KAUNO TECHNOLOGIJOS UNIVERSITETAS

MANTAS LUKAUSKAS

ROBASTINIŲ KLASTERIZAVIMO ALGORITMŲ PLĖTOJIMAS

Daktaro disertacija
Gamtos mokslai, informatika (N 009)

2024, Kaunas

Disertacija rengta 2019–2023 metais Kauno technologijos universiteto Matematikos ir gamtos mokslų fakulteto Taikomosios matematikos katedroje. Mokslinius tyrimus rėmė Lietuvos mokslo taryba.
Doktorantūros teisė Kauno technologijos universitetui suteikta kartu su Vytauto Didžiojo universitetu ir Vilniaus Gedimino technikos universitetu (VILNIUS TECH).

Mokslinis vadovas:
doc. dr. Tomas RUZGAS (Kauno technologijos universitetas, gamtos mokslai, informatika, N 009).

Disertaciją redagavo: anglų kalbos redaktorius dr. Armandas Rumšas (leidykla „Technologija“), lietuvių kalbos redaktorė Rita Malikėnienė (leidykla „Technologija“).

Informatikos mokslo krypties disertacijos gynimo taryba:
prof. habil. dr. Rimantas BARAUSKAS (Kauno technologijos universitetas, gamtos mokslai, informatika, N 009) – **pirmininkas**;
vyresn. m. d., dr. Gražina KORVEL (Vilniaus universitetas, gamtos mokslai, informatika, N 009);
dr. Mantas MIKAITIS (Lidso universitetas, Didžioji Britanija, gamtos mokslai, matematika, N 001);
prof. dr. Alfonsas MISEVIČIUS (Kauno technologijos universitetas, gamtos mokslai, informatika, N 009);
prof. dr. Agnė PAULAUSKAITĖ-TARASEVIČIENĖ (Kauno technologijos universitetas, gamtos mokslai, informatika, N 009).

Disertacija bus ginama viešame Informatikos mokslo krypties disertacijos gynimo tarybos posėdyje 2024 m. rugpjūčio 27 d. 15 val. Kauno technologijos universiteto Rektorato salėje.

Adresas: K. Donelaičio g. 73-402, LT-44249 Kaunas, Lietuva.
Tel. +370 608 28 527; el. paštas doktorantura@ktu.lt

Disertacija išsiųsta 2024 m. liepos 26 d.
Su disertacija galima susipažinti interneto svetainėje http://ktu.edu, Kauno technologijos universiteto bibliotekoje (Gedimino g. 50, LT-44239 Kaunas), Vytauto Didžiojo universiteto bibliotekoje (K. Donelaičio g. 52, LT-44244 Kaunas, Lietuva) ir Vilniaus Gedimino technikos universiteto (VILNIUS TECH) bibliotekoje (Saulėtekio al. 14, LT-10223 Vilnius, Lietuva).

KAUNAS UNIVERSITY OF TECHNOLOGY

MANTAS LUKAUSKAS

DEVELOPMENT OF ROBUST CLUSTERING ALGORITHMS

Doctoral dissertation
Natural Sciences, Informatics (N 009)

2024, Kaunas

This doctoral dissertation was prepared at Kaunas University of Technology, Faculty of Mathematics and Natural Sciences, Department of Applied Mathematics, during the period of 2019–2023. The studies were supported by the Research Council of Lithuania.
The doctoral right has been granted to Kaunas University of Technology together with Vytautas Magnus University and Vilnius Gediminas Technical University (VILNIUS TECH).

Scientific Supervisor:
Assoc. Prof. Dr. Tomas RUZGAS (Kaunas University of Technology, Natural Sciences, Informatics, N 009).

The dissertation was edited by: English language editor Dr. Armandas Rumšas (Publishing House *Technologija*), Lithuanian language editor Rita Malikėnienė (Publishing House *Technologija*).

The public defence of the dissertation will be held at 3 p.m. on 27 August 2024 at the public meeting of Dissertation Defence Board of Informatics Science Field in Rectorate Hall at Kaunas University of Technology.

Address: K. Donelaičio 73-402, LT-44249 Kaunas, Lithuania.
Phone: +370 608 28 527; email: doktorantura@ktu.lt

The doctoral dissertation was sent out on July 26, 2024.
The doctoral dissertation is available on the internet at http://ktu.edu and at the libraries of Kaunas University of Technology (Gedimino 50, LT-44239 Kaunas, Lithuania), Vytautas Magnus University (K. Donelaičio 52, LT-44244 Kaunas, Lithuania) and Vilnius Gediminas Technical University (VILNIUS TECH) (Saulėtekio 14, LT-10223 Vilnius, Lithuania).

TURINYS

LENTELIŲ SĄRAŠAS

PAVEIKSLŲ SĄRAŠAS

SANTRUMPŲ IR TERMINŲ SĄRAŠAS

Santrumpos:

AE – automatinis enkoderis (angl. *autoencoder*)
AE-CBMIDE – automatinio enkoderio klasterizavimas paremtas modifikuotos apvertimo formulės tankio įverčiu (angl. *autoencoder clustering based on the modified inversion formula density estimation*)
AKDE – adaptuotas branduolinis (angl. *adaptive kernel*) tankio įvertis
ANN – dirbtiniai neuroniniai tinklai (angl. *Artificial Neural Networks*)
BIRCH – BIRCH metodas (angl. *Balanced Iterative Reducing and Clustering Using Hierarchies*)
CBMIDE – klasterizavimas paremtas modifikuotos apvertimo formulės tankio įvertiniu (angl. *clustering based on the modified inversion formula density estimation*)
DBSCAN – DBSCAN metodas (angl. *Density Based Spatial Clustering of Applications with Noise*)
DCBMIDE – gilusis klasterizavimas paremtas modifikuotos apvertimo formulės tankio įvertiniu (angl. *deep clustering based on the modified inversion formula density estimation*)
EM – didžiausio tikėtinumo metodas (angl. *Expectation Maximization*)
FCM – neraiškiąja logiką grįstas c vidurkių klasterizavimas (angl. *Fuzzy c-means (FCM)*)
GDBSCAN – apibendrintas DBSCAN metodas (angl. *Generalized Density Based Spatial Clustering of Applications with Noise*)
GMM – Gauso mišinių metodas (angl. *Gaussian mixture model*)
HDBSCAN – hierarchinis DBSCAN metodas (angl. *Hierarchical Density-Based Spatial Clustering of Applications with Noise*)
IFDE – apvertimo formulės (angl. *inversion formula*) tankio įvertis
LSDE – logsplaino (angl. *logspline*) tankio įvertinys
MIDE – modifikuotas apvertimo formulės tankio įvertinys (angl. *modified inversion formula estimator*)
PPDE – tikslinio projektavimo (angl. *projection pursuit*) tankio įvertinys
RCBMIDE – sumažintas klasterizavimas, paremtas modifikuotos apvertimo formulės tankio įvertiniu (angl. *reduced clustering based on the modified inversion formula density estimation*)
SKDE – iš dalies parametrinis branduolinis (angl. *semi-parametric kernel*) tankio įvertinys
SUBClu – tankiu sujungtas poerdvio klasterizavimas (angl. *Density-connected Subspace Clustering*)
TF-IDF (angl. Term Frequency - Invert Document Frequency) – algoritmas, skirtas žodžio svarbai dokumente nustatyti
VAE – variacinis automatinis enkoderis (angl. *variational autoencoder*)
VAE-CBMIDE – automatinio enkoderio klasterizavimas, paremtas modifikuotos apvertimo formulės tankio įvertiniu (angl. *variational autoencoder clustering based on the modified inversion formula density estimation*)

Terminai:

Automatinis koderis (angl. *autoencoder*) – dirbtinio neuroninio tinklo tipas, sudarytas iš dviejų dalių: pirma dalis transformuoja daugiamatės erdvės analizuojamus duomenis į mažesnio matavimo erdvę, antra dalis rekonstruoja pradinius duomenis iš gautų projekcijų.
Dekoderis (angl. *decoder*) – automatinio enkoderio dalis, kurioje atkuriama informacija.
Enkoderis (angl. *encoder*) – automatinio enkoderio dalis, kurioje vyksta informacijos glaudinimas.
Klasteris (angl. *cluster*) – panašiomis charakteristikomis pasižyminčių elementų grupė.
Metodas (angl. *method*) – nurodymų rinkinys, skirtas tam tikrai užduočiai atlikti arba problemai išspręsti. Tai aiški žingsnių seka, padedanti pasiekti norimą rezultatą. Metodas yra abstrakčios sąvokos, kurios nepriklauso nuo konkrečių duomenų ar problemos, kuriai jie taikomi, specifikos.
Modelis (angl. *model*) – kalbant apie mašininį mokymąsi arba statistiką, paprastai yra matematinės arba statistinės struktūros, naudojamos duomenims apibūdinti arba numatyti (prognozuoti). Modelis sukuriamas ir apmokomas naudojant konkrečius duomenis ir galbūt naudojant tam tikrus algoritmus. Modelio tikslas – suvokti, kaip vieni kintamieji yra susiję su kitais ir kaip pagal tuos ryšius galima daryti išvadas arba numatymus apie nežinomus ar ateities duomenis.
Robastinis – savybė, padedanti sistemai arba modeliui išlikti efektyviems ir tiksliems net esant įvairiems duomenų trikdžiams ar klaidoms. Robastiniai algoritmai yra itin vertinami, nes užtikrina stabilumą ir patikimumą net neidealiomis sąlygomis.
Robastiniai algoritmai – tai algoritmai, kurie sukurti tam, kad gerai veiktų net esant duomenų triukšmui ar klaidoms. Šie algoritmai yra itin atsparūs ekstremaliems duomenų taškams (angl. *outliers*) ir kitoms anomalijoms. Tai užtikrina stabilų ir patikimą rezultatyvumą įvairiomis sąlygomis.
Sentimentai – žmonių emocinės reakcijos, nuomonės arba požiūriai, išreikšti tekstuose. Jie gali būti teigiami, neigiami arba neutralūs, ypač vertinami siekiant suprasti viešąją nuomonę arba emocinę reakciją į tam tikrus įvykius, produktus ar paslaugas.
Sentimentų analizė – natūralios kalbos apdorojimo (angl. NLP) technologijų taikymo sritis, kurios tikslas – atpažinti ir interpretuoti emocinį turinį tekste. Sentimentų analizė apima teksto rinkimą, apdorojimą ir klasifikavimą pagal emocinį turinį, siekiant sužinoti, kaip žmonės vertina tam tikrus objektus ar įvykius.

1. ĮVADAS

Darbo aktualumas ir svarba

Besikeičiantys duomenys ir didėjantis duomenų analizės poreikis pabrėžia efektyvių duomenų analizės metodų svarbą [1,2]. Klasterizavimo algoritmai tapo esmine mašininio mokymosi ir duomenų gavybos priemone, padedančia spręsti iššūkius, susijusius su duomenų apdorojimu ir interpretavimu, ypač kai nėra jokios apriorinės informacijos apie duomenis. Klasterizavimas yra procesas, kuriuo duomenys grupuojami remiantis jų charakteristikomis, panašumais arba ryšiais, siekiant sukurti viduje vienalytes ir tarpusavyje skirtingas grupes. Toks duomenų suskirstymas į grupes gali supaprastinti sudėtingas problemas, palengvinti žinių atradimą ir suteikti gilesnių įžvalgų apie pagrindinę duomenų struktūrą [3,4].

Pastaraisiais metais klasterizavimas buvo pritaikytas įvairiose srityse, tokiose kaip bioinformatika [5], vaizdo apdorojimas [6,7], natūralios kalbos apdorojimas [8,9], socialinių tinklų analizė [10] ir anomalijų aptikimas [11]. Klasterizavimo svarba – tai jo gebėjimas padėti lengviau suprasti sudėtingas duomenų struktūras ir atskleisti reikšmingus duomenų modelius. Be to, klasterizavimas sudaro sąlygas kurti pažangesnius ir veiksmingesnius metodus, padedančius spręsti sudėtingas problemas įvairiose srityse, taip pagerinant bendrą tų sričių supratimą.

Moksliniai tyrimai, kuriuos atliekant tiriamas patikimas duomenų klasterizavimas ir jo taikymas, dėmesį skiria realiems duomenims [12–14]. Jie apima tokias problemas kaip triukšmas, ekstremalios reikšmės, išskirtys, trūkstami arba sugadinti duomenys. Realaus pasaulio duomenų rinkiniai gali turėti sudėtingas geometrines klasterių formas, dėl kurių tradiciniai grupavimo metodai gali būti neefektyvūs. Patikimas robastinis klasterizavimas siekia pašalinti šiuos apribojimus taikant pažangius metodus, kurie gali suskirstyti įvairius duomenis ir atskleisti sudėtingas jų struktūras. Siekiant išspręsti triukšmo problemą, plėtojami skirtingi robastiniai duomenų klasterizavimo algoritmai, leidžiantys tiksliai atlikti grupavimą net esant dideliam triukšmo ar išskirtų duomenų kiekiui.

Darbo objektas, subjektas ir matai

Šio tyrimo objektu laikomas heteroskedastinių duomenų aibė, kurią reikia suskirstyti į homogenines grupes. Darbo subjektai yra robastiniai duomenų klasterizavimo algoritmai / metodai, kurie būtų nejautrūs triukšmui ir išskirtims. Šiame darbe robastiniai duomenų klasterizavimo metodai apibrėžiami kaip metodai, nejautrūs triukšmui ir išskirtims. Duomenų klasterizavimo metodų tikslumui palyginti naudojami įvairūs matai: tikslumo matas (angl. *Accuracy*), „JScore“ ir Silueto koeficientas.

Darbo tikslas

Sukurti ir ištirti duomenų klasterizavimo metodus, kurie būtų robastiniai ir efektyvūs, lyginant su kitais šiuo metu taikomais duomenų klasterizavimo metodais naudojant nevienalyčius duomenis, jei duomenyse yra triukšmo ir ekstremalių reikšmių, bei įvertinti metodų efektyvumą, jei triukšmo duomenyse nėra.

Darbo uždaviniai

1. Apžvelgti ir ištirti šiuo metu taikomus duomenų klasterizavimo metodus, jų veikimo principus ir skirtumus bei realaus naudojimo atvejus, juose kylančias problemas, šių metodų modifikacijas, galinčias veikti esant triukšmui ar trūkstamoms reikšmėms.
2. Sudaryti naujus robastinius algoritmus / metodus, skirtus duomenų klasterizavimui atlikti.
3. Atlikti pasiūlytų ir kitų populiarių klasterizavimo algoritmų / metodų lyginamąją analizę.
4. Pritaikyti naujus pasiūlytus klasterizavimo algoritmus / metodus realių duomenų rinkinių atvejais, nustatyti taikymo privalumus ir trūkumus, lyginant su kitais metodais.

Darbo metodika

Disertacijoje taikomi įvairūs tikimybių teorijos, matematinės statistikos, duomenų dimensijų mažinimo, vizualizavimo metodai. Pristatyti klasterizavimo metodai paremti apvertimo formulės tankio įverčiu.

Ginamieji teiginiai

- Pasiūlyti robastiniai duomenų klasterizavimo metodai, paremti apvertimo formulės tankio įverčiu, yra tikslesni, lyginant su šiuo metu taikomais metodais atskirais sintetinių ir realių duomenų atvejais.
- Pasiūlytos robastinių klasterizavimo metodų modifikacijos, pritaikančios duomenų dimensijų mažinimo metodus, su sukurtais duomenų klasterizavimo metodais nuosekliai teikia geresnius rezultatus.

Darbo mokslinis naujumas ir praktinė reikšmė

Sukurti duomenų klasterizavimo algoritmai, paremti apvertimo formulės tankio įverčiu, pratęsia M. Kavaliausko [15] ir T. Ruzgo [16] darbus. Pasiūlyti duomenų klasterizavimo metodai ir tyrimų rezultatai buvo pristatyti mokslinėse konferencijose ir publikuoti moksliniuose straipsniuose. Sukurti duomenų klasterizavimo algoritmai buvo panaudoti mokslinių projektų metu ir praktinėje įmonės UAB „Hostinger International“ veikloje. Duomenų klasterizavimo taikymas praktinėje veikloje leido sudaryti verslui svarbias duomenų grupes, jas interpretuoti ir pritaikyti įmonės procesuose. Moksliniuose projektuose duomenų klasterizavimas buvo pritaikytas su ekonominiais duomenimis ir yra realizuotas sukurtoje komercinėje platformoje.

Šiame darbe pristatytus naujus duomenų klasterizavimo metodus ir jų taikymo rekomendacijas savo tyrimuose cituoja ir naudoja: Farmer ir kt. (2023) [17], Powroźnik ir kt. (2022) [18], Yu ir kt. (2023) [19], Chen ir kt. (2023) [20], Gebrael ir kt. (2023) [21], Wei ir Song (2023) [22], Eltayeb ir kt. (2023) [23], Lin ir kt. (2023) [24], Roeksiri ir kt. (2023) [25], Leon ir kt. (2024) [26], Uskenov ir kt. (2024) [27] , Nurduhan ir Kuleyin (2024) [28], Senger ir kt. (2024) [29], Chweidan ir kt. (2024) [30], Goldshein ir kt. (2024) [31].

Darbo rezultatų aprobavimas

Šios disertacijos tema publikuoti penki moksliniai straipsniai periodiniuose leidiniuose su *Web of Science* citavimo indeksu. Trijuose moksliniuose straipsniuose pristatomi nauji klasterizavimo metodai, dviejuose pateikiami sukurtų klasterizavimo metodų praktinio taikymo pavyzdžiai. Duomenų klasterizavimo metodai publikuoti dar trijuose moksliniuose straipsniuose su citavimo indeksu, kurie nėra įtraukiami į šią daktaro disertaciją kaip jos pagrindas.

Atliktų tyrimų rezultatai taip pat pristatyti 12-oje tarptautinių ir nacionalinių mokslinių konferencijų:

1. M. Lukauskas ir T. Ruzgas. A review of clustering algorithms and application // 9th (online) international conference on applied analysis and mathematical modeling (ICAAMM21) June 11-13, 2021, Istanbul-Turkey:
2. M. Lukauskas ir T. Ruzgas. Bank credit card default classification based on clustering using machine learning algorithms // 9th world sustainability forum, virtual, Switzerland, 13–15 September 2021
3. J. Bruneckienė, V. Varaniūtė, L. Dagilienė ir M. Lukauskas. What profiles of circular economy implementation strategies dominate in advanced small open economies? // 2021 IEEE international conference on technology and entrepreneurship (ICTE) "Leading digital transformation in business and society"
4. M. Lukauskas ir T. Ruzgas. Data clustering and its applications in medicine // Online international symposium on applied mathematics and engineering (ISAME22), January 21-23, 2022, Istanbul-Turkey
5. M. Lukauskas ir T. Ruzgas. Mixtures models for clustering: review and comparison // 10th (online) international conference on applied analysis and mathematical modeling (ICAAMM22), July 1-3, 2022, Istanbul, Turkey
6. M. Lukauskas ir T. Ruzgas. Analysis of clustering methods performance across multiple datasets // DAMSS 2021: 12th conference on data analysis methods for software systems, Druskininkai, Lithuania, December 2–4, 2021
7. M. Lukauskas ir T. Ruzgas. Data clustering based on the modified inversion formula density estimation // DAMSS 2022: 13th conference on data analysis methods for software systems, Druskininkai, Lithuania, December 1–3, 2022
8. M. Lukauskas, V. Pilinkienė, J. Bruneckienė, A. Stundžienė, A. Grybauskas ir T. Ruzgas. Big data processing system for Lithuania economic activity nowcasting // DAMSS 2022: 13th conference on data analysis methods for software systems, Druskininkai, Lithuania, December 1–3, 2022
9. M. Lukauskas ir T. Ruzgas. Review and comparative analysis of unsupervised machine learning application in health care // Data intelligence and cognitive informatics: proceedings of ICDICI 2022.

10. M. Lukauskas, V. Pilinkienė, J. Bruneckienė, A. Stundžienė ir A.Grybauskas. Automated system and machine learning application in economic activity monitoring and nowcasting // Information and software technologies: 28th international conference, ICIST 2022, Kaunas, Lithuania, October 13–15, 2022
11. M. Lukauskas. *Comparative analysis of clustering algorithms for synthetic and real data*. 7th International Conference on Machine Learning, Optimization & Data Science - LOD 2021.
12. M. Lukauskas, V. Pilinkienė, J. Bruneckienė, A. Stundžienė, A. Grybauskas ir T. Ruzgas. Evaluation of news sentiment in economic activity forecasting. 3rd International Electronic Conference on Applied Sciences session Computing and Artificial Intelligence, 2022.

Tyrimo rezultatai taip pat pristatyti Lietuvos statistikų sąjungos organizuojamame seminare.

Įtraukti straipsniai ir bendraautorių indėlis į darbus

Pateikiamos penkių toliau išvardytų straipsnių, kurie yra šios disertacijos pagrindas, kopijos. Straipsniai yra atvirosios prieigos (angl. *Open Access*).

Straipsnis 1. T. Ruzgas, **M. Lukauskas**, and G. Čepkauskas. 2021. “Nonparametric Multivariate Density Estimation: Case Study of Cauchy Mixture Model" Mathematics 9, no. 21: 2717. https://doi.org/10.3390/math9212717

Straipsnis 2. **M. Lukauskas** ir T. Ruzgas. 2022. “A New Clustering Method Based on the Inversion Formula” Mathematics 10, no. 15: 2559. https://doi.org/10.3390/math10152559.

Straipsnis 3. **M. Lukauskas** ir T. Ruzgas. 2023. “Reduced Clustering Method Based on the Inversion Formula Density Estimation” Mathematics 11, no. 3: 661. https://doi.org/10.3390/math11030661

Straipsnis 4. **M. Lukauskas**, V. Pilinkienė, J. Bruneckienė, A. Stundžienė, A. Grybauskas ir T. Ruzgas. 2022. “Economic Activity Forecasting Based on the Sentiment Analysis of News” Mathematics 10, no. 19: 3461. https://doi.org/10.3390/math10193461

Straipsnis 5. **M. Lukauskas**, V. Šarkauskaitė, V. Pilinkienė, A. Stundžienė, A. Grybauskas ir J. Bruneckienė. “Enhancing Skills Demand Understanding through Job Ad Segmentation using NLP and Clustering Techniques” Applied Sciences. 2023; 13(10):6119. https://doi.org/10.3390/app13106119

Toliau išvardyti straipsniai ir disertacijos autoriaus indėlis į šiuos straipsnius:

1) Straipsnis 1. T. Ruzgas, **M. Lukauskas**, ir G. Čepkauskas. Nonparametric Multivariate Density Estimation: Case Study of Cauchy Mixture Model.

- M. Lukauskas – straipsnio pateikimo kontaktinis asmuo (angl. *Correspondence author*), idėjos konceptualizavimas, reikalingų programinių priemonių rengimas, duomenų analizė ir jų tyrimas, pradinio straipsnio šablono rašymas, taisymų atlikimas po recenzentų ir redaktoriaus komentarų.

2) Straipsnis 2. **M. Lukauskas** ir T. Ruzgas. A New Clustering Method Based on the Inversion Formula.
    - M. Lukauskas – pirmas ir kontaktinis straipsnio asmuo, idėjos konceptualizavimas, metodologijos sudarymas, programinės įrangos, reikalingos eksperimentams atlikti, parengimas, duomenų analizė ir tyrimas, pradinio straipsnio rankraščio rengimas, taisymų atlikimas atsižvelgiant į recenzentų ir redaktoriaus komentarus.
3) Straipsnis 3. **M. Lukauskas** ir T. Ruzgas. Reduced Clustering Method Based on the Inversion Formula Density Estimation.
    - M. Lukauskas – pirmas ir kontaktinis straipsnio asmuo, idėjos konceptualizavimas, metodologijos sudarymas, programinės įrangos, reikalingos eksperimentams atlikiti, parengimas, duomenų analizė ir tyrimas, pradinio straipsnio rankraščio rengimas, taisymų atlikimas atsižvelgiant į recenzentų ir redaktoriaus komentarus.
4) Straipsnis 4. **M. Lukauskas**, V. Pilinkienė, J. Bruneckienė, A. Stundžienė, A. Grybauskas, ir T. Ruzgas. Economic Activity Forecasting Based on the Sentiment Analysis of News.
    - M. Lukauskas – pirmas ir kontaktinis straipsnio asmuo, idėjos konceptualizavimas, metodologijos sudarymas, programinės įrangos, reikalingos eksperimentų atlikimui parengimas, duomenų analizė ir tyrimas, pradinio straipsnio rankraščio rengimas, pataisymų vykdymas atsižvelgiant į recenzentų ir redaktoriaus komentarus.
5) Straipsnis 5. **M. Lukauskas**, V. Šarkauskaitė, V. Pilinkienė, A. Stundžienė, A. Grybauskas ir J. Bruneckienė. Enhancing Skills Demand Understanding through Job Ad Segmentation using NLP and Clustering Techniques.
    - M. Lukauskas – pirmas ir kontaktinis straipsnio asmuo, idėjos konceptualizavimas, metodologijos sudarymas, programinės įrangos, reikalingos eksperimentams atlikti, parengimas, duomenų analizė ir tyrimas, pradinio straipsnio rankraščio rengimas, pataisymų atlikimas atsižvelgiant į recenzentų ir redaktoriaus komentarus.

2. LITERATŪROS APŽVALGA

Šio mokslinio darbo dalyje nagrinėjamas tyrimo objektas – duomenų klasterizavimas, pateikiant informaciją apie taikomus metodus, jų pranašumus ir trūkumus. Skyriuje aptariamas duomenų klasterizavimo taikymas praktikoje, remiantis straipsniais, kuriuose siekiama pritaikyti naujai kuriamus klasterizavimo metodus. Skyrius baigiamas pagrindinių mokslinės literatūros analizės išvadų apžvalga.

2.1. Duomenų klasterizavimas ir jo metodai

Dirbtinis intelektas pirmą kartą buvo paminėtas dar 1956 m. [32], tačiau ryškiausias jo naudojimo šuolis pastebimas tik per paskutinį dešimtmetį [33–35]. Nuolat didėjanti kompiuterių skaičiavimo galia skatina vis didesnį dirbtinio intelekto prieinamumą, plėtrą ir pritaikymą įvairiose praktikos / mokslo srityse. Viena iš dirbtinio intelekto sričių, žinoma plačiausiai, yra mašininis mokymasis, kuriuo siekiama sukurti skirtingus modelius naudojant turimus duomenis. Vienas iš mašininio mokymosi tipų – neprižiūrimas mokymasis, kurio pagrindinė sritis yra duomenų klasterizavimas. Duomenų klasterizavimo tikslas – suskirstyti turimus stebinius į kuo geriau atskiriamas grupes / klases, taip atrandant vidinę duomenų rinkinio objektų / stebinių struktūrą ir ryšius tarp jų. Prižiūrimo mašininio mokymosi atveju / klasifikavimo atveju apmokant modelius iš anksto yra žinomos klasės ir siekiama, kad modelis kuo geriau prisitaikytų prie šių klasių ir galėtų naujus duomenis suskirstyti į šias klases. Klasterinės analizės tikslas – suskirstyti panašius elementus / stebinius į atskiras grupes, įvertinant stebinių panašumo laipsnį. Klasterizuojant duomenis tikimasi, kad to paties klasterio elementai bus kaip įmanoma panašesni vienas į kitą, o skirtingų grupių elementai bus kiek įmanoma skirtingesni [36]. Matematinė uždavinio formulė gali būti apibrėžta taip: turint duomenų rinkinį $X=\{x_1, x_2, ..., x_n\}$, čia x_i – duomenų taškas i-tojoje pozicijoje, klasterizavimo uždavinys yra priskirti kiekvieną x_i tam tikram klasteriui C_j taip, kad panašūs taškai būtų tame pačiame klasteryje ir kuo arčiau klasterio centro. Siekiamas kriterijus (tikslo funkcija) priklauso nuo pasirinkto klasterizavimo algoritmo, tačiau dažnai tai funkcija, kurią siekiama minimizuoti arba maksimizuoti. Pavyzdžiui, klasterizuojant *K* vidurkius (angl. *k-means*), tikslo funkcija gali būti suma kvadratinių atstumų tarp klasterių centrų ir jiems priklausančių taškų. Tai reiškia, kad algoritmas sieks surasti tokį klasterių centrų rinkinį, kad ši suma būtų kuo mažesnė.

Duomenų klasterizavimo sąvokos atsiradimas siejamas su įvairiomis mokslo disciplinomis, tokiomis kaip biologija, statistika ir psichologija. Klasterinės analizės istoriją galima atsekti iki žymaus britų biologo sero Ronaldo Fisherio, kuris 1936 m. pristatė vilkdalgių (angl. *iris*) duomenų rinkinį, kuris vėliau tapo įvairių klasterizavimo algoritmų kūrimo pagrindu. Bėgant metams grupavimas tapo svarbia analitikos priemone, ypač duomenų tyrybos, mašininio mokymosi ir modelių atpažinimo srityse. Klasterinė analizė yra nepaprastai svarbi akademiniams tyrimams, nes leidžia mokslininkams atskleisti reikšmingus modelius ir tendencijas dideliuose ir sudėtinguose duomenų rinkiniuose [37–39]. Grupuojant panašius duomenų taškus pagal jiems būdingas savybes, grupavimas padeda identifikuoti pagrindines duomenų

struktūras, ryšius ir asociacijas, o tai galiausiai palengvina žinių atskleidimo procesą. Dėl to klasterių analizė yra plačiai pritaikoma įvairiose srityse, tokiose kaip rinkodara [40], finansai [38], medicina [41] ir socialiniai mokslai [42], prisidėdama prie labiau informuotų, duomenimis pagrįstų sprendimų priėmimo paradigmų kūrimo [10, 43].

Norint praktiškai pritaikyti duomenų klasterizavimą, buvo pasiūlyta nemažai klasterizavimo metodų, kuriuos galima suskirstyti į penkias pagrindines grupes: dalijimo metodai, hierarchiniai metodai, tankiu grįsti metodai, tinkleliu grįsti metodai ir modeliais grįsti metodai (angl. *model-based clustering*). Šiame poskyryje pateikiame išsamų kiekvienos duomenų klasterizavimo metodų kategorijos paaiškinimą ir populiariausius šios kategorijos klasterizavimo metodus, jų veikimą, apžvelgtus pranašumus ir trūkumus.

Nepaisant to, kad yra daug skirtingų duomenų klasterizavimo metodų, klasterizavimas ir toliau lieka svarbia duomenų mokslo sritimi, kuri nuolatos tobulinama. Labai svarbu kurti naujus metodus, kurie būtų pritaikyti prie šiuo metu esančių duomenų, gebėtų susitvarkyti su kylančiais šių duomenų iššūkiais. Vienas didžiausių duomenų klasterizavimo iššūkių – nuolat didėjantys duomenų kiekiai bei jų sudėtingumas, dėl ko tampa svarbu turėti metodus, pasižyminčius greitaveika. Kitas svarbus veiksnys yra triukšmas, nuokrypiai, išskirtys duomenyse, kuo labai dažnai pasižymi praktikoje naudojami duomenys. Todėl ypač svarbu kurti patikimus robastinius duomenų klasterizavimo metodus, kurie galėtų veiksmingai pašalinti tokius trūkumus. Be to, didėjanti duomenų tipų, pvz., teksto, vaizdų ir grafikų, įvairovė reikalauja universalesnių klasterizavimo metodų, galinčių tvarkyti skirtingas duomenų struktūras ir atvaizdus.

Nė vienas duomenų klasterizavimo metodas nėra visuotinai veiksmingas visų tipų duomenų rinkiniams ar net atskiriems duomenų rinkiniams, todėl tai dar viena priežastis, dėl kurios klasterizavimo metodai nuolatos tobulinami. Kiekvienas klasterizavimo metodas turi išskirtinių savybių, privalumų ir trūkumų. Kai kurie duomenų klasterizavimo metodai veikia gerai tik su atskiromis sferinėmis duomenų grupėmis, tačiau prastai veikia su iš dalies sutampančiais ar kitokios formos klasteriais. Tai rodo, kad duomenų klasterizavimo metodo parinkimas dažniausiai priklauso nuo duomenų rinkinio ir pagrindinių jo savybių. Be to, tolesnis duomenų klasterizavimo metodų tobulinimas prisideda ir prie kitų duomenų mokslo bei dirbtinio intelekto sričių tobulėjimo, nes taikant nemažą dalį įvairių metodų naudojamas ir duomenų klasterizavimas. Toliau pateikiama informacija apie skirtingas duomenų klasterizavimo grupes ir kokie metodai priskiriami tai grupei.

2.1.1. Dalijimo metodai

Dalijimo metodais siekiama suskaidyti duomenis į iš anksto nustatytą skaičių nesutampančių grupių (K), kad kiekvienas objektas priklausytų tiksliai vienai jų. k vidurkių (angl. *k-means*) duomenų klasterizavimo metodas yra vienas iš populiariausių duomenų klasterizavimo metodų. Šis duomenų klasterizavimo metodas pirmą kartą buvo pasiūlytas Stuarto Lloydo 1957 m. [44], o vėliau jį patobulino J. B. MacQueenas 1967 m. [45]. Tai standartinis duomenų dalijimo metodas, kuriuo siekiama visą duomenų aibę suskirstyti į k skirtingų grupių.

Kiekvienas duomenų taškas priklauso vienam klasteriui. Per daugelį metų buvo pasiūlyta keletas modifikacijų ir patobulinimų, siekiant pagerinti *k* vidurkių algoritmo veikimą ir pašalinti jo ribotumus. Pavyzdžiui, *k* vidurkių ++ inicijavimo metodas, kurį 2007 m. pristatė Davidas Arthuras ir Sergejus Vassilvitskis, pagerina pradinių centroidų pasirinkimą, paskirstydamas juos tolygiau visoje duomenų erdvėje, taip sumažindamas konvergavimo į lokalų minimumą tikimybę [46]. *K* vidurkių algoritmo skaičiavimo laiko problemą ir jo pritaikymą dideliems duomenų kiekiams išsprendė Bahmani et al. (2012). Jis pristatė lygiagretaus inicijavimo metodą (angl. *scalable k-means++*), leidžiantį efektyviai apdoroti didelius duomenų rinkinius [47]. Mažo paketo *k* vidurkių (*angl. mini-batch*) variantas, kurį pasiūlė Sculley (2010), kiekvienoje iteracijoje naudoja atsitiktinę imtį arba mini duomenų taškų paketą, todėl sumažėja skaičiavimo sąnaudos ir užtikrinamas greitesnis konvergavimas, ypač dideliems duomenų rinkiniams [48]. Neraiškiąja logiką grįstas *C* vidurkių (angl. *Fuzzy C-means,* FCM) metodas leidžia duomenų taškams priklausyti keliems klasteriams, turintiems skirtingą narystės laipsnį kiekviename iš jų ir vaizduojamą narystės matrica [49]. FCM algoritmas naudoja neapibrėžtumo parametrą, skirtą klasterio sutapimo lygiui valdyti, ir yra ypač naudingas dirbant su duomenimis, kuriems būdingas neapibrėžtumas ir sutapimas. Taip pat galima rasti ir kitų *k* vidurkių klasterizavimo metodų modifikacijų, pavyzdžiui, apribotas *k* vidurkis (*angl. Constrained K-means*), kuriame į *k* vidurkių algoritmą įtraukiami papildomi apribojimai, siekiant užtikrinti konkrečius reikalavimus, pvz., klasterio dydį, klasterio formą arba porinius duomenų taškų apribojimus [50]. Apribotas *k* vidurkių metodas gali būti naudingas tais atvejais, kai grupavimo procese reikia atsižvelgti į išankstines žinias arba specifinius srities apribojimus. Sferinis *k* vidurkių metodas, taip pat žinomas kaip *k* vidurkio metodas su kosinuso panašumu, naudoja kosinuso panašumo matą, o ne Euklido atstumą, kad palygintų duomenų taškus. Sferinis *k* vidurkių metodas veikia normalizuotuose duomenų taškuose, kurie turi būti ant hipersferos paviršiaus. Dėl šio apribojimo algoritmas yra atsparesnis reikšmingiems duomenų taškų tankio skirtumams [51, 52]. Jis tinka didelės apimties duomenims, pvz., tekstiniams dokumentams ar genų profiliams, kur kampas tarp duomenų taškų yra reikšmingesnis nei tikrasis atstumas. Viena iš robastinių modifikacijų yra apkarpytas *k* vidurkių metodas (angl. *Trimmed K-means,* TKM), kurį pristatė García-Escudero ir kt. 2008 m. [53]. Pagal šį metodą apskaičiuojant naujų centro taškų padėtį, tam tikra procentinė dalis nutolusių duomenų taškų iš kiekvieno klasterio centroido neįtraukiama. Šis apkarpymo procesas padeda sumažinti galimų nuokrypių įtaką klasterio centroidams. Pagrindinis apkarpyto *k* vidurkių algoritmo pranašumas yra jo gebėjimas sukurti patikimesnius klasterizavimo rezultatus, esant triukšmui ar išskirtims, todėl šis metodas laikomas robastine *k* vidurkių metodo modifikacija. Tačiau taikant metodą reikia nurodyti apkarpytinų duomenų taškų proporciją, kuri gali būti nežinoma iš anksto ir sudėtinga įvertinti. Paskutiniais metais Dorabiala ir kt. (2022) [54] pristatė dar vieną patobulintą robastinį apkarpytą *k* vidurkių metodą (angl. *Robust trimmed k-means,* RTKM). Robastinis apkarpytas *k* vidurkių (RTKM) metodas yra tradicinio *k* vidurkių klasterizavimo metodo plėtinys, kuriuo siekiama pagerinti jo patikimumą. Skirtingai nuo kitų metodų, kurių specializuoja – vieno tipo duomenys, RTKM sukurtas taip, kad būtų universalesnis ir veiktų tiksliau įvairiais

atvejais. Šis algoritmas nustato išskirčių taškus, todėl yra tinkamas įvairiai taikyti praktikoje. Eksperimentai su įvairiais realaus pasaulio duomenų rinkiniais rodo, kad RTKM veikia konkurencingai, palyginti su kitais metodais [54]. Be to, RTKM pranoksta kitus metodus dirbant su negriežtos klasterizacijos duomenimis, nes išnaudoja santykinius taškų pranašumus. Kitas robastinis klasterizavimo metodas yra *k* vidurkių metodas pašalinant išskirtis (angl. *K-means with Outlier Removal,* KMOR). Šis metodas buvo sukurtas kaip standartinio *k* vidurkių klasterizavimo metodo variantas, siekiant išspręsti iššūkį, susijusį su išskirčių aptikimu ir pašalinimu klasterizavimo proceso metu. Pagrindinis KMOR tikslas – pagerinti *k* vidurkių metodo atsparumą ir našumą pakartotinai aptinkant ir šalinant galimas išskirtis, kurios kitaip galėtų neigiamai paveikti klasterizavimo rezultatus [55]. KMOR algoritmas veikia identifikuodamas galimas išskirtis pagal jų atstumą nuo centroidų per kiekvieną *k* vidurkių metodo iteraciją. Šios galimos išskirtys yra laikinai pašalinamos iš duomenų rinkinio, leidžiant *k* vidurkių metodui tęsti iteracijas su likusiais stebiniais. Šis procesas kartojamas tol, kol pasiekiamas konvergavimas, todėl gaunamas patikimesnis klasterizavimo sprendinys. Vienas pagrindinių KMOR metodo privalumų – jo robastiškumas ir gebėjimas efektyviai tvarkyti triukšmingus duomenis arba duomenų rinkinius, kuriuose yra daug išskirčių. Identifikuodamas ir šalindamas kraštutinius taškus, KMOR padeda išvengti centroidų iškraipymo dėl šių nuokrypių, o tai gali lemti geresnius klasterizavimo rezultatus. Tačiau yra ir tam tikrų galimų KMOR metodo trūkumų. Vienas pagrindinių iššūkių – tinkamo nukrypimų aptikimo slenksčio nustatymas, kuris gali turėti didelę įtaką algoritmo efektyvumui. Jei riba yra per griežta, gali būti pašalinta per daug taškų, todėl prarandama vertinga informacija. Kita vertus, per švelnus slenkstis gali leisti faktiniams nuokrypiams likti neaptiktiems, todėl bus sumenkintas algoritmo patikimumas. Be to, taikant KMOR metodą gali prireikti daugiau skaičiavimo išteklių ir laiko dėl papildomo apdorojimo pašalinant išskirtis.

Kitas pasiūlytas robastinis klasterizavimo metodas, kuris yra *k* vidurkių metodo modifikacija yra NEO *k* vidurkių (angl. *NEO-K-Means (Non-Exhaustive Overlapping K-means*) [56]. Šiame darbe autoriai pasiūlė naują tikslo funkciją. Ji papildo tradicinę *k* vidurkių tikslo formuluotę, naudojant lengvai suprantamus parametrus, atspindinčius sutapimo ir neišsamumo (angl. *non-exhaustiveness*) laipsnius. Metodas pasižymi gerais rezultatais esant įvairiam sutapimo laipsniui. Gauti tyrimo rezultatai rodo geresnius rezultatus lyginant su kitais metodais [56].

2.1.2. Tankiu grįsti metodai

Kaip minėta, nors dalijimo metodai yra paprasčiausi ir lengviausiai pritaikomi, tačiau nemažas dėmesys taip pat skiriamas tankiu grįstiems metodams. Jie grupes identifikuoja pagal vietinį stebinių tankį duomenų erdvėje. Šiuo atveju klasteris apibrėžiamas kaip tankus objektų regionas, kurį supa mažesnio tankio objektų sritys. Vienas žinomiausių tankiu grįstų algoritmų yra DBSCAN (angl. *Density-Based Spatial Clustering of Applications with Noise*). Jis veiksmingai identifikuoja įvairių formų bei dydžių grupes, yra atsparus triukšmui ir išskirtims, todėl laikomas robastiniu klasterizavimo metodu. Kitas populiarus algoritmas – OPTICS (angl.

Ordering Points to Identify the Clustering Structure), kuris papildo DBSCAN, sukurdamas hierarchinį duomenų atvaizdavimą. Tankiu grįsti metodai tinka duomenų rinkiniams su triukšmu ir įvairios formos klasteriams. Tankiu grįstas klasterizavimas gali užtrukti ilgą laiką, ypač kai naudojami dideli duomenų rinkiniai. DBSCAN – tankiu grįstas erdvinis klasterizavimas su triukšmu (angl. *Density-Based Spatial Clustering of Applications with Noise*) [57,58]. DBSCAN yra duomenų klasterizavimo metodas, kurį 1996 m. pasiūlė Martinas Ester, Hans-Peter Kriegel, Jörg Sander ir Xiaowei Xu [57]. DBSCAN yra vienas labiausiai paplitusių ir dažniausiai cituojamų klasterizavimo metodų mokslinėje literatūroje. Pagrindinis DBSCAN tikslas – identifikuoti duomenų rinkinio grupes, remiantis taškų tankiu nustatytame spindulyje. Skirtingai nuo kitų klasterizavimo metodų, pvz., *k* vidurkių, DBSCAN nereikia iš anksto nurodyti klasterių skaičiaus, todėl jis labiau prisitaiko prie pagrindinės duomenų struktūros. Šio klasterizavimo metodo algoritmas abstrakčiai gali būti aprašomas trimis žingsniais. Pirmiausia, kiekvienam taškui nustatomi kaimyniniai taškai, atitinkantys minimalų atstumą, o vėliau identifikuojami pagrindiniai taškai, turintys daugiau nei minimalų kaimynų skaičių. Tada identifikuojami taškai, sujungti su pagrindiniais taškais, ir visi likę ne pagrindiniai taškai priskiriami klasteriams, jei tenkinama minimalaus atstumo sąlyga. Jeigu ši sąlyga nėra tenkinama, toks taškas priskiriamas triukšmo arba išskirčių klasteriui. Kadangi šis metodas veikia šiuo būdu, jis laikomas robastiniu duomenų klasterizavimo metodu. Kaip ir kiti metodai, DBSCAN taip pat turi tam tikrų apribojimų. Kaip ir kiekviename duomenų analizės metode, šiame taip pat viena pagrindinių problemų – tinkamas skirtingų parametrų parinkimas. Algoritmas labai priklauso nuo atitinkamų parametrų pasirinkimo, būtent spindulio ilgio *eps* ir minimalaus taškų skaičiaus. Tik sėkmingas parametrų parinkimas užtikrina, kad bus gauti tinkami klasterizavimo rezultatai. Parinkus netinkamas šių parametrų vertes, klasterizavimo rezultatai gali būti itin prasti. DBSCAN metodas turi ne vieną modifikaciją, kurios leido šiam metodui dar labiau išpopuliarėti ir pašalinti šio metodo trūkumus. Tie patys autoriai kiek vėliau pasiūlė ir apibendrintą DBSCAN metodą (GDBSCAN) [59, 60], ir OPTICS. DBSCAN taip pat naudojamas kaip poerdvio klasterizavimo metodų, tokių kaip PreDeCon ir SUBCLU, dalis. OPTICS (angl. *Ordering Points To Identify the Clustering Structure*) metodas yra glaudžiai susijęs su DBSCAN metodu, kuris taip pat randa didelio tankio pagrindinį klasterį ir tuomet išplečia šiuos klasterius [61]. OPTICS, pasiūlytas Ankerst ir kt. 1999 m., yra DBSCAN plėtinys, sprendžiantis tinkamų parametrų parinkimo problemą [61]. Skirtingai nei DBSCAN šis metodas pasižymi tuo, kad išlaiko hierarchinius stebinių ryšius. Teigiama, kad šis metodas geriau tinkamas naudoti dideliems duomenų rinkiniams lyginant su DBSCAN [62]. HDBSCAN (hierarchinis DBSCAN): HDBSCAN, kurį pristatė Campello ir kt. 2013 m., yra hierarchinis klasterizavimo algoritmas, pagrįstas DBSCAN [63]. Šis algoritmas verčia DBSCAN procedūrą į hierarchinį klasterizavimą, formuojant minimalų apimantį medį, grindžiamą duomenų taškų tankiu (angl. *minimum spanning tree,* MST). Keičiant tankio slenkstį, sukuriama klasterių hierarchija, leidžianti parinkti tinkamiausią duomenų klasterizavimo struktūrą. HDBSCAN gali automatiškai nustatyti klasterių skaičių ir yra patikimesnis parametrų pasirinkimui, lyginant su pradiniu DBSCAN. HDBSCAN [64] yra

hierarchinė DBSCAN versija, kuri taip pat yra greitesnė už OPTICS, iš kurios hierarchijos galima išskirti plokščią skaidinį, susidedantį iš ryškiausių grupių. DBSCAN++ yra dar vienas iš naujesnių duomenų klasterizavimo metodų, kurį pristatė Jennifer Jang ir Heinrichas Jiangas [65]. Šis metodas pagrįstas pastebėjimu, kad tik poaibiui (*m*) visų duomenų taškų (*n*) reikia apskaičiuoti jų tankį, kad būtų galima efektyviai suskirstyti į grupes. DBSCAN++ siūlo dvi paprastas šių *m* taškų atrankos strategijas: tolydžią ir godžią *K* centrais (angl. *greedy k-centers*) pagrįstą atranką. Esant blogiausio atvejo vykdymo laikui O(mn) [66], DBSCAN++ geba išlaikyti aukštus vertinimo rezultatus, kartu greičiau atliekant skaičiavimus. MR-DBSCAN (angl. *MapReduce DBSCAN*), kurį pasiūlė He ir kt. 2011 m., yra lygiagreti ir paskirstyta DBSCAN versija, skirta didelio masto duomenų rinkiniams tvarkyti naudojant „MapReduce“ programavimo modelį [67]. Paskirstydamas klasterizavimo užduotis keliuose mazguose, MR-DBSCAN gali efektyviai apdoroti didžiulius duomenų rinkinius. MR-DBSCAN skaičiavimo sudėtingumas priklauso nuo mazgų skaičiaus ir naudojamos skaidymo strategijos. Esant dideliems duomenų rinkiniams, šis metodas veikia geriau nei DBSCAN. ST-DBSCAN (angl. *Spatial and Temporal DBSCAN*), kurį 2007 m. pasiūlė Birant ir Kut, yra DBSCAN plėtinys, skirtas erdviniams ir laiko duomenims [68]. Apibrėždamas duomenų taškų kaimynystę ir tankį, algoritmas atsižvelgia ir į erdvinius, ir į laiko matmenis. Dėl šios modifikacijos ST-DBSCAN yra tinkamas tokiems taikymams, kaip trajektorijų analizė, įvykių nustatymas laike ir erdvės bei laiko duomenų tyryba. Tačiau tankiu grįsti duomenų klasterizavimo metodai neapsiriboja tik DBSCAN ir jo modifikacijomis. Vienas dažniausių tankiu grįstų metodų DENCLUE (angl. *Density-Connected Clustering*). DENCLUE, kurį 1998 m. pasiūlė Hinneburgas ir Keimas, yra dar vienas tankiu pagrįstas klasterizavimo metodas, kuris naudoja branduolio tankio įvertinimo metodą pagrindiniam duomenų pasiskirstymui modeliuoti [69]. Skirtingai nuo DBSCAN, kuris remiasi fiksuotu kaimynystės spinduliu, DENCLUE įvertina kiekvieno duomenų taško tankį, atsižvelgdamas į jo kaimynų įtaką, naudodamas Gauso branduolio funkciją. Algoritmas identifikuoja grupes kaip didelio tankio regionus, atskirtus mažesnio tankio sritimis. Pagrindinis DENCLUE pranašumas – galimybė efektyviau nei DBSCAN apdoroti įvairaus tankio ir triukšmo lygio duomenų rinkinius, todėl galima teigti, kad jis pasižymi geresniu robastiškumu.

2.1.3. Hierarchinis duomenų klasterizavimas

Kita duomenų klasterizavimo metodų grupė – hierarchinis duomenų klasterizavimas. Hierarchiniai klasterizavimo metodai sukuria į medį panašią klasterių struktūrą, kurią galima vizualizuoti kaip dendrogramą. Klasteriai formuojami iš viršaus į apačią arba iš apačios į viršų, kad būtų sukurta dendrograma, vaizduojanti sudarytų klasterių hierarchinę struktūrą [70]. Hierarchinį duomenų klasterizavimą galima suskirstyti į du tipus – aglomeracinį / jungiamąjį ir dalijamąjį. Aglomeraciniai / jungiamieji metodai pradedami nuo kiekvieno objekto kaip pavienio klasterio ir kartotiniu būdu sujungiamos artimiausios klasterių poros. Tai vykdoma tol, kol visi objektai priklauso vienam klasteriui arba yra tenkinamas nustatytas sustabdymo kriterijus. Dalijimo (iš viršaus į apačią) metodai prasideda nuo vieno

klasterio, kuriame yra visi objektai, ir rekursyviai atliekamas grupių skaidymas į mažesnes. Hierarchiniams metodams nereikia iš anksto nustatyto klasterių skaičiaus. Tačiau jie yra brangūs skaičiavimo požiūriu ir jautrūs panašumo matavimo ir susiejimo kriterijų pasirinkimui. Klasterizuojant hierarchiją taško poaibiai jungiami arba dalijami apibendrinus atstumą tarp atskirų taškų į atstumą tarp taško pogrupių. Tai nustatoma naudojant artumo matą, vadinamą susiejimo metrika. Hierarchiniam klasterizavimui naudojamos trys pagrindinės sąsajų metrikos: vienetinis ryšys (angl. *single linkage*), vidutinis ryšys (angl. *average linkage*) ir pilnasis ryšys (angl. *complete linkage*) [70–73]. Hierarchinis klasterizavimo algoritmas naudoja $N \times N$ jungiamumo matricos formą, kurioje sukonstruojama klasterizuoti naudojama susiejimo metrika. Panašumo matricos konstrukcija pasiekiama ieškant panašumo tarp kiekvienos duomenų taškų poros. Tada sujungimo kriterijus apskaičiuojamas randant porinį atstumą tarp grupių. Panašumo metrika naudojama norint nustatyti atstumą tarp klasterių aibių. Jis taip pat naudojamas nustatant klasterių formą. Vienetinės jungties klasterizavimas taip pat vadinamas artimiausio kaimyno metodu. Jis matuoja artimiausią atstumą nuo bet kurio klasterio nario iki bet kurio kito klasterio nario. Jis matuoja dviejų grupių panašumą matuodamas artimiausią atstumą tarp vienos elementų poros. Vienetinės jungties klasterizavimas turi grandininį efektą, dėl kurio susidaro pailgi klasteriai [74]. Vidutinis susiejimo klasterizavimas taip pat laikomas minimalios dispersijos ryšiu [70, 75]. Jis nustato atstumų tarp visų duomenų taškų tarp grupių vidurkį arba medianą [74]. Pilnasis ryšys, taip pat vadinamas didžiausio skersmens arba tolimiausio kaimyno metodu, nustato atstumą tarp dviejų grupių, matuojant ilgiausią atstumą nuo bet kurio vienos grupės nario iki kito klasterio nario. Pilnojo ryšio algoritmų klasteriai yra kompaktiškesni ir glaudžiau susieti nei vienetinės jungties klasterizavimas [75].

2.1.4. Tinkleliu ir modeliais grįsti metodai

Viena iš duomenų klasterizavimo metodų grupių, kuri pasitaiko kur kas rečiau ir naudojama praktikoje, yra tinkleliu grįsti duomenų klasterizavimo metodai. Šie metodai dalija duomenų erdvę į baigtinį skaičių langelių (tinklelio elementų) ir struktūroje atlieka klasterizavimą. Pagrindinis tinklelio metodų pranašumas – greitas apdorojimo laikas, kuris nepriklauso nuo objektų skaičiaus duomenų rinkinyje. Populiarūs tinkleliu grįsti algoritmai yra STING (angl. *Statistical Information Grid;* liet. statistinės informacijos tinklelis)[76], CLIQUE (angl. *Clustering in Quest*) [77] ir „WaveCluster“ [78]. Tinklelio metodai gerai tinka didelių matmenų duomenims ir erdviniams duomenų rinkiniams, šių metodų našumas labai priklauso nuo tinklelio struktūros detalumo.

Paskutinė duomenų klasterizavimo metodų grupė yra modeliais grįsti (angl. *model-based*) duomenų klasterizavimo metodai. Modeliais grįsti klasterizavimo metodai yra metodų šeima, kuria siekiama identifikuoti pagrindinę duomenų rinkinio struktūrą prie duomenų pritaikant tikimybinį modelį [79]. Šiuose metoduose daroma prielaida, kad duomenys yra iš tikimybių skirstinių, paprastai Gauso ar kitų parametrinių skirstinių, mišinio, o kiekvienas skirstinys atitinka atskirą klasterį. Modeliu grįsto klasterizavimo tikslas – įvertinti šių skirstinių parametrus ir priskirti

duomenų taškus labiausiai tikėtiniems klasteriams pagal jų tikimybę. Labiausiai paplitęs modeliu grįstas klasterizavimo metodas yra Gauso mišinio modelis (GMM) [80, 81]. Čia daroma prielaida, kad duomenys yra iš Gauso skirstinių mišinio. Didžiausio tikėtinumo (EM) algoritmas paprastai naudojamas Gauso skirstinių parametrams įvertinti, tokiems kaip vidurkis, kovariacijos matrica ir mišinio komponentų proporcijos. EM algoritmas yra kartotinė procedūra, kuri vyksta dviem etapais: tikėtinumo (E) žingsniu, kurio metu apskaičiuojama kiekvieno duomenų taško, priklausančio kiekvienam klasteriui, aposteriorinė tikimybė, ir maksimizavimo (M) žingsnio, kuriame Gauso skirstiniai atnaujinami, kad būtų maksimaliai padidinta stebimų duomenų priskyrimo klasteriui tikimybė. Kai EM algoritmas konverguoja, duomenų taškai priskiriami klasteriui su didžiausia tikimybe [81].

Apibendrinant pažymėtina, kad šioje dalyje pateikta išsami įvairių klasterizavimo metodų, tokių kaip padalinimo, hierarchinių ir kitų metodų apžvalga. Aptariama keletas metodų modifikacijų ir patobulinimų, kurie buvo sukurti laikui bėgant, siekiant pagerinti jų veikimą ir pritaikomumą. Klasterizavimas atlieka gyvybiškai svarbų vaidmenį daugelyje sričių, tokių kaip duomenų tyryba, šablonų atpažinimas, vaizdo apdorojimas ir bioinformatika. Tolesniame skyriuje bus išsamiau nagrinėjami įvairūs klasterizavimo metodų taikymai, aptariant jų universalumą ir reikšmę įvairiose srityse.

2.2. Duomenų klasterizavimo taikymas

Kaip minėta, duomenų klasterizavimas taikomas ypač plačiai, tad galima teigti, kad taikomas beveik visose gyvenimo srityse. Atsižvelgiant į tai, jog šiame disertaciniame darbe dalis pateikiamų straipsnių orientuoti į praktinį klasterizavimo taikymą, šiame poskyryje pateikiama duomenų klasterizavimo taikymo apžvalga. Ji naudinga siekiant palyginti šiuo metu gerai žinomo bei disertacijos moksliniuose straipsniuose siūlomo klasterizavimo metodų taikymo atvejus. Tai reikšminga, nes svarbu įvertinti, kuo turi pasižymėti naujai kuriami metodai ir kaip jie gali būti pritaikomi. Tai analizuojant tikimasi atskleisti besikeičiančias duomenų klasterizavimo naudojimo tendencijas ir nustatyti esamas metodų tobulinimo ribas.

Atsižvelgiant į straipsnių apžvalgoje pateikiamus praktinius taikymo atvejus, čia pirmiausia apžvelgiamas duomenų klasterizavimo taikymas finansuose / ekonomikoje. Portfelio valdymas apima finansinio turto atranką ir paskirstymą, siekiant maksimaliai padidinti grąžą ir sumažinti riziką. Klasterizavimo metodai buvo naudojami portfeliui valdyti, siekiant sugrupuoti akcijas ar kitas finansines priemones pagal jų našumą, riziką ir koreliacijos ypatybes [82]. Klasterizavimo algoritmai, tokie kaip *k* vidurkių, hierarchinis klasterizavimas ir spektrinis klasterizavimas, buvo naudojami siekiant nustatyti turto grupes su panašiais rizikos ir grąžos profiliais, kurie gali padėti sukurti diversifikuotus ir efektyvius portfelius [83]. Finansų ir ekonomikos kontekste klientų segmentavimas reiškia klientų skirstymą į grupes pagal jų finansines charakteristikas, tokias kaip pajamos, išlaidų įpročiai, kredito istorija ir rizikos pasirinkimai. Klasterizavimo metodai buvo plačiai taikomi atliekant klientų segmentavimo užduotis, siekiant nustatyti vienarūšes klientų grupes ir informuoti apie tikslines rinkodaros, skolinimo ir investavimo strategijas [84]. Naujausi klientų

segmentavimo pasiekimai apima giliojo mokymosi metodus, tokius kaip enkoderiai ir neuroniniai tinklai, siekiant pagerinti klasterizavimo našumą atpažįstant sudėtingus klientų duomenų modelius [85]. Kitas svarbus taikymas, kuriame dažnu atveju naudojami būtent robastiniai duomenų klasterizavimo metodai, yra sukčiavimo aptikimas. Sukčiavimo aptikimas – tai įtartinos ar anomalios veiklos, susijusios su finansinėmis operacijomis, nustatymo procesas, pvz., sukčiavimas kredito kortelėmis, prekyba pasinaudojant viešai neatskleista informacija ir mokesčių slėpimas. Klasterizavimo metodai buvo taikomi atliekant sukčiavimo aptikimo užduotis, siekiant sugrupuoti sandorius arba vartotojus pagal jų elgesio modelius, palengvinančius nuokrypių ir galimų sukčiavimo atvejų identifikavimą [86]. Šis taikymas taip pat aptariamas ir viename straipsnyje pateiktame straipsnių apžvalgos skyriuje. Naujausi sukčiavimo nustatymo tyrimai išnagrinėjo klasterizavimo algoritmų naudojimą kartu su kitais metodais, siekiant pagerinti nesąžiningos veiklos nustatymą. Subudhi ir Panigrahi (2020) pasiūlė hibridinį automobilių draudimo sukčiavimo aptikimo metodą, taikant genetiniu algoritmu ir neraiškiąja logiką grįstą *c* vidurkių (angl. *fuzzy c-means*) klasterizavimą ir įvairius prižiūrimus klasifikatoriaus modelius, kurie buvo efektyvūs naudojant realaus pasaulio automobilių draudimo duomenų rinkinį [87]. Ekonominėje analizėje duomenų klasterizavimas taip pat yra paplitęs. Makroekonominė analizė apima ekonominių rodiklių, tokių kaip BVP, infliacija ir nedarbas, tyrimą, siekiant suprasti bendrus ekonomikos rezultatus ir tendencijas. Klasterizavimo metodai buvo taikomi makroekonominėje analizėje, siekiant sugrupuoti šalis ar regionus pagal jų ekonomines charakteristikas, o tai gali padėti nustatyti augimo modelius, politikos poveikį ir galimas investavimo galimybes [88–91]. Šiuo metu beveik kiekviena įmonė duomenų klasterizavimą taiko savo veikloje segmentuodama vartotojus. Vartotojų segmentavimas, taip pat žinomas kaip klientų segmentavimas, yra vartotojų skirstymas į atskiras grupes pagal jų bendras savybes, pvz., demografinius rodiklius, nuostatas ir elgesį. Klasterizavimo metodai, tokie kaip *k* vidurkių, hierarchinis klasterizavimas ir DBSCAN, buvo plačiai naudojami atliekant vartotojų segmentavimo užduotis, siekiant nustatyti vienarūšes vartotojų grupes ir informuoti apie tikslines rinkodaros strategijas, suasmenintas rekomendacijas ir vartotojo patirtis [92, 93]. Vertinant vartotojus dėl didelio duomenų kiekio rinkimo atsiranda ir naujos analizės galimybės – jų elgesio analizė. Naudotojų elgesio analizės tikslas – suprasti vartotojų veiksmus, nuostatas ir sąveiką su skaitmeninėmis platformomis, tokiomis kaip svetainės, mobiliosios programos ir internetinės paslaugos. Klasterizavimo metodai buvo naudojami analizuojant vartotojų elgseną, siekiant sugrupuoti vartotojus pagal jų elgesio modelius, palengvinančius vartotojų pageidavimų, poreikių ir problemų identifikavimą [94, 95]. Tačiau svarbu nustatyti ne tik vartotojų elgesį, bet ir sugebėti reaguoti remiantis gauta informacija apie vartotojus, todėl klasterizavimas atlieka svarbų vaidmenį kuriant ir tobulinant rekomendacijų sistemas, kurios plačiai naudojamos įvairiose srityse, siekiant pasiūlyti vartotojams atitinkamus elementus, tokius kaip produktai, filmai ir muzika, atsižvelgiant į jų pageidavimus [96]. Pagrindinis klasterizavimo rekomendacijų sistemose aspektas – vartotojų arba elementų su panašiomis savybėmis klasterizavimas, siekiant teikti individualizuotas rekomendacijas naudojant skirtingus klasterizavimo algoritmus, tokius kaip *k* vidurkių, hierarchinis klasterizavimas ir

DBSCAN [97]. Įrodyta, kad klasterizavimo metodų taikymas rekomendacijų sistemose padeda išspręsti mastelio ir retumo problemas, kurios vyrauja didelės apimties duomenų rinkiniuose [98]. Pavyzdžiui, naudojant klasterizavimo algoritmus, galima reikšmingai sumažinti duomenų rinkinio matmenis, identifikuojant reprezentatyvius vartotojus ar elementus, taip pagerinant rekomendacijų teikimo proceso efektyvumą ir greitį [99].

Klasterizavimo taikymas tampa vis svarbesnis bioinformatikos srityje, atsižvelgiant į spartų biologinių duomenų augimą ir poreikį iš sudėtingų duomenų rinkinių gauti reikšmingų įžvalgų [100]. Klasterizavimas buvo naudojamas atliekant įvairias bioinformatikos užduotis, tokias kaip genų ekspresijos analizė, baltymų struktūros prognozavimas ir metagenomika, siekiant nustatyti modelius ir ryšius tarp biologinių subjektų [101]. Genų ekspresijos analizės kontekste buvo taikomi klasterizavimo metodai, skirti genams su panašiais ekspresijos modeliais sugrupuoti, kurie dažnai atitinka genus, turinčius panašias funkcijas arba dalyvaujančius tuose pačiuose biologiniuose procesuose [102]. Pavyzdžiui, hierarchinio ir *k* vidurkių klasterizavimo algoritmai buvo plačiai taikomi analizuojant mikrogardelių duomenis, leidžiančius identifikuoti ir apibūdinti įvairius vėžio tipus pagal genų ekspresijos profilius [103]. Be to, naujausia pažanga vienos ląstelės RNR sekos (scRNA-seq) srityje paskatino sukurti naujus klasterizavimo algoritmus. Tokius kaip grafais pagrįsti metodai ir matmenų mažinimo metodai, siekiant atskleisti ląstelių tipui būdingus genų ekspresijos modelius ir pagerinti mūsų supratimą apie ląstelių heterogeniškumą [104, 105].

Klasterizavimas taip pat tapo svarbiu teksto tyrybos ir informacijos gavimo būdu, atlikus daugybę tyrimų, kuriuose nagrinėjami įvairūs metodai, kaip padidinti teksto informacinių sistemų efektyvumą, tikslumą ir aktualumą [106]. Šie metodai leidžia organizuoti ir suskirstyti į kategorijas nestruktūrizuoto teksto duomenis, taip pat išgauti prasmingus modelius ir ryšius. Jie taikomi įvairiose srityse, tokiose kaip dokumentų paieška, temų modeliavimas, nuotaikų analizė ir teksto klasifikavimas [107]. Daugiau apie šį taikymą informacijos pateikiama viename iš aptariamų straipsnių, publikacijų apžvalgos skyriuje. Populiarus klasterizavimo metodas, taikomas teksto tyrybai ir informacijai gauti, yra *k* vidurkių algoritmas, kuris suskirsto dokumentus į *k* grupių, atsižvelgiant į jų panašumą pagal ypatybes, tokias kaip terminų dažnis – atvirkštinis dokumentų dažnis (TF-IDF) [108]. *k* vidurkių algoritmo veiksmingumas klasterizuojant tekstą buvo įrodytas daugelyje tyrimų, patobulinus ir modifikavus algoritmą, siekiant išspręsti tokias problemas kaip pradinių centroidų ir klasterių skaičiaus parinkimas [46]. Kitas plačiai taikomas klasterizavimo metodas šioje srityje yra hierarchinis klasterizavimas, kuris sukuria į medį panašią struktūrą, vaizduojančią įdėtus ryšius tarp dokumentų [109]. Šis metodas buvo ypač naudingas atliekant taikymus, kuriuose reikalingas hierarchinis dokumentų ar temų organizavimas, pvz., interneto paieškos sistemose ir skaitmeninėse bibliotekose [107]. Pavyzdžiui, buvo įrodyta, kad DBSCAN taikymas naujienų straipsniams klasterizuoti suteikia tikslesnes ir nuoseklesnes grupes, palyginti su *k* vidurkių ir hierarchiniu klasterizavimu, nes jis gali identifikuoti ir išskirti triukšmingus duomenų taškus ir yra robastinis [110]. Mašininio mokymosi ir natūralios kalbos apdorojimo pažanga paskatino sukurti sudėtingesnius teksto duomenų grupavimo metodus. Vienas iš tokių

patobulinimų – temų modeliavimo metodas, toks kaip latentinis dirichleto paskirstymas (LDA). Šis metodas priskiria dokumentus temoms pagal latentinių kintamųjų požymius [111]. Šis metodas įrodė savo veiksmingumą fiksuojant pagrindinę teksto žodyno struktūrą, leidžiančią tiksliau ir geriau interpretuoti dokumentų grupavimą ir paiešką [112]. Be to, giliuoju mokymusi pagrįsti grupavimo būdai, tokie kaip automatiniai koderiai ir gilieji neuroniniai tinklai, buvo pritaikyti teksto tyrybai ir informacijai gauti, siekiant sukurti prasmingesnes ir patikimesnes dokumentų vaizdavimo priemones, galinčias pagerinti klasterizavimo našumą [113].

Kalbant apie nestruktūrizuotą informaciją, kuri buvo apžvelgiama anksčiau, svarbu atsižvelgti ir į kitą nestruktūrizuotos informacijos tipą – vaizdinę informaciją. Klasterizavimo metodai atlieka svarbų vaidmenį įvairiose srityse, įskaitant vaizdo apdorojimą ir kompiuterinę regą, kur jie yra galingi įrankiai savybėms išgauti, šablonams atpažinti, vaizdui segmentuoti. Skirtingai nuo klasifikavimo, klasterizavimas nesiremia iš anksto suformuotomis nuostatomis ar kategorijomis, o grupuoja duomenis pagal būdingus panašumus, todėl tinka neprižiūrimo mokymosi scenarijams įgyvendinti [114]. Šiame skyriuje aptariame keletą klasterizavimo pritaikymų, skirtų vaizdams apdoroti ir kompiuterinei regai, pabrėždami šių metodų svarbą ir universalumą. Vienas pagrindinių vaizdų apdorojimo klasterizavimo taikymo būdų yra vaizdo segmentavimas, kurio pagrindinis tikslas – padalinti vaizdą į atskirus regionus, turinčius nuoseklias vaizdines savybes [115]. Šis procesas yra būtinas įvairiems taikymams, įskaitant objektų atpažinimą, ir medicininių vaizdų analizę. *k* vidurkių klasterizavimas yra populiarus vaizdų segmentavimo užduočių pasirinkimas dėl savo paprastumo ir galimybės tvarkyti didelius duomenų rinkinius [116]. Kiti klasterizavimo algoritmai, tokie kaip hierarchinis klasterizavimas, DBSCAN ir vidurkio poslinkis, taip pat sėkmingai buvo pritaikyti vaizdams segmentuoti [117]. Kitas svarbus klasterizavimo taikymas kompiuterinėje regoje yra informacijos iš vaizdo išskyrimas. Pavyzdžiui, klasterizavimo algoritmai, tokie kaip hierarchinis klasterizavimas ir spektrinis klasterizavimas, buvo naudojami vaizdų ypatybėms automatiškai išmokti, kurios gali būti naudojamos tokioms užduotims kaip objekto atpažinimas [118]. BoVW (angl. *Bag of Visual Words*) yra vienas iš tokių vaizdų, kai vietiniai ypatybių aprašai yra sugrupuoti, kad sudarytų vaizdinį žodyną, o vaizdai vaizduojami kaip šių vaizdinių žodžių histogramos [119]. Šis metodas buvo sėkmingas atliekant įvairias kompiuterinės regos užduotis, tokias kaip vaizdų klasifikavimas ir paieška [120].

Apibendrinant šią trumpą duomenų klasterizavimo taikymo apžvalgą pažymėtina, kad duomenų klasterizavimas taikomas daugumoje mokslo ir verslo sričių. Didėjant duomenų kiekiams, duomenų klasterizavimas, kaip neprižiūrimo mašininio mokymosi metodas, tampa vis aktualesnis, nes leidžia gauti naudingą informaciją iš anksto neturint aiškių klasių. Svarbu ir tai, kad duomenų klasterizavimas dažnu atveju yra kaip didesnės duomenų analizės sistemos sudedamoji dalis, leidžianti atskirti tam tikras grupes, kurias tampa kur kas paprasčiau interpretuoti ar atlikti nuodugnesnę reikšmingų grupių analizę.

3. STRAIPSNIŲ APŽVALGA

Šiame skyriuje atskirai apžvelgiami straipsniai, įtraukti į šią disertaciją, ir kaip jie atskleidžia disertacijos temą.

3.1. Straipsnio "Nonparametric Multivariate Density Estimation: Case Study of Cauchy Mixture Model" apžvalga

M. Lukausko įnašas į straipsnį: M. Lukauskas yra kontaktinis straipsnio asmuo (angl. *corresponding author*), atliko eksperimentinius skaičiavimus, kartu su bendraautoriais pateikė pradinę rankraščio versiją, bendraudamas su recenzentais ir redaktoriumi atliko reikalingus straipsnio taisymus.

Straipsnio apibendrinimas. Šiame darbe empiriniu būdu tiriami penki pagrindiniai neparametrinių daugiamačių tankių įvertinių metodai, kai nedaroma prielaida, kad duomenys yra iš bet kurios žinomos parametrinės pasiskirstymo šeimos. Buvo sukurtas apvertimo formulės tankio įvertinimo metodas, į bendrą mišinio modelį įtraukiant triukšmo klasterį. Toks triukšmo klasterio įtraukimas suteikia metodui geresnes galimybes robastiškai įvertinti tankio įvertį. Šio metodo efektyvumas įrodomas atliekant lyginamąją analizę ir modeliavimą. Tankio funkcijos įvertis laikomas vienu pagrindinių statistinio modeliavimo uždavinių. Jis išreiškia atsitiktinius dydžius kaip kitų kintamųjų funkcijas, nes nurodo, kaip kintamieji yra susiję, galima nustatyti ryšius tarp kintamųjų ir taip gauti nuodugnesnę duomenų supratimo analizę.

Sprendžiant įvairius taikomuosius uždavinius, daugelis algoritmų veikia gerai, jei žinoma pasiskirstymo tankių šeima. Deja, tikrovėje šie tankiai paprastai nežinomi, ir tankių vertinimas tampa ypač reikšmingas. Priešingai nei atliekant parametrinį tankio vertinimą, kuriame daromos prielaidos apie duomenų parametrinį pasiskirstymą, atliekant neparametrinį tankio vertinimą prielaidos apie duomenų pasiskirstymą yra ne tokios griežtos. Turint d-mačius stebėtus duomenis $\{X_i, i = 1, \ldots, d\}$, daugiamačio tankio vertinimo uždavinys – rasti įvertinį $\hat{f}$, kuris geriausiai aproksimuoja tikrąjį tankį f.

Šio tyrimo pagrindinis metodas, taikomas atliekant modeliavimą ir palyginimą, yra Monte Karlo metodas. Minėtas palyginimas leidžia įvertinti realias tankio reikšmes ir įvertinti atskirų tiriamų algoritmų efektyvumą. Tyrimui naudojami daugiamačiai Koši skirstiniai (d = 2, 5, 10, 15). 1 lentelėje pateikiami pagrindiniai parametrai.

1 lentelė. Tyrime naudotų komponentų ir parametrų reikšmės

Komponentų skaičius	Komponentų tikimybės	Centrų parametrai	Centrų atskyrimo atstumas
$q = 2$	$p_1 = (1 - p_2)$, $p_2 = 0{,}1, 0{,}3, 0{,}5$	$m_1 = (0, 0)$, $m_2 = (0{,}5i, 0{,}5i)$	$i = 1, 2, \ldots, 6$
$q = 3$	$p_1 = p_2 = (1 - p_3)/2$,	$m_1 = (0, 0)$,	$i = 1, 2, \ldots, 6$

	$p_3 = 0{,}1,\ 1/3,\ 0{,}8$	$m_2 = (0{,}5i,\ 0{,}5i)$, $m_3 = (0{,}5i,\ 0)$	
$q = 4$	$p_1 = p_2 = p_3 = (1 - p_4)/3$, $p_4 = 0{,}1,\ 0{,}25,\ 0{,}7$	$m_1 = (0,\ 0)$, $m_2 = (0{,}5i,\ 0{,}5i)$, $m_3 = (0{,}5i,\ 0)$, $m_4 = (0,\ 0{,}5i)$	$i = 1, 2, ..., 6$
$q = 2$	$p_1 = (1 - p_2)$, $p_2 = 0{,}1,\ 0{,}2,\ 0{,}3,\ 0{,}4,\ 0{,}5$	$m_1 = (0,\ 0,\ 0,\ 0,\ 0)$, $m_2 = (0{,}5i,\ 0{,}5i,\ 0{,}5i,\ 0{,}5i,\ 0{,}5i)$	$i = 1, 2, ..., 6$
$q = 3$	$p_1 = p_2 = (1 - p_3)/2$, $p_3 = 0{,}1,\ 0{,}2,\ 1/3,\ 0{,}4,\ 0{,}6,\ 0{,}8$	$m_1 = (0,\ 0,\ 0,\ 0,\ 0)$, $m_2 = (0{,}5i,\ 0{,}5i,\ 0{,}5i,\ 0{,}5i,\ 0{,}5i)$, $m_3 = (0{,}5i,\ 0{,}5i,\ 0,\ 0,\ 0)$	$i = 1, 2, ..., 6$
$q = 4$	$p_1 = p_2 = p_3 = (1 - p_4)/3$, $p_4 = 0{,}1,\ 0{,}16,\ 0{,}25,\ 0{,}4,\ 0{,}7$	$m_1 = (0,\ 0,\ 0,\ 0,\ 0)$, $m_2 = (0{,}5i,\ 0{,}5i,\ 0{,}5i,\ 0{,}5i,\ 0{,}5i)$, $m_3 = (0{,}5i,\ 0{,}5i,\ 0,\ 0,\ 0)$, $m_4 = (0,\ 0,\ 0{,}5i,\ 0{,}5i,\ 0{,}5i)$	$i = 1, 2, ..., 6$

Tyrime panaudoti algoritmai: AKDE – adaptuotas branduolinis (angl. *adaptive kernel*), PPDE – tikslinio projektavimo (angl. *projection pursuit*), LSDE – logsplainų (angl. *logsline*), SKDE – pusiau parametrinis branduolinis (angl. *semi-parametric kernel*), IFDE – apvertimo formulė (angl. *inversion formula*), MIDE – apvertimo formulė su triukšmo klasteriu (angl. *inversion formula with noise cluster*). Šiame tyrime Monte Karlo metodu buvo siekiama atlikti anksčiau minėtų (AKDE, PPDE, LSDE, SKDE, IFDE, MIDE) neparametrinių pasiskirstymo tankio įverčių tikslumo tyrimą. Autoriai [121] siūlo rinkti AKDE metode naudojamo jautrumo parametro reikšmę γ iš aibės $\{0{,}2;\ 0{,}4;\ 0{,}6;\ 0{,}8\}$, todėl šiame tyrime taip pat buvo naudojamos šios reikšmės. Konkreti parametro reikšmė nustatoma tikimybiniu kryžminiu patikrinimu [122, 123].

SKDE buvo iš naujo pasirinktos visos galimos subvektoriaus Y dimensijos s reikšmės ($1 \leq s \leq d - 1$, kur d yra matmenys) ir jas atitinkančios koordinatės. LSDE metode parenkant bazinio splaino taškų skaičių, minimizuojamas Akaike informacijos kriterijus [124]. Akaike informacijos kriterijus $\mathrm{AIC} = -2l(t) + a\,J(t)$, J – splaino laipsnis, $a = \log(n)$, l – tikėtinumo funkcija, naudojama splaino koeficientams parinkti. MIDE metodas turi glodumo parametrą, h. Pasirinkta glodumo daugiklio e $\{-hu^2\}$ forma leidžia susieti glodumo parametrą h su projekcijų klasterių dispersijomis. Modeliavimo tyrimai parodė, kad šis metodas yra jautrus parametrų pasirinkimui, todėl parametrų parinkimas yra ypač svarbus žingsnis. Pavyzdžiui, jei h parametras nustatomas per mažas, tuomet tankio įvertis tampa labai aptakus ir sukelia dideles paklaidas. Jei glodumo reikšmė yra per didelė, tankio įvertinys nebūna labai paveiktas. Tyrimuose pastebėta, kad vertinimas tampa netolygus dėl projektuojamų stebėjimų reikšmių panašumo, taip išskiriant mažo svorio komponentus su mažomis dispersijomis. Glodumo parametras (h), taip pat specifinė triukšmo klasterio svorio reikšmė (tikimybė) iš aibės $\{0{,}05;\ 0{,}1;\ 0{,}15;\ 0{,}2;\ 0{,}3;\ 0{,}4\}$ atrenkami kryžminiu būdu minimizuojant kvadratinę paklaidą.

Skirtingų metodų tankio įverčio tikslumo vertinimui skaičiuojama vidutinė absoliučioji paklaida ir (3.1) vidutinė absoliučioji procentinė paklaida (3.2).

$$\delta_1 = \frac{1}{n}\sum_{t=1}^{n}\left|f(x(t)) - \hat{f}(x(t))\right| \cong \int\left|f(x) - \hat{f}(x)\right| f(x)dx. \qquad (3.1)$$

$$\delta_2 = \frac{2}{n}\sum_{t=1}^{n}\left|\frac{f(x(t)) - \hat{f}(x(t))}{f(x(t)) + \hat{f}(x(t))}\right| \cong \int\left|f(x) - \hat{f}(x)\right| dx. \qquad (3.2)$$

Toliau pateikiama viena iš gautų rezultatų lentelių. Visi rezultatai pateikiami visateksčiame straipsnyje. Rezultatų lentelėse pateikiami duomenys, atspindintys 100 000 imčių absoliučiosios procentinės paklaidos vidurkį bei standartinį nuokrypį. Tyrimo metu buvo pastebėta, kad kai $q = 2$, $n = 100$, $d = 5$, geriausi rezultatai gaunami SKDE ir MIDE metodais. Kai $q = 2$, $n = 200$, geriausi rezultatai taip pat gaunami taikant SKDE ir MIDE metodus. Gauti rezultatai rodo, kad kai $q = 3$, $n = 200$, esant labai sutampantiems skirstiniams ($i = 1, 2$), geriausi rezultatai gaunami taikant SKDE metodą, o esant labiau atskirtiems skirstiniams ($i \geq 3$) – MIDE metodu. Šiuo atveju i nusako atstumą tarp mišinio komponentų centrų. Kuo tie komponentai arčiau vienas kito (i mažas), tuo labiau jie sutampa. Kai $i \geq 5$, tai jie jau smarkiai atsiskyrę (kai $i = 6$, juos jau galima visiškai atskirti su hiperplokštuma). Galima pastebėti, kad kai $q = 3$, $n \geq 400$, geriausi rezultatai gaunami SKDE metodu, o antras geriausias metodas yra MIDE. Pastebima, kad kai $q = 4$, $n = 400$, esant labiau sutampantiems skirstiniams ($i < 3$), geriausi rezultatai gaunami SKDE metodu, o esant labiau atskirtiems skirstiniams ($i \geq 4$) – MIDE metodas. Galima pastebėti, kad esant mažiau sutampantiems arba vidutiniškai atskirtiems skirstiniams ($i \leq 5$), geriausi rezultatai gaunami taikant SKDE metodą, o esant labai atskirtiems skirstiniams ($i = 6$) – MIDE metodu. Kai $q = 2$ ir $n = 50$, visais atvejais tiek sutampantiems, tiek atskirtiems skirstiniams geriausi rezultatai gaunami AKDE metodu, o esant labiau atskirtiems skirstiniams ($i = 6$), kai $p_1 = 0{,}6$, $p_2 = 0{,}4$ – MIDE metodu. Kai $q = 3$ ir $n = 50$, geriausi rezultatai visais atvejais gaunami taikant AKDE metodą (labai persidengiantys arba izoliuoti skirstiniai). LSDE metodas su didžiuliais nuokrypiais ($|x–m_j| > 100u_j$) sugrupuojamas su reikšmingesniu reikšmių skaičiumi arčiau centro.

2 lentelė. Tyrimo metu gauti pagrindiniai rezultatai

Įvertinimo metodas		**Tankis**					
		***d* = 5; $p_1 = p_2 = p_3$ = 1/3; *n* = 100**					
		***i* = 1**	***i* = 2**	***i* = 3**	***i* = 4**	***i* = 5**	***i* = 6**
AKDE	Vidurkis	0,8268	0,8257	0,8198	0,8128	0,8066	0,8075
	SN	(0,0760)	(0,0814)	(0,0848)	(0,0827)	(0,0788)	(0,0731)
PPDE	Vidurkis	0,9243	0,9319	0,9303	0,9300	0,9284	0,9250
	SN	(0,0500)	(0,0364)	(0,0375)	(0,0387)	(0,0410)	(0,0433)
LSDE	Vidurkis	0,8043	0,8162	0,8583	0,8611	0,8613	0,8711
	SN	(0,0534)	(0,0540)	(0,0490)	(0,0349)	(0,0434)	(0,0577)
SKDE	Vidurkis	**0,7158**	**0,7144**	**0,7088**	**0,7071**	0,7179	0,7227
	SN	(0,0260)	(0,0905)	(0,0905)	(0,0830)	(0,0631)	(0,0499)

IFDE	Vidurkis	0,94593	0,8886	0,7857	0,8463	0,8761	0,8312
	SN	(0,0362)	(0,1318)	(0,0706)	(0,0380)	(0,1110)	(0,0538)
MIDE	Vidurkis	0,7389	0,7332	0,7235	0,7149	**0,7121**	**0,7219**
	SN	(0,0280)	(0,0221)	(0,0338)	(0,0195)	(0,0208)	(0,0203)

Mažesnių matmenų ($d = 2$) rezultatai rodo, kad geriausi rezultatai gaunami taikant SKDE metodą sutampant tiek didelei, tiek mažai apimčiai ($i < 4$). Kita vertus, esant izoliuotam pasiskirstymui ($i \geq 5$), geri rezultatai gauti taikant MIDE metodą.

Išvados. Šiame darbe apžvelgti populiariausi ir dažniausiai naudojami neparametrinio tankio įvertinimo algoritmai. Straipsnyje taip pat pateikta tankio įvertinimo apvertimo formulė. Pastebėta, kad įtraukus triukšmo klasterį, apvertimo formulės rezultatai tampa statistiškai reikšmingai geresni ($p < 0,05$). Tai patvirtina, kad įtrauktas triukšmo klasteris padeda įvertinti įverčius, atsparius išskirtinėms reikšmėms. Remiantis vidutine absoliučiąja paklaida, esant didesnėms dimensijoms ($d \sim 5$) ir mažoms imtims ($n \sim 50$), rekomenduojama taikyti adaptyvų branduolio metodą. Esant didesniam stebinių skaičiui $n \sim 100$, pastebima, kad tiksliausi rezultatai gaunami, kai naudojamas modifikuotos apvertimo formulės tankio įverčio metodas (MIDE), kuris buvo plėtojamas. Didesnėms imtims su sutampančiais skirstiniais rekomenduojama naudoti iš dalies parametrinį branduolinį, o labiau izoliuotam paskirstymui – modifikuotą apvertimo formulės metodą. Remiantis vidutine absoliučiąja procentine paklaida, rekomenduojama naudoti iš dalies parametrinį branduolinį, kai imtis yra su sutampančiais skirstiniais. Esant dviem dimensijoms ($d \sim 2$), kai imtis yra su sutampančiais skirstiniais, rekomenduojama taikyti iš dalies parametrinį branduolinį metodą. Atskiriems pasiskirstymams rekomenduojama taikyti adaptyvų branduolio metodą.

3.2. Straipsnio "A New Clustering Method Based on the Inversion Formula" apžvalga

M. Lukausko įnašas į straipsnį: M. Lukauskas yra kontaktinis straipsnio asmuo (angl. *corresponding author*), sudarė duomenų klasterizavimo metodą, pagrįstą modifikuotu apvertimo formulės tankio įvertiniu, ir jį programiškai realizavo, atliko eksperimentinius skaičiavimus, parengė straipsnį.

Straipsnio apibendrinimas. Tobulėjant duomenų mokslo sričiai, daug dėmesio skiriama naujų tankio įvertinimo procedūrų kūrimui [125, 126]. Ankstesniame straipsnyje "Nonparametric Multivariate Density Estimation: Case Study of Cauchy Mixture Model" buvo pateiktas palyginimas tarp modifikuoto tankio įverčio tikslumo, lyginant su kitais tankio įvertinimo metodais, o šiame straipsnyje tankio įvertinys pritaikomas duomenims klasterizuoti. Pastaraisiais metais mokslininkai pradėjo siūlyti skirtingus robastinius tankio įvertinimo metodus, pagrįstus neuroniniais tinklais. Vieni iš šių tyrimų yra Parzen neuroniniai tinklai [127], save ribojantis neuroninis tinklas (angl. *Self constrained neural network*) [128] ir kiti [129, 130]. Tačiau yra ir daugiau naujų tankio įvertinimo metodų, kurie buvo pristatyti paskutiniais metais [131–133] Tikimybių tankio funkcijų (angl. *probability density function, pdf)* įvertinimas laikomas viena svarbiausių statistinio modeliavimo dalių.

Ši dalis atsitiktinius dydžius pateikia kaip kitų kintamųjų funkcijas, todėl galima aptikti paslėptus ryšius tarp duomenų [134]. Nemažoje dalyje mašininio mokymosi algoritmų būtina nustatyti duomenų pasiskirstymo tankį. Tikimybių tankio funkcija taikoma Bajeso klasifikatoriaus [135, 136] tankiu pagrįstuose klasterizavimo algoritmuose [137–140] arba informacija pagrįstuose stebinio savybių / kintamųjų pasirinkimo algoritmuose [141, 142]. Veiksmingi tankio įverčiai turi būti kruopščiai įvertinti iš anksto, kad gautume nežinomas tikimybių tankio funkcijas. Daug dėmesio vis dar skiriama naujų tankio įvertinimo procedūroms kurti [125, 126].

Šiame straipsnyje iškėlėme hipotezę, kad tankio įvertinys, pagrįstas modifikuota apvertimo formule, yra tinkamas duomenims klasterizuoti. Atsižvelgiant į tai, kad duomenų klasterizavimo algoritme / metode naudojamas triukšmo klasteris, šis metodas šiame kontekste laikomas robastiniu duomenų klasterizavimo metodu. Manoma, kad sukurtas robastinis duomenų klasterizavimo metodas sėkmingai suskirstys duomenis į atskiras grupes net esant triukšmui ar išskirtinėms reikšmėms. Šiame straipsnyje buvo siekiama pristatyti naują duomenų klasterizavimo metodą, pagrįstą modifikuotos apvertimo formulės tankio įvertiniu. Buvo manoma, kad naujas metodas leis atlikti tikslesnį klasterizavimą, lyginant su populiariausiais šiuo metu taikomais metodais: *k* vidurkių, Gauso mišinių metodu ir kitais metodais, pristatytais literatūros apžvalgoje.

Jei turimas atsitiktinis vektorius $X \in \mathbf{R}^d$ tenkina toliau pateiktą lygybę $f(x)$, galima teigti, kad atsitiktinis vektorius tenkina skirstinių mišinio modelį:

$$f(x) = \sum_{k=1}^{q} p_k f_k(x) = f(x, \theta), \tag{3.3}$$

čia $f_k(x)$ – pasiskirstymo tankio funkcija; θ – daugiamatis modelio parametras. Mašininių klasterių (komponentų, grupių) skaičius apibrėžiamas parametru q, o p_k yra apriorinės tikimybės, kurios tenkina sąlygas:

$$p_k > 0, \sum_{k=1}^{q} p_k = 1. \tag{3.4}$$

Duomenų dimensijų didėjimas sukelia daug problemų aproksimuojant parametrinius metodus, nes didėjantis dimensijų skaičius smarkiai didina parametrų skaičių. Toks parametrų skaičiaus augimas apsunkina tikslių parametrų įverčių radimą. Straipsnyje pristatomas modifikuotos apvertimo formulės tankio įvertinys ir jo naudojimas.

$$f_\tau(x) = \sum_{k=1}^{q} p_{k,\tau} \varphi_{k,\tau}(x) = f_\tau(x, \theta_\tau), \tag{3.5}$$

čia $\varphi_{k,\tau}(x) = \varphi\left(x;\, m_{k,\tau}, \sigma_{k,\tau}^2\right)$ – vienmatis Gauso skirstinio tankis. Daugiamatis mišinio parametras ir duomenų skirstinio parametrai $\theta_\tau = \left(p_{k,\tau}, m_{k,\tau}, \sigma_{k,\tau}^2\right),\ k = 1, \ldots, q$ susieti lygybėmis:

$$\begin{aligned} p_{j,\tau} &= p_j \\ m_{j,\tau} &= \tau' M_j \\ \sigma_{j,\tau}^2 &= \tau' R_j \tau \end{aligned} \tag{3.6}$$

Tuomet pasinaudojus apvertimo formulės tankio įvertiniu,

$$f(x) = \frac{1}{(2\pi)^d} \int\limits_{\mathbf{R}^d} e^{-it'x} \psi(t) dt, \tag{3.7}$$

čia $\psi(t) = \mathrm{E} e^{it'x}$ žymi atsitiktinio kintamojo X charakteristinę funkciją. Taikant apvertimo formulę parinkus aibę *T*, kuri apibrėžiama kaip projektavimo kryptys, išsidėsčiusios ant sferos, ir charakteristinę funkciją keičiant jos įvertiniu, gaunama formulė įverčiui apskaičiuoti [15, 143]:

$$\hat{f}(x) = \frac{A(d)}{\#T} \sum_{\tau \in T} \int\limits_0^\infty e^{-iu\tau' x} \hat{\psi}_\tau(u) u^{d-1} e^{-hu^2} du, \tag{3.8}$$

čia ir toliau esančiose formulėse # žymi elementų skaičių rinkinyje. Su *d* matmens sferos tūrio formule

$$V_d(R) = \frac{\pi^{\frac{d}{2}} R^d}{\Gamma\left(\frac{d}{2} + 1\right)} = \begin{cases} \dfrac{\pi^{\frac{d}{2}} R^d}{\left(\frac{d}{2}\right)!}, & \text{kai } d \bmod 2 \equiv 0 \\ \dfrac{2^{\frac{d+1}{2}} \pi^{\frac{d-1}{2}} R^d}{d!!}, & \text{kai } d \bmod 2 \equiv 1 \end{cases} \tag{3.9}$$

Šiuo atveju konstantą *A(d)* galima apskaičiuoti pagal toliau pateiktą formulę:

$$A(d) = \frac{\left(V_d(1)\right)'_R}{(2\pi)^d} = \frac{d 2^{-d} \pi^{-\frac{d}{2}}}{\Gamma\left(\frac{d}{2} + 1\right)}. \tag{3.10}$$

Svarbu, kad pasiūlytoje formulėje gauti tankio įverčiai yra neglotnūs, dėl šios priežasties 3.8 formulėje po integralo ženklu naudojamas papildomas glodinimo daugiklis e^{-hu^2}. Daugiklis e^{-hu^2} papildomai glodina įvertį $\hat{f}(x)$ su Gauso branduolio funkcija, čia *h* – glodinimo parametras. Tyrimai Monte Karlo metodu parodė, kad naudojant tokį daugiklį sumažėja įverčių paklaidos. Taip pat svarbu, kad tokia daugiklio forma leidžia analitiškai apskaičiuoti integralo reikšmę. Tačiau pateiktos formulės trūkumas tas, kad šiuo įvertiniu aprašomas Gauso skirstinių mišinio modelis

(kai $f_k = \varphi_k$) (3.8 formulė) gerai vertina tik Gauso skirstiniui artimo pasiskirstymo stebinių tankį. Taip pat pažymėtina, kad aproksimuojant tankį Gauso skirstinių mišiniu apvertimo formulės tankio įvertinys tampa sudėtingas. Šis įvertinys tampa sudėtingas dėl didelio komponentų skaičiaus su mažomis pradinėmis tikimybėmis. Šių tikimybių skaičių galima kontroliuoti ir sumažinti naudojant triukšmo klasterio komponentą.

Toliau kaip straipsnio pagrindą galime aptarti modifikuotą algoritmą, kuris remiasi daugiamačiu Gauso skirstinio mišinio modeliu. Tam gali būti taikoma apvertimo formulė. Tokiu atveju tolydžiojo skirstinio tankio charakteristinės funkcijos parametrinis įvertinys aprašomas taip:

$$\hat{\psi}(u) = \frac{2}{(b-a)u} \sin\frac{(b-a)u}{2} \cdot e^{\frac{iu(a+b)}{2}}. \tag{3.11}$$

Siekiant įvertinti tankį, buvo sukonstruotas charakteristinės funkcijos tankio įvertinys. Šis tankio įvertinys yra Gauso skirstinių mišinio ir tolydžiojo skirstinio charakteristinių funkcijų sąjunga su atitinkamomis *apriorinėmis* tikimybėmis:

$$\hat{\psi}_\tau(u) = \sum_{k=1}^{\hat{q}_\tau} \hat{p}_{k,\tau} e^{iu\hat{m}_{k,\tau} - u^2\hat{\sigma}^2_{k,\tau}/2} + \hat{p}_{0,\tau} \frac{2}{(b-a)u} \sin\frac{(b-a)u}{2} \cdot e^{\frac{iu(a+b)}{2}}. \tag{3.12}$$

Pateiktoje formulėje antrasis narys aprašo tolygaus pasiskirstymo triukšmo klasterį. Galima pastebėti ir tai, kad $\hat{p}_0$ yra anksčiau aptartas triukšmo klasterio svoris, o $a = a(\tau)$, $b = b(\tau)$. Remdamiesi nustatytais tolydžiojo skirstinio parametrų įverčiais ir projektuotais duomenimis, galime užrašyti:

$$a = (\tau' x)_{\min} - \frac{(\tau' x)_{\max} - (\tau' x)_{\min}}{2(n-1)} \tag{3.13}$$

ir

$$b = (\tau' x)_{\max} + \frac{(\tau' x)_{\max} - (\tau' x)_{\min}}{2(n-1)}. \tag{3.14}$$

Įrašę

$$\hat{\psi}_\tau(u) = \sum_{k=1}^{\hat{q}_\tau} \hat{p}_{k,\tau} e^{iu\hat{m}_{k,\tau} - u^2\hat{\sigma}^2_{k,\tau}/2}. \tag{3.15}$$

į šio skyriaus 3.3 formulę gauname:

$$\hat{f}(x) = \frac{A(d)}{\#T} \sum_{\tau \in T} \left[\sum_{k=1}^{\hat{q}_\tau} \hat{p}_{k,\tau} \int_0^\infty e^{iu\left(\hat{m}_{k,\tau} - \tau' x\right) - u^2\left(h + \hat{\sigma}_{k,\tau}^2/2\right)} u^{d-1}\, du \right.$$
$$\left. + \frac{2\hat{p}_{0,\tau}}{b-a} \int_0^\infty e^{iu\left(\frac{a+b}{2} - \tau' x\right) - u^2 h} \cdot \sin\frac{(b-a)u}{2} \cdot u^{d-2} du \right]. \tag{3.16}$$

Pasinaudojus modifikuota apvertimo formule (3.16) ir EM (didžiausio tikėtinumo) algoritmu atliekamas duomenų klasterizavimas paremtas modifikuotu apvertimo formulės tankio įvertiniu. Tarkime, kad *X* skirstinys priklauso nuo atsitiktinio dydžio *v*, kuris įgyja reikšmes 1, …, *q* su atitinkamomis tikimybėmis $p_1, \ldots, p_q$. Čia *v* interpretuojamas kaip klasės, kuriai priklauso stebimas objektas, skaičius. Taigi *X(t)* stebiniai atitiktų *v(t)*, *t* = 1, …, *n*. Funkcijos f_k traktuojamos kaip sąlyginio skirstinio *X* tankis esant sąlygai *v* = *k*

$$\pi_k(x) = P\{v = k | X = x\} \tag{3.17}$$

vertinimas, kai visi $x \epsilon \{X(1), \ldots, X(n)\}$. Griežtas imties klasterizavimas būtų atsitiktinių dydžių *v*(1), …, *v*(*n*) įvertinimas, suskirstytas į poaibius remiantis lygybe:

$$\hat{v}(t) = arg \max_{k=1,\ldots,q} \hat{\pi}_k(X(t)). \tag{3.18}$$

(19)

Įverčiai $\hat{\pi}_k$ gaunami aproksimuojant nežinomus pasiskirstymo tankio komponentus inversijos formulės tankio įverčiais ir naudojant EM algoritmą. Tarkime, kad f_k yra anksčiau aptarta tankio funkcija, *k* = 1, …, *q*. Čia *q* – klasterių skaičius. Šiuo atveju (17) dešiniąją lygties pusę pažymėkime $f(x,\theta)$, čia $\theta = (p_k, M_k, R_k, a, b, k = 1, \ldots, q)$. Taikoma lygybė:

$$\pi_k(x) = \frac{p_k f_k(x)}{f(x,\theta)}. \tag{3.19}$$

Turint θ įvertį, tikimybių π_k (*k*-tosios klasterio tikimybės) įverčiai gaunami iš 3.18 formulės, dešinėje esančius nežinomus parametrus pakeičiant jų statistiniais įverčiais.

$$\theta^* = arg \max_{\theta} L(\theta) \tag{3.20}$$

$$L(\theta) = \prod_{t=1}^{n} f(X(t), \theta), \tag{3.21}$$

EM algoritmas buvo plačiai analizuotas įvairiuose apžvalginiuose straipsniuose ir monografijose [144–146]. Tarkime, kad po *r* ciklų gavome įverčius $\hat{\pi}_k = \hat{\pi}_k^{(r)}$. Tada naujas įvertis $\hat{\theta} = \hat{\theta}^{(r+1)}$ apibrėžiamas lygtimis:

$$\hat{p}_k = \frac{1}{n}\sum_{t=1}^{n} \hat{\pi}_k(X(t)), \tag{3.22}$$

$$\widehat{M}(k) = \frac{1}{n\hat{p}_k}\sum_{t=1}^{n} \hat{\pi}_k\big(X(t)\big) \cdot X(t), \tag{3.23}$$

$$\hat{R}(k) = \frac{1}{n\hat{p}_k}\sum_{t=1}^{n} \hat{\pi}_k\big(X(t)\big)\big[X(t) - \widehat{M}(k)\big] \cdot \big[X(t) - \widehat{M}(k)\big]', \tag{3.24}$$

čia $k = 1, \ldots, q$. Įrašius $\hat{\theta}^{(r+1)}$ į 3.19 formulę, randami $\hat{\pi}^{(r+1)}(X(t))$, $k = \overline{1,\ q}$, $t = \overline{1,\ n}$. Atlikus šią rekursinę procedūrą gaunama nemažėjanti seka $L(\hat{\theta}^{(r)})$, bet ar ji konverguoja, labai priklauso nuo pradinio įvertinimo $\hat{\theta}^{(0)}$ (ar $\hat{\pi}^{(0)}$). Atliekant eksperimentinius tyrimus, darbe naudojami įvairūs populiarūs klasterizavimo duomenų rinkiniai, o sukurtas duomenų klasterizavimo metodas lyginamas su kitais duomenų klasterizavimo metodais. Kiti duomenų klasterizavimo metodai buvo atrinkti remiantis populiariausiais duomenų klasterizavimo metodais, taikomais kituose moksliniuose straipsniuose.

1 algoritmas. Duomenų klasterizavimas remiantis modifikuota apvertimo formule (MIDE)

1 **Įvestis:** Duomenų rinkinys $X = [X_1, X_2, \ldots, X_n]$, klasterių skaičius K
Išvestis: $C_1, C_2, \ldots, C_t$
Inicializuojamas vidurkių vektorius:
1) atsitiktinis tolydaus skirstinio inicializavimas;
2 2) K vidurkių inicializavimas;
3) atsitiktinių taškų inicializavimas.
Generuojama *T* matrica. Aibė T apskaičiuojama, kai projektavimo kryptys yra tolygiai išdėstytos sferoje.
3 **Kol *i* = 1: *t* vykdyti**
4 Tankio įvertinimams kiekvienam taškui ir klasteriui.
Atnaujinamos $\widehat{M}$, $\hat{p}_k$, $\hat{R}$ reikšmės.
5 **Pabaiga**
6 $C_1, C_2, \ldots, C_t$ ir $\widehat{M}$, $\hat{p}_k$, $\hat{R}$

Šiame straipsnyje duomenų klasterizavimo tikslumas buvo vertinamas naudojant skirtingas metrikas, pagal kurias buvo atliekama lyginamoji analizė. J-Score [147], NMI (angl. *Normalized Mutual Information*) [148], ARI (angl. *Adjusted Rand Index*) [149] ir tikslumas (angl. *Accuracy (ACC)*) [150] ir FMI (angl. *Fowlkes-Mallows index*) [151]. Šios metrikos gali būti naudojamos, nes tyrimo metu žinomos tikrosios grupės, kurios gali būti naudojamos įvertinti. Jei klasteriai nebūtų žinomi, tada naudojamos vertinimo metrikos gali būti: Calinski ir Harabasz, dar žinoma kaip variacijos santykio kriterijus (angl. *Variance Ratio Criterion*) [152], ir Davies-Bouldin metrika [153].

Tyrime buvo naudoti dvidešimt penki skirtingi duomenų rinkiniai siekiant palyginti sukurtą duomenų klasterizavimo metodą su kitais šiuo metu taikomais metodais. Naudojamus duomenų rinkinius galima išskirti į tris atskiras grupes: sintetinius, tikrus (angl. *real*) ir generuotus su triukšmu duomenų rinkinius. Sintetiniai duomenų rinkiniai apibrėžiami kaip kitų autorių sudaryti rinkiniai, kurie dažnai naudojami duomenis klasterizuojant ar klasifikuojant, pavyzdžiui, D31, R15, Threenorm ir kiti. Šie duomenų rinkiniai buvo surinkti remiantis kitų autorių moksliniais straipsniais ir juose dažniausiai naudojamais duomenų rinkiniais. Realūs duomenų rinkiniai apibrėžiami kaip rinkiniai, sudaryti remiantis realiais pasaulio stebiniais. Tokie duomenų rinkiniai yra Iris, Wine, Diabetes ir kiti. Šie duomenų rinkiniai taip pat buvo surinkti pagal kitų autorių naudojamus duomenų rinkinius. Paskutinė duomenų rinkinių grupė – generuoti duomenų rinkiniai su triukšmo komponentu. Šie duomenys generuojami kaip pasiskirstę pagal Gauso skirstinį, įskaitant tam tikrą triukšmo procentą: 0,5 %, 1 %, 2 % ir 4 %. Šio duomenų rinkinio tikslas – įvertinti, kaip skirtingi metodai veikia su duomenimis, pasiskirsčiusiais pagal Gauso skirstinį ir turinčiais iš anksto apibrėžtą triukšmo santykį. 3 lentelėje pateikiama informacija apie naudotus duomenų rinkinius.

3 lentelė. Tyrimo duomenų rinkinių informacija (imties dydis, dimensijos, grupių skaičius).

	Duomenų rinkinys	**Imties dydis (N)**	**Dimensijos (D)**	**Grupių skaičius**
Sintetiniai				
1	Aggregation	788	2	7
2	Atom	800	3	2
3	D31	3100	2	31
4	R15	600	2	15
5	Gaussians1	100	2	2
6	Threenorm	1000	2	2
7	Twenty	1000	2	20
8	Wingnut	1016	2	2
Realūs				
9	Breast	570	30	2
10	CPU	209	6	4
11	Dermatology	366	17	6

12	Diabetes	442	10	4
13	Ecoli	336	7	8
14	Glass	214	9	6
15	Heart-statlog	270	13	2
16	Iono	351	34	2
17	Iris	150	4	3
18	Wine	178	13	3
19	Thyroid	215	5	3
Generuoti duomenų rinkiniai				
20	2 klasteriai (0,5 % išskirčių)	1005	2	2
21	2 klasteriai (1 % išskirčių)	1010	2	2
22	2 klasteriai (2 % išskirčių)	1020	2	2
23	2 klasteriai (4 % išskirčių)	1040	2	2
25	3 klasteriai (0,5 % išskirčių)	1005	2	3
26	3 klasteriai (1 % išskirčių)	1010	2	3
27	3 klasteriai (2 % išskirčių)	1020	2	3
28	3 klasteriai (4 % išskirčių)	1040	2	3

Visi eksperimentai buvo atlikti 10 000 kartų, siekiant minimizuoti galimą inicializavimo įtaką klasterizavimo rezultatų interpretacijai. Taip pat tyrimo metu buvo taikyti skirtingi duomenų klasterizavimo metodai, kurie dažnai pasirenkami moksliniuose tyrimuose atliekant duomenų klasterizavimą. Vienas populiariausių metodų, kuris įtraukiamas į visus palyginimus, yra *k* vidurkių metodas. Tyrime taip pat buvo įtraukti GMM ir BGMM metodai, nes jų veikimas yra panašesnis į kuriamų klasterizavimo metodų veikimą. 4 lentelėje pateikiami pagrindiniai straipsnio rezultatai, daugiau rezultatų pateikiama aptariamajame straipsnyje.

4 lentelė. Pagrindiniai straipsnio rezultatai. Duomenų klasterizavimo rezultatai pagal tikslumo (angl. *Accuracy*) metriką

Duomenų rinkinys	**K-vidurkių**		**GMM**		**BGMM**		**MIDEv1**		**MIDEv2**	
	Vid.	SN	Vid.	SN	Vid.	SN	Vid.	SN	Vid.	SN
Sintetiniai										
Aggregation	0,857	0,005	0,835	0,075	**0,907**	0,042	0,889	0,008	0,895	0,009
Atom	0,710	0,002	0,618	0,028	0,637	0,022	0,723	0,002	**0,746**	0,004
D31	**0,972**	0,015	0,928	0,028	0,601	0,022	0,721	0,017	0,723	0,013
R15	**0,997**	0,000	0,979	0,036	0,669	0,011	0,768	0,008	0,855	0,007
Gaussians1	1,000	0,000	1,000	0,000	1,000	0,000	1,000	0,000	**1,000**	0,000
Threenorm	0,591	0,001	0,612	0,047	0,549	0,006	0,649	0,003	**0,679**	0,003
Twenty	1,000	0,000	0,985	0,029	0,838	0,075	–	–	–	–
Wingnut	0,909	0,000	0,964	0,000	**0,965**	0,000	0,876	0,000	0,880	0,000
Realūs										
Breast	0,908	0,003	**0,940**	0,001	0,933	0,001	–	–	–	–
CPU	0,738	0,008	0,574	0,073	0,590	0,093	0,808	0,007	**0,828**	0,006
Dermatology	0,739	0,044	0,737	0,080	**0,756**	0,109	–	–	–	–
Diabetes	0,356	0,010	0,419	0,043	0,439	0,033	0,420	0,008	**0,448**	0,007
Ecoli	0,649	0,013	0,753	0,018	0,739	0,006	0,714	0,011	**0,754**	0,009
Glass	0,447	0,016	0,468	0,025	0,483	0,025	0,465	0,013	**0,487**	0,017

Heart-statlog	**0,837**	0,002	0,794	0,045	0,791	0,045	–	–	–	–
Iono	0,707	0,000	**0,810**	0,029	0,803	0,023	–	–	–	–
Iris	0,831	0,007	0,953	0,065	0,838	0,049	0,933	0,006	**0,955**	0,005
Wine	0,966	0,000	0,953	0,048	**0,977**	0,038	0,943	0,003	0,953	0,004
Thyroid	0,874	0,000	**0,953**	0,029	0,917	0,035	0,754	0,007	0,778	0,009
Generuoti duomenys su išskirtimis										
2 klasteriai (0,5 % išskirčių)	0,995	0,000	0,995	0,000	0,995	0,000	0,995	0,000	**1,000**	0,000
2 klasteriai (1 % išskirčių)	0,989	0,000	0,990	0,000	0,990	0,000	0,990	0,000	**0,996**	0,000
2 klasteriai (2 % išskirčių)	0,979	0,000	0,980	0,000	0,980	0,000	0,981	0,000	**0,997**	0,000
2 klasteriai (4 % išskirčių)	0,961	0,000	0,962	0,000	0,962	0,000	0,964	0,000	**0,996**	0,000
3 klasteriai (0,5 % išskirčių)	0,994	0,000	0,994	0,000	0,994	0,000	0,994	0,000	**0,999**	0,000
3 klasteriai (1 % išskirčių)	0,989	0,000	0,989	0,000	0,989	0,000	0,989	0,000	**0,997**	0,000
3 klasteriai (2 % išskirčių)	0,979	0,000	0,979	0,000	0,979	0,000	0,981	0,000	**0,997**	0,000
3 klasteriai (4 % išskirčių)	0,961	0,000	0,951	0,000	0,945	0,000	0,965	0,000	**0,996**	0,000

Išvados. Šiame darbe buvo pateikta naujo duomenų klasterizavimo metodo idėja, paremta modifikuotos apvertimo formulės tankio robastiniu įvertiniu. Šiame tyrime buvo sukurtas, išbandytas ir palygintas naujas duomenų klasterizavimo metodas. Galima pastebėti, kad naujasis metodas, pagrįstas modifikuota apvertimo formule, gana gerai veikia su skirtingais duomenų rinkiniais, palyginti su *k* vidurkių, Gauso mišinio modeliu ir Bajeso Gauso mišinio modeliu Tyrimo metu buvo pastebėti ir tam tikri sukurto duomenų klasterizavimo metodo apribojimai. Šis metodas sunkiai veikia esant didesnėms duomenų dimensijoms ($d > 15$), nes susiduriama su matricos T generavimo problemomis. Didėjant dimensijų skaičiui pastebima, kad atmetama vis daugiau reikšmių, nes daugiamatė sfera artėja prie hiperkubo, pagal kurį atrenkamos reikšmės. Vienas iš sprendimų galėtų būti krypties parinkimas iš Gauso skirstinių ir tuomet reikšmių normalizavimas. Tyrimo rezultatai parodė, kad MIDEv2 metodas geriausiai veikia, kai duomenų rinkiniai turi triukšmą (0,5 %, 1 %, 2 %, 4 % triukšmo). Verta pažymėti, kad naujas metodas, pagrįstas apvertimo formule, gali sugrupuoti duomenis, net jei jie neturi triukšmo ar nuokrypių. Pavyzdžiui, vieno populiariausių Iris duomenų rinkinių grupavimo tikslumas yra didesnis lyginant su kitais metodais.

3.3. Straipsnio "Reduced Clustering Method Based on the Inversion Formula Density Estimation" apžvalga

M. Lukausko įnašas į straipsnį: M. Lukauskas yra straipsnio kontaktinis asmuo (angl. *corresponding author*), sudarė duomenų klasterizavimo metodą, pagrįstą apvertimo formulės tankio įvertiniu ir duomenų dimensijų mažinimo metodais, atliko eksperimentinius skaičiavimus, parengė straipsnį, bendraudamas su bendraautoriais, recenzentais, redaktoriumi atliko straipsnio taisymus.

Straipsnio apibendrinimas. Šiame moksliniame straipsnyje tęsiamas ankstesnio mokslinio straipsnio įdirbis, kuriant robastinius duomenų klasterizavimo metodus, pagrįstus modifikuotos apvertimo formulės tankio įvertiniu. Anksčiau pateiktame mokslinime straipsnyje buvo pastebėta, kad naujai sukurtas robastinis klasterizavimo metodas (CBMIDE) pasižymi apribojimais, kai duomenų matmenys yra didesni ($d > 10$). Šiame moksliniame straipsnyje siūlomas CBMIDE (angl. *Clustering based on modified inversion formula density estimation*) metodo patobulinimas didesnių dimensijų duomenims, kad metodas veiktų su didesniais matmenimis ir (arba)

padidintų metodo tikslumą mažesnio matmens atveju. Siekiant įgyvendinti šį tikslą, moksliniame straipsnyje tiriama duomenų dimensijų mažinimo metodų įtaka duomenų klasterizavimo rezultatams, naudojant modifikuotą tankio įvertinio formulę. Duomenų dimensijos dažnai lemia klasterizavimo metodų tikslumą, skaičiavimo laiką ir reikalingus išteklius. Įprastu atveju pastebima, kad daugėjant duomenų dimensijų, klasterizavimo laikas taip pat žymiai padidėja. Tokiu atveju vienas iš sprendimų – sumažinti duomenų dimensijas. Tai galima padaryti taikant įvairius matmenų mažinimo metodus. Duomenų matmenų mažinimo derinimas su klasterizavimu yra gana dažnas, nes taip sutaupoma daug skaičiavimo išteklių. Duomenų matmenų mažinimas ir sumažintų dimensijų taikymas klasterizuojant taip pat aptartas mokslinėje literatūroje, kur pirmą kartą buvo sujungti paprasčiausi *K* vidurkių ir PCA metodai [52]. Studijos, jungiančios duomenų dimensijų mažinimą ir duomenų klasterizavimą, neprarado populiarumo ir dabar. Pagrindinė to priežastis – kasmet didėjantis duomenų kiekis. Gana populiari šių metodų kombinacijos derinimo sritis yra genų analizė. Duomenų matmenų mažinimo metodai, tokie kaip pagrindinės komponentės (angl. *principal components*, PCA) [53], neneigiamų matricų faktorizavimas (angl. *non-negativve matrix factorization,* NMF) [54], nepriklausomų komponenčių analizė (angl. *independant component analysis,* ICA) [55], ir klasterizavimo metodai, tokie kaip *K* vidurkių, DBSCAN ir kiti, naudojami genų sekoms tirti [56]. Taip pat pastebimi sudėtingesni metodai, leidžiantys sumažinti duomenų matmenis, pvz., t-SNE [57], UMAP [58] ir įvairūs metodų deriniai [59–64].

Šiame moksliniame darbe buvo modifikuotas robastinio duomenų klasterizavimo metodas CBMIDE, siekiant jį pritaikyti didelių dimensijų duomenims klasterizuoti. Straipsnyje buvo keliama hipotezė, kad duomenų klasterizavimo tikslumas yra toks pats arba netgi geresnis sumažinus duomenų matmenis matmenų mažinimo metodais ir kad modifikuotas CBMIDE metodas turi pranašumą, palyginti su ankstesniu metodu ir kitais populiariais metodais. Norint palyginti rezultatus, šiame darbe naudojamos skirtingos duomenų dimensijų mažinimo metodų ir klasterizavimo metodų kombinacijos.

Didžiausias anksčiau pasiūlyto metodo CBMIDE trūkumas yra jo pritaikymas didelių matmenų duomenims. Norėdami naudoti apvertimo formulę, pirmiausia pasirenkame projektavimo krypčių skaičių (aibę *T*), kuris turi būti tolygiai paskirstytas vienetinėje sferoje. Didėjant matmenims, yra sunku rasti projektavimo kryptis vienetinėje sferoje, kadangi dalis generuojamų reikšmių yra atmetamos, nes neatitinka iš anksto apibrėžtų sąlygų. Didėjant dimensijų skaičiui pastebima tai, kad atmetama vis daugiau reikšmių, tai atsitinka dėl to, kad daugiamatė sfera artėja prie hiperkubo, pagal kurį atrenkamos reikšmės. Todėl sunku sukurti reikiamą T matricą. Dėl šios priežasties reikia tobulinti CBMIDE metodo, leidžiantį taikyti jį didelių matmenų duomenims. Šiame darbe siūloma metodą išplėsti – duomenų matmenis sumažinti pirmajame klasterizavimo etape.

$$Z = f(X, d), \tag{3.25}$$

čia Z – sumažintų matmenų duomenys latentinėje erdvėje; X – pradiniai duomenys pradinėje erdvėje; d – matmenų skaičius. Toliau duomenys klasterizuojami naudojant sumažintų dimensijų duomenis. Duomenų dimensijų mažinimas leidžia atskirti

perteklines stebinių charakteristikas, sumažinant duomenų multikolinearumo problemą. Duomenų matmenų sumažinimas taip pat leidžia lengviau vizualizuoti duomenis, kai dimensijos sumažinamos iki dviejų ar trijų, kurias galima pavaizduoti vizualiai. Duomenų dimensijų mažinimas gali būti atliktas pasirenkant savybes iš visos savybių aibės arba metodus, kurie sudaro naujus matmenis, geriausiai atspindinčius pradinius duomenis. Norint išplėsti pradinį CBMIDE, naujame algoritme buvo įdiegtas duomenų dimensijų mažinimas, kuris gali būti atliekamas įvairiais duomenų dimensijų mažinimo metodais. Analizuojant ir lyginant šių metodų rezultatus buvo integruoti šie duomenų dimensijų mažinimo metodai:

1) pagrindinių komponenčių analizė (PCA);
2) nepriklausomųjų komponenčių analizė (ICA);
3) faktorinė analizė (FA);
4) *T* paskirstytas stochastinis kaimynų įterpimas (t-SNE);
5) TriMap;
6) vienodas kolektoriaus aproksimavimas ir projekcija (UMAP);
7) ISOMAP;
8) daugiamatis mastelio keitimas (MDS);
9) vietinis tiesinis įterpimas (LLE).

Skirtingi variantai naudojami siekiant nustatyti, kuris iš patobulinimų duoda geriausius klasterizavimo rezultatus, modifikuojant bendrą duomenų klasterizavimo metodą, pagrįstą modifikuotu apvertimo formulės tankio įvertiniu.

Siekiant tinkamai įvertinti duomenų dimensijų mažinimo tikslumą, šiame algoritme naudojama duomenų dimensijų mažinimo patikimumo metrika – *trustworthiness* [154]. Ji apskaičiuojama pagal toliau pateiktą formulę:

$$T(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{j \in N_i^k} \max\left(0, (r(i,j) - k)\right) \qquad (3.26)$$

Čia kiekvienam stebiniui i, N_i^k yra jo k artimiausių kaimynų (išvesties erdvėje po pritaikyto duomenų dimensijų mažinimo metodo) ir kiekvienam stebiniui j, $r(i,j)$-tasis artimiausias kaimynas įvesties erdvėje. Jei išvesties duomenų erdvėje nustatomi duomenų stebiniai, kurie neturėtų būtų prie kitų stebinių, jie yra „baudžiami“. Šiame tyrime naudojamas skirtingas kaimynų skaičius vertinant šią metriką (5, 10, 15, 20). Bendras duomenų klasterizavimo algoritmas pateikiamas toliau.

2 algoritmas. RCBMIDE duomenų klasterizavimo algoritmas

1 **Įvestis:** Duomenų rinkinys X= [X_1, X_2, …, X_d], klasterių skaičius K, glodumo parametras h, išskirčių procentinis santykis p_0, iteracijų skaičius t, dimensijų skaičius d, duomenų dimensijų mažinimo metodas, artimiausių kaimynų skaičius naudojamas patikimumo metrikai apskaičiuoti.

2 **Išvestis:** C_1, C_2, ..., C_t

Įvestis duomenų dimensijoms mažinti:

Kol $j = 2$: $d < 15$:

Mažinti duomenų dimensijas pasirinktu duomenų dimensijų mažinimo metodu. Apskaičiuoti patikimumo metriką sumažintiems duomenims (13)

Pasirinkti duomenų rinkinio dimensijas remiantis duomenų dimensijų mažinimo žingsniu.

Sudaromas pradinis vidurkių vektorius naudojant *k* vidurkių ar *k* vidurkių ++ metodą.

Sudaroma matrica *T*. Matrica *T* apskaičiuojama, kai projektavimo kryptys yra tolygiai išdėstytos sferoje.

Kol $i = 1$: t vykdyti

Tankio apskaičiavimas kiekvienam duomenų stebiniui.

Atnaujinamos $\widehat{M}$, $\hat{p}_k$, $\hat{R}$ reikšmės

Pabaiga

Grąžinti C_1, C_2, ..., C_t ir $\widehat{M}$, $\hat{p}_k$, $\hat{R}$

Šiame tyrime buvo naudojama 20 skirtingų duomenų rinkinių, siekiant įvertinti duomenų klasterizavimo metodų tikslumą skirtingais duomenų atvejais bei metodų apibendrinimo galimybes. Taip pat skirtingi duomenų standartizavimo metodai buvo taikyti įvertinant šių metodų įtaką duomenų klasterizavimo rezultatams: nestandartizuoti duomenys (angl. *Raw*), minimalios ir maksimalios reikšmės normalizavimas (angl. *MinMax*), standartizavimas (angl. Standard), nejautrus išskirtims normalizavimas (angl. *Robust*), absoliučiosios maksimalios reikšmės normalizavimas (angl. *Max-Abs*), kvantilių normaliojo skirstinio normalizavimas (angl. *QuantileNormal*), kvantilių tolydus normalizavimas (angl. *QuantileUniform*), galios transformavimas (angl. *PowerTransformer*) ir *Scikit-learn Normalizer* metodas. Šiuo atveju kiekvienas stebinys (t. y. kiekviena duomenų matricos eilutė) su bent vienu komponentu, kuris nėra nulis, perskaičiuojamas nepriklausomai nuo kitų imčių, kad jo norma (l1, l2 arba inf) būtų lygi vienetui. Visas aprašymas pateikiamas aptariamajame straipsnyje. Tyrimo metu gauti svarbiausi rezultatai pateikiami tolesnėse lentelėse.

Duomenų analizė buvo suskirstyta į dvi atskiras dalis: mažų dimensijų duomenų klasterizavimą ($d < 10$) ir didelių dimensijų duomenų klasterizavimą. Atsižvelgiant į tai, kad robastinis duomenų klasterizavimo metodas, pristatytas ankstesniuose straipsniuose, CBMIDE kenčia nuo didelių dimensijų problemos, pirmiausia klasterizavimo metodai buvo lyginti naudojant mažų dimensijų duomenų rinkinius. Gauti rezultatai parodė, kad duomenų klasterizavimo tikslumas, naudojant modifikuotą metodą RCBMIDE, yra geresnis, palyginti su CBMIDE metodu. Šie rezultatai patvirtino straipsnyje iškeltą idėją apie modifikuoto klasterizavimo metodo pranašumus prieš pradinį duomenų klasterizavimo metodą. Taip pat pastebima, kad, lyginant su platesniu skirtingų metodų spektru, RCBMIDE pateikė geriausius duomenų klasterizavimo rezultatus net su 5 duomenų rinkiniais iš 11. Kitais atvejais tikslumo rezultatai yra artimi geriausiems atitinkamo duomenų rinkinio klasterizavimo rezultatams. Pavyzdžiui, „diabetes“ duomenų rinkinyje geriausias

rezultatas buvo gautas taikant BIRCH metodą (0,514), o RCBMIDE rezultatai parodė 0,512 tikslumą, patvirtinantį idėją apie modifikuoto metodo taikymą duomenims klasterizuoti. Geriausi rezultatai buvo pasiekti daugiausia naudojant 6-ąją modifikaciją (TriMap) ir 5-ąją modifikaciją (UMAP).

5 lentelė. Duomenų klasterizavimo tikslumo (angl. *Accuracy*) rezultatai mažų dimensijų duomenų rinkiniams.

Duomenų	Duomenų klasterizavimo metodas								
rinkinys	Agg	BIRCH	GMM	BGMM	DBSCAN	K-means	HDBSCAN	CBMIDE	RCBMIDE
1balance	0,624	0,658	0,568	0,586	0,464	0,603	0,597	0,576	**0,693**
atom	**1,000**	0,868	0,883	0,960	**1,000**	0,719	**1,000**	0,891	**1,000**
cpu	0,823	0,828	0,746	0,641	0,833	0,761	0,813	0,815	**0,858**
diabetes	0,507	**0,514**	0,459	0,455	0,482	0,428	0,477	0,502	0,512
ecoli	0,804	**0,845**	0,762	0,747	0,682	0,688	0,646	0,754	0,817
glass	0,514	0,565	0,509	0,528	0,514	0,547	0,528	0,527	**0,607**
Haberman	0,748	**0,761**	0,667	0,716	0,758	0,748	0,739	0,735	0,742
iris	0,967	0,973	0,967	0,893	0,94	0,967	0,700	0,975	**0,983**
pmf	0,977	0,977	0,92	0,977	**0,983**	0,844	0,978	0,934	0,981
thyroid	0,93	0,949	**0,963**	0,949	0,874	0,944	0,823	0,778	0,834
Wine	0,978	**0,994**	0,972	0,983	0,949	0,978	0,876	0,953	0,963

Paryškintos ir pabrauktos reikšmės rodo geriausius kiekvieno duomenų rinkinio tikslumo mato rezultatus.

Toliau straipsnyje nagrinėjami metodai esant didelėms duomenų dimensijoms, todėl svarbu pažymėti, kad tolesniame tyrime pradinis CBMIDE metodas nebebuvo įtrauktas į palyginimą, nes to padaryti nebuvo įmanoma dėl šio metodo nepritaikymo didelių dimensijų duomenims. Tokie rezultatai parodo, kad, siekiant panaudoti duomenų klasterizavimą, pagrįstą modifikuotu apvertimo formulės tankio įvertiniu, tokios robastinio klasterizavimo metodo modifikacijos buvo reikšmingos. Gauti rezultatai patvirtino, kad sudarytas naujas RCBMIDE duomenų klasterizavimo metodas pasižymėjo geriausiais duomenų klasterizavimo rezultatais „german“ ir „segment“ duomenų rinkiniams. Šiuo atveju geriausios metodo modifikacijos buvo 8-oji modifikacija (daugiamatė skalė (MDS)) ir 6-oji modifikacija (TriMap). Detalesnis Duomenų rezultatai nuodugniau aprašomi ir lyginami aptariamajame straipsnyje.

6 lentelė. Duomenų klasterizavimo tikslumo mato (angl. *Accuracy*) reikšmės didelių dimensijų duomenų rinkiniams

Duomenų	Duomenų klasterizavimo metodas							
rinkinys	Agg	BIRCH	GMM	BGMM	DBSCAN	K-means	HDBSCAN	RCBMIDE
arrhythmia	**0,600**	0,571	0,485	0,431	0,573	0,438	0,582	0,597
Breast	0,942	**0,954**	0,951	0,953	0,903	0,928	0,743	0,909
Coil20	0,738	0,738	0,638	0,675	0,867	0,733	**0,884**	0,882
dermatology	0,956	**0,978**	0,91	0,855	0,694	0,962	0,809	0,867
german	0,705	0,713	0,696	0,638	0,712	0,673	0,704	**0,716**

heart-statlog	0,807	0,811	0,819	0,826	0,815	**0,848**	0,626	0,831
iono	0,729	0,795	0,849	0,809	**0,929**	0,712	0,906	0,899
segment	0,708	0,732	0,631	0,612	0,529	0,665	0,530	**0,789**
spambase	0,878	**0,918**	0,856	0,857	0,693	0,854	0,690	0,905
wdbc	0,942	0,954	0,951	**0,958**	0,903	0,928	0,743	0,951

Paryškintos ir pabrauktos reikšmės rodo geriausius kiekvieno duomenų rinkinio tikslumo mato rezultatus.

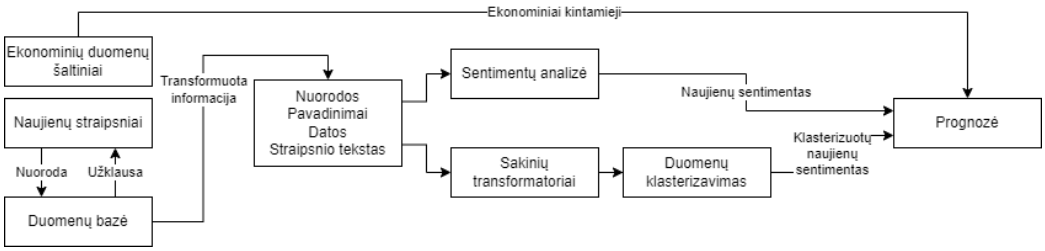
Išvados. Šiame moksliniame straipsnyje pristatytas modifikuotas robastinis duomenų klasterizavimo metodas (RCBMIDE), siekiant išvengti ankstesnio metodo (CBMIDE) duomenų didelių dimensijų problemos. Siekiant tai įgyvendinti buvo modifikuotas duomenų klasterizavimo algoritmas, kuriame pasiūlomas duomenų dimensijų mažinimas, iš dalies padedantis išspręsti didelių duomenų dimensijų problemą, su kuria susiduria CBMIDE metodas. Duomenų dimensijoms mažinti buvo taikomi skirtingi metodai, kurie tapo atskiromis klasterizavimo metodo modifikacijomis. Naujo metodo taikymas duomenims klasterizuoti parodė, kad modifikuotas metodas pasižymi geresniais rezultatais, palyginti su pradiniu duomenų klasterizavimo metodu. Palyginus su kitais duomenų klasterizavimo metodais, sukurtas RCBMIDE pasižymėjo gerais rezultatais. Naudojant mažų dimensijų duomenų rinkinius, modifikuoto klasterizavimo metodas buvo palygintas su pradiniu duomenų klasterizavimo metodu. Buvo pastebėta, kad naujas duomenų klasterizavimo metodas (RCBMIDE) pasižymi geresniais rezultatais net esant mažų dimensijų duomenims, palyginti su ankstesniu metodu. Dėl šios priežasties galima teigti, kad hipotezė apie metodo modifikavimo įtaką geresniems klasterizavimo rezultatams buvo patvirtinta. Taip pat verta paminėti, kad visais atvejais taikyti duomenų matmenų mažinimo metodai yra skirtingi. Patikimumo metrika (angl. *Trustworthiness*) leido palyginti, kaip sėkmingai buvo sumažintas matmenų dydis (dimensijos). Taip pat pastebima, kad UMAP ir TriMap metodai, kurie veikia panašiu principu, dažniausiai duoda patikimiausią duomenų sumažinimo rezultatą mažų matmenų atveju. Šis darbas įrodė iškeltą hipotezę apie sukurto metodo išplėtimą naudojant duomenų dimensijų mažinimo metodus ir leido pritaikyti sukurtą duomenų klasterizavimo metodą didelių dimensijų duomenims.

3.4. Straipsnio “Economic Activity Forecasting Based on the Sentiment Analysis of News” apžvalga

M. Lukausko įnašas į straipsnį: M. Lukauskas yra straipsnio kontaktinis asmuo (angl. *corresponding author*) ir pirmasis straipsnio autorius. Autorius surinko duomenų rinkinį, konceptualizavo duomenų klasterizavimo pritaikymo idėją vykdomiems tyrimams, atlikto eksperimentinius tyrimus, parengė straipsnį, bendraudamas su bendraautoriais, recenzentais, redaktoriumi atliko straipsnio taisymus.

Straipsnio apibendrinimas. Šiame straipsnyje buvo pereita nuo teorinių tyrimų ir metodų kūrimo prie šių metodų taikymo sprendžiant realius uždavinius. Pateikiamas straipsnis yra dalis Europos regioninės plėtros fondų finansuojamo projekto (No. 13.1.1-LMT-K-718-05-0012), kuris buvo finansuotas kaip atsakas į

COVID-19 pandemiją. Duomenų klasterizavimas taikomas ypač plačiai ir šiuo atveju duomenų klasterizavimo metodai buvo pritaikyti prognozuojant ekonominį aktyvumą (atskirus ekonominio aktyvumo rodiklius). Vertinant ekonominį šalies aktyvumą, jis dažniausiai susiejamas su bendruoju vidaus produktu ar pramonės gamybos pokyčiais, nes leidžia įvertinti šalies pramonėje / gamyboje vykstančius veiksmus [155–157]. Nagrinėjamame straipsnyje duomenų klasterizavimo metodai taikomi kartu su natūralios kalbos apdorojimo (NLP) metodais. Pirmajame tyrimo etape buvo surinkta ypač daug tekstinių duomenų, kurie saugomi duomenų bazėje, tuomet skirtingais metodais atliekama tekstinių duomenų sentimentų analizė. Duomenų klasterizavimas šiuo atveju padeda tuo, kad duomenys gali būti suskirstyti į iš anksto nežinomas grupes / kategorijas, kas padeda geriau interpretuoti turimus rezultatus. 1 pav. pateikiama apibendrinta tyrimo schema.



1 pav. Pagrindinė taikomojo tyrimo schema

Šiame tyrime buvo naudojami tekstiniai duomenys, todėl jų pradiniam apdorojimui panaudoti transformerių struktūros modeliai. Duomenys į skaitines vertes buvo paversti naudojant „sentence-transformers“ modelį „all-MiniLM-L6-v2“. Šis modelis neapdorotą tekstinę informaciją transformuoja į 384 matmenų tankią vektorinę erdvę. Transformuoti skaitinio pavidalo duomenys gali būti naudojami duomenims klasterizuoti, atliekant semantinę paiešką ir kitas užduotis. Įvertinant skirtingų tekstinių duomenų sentimentus nemažą reikšmę tam gali turėti ir duomenų rinkinys, su kuriuo buvo apmokyti naudojami modeliai. Dėl šios priežasties sentimentų analizės metu buvo nuspręsta naudoti ne vieną specifinį modelį, tačiau skirtingų modelių derinį. Šiame tyrime nustatant teksto sentimentą buvo naudojami 4 skirtingi iš anksto apmokyti modeliai: „DistilBERT-base-uncased“, „FinBERT“, „Twitter-roBERTa-base“, „FinBERT-tone“. „DistilBERT-base-uncased“ modelis yra sumažinta „Bert-base-uncased“ modelio versija, tačiau pasižyminti ypač aukštais rezultatais. Šis modelis yra apmokytas naudojant SST-2 duomenų rinkinį ir jo tikslumas šiam duomenų rinkiniui siekia 91,3 procentus. „Prosus“ sukurtas „FinBERT“ modelis, skirtas būtent finansinių tekstų analizei [158]. Šis modelis iš tiesų yra „BERT“ modelis, tačiau jo apmokymui buvo naudojami būtent finansiniai tekstiniai duomenys, kurie šiam modeliui leidžia geriau nustatyti sentimentus tekstuose, susijusiuose su finansine informacija. Modeliui apmokyti buvo naudojami „Financial PhraseBank“ duomenys [159]. Kitas tyrime naudojamas modelis yra „FinBERT-tone“. Jam apmokyti taip pat naudojami būtent finansinės informacijos tekstiniai duomenys [160]. Šis modelis apmokytas naudojant net tris skirtingus finansinės tekstinės informacijos rinkinius. Bendras žodyno dydis yra 4,9 B žodžių (angl. *tokens*): įmonių ataskaitos 10-K ir 10-Q: 2,5 mlrd. žodžių, uždarbio nuorašai: 1,3 mlrd. žetonų, analitikų ataskaitos: 1,1 mlrd. žetonų. Tuomet būtent „FinBERT-

tone“ modelis yra apmokytas su rankiniu būdu sužymėtais duomenimis. Šiuo modeliu pasiekiamas geresnis našumas, atliekant finansinio duomenų analizės užduotį. Paskutinis modelis, tačiau ne ką mažiau svarbus šiame tyrime yra „Twitter-roBERTa-base“ modelis, skirtas būtent sentimentų analizei [161]. Šis modelis yra apmokytas naudojant net 124 milijonus „Twitter“ žinučių, kurios buvo surinktos net trejų metų laikotarpiu. Šis modelis leidžia įvertinti sentimentus ne tik finansiniams duomenims, bet ir bendriems tekstams.

Šiame tyrime siekiama suskirstyti visus naujienų tekstus į grupes, kad būtų galima nustatyti šių grupių sentimentus. Šiam tikslui naudojama klasterinė analizė, kurios metu buvo panaudotas populiariausias duomenų klasterizavimo metodas *k* vidurkių bei ankstesniuose straipsniuose pristatytas CBMIDE robastinis klasterizavimo metodas. Siekiant nustatyti tinkamiausią klasterizavimo metodą, būtent analizuojamiems duomenims naudojami ir kiti klasterizavimo metodai: Gauso mišinių metodas (angl. *Gaussian Mixture Models*), Bajeso Gauso mišinių metodas (angl. *Bayesian Gaussian Mixture Models*), DBSCAN (angl. *Density-Based Spatial Clustering of Applications with Noise*) [57], BIRCH (angl. *Balanced Iterative Reducing and Clustering using Hierarchies*) [162], OPTICS (angl. *Ordering Points To Identify the Clustering Structure*) [61]. Kaip galima pastebėti, į analizę įtraukti ir kiti populiariausi robastiniai duomenų klasterizavimo metodai (kurie yra atsparūs triukšmo įtakai). Šie skirtingi modeliai apmokomi keičiant jų parametrus ir taip siekiant nustatyti geriausius klasterizavimo modelius. Pirmuoju naujienų klasterizavimo žingsniu visa tekstinė informacija buvo paversta skaitine informacija naudojant „sentence-transformers“. Naudojant „sentence-transformers“ tekstiniai duomenys paverčiami 384 dimensijos duomenimis, kur kiekvienas tekstas atitinka tam tikrą tašką šioje erdvėje pagal sakinyje esančius žodžius, jų reikšmes, semantinę prasmę. Tuomet šie taškai klasterizuojami remiantis skirtingais klasterizavimo metodais, gauti rezultatai palyginami remiantis skirtingomis metrikomis, aptartomis ankstesniame skyriuje. Metodai, pasižymintys geriausiais rezultatais, naudojami tolesniame tyrime.

7 lentelė. Skirtingų duomenų klasterizavimo metodų palyginimas naudojant Calinski-Harabasz ir Davies-Bouldin metrikas (100 bandymų)

	Calinski ir Harabasz metrikos reikšmė	Davies-Bouldin metrikos reikšmė
GMM	11648	5,841
BGMM	13547	6,148
K-means	13387	4,627
MIDE	12542	5,314

Remiantis gautais rezultatais galima pastebėti tai, kad vertinant Davies-Bouldin metriką sukurtas klasterizavimo metodas, analizuojant realius duomenis, pasižymėjo antru geriausiu rezultatu. Toliau straipsnyje pateikiama informacija apie skirtingų ekonominių rodiklių prognozavimą remiantis klasterizavimu bei kitomis prognozavimo variacijomis. Remiantis straipsnyje gautais rezultatais pastebima, kad pritaikius klasterizuotų naujienų sentimentų analizės indeksą prognozuojant skirtingus rodiklius prognozavimo paklaidos nėra geresnės nei kitais atvejais. Dėl šios

priežasties nors klasterizavimas ir pasižymėjo gana gerais rezultatais, tačiau nepadėjo geriau prognozuoti ekonominio aktyvumo rodiklių.

8 lentelė. Ekonominio aktyvumo prognozavimo rezultatai remiantis vienmatėmis laiko eilutėmis, kategorijų sentimentų laiko eilutėmis ir klasterizuotų žinių laiko eilutėmis

Duomenys/modelis	**RMSE**		**MAE**		**MAPE**	
	Vidurkis	SN	Vidurkis	SN	Vidurkis	SN
Jaunimo nedarbo lygis						
Vienmatis	**0,038**	0,010	**0,027**	0,006	**0,042**	0,009
Verslo sentimentas	0,119	0,017	0,105	0,018	0,336	0,052
Visi sentimentai	0,207	0,041	0,186	0,040	0,435	0,061
Didžiausias klasteris	0.221	0.045	0.196	0.043	0.458	0.068
Bendras nedarbo lygis						
Vienmatis	**0,045**	0,009	**0,035**	0,006	**0,047**	0,008
Verslo sentimentas	0,121	0,023	0,110	0,025	0,338	0,061
Visi sentimentai	0,264	0,023	0,256	0,025	0,533	0,040
Didžiausias klasteris	0.267	0.035	0.278	0.037	0.576	0.045
Vartotojų pasitenkinimas						
Vienmatis	0,079	0,010	0,057	0,007	0,079	0,009
Bendras sentimentas	**0,048**	0,005	**0,040**	0,003	**0,066**	0,006
Visi sentimentai	0,119	0,022	0,095	0,021	0,138	0,036
Didžiausias klasteris	0,125	0,023	0,098	0,023	0,147	0,037
Mėnesinis infliacijos lygis						
Vienmatis	0,187	0,010	0,143	0,008	0,340	0,015
Bendras sentimentas	**0,123**	0,003	**0,093**	0,002	**0,255**	0,008
Visi sentimentai	0,209	0,010	0,162	0,006	0,365	0,032
Didžiausias klasteris	0,208	0,013	0,158	0,007	0,356	0,045
Metinis infliacijos lygis						
Vienmatis	0,087	0,018	0,065	0,013	0,298	0,030
Bendras sentimentas	**0,062**	0,007	**0,053**	0,006	**0,283**	0,035
Visi sentimentai	0,106	0,059	0,081	0,043	0,285	0,117
Didžiausias klasteris	0,156	0,068	0,098	0,056	0,305	0,158
Produkcijos indeksas						
Vienmatis	0,214	0,011	0,177	0,014	0,402	0,031
Lietuvos sentimentas	**0,099**	0,003	**0,076**	0,004	**0,166**	0,010
Visi sentimentai	0,112	0,011	0,086	0,010	0,171	0,014
Didžiausias klasteris	0,156	0,023	0,105	0,021	0,205	0,026

Išvados. Šiame moksliniame straipsnyje buvo pristatytas mokslinis tyrimas, kurio metu buvo atliktas duomenų klasterizavimas ir panaudotas ekonominio aktyvumo rodikliams prognozuoti. Šiame tyrime gali būti panaudota tik dalis

numatytų klasterizavimo modelių, tačiau tai yra dažniausiai naudojami modeliai praktikoje. Tai leido mums įvertinti klasterizavimo poveikį naujienų sentimentų analizei ir ekonominių rodiklių prognozavimui. paremtam sentimentų rodikliu ir duomenų klasterizavimu. Kitas svarbus veiksnys ir šio darbo apribojimas yra tai, kad darbe buvo panaudoti straipsnių pavadinimai ir jų santraukos, bet ne visa straipsnio struktūra. Taip pat šiuo metu kuriamas lietuviškas sentimentų analizės modelis, kuriam nebereikėtų papildomo tekstų vertimo, o neigiamoms nuotaikoms išgauti būtų galima naudoti neapdorotus tekstus. Apibendrinus kitus tyrimo metu gautus rezultatus pastebėta, kad neigiamas naujienų sentimentas susijęs su ekonomine veikla, leidžia tiksliau prognozuoti įvairius ekonominio aktyvumo rodiklius ir kad duomenų klasterizavimas prisideda prie tikslesnio prognozavimo lyginant su vienmačiu modeliu.

3.5. Straipsnio "Enhancing Skills Demand Understanding through Job Ad Segmentation using NLP and Clustering Techniques" apžvalga

M. Lukauskas įnašas į straipsnį: M. Lukauskas yra pirmasis straipsnio autorius ir kontaktinis straipsnio asmuo. Autorius sudarė duomenų rinkinį, konceptualizavo duomenų klasterizavimo pritaikymo idėją vykdomiems tyrimams, atliko eksperimentinius tyrimus, parengė straipsnį, bendraudamas su bendraautoriais, recenzentais, redaktoriumi atliko straipsnio taisymus.

Straipsnio apibendrinimas. Šiame straipsnyje buvo tęsiamas sukurtų metodų praktinis pritaikymas sprendžiant realias užduotis. Pateikiamas straipsnis yra dalis Europos regioninės plėtros fondų finansuojamo projekto (Nr. 13.1.1-LMT-K-718-05-0012), kuris buvo finansuotas kaip atsakas į COVID-19 pandemiją. Šiame straipsnyje pristatomas automatizuotas būdas, siekiant išsiaiškinti reikalingus darbuotojų įgūdžius darbo rinkoje. Skaitmeninimas verslui ir darbuotojams kelia daug iššūkių, anksčiau žmonių atliekamas užduotis, tokias kaip duomenų įvedimas, apskaita ir darbas prie konvejerio, dabar palaiko skaitmeninės technologijos ir dirbtinis intelektas [1,2]. Paskutiniais metais pastebimas ypač didelis įmonių skaitmenizavimo tempas, pritaikant įvairius IT sprendimus įmonių veikloje, taip didinant įmonės konkurencinį pranašumą. Tačiau kartu kyla ir naujų iššūkių. Vienas iš jų – naujų profesijų ir įvairių naujų įgūdžių, savybių paklausa [5]. Esant tokiam dideliam skaitmenizavimo tempui įmonės susiduria su kvalifikuotų darbuotojų trūkumu. Šių darbuotojų pagrindinė užduotis yra diegti, palaikyti ir prižiūrėti naujausias technologijas organizacijoje, tačiau tam reikalingi ir specifiniai įgūdžiai. Naujų įgūdžių rinkoje nustatymas dažniausiai atliekamas naudojant įprastas darbuotojų ir įmonių apklausas. Dėl tokio duomenų surinkimo būdo susiduriama su vėluojančių duomenų problema, įgūdžiai ir jų poreikis įvertinami tik tam tikru laiko momentu, o ne dinamiškai. Vienas iš galimų sprendimų, norint geriau suprasti reikiamus įgūdžius, yra dirbtinio intelekto metodų taikymas analizuojant šiuos įgūdžius. Dirbtinio intelekto metodų kūrimas leidžia išgauti, apdoroti ir interpretuoti dinamiškus įgūdžių poreikius, leidžiant itin greitai ir automatizuotai atlikti šiuos procesus.

Šio tyrimo tikslas – naudojant natūralios kalbos apdorojimo technologijas ir duomenų klasterizavimą, analizuojant Lietuvos darbuotojų įgūdžių poreikius, atlikti šių įgūdžių klasterinę analizę, kurti naujus darbuotojų profilius. Pagrindinės šio

mokslinio darbo užduotys – ištirti specifinių įgūdžių dinamiką, palyginti skirtingus klasterizavimo metodus, nustatyti pagrindinius darbuotojų profilius Lietuvos darbo rinkoje. Šio tyrimo socialinis ir mokslinis indėlis apima vertingų įžvalgų apie kintančius Lietuvos darbo rinkos poreikius ir pasiūlymas duomenimis pagrįstą metodą, leidžiantį įvertinti darbo įgūdžių paklausos pokyčius realiuoju laiku.

Šiam tyrimui reikalingi duomenys buvo renkami iš laisvai prieinamų lietuviškų darbo skelbimų didžiausiuose Lietuvos portaluose. Duomenims rinkti buvo naudojamos įvairios „Python“ bibliotekos, įskaitant „Playwright“, „BeautifulSoup“ ir „Selenium“. Svarbu suprasti, kad surinkti duomenys buvo nestruktūrizuoti, todėl juos reikia apdoroti daugiau nei struktūrizuotus duomenis. Be to, dėl itin nestruktūrizuotų duomenų buvo galimos ir įvairios duomenų / reklamos struktūros, o tai tik dar labiau apsunkina duomenų apdorojimo darbą. Surinkus visus duomenis didelę problemą kelia šių duomenų apdorojimas, nes jie yra labai nestruktūrizuoti. Dėl šios priežasties buvo išbandyta daug skirtingų būdų duomenims paruošti: rinktiniais žodžiais pagrįstas teksto atskyrimas, generatyviniu dirbtiniu intelektu pagrįstas išskyrimas, NER ir kt. Po šio žingsnio buvo atliekamas duomenų vektorizavimas, kad vėliau turėtume galimybę atlikti klasterinę analizę su šiais duomenimis. Duomenims vektorizuoti buvo taikomi tokie metodai: TF-IDF (angl. *Term Frequency - Inverse Document Frequency*), sakinių transformatorių (angl. *Sentence transformers*) modeliai. Atsižvelgiant į ypač dideles dimensijas tolesniame darbo etape buvo mažinamos dimensijos, tuo siekiama nustatyti, kuris duomenų dimensijų mažinimo metodas yra tinkamiausias būtent šiems duomenims.

9 lentelė. Straipsnyje analizuotų duomenų dimensijų mažinimo metodų „Trustworthiness“ metrikos reikšmės skirtingiems dimensijų dydžiams ir metodams

Dimensijų skaičius	Duomenų dimensijų mažinimo metodas				
	PCA	**UMAP**	**Trimap**	**t-SNE**	**ISOMAP**
2	0,756	0,933	0,831	0,871	0,748
3	0,805	0,950	0,882	0,881	0,821
4	0,844	0,955	0,908	0,883	0,859
5	0,876	0,961	0,931	0,886	0,896
6	0,898	0,964	0,942	0,902	0,908
7	0,915	0,966	0,950	0,912	0,920
8	0,928	0,969	0,955	0,928	0,938
9	0,940	0,971	0,961	0,937	0,949
10	0,948	0,973	0,964	0,954	0,965
15	0,968	0,974	0,971	0,966	0,977
20	0,980	0,975	0,974	0,975	0,989
25	0,986	0,978	0,976	0,978	0,989
30	0,990	0,980	0,977	0,979	0,990
35	0,992	0,983	0,978	0,981	0,991
40	0,993	0,984	0,978	0,982	0,991
50	0,996	**0,986**	0,978	0,984	0,992

Remiantis sumažintų dimensijų duomenimis, toliau duomenys buvo klasterizuojami, taip siekiant išskirti pagrindinius darbo profilius Lietuvos darbo rinkoje. Atliekant tyrimą buvo taikomi skirtingi duomenų klasterizavimo metodai – sukurti robastiniai triukšmui atsparūs metodai ir kiti šiuo metu praktikoje paplitę duomenų klasterizavimo metodai. Robastiniai duomenų klasterizavimo metodai leidžia sudaryti skirtingus darbo profilius ir kartu neprisitaikyti prie pavienių darbo skelbimų. Ieškant optimalaus duomenų klasterizavimo metodo, buvo keičiami šių metodų hiperparametrai:

- taikant *k* vidurkių duomenų klasterizavimo metodą buvo keičiamas parametras *k* – klasterių skaičius {2,3,4,5,10,20,30};
- taikant DBSCAN metode keičiami parametrai: *eps* {0.1, 0.2, 0.3, 0.4, 0.5, 0.75} ir minimalus stebinių skaičius klasteryje (angl. *min samples*) {5, 10, 20, 30, 50, 100};
- HDBSCAN keičiami parametrai: minimalaus klasterio dydis (mkl) {10, 20, 30, 40, 50, 100}, minimalus stebinių skaičius klasteryje (angl. *min samples*) {5, 10, 20, 50} ir *eps* {0.1,0.2,0.3,0.4,0.5, 1};
- BIRCH keičiami parametrai: *k* – klasterių skaičius {2, 3, 4, 5, 10, 20, 50} ir išsišakojimo koeficientas {10, 20, 50, 100};
- afiniteto sklidimo atveju keičiamas parametras – slopinimo koeficientas (angl. *damping*) {0,5, 0,6, 0,7, 0,8, 0,9};
- CBMIDE ir RCBMIDE parametrai: *k* – klasterio skaičius kaip ir ankstesniuose metoduose, projekcijų krypčių skaičius *T* {5, 10, 20, 50, 100}, glodumo parametras *h* {0,05, 0,1, 0,2, 0,3, 0,4, 0,5} ir triukšmo klasterių rinkinio svoris {0, 0,05, 0,1, 0,2, 0,3, 0,4, 0,5}.

10 lentelėje pateikiami geriausi modelių rezultatai (žr. 9 lentelę). Buvo sukurta daugiau nei 3000 modelių, siekiant įvertinti geriausią turimą modelį šiai užduočiai atlikti.

10 lentelė. Straipsnyje nagrinėtų geriausių duomenų klasterizavimo metodų rezultatai

Metodas	Parametrai	Davies–Bouldin	Calinski–Harabasz
K-means	{k: 5}	0,9143	**3386**
	{k: 5}	1,0041	2995
	{k: 5}	1,0487	2551
DBSCAN	{eps: 0.2, min_samples: 30}	1,1245	1352
	{eps: 0.3, min_samples: 20}	1,1568	1458
	{eps: 0.3, min_samples: 50}	1,2658	1589
HDBSCAN	{eps: 0.3, mkl: 50,	**0,4475**	2698

	min_samples: 20}		
	{eps: 0.2, mkl: 20, min_samples: 5}	0,7968	1398
	{eps: 0.3, mkl: 30, min_samples: 5}	0,9033	1548
BIRCH	{bf: 100, n_clusters: 5, threshold: 0.4}	1,1823	3216
	{bf: 10, n_clusters: 4, threshold: 0.3}	1,2641	2515
	{bf: 20, n_clusters: 30, threshold: 0.4}	1,2951	1927
Affinity propagation	{damping: 0.5}	1,1374	1011
	{damping: 0.8}	1,2493	1265
	{damping: 0.7}	1,2623	1255
CBMIDE	{k: 2}	1,1731	1689
	{k: 30}	1,1875	1456
	{k: 20}	1,2041	1265
RCBMIDE	{k: 4}	1,0931	2035
	{k: 20}	1,1175	1689
	{k: 10}	1,1540	1356

Sumažintas duomenų rinkinys leido greičiau atlikti skaičiavimus. Ankstesnėje lentelėje pateikti rezultatai įrodo, kad tai galima padaryti beveik neprarandant informacijos, todėl tai patvirtina, kad šiuos duomenis galima naudoti klasterizavimui. Duomenų klasterizavimo vertinimas atliktas remiantis Davies-Bouldin ir Calinski-Harabasz metrikomis. Šiuo atveju pagrindinis rodiklis buvo Davies-Bouldin. Esant labai panašioms reikšmėms, buvo atsižvelgta ir į Calinski–Harabasz metriką. Remiantis gautais rezultatais, buvo nustatyta, kad HDBSCAN metodas parodė geriausius rezultatus, kai Davies-Bouldin vertė yra 0,4475. HDBSCAN pasirodė kaip geriausiai veikiantis metodas pagal šį scenarijų dėl savo unikalaus gebėjimo valdyti įvairaus tankio ir formų grupes bei robastiškumo nustatant triukšmo taškus. Skirtingai nuo kitų klasterizavimo algoritmų, kuriems reikalingas iš anksto nustatytas grupių skaičius, HDBSCAN automatiškai nustato optimalią klasterio struktūrą pagal duomenų tankio pasiskirstymą. Dėl šio pritaikomumo jis ypač tinka sudėtingiems duomenų rinkiniams su nevienodu tankiu ir nesferinėmis grupėmis.

Tačiau nors geriausi rezultatai buvo gauti taikant HDBSCAN metodu, tačiau teigiamais rezultatais pasižymėjo ir ankstesniuose straipsniuose sukurtas RCBMIDE metodas. Atlikus detalesnį palyginimą buvo matoma, kad išskirti / sudaryti profiliai,

remiantis RCBMIDE robastiniu klasterizavimu, pasižymi tiksliais rezultatais. Tačiau nors tokie rezultatai būtų ir tenkinami, svarbu paminėti ir pastebėtus metodo trūkumus, šiuo atveju vienas iš trūkumų yra išankstinis klasterių skaičiaus nustatymas. Pritaikydami šio tyrimo rezultatus praktiškai susiduriame su situacija, kai klasterių skaičius nėra žinomas ir nustatyti tam tikrą fiksuotą klasterių skaičių tampa sudėtinga. Dėl to tampa sudėtingiau taikyti šį metodą.

Išvados. Šiame moksliniame tyrime buvo iškeltas tikslas pritaikyti natūralios kalbos apdorojimo, duomenų klasterizavimo ir kitus mašininio mokymosi metodus darbuotojų įgūdžių poreikio analizėje. Šio tyrimo darbuotojų įgūdžių vertinimo erdvė buvo Lietuvos darbo skelbimai. Pradiniame tyrimo etape duomenys buvo automatizuotai paimami, apdoroti ir paruošti tolesniems darbams. Duomenys buvo išgaunami nuolatos, taip įvertinant įgūdžių poreikio pokyčius dinamiškai, o ne tik tam tikru laiko momentu. Moksliniame darbe apžvelgiami įvairūs duomenų paruošimo ir apdorojimo metodai, pritaikant įvairius duomenų vektorizavimo metodus, nagrinėjami šių metodų pranašumai ir trūkumai.

Pradiniu duomenų analizės etapu, surinkus duomenis, buvo siekiama iš darbo skelbimų išskirti darbuotojų įgūdžius. Šis procesas yra sudėtingas, nes darbo skelbimai neturi bendro standarto ir įgūdžiai gali būti pateikiami įvairiose darbo skelbimo vietose. Šiame tyrime geriausi rezultatai buvo gauti darbuotojų įgūdžius iš pradinio teksto išskiriant „regex“ metodu. Šiuo metu iš pradinio surinkto teksto buvo išrinkta tik reikalinga informacija, kuri naudojama tolesniame tyrime. Duomenims vektorizuoti buvo taikomi du pagrindiniai metodai, davę geriausius rezultatus – TF-IDF ir sakinių transformeriai (BERT modelis). Dėl surinkto didelio darbo skelbimų skaičiaus sudaryti vektoriai tapo didelės apimties ir apsunkino skaičiavimą bei skirtingų modelių palyginimą. Dėl sakinių transformatoriaus pateikiamų vektorių matmenų (384) tyrimo metu buvo taikomi ir skirtingi duomenų dimensijų mažinimo metodai. Geriausi rezultatai buvo pastebėti taikant ISOMAP duomenų dimensijų mažinimo metodą. Iš viso buvo sukurta 11 000 matmenų mažinimo modelių, skirtų skirtingiems matmenų mažinimo būdams ir jų parametrams įvertinti bei kiekvieno modelio optimaliems parametrams nustatyti. Pažymėtina, kad gerais rezultatais pasižymėjo ir kiti duomenų klasterizavimo metodai, pvz., UMAP. Skirtumas tarp šių metodų rezultatų buvo palyginti nedidelis, todėl pagrindinis veiksnys, lemiantis automatizuotą darbo skelbimų analizę, tapo apdorojimo greitis. Palyginus skirtingų duomenų dimensijų mažinimo metodų greitį, pastebėta, kad greičiausiai veikė šiuo specifiniu atveju UMAP metodas. Šis metodas buvo greitesnis už kitus tyrimo metu taikytus metodus – t-SNE, ISOMAP. Norint pasiekti geriausią greitį, galėjo būti taikomas PCA metodas, tačiau jo rezultatai, palyginti su UMAP, būtų buvę mažesnio tikslumo. Pažymėtina, kad padidinus latentinių dimensijų skaičių, duomenų matmenų mažinimo kontekste nebuvo geresnių rezultatų, o tai reiškia, kad nėra prasmės naudoti daugiau resursų skaičiavimams gaunant tuos pačius rezultatus. 20 latentinių dimensijų puikiai atspindėjo visą reikšmingą informaciją, o didesnis latentinių dimensijų skaičius nesuteikė papildomos informacijos duomenų klasterizavimo metodams. Šiame tyrime matmenims mažinti buvo naudojama 50 latentinių matmenų, gautų taikant anksčiau minėtą UMAP metodą. Tyrimo metu buvo pastebėta tai, kad mažinant šių duomenų dimensijas, latentinių dimensijų skaičiaus sumažinimas iki 2

ar 5 pasižymi aukšta patikimumo metrika, patikimumo metrikos vertė buvo didesnė nei 0,9.

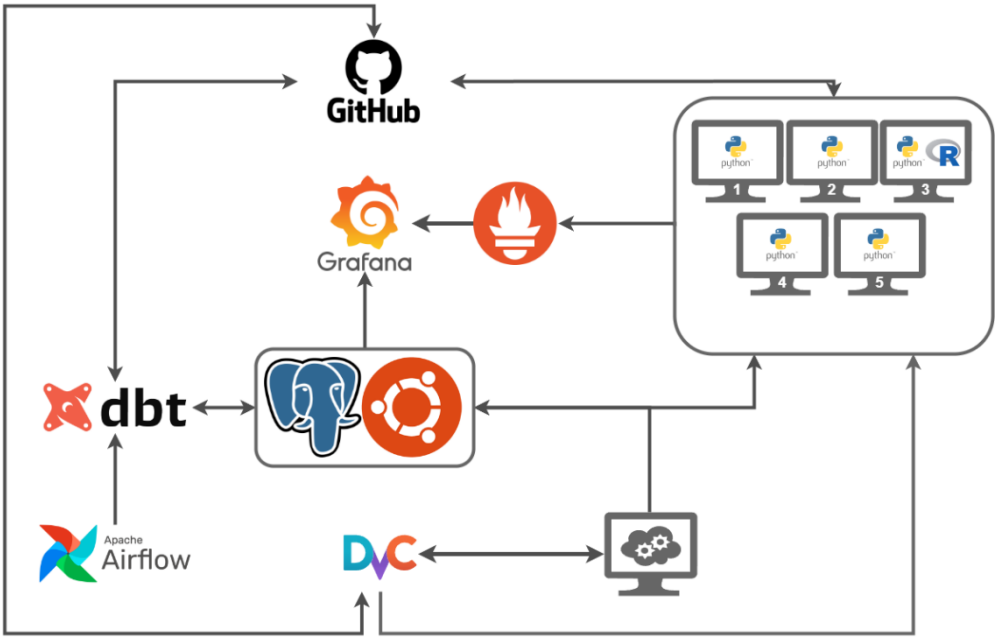
Tyrimo metu buvo taikomi skirtingo veikimo duomenų klasterizavimo metodai, kurie buvo aptariami šiame straipsnyje. Klasterizuojant duomenis sukurta daugiau nei trys tūkstančiai skirtingų duomenų klasterizavimo modelių, siekiant rasti tinkamiausią duomenų klasterizavimą šiam praktiniam taikymui ir skirtingų metodų optimalus parametrų rinkinį. HDBSCAN metodas pasirodė efektyviausias klasterizuojant duomenis daugiausia dėl jo hierarchinės struktūros. Be to, RCBMIDE metodas taip pat parodė gana tikslius / sėkmingus rezultatus, leido pašalinti triukšmą ir įtraukti duomenis, kurie geriausiai atspindi apibendrintą informaciją. Remiantis abiem metodais gauti darbo profiliai buvo gana panašūs. Nors sudaryti darbo profiliai ir buvo panašūs ir galėjo būti interpretuojami abiejų duomenų klasterizavimo metodų, remiantis vertintomis metrikomis HDBSCAN metodas pasižymėjo geresnėmis tikslumo metrikomis. Remiantis pasirinktomis metrikomis HDBSCAN metodas buvo taikomas kaip pagrindinis tyrimo duomenų klasterizavimo metodas. Įdomu tai, kad šiuo metu dažniausiai taikomas duomenų klasterizavimo *k* vidurkių metodas parodė, kad geriausi rezultatai gauti naudojant $k = 5$. Galima teigti, kad toks darbo profilių skaičius yra iš tiesų per mažas atsižvelgiant į didelį darbo skelbimų skaičių ir skirtingų specialybių / profesijų skaičių. Remiantis tokiais rezultatais galima daryti išvadą, kad nors *k* vidurkių metodas ir yra dažniausiai taikomas, tačiau šiuo specifiniu atveju nedavė gerų rezultatų. Taip pat tai parodo, kad naujai sukurtas duomenų klasterizavimo metodas pasižymėjo kur kas geresniais rezultatais ir buvo konkurencingas HDBSCAN metodui pagal savo veikimą ir interpretuojamumą.

Atliekant sudarytų klasterių analizę pastebima tai, kad automatizuotai sudaryti darbo profiliai pasižymi aukštos kokybės apibendrinimu, gerais rezultatais. Tačiau remiantis sudarytais profiliais pastebėta, kad kai kuriuose profiliuose buvo sunkiau suprasti įgūdžius, pavyzdžiui, vokiečių kalbos („germany“) ir kt. Tokie rezultatai rodo, kad šis automatizuotas darbo profilių sudarymas turi apribojimų. Vienas iš apribojimų – didelė profilių tikslumo priklausomybė nuo pradinių duomenų paruošimo, nes neteisingai paruošti duomenys, kuriuose yra likę daug triukšmo (nereikalingos informacijos), gali iškraipyti sudarytus profilius. Aptariant pastebėtas situacijas galima teigti, kad tokie išryškėję įgūdžiai, kaip vokiečių kalba, apibendrinta žodžiu „Vokietija“, nors ir yra sunkiau suprantami, bet gali būti interpretuojami. Siekiant, kad darbo profilių sudarymas būtų automatizuotas maksimaliai, šie darbo profiliai buvo apibendrinti naudojant generatyvinį dirbtinį intelektą. Paskutiniu darbo profilių automatizavimo žingsniu klasteriams / grupėms apibendrinti / apibūdinti buvo naudojami OpenAI dideli kalbos modeliai (angl. *large language models*). Pastebima, kad geriausi rezultatai buvo gauti naudojant naujausią LLM modelį – GPT-4, tačiau didesniu greičiu ir mažesne kaina pasižymintis GPT-3.5 modelis davė taip pat aukštos kokybės rezultatus. Tyrimo metu pritaikius kitus senesnius kalbos modelius buvo pastebėta, kad šie modeliai nepasiekia norimo darbo profilių apibendrinimo rezultato, t. y. profiliai yra neinformatyvūs, apibūdinimai nėra tikslūs, ne visada naudojama visa klasterio informacija.

3.6. Straipsniuose nepublikuotų tyrimų rezultatų aptarimas

Šiame poskyryje pateikiama informacija apie tyrimo vykdymą ir jo metu gautus rezultatus, kurie nebuvo publikuoti disertacijoje pristatytuose straipsniuose. Tyrimo metu buvo įtraukti šiuo metu daugiausia taikomi duomenų klasterizavimo metodai. Jie buvo pasirinkti remiantis kitų mokslinių darbų pavyzdžiu bei tuo, kokie metodai taikomi moksliniuose darbuose [163, 164]. Taip pat tyrime buvo naudoti skirtingi duomenų rinkiniai, leidžiantys įvertinti duomenų klasterizavimo tikslumą. Šie rinkiniai taip pat buvo pasirinkti remiantis kitų mokslinių tyrimų pavyzdžiais ir dažniausiai naudojamais duomenų rinkiniais [163, 165, 166].

Tyrimo metu buvo atliekami didelės apimties skaičiavimai, todėl buvo naudoti keli serveriai. Rezultatams skaičiuoti buvo panaudoti Kauno technologijos universiteto skaičiavimo centro virtualūs serveriai, asmeniniai kompiuteriai (i7-12700K, RTX 3080Ti, 128 GB RAM) ir „Google Cloud“ virtualūs serveriai. Toliau pateikiama techninė naudotų įrankių schema ir šių įrankių paaiškinimas (2 pav.).



2 pav. Techninė tyrimo schema

Trumpas naudojamų įrankių aprašymas:

- „Github“ – kodo versijų valdymo sistema, naudojama viso tyrimo metu duomenims bei programiniam kodui saugoti ir atnaujinti. Ji leido užtikrinti, kad visi serveriai turėtų priėjimą prie naujausių kodo versijų.
- Serveriai – didelio našumo skaičiavimo vienetai, skirti duomenims apdoroti ir analizuoti. Jais naudojantis buvo įmanoma atlikti reikalingus skaičiavimus

greitai ir efektyviai, nepriklausomai nuo duomenų kiekių. Šiuose serveriuose buvo naudojamos dvi pagrindinės programavimo kalbos: „Python“ ir R.

- „Grafana“ – atvirojo kodo platforma, skirta vizualizuoti „Prometheus“ ir kitų sistemų renkamiems duomenims. Ja naudojantis buvo galima lengvai stebėti ir analizuoti skaičiavimo proceso efektyvumą ir kitus svarbius parametrus.
- „Prometheus“ – atvirojo kodo sistema, naudojama serverių ir aplikacijų veikimo rodikliams realiuoju laiku stebėti ir analizuoti. Ji leido greitai reaguoti į bet kokias sistemos veikimo problemas ir užtikrinti sklandų tyrimo procesą.
- „Ubuntu“ – populiariausia „Linux“ sistemos versija, pasirinkta dėl jos stabilumo, saugumo ir plačiai prieinamų dokumentų.
- DVC (angl. *Data Version Control*) – įrankis duomenų versijoms valdyti, leidžiantis sekti ir kontroliuoti eksperimentų duomenų pakeitimus. DVC buvo naudingas organizuojant duomenis, ypač atliekant daug eksperimentų ir analizuojant skirtingus duomenų rinkinius.
- dbt (angl. *data build tool*) – įrankis duomenims duomenų bazėse transformuoti. dbt leidžia rašyti transformacijas, kurios yra lengvai perduodamos ir testuojamos, užtikrinant aukštą duomenų kokybę ir patikimumą.
- „Airflow“ – platforma, skirta duomenų apdorojimo užduotims planuoti, koordinuoti ir vykdyti. Ją naudojant buvo galima automatizuoti ir optimizuoti duomenų apdorojimo darbo eigą, taip užtikrinant efektyvesnį ir sklandesnį duomenų analizės procesą.
- „PostgreSQL“ – viena labiausiai vertinamų duomenų bazių valdymo sistemų mokslinių tyrimų srityje, nes pasižymi dideliu stabilumu, saugumu ir efektyvumu dirbant su dideliais duomenų kiekiais. Mūsų tyrimo metu „PostgreSQL“ buvo pagrindinė platforma, kurioje buvo saugomi visi gauti tyrimo rezultatai. Naudojant šią sistemą, buvo užtikrintas duomenų vientisumas ir prieinamumas analizės procese. Tyrimo metu „PostgreSQL“ darbuotis buvo glaudžiai susijusi su dbt ir „Airflow“. dbt buvo naudojamas duomenims transformuoti ir paruošti analizei tiesiai duomenų bazėje. Tai leido efektyviai valdyti duomenų modelius, automatizuoti jų atnaujinimą ir užtikrinti, kad analizė vyktų naudojant tikslius ir patikrintus duomenis. dbt taip pat suteikė galimybę vykdyti duomenų testavimą, kuris yra gyvybiškai svarbus užtikrinant duomenų kokybę mokslinių tyrimų kontekste.

Visi šie įrankiai sudarė tvirtą technologinę tyrimo bazę, leidžiančią efektyviai tvarkyti, analizuoti ir pateikti duomenis, taip užtikrinant tyrimo rezultatų patikimumą ir mokslinį naudingumą. Norint įvertinti klasterizavimo rezultatus, būtina pasirinkti tinkamus vertinimo rodiklius, nes jie gali nulemti klasterizavimo vertinimą. Šiame tyrime klasterizavimo metodai lyginami naudojant J-Score [147], NMI [148], ARI [149], tikslumą (angl. *Accuracy*) [150] ir Fowlkes-Mallows indeksą (FMI) [151]. Šios metrikos tinka tais atvejais, kai žinomos tikros stebinių grupės. Kartu įtraukiamos ir metrikos, kurios yra tinkamos tuo atveju, jei stebinių grupės nėra žinomos: Calinski ir Harabasz rodiklis, dar žinomas kaip dispersijos santykio kriterijus [152], Davies-

Bouldin metrika [153], Silueto koeficientas [167] ir kt. Duomenų bazėje visos šios metrikos saugomos kiekvienam eksperimentui, taip pat išsaugant ir daugiau skirtingų metrikų kaip informaciją apie kiekvieną stebinį – kokiai grupei jis buvo priskirtas eksperimento metu. Toliau pateikiamas trijų pagrindinių vertinimo metrikų, pateiktų šiame darbe, apibrėžimai ir apskaičiavimo logika. Ahmadinejad and Liu [147] pasiūlė naują klasterizavimo vertinimo metriką – „JScore“. „Jscore“ yra paprastas ir išskirtims atsparus klasterizavimo tikslumo matas. Ši metrika leidžia išspręsti keletą problemų, su kuriomis susiduria kitos vertinimo metrikos: atitikmenų ir persimokymo problemas [147]. Dviejų krypčių rinkinio atitiktis: tarkime, kad duomenų rinkinyje yra N duomenų taškai, priklausantys T tikrosioms klasėms, o klasterių analizė sukuria K grupes. Norėdami nustatyti atitiktį tarp T ir K, pirmiausia nustatome geriausiai atitinkančią grupę ($T \rightarrow K$). Tiksliau, klasei $t \in T$ ieškome klasterio $k \in K$, turinčio didžiausią „Jaccard“ indeksą,

$$I_t = \max_{k \in K} \frac{|V_t \cap V_k|}{|V_t \cup V_k|}, \tag{3.27}$$

čia V_t ir V_k – stebinių rinkiniai, kurie atitinkamai priklauso klasei / grupei t ir klasteriui k ir $|\cdot|$ žymi rinkinių dydį. Tuomet kiekvienam klasteriui ieškome geriausiai atitinkančias klases ($K \rightarrow T$), naudodami panašumą. Kiekvienam klasteriui $k \in K$ ieškome klasės $t \in T$ su didžiausiu „Jaccard“ indeksu.

$$I_k = \max_{t \in T} \frac{|V_t \cap V_k|}{|V_t \cup V_k|}. \tag{3.28}$$

Tuomet siekiant apskaičiuoti bendrą tikslumą visi apskaičiuoti kiekvieno klasterio ir kiekvienos klasės „Jaccard“ indeksai agreguojami atsižvelgiant į jų dydžius (t. y. stebinių skaičių). Pirmiausia apskaičiuojama svertinė visų klasių I_t suma $R = \sum_{t \in T} \left(\frac{V_t}{N} I_t \right)$ ir svertinė visų klasterių I_k suma $P = \sum_{k \in K} \left(\frac{V_k}{N} I_k \right)$. „Jscore“ metrika apskaičiuojama kaip harmoninis šių reikšmių vidurkis:

$$J = \frac{2 \times R \times P}{R + P}. \tag{3.29}$$

Ši metrika buvo realizuota „Python“ programavimo kalba (žr. 6 priedą). Tikslumas (angl. *Accuracy*) dažnai naudojamas klasifikavimo kokybei matuoti. Jis taip pat naudojamas grupuoti, tačiau svarbu pabrėžti tai, kad klasterizavimo klasių numeriai gali skirtis nuo tikrųjų. Dėl šios priežasties svarbu naudoti sumaišymo matricą ir jos įstrižainę, ją išrikiavus. Tikslumo metrika apskaičiuojama taip:

$$ACC = \frac{1}{N} \sum_{i=1}^{k} n_i, \tag{3.30}$$

čia N – bendras stebinių skaičius, esantis duomenų rinkinyje; n_i – teisingai klasteriui I priskirtų stebinių skaičius; k – klasterių skaičius.

Silueto koeficientas yra plačiai naudojama metrika, skirta klasterizavimo rezultatų kokybei įvertinti. Jis matuoja, kiek objektas yra panašus į savo klasterį, palyginti su kitais klasteriais. Ši metrika suteikia informaciją apie suformuotų grupių kompaktiškumą ir atskyrimą. Koeficientas gali turėti reikšmes nuo –1 iki 1, kur didelė reikšmė rodo, kad objektas yra gerai suderintas su savo klasteriu ir prastai suderintas su kaimyniniais klasteriais. Silueto koeficientas apskaičiuojamas, kaip pateikta toliau:

$$s_i = \frac{b_i - a_i}{max(b_i, a_i)}, \tag{3.31}$$

čia b_i – atstumas, apibūdinamas kaip vidutinis atstumas iki artimiausio klasterio stebiniui i, kuriam šis stebinys nepriklauso, apskaičiuojamas taip:

$$b_i = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i.j). \tag{3.32}$$

a_i apibūdinamas kaip atstumas iki kitų stebinių, esančių tame pačiame klasteryje, kuris apskaičiuojamas taip:

$$a_i = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i.j). \tag{3.33}$$

Tuomet bendras Silueto koeficientas apskaičiuojamas kaip vidutinė kiekvieno stebinio Silueto koeficiento reikšmė.

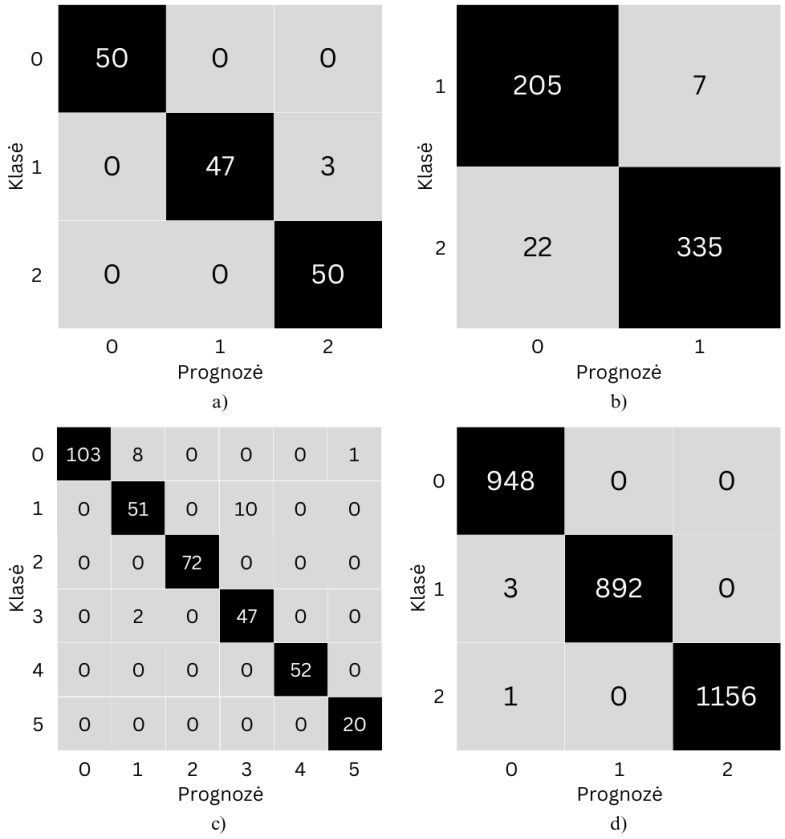
Kaip minėta, šiame skyriuje pateikiama visų tyrimų rezultatų informacija. Šie rezultatai yra išplėstiniai, nes moksliniai tyrimai buvo vykdomi ir toliau, paskelbus mokslinę publikacijos. Bendras eksperimentų skaičius viršija 70 milijonų. Duomenų bazėje saugomos anksčiau aprašytos metrikos ir kita tyrimui vertinga informacija. 2, 3 ir 4 prieduose pateikiami geriausi rezultatai visiems tyrime naudotiems duomenų rinkiniams. Šiuose prieduose pateikiamos trys pagrindinės tyrime naudotos metrikos. Galima pastebėti, kad šie rezultatai rodo daug didesnę duomenų rinkinių imtį, lyginant su straipsniuose pateiktais duomenų rinkiniais. Visi duomenų rinkiniai aprašyti 1 priede. 11 lentelėje pateikiamos vidutinės tikslumo, „JScore“ ir Silueto koeficiento reikšmės, remiantis visais duomenų rinkiniais.

11 lentelė. Vidutinės tikslumo (angl. *Accuracy*), „JScore“, Silueto koeficiento reikšmės, apskaičiuotos remiantis visais duomenų rinkiniais

	Tikslumas	**Jscore**	**Silueto koeficientas**
Agg	0,882	0,824	0,405
BGM	0,809	0,729	0,437
BIRCH	0,868	0,802	0,406

CBMIDE	0,868	0,797	0,399
DBSCAN	0,881	0,833	0,406
FCM	0,673	0,599	0,44
GMM	0,797	0,720	0,461
HDBSCAN	0,847	0,791	0,342
K-vidurkių	0,795	0,715	0,459
MIDE	0,606	0,536	0,28
OPTICS	0,822	0,762	0,411
ST-DBSCAN	0,741	0,664	0,369

Remiantis gautais rezultatais galima pastebėti, kad kai kurių tyrimo rezultatai (žr. 2, 3, 4 priedus) yra geresni, lyginant su straipsniuose publikuotais rezultatais. Šie geresni rezultatai buvo gauti, nes tyrimas buvo tęsiamas ieškant geriausių parametrų kiekvienam duomenų klasterizavimo metodui ir siekiant optimizuoti rezultatus. Pastebima, kad ne tik naujai sukurtas metodas, bet ir kiti metodai, tokie kaip *k* vidurkių ar DBSCAN, pasižymi geresnėmis metrikų reikšmėmis, lyginant su straipsniuose pateiktais rezultatais. Prieduose taip pat pateiktas pavyzdinis kodas, kuris gali būti naudojamas šiems rezultatams atkurti. Remiantis gautais rezultatais, pastebima, kad naujai sukurtas metodas veikia konkurencingai, lyginant su pripažintais praktikoje duomenų klasterizavimo metodais, tokiais kaip DBSCAN ar *k* vidurkių, ir pasižymi vienomis iš geriausių tikslumo bei „J-Score“ metrikų reikšmėmis. Tyrimų metu pastebėta, kad naujai sukurtas metodas pasižymi prastais rezultatais su tokiais duomenų rinkiniais kaip „Spiral“. Naujai sukurtam metodui tampa sudėtinga atskirti sunkiai atskiriamos struktūros duomenų taškus į klasterius. Taip pat sudaromas triukšmo klasteris naujo metodo atveju yra tolydus skirstinys. Pastebima, kad tokiems duomenų rinkiniams kaip „glass“ ar „Iris“ naujai sukurtas metodas duoda net geriausius duomenų klasterizavimo rezultatus. Toliau pateikiamos CBMIDE rezultatų sumaišymo matricos keturiems duomenų rinkiniams (žr. 3 pav.). Šiems duomenų rinkiniams CBMIDE metodas pasižymi ypač gerais klasterizavimo rezultatais.



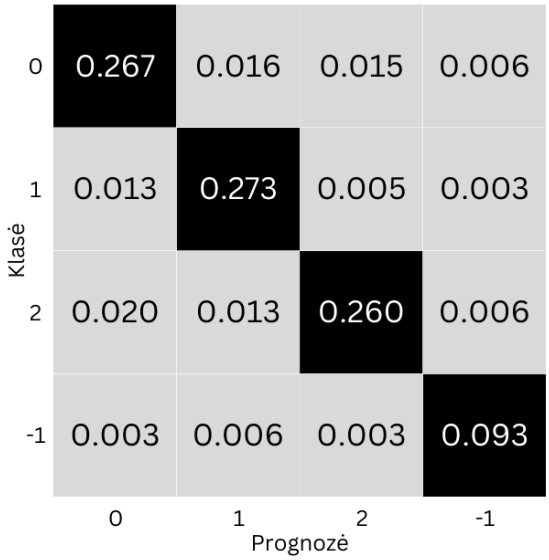
3 pav. Duomenų klasterizavimo rezultatų sumaišymo matricos taikant CBMIDE metodą: a) „Iris“ duomenų rinkiniui; b) „Breast“ duomenų rinkiniui; c) „Dermatology“ duomenų rinkiniui; d) „Xclara“ duomenų rinkiniui

Siekiant nuodugniau palyginti duomenų klasterizavimą buvo panaudoti 300 000 tūkstančių stebinių duomenų rinkiniai. Jie buvo sudaryti taip pat kaip 2 straipsnyje generuoti duomenų rinkiniai. Metodai buvo geriau palyginti naudojant šiuos duomenų generavimo parametrus:

- triukšmo lygis – procentinė išraiška, kiek duomenų taškų duomenų rinkinyje vertinami kaip išskirtys;
- klasterių dalinis sutapimas – generuotų klasterių sutapimas buvo vertinamas taip: nesutampantys, mažai sutampantys, stipriai sutampantys.

12–14 lentelėse pateikiami pagrindiniai tyrimo rezultatai: klasterizavimo tikslumas (angl. *Accuracy*) ir Silueto koeficientas. Taip pat prieduose papildomai pateiktos ir „Jscore“ metrikos reikšmės visiems metodams ir duomenų rinkiniams (žr. 7–9 priedus). Vienas iš svarbių šio tyrimo rezultatų yra tas, kad tokie metodai kaip DBSCAN ar HDBSCAN, paremti stebinių porų atstumo skaičiavimu, pasižymi dideliu atminties poreikiu. Štai klaidos pavyzdys: „*numpy. core. _exceptions. _ArrayMemoryError: Unable to allocate 671 GiB for an array with shape (300000,*

300000) and data type float64“. Dėl šios priežasties tokius metodus sunku pritaikyti dideliems duomenų rinkiniams. 4 pav. pateikiama CBMIDE metodo sumaišymo matrica, kai naudojamas 300 000 stebinių rinkinys su 10 proc. triukšmo.



4 pav. CBMIDE metodo sumaišymo matrica naudojant 300 000 stebinių duomenų rinkinį su 10 proc. triukšmo

12 lentelė. Duomenų klasterizavimo tikslumas (angl. *Accuracy*) esant nesutampantiems klasteriams, kai yra skirtingi triukšmo lygiai

	Triukšmo lygis (%)					
Metodas	1 %	2 %	5 %	10 %	20 %	50 %
Affinity	0,581	0,534	0,605	0,500	0,512	0,526
Agglomerative	0,994	0,987	0,964	0,919	0,877	0,658
BGMM	1,000	0,999	0,994	0,977	0,897	0,744
Birch	0,990	0,980	0,950	0,900	0,800	0,615
CBMIDE	1,000	0,998	0,982	0,975	0,965	0,913
DBSCAN	1,000	0,999	0,997	0,993	0,993	0,995
FCM	0,990	0,980	0,950	0,910	0,847	0,617
GMM	1,000	0,980	0,998	0,986	0,971	0,710
HDBSCAN	0,998	0,992	0,983	0,957	0,937	0,876
k-means	0,990	0,986	0,964	0,917	0,844	0,613
MeanShift	0,992	0,985	0,958	0,914	0,828	0,570
OPTICS	0,993	0,986	0,966	0,932	0,897	0,984
RobustLinkage	0,969	0,970	0,909	0,605	0,855	0,721
Spectral	0,993	0,985	0,950	0,900	0,800	0,581
ST-DBSCAN	0,989	0,971	0,644	0,620	0,859	0,463

13 lentelė. Duomenų klasterizavimo tikslumas *(angl. Accuracy*) esant mažai sutampantiems klasteriams, kai yra skirtingi triukšmo lygiai

	Triukšmo lygis (%)					
Metodas	1 %	2 %	5 %	10 %	20 %	50 %
Affinity	0,280	0,294	0,291	0,300	0,325	0,237
Agglomerative	0,991	0,983	0,957	0,916	0,855	0,585
BGMM	0,991	0,986	0,973	0,930	0,876	0,586
Birch	0,990	0,979	0,950	0,899	0,799	0,565
CBMIDE	0,992	0,986	0,974	0,935	0,888	0,754
DBSCAN	0,992	0,989	0,973	0,963	0,930	0,888
FCM	0,989	0,979	0,949	0,899	0,848	0,601
GMM	0,989	0,979	0,963	0,919	0,849	0,580
HDBSCAN	0,991	0,989	0,971	0,962	0,930	0,871
k-means	0,989	0,979	0,949	0,919	0,846	0,597
MeanShift	0,991	0,975	0,949	0,911	0,807	0,534
OPTICS	0,991	0,988	0,969	0,961	0,917	0,631
RobustLinkage	0,977	0,948	0,904	0,827	0,847	0,841
Spectral	0,989	0,979	0,949	0,911	0,818	0,549
ST-DBSCAN	0,648	0,636	0,621	0,591	0,523	0,459

14 lentelė. Duomenų klasterizavimo tikslumas (angl. *Accuracy*) stipriai sutampantiems klasteriams, kai yra skirtingi triukšmo lygiai

	Triukšmo lygis (%)					
Metodas	1 %	2 %	5 %	10 %	20 %	50 %
Affinity	0,174	0,182	0,179	0,187	0,186	0,171
Agglomerative	0,896	0,880	0,856	0,805	0,727	0,520
BGMM	0,907	0,912	0,843	0,836	0,738	0,526
Birch	0,906	0,891	0,863	0,824	0,733	0,523
CBMIDE	0,916	0,914	0,866	0,842	0,749	0,559
DBSCAN	0,597	0,588	0,556	0,581	0,522	0,694
FCM	0,911	0,901	0,876	0,837	0,740	0,534
GMM	0,912	0,899	0,879	0,827	0,769	0,527
HDBSCAN	0,594	0,587	0,559	0,538	0,523	0,694
k-means	0,910	0,904	0,879	0,834	0,760	0,531
MeanShift	0,875	0,758	0,722	0,704	0,506	0,247
OPTICS	0,373	0,368	0,340	0,338	0,317	0,521
RobustLinkage	0,614	0,635	0,521	0,561	0,660	0,664
Spectral	0,911	0,902	0,616	0,604	0,768	0,506
ST-DBSCAN	0,561	0,550	0,533	0,382	0,443	0,464

Toliau pateikiami rezultatai vertinant duomenų klasterizavimo metodus remiantis Silueto koeficientu. 15 lentelėje pateiktas apskaičiuotas Silueto koeficientas naudojant tikrąsias klasterių reikšmes, t. y. kokia būtų Silueto metrikos reikšmė, jei klasteriai būtų atskirti 100 % tiksliai. Klasterių standartinis nuokrypis (SN) rodo, kaip labai klasteriai sutampa. 0,25 klasterio SN reikšmė rodo, kad klasteriai yra gerai atskirti vienas nuo kito, 0,5 – vidutiniškai, 1 – stipriai sutampa. Galima pastebėti, kad nė viename duomenų rinkinyje Silueto reikšmė nėra 1. Taip pat didėjant klasterių standartiniam nuokrypiui Silueto reikšmė mažėja, tokia pati tendencija matoma ir didėjant triukšmo santykiui.

15 lentelė. Idealus Silueto koeficientas visiems sudarytiems duomenų rinkiniams su skirtingu triukšmo lygiu.

	Triukšmo lygis (%)					
Klasterio SN	1 %	2 %	5 %	10 %	20 %	50 %
0,25	0,8347	0,8257	0,7866	0,7102	0,5837	0,1741
0,5	0,6785	0,6729	0,6483	0,5894	0,4670	0,1021
1,0	0,4174	0,4117	0,3871	0,3430	0,2518	-0,0065

16 lentelė. Duomenų klasterizavimo Silueto koeficiento reikšmės esant nesutampantiems klasteriams, kai yra skirtingi triukšmo lygiai

	Triukšmo lygis (%)					
Metodas	1 %	2 %	5 %	10 %	20 %	50 %
Affinity	0,302	0,381	0,311	0,298	0,338	0,451
Agglomerative	0,834	0,781	0,795	0,754	0,673	0,486
BGMM	0,835	0,829	0,795	0,742	0,604	0,370
Birch	0,848	0,840	0,806	0,788	0,686	0,476
CBMIDE	0,845	0,839	0,809	0,764	0,687	0,368
DBSCAN	0,835	0,829	0,785	0,717	0,588	0,179
FCM	0,848	0,840	0,817	0,781	0,704	0,168
GMM	0,835	0,840	0,790	0,730	0,580	0,319
HDBSCAN	0,841	0,836	0,805	0,767	0,653	0,322
k-means	0,848	0,832	0,802	0,764	0,688	0,550
MeanShift	0,733	0,761	0,717	0,696	0,631	0,426
OPTICS	0,843	0,839	0,812	0,780	0,679	0,169
RobustLinkage	0,769	0,775	0,556	0,104	0,454	-0,280
Spectral	0,826	0,769	0,817	0,788	0,710	0,515
ST-DBSCAN	0,837	0,792	0,595	0,590	0,597	0,374

17 lentelė. Duomenų klasterizavimo Silueto koeficiento reikšmės esant mažai sutampantiems klasteriams, kai yra skirtingi triukšmo lygiai

	Triukšmo lygis (%)					
Metodas	1 %	2 %	5 %	10 %	20 %	50 %
Affinity	0,306	0,310	0,316	0,307	0,322	0,333

Agglomerative	0,702	0,597	0,641	0,650	0,589	0,393
BGMM	0,702	0,693	0,680	0,605	0,549	0,464
Birch	0,708	0,702	0,686	0,654	0,595	0,395
CBMIDE	0,708	0,702	0,686	0,651	0,589	0,432
DBSCAN	0,699	0,688	0,675	0,619	0,508	0,081
FCM	0,708	0,702	0,688	0,656	0,611	0,476
GMM	0,708	0,702	0,674	0,609	0,609	0,422
HDBSCAN	0,695	0,693	0,681	0,622	0,515	0,197
k-means	0,708	0,702	0,688	0,656	0,613	0,478
MeanShift	0,657	0,629	0,559	0,582	0,469	0,360
OPTICS	0,693	0,692	0,681	0,629	0,554	0,121
RobustLinkage	0,468	0,461	0,573	0,422	0,376	0,162
Spectral	0,708	0,702	0,688	0,659	0,602	0,478
ST-DBSCAN	0,562	0,522	0,530	0,500	0,412	0,239

18 lentelė. Duomenų klasterizavimo Silueto koeficiento reikšmės esant stipriai sutampantiems klasteriams, kai yra skirtingi triukšmo lygiai

	Triukšmo lygis (%)					
Metodas	1 %	2 %	5 %	10 %	20 %	50 %
Affinity	0,336	0,329	0,336	0,339	0,348	0,369
Agglomerative	0,335	0,419	0,373	0,317	0,308	0,295
BGMM	0,475	0,471	0,440	0,420	0,357	0,357
Birch	0,464	0,454	0,451	0,427	0,410	0,327
CBMIDE	0,475	0,472	0,466	0,452	0,417	0,361
DBSCAN	-0,041	-0,176	0,265	-0,134	0,127	-0,102
FCM	0,477	0,475	0,469	0,443	0,419	0,370
GMM	0,475	0,472	0,467	0,401	0,412	0,361
HDBSCAN	0,333	0,335	0,312	0,033	0,131	-0,085
k-means	0,477	0,475	0,470	0,444	0,411	0,365
MeanShift	0,315	0,225	0,283	0,238	0,245	0,275
OPTICS	0,026	0,015	-0,169	-0,430	-0,389	-0,395
RobustLinkage	0,036	0,138	-0,042	0,035	0,101	0,160
Spectral	0,476	0,471	0,395	0,437	0,378	0,359
ST-DBSCAN	0,289	0,263	0,277	0,284	0,166	0,293

Remiantis tyrimo rezultatais esant dideliam duomenų kiekiui ir skirtingam triukšmo santykiui su stebinių skaičiumi pastebima, kad ne visi metodai lengvai pritaikomi dėl didelio atminties poreikio. Tokie metodai, kaip DBSCAN, HDBSCAN ir kiti, kurie skaičiuoja atstumų matricą, reikalauja ypač didelio atminties kiekio. Taip pat pastebima, kad sukurtas CBMIDE metodas pasižymi gerais rezultatais, lyginant su kitais duomenų klasterizavimo metodais, kurie šiuo metu yra populiariausi ir taikomi dažniausiai.

4. IŠVADOS

1. Mokslinio tyrimo metu buvo išsamiai ištirti populiarūs duomenų klasterizavimo metodai, jų veikimo principai, apribojimai ir taikymas tiek moksliniuose tyrimuose, tiek praktinėje veikloje, taip pat analizuotos skirtingų metodų / algoritmų robastinės modifikacijos. Atliekant mokslinės literatūros analizę nustatyta, kad duomenų klasterizavimo tyrimuose naudojama daugiau nei 100 skirtingų duomenų rinkinių, kurie skirstomi į dvi pagrindines grupes: realius duomenis ir sintetinius. Pastebėta, kad šiuose tyrimuose dažnai naudojami duomenų rinkiniai, turintys tikrąsias klasių reikšmes, nes tai padeda lengviau įvertinti klasterizavimo tikslumą. Dėl šios priežasties galima teigti, kad, kuriant naujus duomenų klasterizavimo metodus, moksliniuose tyrimuose dažnai naudojami duomenų klasifikavimo rinkiniai. Duomenų klasterizavimo metodai skirstomi į kelias pagrindines grupes: padalinimo, tankio pagrindu veikiantys metodai, tinkleliu ir modeliais paremti metodai bei hierarchiniai klasterizavimo metodai. Atliekant mokslinius tyrimus pastebėta, kad dažniausiai naudojamas duomenų klasterizavimo metodas vis dar yra *k* vidurkių metodas.

2. Remiantis tyrimų rezultatais buvo sukurtas ir pasiūlytas robastinis duomenų klasterizavimo metodas, pagrįstas modifikuotos apvertimo formulės taikymu. Tyrimai įrodė, kad apvertimo formulės tankio įvertinys veikia tiksliau, lyginant su kitais metodais, todėl, remiantis šiuo tankio įvertiniu, buvo sukurtas duomenų klasterizavimo metodas. Sukurtas duomenų klasterizavimo metodas dėl savo veikimo turėjo apribojimų esant didelėms duomenų dimensijoms, todėl, atsižvelgiant į tai, buvo sukurtos skirtingos metodo modifikacijos. Sukurtos duomenų klasterizavimo metodo modifikacijos grindžiamos duomenų dimensijų mažinimu. Duomenų klasterizavimo metodo modifikacijos pasižymėjo geresniais rezultatais, lyginant su nemodifikuotu metodu. Galima teigti, kad duomenų dimensijų mažinimo įtraukimas į duomenų klasterizavimo procesą turėjo teigiamą įtaką duomenų klasterizavimo metodo tikslumui.

3. Atlikus pasiūlytų ir kitų populiarių duomenų klasterizavimo metodų lyginamąją analizę, kuri buvo paremta 100 skirtingų duomenų rinkinių, paaiškėjo, kad sukurtas CBMIDE robastinis klasterizavimo metodas pasižymėjo geresniais rezultatais lyginant su *k* vidurkių klasterizavimu, GMM ir BGMM naudojant „Atom“, „Diabetes“, „Ecoli“ ir CPU duomenų rinkinius. Rezultatų analizė rodo, kad CBMIDE metodas pasiekė vidutinį tikslumą (*Accuracy*) 0,868, „JScore“ – 0,797 ir Silueto koeficientą 0,399, o *k* vidurkių metodas – atitinkamai 0,795, 0,715 ir 0,459, GMM – 0,797, 0,720 ir 0,461, o BGMM – 0,809, 0,729 ir 0,437. Modifikuotas duomenų klasterizavimo metodas (RCBMIDE) pasiekė didesnį tikslumą, naudojant „German“ ir „Segment“ duomenų rinkinius, palyginti su konkurentais. Pavyzdžiui, CBMIDE

metodo tikslumas „German“ rinkinyje buvo 0,895, segmentų rinkinyje – 0,885, lyginant su *k* vidurkių metodais, kurie atitinkamai pasiekė 0,79 ir 0,81. Tačiau, analizuojant daugiau duomenų rinkinių (100 duomenų rinkinių), pastebėta, kad CBMIDE metodas nebuvo toks efektyvus naudojant specifinius duomenų rinkinius, tokius kaip „Spiral“, kuriuose yra spiralinė duomenų struktūra. Šiuose rinkiniuose CBMIDE tikslumas buvo kur kas mažesnis (0,59), lyginant su kitais metodais, pvz., DBSCAN ir HDBSCAN, kurie geriau tvarkėsi su sudėtingomis duomenų struktūromis. Analizuojant didelį duomenų rinkinį (300 000 stebinių) su skirtingais triukšmo lygiais ir klasterių daliniais sutapimais, pastebėta, kad CBMIDE metodas buvo taip pat efektyvus. Vertinant klasterizavimo tikslumą (*Accuracy*), esant skirtingiems triukšmo lygiams ir klasterių daliniams sutapimams, CBMIDE metodas pasiekė reikšmingus rezultatus. Kai klasteriai nesutampa, esant 1% triukšmo lygiui, CBMIDE metodo tikslumas buvo 1,000, o esant 50 % triukšmo lygiui – 0, 913. Esant mažai sutampantiems klasteriams, kai yra 1 % triukšmo lygis, CBMIDE metodas pasiekė 0,992 tikslumą, o esant 50 % triukšmo – 0,754. Esant stipriai sutampantiems klasteriams, CBMIDE metodo tikslumas, kai yra 1% triukšmo, buvo 0,916, o kai 50 % triukšmo – 0,559.

4. Nauji pasiūlyti duomenų klasterizavimo metodai buvo pritaikyti sprendžiant realias problemas su dideliais duomenų kiekiais. Duomenų klasterizavimas naudojant tekstinius darbo skelbimų ir naujienų duomenis parodė, kad sukurti duomenų klasterizavimo metodai yra labai tikslūs ir našūs net esant dideliam nestruktūruotų duomenų kiekiui. Atliekant klasterizavimą su tekstiniais darbo skelbimų duomenimis, CBMIDE metodas pasiekė aukštesnį tikslumą, lyginant su kitais metodais. Taip pat pastebima, kad tokie metodai, kaip DBSCAN ar HDBSCAN, negalėjo būti pritaikyti dėl didelio duomenų kiekio. Tačiau buvo pastebėta ir tam tikrų trūkumų. Pavyzdžiui, CBMIDE metodas reikalauja daugiau skaičiavimo laiko, dėl šios priežasties taikant šį metodą svarbu įvertinti, ar skaičiavimų laikas yra reikšmingas naudojimui.

5. SUMMARY

The evolving nature of data and the growing demand for data analysis underscore the importance of effective data analysis methods [1,2]. Clustering algorithms have become an essential tool in machine learning and data mining, by aiding in addressing challenges related to data processing and interpretation, especially when there is no prior information about the data. Clustering is the process of grouping data based on their characteristics, similarities, or relationships, with the aim of creating internally homogeneous and mutually distinct groups. Such data grouping can simplify complex problems, facilitate knowledge discovery, and provide deeper insights into the underlying data structure [3,4].

In recent years, clustering has been applied in various fields such as bioinformatics [5], image processing [6,7], natural language processing [8,9], social network analysis [10], and anomaly detection [11]. Clustering is significant because it helps understand complex data structures and reveals significant data patterns. Additionally, clustering enables the development of more advanced and effective methods for solving complex problems in various fields, thereby enhancing the overall understanding of those areas.

Scientific research investigating robust data clustering and its applications particularly focuses on real data [12–14]. This includes challenges such as noise, outliers, extreme values, and missing or corrupted data. Real-world data sets may have complex geometric cluster shapes, thereby rendering traditional clustering methods inefficient. Robust clustering aims to eliminate these limitations by using advanced methods that can cluster diverse data and uncover complex structures. To address the noise problem, different robust data clustering algorithms are being developed, allowing for accurate clustering even in the presence of high noise or outlier data.

Object, Subject, and Measures of the Study

The object of this study is a set of heteroscedastic data that needs to be clustered into homogeneous groups. The subjects of the study are robust data clustering algorithms/methods that are resistant to noise and outliers. In this study, robust data clustering methods are defined as methods resistant to noise and outliers. Various measures such as Accuracy, JScore, and the Silhouette coefficient are used to compare the Accuracy of data clustering methods.

Aim of the Study

To create and investigate data clustering methods that are robust and efficient compared to other currently existing data clustering methods in cases of heterogeneous data and/or when noise and extreme values are present in the data, and to evaluate the effectiveness of the methods in the absence of noise.

Objectives of the Study

1. Review and investigate the already existing data clustering methods, their operating principles, differences, and practical applications currently, by addressing

their encountered problems and modifications that can function in case of noise or missing values.

2. Develop new robust algorithms/methods for data clustering.

3. Conduct a comparative analysis of the proposed and other popular clustering algorithms/methods.

4. Apply the newly proposed clustering algorithms/methods to real data sets, and determine their advantages and disadvantages compared to alternative methods.

Methodology of the Study

This dissertation applies various methods of the probability theory, mathematical statistics, data dimension reduction, and visualization techniques. The presented clustering methods are based on introducing the density estimate of the reverse formula.

Defended Statements

1. The proposed robust data clustering methods, based on the density estimate of the reverse formula, exhibit more accurate clustering compared to the currently used methods in the case of synthetic and real data.
2. The proposed modifications of robust clustering methods that apply data dimension reduction techniques consistently show better results than the developed data clustering method.

Scientific Novelty and Practical Significance

The developed data clustering algorithms based on the density estimate of the reverse formula extend the works of M. Kavaliauskas [15] and T. Ruzgas [16]. The proposed data clustering methods and research results have been presented at 12 scientific conferences and published in 5 scientific articles. Additionally, the developed data clustering algorithms were employed during scientific projects and in practical activities of the company *UAB Hostinger*. The application of data clustering in practical activities enabled the formation of business-significant data groups, their interpretation, and their application in company processes. In scientific projects, data clustering has been applied to economic data and is being implemented in the developed commercial platform.

The new data clustering methods presented in this work, as well as their application recommendations, have been cited and used in their research by Farmer *et al*. (2023) [17], Powroźnik *et al*. (2022) [18], Yu *et al*. (2023) [19], Chen *et al*. (2023) [20], Gebrael *et al*. (2023) [21], Wei and Song (2023) [22], Eltayeb *et al*. (2023) [23], Lin *et al*. (2023) [24], Roeksiri *et al*. (2023) [25], Leon *et al*. (2024) [26], Uskenov *et al*. (2024) [27], Nurduhan and Kuleyin (2024) [28], Senger *et al*. (2024) [29], Chweidan *et al*. (2024) [30], Goldstein *et al*. (2024) [31].

Included articles and contributions of co-authors to papers

Copies of the five articles listed below form the basis of this thesis. Articles X1, X2, X3, X4, X5 were published with open access (Open Access). Therefore, no permission from the publisher is required.

Article 1. T.Ruzgas, M.Lukauskas and G.Čepkauskas. (2021). Nonparametric Multivariate Density Estimation: Case Study of Cauchy Mixture Model. *Mathematics 9*, No. 21: 2717. Access online: https://doi.org/10.3390/math9212717

Article 2. M.Lukauskas and T.Ruzgas. (2022). A New Clustering Method Based on the Inversion Formula. *Mathematics 10*, No. 15: 2559. Access online: https://doi.org/10.3390/math10152559

Article 3. M.Lukauskas and T.Ruzgas. (2023). Reduced Clustering Method Based on the Inversion Formula Density Estimation. *Mathematics 11*, No. 3: 661. Access online: https://doi.org/10.3390/math11030661

Article 4. M. Lukauskas, V. Pilinkienė, J. Bruneckienė, A. Stundžienė, A. Grybauskas and T. Ruzgas. 2022. Economic Activity Forecasting Based on the Sentiment Analysis of News. *Mathematics 10*, No. 19: 3461. Access online: https://doi.org/10.3390/math10193461

Article 5. M. Lukauskas, V. Šarkauskaitė, V. Pilinkienė, A. Stundžienė, A. Grybauskas and J. Bruneckienė. Enhancing Skills Demand Understanding through Job Ad Segmentation using NLP and Clustering Techniques. *Applied Sciences*, 2023, 13(10):6119. Access online: https://doi.org/10.3390/app13106119

Artificial intelligence was first mentioned in 1956, but the sharpest jump in its use can be seen only in the last decade. The ever-increasing computing power of computers is driving the ever-increasing availability, development, and application of artificial intelligence in various fields of practice as well as science. Data mining is one of the most important areas of analytics because it is not limited to business, manufacturing, or service areas. For this reason, data mining has attracted a large number of researchers. Data clustering is one area of data mining that falls under unsupervised machine learning. Cluster analysis aims to divide data into separate classes/groups by discovering the internal structure of the objects/observations in the data set and their relationships. The purpose of cluster analysis is to divide similar elements/observations into separate groups by evaluating the degree of similarity of these observations. In data clustering, the expectation is that elements within the same cluster will be as similar as possible to each other, whereas elements from different clusters will be as dissimilar as possible [36]. The emergence of data clustering is associated with various scientific disciplines such as biology, statistics, and psychology. It dates back to the early 20th century. The history of cluster analysis can be traced back to the famous British biologist Sir Ronald Fisher who, in 1936, presented the *Iris* data set, which later became the basis for developing various clustering algorithms. Over the years, clustering has become an important tool in analytics, especially in data mining, machine learning, and pattern recognition. Cluster analysis is extremely important for academic research because it allows researchers to uncover meaningful patterns and trends in large and complex data sets. Clustering helps identify the underlying data structures, relationships, and

associations by grouping similar data points based on their inherent characteristics, ultimately facilitating knowledge discovery. As a result, cluster analysis has found wide application in fields as diverse as marketing, finance, medicine, and the social sciences, greatly enriching the academic literature and developing more informed, data-driven decision-making paradigms.

In order to apply data clustering in practice, many clustering methods have been proposed, which can be divided into five main groups: partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based clustering methods. In this subsection, we provide a detailed explanation of each category of data clustering methods, the most popular clustering methods in this category, their performance, and review their advantages and disadvantages.

Although there are many different approaches to data clustering, clustering is an important area of data science which is constantly evolving. It is of top importance to develop new methods adapted to the currently available data to cope with the emerging challenges of this data. One of the biggest challenges of data clustering is the ever-increasing amount of data and its complexity, which makes it important to have methods with high speed. Another important factor is noise; data outliers are often present in real-world datasets. Therefore, developing reliable data clustering methods to effectively eliminate such shortcomings becomes extremely important. In addition, the increasing diversity of data types such as text, images, and graphs requires more versatile clustering techniques that can handle different data structures and representations.

No single data clustering method is universally effective for all types of data sets or individual data sets, which is another reason why clustering methods continue evolving. Each clustering method has its distinctive features, advantages, and disadvantages. Some data clustering methods work well only with discrete spherical data clusters but perform poorly with overlapping or differently shaped clusters. This shows that the data clustering method depends on the data set and its main characteristics. In addition, the further development of data clustering methods contributes to the development of other areas of data science and artificial intelligence since many methods also use data clustering.

In recent years, data clustering has been particularly growing in various fields of business and science, as the amount and complexity of data is also growing rapidly. Considering this, it becomes important to evaluate the current applications of clustering methods to understand their impact, what challenges data clustering methods are characterized by, and in which directions data clustering methods can be improved. This chapter provides an overview of the applications of clustering techniques, the benefits of these applications, and the challenges faced by these application cases. This assessment is significant because it is important to assess what the new methods being developed should be characterized by and how these methods can be applied.

Review of the article "Nonparametric Multivariate Density Estimation: Case Study of Cauchy Mixture Model."

Contribution to the article by M. Lukauskas: the article presents M. Lukauskas as the contact person (the corresponding author) who performed experimental calculations, submitted the initial version of the manuscript together with the co-authors, and made the necessary corrections to the article in communication with the reviewers and the editor.

Summary of the article: This paper thoroughly examines five primary nonparametric multivariate density estimation methods, with no assumptions made regarding the nature of the data from any known parametric distribution family. A formulation method is devised, including the noise cluster in the collective mixture model. The efficacy of this method is illustrated through a simulation study. Evaluating the probability density function is deemed a crucial task in statistical modeling, as it depicts random variables as functions of other variables, thereby establishing relationships within the data.

The results are presented in tables detailing the mean, the standard deviation of the density from 100,000 samples, and the absolute percentage error of the estimate. It is evident from the findings that the most accurate results are secured by the SKDE and MIDE methods when the variables are such that n=100, d=5. Similarly, when the SKDE and MIDE methods again obtain q=2 and n=200, superior results are observed.

The outcomes exhibit that when variables are assigned as q=3, and n=200, with highly overlapping distributions (i=1, 2), the highest possible results can be attained by implementing the SKDE method. In the case of more separate distributions ($i \geq 3$), the MIDE method provides the best results. Notably, when q=3, n=400, it is evident that the SKDE method offers the best results, with the MIDE method coming in second.

Furthermore, when q=4 and n=400 with highly overlapping distributions ($i \leq 3$), the best results are secured via the SKDE method. Meanwhile, the MIDE method is more effective for increasingly separated distributions ($i \geq 4$). The SKDE method delivers the best results for highly overlapping or moderately separated distributions ($i \leq 5$). However, the MIDE method is the best for more separated distributions (i=6). When examining results with q=2 and n=50, it is discernible that the best overall results can be achieved by using the AKDE method for overlapping and separated distributions. More specifically, regarding significantly separated distributions ($i = 6$), the MIDE method is ideal when p1=0.6 and p2=0.4; when considering results where q=3 and n=50, the AKDE method consistently yields the best outcomes across all instances, embracing highly overlapping and isolated distributions.

The LSDE method, when dealing with significant outliers ($| x\text{-}mj |> 100uj$), tends to cluster a larger number of values closer to the centroid. For lower-dimensional distributions (d=2), the SKDE method produces the most effective results for cases of both large and small overlap ($i < 4$). Conversely, the MIDE method yields satisfying outcomes for isolated distributions ($i \geq 5$).

This paper critically examines the most prevalent and extensively utilized nonparametric density estimation algorithms and devises an inversion formula for

density estimation. It has been determined that integrating the noise cluster into the method significantly improves the results of the inversion formula. Given its performance in terms of the mean absolute error, the adaptive kernel method is suggested for higher-dimensional cases (d~5) as well as for smaller sample sets (n~50).

When the number of observations exceeds approximately n~100, the modified inversion formula method offers the most favorable results. The semiparametric kernel method is advised for larger sample sets with overlapping distributions. In contrast, the modified inversion formula method is recommended for more isolated distributions. Based on the mean absolute percentage error, the semiparametric kernel method is also suggested for samples with overlapping distributions.

In the case of two dimensions (d~2) and samples with overlapping distributions, the semiparametric kernel method is the best choice. Meanwhile, for discrete distributions, the adaptive kernel method is advised.

Review of the article "A New Clustering Method Based on the Inversion Formula."

Contribution to the article by M. Lukauskas: M. Lukauskas is presented as the contact person (the corresponding author). M. Lukauskas created a data clustering method based on the inversion formula density estimation, performed experimental calculations, and prepared the final version of the article.

The burgeoning fields of data science continue to emphasize the advancement of new density estimation processes [86,87]. While a previous article contrasted density estimation with other methods, the current one extends density estimation to data clustering. Numerous robust density estimation methods have been proposed in recent years, all of which are based on neural networks. Various studies have been dedicated to this area, including Parzen neural networks [88], soft-constrained neural networks [89], and several other alternatives [90].

Evaluation of probability density functions (pdf) is deemed to represent a fundamental aspect of statistical modeling, by transforming random variables into functions of other variables. This facilitates the discovery of covert relationships within data [91]. In many machine learning algorithms, defining a hitherto unknown function – the density of the data distribution – has been found to be of critical importance. Notably, the distribution density function plays a pivotal role in Bayesian classifiers [92,93], density-based clustering algorithms [94–97], and information-based feature selection algorithms [98,99].

Effective density estimates require meticulous pre-construction to successfully extract unknown probability density functions. There is sustained interest in developing novel density estimation procedures [86,87].

This paper proposes that a density estimation founded on a modified inversion formula is aptly suited for data clustering. A primary objective of this document is to introduce a new density clustering technique derived from a modified estimation of the inversion formula's density. This advanced method allows for a higher degree of

precise clustering than popular methodologies like K-means and the Gaussian mixture, as well as other techniques outlined in the literature review.

In conclusion, this paper outlines a novel clustering approach based on the modified inversion formula density estimation. According to a preceding article, the density estimator of the modified inversion formula surpasses other methodologies in estimating density. Consequently, a new clustering technique has been devised and examined in this paper. The updated method, derived from the modified inversion formula, apparently registers commendable performance on various datasets relative to the K-means, Gaussian, and Bayesian Gaussian mixture models.

Despite the noteworthy outcomes, there are limitations to consider. The new method cannot process higher-dimensional data (d>15). Additionally, based on the results, it is apparent that the MIDEv2 method exhibits the highest-level performance on the generated data with noise across all datasets (0.5%, 1%, 2%, and 4% noise). Intriguingly, even without noise/outliers, the new method (buttressed by the inversion formula) demonstrates a notable proficiency at clustering data – as witnessed by using one of the most popular datasets, the *Iris*.

Review of the article "Reduced Clustering Method Based on the Inversion Formula Density Estimation."

Contribution to the article by M. Lukauskas: in the article, M. Lukauskas is presented as the contact person (the corresponding author). M. Lukauskas has created a data clustering method based on the inversion formula density estimation and data dimensionality reduction methods, performed experimental calculations, prepared the article, communicated with co-authors, reviewers, and edited the article as an editor.

Article summary. This scholarly research paper advances the work conducted in a previous study by introducing data clustering methodologies derived from the modified inversion formula density estimation. It was observed in the prior study that the newly instituted data clustering method, referred to as *CBMIDE* (*Clustering Based on Modified Inversion Formula Density Estimation*), exhibited subpar performance for higher data dimensionality (d > 10).

The research paper under examination proposes modifying the CBMIDE methodology to better accommodate high-dimensional data – thus enhancing the method's accuracy for both high and low dimensions. The paper explores the effects of data dimensionality reduction techniques on the outcomes of data clustering, specifically when using the modified density estimation formula.

The dimensions of a given dataset often crucially influence the precision of the clustering methods and the calculation time and resources necessary for implementing data clustering techniques. It has been typically observed that, as the data dimension increases, the time required for clustering also proportionally escalates – potentially even exponentially.

A viable solution is data dimensionality reduction, which is achievable via various techniques. Integration of this reduction approach with data clustering is a prevalent practice which allows for substantial resource conservation. The usage of dimensionality reduction in data clustering and the application of reduced dimensions

have been extensively discussed in scientific literature. Initial methods combined simple K-means and PCA techniques [52].

A combination of these methods continues to be relevant today, primarily attributed to the annual data surge. A prominent field for the application of these combined techniques is gene analysis, where dimensionality reduction methods like *Principal Components* (PCA) [53], *Non-Negative Matrix Factorization* (NMF) [54], and *Independent Component Analysis* (ICA) [55] are utilized. Furthermore, clustering methods, such as K-means and DBSCAN, are employed to study gene sequences [56]. There is also the observation of more intricate data dimensionality reduction methods, including t-SNE [57], UMAP [58], and various method combinations [59–64].

In the research under consideration, modifications were made to the data clustering methodology CBMIDE to make it applicable for clustering high-dimensional data. This paper proposes that the employment of data dimensionality reduction techniques could maintain the accuracy of data clustering. Furthermore, it asserts that the novel RCBMIDE method holds a comparative advantage over the preceding procedure and other popular methodologies. This paper employs a more extensive range of data dimensionality reduction and clustering methods to contrast the results.

In conclusion, this study modifies the previously presented data clustering method, RCBMIDE, to circumvent the high-dimensionality issues inherent in the earlier method (CBMIDE). To achieve this change, the data clustering algorithm underwent a significant modification, thereby suggesting that data dimensionality reduction could solve the problem of data dimensionality. Various data dimensionality reduction methods were utilized individually to reduce the dimensions, each presenting a unique modification to the clustering method. Applying the new process in data clustering indicated a favorable impact on the results of data clustering.

The developed data clustering approach displayed competitive outcomes compared to alternative data clustering methodologies. The altered clustering method was compared with the original data clustering framework using low-dimensional datasets. This comparison showed that the newly developed data clustering method showed superior results, even for low-dimensional data, in relation to the preceding one.

The findings support the initial hypothesis regarding the beneficial effects of the method modification on improving data clustering outcomes. However, it is important to note that there is no straightforward decision on the best method as the applied data dimensionality reduction methods vary in every scenario. The reliability metric enabled the comparison of the extent of dimensionality reduction and the subsequent clustering performance. It was also observed that the reduction methods from the same family, UMAP and TriMap, usually yielded the best results for low dimensions. This study substantiates the hypothesis concerning the application of the developed method to high-dimensional data.

Review of the article "Economic Activity Forecasting Based on the Sentiment Analysis of News."

Contribution to the article by M. Lukauskas: M. Lukauskas is presented as the corresponding author and is the first author of the article. M. Lukauskas compiled the data set, conceptualized the idea of applying data clustering to the ongoing research, conducted experimental research, prepared the article, communicated with the co-authors and reviewers, and edited the article.

Article summary. This article represents a shift from the theoretical research and method development towards practical problem-solving applications. The article was implemented as part of a project backed by the *European Regional Development Funds* (No. 13.1.1-LMT-K-718-05-0012) as a response to the COVID-19 pandemic. Data clustering, a widely employed technique, is used here to model economic activity. Evaluations of a country's economic vitality are often linked with gross domestic product measurements or changes in the industrial production, as these parameters reflect the nation's industrial activities [109–111].

The paper utilizes data clustering and natural language processing techniques. In the initial research stage, extensive textual data was collected and stored. Sentiment analysis was subsequently performed on this data by using various methods. Data clustering was advantageous here, as the pre-segmentation of data into unidentified categories facilitated a better interpretation of the results.

The core aim of this study was to classify all news texts to ascertain the sentiments within these groups. This task was accomplished through cluster analysis. As presented in prior articles, the most prevalent data clustering method, K-means, was used alongside the CBMIDE clustering method. Other specific clustering methods were also used for the analyzed data to determine the most appropriate clustering technique. These included Gaussian Mixture Models, Bayesian Gaussian Mixture Models, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [24], BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) [116], and OPTICS (Ordering Points To Identify the Clustering Structure) [28]. Various parameter adjustments were applied to these models with the objective to identify the most effective clustering approach.

In the initial news clustering phase, all textual data was numerically converted by using Sentence-Transformers. This transformed textual data into a 384-dimensional dataset, where each text was allocated a specific point in the space based on sentence words, meanings, and semantic implications. These points were then clustered by employing different clustering methods. The results were compared by using the metrics discussed earlier, with the best-performing methods to be used in further research.

In conclusion, this study involved substantial data clustering. However, the performance of data clustering with such a large dataset was not as favorable as initially anticipated. Despite this, some proposed clustering models, commonly used in practice, were still applicable to this research, allowing assessment of the impact of clustering on sentiment analysis and forecasting in news stories. An important factor and limitation is that this paper utilized article headings and leads, thus omitting the article's full structure. Nonetheless, based on the title and sentiment analysis, the paper confirmed the hypothesis of the influence of sentiments on economic activity.

Furthermore, the development of a Lithuanian sentiment analysis model is underway, thus eliminating the need for text translation. This would enable pure texts to be used directly for negative sentiment extraction. To summarize, the research found supporting evidence for the hypothesis that negative news sentiment is linked to economic activity.

Review of the article "Enhancing Skills Demand Understanding through Job Ad Segmentation using NLP and Clustering Techniques."

Contribution to the article by M. Lukauskas: In the article, M. Lukauskas is presented as the first author of the article and the contact person of the article. The author compiled a data set, conceptualized the idea of applying data clustering to ongoing research, conducted experimental research, prepared the article, communicated with the co-authors and reviewers, and edited the final version of the article.

Article summary. This article proposes continuing the application of the developed methodologies for real-world problem-solving. The presented research is part of a project initiated in response to the COVID-19 pandemic, backed by the *European Regional Development Funds* (No. 13.1.1-LMT-K-718-05-0012). Specifically, this article introduces an advanced automated method to assess the requisite skillsets of employees within the labor market context.

The ongoing digitization process poses numerous challenges not only for companies, but also for their workforce. Many tasks traditionally performed by human operators, such as data entry, accounting, and assembly line work, are now assisted or even replaced by digital technologies and artificial intelligence, leading to a significant shift in the work ecosystem [1,2]. This situation is further complicated by the rising demand for new vocations and skill sets due to rapid digitalization [5].

Automating large sections of the workforce precipitates issues that need to be urgently addressed. Companies are increasingly facing a workforce that is insufficiently equipped with the necessary skills to effectively implement and sustain the introduction and integration of emerging technologies in their operations. This article sees the addressing of this skill gap as a major challenge that various organizations are presently being confronted with, intensified by the ongoing digital evolution. However, it presents a unique solution in the form of an automated method for determining the required skills of employees in the rapidly changing labor market. This innovation could prove instrumental in offering a strategic direction for training and education strategies, thereby ensuring that the workforce remains relevant in a rapidly evolving job market.

Typically, skill analysis in organizations has been conducted via traditional survey methodologies. However, such methods present limitations, such as inflexible schedules and difficulty tracking the dynamism intrinsic to skill evolution. One avenue to efficiently comprehend the requisite skill involves the application of *Artificial Intelligence* (AI) methodologies. With advancements in AI, it is now possible to extract, process, and interpret the varying needs for skills in a rapidly changing workforce with remarkable speed and automation.

The heart of this research is the use of *Natural Language Processing* (NLP) and data clustering methodologies. The intention is to analyze the skill requirements of employees in Lithuania, conduct cluster analysis on the derived skill data, and construct nuanced employee profile archetypes. The primary objectives of this study encompass the analysis of specific skill dynamics, the juxtaposition of various clustering methodologies, and the identification of the dominant employee profile types within the Lithuanian labor market.

A significant takeaway from this study is its contribution of valuable insights into the evolving needs of the Lithuanian labor market. Further, it proposes an innovative, data-driven approach capable of evaluating changes in the labor skill demand in real time. This mitigates the efficiency issues of traditional survey-based methods and ensures that businesses stay tuned with the trending and future skill requirements, thus enabling them to make informed decisions about workforce training and development.

Conclusions. The primary objective of this article was to investigate the potential of natural language processing, data clustering, and other machine learning techniques in determining the job profile requirements, specifically within the Lithuanian context. The discussion primarily emphasizes basic strategies for extracting data from publicly accessible unstructured sources. Further, the article explores the subsequent stages of data processing using a range of vectorization strategies, evaluating the pros and cons of each method. The research found that the regex method yielded optimum results for extracting data from Lithuanian job advertisements. Regarding data vectorization and feature extraction, the study employed two primary techniques – TF-IDF and sentence transformers (the BERT model). Due to the high dimensionality of the feature-extracted data (384 dimensions), dimensionality reduction methods were utilized, with the ISOMAP method producing the most favorable outcomes. Eleven thousand dimensionality reduction models were constructed and evaluated in total to discover the most effective dimensional reduction methods and their associated parameters, and to ascertain the optimal parameters for each technique.

However, other methods, including UMAP, also yielded commendable results. The crucial factor for automated job advertisement analysis was the speed, resulting in the selection of the UMAP method for the research. The UMAP method demonstrated a higher processing speed than the alternatives, such as t-SNE or ISOMAP, which is a characteristic that is crucial for regularly analyzing job adverts. While the PCA method could have been employed for enhanced speed, it would have likely delivered lower-quality results than UMAP. It should be noted that a mere increase in the hidden dimensions did not significantly enhance the data dimensionality reduction results. The applied data dimensionality reduction techniques did not yield any improved outcomes beyond 20 latent dimensions.

Consequently, 20 latent dimensions may also be employed under more extensive system constraints. This study utilized 50 latent dimensions extracted from the earlier-discussed UMAP method for dimensionality reduction. It was ascertained that the dimensionality reduction of this specific data, even for extremely low quantities of latent dimensions such as 2.5, exhibited confidence metric values

exceeding 0.9. The article systematically reviews a multitude of data clustering techniques. Over 3000 distinctive models were developed for data clustering purposes to optimize parameters about various methods. The HDBSCAN technique emerged as the most effective choice in data clustering, primarily owing to its hierarchical organizational structure. Concurrently, the RCBMIDE technique also demonstrated appreciable results about the metrics utilized in the research, facilitating the exclusion of outliers and incorporating data that accurately represents the information. The job profiles derived based on both methods exhibited significant similarity. However, based on metrics, HDBSCAN was chosen as the principal method for data clustering. It was observed that, when implementing the widely utilized K-means method, the most effective clustering result was achieved with a mere 5 clusters, which is deemed an exceptionally small number for representing country job profiles.

Following a comprehensive evaluation of job profiles, it was clear that the extraction quality was commendable, further suggesting the feasibility of these methods for automated profiling. Analysis of the delineated profiles indicated that understanding some skills, such as 'German' and 'cats', proved more intricate. These findings underscore the limitations of this approach when dealing with harder-to-interpret skills.

In this context, 'German' was observed to denote expertise in the German language, while 'Cat' correlated with job advertisements oriented towards animal care or similar roles. Leveraging the resultant job profiles, these were detailed by using an automated, generative artificial intelligence algorithm. Various models of generative artificial intelligence were employed to articulate these job profiles.

The cutting-edge GPT-4 language model displayed the most promising results, although the GPT-3.5 model also exhibited respectable performance. However, older variants of generative AI models yielded less satisfactory outcomes, and the crafted job descriptions need additional refinement and analysis for real-world applicability.

CONCLUSIONS

1. During the scientific research, popular data clustering methods, their principles of operation, limitations, and applications both in scientific research and practical activities were thoroughly examined, and robust modifications of different methods/algorithms were analyzed. Analysis of scientific literature revealed that more than 100 different data sets are used in the data clustering research, which is divided into two main groups: real data, and synthetic data. It was observed that data sets with true class values are often used in these studies because they help assess clustering accuracy more easily. For this reason, it can be argued that data classification sets are often used in scientific research when developing new data clustering methods. Data clustering methods are divided into several main groups: partitioning methods, density-based methods, grid-based and model-based methods, and hierarchical clustering methods. It is noted in the scientific research that the

K-means method is still the most commonly used data clustering method in comparisons.

2. Based on the research results, a robust data clustering method based on the application of a modified reverse formula has been created and proposed. Studies have proven that the density estimate of the reverse formula operates more accurately compared to other methods. Therefore, a data clustering method was developed based on this created density estimate. The developed data clustering method had limitations when dealing with high data dimensions; therefore, different method modifications were created to address this issue. Modifications to the created data clustering method are based on reducing data dimensions. The data clustering method modifications showed better results compared to the unmodified method. It can be stated that the inclusion of data dimension reduction in the data clustering process had a positive impact on the accuracy of the data clustering method.
3. A comparative analysis of the proposed and other popular data clustering methods, based on 100 different data sets, revealed that the developed CBMIDE robust clustering method showed better results compared to K-means clustering, GMM, and BGMM using Atom, Diabetes, Ecoli, and CPU data sets. The analysis of the results shows that the CBMIDE method achieved an average accuracy of 0.868, a JScore of 0.797, and a Silhouette coefficient of 0.399, while the K-means method yielded 0.795, 0.715, and 0.459, respectively, whereas, the scores for GMM were 0.797, 0.720, and 0.461, and those of BGMM were 0.809, 0.729, and 0.437. The modified data clustering method (RCBMIDE) achieved higher accuracy when using the German and Segment data sets compared to competitors. For example, the CBMIDE method's accuracy in the German set was 0.895, and in the segment set it measured 0.885, compared to the K-means methods, which respectively achieved 0.79 and 0.81. However, when analyzing more data sets (100 data sets), it was observed that the CBMIDE method was not as effective when using specific data sets, such as Spiral, which has a spiral data structure. In these sets, the CBMIDE accuracy was significantly lower (0.59) compared to other methods, such as DBSCAN and HDBSCAN, which handled complex data structures better. When analyzing a large data set (300,000 observations) with different levels of noise and cluster overlaps, it was observed that the CBMIDE method was also effective. By evaluating the clustering accuracy (Accuracy) at different noise levels and cluster overlaps, the CBMIDE method achieved significant results. In the case of non-overlapping clusters with 1% noise level, the CBMIDE method's accuracy was 1.000, whereas, at 50% noise level, it scored 0.913. In the case of slightly overlapping clusters,

at 1% noise level, the CBMIDE method achieved an accuracy of 0.992, and, at 50% noise, it yielded 0.754. In the case of highly overlapping clusters, the CBMIDE method's accuracy at 1% noise was 0.916, while, at 50% noise, it was 0.559.

4. The newly proposed data clustering methods were applied to solve real-world problems with large data volumes. Performing data clustering using textual job advertisements and news data showed that the created data clustering methods are denoted by high accuracy and performance even with a large amount of unstructured data. During clustering with textual job advertisement data, the CBMIDE method achieved higher accuracy compared to other methods. It is also notable that methods like DBSCAN or HDBSCAN could not be used due to the large amount of data. However, some downsides have also been observed. For example, the CBMIDE method requires more computation time; therefore, it is important to assess whether the computation time is significant for the application when using this method.

LITERATŪROS SĄRAŠAS

1. Naeem, M.; Jamal, T.; Diaz-Martinez, J.; Butt, S.A.; Montesano, N.; Tariq, M.I.; De-la-Hoz-Franco, E.; De-La-Hoz-Valdiris, E. Trends and future perspective challenges in big data. In Proceedings of the Advances in Intelligent Data Analysis and Applications: Proceeding of the Sixth Euro-China Conference on Intelligent Data Analysis and Applications, 15–18 October 2019, Arad, Romania, 2022; pp. 309-325.
2. Igual, L.; Seguí, S. Introduction to data science. In *Introduction to Data Science: A Python Approach to Concepts, Techniques and Applications*; Springer: 2024; pp. 1-4.
3. Oyewole, G.J.; Thopil, G.A. Data clustering: application and trends. *Artificial Intelligence Review* **2023**, *56*, 6439-6475.
4. Hashemi, S.E.; Gholian-Jouybari, F.; Hajiaghaei-Keshteli, M. A fuzzy C-means algorithm for optimizing data clustering. *Expert Systems with Applications* **2023**, *227*, 120377.
5. Xu, Z.; Park, T.-J.; Cao, H. Advances in mining and expressing microbial biosynthetic gene clusters. *Critical reviews in microbiology* **2023**, *49*, 18-37.
6. Fang, U.; Li, J.; Lu, X.; Mian, A.; Gu, Z. Robust image clustering via context-aware contrastive graph learning. *Pattern Recognition* **2023**, *138*, 109340.
7. Huang, X.; Hu, Z.; Lin, L. Deep clustering based on embedded auto-encoder. *Soft Computing* **2023**, *27*, 1075-1090.
8. George, L.; Sumathy, P. An integrated clustering and BERT framework for improved topic modeling. *International Journal of Information Technology* **2023**, *15*, 2187-2195.
9. Chen, W.; Hu, H.; Li, Y.; Ruiz, N.; Jia, X.; Chang, M.-W.; Cohen, W.W. Subject-driven text-to-image generation via apprenticeship learning. *Advances in Neural Information Processing Systems* **2024**, *36*.
10. Trillo, J.R.; Herrera-Viedma, E.; Morente-Molinera, J.A.; Cabrerizo, F.J. A large scale group decision making system based on sentiment analysis cluster. *Information Fusion* **2023**, *91*, 633-643.
11. Samariya, D.; Thakkar, A. A comprehensive survey of anomaly detection algorithms. *Annals of Data Science* **2023**, *10*, 829-850.
12. Heaton, H.; Talman, A.M.; Knights, A.; Imaz, M.; Gaffney, D.J.; Durbin, R.; Hemberg, M.; Lawniczak, M.K. Souporcell: robust clustering of single-cell RNA-seq data by genotype without reference genotypes. *Nature methods* **2020**, *17*, 615-620.
13. Yang, B.; Wu, J.; Sun, A.; Gao, N.; Zhang, X. Robust landmark graph-based clustering for high-dimensional data. *Neurocomputing* **2022**, *496*, 72-84.
14. Palomino-Echeverria, S.; Huergo, E.; Ortega-Legarreta, A.; Uson, E.M.; Aguilar, F.; de la Pena, C.; Lopez-Vicario, C.; Alessandria, C.; Laleman, W.; Farias Queiroz, A. ClustALL: A robust clustering strategy for stratification of patients with acutely decompensated cirrhosis. *medRxiv* **2023**, 2023.2011. 2017.23298672.

15. Kavaliauskas, M.; Rudzkis, R.; Ruzgas, T. The projection-based multivariate density estimation. *Acta et Commentationes Universitatis Tartuensis de Mathematica* **2004**, *8*, 135-141.
16. Ruzgas, T. The Nonparametric Estimation of Multivariate Distribution Density Applying Clustering Procedures. Institute of Mathematics and Informatics Vilnius, Lithuania, 2007.
17. Farmer, J.; Allen, E.; Jacobs, D.J. Quasar Identification Using Multivariate Probability Density Estimated from Nonparametric Conditional Probabilities. *Mathematics* **2023**, *11*, 155.
18. Powroźnik, P.; Szcześniak, P.; Turchan, K.; Krysik, M.; Koropiecki, I.; Piotrowski, K. An Elastic Energy Management Algorithm in a Hierarchical Control System with Distributed Control Devices. *Energies* **2022**, *15*, 4750.
19. Yu, J.; Duan, Q.; Huang, H.; He, S.; Zou, T. Effective Incomplete Multi-View Clustering via Low-Rank Graph Tensor Completion. *Mathematics* **2023**, *11*, 652.
20. Chen, J.; Shi, Y.; Sun, J.; Li, J.; Xu, J. Base Station Planning Based on Region Division and Mean Shift Clustering. *Mathematics* **2023**, *11*, 1971.
21. Gebrael, G.; Sahu, K.K.; Chigarira, B.; Tripathi, N.; Mathew Thomas, V.; Sayegh, N.; Maughan, B.L.; Agarwal, N.; Swami, U.; Li, H. Enhancing Triage Efficiency and Accuracy in Emergency Rooms for Patients with Metastatic Prostate Cancer: A Retrospective Analysis of Artificial Intelligence-Assisted Triage Using ChatGPT 4.0. *Cancers* **2023**, *15*, 3717.
22. Wei, C.; Song, Z. Real-Time Forecasting of Subsurface Inclusion Defects for Continuous Casting Slabs: A Data-Driven Comparative Study. *Sensors* **2023**, *23*, 5415.
23. Eltayeb, R.Y.; Karrar, A.E.; Osman, W.I.; Ali, M.M. Handling Imbalanced Data through Re-sampling: Systematic Review. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)* **2023**, *11*.
24. Lin, J.C.-W.; Tomasiello, S.; Srivastava, G. Integrated Artificial Intelligence in Data Science. **2023**, *13*, 11612.
25. Roeksiri, N.; Amphawan, K. CID-CJA: Co-Occurrence Information Discovery in Computer-Related Job Advertisements. In Proceedings of the 2023 27th International Computer Science and Engineering Conference (ICSEC), 2023; pp. 247-252.
26. Leon, F.; Gavrilescu, M.; Floria, S.-A.; Minea, A.-A. Hierarchical Classification of Transversal Skills in Job Ads Based on Sentence Embeddings. *arXiv preprint arXiv:2401.05073* **2024**.
27. Uskenov, R.; Yengsebek, T.; Kurzhykaev, Z.; Bostanova, S.; Akkair, B. The relationship between dry matter intake and the average daily gain. In Proceedings of the BIO Web of Conferences, 2024; p. 01006.
28. Nurduhan, M.; Kuleyin, B. Cluster-based Visualization of human element interactions in marine accidents. *Ocean Engineering* **2024**, *298*, 117153.
29. Senger, E.; Zhang, M.; van der Goot, R.; Plank, B. Deep Learning-based Computational Job Market Analysis: A Survey on Skill Extraction and Classification from Job Postings. *arXiv preprint arXiv:2402.05617* **2024**.

30. Chweidan, H.; Rudyuk, N.; Tzur, D.; Goldstein, C.; Almoznino, G. Statistical Methods and Machine Learning Algorithms for Investigating Metabolic Syndrome in Temporomandibular Disorders: A Nationwide Study. *Bioengineering* **2024**, *11*, 134.
31. Goldstein, A.; Shahar, Y.; Weisman Raymond, M.; Peleg, H.; Ben-Chetrit, E.; Ben-Yehuda, A.; Shalom, E.; Goldstein, C.; Shiloh, S.S.; Almoznino, G. Multi-Dimensional Validation of the Integration of Syntactic and Semantic Distance Measures for Clustering Fibromyalgia Patients in the Rheumatic Monitor Big Data Study. *Bioengineering* **2024**, *11*, 97.
32. McCarthy, J.; Minsky, M.L.; Rochester, N.; Shannon, C.E. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine* **2006**, *27*, 12-12.
33. Wu, T.; He, S.; Liu, J.; Sun, S.; Liu, K.; Han, Q.-L.; Tang, Y. A brief overview of ChatGPT: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica* **2023**, *10*, 1122-1136.
34. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Advances in neural information processing systems* **2017**, *30*.
35. Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* **2023**.
36. Ding, S.; Jia, H.; Du, M.; Xue, Y. A semi-supervised approximate spectral clustering algorithm based on HMRF model. *Information Sciences* **2018**, *429*, 215-228.
37. Koretsky, M.J.; Alvarado, C.; Makarious, M.B.; Vitale, D.; Levine, K.; Bandres-Ciga, S.; Dadu, A.; Scholz, S.W.; Sargent, L.; Faghri, F. Genetic risk factor clustering within and across neurodegenerative diseases. *Brain* **2023**, *146*, 4486-4494.
38. Huang, Z.; Zheng, H.; Li, C.; Che, C. Application of Machine Learning-Based K-Means Clustering for Financial Fraud Detection. *Academic Journal of Science and Technology* **2024**, *10*, 33-39.
39. Barrio-Hernandez, I.; Yeo, J.; Jänes, J.; Mirdita, M.; Gilchrist, C.L.; Wein, T.; Varadi, M.; Velankar, S.; Beltrao, P.; Steinegger, M. Clustering predicted structures at the scale of the known protein universe. *Nature* **2023**, *622*, 637-645.
40. Kasem, M.S.; Hamada, M.; Taj-Eddin, I. Customer profiling, segmentation, and sales prediction using AI in direct marketing. *Neural Computing and Applications* **2024**, *36*, 4995-5005.
41. Fernández-de-Las-Peñas, C.; Martín-Guerrero, J.D.; Florencio, L.L.; Navarro-Pardo, E.; Rodríguez-Jiménez, J.; Torres-Macho, J.; Pellicer-Valero, O.J. Clustering analysis reveals different profiles associating long-term post-COVID symptoms, COVID-19 symptoms at hospital admission and previous medical co-morbidities in previously hospitalized COVID-19 survivors. *Infection* **2023**, *51*, 61-69.

42. Thakur, J.; Kushwaha, B.P. Artificial intelligence in marketing research and future research directions: Science mapping and research clustering using bibliometric analysis. *Global Business and Organizational Excellence* **2024**, *43*, 139-155.
43. de Oliveira, M.S.; Steffen, V.; de Francisco, A.C.; Trojan, F. Integrated data envelopment analysis, multi-criteria decision making, and cluster analysis methods: Trends and perspectives. *Decision Analytics Journal* **2023**, 100271.
44. Lloyd, S. Least squares quantization in PCM. *IEEE transactions on information theory* **1982**, *28*, 129-137.
45. MacQueen, J. Classification and analysis of multivariate observations. In Proceedings of the 5th Berkeley Symp. Math. Statist. Probability, 1967; pp. 281-297.
46. Arthur, D.; Vassilvitskii, S. K-means++ the advantages of careful seeding. In Proceedings of the Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, 2007; pp. 1027-1035.
47. Bahmani, B.; Moseley, B.; Vattani, A.; Kumar, R.; Vassilvitskii, S. Scalable k-means++. *arXiv preprint arXiv:1203.6402* **2012**.
48. Sculley, D. Web-scale k-means clustering. In Proceedings of the Proceedings of the 19th international conference on World wide web, 2010; pp. 1177-1178.
49. Bezdek, J.C. *Pattern recognition with fuzzy objective function algorithms*; Springer Science & Business Media: 2013.
50. Wagstaff, K.; Cardie, C.; Rogers, S.; Schrödl, S. Constrained k-means clustering with background knowledge. In Proceedings of the Icml, 2001; pp. 577-584.
51. Dhillon, I.S.; Modha, D.S. Concept decompositions for large sparse text data using clustering. *Machine learning* **2001**, *42*, 143-175.
52. Hornik, K.; Feinerer, I.; Kober, M.; Buchta, C. Spherical k-means clustering. *Journal of statistical software* **2012**, *50*, 1-22.
53. García-Escudero, L.A.; Gordaliza, A.; Matrán, C.; Mayo-Iscar, A. A general trimming approach to robust cluster analysis. **2008**.
54. Dorabiala, O.; Kutz, J.N.; Aravkin, A.Y. Robust trimmed k-means. *Pattern Recognition Letters* **2022**, *161*, 9-16.
55. Gan, G.; Ng, M.K.-P. K-means clustering with outlier removal. *Pattern Recognition Letters* **2017**, *90*, 8-14.
56. Whang, J.J.; Dhillon, I.S.; Gleich, D.F. Non-exhaustive, overlapping k-means. In Proceedings of the Proceedings of the 2015 SIAM international conference on data mining, 2015; pp. 936-944.
57. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the kdd, 1996; pp. 226-231.
58. Schubert, E.; Sander, J.; Ester, M.; Kriegel, H.P.; Xu, X. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)* **2017**, *42*, 1-21.

59. Sander, J.; Ester, M.; Kriegel, H.-P.; Xu, X. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data mining and knowledge discovery* **1998**, *2*, 169-194.
60. Sander, J. *Generalized density based clustering for spatial data mining*; Herbert Utz Verlag: 1999.
61. Ankerst, M.; Breunig, M.M.; Kriegel, H.-P.; Sander, J. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod record* **1999**, *28*, 49-60.
62. Schubert, E.; Gertz, M. Improving the cluster structure extracted from optics plots. In Proceedings of the LWDA, 2018.
63. Campello, R.J.; Moulavi, D.; Sander, J. Density-based clustering based on hierarchical density estimates. In Proceedings of the Advances in Knowledge Discovery and Data Mining: 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part II 17, 2013; pp. 160-172.
64. Campello, R.J.; Moulavi, D.; Zimek, A.; Sander, J. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **2015**, *10*, 1-51.
65. Jang, J.; Jiang, H. DBSCAN++: Towards fast and scalable density clustering. In Proceedings of the International conference on machine learning, 2019; pp. 3019-3029.
66. Gonzalez, T.F. Clustering to minimize the maximum intercluster distance. *Theoretical computer science* **1985**, *38*, 293-306.
67. He, Y.; Tan, H.; Luo, W.; Mao, H.; Ma, D.; Feng, S.; Fan, J. Mr-dbscan: an efficient parallel density-based clustering algorithm using mapreduce. In Proceedings of the 2011 IEEE 17th international conference on parallel and distributed systems, 2011; pp. 473-480.
68. Birant, D.; Kut, A. ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & knowledge engineering* **2007**, *60*, 208-221.
69. Hinneburg, A.; Keim, D.A. *An efficient approach to clustering in large multimedia databases with noise*; Bibliothek der Universität Konstanz: 1998; Volume 98.
70. Saxena, A.; Prasad, M.; Gupta, A.; Bharill, N.; Patel, O.P.; Tiwari, A.; Er, M.J.; Ding, W.; Lin, C.-T. A review of clustering techniques and developments. *Neurocomputing* **2017**, *267*, 664-681.
71. Olson, C.F. Parallel algorithms for hierarchical clustering. *Parallel computing* **1995**, *21*, 1313-1325.
72. Jain, A.K.; Murty, M.N.; Flynn, P.J. Data clustering: a review. *ACM computing surveys (CSUR)* **1999**, *31*, 264-323.
73. Murtagh, F. A survey of algorithms for contiguity-constrained clustering and related problems. *The computer journal* **1985**, *28*, 82-88.
74. Rathore, P. Big data cluster analysis and its applications. University of Melbourne, Parkville, Victoria, Australia, 2018.
75. Jain, A.K.; Dubes, R.C. *Algorithms for clustering data*; Prentice-Hall, Inc.: 1988.

76. Wang, W.; Yang, J.; Muntz, R. STING: A statistical information grid approach to spatial data mining. In Proceedings of the Vldb, 1997; pp. 186-195.
77. Ouyang, G.; Dey, D.K.; Zhang, P. Clique-based method for social network clustering. *Journal of Classification* **2020**, *37*, 254-274.
78. Sheikholeslami, G.; Chatterjee, S.; Zhang, A. Wavecluster: A multi-resolution clustering approach for very large spatial databases. In Proceedings of the VLDB, 1998; pp. 428-439.
79. McNicholas, P.D. Model-based clustering. *Journal of Classification* **2016**, *33*, 331-373.
80. McLachlan, G.J.; Rathnayake, S. On the number of components in a Gaussian mixture model. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2014**, *4*, 341-355.
81. Yang, M.-S.; Lai, C.-Y.; Lin, C.-Y. A robust EM clustering algorithm for Gaussian mixture models. *Pattern Recognition* **2012**, *45*, 3950-3961.
82. Iorio, C.; Frasso, G.; D'Ambrosio, A.; Siciliano, R. A P-spline based clustering approach for portfolio selection. *Expert Systems with Applications* **2018**, *95*, 88-103.
83. León, D.; Aragón, A.; Sandoval, J.; Hernández, G.; Arévalo, A.; Niño, J. Clustering algorithms for risk-adjusted portfolio construction. *Procedia Computer Science* **2017**, *108*, 1334-1343.
84. Phan, T.C.; Rieger, M.O.; Wang, M. Segmentation of financial clients by attitudes and behavior: A comparison between Switzerland and Vietnam. *International Journal of Bank Marketing* **2019**, *37*, 44-68.
85. Filchenkov, A.; Khanzhina, N.; Tsai, A.; Smetannikov, I. Regularization of autoencoders for bank client profiling based on financial transactions. *Risks* **2021**, *9*, 54.
86. Bolton, R.J.; Hand, D.J. Statistical fraud detection: A review. *Statistical science* **2002**, *17*, 235-255.
87. Subudhi, S.; Panigrahi, S. Use of optimized Fuzzy C-Means clustering and supervised classifiers for automobile insurance fraud detection. *Journal of King Saud University-Computer and Information Sciences* **2020**, *32*, 568-575.
88. Augustyński, I.; Laskoś-Grabowski, P. Clustering macroeconomic time series. *Econometrics. Ekonometria. Advances in Applied Data Analytics* **2018**, *22*, 74-88.
89. Zuhroh, I.; Rofik, M.; Echchabi, A. Banking stock price movement and macroeconomic indicators: k-means clustering approach. *Cogent Business & Management* **2021**, *8*, 1980247.
90. Rapsikevičius, J.; Bruneckienė, J.; Krušinskas, R.; Lukauskas, M. The Impact of Structural Reforms on Sustainable Development Performance: Evidence from European Union Countries. *Sustainability* **2022**, *14*, 12583.
91. Rapsikevicius, J.; Bruneckiene, J.; Lukauskas, M.; Mikalonis, S. The impact of economic freedom on economic and environmental performance: evidence from European countries. *Sustainability* **2021**, *13*, 2380.

92. Long, G.; Xie, M.; Shen, T.; Zhou, T.; Wang, X.; Jiang, J. Multi-center federated learning: clients clustering for better personalization. *World Wide Web* **2023**, *26*, 481-500.
93. Razavi, R. Personality segmentation of users through mining their mobile usage patterns. *International Journal of Human-Computer Studies* **2020**, *143*, 102470.
94. Shensa, A.; Sidani, J.E.; Dew, M.A.; Escobar-Viera, C.G.; Primack, B.A. Social media use and depression and anxiety symptoms: A cluster analysis. *American journal of health behavior* **2018**, *42*, 116-128.
95. Anitha, P.; Patil, M.M. RFM model for customer purchase behavior using K-Means algorithm. *Journal of King Saud University-Computer and Information Sciences* **2022**, *34*, 1785-1792.
96. Adomavicius, G.; Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering* **2005**, *17*, 734-749.
97. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Item-based collaborative filtering recommendation algorithms. In Proceedings of the Proceedings of the 10th international conference on World Wide Web, 2001; pp. 285-295.
98. Rashid, A.M.; Albert, I.; Cosley, D.; Lam, S.K.; McNee, S.M.; Konstan, J.A.; Riedl, J. Getting to know you: learning new user preferences in recommender systems. In Proceedings of the Proceedings of the 7th international conference on Intelligent user interfaces, 2002; pp. 127-134.
99. Xue, G.-R.; Lin, C.; Yang, Q.; Xi, W.; Zeng, H.-J.; Yu, Y.; Chen, Z. Scalable collaborative filtering using cluster-based smoothing. In Proceedings of the Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, 2005; pp. 114-121.
100. Jiang, D.; Tang, C.; Zhang, A. Cluster analysis for gene expression data: a survey. *IEEE Transactions on knowledge and data engineering* **2004**, *16*, 1370-1386.
101. Kriegel, H.-P.; Kröger, P.; Zimek, A. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *Acm transactions on knowledge discovery from data (tkdd)* **2009**, *3*, 1-58.
102. Eisen, M.B.; Spellman, P.T.; Brown, P.O.; Botstein, D. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences* **1998**, *95*, 14863-14868.
103. Alon, U.; Barkai, N.; Notterman, D.A.; Gish, K.; Ybarra, S.; Mack, D.; Levine, A.J. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences* **1999**, *96*, 6745-6750.
104. Kiselev, V.Y.; Kirschner, K.; Schaub, M.T.; Andrews, T.; Yiu, A.; Chandra, T.; Natarajan, K.N.; Reik, W.; Barahona, M.; Green, A.R. SC3: consensus clustering of single-cell RNA-seq data. *Nature methods* **2017**, *14*, 483-486.

105. Stuart, T.; Butler, A.; Hoffman, P.; Hafemeister, C.; Papalexi, E.; Mauck, W.M.; Hao, Y.; Stoeckius, M.; Smibert, P.; Satija, R. Comprehensive integration of single-cell data. *Cell* **2019**, *177*, 1888-1902. e1821.
106. Aggarwal, C.C.; Zhai, C. A survey of text clustering algorithms. *Mining text data* **2012**, 77-128.
107. Schütze, H.; Manning, C.D.; Raghavan, P. *Introduction to information retrieval*; Cambridge University Press Cambridge: 2008; Volume 39.
108. Steinbach, M.; Karypis, G.; Kumar, V. A comparison of document clustering techniques. **2000**.
109. Zhao, Y.; Karypis, G. Evaluation of hierarchical clustering algorithms for document datasets. In Proceedings of the Proceedings of the eleventh international conference on Information and knowledge management, 2002; pp. 515-524.
110. Hotho, A.; Nürnberger, A.; Paaß, G. A brief survey of text mining. Ldv Forum 20 (1): 19–62. **2005**.
111. Blei, D.; Ng, A.; Jordan, M. Latent dirichlet allocation, journal of machine learning research 3 (Jan). **2003**.
112. Griffiths, T.; Steyvers, M. Finding scientific topics. Proceedings of the National Academy of Sciences of the United States of America. *vol* **2004**, *101*, p9.
113. Yang, B.; Fu, X.; Sidiropoulos, N.D.; Hong, M. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In Proceedings of the international conference on machine learning, 2017; pp. 3861-3870.
114. Jain, A.K. Data clustering: 50 years beyond K-means. *Pattern recognition letters* **2010**, *31*, 651-666.
115. Shi, J.; Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence* **2000**, *22*, 888-905.
116. Kanungo, T.; Mount, D.M.; Netanyahu, N.S.; Piatko, C.D.; Silverman, R.; Wu, A.Y. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE transactions on pattern analysis and machine intelligence* **2002**, *24*, 881-892.
117. Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence* **2002**, *24*, 603-619.
118. Ng, A.; Jordan, M.; Weiss, Y. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems* **2001**, *14*.
119. Abonyi, J.; Feil, B. *Cluster analysis for data mining and system identification*; Springer Science & Business Media: 2007.
120. Lazebnik, S.; Schmid, C.; Ponce, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In Proceedings of the 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06), 2006; pp. 2169-2178.
121. Tarter, M.; Kronmal, R. On multivariate density estimates based on orthogonal expansions. *The Annals of Mathematical Statistics* **1970**, 718-722.

122. van deLaan, M. Efficient and inefficient estimation in semiparametric models. *CWI Tracts* **1995**.
123. van der Laan, M.J.; Dudoit, S.; Keles, S. Asymptotic optimality of likelihood-based cross-validation. *Statistical Applications in Genetics and Molecular Biology* **2004**, *3*.
124. Kooperberg, C.; Stone, C.J. A study of logspline density estimation. *Computational Statistics & Data Analysis* **1991**, *12*, 327-347.
125. Li, D.; Yang, K.; Wong, W.H. Density estimation via discrepancy based adaptive sequential partition. *Advances in neural information processing systems* **2016**, *29*.
126. Rothfuss, J.; Ferreira, F.; Walther, S.; Ulrich, M. Conditional density estimation with neural networks: Best practices and benchmarks. *arXiv preprint arXiv:1903.00954* **2019**.
127. Trentin, E.; Lusnig, L.; Cavalli, F. Parzen neural networks: Fundamentals, properties, and an application to forensic anthropology. *Neural Networks* **2018**, *97*, 137-151.
128. Trentin, E. Soft-constrained neural networks for nonparametric density estimation. *Neural Processing Letters* **2018**, *48*, 915-932.
129. Huynh, H.T.; Nguyen, L. Nonparametric maximum likelihood estimation using neural networks. *Pattern Recognition Letters* **2020**, *138*, 580-586.
130. Liu, Q.; Xu, J.; Jiang, R.; Wong, W.H. Density estimation using deep generative neural networks. *Proceedings of the National Academy of Sciences* **2021**, *118*, e2101344118.
131. Kennedy, E.H.; Balakrishnan, S.; Wasserman, L. Semiparametric counterfactual density estimation. *Biometrika* **2023**, *110*, 875-896.
132. Bilodeau, B.; Foster, D.J.; Roy, D.M. Minimax rates for conditional density estimation via empirical entropy. *The Annals of Statistics* **2023**, *51*, 762-790.
133. Park, S.; Pardalos, P.M. Deep data density estimation through donsker-varadhan representation. *Annals of Mathematics and Artificial Intelligence* **2024**, 1-11.
134. Duda, R.O.; Hart, P.E. *Pattern classification and scene analysis*; Wiley New York: 1973; Volume 3.
135. John, G.H.; Langley, P. Estimating continuous distributions in Bayesian classifiers. *arXiv preprint arXiv:1302.4964* **2013**.
136. Wang, X.-Z.; He, Y.-L.; Wang, D.D. Non-naive Bayesian classifiers for classification problems with continuous attributes. *IEEE Transactions on Cybernetics* **2013**, *44*, 21-39.
137. Azzalini, A.; Menardi, G. Clustering via nonparametric density estimation: The R package pdfCluster. *arXiv preprint arXiv:1301.6559* **2013**.
138. Azzalini, A.; Torelli, N. Clustering via nonparametric density estimation. *Statistics and Computing* **2007**, *17*, 71-80.
139. Cuevas, A.; Febrero, M.; Fraiman, R. Cluster analysis: a further approach based on density estimation. *Computational Statistics & Data Analysis* **2001**, *36*, 441-459.

140. Campello, R.J.; Kröger, P.; Sander, J.; Zimek, A. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2020**, *10*, e1343.
141. Kwak, N.; Choi, C.-H. Input feature selection by mutual information based on Parzen window. *IEEE transactions on pattern analysis and machine intelligence* **2002**, *24*, 1667-1671.
142. Peng, H.; Long, F.; Ding, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence* **2005**, *27*, 1226-1238.
143. Tomas, R. The nonparametric estimation of multivariate distribution density applying clustering procedures. *Vilnius* **2007**, *1*, 12-13.
144. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* **1977**, *39*, 1-22.
145. Everitt, B. *Finite mixture distributions*; Springer Science & Business Media: 2013.
146. Redner, R.A.; Walker, H.F. Mixture densities, maximum likelihood and the EM algorithm. *SIAM review* **1984**, *26*, 195-239.
147. Ahmadinejad, N.; Liu, L. J-Score: A Robust Measure of Clustering Accuracy. *arXiv preprint arXiv:2109.01306* **2021**.
148. Zhong, S.; Ghosh, J. Generative model-based document clustering: a comparative study. *Knowledge and Information Systems* **2005**, *8*, 374-384.
149. Lawrence, H.; Phipps, A. Comparing partitions. *Journal of classification* **1985**, *2*, 193-218.
150. Wang, P.; Shi, H.; Yang, X.; Mi, J. Three-way k-means: integrating k-means and three-way decision. *International Journal of Machine Learning and Cybernetics* **2019**, *10*, 2767-2777.
151. Fowlkes, E.B.; Mallows, C.L. A method for comparing two hierarchical clusterings. *Journal of the American statistical association* **1983**, *78*, 553-569.
152. Caliński, T.; Harabasz, J. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods* **1974**, *3*, 1-27.
153. Davies, D.L.; Bouldin, D.W. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence* **1979**, 224-227.
154. Venna, J.; Kaski, S. Neighborhood Preservation in Nonlinear Projection Methods: An Experimental Study. Berlin, Heidelberg, 2001; pp. 485-491.
155. Cooper, I.; Priestley, R. The world business cycle and expected returns. *Review of Finance* **2013**, *17*, 1029-1064.
156. Baumeister, C.; Hamilton, J.D. Structural interpretation of vector autoregressions with incomplete identification: Revisiting the role of oil supply and demand shocks. *American Economic Review* **2019**, *109*, 1873-1910.

157. Herrera, A.M.; Rangaraju, S.K. The effect of oil supply shocks on US economic activity: What have we learned? *Journal of Applied Econometrics* **2020**, *35*, 141-159.
158. Araci, D. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063* **2019**.
159. Malo, P.; Sinha, A.; Korhonen, P.; Wallenius, J.; Takala, P. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology* **2014**, *65*, 782-796.
160. Huang, A.; Wang, H.; Yang, Y. FinBERT—A Deep Learning Approach to Extracting Textual Information. *Available at SSRN 3910214* **2020**.
161. Rosenthal, S.; Farra, N.; Nakov, P. SemEval-2017 task 4: Sentiment analysis in Twitter. *arXiv preprint arXiv:1912.00741* **2019**.
162. Zhang, T.; Ramakrishnan, R.; Livny, M. BIRCH: an efficient data clustering method for very large databases. *ACM sigmod record* **1996**, *25*, 103-114.
163. Ikotun, A.M.; Ezugwu, A.E.; Abualigah, L.; Abuhaija, B.; Heming, J. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences* **2023**, *622*, 178-210.
164. Ghazal, T.M. Performances of k-means clustering algorithm with different distance metrics. *Intelligent Automation & Soft Computing* **2021**, *30*, 735-742.
165. Zhang, Y.; Li, M.; Wang, S.; Dai, S.; Luo, L.; Zhu, E.; Xu, H.; Zhu, X.; Yao, C.; Zhou, H. Gaussian mixture model clustering with incomplete data. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* **2021**, *17*, 1-14.
166. Anand, S.K.; Kumar, S. Experimental comparisons of clustering approaches for data representation. *ACM Computing Surveys (CSUR)* **2022**, *55*, 1-33.
167. Rousseeuw, P.J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* **1987**, *20*, 53-65.

CURRICULUM VITAE

Mantas Lukauskas
mantas.lukauskas@ktu.lt

Išsilavinimas:
2001–2013 Marijampolės Sūduvos gimnazija
2013–2017 Kauno technologijos universitetas, Matematikos ir gamtos mokslų fakultetas, Medžiagų ir nanotechnologijų bakalauro laipsnis
2014–2017 Kauno technologijos universitetas, Ekonomikos ir verslo fakultetas, ekonomikos bakalauro laipsnis
2017–2019 Kauno technologijos universitetas, Matematikos ir gamtos mokslų fakultetas, Matematikos magistro laipsnis (didžiųjų verslo duomenų analitika).
2019–dabar Kauno technologijos universitetas, Matematikos ir gamtos mokslų fakultetas, Informatikos doktorantūra

Profesinė patirtis:
2017–2017 Apmokamas praktikantas UAB „Littelfuse LT“
2019–2020 Jaunesnysis mokslo darbuotojas. Projektas „Patterns of smart economic development in Europe: which approach is more successful nowadays?“. Kauno technologijos universitetas.
2020–2021 Projekto technikas. Kauno technologijos universitetas
2017–2021 Produkto valdymo analitikas / jaunesnysis produktų vadovas. UAB „Littelfuse LT“.
2021–2023 Jaunesnysis mokslo darbuotojas. Projektas „Ekonomikos aktyvumo vertinimas ir prognozavimas realiuoju laiku, naudojant didžiuosius duomenis“. Kauno technologijos universitetas.
2021 – dabar Duomenų mokslininkas / Techninis vadovas. UAB „Hostinger“

Mokslinių projektų patirtis:
2019-04-01 iki 2019-12-31 Sumanaus ekonominio vystymosi modeliai Europoje: kurie labiau sėkmingi šiandien? (SMART_PROG). Projekto Nr.: PP91R/19, projekto vertė 10 000 Eur.
2019-06-21 iki 2021-12-31 Studijų programų optimizavimas ir fizinių bei technologinių mokslų dėstytojų pedagoginių kompetencijų stiprinimas. SV3/1095.
2021-11-03 iki 2023-08-31 Ekonomikos aktyvumo vertinimas ir prognozavimas realiuoju laiku, naudojant didžiuosius duomenis. Projekto Nr.: 13.1.1-LMT-K-718-05-0012
2023-04-01 iki 2026-03-31 Dirbtiniu intelektu grįstas įmonių inovatyvumo vertinimas skaitmeninės transformacijos kontekste. Projekto Nr.: S-MIP-23-54

Mokslinių interesų sritys:

Dirbtinis intelektas, mašininis mokymasis, duomenų klasterizavimas, dirbtiniu intelektu grįstų produktų kūrimas

STRAIPSNIAI RECENZUOJAMUOSE MOKSLO LEIDINIUOSE

Web of Science duomenų bazėje indeksuotuose žurnaluose su cituojamumo rodikliu (JCR SCIE), kai IF/AIF > 0,25

Užsienio šalių leidyklose

1. Stundziene, A., Pilinkiene, V., Bruneckiene, J., Grybauskas, A., Lukauskas, M., & Pekarskiene, I. (2023). Future directions in nowcasting economic activity: A systematic literature review. Journal of Economic Surveys, 00, 1–35. https://doi.org/10.1111/joes.12579

2. [S1; CH; OA] Lukauskas, Mantas; Šarkauskaitė, Viktorija; Pilinkienė, Vaida; Stundžienė, Alina; Grybauskas, Andrius; Bruneckienė, Jurgita. Enhancing skills demand understanding through job ad segmentation using NLP and clustering techniques // Applied sciences. Basel : MDPI. ISSN 2076-3417. 2023, vol. 13, iss. 10, art. no. 6119, p. 1-29. DOI: 10.3390/app13106119. [Science Citation Index Expanded (Web of Science); Scopus] [IF: 2,838; AIF: 5,795; IF/AIF: 0,489; Q2 (2021, InCites JCR SCIE)] [M.kr.: S 004, N 009] [Indėlis: 0,170] SCIE)] [M.kr.: N 001, S 004]

3 [S1; CH; OA] Ruzgas, Tomas; Kižauskienė, Laura; Lukauskas, Mantas; Sinkevičius, Egidijus; Frolovaitė, Melita; Arnastauskaitė, Jurgita. Tax fraud reduction using analytics in an East European country // Axioms. Basel : MDPI. ISSN 2075-1680. 2023, vol. 12, iss. 3, art. no. 288, p. 1-29. DOI: 10.3390/axioms12030288. [Science Citation Index Expanded (Web of Science); Scopus; DOAJ] [IF: 1,824; AIF: 2,042; IF/AIF: 0,893; Q2 (2021, InCites JCR SCIE)] [M.kr.: N 001, N 009] [Indėlis: 0,240]

4 [S1; CH; OA] Lukauskas, Mantas; Ruzgas, Tomas. Reduced clustering method based on the inversion formula density estimation // Mathematics. Basel : MDPI. ISSN 2227-7390. 2023, vol. 11, iss. 3, art. no. 661, p. 1-15. DOI: 10.3390/math11030661. [Science Citation Index Expanded (Web of Science); Scopus; DOAJ] [IF: 2,592; AIF: 1,291; IF/AIF: 2,007; Q1 (2021, InCites JCR SCIE)] [M.kr.: N 001, N 009] [Indėlis: 0,500]

5 [S1; CH; OA] Rapsikevičius, Jonas; Bruneckienė, Jurgita; Krušinskas, Rytis; Lukauskas, Mantas. The impact of structural reforms on sustainable development performance: evidence from European Union countries // Sustainability. Basel : MDPI. ISSN 2071-1050. 2022, vol. 14, iss. 19, art. no. 12583, p. 1-18. DOI: 10.3390/su141912583. [Social Sciences Citation Index (Web of Science); Scopus; DOAJ] [IF: 3,889; AIF: 6,732; IF/AIF: 0,577; Q2 (2021, InCites JCR SCIE)] [M.kr.: S 004, N 009] [Indėlis: 0,250]

6 [S1; CH; OA] Lukauskas, Mantas; Pilinkienė, Vaida; Bruneckienė, Jurgita; Stundžienė, Alina; Grybauskas, Andrius; Ruzgas, Tomas. Economic activity forecasting based on the sentiment analysis of news // Mathematics. Basel : MDPI. ISSN 2227-7390. 2022, vol. 10, iss. 19, art. no. 3461, p. 1-22. DOI: 10.3390/math10193461. [Science Citation Index Expanded (Web of Science); Scopus; Dimensions] [IF: 2,592; AIF: 1,291; IF/AIF: 2,007; Q1 (2021, InCites JCR SCIE)] [M.kr.: S 004, N 001, N 009] [Indėlis: 0,170]

7 [S1; CH; OA] Lukauskas, Mantas; Ruzgas, Tomas. A new clustering method based on the inversion formula // Mathematics. Basel : MDPI. ISSN 2227-7390. 2022, vol. 10, iss. 15, art. no. 2559, p. 1-16. DOI: 10.3390/math10152559. [Science Citation Index Expanded (Web of Science); Scopus] [IF: 2,592; AIF: 1,291; IF/AIF: 2,007; Q1 (2021, InCites JCR SCIE)] [M.kr.: N 001, N 009] [Indėlis: 0,500]

8 [S1; CH; OA] Ruzgas, Tomas; Lukauskas, Mantas; Čepkauskas, Gedmantas. Nonparametric multivariate density estimation: case study of Cauchy mixture model // Mathematics. Basel : MDPI. ISSN 2227-7390. 2021, vol. 9, iss. 21, art. no. 2717, p. 1-23. DOI: 10.3390/math9212717. [Science Citation Index Expanded (Web of Science); Scopus; DOAJ] [IF: 2,592; AIF: 1,291; IF/AIF: 2,007; Q1 (2021, InCites JCR SCIE)] [M.kr.: N 001, N 009] [Indėlis: 0,333]

9 [S1; CH; OA] Rapsikevicius, Jonas; Bruneckiene, Jurgita; Lukauskas, Mantas; Mikalonis, Sarunas. The impact of economic freedom on economic and environmental performance: evidence from European countries // Sustainability. Basel : MDPI. ISSN 2071-1050. 2021, vol. 13, iss. 4, art. no. 2380, p. 1-22. DOI: 10.3390/su13042380. [Social Sciences Citation Index (Web of Science); Scopus; DOAJ] [IF: 3,889; AIF: 6,732; IF/AIF: 0,577; Q2 (2021, InCites JCR SCIE)] [M.kr.: S 004, N 009] [Indėlis: 0,250]

10 [S1; CH; OA] Bruneckienė, Jurgita; Jucevicius, Robertas; Zykienė, Ineta; Rapsikevičius, Jonas; Lukauskas, Mantas. Assessment of investment attractiveness in European countries by artificial neural networks: what competences are needed to make a decision on collective well-being? // Sustainability. Basel : MDPI. ISSN 2071-1050. 2019, vol. 11, iss. 24, art. no.6892, p. 1-23. DOI: 10.3390/su11246892. [Science Citation Index Expanded (Web of Science); Social Sciences Citation Index (Web of Science); Scopus] [IF: 2,576; AIF: 5,046; IF/AIF: 0,510; Q2 (2019, InCites JCR SCIE)] [M.kr.: S 003, N 009] [Indėlis: 0,200]

Web of Science duomenų bazėje indeksuotuose Q4 kvartilio žurnaluose (JCR

Užsienio šalių leidyklose

1. [S1; SK; OA] Lukauskas, Mantas; Rasymas, Tomas; Minelga, Matas; Vaitmonas, Domas. Large scale fine-tuned transformers models application for business names generation // Computing and informatics. Bratislava : Institute of Informatics Slovak Academy of Sciences. ISSN 1335-9150. eISSN 2585-8807. 2023, vol. 42, no. 3, p. 525-545. DOI: 10.31577/cai_2023_3_525. [Science Citation Index Expanded (Web of

Web of Science duomenų bazėje indeksuotuose leidiniuose be cituojamumo rodiklio (JCR SCIE)

Užsienio šalių leidyklose

1 [S1; CH; OA] Stundziene, Alina; Pilinkiene, Vaida; Bruneckiene, Jurgita; Grybauskas, Andrius; Lukauskas, Mantas. Nowcasting economic activity using electricity market data: the case of Lithuania // Economies. Basel : MDPI. ISSN 2227-7099. 2023, vol. 11, iss. 5, art. no. 134, p. 1-21. DOI: 10.3390/economies11050134. [Emerging Sources Citation Index (Web of Science); Scopus; DOAJ] [M.kr.: S 004] [Indėlis: 0,200]

2 [S1; CH; OA] Grybauskas, Andrius; Pilinkienė, Vaida; Lukauskas, Mantas; Stundžienė, Alina; Bruneckienė, Jurgita. Nowcasting unemployment using neural networks and multi-dimensional Google trends data // Economies. Basel : MDPI. ISSN 2227-7099. 2023, vol. 11, iss. 5, art. no. 130, p. 1-23. DOI: 10.3390/economies11050130. [Emerging Sources Citation Index (Web of Science); Scopus; DOAJ] [M.kr.: S 004]

3 [S1; GB] Bruneckienė, Jurgita; Rapsikevicius, Jonas; Lukauskas, Mantas; Zykienė, Ineta; Jucevicius, Robertas. Smart economic development patterns in Europe: interaction with competitiveness // Competitiveness review. Bingley : Emerald. ISSN 1059-5422. eISSN 2051-3143. 2023, vol. 33, iss. 2, p. 302-331. DOI: 10.1108/CR-02-2021-0026. [Emerging Sources Citation Index (Web of Science); Scopus] [M.kr.: S 004, N 009] [Indėlis: 0,200]

4 [S1; CH] Dagilienė, Lina; Bruneckienė, Jurgita; Varaniūtė, Viktorija; Lukauskas, Mantas. The circular economy for sustainable development: implementation strategies in advanced small open economies // International journal of environment and sustainable development. Geneva : Inderscience. ISSN 1474-6778. eISSN 1478-7466. 2023, vol. 22, iss. 1, p. 51-76. DOI: 10.1504/IJESD.2021.10040657. [Emerging Sources Citation Index (Web of Science); Scopus; Academic OneFile] [M.kr.: S 004, N 009]

5 [S1; GB] Dagilienė, Lina; Bruneckienė, Jurgita; Jucevičius, Robertas; Lukauskas, Mantas. Exploring smart economic development and competitiveness in Central and Eastern European countries // Competitiveness review. Bingley : Emerald. ISSN 1059-5422. eISSN 2051-3143. 2020, vol. 30, iss. 5, p. 485-505. DOI: 10.1108/CR-04-2019-0041. [Emerging Sources Citation Index (Web of Science); Scopus] [M.kr.: S 004, N 009] [Indėlis: 0,250]

Kitose tarptautinėse duomenų bazėse referuojamuose recenzuojamuose mokslo leidiniuose

Užsienio šalių leidyklose

1 [S2; CH; OA] Lukauskas, Mantas; Pilinkienė, Vaida; Bruneckienė, Jurgita; Stundžienė, Alina; Grybauskas, Andrius; Ruzgas, Tomas. Evaluation of news sentiment in economic activity forecasting // Engineering proceedings. Basel : MDPI. ISSN 2673-4591. 2023, vol. 31, iss. 1, art. no. 7, p. 1-6. DOI: 10.3390/ASEC2022-13790. [Scopus; Dimensions] [M.kr.: S 004, N 001, N 009] [Indėlis: 0,170]

2 [S3; TR; OA] Lukauskas, Mantas; Ruzgas, Tomas. Data clustering and its applications in medicine // New trends in mathematical science: ISAME 2022 proceedings. Istanbul : BİSKA bilisim technology. ISSN 2147-5520. 2022, vol. 10, spec. iss. 1, p. 67-70. DOI: 10.20852/ntmsci.2022.465. [Academic Search Ultimate; Scilit; DOAJ] [M.kr.: N 001, N 009] [Indėlis: 0,500]

Kituose recenzuojamuose mokslo leidiniuose

Užsienio šalių leidyklose

1 [P1d; SG] Lukauskas, Mantas; Ruzgas, Tomas. Review and comparative analysis of unsupervised machine learning application in health care // Data intelligence and cognitive informatics: proceedings of ICDICI 2022, [Tirunelveli, India, July 6–7, 2022] / I.J. Jacob, S.K. Shanmugam, I. Izonin (eds.). Singapore : Springer Nature, 2023. ISBN 9789811960031. eISBN 9789811960048. p. 751-759. (Algorithms for intelligent systems, ISSN 2524-7565, eISSN 2524-7573). DOI: 10.1007/978-981-19-6004-8_56. [M.kr.: N 001, N 009] [Indėlis: 0,500]

2 [P1d; CH] Lukauskas, Mantas; Pilinkienė, Vaida; Bruneckienė, Jurgita; Stundžienė, Alina; Grybauskas, Andrius. Automated system and machine learning application in economic activity monitoring and nowcasting // Information and software technologies: 28th international conference, ICIST 2022, Kaunas, Lithuania, October 13–15, 2022: proceedings / A. Lopata, D. Gudonienė, R. Butkienė (eds.). Cham : Springer, 2022. ISBN 9783031163012. eISBN 9783031163029. p. 102-113. (Communications in computer and information science, ISSN 1865-0929, eISSN 1865-0937 ; vol. 1665). DOI: 10.1007/978-3-031-16302-9_8. [M.kr.: S 004, N 001, N 009] [Indėlis: 0,200]

3 [S4; US] Bruneckiene, Jurgita; Jucevicius, Robertas; Zykiene, Ineta; Rapsikevicius, Jonas; Lukauskas, Mantas. Quantum theory and artificial intelligence in the analysis of the development of socio-economic systems: theoretical insights // Developing countries and technology inclusion in the 21st century information society / editor A. S. Etim. Hershey, PA : IGI Global, 2021. ISBN 9781799834687. eISBN 9781799834700. p. 1-16. DOI: 10.4018/978-1-7998-3468-7. [M.kr.: S 003, N 009] [Indėlis: 0,200]

MOKSLINIŲ TYRIMŲ REZULTATŲ SKELBIMAS KONFERENCIJOSE

Konferencijų pranešimų tezės tarptautinių duomenų bazių leidiniuose

1 [T1; LT; OA] Lukauskas, Mantas; Pilinkienė, Vaida; Bruneckienė, Jurgita; Stundžienė, Alina; Grybauskas, Andrius; Ruzgas, Tomas. Big data processing system for Lithuania economic activity nowcasting // DAMSS 2022: 13th conference on data analysis methods for software systems, Druskininkai, Lithuania, December 1–3, 2022 / Lithuanian computer society, Vilnius university Institute of data science and digital technologies, Lithuanian academy of sciences. Vilnius : Vilnius university press, 2022. ISBN 9786090707944. eISBN 9786090707951. p. 55. (Vilnius university proceedings, eISSN 2669-0233). [Dimensions] [M.kr.: S 004, N 001, N 009]

2 [T1; LT; OA] Lukauskas, Mantas; Ruzgas, Tomas. Data clustering based on the modified inversion formula density estimation // DAMSS 2022: 13th conference on data analysis methods for software systems, Druskininkai, Lithuania, December 1–3, 2022 / Lithuanian computer society, Vilnius university Institute of data science and digital technologies, Lithuanian academy of sciences. Vilnius : Vilnius university press, 2022. ISBN 9786090707944. eISBN 9786090707951. p. 56-57. (Vilnius university proceedings, eISSN 2669-0233). [Dimensions] [M.kr.: N 001, N 009]

3 [T1; LT; OA] Lukauskas, Mantas; Ruzgas, Tomas. Analysis of clustering methods performance across multiple datasets // DAMSS 2021: 12th conference on data analysis methods for software systems, Druskininkai, Lithuania, December 2–4, 2021 / Lithuanian computer society, Vilnius university Institute of data science and digital technologies, Lithuanian academy of sciences. Vilnius : Vilnius university press, 2021. ISBN 9786090706732. eISBN 9786090706749. p. 45-46. (Vilnius university proceedings, eISSN 2669-0233). [Dimensions; Scilit] [M.kr.: N 001, N 009]

4 [T1; LT; OA] Lukauskas, Mantas; Rasymas, Tomas; Vaitmonas, Domas; Minelga, Matas. Natural language generation with architecture of transformers: a case study of business names generation // DAMSS 2021: 12th conference on data analysis methods for software systems, Druskininkai, Lithuania, December 2–4, 2021 / Lithuanian computer society, Vilnius university Institute of data science and digital technologies, Lithuanian academy of sciences. Vilnius: Vilnius university press, 2021. ISBN 9786090706732. eISBN 9786090706749. p. 43-44. (Vilnius university proceedings, eISSN 2669-0233). [Dimensions; Scilit] [M.kr.: N 009]

Kitos konferencijų pranešimų tezės ir straipsniai nerecenzuojamoje konferencijų pranešimų medžiagoje

1 [P2; TR; OA] Lukauskas, Mantas; Ruzgas, Tomas. Mixtures models for clustering: review and comparison // 10th (online) international conference on applied analysis and mathematical modeling (ICAAMM22), July 1-3, 2022, Istanbul, Turkey: abstracts and proceedings book / Mustafa Bayram, Aydın Seçer (eds.). [S.l.] : [s.n.], 2022. ISBN 9786056918162. p. 106-109. [M.kr.: N 001, N 009]

2 [T2; TR; OA] Lukauskas, Mantas; Ruzgas, Tomas. Data clustering and its applications in medicine // Online international symposium on applied mathematics and engineering (ISAME22), January 21-23, 2022, Istanbul-Turkey: abstracts book / M. Bayram, A. Secer (eds.). Istanbul : [s.n.], 2022. ISBN 9786056918155. p. 60-61. [M.kr.: N 001, N 009]

3 [T3; CH; OA] Lukauskas, Mantas; Rasymas, Tomas; Vaitmonas, Domas; Minelga, Matas. Transformers architecture application in high-quality business names generation // IECI 2021: the 1st international electronic conference on information, 1–15 December 2021, online. Basel : Sciforum. 2021, p. 1. DOI: 10.3390/IECI2021-11960. [M.kr.: N 009, N 001]

4 [T2; LT; OA] Bruneckiene, Jurgita; Varaniute, Viktorija; Dagiliene, Lina; Lukauskas, Mantas. What profiles of circular economy implementation strategies dominate in advanced small open economies? // 2021 IEEE international conference on technology and entrepreneurship (ICTE) "Leading digital transformation in business and society": book of abstracts / edited by A. Gadeikiene, A. Pundziene, J. Banyte. Kaunas : Technologija. ISSN 2783-6037. 2021, p. 36. [M.kr.: S 004, N 009]

5 [T3; CH] Lukauskas, Mantas; Ruzgas, Tomas. Bank credit card default classification based on clustering using machine learning algorithms // 9th world sustainability forum, virtual, Switzerland, 13–15 September 2021: program and abstract book / organised by MDPI. Basel : MDPI. 2021, p. 19. [M.kr.: N 001, N 009]

6 [T2; TR] Lukauskas, Mantas; Ruzgas, Tomas. A review of clustering algorithms and application // 9th (online) international conference on applied analysis and mathematical modeling (ICAAMM21) June 11-13, 2021, Istanbul-Turkey: abstracts book /M. Bayram, A. Seçer (eds.). Istanbul : Biruni University, 2021. ISBN 9786056918148. p. 45. [M.kr.: N 001, N 009]

7 [T2; TR] Rapsikevičius, Jonas; Bruneckiene, Jurgita; Lukauskas, Mantas; Mikalonis, Šarūnas. The impact of economic freedom on economic performance: evidence from high-income countries // 34th EBES conference, 6-8, January 2021, Athens, Greece (online/virtual presentation only): program and abstract book. Istanbul : EBES Publications, 2021. ISBN 9786058004245. p. 69. [M.kr.: S 004, N 009]

8 [T2; LT] Bruneckienė, Jurgita; Jucevicius, Robertas; Lukauskas, Mantas; Rapsikevicius, Jonas; Zykienė, Ineta. Quantum theory and artificial intelligence in economic development patterns: theoretical insights // AIB-CEE 2019 : 6th Academy of International Business Central Eastern European (AIB-CEE) chapter annual conference: „International business in the dynamic environment: changes in digitalization, innovation and entrepreneurship", 25-27 September, 2019, Kaunas, Lithuania : book of abstracts / Edited by Jurgita Sekliuckienė, Rozita Susnienė. Kaunas : Kaunas University of Technology, 2019. eISBN 9786090216378. p. 102. [M.kr.: S 003, N 009]

9 [P2; LT] Lukauskas, M.; Bruneckienė, J. Mašininio mokymosi panaudojimo galimybės regionų investicinio patrauklumo vertinime // Technologijų ir verslo aktualijos – 2018: studentų mokslinių darbų konferencijos pranešimų medžiaga, 2018 m. gegužės 4 d., Panevėžys, Lietuva. Kaunas : Kauno technologijos universitetas. ISSN 2538-8045. 2018, p. 184-191. [M.kr.: S 004]

10 [P2; LT] Lukauskas, M.; Abakevičienė, B. Plonasluoksnių polimerinių plėvelių mechaninių savybių tyrimas // Technologijų ir verslo aktualijos – 2018: studentų mokslinių darbų konferencijos pranešimų medžiaga, 2018 m. gegužės 4 d., Panevėžys, Lietuva. Kaunas : Kauno technologijos universitetas. ISSN 2538-8045. 2018, p. 330-337. [M.kr.: T 008]

11 [T2; LT] Lukauskas, M.; Abakevičienė, B. (temos vadovas). Išorinių poveikių įtaka fluorinto etileno-tetrafluoretileno mechaninėms savybėms // Matematika ir gamtos mokslai: teorija ir taikymai : XVI studentų konferencijos pranešimų medžiaga = Mathematics and natural sciences: theory and applications. Kaunas : Kauno technologijos universitetas, 2018. ISBN 9786090214534. p. 41-42. [M.kr.: T 008]

12 [T3; LT] Lukauskas, Mantas; Abakevičienė, Brigita. Fluoropolimerų mechaninių savybių tyrimas mikrotempimo įrenginiu // Fizinių ir technologijos mokslų tarpdalykiniai tyrimai : 8-oji jaunųjų mokslininkų konferencija, 2018 m. vasario 8 d. : pranešimų santraukos / Lietuvos mokslų akademija. [S.l.] : [s.n.]. 2018, p. 19. [M.kr.: T 008, N 002]

MOKSLO, MENO POPULIARINIMO LEIDINIAI IR JŲ DALYS, MULTIMEDIJA

1. [S9; LT] Serackis, Artūras (interviu, duodantis); Žemaitis, Eigirdas (interviu, duodantis); Eriksonas, Linas (interviu, duodantis); Kapočiūtė-Dzikienė, Jurgita (interviu, duodantis); Lukauskas, Mantas (interviu, duodantis); Piepaliūtė, Agnė (interviu, imantis). (Ne) pakeičiamieji // IQ: DI ir darbo rinka : UAB „Naujienų centras". ISSN 2029-4417. 2024,

MOKSLINIŲ STRAIPSNIŲ KOPIJOS

Tolesniuose puslapiuose yra penkių straipsnių, sudarančių šios disertacijos pagrindą, kopijos.

Straipsnis 1. T. Ruzgas, **M. Lukauskas**, and G. Čepkauskas. 2021. “Nonparametric Multivariate Density Estimation: Case Study of Cauchy Mixture Model" Mathematics 9, no. 21: 2717. https://doi.org/10.3390/math9212717

Straipsnis 2. **M. Lukauskas** ir T. Ruzgas. 2022. “A New Clustering Method Based on the Inversion Formula” Mathematics 10, no. 15: 2559. https://doi.org/10.3390/math10152559.

Straipsnis 3. **M. Lukauskas** ir T. Ruzgas. 2023. “Reduced Clustering Method Based on the Inversion Formula Density Estimation” Mathematics 11, no. 3: 661. https://doi.org/10.3390/math11030661

Straipsnis 4. **M. Lukauskas**, V. Pilinkienė, J. Bruneckienė, A. Stundžienė, A. Grybauskas ir T. Ruzgas. 2022. “Economic Activity Forecasting Based on the Sentiment Analysis of News” Mathematics 10, no. 19: 3461. https://doi.org/10.3390/math10193461

Straipsnis 5. **M. Lukauskas**, V. Šarkauskaitė, V. Pilinkienė, A. Stundžienė, A. Grybauskas ir J. Bruneckienė. “Enhancing Skills Demand Understanding through Job Ad Segmentation using NLP and Clustering Techniques” Applied Sciences. 2023; 13(10):6119. https://doi.org/10.3390/app13106119

mathematics

MDPI

Article

Nonparametric Multivariate Density Estimation: Case Study of Cauchy Mixture Model

Tomas Ruzgas *, Mantas Lukauskas * and Gedmantas Čepkauskas

Department of Applied Mathematics, Faculty of Mathematics and Natural Sciences, Kaunas University of Technology, 44249 Kaunas, Lithuania
* Correspondence: tomas.ruzgas@ktu.lt (T.R.); mantas.lukauskas@ktu.lt (M.L.)

Abstract: Estimation of probability density functions (pdf) is considered an essential part of statistical modelling. Heteroskedasticity and outliers are the problems that make data analysis harder. The Cauchy mixture model helps us to cover both of them. This paper studies five different significant types of non-parametric multivariate density estimation techniques algorithmically and empirically. At the same time, we do not make assumptions about the origin of data from any known parametric families of distribution. The method of the inversion formula is made when the cluster of noise is involved in the general mixture model. The effectiveness of the method is demonstrated through a simulation study. The relationship between the accuracy of evaluation and complicated multidimensional Cauchy mixture models (CMM) is analyzed using the Monte Carlo method. For larger dimensions ($d \sim 5$) and small samples ($n \sim 50$), the adaptive kernel method is recommended. If the sample is $n \sim 100$, it is recommended to use a modified inversion formula (MIDE). It is better for larger samples with overlapping distributions to use a semi-parametric kernel estimation and more isolated distribution-modified inversion methods. For the mean absolute percentage error, it is recommended to use a semi-parametric kernel estimation when the sample has overlapping distributions. In the smaller dimensions ($d = 2$) and a sample is with overlapping distributions, it is recommended to use the semi-parametric kernel method (SKDE) and for isolated distributions, it is recommended to use modified inversion formula (MIDE). The inversion formula algorithm shows that with noise cluster, the results of the inversion formula improved significantly.

Keywords: Cauchy mixture model; nonparametric density estimation; density estimation algorithms; adapted kernel density estimate; logspline estimation

MSC: 62G05; 62G07; 62G30

Citation: Ruzgas, T.; Lukauskas, M.; Čepkauskas, G. Nonparametric Multivariate Density Estimation: Case Study of Cauchy Mixture Model. *Mathematics* **2021**, *9*, 2717. https://doi.org/10.3390/math9212717

Academic Editor: Antonio Di Crescenzo

Received: 1 September 2021
Accepted: 21 October 2021
Published: 26 October 2021

1. Introduction

Estimation of probability density functions (pdf) is considered an essential part of statistical modelling. It expresses random variables as functions of other variables, making it possible to detect hidden relationships between data [1]. In a significant number of machine learning algorithms, it is essential to determine a previously unknown function of the distribution density of the data. The function of the distribution density is applied in the Bayesian classifier [2,3], in density-based clustering algorithms [4–6], or information-based feature selection algorithms [7,8]. Effective density estimates must be carefully created in advance to obtain unknown functions of probability density. Nowadays, there is still much focus on developing innovative density estimation procedures [9,10]. Density estimation is an open research topic in the fast-growing area of deep learning. Scientists have begun proposing robust density estimators based on neural networks such as Parzen neural networks [11], soft-constrained neural networks [12], and others [13].

Let us say that the random vector $X \in R^d$ satisfies the distribution mixture model if its distribution density $f(x)$ satisfies the equation $f(x) = \sum_{k=1}^{q} p_k f_k(x) = f(x, \theta)$. The

parameter q is the number of mixture clusters, and p_k is the a priori probability. These conditions must also be met: $p_k > 0$ and $\sum_{k=1}^{q} p_k = 1$. The function $f_k(x)$ is a function of the distribution density, and θ is a multidimensional parameter of the model. Suppose X is a d-dimensional random vector with a distribution density $f(x)$, and there is a sample of independent copies of X, where $X = (X(1), \ldots, X(n))$. It can be argued that the sample satisfies the mixture model if $X(t)$ satisfies $f(x) = \sum_{k=1}^{q} p_k f_k(x) = f(x, \theta)$.

One of the statistical tasks is to estimate the density of the observed random variable. Suppose the available sample's distribution type is known (Normal, Poisson, and others). In that case, the distribution density of the data can be estimated simply using mean and covariance matrix estimates, fitting them to a defined distribution [14–16]. Thus, the standard parametric method is applied when the assumptions about the density form are met. When estimating density in a parametric way, the value of the multidimensional distribution parameter θ needs to be found, which is not straightforward because the number of parameters increases rapidly as the dimension d increases. For example, in the case of a mixture of Gaussian distributions, $\dim\theta = \frac{1}{2qd(d+1)} + qd + q - 1$, and even with a small dimension $d = q = 5$, the model will consist of $\dim(\theta) = 104$ parameters. When searching for parameter estimates, it may be necessary to solve the optimization problem in the 104-dimensional space. In practice, the number of clusters q may also be unknown, and it needs to be estimated. The parametric method is not proper when the random size distribution is unknown. In this case, non-parametric methods are used to determine certain forms of density estimates [17–19].

The histogram is one of the simplest and oldest estimates of density. To the best of our knowledge, data in the form of histograms (without graphical representation) were first presented in 1661 to determine mortality probabilities in different age groups [20]. To approximate the density $f(x)$ in the area Ω, the number of observations $X(t)$ falling into Ω is calculated and divided by n and the volume of the area Ω. The area of space to which all observations fall is first found. That means the fluctuation intervals of all X projections on the axes $X^{(1)}, X^{(2)}, \ldots, X^{(d)}$ are found. The fluctuation intervals of the observations are divided into l partial intervals and in the hypercubes $\Omega_j (j = 1, \ldots, r)$ bounded by them, the density estimate is calculated as

$$\hat{f}(x) = \frac{n(\Omega_j)}{n \cdot h_1 \cdot h_2 \ldots \cdot h_d}. \quad (1)$$

Here $n(\Omega_j)$ is the number of observations entering the hypercube Ω_j and $h_j, j = 1, \ldots, d$ are the edges of the hypercube [21,22]. It is recommended to select the number of hypercubes [17,23,24], and to choose $r \cong 1 + 3.32 \log(n)$, and $l = \sqrt[d]{r}$ has to be an integer number, so r is chosen that $\left\lceil \sqrt[d]{1 + 3.32 log n}^{\,d} \right\rceil$.

A histogram is one of the simplest means of presenting data that is easy to understand and convenient. This estimate is a function that acquires non-negative values, and its integral is equal to one. However, it is not continuous. That poses problems when knowing the density estimate derivatives is essential, mainly when density estimation is used in intermediate steps of other methods, such as clustering using a gradient algorithm or plotting high-measurement data-level lines. Remarkably, the histogram stood as the only non-parametric density estimator until the 1950's when substantial and simultaneous progress was made in density estimation and spectral density estimation. In 1951, in a little-known paper, Fix and Hodges [25] introduced the basic algorithm of non-parametric density estimation; an unpublished technical report was formally published as a review by Silverman and Jones in 1989 [26]. They addressed the problem of statistical discrimination when the parametric form of the sampling density was not known. During the following decade, several general algorithms and alternative theoretical modes of analysis were introduced by Rosenblatt in 1956 [27], Parzen in 1962 [28], and Cencov in 1962 [29]. Then followed the second wave of essential and primarily theoretical papers by Watson and Leadbetter in 1963 [30], Loftsgaarden and Quesenberry in 1965 [31], Schwartz in 1967 [32],

Epanechnikov in 1969 [33], Tarter and Kronmal in 1970 [34], and Kimeldorf and Wahba in 1971 [35]. Next, Cacoullos introduced the natural multivariate generalization in 1966 [36]. Finally, in the 1970s, the first papers focusing on the practical application of these methods were published by Scott et al. in 1979 [24] and Silverman in 1978 [37]. These and later multivariate applications awaited the computing revolution.

Modern data analysis uses several non-parametric methods for statistically estimating the distribution density of multivariate random variables. Kernel estimates are particularly prevalent [38,39]. Quite popular and spline [40,41] and semi-parametric [42–46] algorithms. However, detailed comparisons of the effectiveness of existing popular estimates for multimodal density are lacking. With the most popular non-parametric estimation procedures, optimal selection of their parameters is encountered in practice. The most crucial element in the design of kernel estimates is the width of the smoothing. It is not easy to select the nodes of the spline estimates. Although several adaptive procedures for the selection of these parameters have been developed [39,47–52], however, they are not efficient enough when the sample volume is not large, especially then the observational dimension is large. In the latter case, it is appropriate to apply data design [53–56] because of the more extensive the dimension of the observed random vectors, the more complex the task of parameter selection.

The main idea of this paper is to estimate the performance of different density estimators by using density mixtures to show another type of problem, which may result from data heteroscedasticity and outliers. The relationship between the accuracy of evaluation and complicated multidimensional Cauchy mixture models (CMM) is analyzed using the Monte Carlo method. For example, Kalantan and Einbeck [57] used engineering data and, for computer vision, used CMM, comparing it with the Gaussian mixture model. Azzari and Foi [58] used harmony between Gauss and heavy-tailed Cauchy to find noise-model parameters that make outlier estimation robust when imaged dominated by texture. Finally, Teimouri [59] analyzed patients with Cushing's syndrome and their diagnostic tests. The focus was on the tetra hydrocortisone urine release rate (mg/24 h) and evaluating parameters in the EM algorithm and Cauchy mixture model.

Scientific novelty. Evaluation accuracy comparative analysis is made by using different probability density estimation procedures. Density function estimates are chosen as popular different technique estimates, which other researchers have already analyzed. This research is essential because it focuses on Cauchy distributions.

2. The Density Estimation Algorithms

This section aims to present the density estimation algorithms used in the study theoretically. All algorithms are presented with algorithms theoretical substantiation. When making the histogram, each $X(t)$ can be imagined as a separate column with a height of $1/n$. Then it makes sense to change the centre of the column to $X(t)$ itself and get the following function:

$$\hat{f}(x) = \frac{1}{n \cdot h_1 \cdot h_2 \ldots \cdot h_d} \sum_{t=1}^{n} I_{C_h(X(t))}(x). \quad (2)$$

Here C_h is a hypercube with centre $X(t)$, and the lengths of the edges are $h_1, \ldots, h_d$. In summary, instead of the indicator function, a smooth "prominence"—the kernel function—can be used at each observed point. The multidimensional fixed-width bandwidth estimate with the kernel function K and the fixed (global) kernel width parameter h, which can be used to estimate the density $\hat{f}(x)$ of the multidimensional data $X \in R^d$, is then defined as follows:

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{t=1}^{n} K\left(\frac{x - X(t)}{h}\right). \quad (3)$$

These are some of the most common non-parametric estimates of distribution density [38,39,60,61]. The kernel function is selected to meet the following condition:

$$\int_{R^d} K(x)dx = 1. \quad (4)$$

The standard normal distribution density function φ is often used as the kernel [62,63]:

$$\Phi(x) = (2\pi)^{-d/2} \exp(-\frac{1}{2}x'x). \quad (5)$$

Often the observations are not evenly distributed in all directions. Therefore, it is desirable to scale the data by eliminating the most significant dispersion differences in the different coordinate directions. One suitable method for this [64] is data standardization. That means the sample's effect on a linear transformation. The mean of the transformed data is zero, the covariance matrix is unitary, and (3) apply the Equation to already standardized data. For example, suppose Z is a standardized random vector,

$$Z = S^{-\frac{1}{2}}(X - \overline{X}), \quad (6)$$

here $\overline{X}$ is the empirical mean of the sample, and $S \in R^{d \times d}$ is the empirical covariance matrix. Based on the fixed kernel width density estimate (3), a more complex standardized data density estimate has been constructed:

$$\hat{f}_z(z) = \frac{1}{nh^d} \sum_{t=1}^{n} K\left(\frac{z - Z(t)}{h}\right). \quad (7)$$

$$\hat{f}(x) = \frac{(detS)^{-1/2}}{nh^d} \sum_{t=1}^{n} K\left(S^{-1/2}\frac{x - X(t)}{h}\right). \quad (8)$$

The optimal kernel width h^* for a fixed core width is determined by minimizing the average integral root mean square error (MISE) [65]. For example, when the distribution of observations is normal with a unit covariance matrix in Gaussian kernel, the expression h^* proposed by [65] is $h^* = An^{-\frac{1}{d+4}}$, here $A = [4/(2d+1)]^{\frac{1}{d+4}}$. More sophisticated kernel width selection methods (such as the least-squares cross-checking method) are obtained by more complex and lengthy calculations [66–70].

In practical research, the kernel width is often selected experimentally. If the value of h is small, the density function estimate has more modes that correspond to the layout of the observed data. A higher value of h means more significant smoothing of the estimate.

Although fixed-core width density estimates are widely used to estimate non-parametric densities, they often have some practical drawbacks [65]. For example, fixed-core width density estimates do not ensure the distribution ends' integrity without over-smoothing the underlying bulk density.

2.1. Adapted Kernel Density Estimate (AKDE)

A good improvement on the fixed kernel width density estimate is the adapted kernel density estimate [65]. The adapted kernel density estimate is constructed similarly to the fixed kernel width density estimate. The kernel describes the density at each observed point. In this case, the kernel width is already considered when moving from one observation to another. In areas of different smoothness, it is appropriate to take different kernel widths. This method consists of two steps: estimation of the adapted kernel width and density estimation by the kernel method, using the information obtained in the first step. The algorithm can be summarized as follows:

Step 1: The elements of sample $X = (X(1), \ldots, X(n))$ are standardized to $Z = (Z(1), \ldots, Z(n))$ such that $\hat{E}[Z] = 0$ and $\hat{E}[ZZ'] = I$.

Step 2: Estimates $\widetilde{f}_Z(z)$ of the fixed kernel density estimate (3) satisfying the condition $\widetilde{f}_Z(Z(t)) > 0, \forall t$.

Step 3: The local width parameter is determined $\lambda_t = \left(\frac{\widetilde{f}_Z(Z(t))}{g}\right)^{-\gamma}$, where g is $\widetilde{f}_Z(z)$ the geometric mean, $\log g = \frac{1}{n}\sum_{i=1}^{n} \log \hat{f}_Z(Z(t))$ and γ is the sensitivity parameter: $0 \leq \gamma \leq 1$.

Step 4: An adapted kernel estimate is made with variable-width kernels: $\hat{f}_Z(z) = \frac{1}{n}\sum_{t=1}^{n} h^{-d}\lambda_t^{-d}K\left(\frac{z-Z(t)}{h\lambda_t}\right)$.

Where h is the same global smoothness parameter as in Equation (3), the higher γ, the more sensitive the density selection. Quite often, the parameter value is selected as follows $\gamma = \frac{1}{2}$ [65,71].

2.2. Semi-Parametric Kernel Density Estimate (SKDE)

When data are scarce, parametric estimates are often applied even when the unknown density is not parameterized. Therefore, it is essential to mention the combination of parametric and non-parametric estimates. For example, one of the semi-parametric estimates of kernel density was examined by F. Hoti and L. Holmström [46]. This estimate divides the random vector into two subvectors and estimates the distribution density of one of them by the kernel method. Afterward, another relative density is approximated by the Normal distribution density [46]. For example, suppose d and s are positive integers $d \geq 2$, $1 \leq s \leq d-1$. Using this method, the d-dimensional vector $X \in R^d$ is decomposed into two s and (d–s) dimensioned subvectors $X = \begin{pmatrix} Y \\ Z \end{pmatrix}$, and the sample is decomposed accordingly: $X = \begin{pmatrix} Y \\ Z \end{pmatrix}$, where $Y \in R^s$, $Z \in R^{d-s}$. The evaluated density function is expressed as the product of the distribution density of the random vector Y and the conditional distribution density of the random vector Z: $f_X(x) = f_{(Y,Z)}(y,z) = f_Y(y)f_{Z|Y=y}\left(z\middle|y\right)$, $x = \begin{pmatrix} y \\ z \end{pmatrix} \in R^d$. Here f_X and f_Y are the densities of X and Y. $f_{Z|Y=y}$ is the density of Z when $Y = y$.

Suppose that the relative density $Y = y$ is multidimensional normal Gaussian, but the density f_Y does not belong to any family of parametric functions. The density f_X is then obtained by estimating f_Y in a non-parametric manner and applying a multidimensional Normal density to each $f_{Z|Y=y}$. The density function $f_Y(y)$, as with (8), is evaluated by the kernel method [65]. Since the sample elements are not standardized, the smoothness parameter is not the same in all directions. Therefore, using the kernel method, it is replaced by the s-dimensional matrix H:

$$\hat{f}(y) = \frac{1}{n}\sum_{t=1}^{n}\frac{1}{\det(H)}K\left(H^{-1}(y - Y(t))\right). \tag{9}$$

Usually, the shape of H is chosen diagonally—$H = diag(h_1, \ldots, h_s)$, and the smoothness parameters are calculated as follow

$$h_j = \left(\frac{4}{s+2}\right)^{1/(s+4)} n^{-1/(s+4)}\sigma_j. \tag{10}$$

It should be noted that this form, when s = 1, was proposed by B. W. Silverman [65]. Replacing the standard deviation σ_j of the component Yj with its estimate $\hat{\sigma}_j = \sqrt{\frac{\sum(X_j - \overline{X_j})^2}{n_j}}$ and by the rule of D. W. Scott [39] first multiplier is always between 0.924 and 1.059 $\hat{h}_j$ can be calculated as follows

$$\hat{h}_j = n^{-1/(s+4)}\hat{\sigma}_j. \tag{11}$$

This Scott's rule is easy to summarize for the smoothness matrix H:

$$\hat{H} = n^{-1/(s+4)}\ \hat{\Sigma}^{1/2}. \tag{12}$$

Here $\hat{\Sigma} = \mathrm{diag}\left(\hat{\sigma}_1^2, \hat{\sigma}_2^2, \ldots, \hat{\sigma}_s^2\right)$ is the diagonal matrix of Y empirical variances.

The conditional density $f_{Z|Y}(\cdot\left|y\right)$ is approximated by the Gaussian distribution $N(m(y), C(y))$, where $m(y)$, $C(y)$ denote the conditional mean of the vector Y and the conditional

covariance matrix: $m(y) = E(Z|Y = y)$, $y \in R^s$, $C(y) = E\left[(Z - m(y))'(Z - m(y))\middle|Y = y\right]$, $y \in R^s$. For the estimation of $m(y)$ and $C(y)$, it is proposed to apply the kernel smoothing:

$$\hat{m}(y) = \frac{\sum_{t=1}^{n} K_{H_2}(y - Y(t))Z(t)}{\sum_{j=1}^{n} K_{H_2}(y - Y(j))} = \sum_{t=1}^{n} W_{H_2}(y - Y(t))Z(t),\ y \in R^s. \tag{13}$$

Here are the weights $W_{H_2}(y - Y(t)) = \frac{K_{H_2}(y - Y(t))}{\sum_{j=1}^{n} K_{H_2}(y - Y(j))}$.

The sum of which is equal to one. The formula (13) can be understood as a regression estimate of the conditional mean function m of Nadaraya and Watson [72,73]. The conditional covariance matrix can be evaluated similarly $\hat{C}(y) = \sum_{t=1}^{n} W_{H_3}(y - Y(t))(Z(t) - \hat{m}(y))'$ $(Z(t) - \hat{m}(y))$, $y \in R^s$. The parametric estimate of the relative density $f(Z \mid Y) = y$ looks akin to this $\hat{f}_{Z|Y=y}(z) = [(2\pi)^{d-s}\det\hat{C}(y)]^{-1/2}\exp\left\{-\frac{1}{2}(z - \hat{m}(y))\hat{C}(y)^{-1}(z - \hat{m}(y))'\right\}$, $z \in R^{d\text{-}s}$. The estimate of the distribution density f_X of X then is: $\hat{f}_X(x) = \hat{f}_{(Y,Z)}(y, z) = \hat{f}_Y(y)\hat{f}_{Z|Y=y}(z)$, $x = (y,z) \in R^d$.

The procedure described above is called the semi-parametric kernel density estimate. In practice, even if the conditional assumption of the normality of several random vector components is satisfied. The decomposition dimensions also influence the accuracy of the density estimation results, and the choice of the coordinates influences the accuracy of the density estimation results. One way to select them is to use the least-squares method or the maximum likelihood cross-entropy method recommended by original method authors [46]. The authors propose the parameters H_2 and H_3 [46] to select $2H$.

2.3. Logspline Estimation (LSDE)

This subsection describes the logspline estimation (LSDE) calculation. One-dimensional polynomial splines are called partial polynomials of a certain degree. Breakpoints that contain a transition from one polynomial to another are called nodes. Suppose that the vector $t = (t_1, \ldots, t_K) \in R^K$ defines a set of such K points. Splines describe smooth connections, showing how different areas are separated by nodes [74]. These constraints are precisely defined by expressing partial polynomials in the number of continuous derivatives s. These include partially linear curves. If there are no restrictions, breakpoints are allowed in the nodes of these functions. Assuming that the functions are globally continuous, it is required that the individual linear parts meet at each node. If greater smoothness is needed (for continuous first-order derivatives), then the flexibility of the nodes is lost. Moreover, the curves are considered simple linear functions. The term "linear spline" is applied to a continuous partial linear function in the literature on approximation theory. Accordingly, the term "cubic spline" is assigned to continuous cubic functions with second-order continuous derivatives and nodes that allow jumps of third-order derivatives. If the polynomial degree is b and the vector of the nodes is t, then the set of polynomial splines with s continuous derivatives forms a linear space. For example, a set of linear splines with nodes in the sequence t is defined by function

$$1, x, (x - t_1)+, \ldots, (x - t_K)+. \tag{14}$$

Here $(\bullet)_+ = \max(\bullet, 0)$. We will rely on this set as the base of space. The base of the spline space of degree b and s smoothness consists of monomial whose form $(x - t_k)_+^{s+j}$, here $1 \le j \le b - s$. Using this formula, in the case of classical cubic splines, where $b = 3$ and $s = 2$, the base consists of elements

$$1, x, x^2, x^3, (x - t_1)^3+, \ldots, (x - t_K)^3+. \tag{15}$$

From the model point of view, this base is convenient because the individual functions at the nodes are merged. In expressions (14) and (15), each function is precisely associated with one of the nodes, and removing this function essentially corresponds to removing the

node itself. It is known that the numerical properties of functions (14) and (15) are poor. For example, the solution matrix deteriorates as rapidly as the number of nodes decreases in linear regression problems. A practical alternative is the so-called B-spline base [75,76]. These functions are designed to be supported in several contiguous intervals defined by nodes ($b + 1$ contiguous intervals are used for the smoothest splines). Suppose we can find the basis for splines of space $B_1(x; t), \ldots, B_J(x; t)$ b with smoothness s and a sequence of nodes t so that any function in space can be written as $g(x; \beta, t) = \beta_1 B_1(x; t) + \ldots + \beta_J B_J(x; t)$. Where the corresponding coefficient vector is $\beta = (\beta_1, \ldots, \beta_J)'$. As seen from (14) and (15), then spline spaces of maximum smoothness are used $J = K + b + 1$.

According to the title of the subsection, the object of this analysis is the logarithmic density. Suppose X is a random vector that takes values from the interval (L, U). In the individual case, L and U can be $\pm\infty$. The parameters L and U are set to $2t_1 - t_2$ and $2t_K - t_{K-1}$, respectively. If $\beta_1 \geq 0$ or $\beta_{K-1} \geq 0$, then the adjustment is made $2L_{old} - t_1$ and $U_{new} = 2U_{old} - t_K$ is performed. The method of Kooperberg and Stone [52,77–79], known as logspline, is implemented with cubic spline. The cubic spline is described in (15). These functions are also continuously differentiated, and the partial polynomials are defined accordingly in the sequence of nodes $t = (t_1, \ldots, t_K)$. In each interval $[t_1, t_2], \ldots, [t_{K-1}, t_K]$ cubic splines are also cubic polynomials, but at the edges $(L, t_1]$ and $[t_K, U)$ are linear functions. The minimum number of nodes is $K \geq 3$ (otherwise, a linear function or constant can be obtained). The basis form is $1, B_1(x; t), \ldots, B_J(x; t)$, where $J = K - 1$.

It is said that the vector $\beta = (\beta_1, \ldots, \beta_J)' \in R^J$ exists then $C(\beta, t) = log\left(\int_L^U exp(\beta_1 B_1(x; t) + \ldots + \beta_J B_J(x; t))dx\right) < \infty$. Suppose B denotes a set of such possible vectors. After selecting $\beta \in B$, the family of positive density functions in the interval (L, U) is defined the form of which is

$$g(x; \beta, t) = exp(\beta_1 B_1(x; t) + \ldots + \beta_J B_J(x; t) - C(\beta, t)),\ L < x < U. \tag{16}$$

Now, having a random sample n of magnitude $X(1), \ldots, X(n)$ from the interval (L, U) with an unknown density function f, the logical probability function corresponding to the model of logsplines (16) is

$$l(\beta, t) = \sum_i log(g(X_i; \beta, t)) = \sum_i \sum_j \beta_j B_j(X_i; t) - nC(\beta, t),\ \beta \in B. \tag{17}$$

where estimation of maximum likelihood $\hat{\beta} = argmax_{\beta \in B} l(\beta, t)$ and an estimate of density $\hat{f} = g(x; \hat{\beta}, t), L < x < U$.

Let us say that during the stepwise determination procedure, the sequence of models is denoted by v, the vth model has J_v base functions. The Generalized Akaike Information Criterion (AIC) selects the best model [80]. Suppose that $\hat{l}_v$ defines the estimate of the logic-likelihood function (17) for the vth model. The Equation defines the Akaike information criterion $AIC_{a,v}(t) = -2\hat{l}_v(t) + aJ_v$ for which the model has a loss parameter a. From many models, the one whose value of v minimizes $AIC_{a,v}$. Stone [52] recommends the use of $a = \log n$.

2.4. PPDE Algorithm. Estimation of the Projection Density of the Target

The projection pursuit density estimator (PPDE) proposed by Friedman is based on the target projection and consistent projection Gaussianization. The essence of J. H. Friedman and coauthors [54,55,81] in estimating the target projection density is to search for "interesting", small-measurement data projections. The distribution structures, where the projections have distributions that are very different (in the sense of some projection index) from Gaussian. Huber [82] made a heuristic proposal to consider the Gaussian distribution as the least interesting. This proposal is based on the facts that:

- The multidimensional Gaussian distribution is entirely defined by its linear structure (mean and covariance matrices). Therefore, it is desired to feel a data structure independent of the correlation and linear transformations such as the scale parameter.

- All projections of a multidimensional Gaussian distribution are also Gaussian distributions. Thus, if the projection differs insignificantly from the Gaussian distribution, it indicates that distribution is also close to the Gaussian.
- For multidimensional data with a structure in multiple projection directions, many projections will have a distribution close to normal. This statement follows from the central limit theorem.
- In the case of constant variance, the Gaussian distribution is considered to be the least informative.

Friedman developed Huber's idea and proposed an algorithm called exploratory target projection to estimate multidimensional non-parametric density. This procedure consists of five steps:

(1) Data standardization simplifies layout, scalability, and correlation structures;
(2) Projection index: the degrees of 'interest' in various directions are determined.
(3) Optimization strategy: search for the direction in which the projection index is the largest.
(4) Data transformation: the one-dimensional density is calculated in the chosen direction, and the data are multiplied.
(5) Density formation: multidimensional density is formed from the calculated one-dimensional densities. Multidimensional density is a function of one-dimensional densities.

The following projection index construction has been proposed. It is known that all projections of a multidimensional Gaussian distribution are one-dimensional Gaussian distributions. If the distribution in one direction is not Gaussian, then the multidimensional distribution is also not Gaussian. Therefore, the projection index $I(\tau)$ shows how far the one-dimensional density $f_\tau(y)$ is in the direction $\tau(Y = \tau' Z)$ from the Gaussian distribution when Z is a standardized quantity [83]:

$$\widetilde{I}(\tau) = \int_{-\infty}^{\infty} (f_\tau(y) - \Phi(y))^2 dy, \text{ where } \Phi(y) = \frac{1}{\sqrt{2\pi}} e^{\frac{-y^2}{2}}. \tag{18}$$

The projection direction τ, which maximizes the projection of a distribution $\widetilde{I}(\tau)$, is chosen to highlight the multimodal or other nonlinear structure of that distribution. We transform the data y by equality $R = 2\Phi(Y) - 1 = 2\Phi(\tau' Z) - 1,\ R \in [-1,\ 1]$, where $\Phi(u)$ is a function of the standard normal distribution $\Phi(u) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{u} e^{\frac{-t^2}{2}} dt$. The distribution density of the transformed quantity R, function $f_R(r)$ can be rewritten as

$$f_R(r) = \frac{f_\tau(y)}{\left|\frac{\partial r}{\partial y}\right|} = \frac{f_\tau(y)}{2\Phi(y)}. \tag{19}$$

Equation (18) can be rewritten by changing the variable y to r : $\widetilde{I}(\tau) = \int_{-1}^{1} 2\Phi(y)(f_R(r) - 1/2)^2 dr = \int_{-1}^{1} 2\Phi\left(\Phi^{-1}\left(\frac{R+1}{2}\right)\right)(f_R(r) - 1/2)^2 dr$. Friedman [55] proposed a slightly different form of the projection index $I(\tau)$, taking the integrated square error as a measure of R inequality:

$$I(\tau) = \int_{-1}^{1} (f_R(r) - 1/2)^2 dr = \int_{-1}^{1} f_R^2(r) dr - 1/2. \tag{20}$$

Note that if the distribution of Y is Gaussian, then $f_R(r) = \frac{1}{2}$, $\forall$r, and the projection index $I(\tau)$ is zero. The more the Y distribution differs from the normal, the higher the value of the index $I(\tau)$. Since $R \in [-1,\ 1]$, $f_R(r)$ can be decomposed by orthogonal Lagrangian polynomials, $\{\psi_j\}_{j=0}^{\infty}$, i.e., $f_R(r) = \sum_{j=0}^{\infty} b_j \psi_j(r)$:

$$I(\tau) = \int_{-1}^{1} f_R^2(r) dr - 1/2 = \int_{-1}^{1} \left[\sum_{j=0}^{\infty} b_j \psi_j(r)\right] f_R(r) dr - 1/2. \tag{21}$$

An iterative expression defines orthogonal Lagrangian polynomials. $\psi_0(r) = 1$ and $\psi_1(r) = r$. $\psi_j(r) = \frac{(2j-1)r\psi_{j-1}(r)-(j-1)\psi_{j-2}(r)}{j}$, then $j \geq 2$. It follows from the orthogonality property that the coefficients b_j can be calculated as follows $b_j = \frac{2j+1}{2}\int_{-1}^{1}\psi_j(r)f_R(r)dr = \frac{2j+1}{2}E_R\left[\psi_j(r)\right] = \frac{2j+1}{2}\frac{1}{n}\sum_{t=1}^{n}\psi_j(2\Phi(Y(t))-1)$, where $\int_{-1}^{1}\psi_j(r)f_R(r)dr = E_R\left[\psi_j(r)\right]$ is the mean of the sample approximates the expression. Thus, equality can be written as

$$I(\tau) = \int_{-1}^{1} f_R^2(r)dr - 1/2 = \sum_{j=1}^{s}\frac{2j+1}{2}E_R^2\left[\psi_j(r)\right]. \tag{22}$$

It should be noted that the infinite amount has been changed to finite. Such a change has advantages: the sum is calculated faster, giving robustness to the projection index. By summing only a finite number of members, the slowly fading "tails" of the projection distributions have a more negligible effect on the value of the projection index. Therefore, it is suggested to choose $4 \leq s \leq 7$.

There are many methods for finding "interesting" projections. The method used in this research for finding the 'most interesting' projection direction is a mixed optimization strategy [55,64,84]. After defining the analytical expression of the projection index, its gradient in the projection direction τ is obtained as follows

$$\frac{\partial I}{\partial \tau} = \frac{2}{\sqrt{2\pi}}\sum_{j=1}^{s}(2j+1)E\left[\psi_j(r)\right]E[\psi_j'(r)e^{-y^2/2}(z-\tau y)\Big]. \tag{23}$$

Here, the Lagrangian polynomial derivative is calculated by an iterative formula: $\psi_1'(r) = 1$, then $\psi_j'(r) = r\psi_{j-1}'(r) + j\psi_{j-1}(r)$, then $j \geq 1$. Initially, an approximate step optimizer is found by searching in the directions of the main components and their combinations so that the initial convergence to the maximum can be achieved quickly. Then, the approximate step optimizer (steepest ascent) quickly selects the projections required to ascend to the (local) maximum of the projection index. The projection index is used to search for 'interesting' data projections. However, it is usually not enough to find a single projection to reasonably estimate the multidimensional density. In general, "interesting" directions do not have to be orthogonal and may require more projection directions than the data dimension. Therefore, when estimating density by targeted projection, the so-called deletion of the data structure is applied. A nonlinear scale transformation is performed, found in the projection direction, so the distribution of the transformed data becomes normal. This operation ensures that the same direction as before was not found when searching for another projection direction.

The deletion of the data structure is based on the fact that if the projection of one-dimensional data projection $\tau'Z$ has a distribution density $f_\tau(y)$ and a corresponding distribution function F_τ, then the random variable is equal to

$$\widetilde{Y} = \Phi^{-1}(F_\tau(Y)), \tag{24}$$

where Φ^{-1} is the inverse of the standard normal distribution. Friedman [55] proposed to calculate the empirical estimate of the distribution function as follows $\hat{F}_\tau(y) = rank(Y)/n - \frac{1}{2n}$, where $rank(y)$ is the rank of Y in the whole sample of size n. Unfortunately, this estimate is not accurate and often results in a very uneven density function. By denoting $Z^{(0)} = Z$, we will discuss how $Z^{(k-1)}$ is obtained from $Z^{(k)}$. Based on Equation (14), $Z^{(k)}$ can be defined as

$$Z^{(k)} = Z^{(k-1)} + \left[\Phi^{-1}\left(F_\tau\left(\tau'Z^{(k-1)}\right)\right) - \tau'Z^{(k-1)}\right]\tau. \tag{25}$$

The same procedure is performed to find the 'most interesting' projection with $Z^{(k)}$ searching for a new direction. This sequence is repeated until the multidimensional distribution becomes close to the Gaussian distribution in all directions. It has been observed [55] that gaussianization in one direction disrupts normalcy in the directions

previously studied so that their projection index $I(\tau)$ is no longer zero. However, studies show [54] that the changes in results are minimal. Multidimensional density is calculated from one-dimensional density estimates.

The relationship between the multidimensional densities $Z^{(k)}$ and $Z^{(k-1)}$ (where $Z^{(k)}$ is the structure of the distant data $Z^{(k-1)}$ along the k-th projection $\tau(k)$) is $f_{\tau(k)}\left(z^{(k)}\right) = \frac{f_{\tau(k-1)}\left(z^{(k-1)}\right)}{\left|J_k\left(z^{(k-1)}\right)\right|}$ and $f_{\tau(k-1)}\left(z^{(k-1)}\right) = f_{\tau(k)}\left(z^{(k)}\right)\left|J_k\left(z^{(k-1)}\right)\right|$, here is the Jacobian $J_k\left(z^{(k-1)}\right) = \frac{\partial z^{(k)}}{\partial z^{(k-1)}} = \frac{\partial\left(Uz^{(k)}\right)}{\partial\left(Uz^{(k-1)}\right)} = \frac{\partial y^{(k)}}{\partial y^{(k-1)}} = \frac{f_{\tau(k)}\left(y^{(k-1)}\right)}{\Phi\left(y^{(k)}\right)} = \frac{f_{\tau(k)}\left(\tau'(k)z^{(k-1)}\right)}{\Phi\left(\tau'(k)z^{(k)}\right)} \geq 0$.

Starting from the initial multidimensional data $Z^{(0)}$ gaussianization procedure is performed for each "interesting" projection found by $I(\tau)$. After a certain number, the projections' multidimensional data $Z^{(M)}$ differ slightly from the normal distribution. Density $Z^{(0)}$ can be calculated as follows

$$\begin{aligned} f\left(z^{(0)}\right) &= f_{\tau(1)}\left(z^{(1)}\right)J_1\left(z^{(0)}\right) = f_{\tau(2)}\left(z^{(2)}\right)J_2\left(z^{(1)}\right)J_1\left(z^{(0)}\right) = f_{\tau(M)}\left(z^{(M)}\right)\prod_{k=1}^{M}J_k\left(z^{(k-1)}\right) \\ &\approx \Phi\left(z^{(M)}\right)\prod_{k=1}^{M}J_k\left(z^{(k-1)}\right) = \Phi\left(z^{(M)}\right)\prod_{k=1}^{M}\frac{f_{\tau(k)}\left(\tau'(k)z^{(k-1)}\right)}{\Phi\left(\tau'(k)z^{(k)}\right)}. \end{aligned} \tag{26}$$

The one-dimensional density of the projected data $f_{\tau(k)}\left(\tau'(k)z^{(k-1)}\right)$ is calculated according to Equation (18) or more precisely

$$f_{\tau(k)}\left(\tau'(k)z^{(k-1)}\right) = \Phi\left(\tau'(k)z^{(k-1)}\right)\sum_{j=0}^{s}\frac{2j+1}{n}\sum_{t=1}^{n}\psi_j\left(r_t^{(k-1)}\right)\psi_j\left(r^{(k-1)}\right). \tag{27}$$

Then, replacing the unknown one-dimensional distribution densities on the right-hand side (26) with their statistical estimates, we obtain

$$\hat{f}(z) = \Phi\left(z^{(M)}\right)\prod_{k=1}^{M}\frac{\hat{f}_{\tau(k)}\left(\tau'(k)z^{(k-1)}\right)}{\Phi\left(\tau'(k)z^{(k)}\right)}. \tag{28}$$

The target projection density estimate is calculated relatively quickly because of the shape of the multivariate projection index and the iterative relationship between polynomials.

2.5. Inversion Formula

When examining approximations of parametric methods, it should be emphasized that as the data dimension increases, the number of model parameters increases rapidly, making it more difficult to find accurate parameter estimates. One-dimensional data projections $X_\tau = \tau' X$ density f_τ is much easier to find than multidimensional data density f because there exists a mutually unambiguous correspondence, $f \leftrightarrow \left\{f_\tau,\ \tau \in R^d\right\}$. It is quite natural to try to find the multidimensional density f using the density estimates $\hat{f}_\tau$ of one-dimensional observational projections. It should be noted that in the case of the mixture, when the distributions are Gaussian, the projections of observations are also distributed according to the (one-dimensional) Gaussian mixture model

$$f_\tau(x) = \sum_{k=1}^{q} p_{k,\tau}\varphi_{k,\tau}(x) = f_\tau(x,\theta_\tau). \tag{29}$$

Here $\varphi_{k,\tau}(x) = \varphi\left(x;\ m_{k,\tau},\ \sigma_{k,\tau}^2\right)$– one-dimensional Gaussian density. The parameter θ of the multidimensional mixture. The distribution parameters of the data projections $\theta_\tau = \left(p_{k,\tau},\ m_{k,\tau},\ \sigma_{k,\tau}^2\right)$, $k = 1,\ldots,q$ are related by equations: $p_{j,\tau} = p_j$, $m_{j,\tau} = \tau' M_j$ and $\sigma_{j,\tau}^2 = \tau' R_j \tau$. Using the inversion formula

$$f(x) = \frac{1}{(2\pi)^d}\int_{R^d} e^{-it'x}\psi(t)dt, \tag{30}$$

where $\psi(t) = Ee^{it'x}$ denotes the characteristic function of the random variable X. Marking $u = |t|$, $\tau = t/|t|$ and changing the variables to a spherical coordinate system, density is written

$$f(x) = \frac{1}{(2\pi)^d} \int_{\tau:\ |\tau|=1} ds \int_0^\infty e^{-iu\tau' x} \psi(u\tau) u^{d-1} du. \tag{31}$$

Here, the first integral is understood as the surface integral of the unit sphere. After noting the characteristic function of the projection of the observed random variable as $\psi_\tau(u) = Ee^{iu\tau' X}$. Then equality $\psi(u\tau) = \psi_\tau(u)$ holds. By selecting the set T of projection directions evenly spaced on the sphere and replacing the characteristic function with its estimate ($\hat{f}(x)$) a formula

$$\hat{f}(x) = \frac{A(d)}{\#T} \sum_{\tau \in T} \int_0^\infty e^{-iu\tau' x} \hat{\psi}_\tau(u) u^{d-1} e^{-hu^2} du, \tag{32}$$

is obtained to calculate the estimate [85,86]. Here and # continue to denote the number of elements in the set T. Using the d-meter ball volume

$$V_d(R) = \frac{\pi^{\frac{d}{2}} R^d}{\Gamma\left(\frac{d}{2}+1\right)} = \begin{cases} \frac{\pi^{\frac{d}{2}} R^d}{\left(\frac{d}{2}\right)!}, & \text{then } d \bmod 2 \equiv 0 \\ \frac{2^{\frac{d+1}{2}} \pi^{\frac{d-1}{2}} R^d}{d!!}, & \text{then } d \bmod 2 \equiv 1 \end{cases}, \tag{33}$$

the constant $A(d)$ depending on the data dimension can be calculated using

$$A(d) = \frac{(V_d(1))'_R}{(2\pi)^d} = \frac{d 2^{-d} \pi^{-\frac{d}{2}}}{\Gamma\left(\frac{d}{2}+1\right)}. \tag{34}$$

Computer simulation studies have shown that the density estimates obtained using the inversion formula are not smooth. Therefore, in formula (32), an additional multiplier e^{-hu^2} is used below the integral sign. This multiplier further smoothes the estimate $\hat{f}(x)$ (32) with the Gaussian kernel function. This form of the multiplier allows the value of the integral to be calculated analytically. The number of clusters and Gaussian mixture parameters was selected using the constructive procedure and software developed at the Lithuanian Institute of Mathematics and Informatics, applying the w^2 type criterion [87]. Formula (32) can be used for various estimates of the characteristic function of the projected data. We will discuss the two methods used in this work.

One of them is based on the density approximation of the Gaussian distribution mixture model. In the present case, after replacing the parameters of the Gaussian mixture with their statistical estimates ($\hat{p}_{k,\tau} = p_k$, $\hat{m}_{k,\tau} = \tau' M_k$, $\sigma^2_{k,\tau} = \tau' R_k \tau$) (Page 10), the following parametric estimate

$$\hat{\psi}_\tau(u) = \sum_{k=1}^{\hat{q}_\tau} \hat{p}_{k,\tau} e^{iu\hat{m}_{k,\tau} - u^2 \hat{\sigma}^2_{k,\tau}/2} \tag{35}$$

of the characteristic function is used, and adding (32) to (35) gives

$$\begin{aligned} \hat{f}(x) &= \frac{A(d)}{\#T} \sum_{\tau \in T} \sum_{k=1}^{\hat{q}_\tau} \hat{p}_{k,\tau} \int_0^\infty e^{iu(\hat{m}_{k,\tau} - \tau' x) - u^2(h + \hat{\sigma}^2_{k,\tau}/2)} u^{d-1} du \\ &= \frac{A(d)}{\#T} \sum_{\tau \in T} \sum_{k=1}^{\hat{q}_\tau} \hat{p}_{k,\tau} I_{d-1}\left(\frac{\hat{m}_{k,\tau} - \tau' x}{\sqrt{\hat{\sigma}^2_{k,\tau} + 2h}}\right) \left(\sqrt{\hat{\sigma}^2_{k,\tau} + 2h}\right)^{-d} \end{aligned} \tag{36}$$

and where $I_j(y)$ can be written as

$$I_j(y) = \mathrm{Re}\left[\int_0^\infty e^{iyz - z^2/2} z^j dz\right]. \tag{37}$$

It should be noted that only the real part of the expression can be considered here. The sum of the imaginary parts must be equal to zero. Because the density estimate $\hat{f}(x)$ can acquire only real values. The chosen form of the smoothing multiplier e^{-hu^2} allows relating the smoothing parameter h to the variances of the projection clusters—in the calculations, the variances are increased by $2h$. How to calculate expression (37) is given in Appendix B.

2.6. Modified Density Estimate of the Inversion Formula

One of the disadvantages of the inversion formula method defined in (32) is that the Gaussian distribution mixture model described by this estimate (where $f_k = \varphi_k$) evaluates well only the density of observations close to it. However, when approximating the density under study with a mixture of Gaussian distributions, the estimation of the density of the inversion formula often becomes complicated due to a large number of components with low a priori probabilities. Their number can be reduced by introducing a noise cluster—the modified algorithm based on a multidimensional Gaussian distribution mixture model. Let us use the inversion formula (30). The parametric estimate of the characteristic function of uniform distribution density can be calculated as follows

$$\hat{\psi}(u) = \frac{2}{(b-a)u} \sin\frac{(b-a)u}{2} \cdot e^{\frac{iu(a+b)}{2}}. \tag{38}$$

In uniform distribution density function (38), b is the maximum value, and a is the minimum value. In the density estimate calculation formula (32), construct the estimation of the characteristic function as a union of the characteristic functions of a mixture of Gaussian distributions and a uniform distribution with corresponding a priori probabilities as follows

$$\hat{\psi}_\tau(u) = \sum_{k=1}^{\hat{q}_\tau} \hat{p}_{k,\tau} e^{iu\hat{m}_{k,\tau} - u^2\hat{\sigma}^2_{k,\tau}/2} + \hat{p}_{0,\tau} \frac{2}{(b(\tau)-a(\tau))u} \sin\frac{(b(\tau)-a(\tau))u}{2} \cdot e^{\frac{iu(a(\tau)+b(\tau))}{2}}. \tag{39}$$

Here the second term describes a uniform distributed noise cluster and $\hat{p}_0$ is the weight of the noise cluster. Based on the parameters of the uniform distribution and the projected data, we can write

$$a(\tau) = (\tau' x)_{\min} - \frac{(\tau' x)_{\max} - (\tau' x)_{\min}}{2(n-1)} \text{ and} \tag{40}$$

$$b(\tau) = (\tau' x)_{\max} + \frac{(\tau' x)_{\max} - (\tau' x)_{\min}}{2(n-1)}. \tag{41}$$

Using notations such as (36), we can write

$$\begin{aligned} \hat{f}(x) = \frac{A(d)}{\#T} \sum_{\tau\in T} \Bigg[& \sum_{k=1}^{\hat{q}_\tau} \hat{p}_{k,\tau} I_{d-1}\left(\frac{\hat{m}_{k,\tau} - \tau' x}{\sqrt{\hat{\sigma}^2_{k,\tau} + 2h}}\right) \left(\hat{\sigma}^2_{k,\tau} + 2h\right)^{-\frac{d}{2}} \\ & + \frac{2\hat{p}_{0,\tau}}{b(\tau)-a(\tau)} I_{d-2}\left(\frac{a(\tau)+b(\tau)-2\tau' x}{2\sqrt{2h}}, \frac{b(\tau)-a(\tau)}{2\sqrt{2h}}\right) \cdot (2h)^{-\frac{d-1}{2}} \Bigg]. \end{aligned} \tag{42}$$

where the expression $I_j(y)$ is the same as (37) and its value is $I_j(y) = C_j(y)$ and

$$J_j(y,z) = \mathrm{Re}\left[\int_0^\infty e^{iyu - u^2/2} \cdot \sin zu \cdot u^j du\right]. \tag{43}$$

By integrating, we get

$$\begin{aligned} \int_0^\infty e^{iyu-u^2/2} \cdot \sin zu \cdot u^j du &= \int_0^\infty (\cos yu + i \sin yu) \cdot \sin zu \cdot e^{-u^2/2} \cdot u^j du \\ &= \int_0^\infty \left(\frac{\sin(y+z)u + \sin(z-y)u}{2} + i\frac{\cos(y-z)u - \cos(y+z)u}{2}\right) \cdot e^{-u^2/2} \cdot u^j du \\ &= \tfrac{1}{2} S_j(y+z) + \tfrac{1}{2} S_j(z-y) + i\tfrac{1}{2} C_j(y-z) - i\tfrac{1}{2} C_j(y+z). \end{aligned} \tag{44}$$

the above formula uses the variables $S_j(y)$ and $C_j(y)$, the calculation of which is given in formulas (52) and (53) in Appendix B.

3. Materials and Methods

Density estimation algorithms were presented in the previous section. The Monte Carlo method was used in this study. Such a comparison of algorithms allows us to measure the real observation density values and evaluate algorithms' efficiency. For the research, we used multidimensional (d = 2, 5, 10, 15) distributions of the Cauchy mixture

$$\textstyle\sum_{j=1}^{q} p_j C\left(x, m_j, u_j\right). \tag{45}$$

Additionally, $C\left(x, m_j, u_j\right)$ is defined as follows

$$C\left(x, m_j, u_j\right) = \prod_{k=1}^{d} \frac{u_{jk}}{\pi[u_{jk}^2 + \left(x_k - m_{jk}\right)^2]}. \quad (46)$$

Calculations were performed using sample sizes of n = 50, 100, 200, 400, 800 while changing the number of distributions, their weights, and centers (see. Table 1). In each case, 100,000 samples were generated.

Table 1. Parameters table.

Number of Components	Proportions of Components	Location Parameters	Separation Size of Locations
$q = 2$	$p_1 = (1 - p_2)$, $p_2 = 0.1, 0.3, 0.5$	$m_1 = (0, 0)$, $m_2 = (0.5i, 0.5i)$	$i = 1, 2, \ldots, 6$
$q = 3$	$p_1 = p_2 = (1 - p_3)/2$, $p_3 = 0.1, 1/3, 0.8$	$m_1 = (0, 0)$, $m_2 = (0.5i, 0.5i)$, $m_3 = (0.5i, 0)$	$i = 1, 2, \ldots, 6$
$q = 4$	$p_1 = p_2 = p_3 = (1 - p_4)/3$, $p_4 = 0.1, 0.25, 0.7$	$m_1 = (0, 0)$, $m_2 = (0.5i, 0.5i)$, $m_3 = (0.5i, 0)$, $m_4 = (0, 0.5i)$	$i = 1, 2, \ldots, 6$
$q = 2$	$p_1 = (1 - p_2)$, $p_2 = 0.1, 0.2, 0.3, 0.4, 0.5$	$m_1 = (0, 0, 0, 0, 0)$, $m_2 = (0.5i, 0.5i, 0.5i, 0.5i, 0.5i)$	$i = 1, 2, \ldots, 6$
$q = 3$	$p_1 = p_2 = (1 - p_3)/2$, $p_3 = 0.1, 0.2, 1/3, 0.4, 0.6, 0.8$	$m_1 = (0, 0, 0, 0, 0)$, $m_2 = (0.5i, 0.5i, 0.5i, 0.5i, 0.5i)$, $m_3 = (0.5i, 0.5i, 0, 0, 0)$	$i = 1, 2, \ldots, 6$
$q = 4$	$p_1 = p_2 = p_3 = (1 - p_4)/3$, $p_4 = 0.1, 0.16, 0.25, 0.4, 0.7$	$m_1 = (0, 0, 0, 0, 0)$, $m_2 = (0.5i, 0.5i, 0.5i, 0.5i, 0.5i)$, $m_3 = (0.5i, 0.5i, 0, 0, 0)$, $m_4 = (0, 0, 0.5i, 0.5i, 0.5i)$	$i = 1, 2, \ldots, 6$

In cases of d = 10, 15, the same weights were used as in d = 5. Additionally, centres are located on the apexes of the hypercube.

Algorithms used in the research: AKDE—adaptive kernel, PPDE—projection pursuit, LSDE—logspline, SKDE—semi-parametric kernel, IFDE—inversion formula, MIDE—inversion formula with noise cluster. In IFDE and MIDE methods are used mixture parameters, calculated with a program made in an institute of Mathematics and Informatics (Vilnius) [87].

Selection of parameters in the density estimation procedure. In this study, the Monte Carlo method aimed to perform the accuracy of the non-parametric estimates of distribution density previously described in the methodological sections (AKDE, PPDE, LSDE, SKDE, IFDE, MIDE) comparative analysis. The authors [34] propose to collect the value of the sensitivity parameter (γ, see. AKDE method step 3) used in the AKDE method from the set {0.2; 0.4; 0.6; 0.8}. The specific value of the parameter is determined by a probabilistic cross-check [88,89]. In the SKDE, all possible values of the sub-vector Y dimension s ($1 \leq s \leq d - 1$, where d is dimensions, see page 5) and their corresponding coordinates were reselected. The most factual errors were used to compare the results with other studied methods. The LSDE method minimizes the Akaike information criterion by selecting the number of baseline spline points [78]. The computer program for calculating this estimate is provided in the R package and was used in the study. Akaike information criterion $AIC = -2l(t) + aJ(t)$, J—degree of spline, $a = \log(n)$, l—probability function used to select the spline coefficients. The MIDE method has a smoothing parameter, h. The chosen form of the smoothing multiplier e^{-hu^2} allows relating the smoothing parameter h to the variances of the projection clusters. Modelling studies have shown that this method is sensitive to parameter selection. If h is set too low, the estimate becomes very slick and has large errors. Excessive smoothing of the density estimate does not greatly affect its quality. In the studies, it was observed that the estimation becomes uneven due to the similarity of the values of the observations projected in some directions, thus distinguishing low-weight components with small dispersions. The smoothing parameter (h) as well as the specific value of the noise cluster weight (probability) from the set {0.05; 0.1; 0.15; 0.2; 0.3;

0.4} are selected by cross-checking the least squares [65]. The vector of the estimate parameters is searched for in such a way that it minimizes the integrated square error

$$\Theta = \underset{\Theta}{\operatorname{argmin}} \int_{-\infty}^{\infty} \left(\hat{f_\Theta}(x) - f(x)\right)^2 dx = \underset{\Theta}{argmin}\left\{ \|\hat{f}_\Theta(x)\|_2^2 - \frac{2}{n}\sum_{t=1}^{n} \hat{f_\Theta}(X(t)) \right\}, \tag{47}$$

where Θ is the evaluated parameter and $F(x)$ is the observed random variable distribution function. Changing an unknown distribution function to an empirical distribution function yields an expression for the parameter estimate

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}}(\|\hat{f}_\Theta(x)\|_2^2 - \frac{2}{n}\sum_{t=1}^{n} \hat{f_\Theta}(X(t)|t), \tag{48}$$

where $\hat{f_\Theta}(x|t)$ is the value of the estimate at point x, which is calculated by subtracting the value of $X(t)$ from the observations. In addition, empirical research suggests that it is better to look for a maximum local minimum point rather than a global minimum [90]. Using PPDE method and following the recommendation of the paper [38], the order of the spread was $4 \le s \le 6$ (see Page 9), and the projection directions were chosen to maximize the value of the estimate of the design index (2) recommended by J. H. Friedman

$$I(\alpha) = \int_{-1}^{1} f_r^2(r)dr - \frac{1}{2} = \sum_{j=1}^{J} \frac{2j+1}{2} E_r^2\left[\psi_j(r)\right]. \tag{49}$$

4. Results and Discussion

This section presents the main results obtained during the simulations. We calculate the mean absolute error and (50) mean absolute percentage error (51) to evaluate the accuracy.

$$\delta_1 = \frac{1}{n}\sum_{t=1}^{n}\left|f(x(t)) - \hat{f}(x(t))\right| \cong \int \left|f(x) - \hat{f}(x)\right| f(x)dx. \tag{50}$$

$$\delta_2 = \frac{2}{n}\sum_{t=1}^{n}\left|\frac{f(x(t)) - \hat{f}(x(t))}{f(x(t)) + \hat{f}(x(t))}\right| \cong \int \left|f(x) - \hat{f}(x)\right| dx. \tag{51}$$

The result tables (Tables 2 and A1, Tables A2–A10) provide 100,000 samples densities mean absolute percentage error. The values in parentheses provide information about the standard deviation of errors. The best results in these tables are bolded and underlined. According to Table 2, it is concluded that when $n = 100$, $d = 5$, the best results are obtained by SKDE and MIDE methods. Based on Table A2, it can be observed that when $q = 2$, $n = 200$, the best results are obtained using SKDE and MIDE methods. According to Table A3, it is concluded that when $q = 3$, $n = 200$, in the case of highly overlapping distributions ($i = 1, 2$), the best results are obtained by the SKDE method, and in the case of more isolated distributions ($i \ge 3$)—by the MIDE method. Based on Table A4, it can be observed that when $q = 3$, $n \ge 400$, the best results are obtained by SKDE, while the second-best method is MIDE. According to Table A5, it is concluded that when $q = 4$, $n = 400$, in the case of highly overlapping distributions ($i \le 3$), the best results are obtained by the SKDE method and in the case of more isolated distributions ($i \ge 4$)—by the MIDE method. Table A6 shows results of $q = 4$, $n \ge 400$, it can be noticed that, in the case of highly overlapping or average isolated distributions ($i \le 5$), the best results are obtained by the SKDE method and in the case of more isolated distributions ($i = 6$)—by the MIDE method. Tables A7 and A8 show results of $q = 2$ and $n = 50$, It can be noticed that in all cases highly overlapping or isolated distributions, the best results are obtained by AKDE method and in the case of more isolated distributions ($i = 6$) with $p_1 = 0.6$; $p_2 = 0.4$— by the MIDE method. Tables A9 and A10 show results $q = 3$ and $n = 50$; the best results are obtained by the AKDE method in all cases (highly overlapping or isolated distributions).The LSDE method with huge outliers ($|x - m_j| > 100$ uj) is grouped with a more significant number of values closer to the centre of the distribution. With the help of the calculated spline coefficients, the density in the outliers is estimated at a value close to 10^{100}. That is incorrect, and in such cases, the use of this method is not recommended.

Table 2. An example of mean absolute percentage error.

Evaluation Methods		Density $d = 5; p_1 = p_2 = p_3 = 1/3; n = 100$					
		$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
AKDE	Mean	0.8268	0.8257	0.8198	0.8128	0.8066	0.8075
	SD	(0.0760)	(0.0814)	(0.0848)	(0.0827)	(0.0788)	(0.0731)
PPDE	Mean	0.9243	0.9319	0.9303	0.9300	0.9284	0.9250
	SD	(0.0500)	(0.0364)	(0.0375)	(0.0387)	(0.0410)	(0.0433)
LSDE	Mean	0.8043	0.8162	0.8583	0.8611	0.8613	0.8711
	SD	(0.0534)	(0.0540)	(0.0490)	(0.0349)	(0.0434)	(0.0577)
SKDE	Mean	**0.7158**	**0.7144**	**0.7088**	**0.7071**	0.7179	0.7227
	SD	(0.0260)	(0.0905)	(0.0905)	(0.0830)	(0.0631)	(0.0499)
IFDE	Mean	0.94593	0.8886	0.7857	0.8463	0.8761	0.8312
	SD	(0.0362)	(0.1318)	(0.0706)	(0.0380)	(0.1110)	(0.0538)
MIDE	Mean	0.7389	0.7332	0.7235	0.7149	**0.7121**	**0.7219**
	SD	(0.0280)	(0.0221)	(0.0338)	(0.0195)	(0.0208)	(0.0203)

The results for the smaller dimensions ($d = 2$) are presented in Table A1. It can be seen that the best results are obtained using the SKDE method, both in large- and small-scale overlapping cases ($i < 4$). On the other hand, in the case of isolated distributions ($i \geq 5$), good results were obtained by the MIDE method.

In the case of mean absolute percentage error, recommended using the semiparametric kernel when the sample has overlapping distributions. In the case of two dimensions ($d \sim 2$) and a sample is with overlapping distributions, it is recommended to use the semiparametric kernel method and for isolated distributions, to use the adaptive kernel method.

5. Conclusions

This paper reviewed the most popular and most often used nonparametric density estimation algorithms. The density estimation inversion formula was also presented in this article. It was observed that when a noise cluster is included, the results of the inversion formula improved statistically significantly. Based on the mean absolute error, in the case of higher dimension ($d \sim 5$) and small samples ($n \sim 50$), it is recommended to use the adaptive kernel method. If the sample is $n \sim 100$, then the modified inversion formula method showed the best results. For larger samples with overlapping distributions it is recommended to use a semi-parametric kernel and for more isolated distribution—modified inversion methods. Based on the mean absolute percentage error, it is recommended to use the semiparametric kernel when the sample is with overlapping distributions. In the case of two dimensions ($d \sim 2$) and a sample is with overlapping distributions, it is recommended to use the semiparametric kernel method. For isolated distributions, it is recommended to use the adaptive kernel method.

Author Contributions: Conceptualization, T.R. and M.L.; methodology, T.R.; software, T.R. and M.L.; formal analysis, T.R. and M.L.; investigation, T.R. and M.L.; writing—original draft preparation, T.R., M.L. and G.Č.; writing—review and editing, M.L. and G.Č.; supervision, T.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors are thankful to the area editor and the reviewers for giving valuable comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. 10^2 times upscaled mean absolute error with $d = 2$; $p_1 = 0.5$; $p_2 = 0.5$; $n = 100$.

Evaluation Methods		Density $d = 2; p_1 = 0.5; p_2 = 0.5; n = 100$					
		$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
AKDE	Mean	1.74	1.41	1.16	1.08	1.02	0.99
	SD	(0.94)	(0.80)	(0.70)	(0.59)	(0.54)	(0.50)
PPDE	Mean	2.21	1.85	1.52	1.32	1.25	1.21
	SD	(0.49)	(0.41)	(0.40)	(0.44)	(0.37)	(0.31)
LSDE	Mean	0.87	0.71	0.78	0.63	0.69	0.69
	SD	(0.43)	(0.20)	(0.08)	(0.08)	(0.04)	(0.09)
SKDE	Mean	**0.63**	**0.61**	**0.52**	**0.52**	**0.51**	**0.51**
	SD	(0.12)	(0.17)	(0.07)	(0.05)	(0.06)	(0.04)
IFDE	Mean	1.69	1.31	0.97	0.75	0.61	0.53
	SD	(0.06)	(0.10)	(0.08)	(0.01)	(0.04)	(0.06)
MIDE	Mean	0.69	0.66	0.57	0.55	**0.51**	**0.51**
	SD	(0.06)	(0.10)	(0.08)	(0.01)	(0.04)	(0.06)

Table A2. 10^4 times upscaled mean absolute error with $d = 5$; $p_1 = 0.7$; $p_2 = 0.3$; $n = 200$.

Evaluation Methods		Density $d=5; p_1 = 0.7; p_2 = 0.3; n = 200$					
		$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
AKDE	Mean	0.979	0.801	0.703	0.664	0.655	0.654
	SD	(0.171)	(0.146)	(0.145)	(0.155)	(0.158)	(0.159)
PPDE	Mean	1.001	0.822	0.722	0.681	0.671	0.669
	SD	(0.174)	(0.152)	(0.151)	(0.160)	(0.163)	(0.163)
LSDE	Mean	5.039	4.185	2.632	0.944	0.665	0.660
	SD	(1.265)	(6.747)	(1.081)	(0.138)	(0.140)	(0.112)
SKDE	Mean	**0.857**	**0.759**	0.705	**0.658**	**0.649**	**0.638**
	SD	(0.087)	(0.069)	(0.076)	(0.085)	(0.083)	(0.083)
IFDE	Mean	0.912	0.801	0.721	0.681	0.667	0.666
	SD	(0.133)	(0.149)	(0.151)	(0.160)	(0.162)	(0.163)
MIDE	Mean	0.956	0.788	**0.694**	0.661	0.657	0.640
	SD	(0.162)	(0.154)	(0.144)	(0.152)	(0.158)	(0.163)

Table A3. 10^4 times upscaled mean absolute error with $d = 5$; $p_1 = 0.45$; $p_2 = 0.45$; $p_3 = 0.1$; n= 200.

Evaluation Methods		Density $d = 5; p_1 = 0.45; p_2 = 0.45; p_3 = 0.1; n = 200$					
		$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
AKDE	Mean	0.970	0.725	0.576	0.533	0.504	0.500
	SD	(0.127)	(0.089)	(0.078)	(0.073)	(0.069)	(0.066)
PPDE	Mean	0.992	0.746	0.594	0.528	0.506	0.500
	SD	(0.137)	(0.092)	(0.077)	(0.072)	(0.069)	(0.066)
LSDE	Mean	1.057	0.775	0.652	0.590	0.508	0.503
	SD	(0.164)	(0.203)	(0.650)	(0.491)	(0.067)	(0.081)
SKDE	Mean	**0.6245**	**0.6274**	0.6312	0.629	0.630	0.628
	SD	(0.072)	(0.025)	(0.027)	(0.049)	(0.049)	(0.050)
IFDE	Mean	0.990	0.743	0.589	0.525	0.497	0.499
	SD	(0.136)	(0.091)	(0.076)	(0.071)	(0.071)	(0.067)
MIDE	Mean	0.993	0.746	**0.574**	**0.525**	**0.496**	**0.490**
	SD	(0.137)	(0.092)	(0.077)	(0.072)	(0.069)	(0.066)

Table A4. 10^4 times upscaled mean absolute error with $d = 5$; $p_1 = 0.4$; $p_2 = 0.4$; $p_3 = 0.2$; $n = 400$.

Evaluation Methods		Density $d = 5; p_1 = 0.4; p_2 = 0.4; p_3 = 0.2; n = 400$					
		$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
AKDE	Mean	0.916	0.689	0.527	0.445	0.410	0.396
	SD	(0.099)	(0.068)	(0.048)	(0.044)	(0.048)	(0.052)
PPDE	Mean	0.937	0.709	0.545	0.461	0.423	0.407
	SD	(0.109)	(0.074)	(0.049)	(0.044)	(0.048)	(0.052)
LSDE	Mean	0.815	0.549	0.511	0.443	0.404	0.401
	SD	(0.007)	(0.063)	(0.151)	(0.094)	(0.040)	(0.030)
SKDE	Mean	**0.655**	**0.499**	**0.413**	**0.388**	**0.385**	**0.384**
	SD	(0.064)	(0.049)	(0.034)	(0.031)	(0.028)	(0.027)
IFDE	Mean	0.937	0.709	0.544	0.460	0.423	0.404
	SD	(0.109)	(0.074)	(0.049)	(0.044)	(0.048)	(0.052)
MIDE	Mean	0.757	0.509	0.415	0.391	0.391	0.388
	SD	(0.109)	(0.074)	(0.049)	(0.044)	(0.048)	(0.052)

Table A5. 10^4 times upscaled mean absolute error with $d = 5$; $p_1 = 0.25$; $p_2 = 0.25$; $p_3 = 0.25$; $p_4 = 0.25$; $n = 400$.

Evaluation Methods		Density $d = 5; p_1 = 0.25; p_2 = 0.25; p_3 = 0.25; p_4 = 0.25; n = 400$					
		$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
AKDE	Mean	0.912	0.645	0.447	0.351	0.309	0.290
	SD	(0.128)	(0.068)	(0.029)	(0.019)	(0.026)	(0.030)
PPDE	Mean	0.934	0.665	0.464	0.365	0.321	0.299
	SD	(0.145)	(0.089)	(0.048)	(0.031)	(0.033)	(0.035)
LSDE	Mean	0.934	0.676	0.464	0.365	0.321	0.293
	SD	(0.145)	(0.064)	(0.048)	(0.031)	(0.033)	(0.039)
SKDE	Mean	**0.658**	**0.472**	**0.372**	0.345	0.316	0.290
	SD	(0.071)	(0.031)	(0.020)	(0.017)	(0.019)	(0.018)
IFDE	Mean	0.933	0.665	0.464	0.365	0.321	0.299
	SD	(0.145)	(0.089)	(0.048)	(0.031)	(0.033)	(0.035)
MIDE	Mean	0.889	0.622	0.433	**0.341**	**0.307**	**0.281**
	SD	(0.118)	(0.074)	(0.037)	(0.026)	(0.019)	(0.027)

Table A6. 10^4 times upscaled mean absolute error with $d = 5$; $p_1 = 0.1$; $p_2 = 0.1$; $p_3 = 0.1$; $p_4 = 0.7$; $n = 400$.

EvaluationMethods		Density $d = 5; p_1 = 0.1; p_2 = 0.1; p_3 = 0.1; p_4 = 0.7; n = 400$					
		$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
AKDE	Mean	0.957	0.800	0.678	0.617	0.587	0.571
	SD	(0.131)	(0.127)	(0.103)	(0.090)	(0.087)	(0.087)
PPDE	Mean	0.979	0.821	0.697	0.634	0.603	0.586
	SD	(0.141)	(0.137)	(0.112)	(0.099)	(0.094)	(0.093)
LSDE	Mean	0.979	0.821	0.697	0.634	0.596	0.586
	SD	(0.141)	(0.137)	(0.112)	(0.099)	(0.098)	(0.093)
SKDE	Mean	**0.687**	**0.580**	**0.514**	**0.496**	**0.491**	0.489
	SD	(0.076)	(0.070)	(0.058)	(0.058)	(0.058)	(0.056)
IFDE	Mean	0.979	0.820	0.697	0.634	0.602	0.585
	SD	(0.141)	(0.137)	(0.112)	(0.098)	(0.094)	(0.093)
MIDE	Mean	0.924	0.770	0.652	0.597	0.533	**0.488**
	SD	(0.135)	(0.131)	(0.108)	(0.093)	(0.092)	(0.091)

Table A7. 10^4 times upscaled mean absolute error with $d = 5$; $p_1 = 0.5$; $p_2 = 0.5$; $n = 50$.

Evaluation Methods		Density $d = 5; p_1 = 0.5; p_2 = 0.5; n = 50$					
		$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
AKDE	Mean	**1.093**	**0.828**	**0.758**	**0.741**	**0.739**	**0.740**
	SD	(0.095)	(0.099)	(0.114)	(0.123)	(0.130)	(0.134)
PPDE	Mean	1.147	0.872	0.794	0.770	0.764	0.762
	SD	(0.157)	(0.150)	(0.157)	(0.156)	(0.159)	(0.160)
LSDE	Mean	2.100	1.997	2.002	2.010	2.013	2.014
	SD	(0.078)	(0.028)	(0.017)	(0.019)	(0.024)	(0.025)
SKDE	Mean	1.149	0.875	0.797	0.773	0.765	0.763
	SD	(0.160)	(0.154)	(0.160)	(0.160)	(0.161)	(0.162)
IFDE	Mean	1.145	0.864	0.780	0.765	0.763	0.757
	SD	(0.163)	(0.137)	(0.140)	(0.150)	(0.163)	(0.156)
MIDE	Mean	1.094	0.860	0.759	0.767	0.742	0.752
	SD	(0.142)	(0.167)	(0.160)	(0.156)	(0.163)	(0.160)

Table A8. 10^4 times upscaled mean absolute error with $d = 5$; $p_1 = 0.6$; $p_2 = 0.4$; $n = 50$.

Evaluation Methods		Density $d = 5; p_1 = 0.6; p_2 = 0.4; n = 50$					
		$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
AKDE	Mean	**1.138**	**0.872**	**0.770**	**0.748**	**0.745**	0.746
	SD	(0.105)	(0.085)	(0.126)	(0.143)	(0.152)	(0.155)
PPDE	Mean	1.192	0.918	0.808	0.778	0.771	0.769
	SD	(0.150)	(0.136)	(0.172)	(0.178)	(0.181)	(0.182)
LSDE	Mean	2.114	1.995	1.977	1.983	1.986	1.987
	SD	(0.101)	(0.083)	(0.080)	(0.079)	(0.082)	(0.084)
SKDE	Mean	1.195	0.919	0.810	0.780	0.772	0.770
	SD	(0.154)	(0.138)	(0.174)	(0.182)	(0.183)	(0.183)
IFDE	Mean	1.183	0.906	0.802	0.778	0.765	0.769
	SD	(0.142)	(0.125)	(0.163)	(0.185)	(0.176)	(0.185)
MIDE	Mean	1.152	0.882	0.782	0.754	0.747	**0.742**
	SD	(0.136)	(0.124)	(0.155)	(0.175)	(0.176)	(0.180)

Table A9. 10^4 times upscaled mean absolute error with $d = 5$; $p_1 = 0.33$; $p_2 = 0.33$; $p_3 = 0.33$; $n = 50$.

Evaluation Methods		Density $d = 5; p_1 = 0.33; p_2 = 0.33; p_3 = 0.33; n = 50$					
		$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
AKDE	Mean	**1.166**	**0.828**	**0.634**	0.547	**0.512**	**0.500**
	SD	(0.120)	(0.086)	(0.057)	(0.064)	(0.074)	(0.080)
PPDE	Mean	1.224	0.879	0.677	0.581	0.540	0.523
	SD	(0.184)	(0.128)	(0.107)	(0.108)	(0.108)	(0.109)
LSDE	Mean	2.075	1.934	1.921	1.939	1.938	1.937
	SD	(0.127)	(0.094)	(0.058)	(0.054)	(0.048)	(0.044)
SKDE	Mean	1.226	0.881	0.678	0.583	0.542	0.524
	SD	(0.186)	(0.130)	(0.109)	(0.110)	(0.110)	(0.110)
IFDE	Mean	1.215	0.839	0.649	0.554	0.522	0.513
	SD	(0.175)	(0.099)	(0.110)	(0.102)	(0.110)	(0.111)
MIDE	Mean	1.182	0.834	0.638	**0.545**	0.518	0.501
	SD	(0.167)	(0.124)	(0.097)	(0.101)	(0.106)	(0.106)

Table A10. 10^4 times upscaled mean absolute error with $d = 5$; $p_1 = 0.45$; $p_2 = 0.45$; $p_3 = 0.1$; $n = 50$.

Evaluation Methods		Density $d = 5$; $p_1 = 0.45$; $p_2 = 0.45$; $p_3 = 0.1$; $n = 50$					
		$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
AKDE	Mean	**1.126**	**0.838**	**0.690**	**0.633**	**0.618**	**0.615**
	SD	(0.112)	(0.126)	(0.063)	(0.053)	(0.061)	(0.067)
PPDE	Mean	1.182	0.882	0.727	0.660	0.640	0.634
	SD	(0.156)	(0.132)	(0.091)	(0.085)	(0.087)	(0.088)
LSDE	Mean	2.101	2.002	1.985	2.010	2.014	2.015
	SD	(0.105)	(0.063)	(0.012)	(0.029)	(0.028)	(0.025)
SKDE	Mean	1.183	0.885	0.729	0.663	0.642	0.635
	SD	(0.157)	(0.134)	(0.094)	(0.089)	(0.090)	(0.090)
IFDE	Mean	1.170	0.859	0.702	0.649	0.624	0.619
	SD	(0.142)	(0.129)	(0.074)	(0.088)	(0.083)	(0.086)
MIDE	Mean	1.142	0.850	0.696	0.639	0.620	0.618
	SD	(0.141)	(0.125)	(0.080)	(0.084)	(0.083)	(0.087)

Appendix B

Calculate expression (36). Marked

$$C_j(y) = \int_0^\infty \cos yz \cdot e^{-z^2/2} \cdot z^j dz \text{ and} \tag{A1}$$

$$S_j(y) = \int_0^\infty \sin yz \cdot e^{-z^2/2} \cdot z^j dz. \tag{A2}$$

The Equation holds

$$\int_0^\infty e^{-iyz-z^2/2} z^j dz = C_j(y) + iS_j(y). \tag{A3}$$

Integration in parts results in

$$\begin{aligned} C_j(y) = e^{-\frac{z^2}{2}} z^{j-1} \cos yz \Big|_0^\infty + \int_0^\infty e^{-\frac{z^2}{2}} \Big((j-1) z^{j-2} \cos yz - yz^{j-1} \sin yz \Big) dz = \\ = 1_{\{j=1\}} + (j-1) C_{j-2}(y) - yS_{j-1}(y),\ j \geq 1. \end{aligned} \tag{A4}$$

Analogously expressing $S_j(y)$ and taking into account the constraints of the j index, recursive equations are obtained

$$C_j(y) = (j-1) C_{j-2}(y) - yS_{j-1}(y),\ j \geq 2 \text{ and} \tag{A5}$$

$$C_1(y) = 1 - yS_0(y) \text{ also} \tag{A6}$$

$$S_j(y) = (j-1) S_{j-2}(y) - yC_{j-1}(y),\ j \geq 2 \text{ and} \tag{A7}$$

$$S_1(y) = yC_0(y)\ ,\ \textit{then } j = 1. \tag{A8}$$

To calculate the functions $C_0(y)$ and $S_0(y)$ it is used that

$$(S_0(y))'_y = \int_0^\infty z \cos yz \cdot e^{-z^2/2} dz = C_1(y). \tag{A9}$$

From (A7) and (A10), it is obtained that S_0 satisfies the differential equation $S'_0(y) = 1 - yS_0(y)$. This Equation is solved by spreading S_0 by Taylor series

$$S'_0(y) = \sum_{l=0}^\infty c_{l+1}(l+1) y^{l+1} = 1 - \sum_{l=2}^\infty c_{l-1} y^l. \tag{A10}$$

Comparing the coefficients to similar members, their values are found $c_0 = 0, c_1 = 1, c_l = -c_{l-2}/l$, $l \geq 2$. Thus,

$$S_0(y) = \sum_{l=0}^{\infty} \frac{(-1)^l y^{2l+1}}{(2l+1)!!} = y - \frac{y^3}{3!!} + \frac{y^5}{5!!} - \frac{y^7}{7!!} + \ldots. \tag{A11}$$

C_0 is found from expression (50)

$$\begin{aligned} C_0(y) &= \int_0^{\infty} \cos yz \cdot e^{-z^2/2} dz = \tfrac{1}{2} \int_{-\infty}^{\infty} \cos yz \cdot e^{-z^2/2} dz \\ &= \tfrac{1}{2} \int_{-\infty}^{\infty} (\cos yz - i \sin yz) \cdot e^{-z^2/2} dz = \sqrt{\tfrac{\pi}{2}} e^{-y^2/2}. \end{aligned} \tag{A12}$$

Seeking integral (32) value $I_j(y) = C_j(y)$.

References

1. Duda, R.O.; Hart, P.E. *Pattern Classification and Scene Analysis*; Wiley: New York, NY, USA, 1973; Volume 3.
2. John, G.; Langley, P. Estimating Continuous Distributions in Bayesian Classifiers. In Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, 18–20 August 1995.
3. Wang, X.-Z.; He, Y.-L.; Wang, D.D. Non-Naive Bayesian Classifiers for Classification Problems with Continuous Attributes. *IEEE Trans. Cybern.* **2013**, *44*, 21–39. [CrossRef] [PubMed]
4. Azzalini, A.; Menardi, G. Clustering via nonparametric density estimation: The R package pdf Cluster. *arXiv* **2013**, arXiv:1301.6559.
5. Cuevas, A.; Febrero-Bande, M.; Fraiman, R. Cluster analysis: A further approach based on density estimation. *Comput. Stat. Data Anal.* **2001**, *36*, 441–459. [CrossRef]
6. Campello, R.J.; Kröger, P.; Sander, J.; Zimek, A. Density-based clustering. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2020**, *10*, e1343. [CrossRef]
7. Kwak, N.; Choi, C.-H. Input feature selection by mutual information based on Parzen window. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 1667–1671. [CrossRef]
8. Peng, H.; Long, F.; Ding, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1226–1238. [CrossRef] [PubMed]
9. Li, D.; Yang, K.; Wong, W.H. Density Estimation via Discrepancy Based Adaptive Sequential Partition. In Proceedings of the 30th Annual Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; Advances in Neural Information Processing Systems 29. pp. 1091–1099.
10. Rothfuss, J.; Ferreira, F.; Walther, S.; Ulrich, M. Conditional density estimation with neural networks: Best practices and benchmarks. *arXiv* **2019**, arXiv:1903.00954.
11. Trentin, E.; Lusnig, L.; Cavalli, F. Parzen neural networks: Fundamentals, properties, and an application to forensic anthropology. *Neural Netw.* **2018**, *97*, 137–151. [CrossRef]
12. Trentin, E. Soft-Constrained Neural Networks for Nonparametric Density Estimation. *Neural Process. Lett.* **2017**, *48*, 915–932. [CrossRef]
13. Huynh, H.T.; Nguyen, L. Nonparametric maximum likelihood estimation using neural networks. *Pattern Recognit. Lett.* **2020**, *138*, 580–586. [CrossRef]
14. Archambeau, C.; Verleysen, M. Fully nonparametric probability density function estimation with finite gaussian mixture models. In Proceedings of the 5th ICPAR Conference, Calcutta, India, 10–13 December 2003; pp. 81–84.
15. Priebe, C.E. Adaptive mixtures. *J. Am. Stat. Assoc.* **1994**, *89*, 796–806. [CrossRef]
16. Scott, D.W. Remarks on fitting and interpreting mixture models. *Comput. Sci. Stat.* **1999**, 104–109.
17. Delicado, P.; Del Río, M. A generalization of histogram type estimators. *J. Nonparametr. Stat.* **2003**, *15*, 113–135. [CrossRef]
18. Peel, D.; MacLahlan, G. *Finite Mixture Models*; Wiley Series in Probability and Statistics; John Wiley & Sons Inc.: Hoboken, NJ, USA, 2000.
19. Minnotte, M.C. Achieving higher-order convergence rates for density estimation with binned data. *J. Am. Stat. Assoc.* **1998**, *93*, 663–672. [CrossRef]
20. Tapia, R.A.; Thompson, J.R. *Nonparametric Probability Density Estimation*; Johns Hopkins University Press: Baltimore, MD, USA, 1978; p. 176.
21. Jones, M.C.; Samiuddin, M.; Al-Harbey, A.H.; Maatouk, T.A.H. The edge frequency polygon. *Biometrika* **1998**, *85*, 235–239. [CrossRef]
22. Simonoff, J.S. The anchor position of histograms and frequency polygons: Quantitative and qualitative smoothing. *Commun. Stat. Simul. Comput.* **1995**, *24*, 691–710. [CrossRef]
23. Scott, D.W. Averaged Shifted Histograms: Effective Nonparametric Density Estimators in Several Dimensions. *Ann. Stat.* **1985**, *13*, 1024–1040. [CrossRef]
24. Scott, D.W. On optimal and data-based histograms. *Biometrika* **1979**, *66*, 605–610. [CrossRef]
25. Fix, E.; Hodges, J. An important contribution to nonparametric discriminant analysis and density estimation. *Int. Stat. Rev.* **1951**, *3*, 233–238.

26. Silverman, B.W.; Jones, M.C.; Fix, E. An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation: Commentary on Fix and Hodges (1951). *Int. Stat. Rev.* **1989**, *57*, 233. [CrossRef]
27. Rosenblatt, M. A central limit theorem and a strong mixing condition. *Proc. Natl. Acad. Sci. USA* **1956**, *42*, 43–47. [CrossRef] [PubMed]
28. Parzen, E. On Estimation of a Probability Density Function and Mode. *Ann. Math. Stat.* **1962**, *33*, 1065–1076. [CrossRef]
29. Cencov, N.N. Estimation of an unknown distribution density from observations. *Soviet Math.* **1962**, *3*, 1559–1566.
30. Watson, G.S.; Leadbetter, M. On the estimation of the probability density, I. *Ann. Math. Stat.* **1963**, *34*, 480–491. [CrossRef]
31. Loftsgaarden, D.O.; Quesenberry, C.P. A Nonparametric Estimate of a Multivariate Density Function. *Ann. Math. Stat.* **1965**, *36*, 1049–1051. [CrossRef]
32. Schwartz, S.C. Estimation of Probability Density by an Orthogonal Series. *Ann. Math. Stat.* **1967**, *38*, 1261–1265. [CrossRef]
33. Epanechnikov, V.A. Non-Parametric Estimation of a Multivariate Probability Density. *Theory Probab. Its Appl.* **1969**, *14*, 153–158. [CrossRef]
34. Tarter, M.; Kronmal, R. On Multivariate Density Estimates Based on Orthogonal Expansions. *Ann. Math. Stat.* **1970**, *41*, 718–722. [CrossRef]
35. Kimeldorf, G.; Wahba, G. Some results on Tchebycheffian spline functions. *J. Math. Anal. Appl.* **1971**, *33*, 82–95. [CrossRef]
36. Cacoullos, T.; Sobel, M. An inverse sampling procedure for selecting the most probable event in a multinomial distribution. In *Multivariate Analysis*; Academic Press: New York, NY, USA, 1966; pp. 423–455.
37. Silverman, B. Choosing the window width when estimating a density. *Biometrika* **1978**, *65*, 1–11. [CrossRef]
38. Hwang, J.-N.; Lay, S.-R.; Lippman, A. Nonparametric multivariate density estimation: A comparative study. *IEEE Trans. Signal. Process.* **1994**, *42*, 2795–2810. [CrossRef]
39. Scott, D. Multivariate Density Estimation. *Ann. Stat.* **1992**, *20*, 1236–1265.
40. Kooperberg, C. Bivariate density estimation with an application to survival analysis. *J. Comput. Graph. Stat.* **1998**, *7*, 322–341.
41. Takada, T. Nonparametric density estimation: A comparative study. *Econ. Bull.* **2001**, *3*, 1–10.
42. Delgado, M.A.; Robinson, P.M. Nonparametric and semiparametric methods for economic research. *J. Econ. Surv.* **1992**, *6*, 201–249. [CrossRef]
43. Gill, R.D.; Wellner, J.A.; Præstgaard, J. Non-and semi-parametric maximum likelihood estimators and the von mises method (part 1) [with discussion and reply]. *Scand. J. Stat.* **1989**, *16*, 97–128.
44. Gill, R.D.; Van Der Vaart, A.W. Non-and semi-parametric maximum likelihood estimators and the von Mises method: II. *Scand. J. Stat.* **1993**, *20*, 271–288.
45. Hyndman, R.J.; Yao, Q. Nonparametric Estimation and Symmetry Tests for Conditional Density Functions. *J. Nonparametr. Stat.* **2002**, *14*, 259–278. [CrossRef]
46. Holmström, L.; Hoti, F. Application of semiparametric density estimation to classification. In Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, 26 August 2004; Volume 3, pp. 371–374.
47. Castellana, J.; Leadbetter, M. On smoothed probability density estimation for stationary processes. *Stoch. Process. Their Appl.* **1986**, *21*, 179–193. [CrossRef]
48. Chiu, S.-T. Bandwidth Selection for Kernel Density Estimation. *Ann. Stat.* **1991**, *19*, 1883–1905. [CrossRef]
49. Gu, C.; Qiu, C. Smoothing Spline Density Estimation: Theory. *Ann. Stat.* **1993**, *21*, 217–234. [CrossRef]
50. Härdle, W.; Müller, M. *Multivariate and Semiparametric Kernel Regression*; SFB 373 Discussion Paper; Springer: Heidelberg/Berlin, Germany, 1997.
51. Jones, M.C.; Marron, J.S.; Sheather, S.J. A brief survey of bandwidth selection for density estimation. *J. Am. Stat. Assoc.* **1996**, *91*, 401–407. [CrossRef]
52. Stone, C.J.; Hansen, M.H.; Kooperberg, C.; Truong, Y.K. Polynomial splines and their tensor products in ex-tended linear modeling: 1994 Wald memorial lecture. *Ann. Stat.* **1997**, *25*, 1371–1470. [CrossRef]
53. Aladjem, M. Projection pursuit mixture density estimation. *IEEE Trans. Signal. Process.* **2005**, *53*, 4376–4383. [CrossRef]
54. Friedman, J.H.; Stuetzle, W.; Schroeder, A. Projection pursuit density estimation. *J. Am. Stat. Assoc.* **1984**, *79*, 599–608. [CrossRef]
55. Friedman, J.H. Exploratory projection pursuit. *J. Am. Stat. Assoc.* **1987**, *82*, 249–266. [CrossRef]
56. Rudzkis, R.; Radavičius, M. Testing Hypotheses on Discriminant Space in the Mixture Model of Gaussian Distributions. *Acta Appl. Math.* **2003**, *79*, 105–114. [CrossRef]
57. Kalantan, Z.I.; Einbeck, J. Quantile-Based Estimation of the Finite Cauchy Mixture Model. *Symmetry* **2019**, *11*, 1186. [CrossRef]
58. Azzari, L.; Foi, A. Gaussian-Cauchy mixture modeling for robust signal-dependent noise estimation. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 5357–5361.
59. Teimouri, M. Statistical Inference for Mixture of Cauchy Distributions. *arXiv* **2018**, arXiv:1809.05722.
60. Jones, M.C. Discretized and interpolated kernel density estimates. *J. Am. Stat. Assoc.* **1989**, *84*, 733–741. [CrossRef]
61. Lambert, C.G.; Harrington, S.E.; Harvey, C.R.; Glodjo, A. Efficient on-line nonparametric kernel density estimation. *Algorithmica* **1999**, *25*, 37–57. [CrossRef]
62. Gasser, T.; Müller, H.-G.; Mammitzsch, V. Kernels for Nonparametric Curve Estimation. *J. R. Stat. Soc. Ser. B* **1985**, *47*, 238–252. [CrossRef]
63. Marron, J.; Nolan, D. Canonical kernels for density estimation. *Stat. Probab. Lett.* **1988**, *7*, 195–199. [CrossRef]

64. Fukunaga, K.; Hostetler, L. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Inf. Theory* **1975**, *21*, 32–40. [CrossRef]
65. Silverman, B.W. Density Estimation for Statistics and Data Analysis. In *Monographs on Statistics and Applied Probability*; Chapman & Hall: London, UK, 1994.
66. Breiman, L.; Meisel, W.; Purcell, E. Variable kernel estimates of multivariate densities. *Technometrics* **1977**, *19*, 135–144. [CrossRef]
67. Hall, P.; Sheather, S.J.; Jones, M.; Marron, J.S. On optimal data-based bandwidth selection in kernel density estimation. *Biometrika* **1991**, *78*, 263–269. [CrossRef]
68. Hart, J.D.; Vieu, P. Data-Driven Bandwidth Choice for Density Estimation Based on Dependent Data. *Ann. Stat.* **1990**, *18*, 873–890. [CrossRef]
69. Marron, J.S. An Asymptotically Efficient Solution to the Bandwidth Problem of Kernel Density Estimation. *Ann. Stat.* **1985**, *13*, 1011–1023. [CrossRef]
70. Wand, M.P.; Jones, M.C. Multivariate plug-in bandwidth selection. *Comput. Stat.* **1994**, *9*, 97–116.
71. Abramson, I.S. On Bandwidth Variation in Kernel Estimates-A Square Root Law. *Ann. Stat.* **1982**, *10*, 1217–1223. [CrossRef]
72. Nadaraya, E.A. On estimating regression. *Theory Probab. Its Appl.* **1964**, *9*, 141–142. [CrossRef]
73. Watson, G.S. Smooth regression analysis. *Sankhyā Indian J. Stat. Ser. A* **1964**, *26*, 359–372.
74. Smith, P.W.; Schumaker, L. Spline Functions: Basic Theory. *Math. Comput.* **1982**, *38*, 652. [CrossRef]
75. De Boor, C.; De Boor, C. *A Practical Guide to Splines*; Springer: New York, NY, USA, 1978; Volume 27.
76. Koo, J.-Y. Bivariate B-splines for tensor logspline density estimation. *Comput. Stat. Data Anal.* **1996**, *21*, 31–42. [CrossRef]
77. Hansen, M.H.; Kooperberg, C. Spline Adaptation in Extended Linear Models (with comments and a rejoinder by the authors). *Stat. Sci.* **2002**, *17*, 2–51. [CrossRef]
78. Kooperberg, C.; Stone, C.J. A study of logspline density estimation. *Comput. Stat. Data Anal.* **1991**, *12*, 327–347. [CrossRef]
79. Kooperberg, C.; Stone, C.J. Logspline density estimation for censored data. *J. Comput. Graph. Stat.* **1992**, *1*, 301–328.
80. Akaike, H. A new look at the statistical model identification. *IEEE Trans. Autom. Control.* **1974**, *19*, 716–723. [CrossRef]
81. Friedman, J.; Tukey, J. A Projection Pursuit Algorithm for Exploratory Data Analysis. *IEEE Trans. Comput.* **1974**, *100*, 881–890. [CrossRef]
82. Huber, P.J. Projection pursuit. *Ann. Stat.* **1985**, *13*, 435–475. [CrossRef]
83. Hall, P. On Polynomial-Based Projection Indices for Exploratory Projection Pursuit. *Ann. Stat.* **1989**, *17*, 589–605. [CrossRef]
84. Goffe, W.L.; Ferrier, G.; Rogers, J. Global optimization of statistical functions with simulated annealing. *J. Econ.* **1994**, *60*, 65–99. [CrossRef]
85. Ruzgas, T. The Nonparametric Estimation of Multivariate Distribution Density Applying Clustering Procedures. Ph.D. Thesis, Matematikos ir Informatikos Institutas, Vilnius, Lithuania, 2007.
86. Kavaliauskas, M.; Rudzkis, R.; Ruzgas, T. The projection-based multi-variate distribution density estimation. *Acta Comment. Univ. Tartu. Math.* **2004**, *8*, 135–141.
87. Rudzkis, R.; Radavičius, M. Statistical estimation of a mixture of Gaussian distributions. *Acta Appl. Math.* **1995**, *38*, 37–54. [CrossRef]
88. Van deLaan, M. *Efficient and Inefficient Estimation in Semiparametric Models*; CWI Tracts: Amsterdam, The Netherlands, 1995.
89. Van Der Laan, M.J.; Dudoit, S.; Keles, S. Asymptotic Optimality of Likelihood-Based Cross-Validation. *Stat. Appl. Genet. Mol. Biol.* **2004**, *3*, 1–23. [CrossRef]
90. Hall, P. Large Sample Optimality of Least Squares Cross-Validation in Density Estimation. *Ann. Stat.* **1983**, *11*, 1156–1174. [CrossRef]

mathematics

MDPI

Article

A New Clustering Method Based on the Inversion Formula

Mantas Lukauskas * and Tomas Ruzgas

Department of Applied Mathematics, Faculty of Mathematics and Natural Sciences, Kaunas University of Technology, 44249 Kaunas, Lithuania; tomas.ruzgas@ktu.lt
* Correspondence: mantas.lukauskas@ktu.lt

Abstract: Data clustering is one area of data mining that falls into the data mining class of unsupervised learning. Cluster analysis divides data into different classes by discovering the internal structure of data set objects and their relationship. This paper presented a new density clustering method based on the modified inversion formula density estimation. This new method should allow one to improve the performance and robustness of the k-means, Gaussian mixture model, and other methods. The primary process of the proposed clustering algorithm consists of three main steps. Firstly, we initialized parameters and generated a T matrix. Secondly, we estimated the densities of each point and cluster. Third, we updated mean, sigma, and phi matrices. The new method based on the inversion formula works quite well with different datasets compared with K-means, Gaussian Mixture Model, and Bayesian Gaussian Mixture model. On the other hand, new methods have limitations because this one method in the current state cannot work with higher-dimensional data (d > 15). This will be solved in the future versions of the model, detailed further in future work. Additionally, based on the results, we can see that the MIDEv2 method works the best with generated data with outliers in all datasets (0.5%, 1%, 2%, 4% outliers). The interesting point is that a new method based on the inversion formula can cluster the data even if data do not have outliers; one of the most popular, for example, is the Iris dataset.

Keywords: artificial intelligence; unsupervised machine learning; clustering; nonparametric density estimation; inversion formula

MSC: 62G05; 62G07; 62G30

check for updates

Citation: Lukauskas, M.; Ruzgas, T. A New Clustering Method Based on the Inversion Formula. *Mathematics* **2022**, *10*, 2559. https://doi.org/10.3390/math10152559

Academic Editor: Alicia Nieto-Reyes

Received: 21 June 2022
Accepted: 18 July 2022
Published: 22 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Artificial intelligence was first mentioned in 1956, but it was not so widely applied for a long time. Artificial intelligence has been widely used in recent decades. The ever-increasing power of possible computations has driven the high availability, development, and applications of artificial intelligence. Data mining is one of the most critical areas, as it is not limited to business, manufacturing, or other services. For this reason, data research has attracted a large number of researchers. Data clustering is one area of data mining that falls into the class of data mining of unsupervised learning. Cluster analysis divides data into different classes by discovering the internal structure of data set objects and their relationship. Clustering aims to create groups of similar observations/elements. The most similar elements, in this case, will be in one cluster and different elements in separate clusters [1].

With the increasing application of data mining, cluster analysis of data is also being applied in many areas: pattern recognition [2,3], bioinformatics [4,5], environment sciences [6], feature selection [7,8], or to solve different healthcare tasks. Clustering algorithms can be used to detect various diseases [9]. For example, different clustering techniques are used to identify breast cancer [10], Parkinson's disease [11,12], various psychological and psychiatric disorders [13], heart diseases and diabetes [14], and Alzheimer's disease [15,16], among many others.

Although there are many clustering methods, this problem is being addressed and remains a complex issue. Different clustering methods often do not work well with all data sets, and different methods are very much needed. Although one of the most widely used algorithms currently used is the k-means, as these methods are fast-acting and work well with certain data sets, there are still a lot of possible improvements to increase this method's accuracy.

Research focuses a lot on developing new density estimation procedures [17,18]. Moreover, in the last years, different scientists started to propose different robust density estimation methods even based on neural networks. There are different researches on this topic: Parzen neural networks [19], soft constrained neural networks [20], and others [21]. Some time ago, we presented a modified inversion formula for density estimation [22]. In this research, we found that this density estimation works better with different data than multiple density estimators. Therefore, we raised the hypothesis that modified inversion formula density estimation would be suitable for data clustering. Due to these facts, this paper aimed to present a new density clustering method based on the modified inversion formula density estimation. This new method should allow one to improve the performance and robustness of the k-means, Gaussian mixture model, and other methods. The main process of the proposed clustering algorithm consists of three main steps. Firstly, we initialized parameters and generated a T matrix. Secondly, we estimated the densities of each point and cluster. Third, we updated mean, sigma, and phi matrices. To compare results in this paper we used k-means, Gaussian mixture model (GMM), and Bayesian Gaussian mixture model (BGMM) clustering methods.

This paper is organized as follows. In the Section 2, we present introduction of the inversion formula and modified inversion formula density estimations and explain the idea behind these estimations. Then, the process of the proposed algorithm is presented in Section 2. In Section 3, we show empirical results with datasets used in the research, evaluation metrics, and experimental results. Finally, conclusions and future work with the new clustering method are given in Section 4

2. Estimation of the Density of the Modified Inversion Formula

Estimating probability density functions (pdf) is considered one of the most important parts of statistical modeling. This feature allows us to express random variables as a function of other variables while simultaneously allowing the detection of potentially hidden relationships in the data. If distribution density $f(x)$ satisfies the equation

$$f(x) = \sum_{k=1}^{q} p_k f_k(x) = f(x, \theta) \tag{1}$$

the random vector *X Rd* satisfies the distribution mixture model. The formula above (1), θ is a multi-dimensional parameter of the model. The function $f_k(x)$ is a function of the distribution density. *X* is a d-dimensional random vector with a distribution density $f(x)$. Additionally, we have independent copies of *X (X (1), . . . , X (n))* (sample of *X*).

We say that the sample satisfies the mixture model if *X (t)* satisfies (1). We call the size n the sample size (volume). The parameter q is called the mixture number of components, and p_k is the a priori probability. They meet the following conditions:

$$p_k > 0, \sum_{k=1}^{q} p_k = 1 \tag{2}$$

2.1. Gaussian Mixture and Inversion Density Estimation

It is important to notice that Projection (3) of the observations of the Gaussian mixture (1) is also distributed by the (one-dimensional) Gaussian mixture model:

$$f_\tau(x) = \sum_{k=1}^{q} p_{k,\tau}\ \varphi_{k,\tau}(x) = f_\tau(x, \theta_\tau) \tag{3}$$

here $\varphi_{k,\tau}(x) = \varphi(x; m_{k,\tau}, \sigma^2_{k,\tau})$—one-dimensional Gaussian density. Multivariate mixture parameter and data projection distribution parameters $\theta_\tau = (p_{k,\tau}, m_{k,\tau}, \sigma^2_{k,\tau})$, $k = 1, \ldots, q$ links equality

$$\begin{array}{c} p_{j,\tau} = p_j \\ m_{j,\tau} = \tau' M_j \\ \sigma^2_{j,\tau} = \tau' R_j \tau \end{array} \tag{4}$$

Using the inversion formula,

$$f(x) = \frac{1}{(2\pi)^d} \int\limits_{\mathbf{R}^d} e^{-it'x} \psi(t) dt \tag{5}$$

where $\psi(t) = \mathrm{E}e^{it'x}$ denotes the characteristic function of the random variable X. First, the set of projections directions T is selected. Additionally, the characteristic function is changed using the following formula:

$$\hat{f}(x) = \frac{A(d)}{\#T} \sum_{\tau \in T} \int\limits_0^\infty e^{-iu\tau'x} \hat{\psi}_\tau(u) u^{d-1} e^{-hu^2} du \tag{6}$$

where here and below, # denotes the number of elements in the set. With the formula for the volume of a d-dimensional sphere

$$V_d(R) = \frac{\pi^{\frac{d}{2}} R^d}{\Gamma\left(\frac{d}{2}+1\right)} = \begin{cases} \frac{\pi^{\frac{d}{2}} R^d}{\left(\frac{d}{2}\right)!}, & \text{kai } d \bmod 2 \equiv 0 \\ \frac{2^{\frac{d+1}{2}} \pi^{\frac{d-1}{2}} R^d}{d!!}, & \text{kai } d \bmod 2 \equiv 1 \end{cases} \tag{7}$$

one can calculate the constant $A(d)$ depending on the dimension of the data:

$$A(d) = \frac{(V_d(1))'_R}{(2\pi)^d} = \frac{d 2^{-d} \pi^{-\frac{d}{2}}}{\Gamma\left(\frac{d}{2}+1\right)} \tag{8}$$

Simulation studies show that the density estimates of the inversion formula are discontinuous/rough. The multiplier e^{-hu^2} in the Formula (6) further smoothes the estimate $\hat{f}(x)$ with the Gaussian kernel function. It is worth noting that this form of the multiplier allows analytical calculation of the value of the integral. Furthermore, results from extended Monte Carlo studies have shown that using this multiplier reduces the estimation errors. Formula (6) can be used for various estimates of the characteristic function of the projected data. A parametric estimate of the characteristic function was used in the present case.

$$\hat{\psi}_\tau(u) = \sum_{k=1}^{\hat{q}_\tau} \hat{p}_{k,\tau} e^{iu\hat{m}_{k,\tau} - u^2 \hat{\sigma}^2_{k,\tau}/2} \tag{9}$$

The chosen form of the smoothing multiplier e^{-hu^2} allows us to relate the smoothing parameter h to the variances of the projection clusters.

2.2. Modified Inversion Density Estimation

It is worth noting that the Gaussian mixture Model (1) described by the estimate (where $f_k = \varphi_k$) only estimates the density of the distribution close to it well. This can be seen as a drawback of the inversion formula method (9). The density estimation of the inversion formula often becomes complicated due to a large number of components with low a priori probability when the aim is to approximate the density under study with a mixture of Gaussian distributions. This problem can be solved by using a noise cluster.

We discuss a modified density estimation algorithm based on a multivariate Gaussian mixture model (Algorithm 1). However, first, let us define the parametric estimate of the characteristic function of a uniform distribution density:

$$\hat{\psi}(u) = \frac{2}{(b-a)u} \sin \frac{(b-a)u}{2} \cdot e^{\frac{iu(a+b)}{2}} \tag{10}$$

In the formula for calculating the density estimate, construct the estimate of the characteristic Function 9 as a union of the characteristic functions of a mixture of Gaussian distributions and a uniform distribution with corresponding a priori probabilities.

$$\hat{\psi}_\tau(u) = \sum_{k=1}^{\hat{q}_\tau} \hat{p}_{k,\tau} e^{iu\hat{m}_{k,\tau} - u^2\hat{\sigma}^2_{k,\tau}/2} + \hat{p}_{0,\tau} \frac{2}{(b-a)u} \sin \frac{(b-a)u}{2} \cdot e^{\frac{iu(a+b)}{2}} \tag{11}$$

Here, the second term describes the noise cluster with an even distribution, and $\hat{p}_0$ is the weight of the noise cluster,

$$a(\tau) = (\tau' x)_{\min} - \frac{(\tau' x)_{\max} - (\tau' x)_{\min}}{2(n-1)} \tag{12}$$

$$b(\tau) = (\tau' x)_{\max} + \frac{(\tau' x)_{\max} - (\tau' x)_{\min}}{2(n-1)}. \tag{13}$$

Algorithm 1: Clustering Algorithm Based on the Modified Inversion Formula Density Estimation (MIDE)

Input: Data set X= [X_1, X_2, . . . , X_n], cluster number K
Output: C1, C2, _, Ct and $\hat{M}$, $\hat{p}_k$, $\hat{R}$
Possible initiation of mean vector:
(1) random uniform initialization
(2) k-means
(3) random point initialization
Generate a T matrix. The set T is calculated when the design directions are evenly spaced on the sphere.
1 **For i = 1: t do**
2 Density estimation for each point and cluster based on (9)
Update $\hat{M}$, $\hat{p}_k$, $\hat{R}$ values based on (22, 23, 24)
4 **End**
5 **Return** C1, C2, _, Ct and $\hat{M}$, $\hat{p}_k$, $\hat{R}$

2.3. Modified Inversion Density Clustering Algorithm

This section aims to overview the critical aspects of the new modified inversion density estimation (MIDE) clustering method (Algorithm 1). This clustering algorithm uses the EM (expectation maximization) algorithm. The selection of the initial parameters of the EM algorithm is of particular importance for the clustering results, as each new combination of parameters can steer the cluster in a different direction. Random parameter selection is one of the most commonly used solutions for parameter initialization [23,24]. Random selection of initial parameters is a reasonably simple solution, as it is easy to implement. However, one of the significant disadvantages of this method is that such initialization

often results in significant deviations in the clustering results. In addition, the algorithm uses a continuous partition to initialize the initial cluster centers.

$$p(x) = \frac{1}{b-a} \tag{14}$$

In addition, another method of initialization presented in the software algorithm solution is the selection of random points. In this case, the initial cluster centers are selected not randomly from the entire space but by randomly selecting a point from the observations in the data set. However, this selection also has several drawbacks, as randomly selected points can be too close to each other, and selected points can also be exceptions in the data.

Hierarchical clustering can also be used to address the potential shortcomings of the random cluster center selection method. For the first time, such a classification algorithm that maintains a Gaussian mixture model to form a cluster tree was described by Fraley in 1998 [25]. Maitra [26] proposed a hierarchical clustering based on mean connectivity to obtain an initial model mean. Moreover, Meila and Heckerman [27] experimentally demonstrated that an algorithm using a pattern-based distance measure is better than a random method. This method is applied to the initial mean of the model. Perhaps the only major drawback it has observed so far is that computer computations take a long time and require a large amount of computer memory if there are many data.

One of the most commonly used methods for selecting cluster centers is the k-means and other heuristic clustering methods. This is one of the most widely used initial parameters selection methods. In the case of the initialization of K-means, firstly, the random cluster centers $\mu_1, \mu_2, \ldots, \mu_k \in \mathcal{R}^n$ are first selected, and then the procedure is performed until convergence is achieved.

$$c^{(i)} = \arg min_j \|x^{(i)} - \mu_j\|^2 \tag{15}$$

$$\mu_j = \frac{\sum_{i=1}^{m} 1\left\{c^{(i)} = j\right\} x^{(i)}}{\sum_{i=1}^{m} \left\{c^{(i)} = j\right\}} \tag{16}$$

Clustering using the modified inversion formula density estimation and the EM algorithm is explained below. If the distribution density X of a random vector has q maxima, then it can be approximated by a mixture of q single-mode distribution densities:

$$f(x) = \sum_{k=1}^{q} p_k f_k(x) \tag{17}$$

Suppose that the distribution of X depends on a random variable v, which acquires the values $1, \ldots, q$ with the corresponding probabilities $p_1, \ldots, p_q$. In classification theory, v is interpreted as the number of the class to which the observed object belongs. Thus, the X(t) observations would correspond to $v(t)$, $t = 1, \ldots, n$. The functions f_k are treated as the density of the conditional distribution X under the condition $v = k$. Based on this approach, loose clustering of the sample is understood as a posteriori probabilities.

$$\pi_k(x) = P\{v = k | X = x\} \tag{18}$$

when all $x \epsilon \{X(1), \ldots, X(n)\}$. Strict clustering of the sample would be an estimate of the random variables $v(1), \ldots, v(n)$ Take a breakdown into subsets based on equality

$$\hat{v}(t) = \arg \max_{k=1,\ldots,q} \hat{\pi}_k(X(t)) \tag{19}$$

The estimates $\hat{\pi}_k$ are obtained by approximating the unknown distribution density components with the density estimates of the inversion formula and using the EM (expectation maximization) algorithm. We briefly describe it as follows. Suppose that

Equation (15) holds and f_k is the density function of the inversion formula for the Gaussian mixture model, $k = 1, \ldots, q$, where q is the number of the clusters. In this case (17), let us denote the right side of the equation by $f(x,\theta)$, where $\theta = (p_k, M_k, R_k, a, b, k = 1, \ldots, q)$. Equality applies:

$$\pi_k(x) = \frac{p_k f_k(x)}{f(x,\theta)} \text{ and } k = \overline{1,q} \tag{20}$$

Having an estimate of θ, the estimates of the probabilities π_k (k-th cluster probability) are obtained from (20) using the "embedding" method, i. y. replacing the unknown parameters on the right with their statistical estimates. The EM algorithm is a reciprocal procedure for estimating the maximum likelihood

$$\theta^* = \arg\max_{\theta} L(\theta),\ L(\theta) = \prod_{t=1}^{n} f(X(t),\theta) \tag{21}$$

and to calculate the corresponding estimates $\hat{\pi}_k$. Several authors have independently proposed this algorithm for Gaussian mixture analysis, including Hasselblad [28] and Behboodian [29]. Its properties were later well examined in refs. [30–32] and other works. The EM algorithm has received much attention in various review articles and monographs [33–35]. Suppose that after r cycles, we obtain the estimates $\hat{\pi}_k = \hat{\pi}_k^{(r)}$. The new estimate $\hat{\theta} = \hat{\theta}^{(r+1)}$ is then defined by the equations:

$$\hat{p}_k = \frac{1}{n}\sum_{t=1}^{n} \hat{\pi}_k(X(t)) \tag{22}$$

$$\hat{M}(k) = \frac{1}{n\hat{p}_k}\sum_{t=1}^{n} \hat{\pi}_k(X(t)) \cdot X(t) \tag{23}$$

$$\hat{R}(k) = \frac{1}{n\hat{p}_k}\sum_{t=1}^{n} \hat{\pi}_k(X(t))[X(t) - \hat{M}(k)] \cdot [X(t) - \hat{M}(k)]' \tag{24}$$

where $k = 1, \ldots, q$. Entering $\hat{\theta}^{(r+1)}$ to the right of (20), we find $\hat{\pi}^{(r+1)}(X(t))$, $k = \overline{1,q}$, $t = \overline{1,n}$. As a result of this recursive procedure, we obtain a non-decreasing sequence $L(\hat{\theta}^{(r)})$, but whether it converges to the point of the global maximum depends very much on the initial estimate $\hat{\theta}^{(0)}$ (or $\hat{\pi}^{(0)}$).

In the case of the high mixture model (GMM), the best number of clusters is selected based on the information criterion. This algorithm's most commonly used information criteria are AIC, BIC, and others. When these information criteria reach their global minimum or maximum, an optimal number of clusters can be said to have been reached. However, there are also some problems in applying these criteria. First, it is necessary to calculate the global maximum of the function as the maximum value of the local maxima, but sometimes this is performed with exceptions. Therefore, applying any procedure cannot guarantee that such a global maximum will be found in such a case.

On the other hand, applying these criteria assumes that one of the parametric methods being compared is correct. This assumption makes the criterion unstable. The arguments presented to raise the question of whether it may be worthwhile to use nonparametric criteria to test the adequacy of the distribution mix model. Several problems can be encountered if the correct number of clusters is not selected. If the number of components selected is too small, then no clear clusters are formed, and one cluster includes more. Meanwhile, if the selected number of clusters is too large, it is much more challenging to calculate clusters in the first place, and less generalizing clusters are also obtained. An attempt to accurately select the number of clusters was provided by Xie, et al. [36], in which an adaptive selection of components/clusters of the Gaussian mixture model was proposed.

3. Experimental Analyses

This section provides information about the modified inversion function based on the proposed clustering method. This section consists of three parts. The first part provides information on the clustering assessment methods used in the empirical study. The second part of this chapter provides information on the data sets used in the study. Finally, the third part of the chapter presents the study's main results.

3.1. Evaluation Metrics

This section presents the main evaluation metrics used in the empirical study. In order to evaluate the results of clustering, it is essential to choose the appropriate evaluation metrics, as they can also determine the evaluation of clustering. In this study, clustering methods were used to compare J-Score [37], Normalized Mutual Information (NMI) [38], Adjusted Rand Index (ARI) [39] and Accuracy (ACC) [40], and The Fowlkes–Mallows index (FMI) [41]. These metrics were chosen based on the fact that the actual data clusters are known in advance because if the clusters were not known in advance, then the evaluation metrics could be: Calinski and Harabasz score, also known as Variance Ratio Criterion [42], Davies–Bouldin score [43], or others.

J-score. Ahmadinejad and Liu [37] suggested a new clustering evaluation metric, J-score. The J-score is a simple and robust measure of clustering accuracy. It addresses the matching problem and reduces the risk of overfitting that challenge existing accuracy measures [37]. Bidirectional set matching: Suppose a dataset contains N datapoints belonging to T true classes, and cluster analysis produces K hypothetical clusters. To establish the correspondence between T and K, we first considered each class as reference and identify its best-matched cluster ($T \rightarrow K$). Specifically, for a class $t \in T$, we searched for a cluster $k \in K$ that has the highest Jaccard index,

$$I_t = \max_{k \in K} \frac{|V_t \cap V_k|}{|V_t \cup V_k|} \tag{25}$$

where Vt and Vk are the set of datapoints belonging to class t and cluster k, respectively, and $|\cdot|$ denotes the size of a set. We then considered each cluster as a reference and identified its best-matched class ($K \rightarrow T$) using a similar procedure. For a cluster $k \in K$, we searched for a class $t \in T$ with the highest Jaccard index,

$$I_k = \max_{t \in T} \frac{|V_t \cap V_k|}{|V_t \cup V_k|} \tag{26}$$

Calculating overall accuracy: To quantify the accuracy, we aggregated Jaccard indices of individual clusters and classes, accounting for their relative sizes (i.e., number of data points). We first calculated a weighted sum of It across all classes as $R = \sum_{t \in T} \left(\frac{V_t}{N} I_t \right)$, and a weighted sum of Ik across all clusters as $P = \sum_{k \in K} \left(\frac{V_k}{N} I_k \right)$. We then took their harmonic mean as J score,

$$J = \frac{2 \times R \times P}{R + P} \tag{27}$$

To work with this metric, we implemented the calculation of the J-score metric in the Python programming language. The program code is available in Appendix A.

Normalized Mutual Information (NMI). The mutual information (MI) of two random variables is a measure of the mutual dependence of the two variables. MI normalization was performed for greater comparability and better interpretation, and the NMI metric was obtained. The values of this metric can range from 0 to 1.0; in this case, zero would indicate no relationship between the variables, while one would indicate a perfect correlation.

$$NMI = \frac{MI(Y', Y)}{\sqrt{H(Y')H(Y)}} \tag{28}$$

Here, Y' are the predicted labels and Y are the actual classes known in advance. $MI(Y', Y)$ is the mutual information between predicted labels and actual labels. This formula also uses the entropy $H(*)$ of predicted labels and actual labels.

Adjusted Rand Index (ARI). The Rand index evaluates the similarity between two clusters. Pairs of all observations are used to calculate this similarity. When calculating this index, there are observations in the assignment to clusters, and how this coincides with the real labels of the clusters.

$$ARI = \frac{2(ad - bc)}{(a + b)(b + d) + (a + c)(c + d)} \tag{29}$$

In the given formula, a is a number that describes how many data points are correctly assigned to a cluster. B is the number of observations in a pair assigned to the same cluster (predicted and actual cluster values match). Here, c is the number of observations in a pair for which the predicted cluster matches, but the actual cluster values do not match. Finally, D is the number of data points in a pair that neither the predicted case nor the actual case belongs to the same cluster.

Accuracy (ACC). Accuracy is often used to measure the quality of classification. It is also used for clustering. It is calculated as the sum of the diagonal elements of the confusion matrix, divided by the number of samples to obtain a value between 0 and 1.

$$ACC = \frac{1}{N} \sum_{i=1}^{k} n_i \tag{30}$$

where N is the total number of data points in the dataset, n_i is the number of data points correctly divided into the corresponding cluster i, and k is the cluster number.

The Fowlkes–Mallows index (FMI). The Fowlkes–Mallows score FMI is defined as the geometric mean of pairwise precision and recall.

$$FMI = \frac{TP}{\sqrt{(TP + FP)(TP + FN)}} \tag{31}$$

True Positive (*TP*) is the number of pairs of points belonging to the same clusters (true label = predicted label). False Positive (*FP*) is the number of the points that belong to the same cluster in the true labels but do not belong to the same cluster in the predicted clusters. False Negative (*FN*) is the number of the pairs of points that belong in the same clusters in the predicted labels and not in the true labels. The higher the metric value, the better the cluster separation is (the maximum possible value of the metric is 1, and the minimum is 0).

3.2. Experimental Datasets

To test the developed method and compare it with other methods, 25 data sets were used in this study. Data sets can be divided into three categories: synthetic, real, and generated data with outliers. Synthetic data sets are data sets that have been generated by other authors and are often used in research on clustering methods. Actual datasets include datasets such as Iris, Wine, Diabetes, and others, and these datasets are also selected based on datasets used by other authors. The third category of generated data with outliers is generated as Gaussian data, including a certain amount of outliers: 0.5%, 1%, 2%, and 4%. These datasets aim to evaluate how different methods work with data with an appropriate amount of outliers. The table below (see Table 1) shows the data sets used.

Table 1. A description of the data set used.

ID	Data Sets	Sample Size (N)	Dimensions (D)	Classes
Synthetic				
1	Aggregation	788	2	7
2	Atom	800	3	2
3	D31	3100	2	31
4	R15	600	2	15
5	Gaussians1	100	2	2
6	Threenorm	1000	2	2
7	Twenty	1000	2	20
8	Wingnut	1016	2	2
Real				
9	Breast	570	30	2
10	CPU	209	6	4
11	Dermatology	366	17	6
12	Diabetes	442	10	4
13	Ecoli	336	7	8
14	Glass	214	9	6
15	Heart-statlog	270	13	2
16	Iono	351	34	2
17	Iris	150	4	3
18	Wine	178	13	3
19	Thyroid	215	5	3
Generated clusters with outliers				
20	2 clusters (0.5% outliers)	1005	2	2
21	2 clusters (1% outliers)	1010	2	2
22	2 clusters (2% outliers)	1020	2	2
23	2 clusters (4% outliers)	1040	2	2
25	3 clusters (0.5% outliers)	1005	2	3
26	3 clusters (1% outliers)	1010	2	3
27	3 clusters (2% outliers)	1020	2	3
28	3 clusters (4% outliers)	1040	2	3

3.3. *Performances of Clustering Methods*

To avoid the possible influence of successful parameter initialization on test results, all experiments were performed 10,000 times. For the k-means method, initial cluster centers were randomly selected based on the 100 runs best solutions. For GMM (Gaussian Mixture Model), BGMM (Bayesian Gaussian Mixture Model) and clustering based on modified inversion density estimation (MIDE) initial center were selected based on the k-means centers initialization. The following table provides information on the Accuracy metric values for the different clustering algorithms. Other evaluation metrics like NMI, ARI, FMI, and J-Score can be found in the Appendix B tables.

The accuracy results for different datasets are presented in Table 2. It can be seen that the new method based on the inversion formula works quite well with different datasets compared with K-means, Gaussian Mixture Model (GMM), and Bayesian Gaussian Mixture model (BGMM).

Table 2. Different models were comparative (means and standard deviation) based on the accuracy (ACC) for 10,000 runs.

Dataset	K-Means		GMM		BGMM		MIDEv1		MIDEv2	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Synthetic										
Aggregation	0.857	0.005	0.835	0.075	**0.907**	0.042	0.889	0.008	0.895	0.009
Atom	0.710	0.002	0.618	0.028	0.637	0.022	0.723	0.002	**0.746**	0.004
D31	**0.972**	0.015	0.928	0.028	0.601	0.022	0.721	0.017	0.723	0.013
R15	**0.997**	0.000	0.979	0.036	0.669	0.011	0.768	0.008	0.855	0.007
Gaussians1	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	**1.000**	0.000
Threenorm	0.591	0.001	0.612	0.047	0.549	0.006	0.649	0.003	**0.679**	0.003
Twenty	1.000	0.000	0.985	0.029	0.838	0.075	-	-	-	-
Wingnut	0.909	0.000	0.964	0.000	**0.965**	0.000	0.876	0.000	0.880	0.000
Real										
Breast	0.908	0.003	**0.940**	0.001	0.933	0.001	-	-	-	-
CPU	0.738	0.008	0.574	0.073	0.590	0.093	0.808	0.007	**0.828**	0.006
Dermatology	0.739	0.044	0.737	0.080	**0.756**	0.109	-	-	-	-
Diabetes	0.356	0.010	0.419	0.043	0.439	0.033	0.420	0.008	**0.448**	0.007
Ecoli	0.649	0.013	0.753	0.018	0.739	0.006	0.714	0.011	**0.754**	0.009
Glass	0.447	0.016	0.468	0.025	0.483	0.025	0.465	0.013	**0.487**	0.017
Heart-statlog	**0.837**	0.002	0.794	0.045	0.791	0.045	-	-	-	-
Iono	0.707	0.000	**0.810**	0.029	0.803	0.023	-	-	-	-
Iris	0.831	0.007	0.953	0.065	0.838	0.049	0.933	0.006	**0.955**	0.005
Wine	0.966	0.000	0.953	0.048	**0.977**	0.038	0.943	0.003	0.953	0.004
Thyroid	0.874	0.000	**0.953**	0.029	0.917	0.035	0.754	0.007	0.778	0.009
Generated blobs with outliers										
2 clusters (0.5% outliers)	0.995	0.000	0.995	0.000	0.995	0.000	0.995	0.000	**1.000**	0.000
2 clusters (1% outliers)	0.989	0.000	0.990	0.000	0.990	0.000	0.990	0.000	**0.996**	0.000
2 clusters (2% outliers)	0.979	0.000	0.980	0.000	0.980	0.000	0.981	0.000	**0.997**	0.000
2 clusters (4% outliers)	0.961	0.000	0.962	0.000	0.962	0.000	0.964	0.000	**0.996**	0.000
3 clusters (0.5% outliers)	0.994	0.000	0.994	0.000	0.994	0.000	0.994	0.000	**0.999**	0.000
3 clusters (1% outliers)	0.989	0.000	0.989	0.000	0.989	0.000	0.989	0.000	**0.997**	0.000
3 clusters (2% outliers)	0.979	0.000	0.979	0.000	0.979	0.000	0.981	0.000	**0.997**	0.000
3 clusters (4% outliers)	0.961	0.000	0.951	0.000	0.945	0.000	0.965	0.000	**0.996**	0.000

Bold underlined values indicate best results for each dataset.

4. Discussion

Research focuses a lot on developing new density estimation procedures [17,18]. Moreover, in the last years, different scientists started to propose different robust density estimation methods even based on neural networks such as Parzen neural networks [19], soft constrained neural networks [20], and others [21]. This paper presented a new clustering method based on the modified inversion formula density estimation (MIDE). This new method improves the performance and robustness of the k-means, Gaussian mixture model, and other methods. Method working: Firstly, we initialized parameters and generated a T matrix. Secondly, we estimated the densities of each point and cluster based on the modified inversion formula. Third, we updated mean, sigma, and phi matrices. Based on the results presented earlier, it is possible to conclude that the newly presented method works well with different clustering datasets even if the datasets do not have any outliers. Results based on the generated clusters data with outliers showed that the newly presented method (MIDEv2) works the best in all situations (0.5%, 1%, 2%, and 4%). Based on the accuracy metric with all of these datasets, accuracy was higher than 0.995. The interesting point is that a new method based on the inversion formula can cluster the data even if data do not have outliers; one of the most popular, for example, is the Iris data set. When we compared the accuracy results in other datasets, it can be mentioned that the MIDE method achieved 0.955 accuracy on the Iris dataset compared with the second-best GMM

method with 0.953 accuracy; using the ARI metric for this dataset, MIDE methods as well showed better results compared with other methods. Based on the NMI, J-Score, and FMI metrics (see Table A1), a better method for the Iris dataset would be GMM. It is hard to compare and use multiple metrics because, in this research, we used accuracy as our main metric. After all, all datasets have labels, and it is possible to calculate accuracy of our clustering methods. Compared with other researchers' results in the past, Sun et al. were able to achieve 0.925 accuracy with the SVC-KM approach [44], and Hyde and Angelov achieved 0.950 accuracy with DDC (Data Density-Based Clustering) [45]. Additonally, it is notable that the MIDE method has a lower standard deviation than other methods used in this research. It is worth mentioning that this method also has limitations. Based on the experimental study, this one method in the current state can not work with higher dimensional data (d > 15). This occurs due to T matrix generation; as dimensions grow, finding a suitable T matrix becomes harder. This one will be solved in the future versions of the model; we will present more about it in future work. Another method problem is speed; the current stage method is slower than other methods, but this problem can be solved with parallelization of the process on the programming side. The future direction of the newly created method is this method application for deep clustering. It can be seen that MIDEv1 and MIDEv2 methods do not work very well with higher-dimension data. Due to that, the deep clustering method with an encoder structure could solve this problem.

Author Contributions: Conceptualization, T.R. and M.L.; methodology, T.R.; software, T.R. and M.L.; formal analysis, T.R. and M.L.; investigation, T.R. and M.L.; writing—original draft preparation, T.R., M.L.; writing—review and editing, M.L.; supervision, T.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank the area editor and the reviewers for giving valuable comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

J-score metric calculation program code with Python language

```
import numpy as np
def JScore(truth, pred):
    if (len(truth) == len(pred)):
        print("Equal lengths")
        A = np.empty([0, len(truth)], bool)
        test = list(set(pred))
        for i in test:
            A = np.vstack([A, (np.array(pred) == i)])
        suma = A.sum(axis=1)
        B = np.empty([0, len(truth)], bool)
        test = list(set(truth))
        for i in test:
            B = np.vstack([B, (np.array(truth) == i)])
        suma2 = B.sum(axis=1)
        C = np.empty([len(suma), len(suma2)], float)
        for i in range(0, len(suma)):
            for j in range(0, len(suma2)):
                C[i, j] = sum(A[i,] & B[j,])/sum(A[i,] | B[j,])
```

Table A2. Comparative analysis of different models (means and standard deviation) based on the adjusted Rand Index (ARI) for 10,000 runs.

Dataset	K-Means		GMM		BGMM		MIDEv1		MIDEv2	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
Synthetic										
Aggregation	0.725	0.008	0.795	0.069	0.860	0.089	0.687	0.035	**0.862**	0.023
Atom	0.176	0.003	0.058	0.028	0.076	0.024	0.204	0.006	**0.221**	0.004
D31	**0.949**	0.016	0.903	0.027	0.634	0.017	0.494	0.037	0.529	0.026
R15	**0.993**	0.000	0.975	0.036	0.608	0.020	0.747	0.021	0.786	0.018
Gaussians1	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	**1.000**	1.000
Threenorm	0.032	0.001	0.058	0.045	0.009	0.002	0.088	0.003	**0.089**	0.002
Twenty	1.000	0.000	0.986	0.028	0.836	0.096	1.000	0.000	**1.000**	0.000
Wingnut	0.670	0.000	0.862	0.001	**0.863**	0.000	0.565	0.007	0.533	0.005
Real										
Breast	0.664	0.008	0.772	0.003	0.747	0.003	-	-	-	-
CPU	0.529	0.014	0.315	0.070	0.336	0.081	0.461	0.043	**0.708**	0.026
Dermatology	0.712	0.038	0.697	0.096	0.728	0.112	-	-	-	-
Diabetes	0.058	0.003	0.059	0.046	0.079	0.028	0.059	0.005	**0.086**	0.002
Ecoli	0.505	0.008	0.649	0.011	**0.665**	0.014	0.551	0.013	0.423	0.015
Glass	0.162	0.014	0.178	0.055	0.211	0.040	0.151	0.024	**0.229**	0.011
Heart-statlog	0.451	0.005	0.352	0.072	0.344	0.075	0.422	0.013	**0.452**	0.011
Iono	0.168	0.000	0.383	0.066	0.368	0.049	-	-	-	-
Iris	0.617	0.009	0.888	0.077	0.654	0.030	0.819	0.029	**0.888**	0.008
Wine	0.897	0.000	0.869	0.072	**0.932**	0.063	0.835	0.031	0.865	0.012
Thyroid	0.583	0.000	**0.850**	0.075	0.735	0.074	0.297	0.045	0.356	0.015
Generated blobs with outliers										
2 clusters (0.5% outliers)	0.991	0.000	0.990	0.000	0.990	0.000	0.993	0.000	**1.000**	0.000
2 clusters (1% outliers)	0.976	0.000	0.980	0.000	0.980	0.000	0.980	0.000	**0.992**	0.000
2 clusters (2% outliers)	0.957	0.000	0.961	0.000	0.961	0.000	0.961	0.000	**0.989**	0.000
2 clusters (4% outliers)	0.920	0.000	0.924	0.000	0.924	0.000	0.928	0.000	**0.990**	0.000
3 clusters (0.5% outliers)	0.990	0.000	0.990	0.000	0.990	0.000	0.991	0.000	**0.997**	0.000
3 clusters (1% outliers)	0.982	0.000	0.982	0.000	0.982	0.000	0.984	0.000	**0.993**	0.000
3 clusters (2% outliers)	0.967	0.000	0.967	0.000	0.967	0.000	0.967	0.000	**0.993**	0.000
3 clusters (4% outliers)	0.938	0.000	0.925	0.000	0.918	0.000	0.941	0.000	**0.992**	0.000

Bold underlined values indicate best results for each dataset.

Table A3. Comparative table of different models (means and standard deviation) based on the J-Score for 10,000 runs.

Dataset	K-Means		GMM		BGMM		MIDEv1		MIDEv2	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
Synthetic										
Aggregation	0.780	0.007	0.800	0.071	0.870	0.062	0.831	0.009	**0.871**	0.012
Atom	0.556	0.002	0.501	0.004	0.503	0.005	0.575	0.004	**0.582**	0.004
D31	**0.951**	0.017	0.901	0.029	0.581	0.019	0.556	0.031	0.609	0.042
R15	**0.993**	0.000	0.975	0.038	0.664	0.011	0.756	0.041	0.834	0.027
Gaussians1	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	**1.000**	0.000
Threenorm	0.420	0.001	0.443	0.050	0.381	0.005	0.481	0.003	**0.496**	0.004
Twenty	1.000	0.000	0.984	0.030	0.838	0.075	**1.000**	0.002	0.986	0.005
Wingnut	0.834	0.000	0.931	0.001	**0.932**	0.000	0.779	0.000	0.808	0.000

```
        M1 = sum(np.amax(C, axis=1) * suma)/A.shape[1]
        M11 = sum(np.amax(C, axis=0) * suma2)/A.shape[1]
        M2 = 2 * M1 * M11/(M1 + M11)
        return M2
    else:
        print('Truth and Pred have different lengths.')
```

Appendix B

Table A1. Comparative table of different models (means and standard deviation) based on Normalized Mutual Information (NMI) for 10,000 runs.

Dataset	K-Means		GMM		BGMM		MIDEv1		MIDEv2	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
Synthetic										
Aggregation	0.836	0.004	0.886	0.035	**0.909**	0.041	0.779	0.006	0.845	0.005
Atom	0.289	0.003	0.170	0.036	0.194	0.028	0.310	0.004	**0.319**	0.003
D31	**0.969**	0.005	0.951	0.008	0.871	0.004	0.791	0.007	0.822	0.006
R15	**0.994**	0.000	0.989	0.012	0.868	0.014	0.881	0.001	0.909	0.001
Gaussians1	1.000	0.000	1.000	0.000	1.000	0.000	**1.000**	0.000	**1.000**	0.000
Threenorm	0.024	0.001	0.047	0.039	0.007	0.002	0.069	0.001	**0.076**	0.001
Twenty	**1.000**	0.000	0.996	0.008	0.956	0.026	**1.000**	0.000	0.988	0.005
Wingnut	0.562	0.000	0.778	0.002	**0.779**	0.000	0.459	0.000	0.420	0.001
Real										
Breast	0.547	0.011	0.659	0.003	0.630	0.003	-	-	-	-
CPU	0.487	0.013	0.398	0.025	0.389	0.033	0.467	0.013	**0.529**	0.011
Dermatology	0.862	0.009	0.809	0.044	0.862	0.049	-	-	-	-
Diabetes	0.090	0.004	0.084	0.041	0.105	0.017	0.089	0.004	**0.106**	0.003
Ecoli	0.636	0.004	0.636	0.016	**0.639**	0.010	0.592	0.004	0.534	0.004
Glass	0.303	0.019	0.327	0.052	0.364	0.042	0.304	0.020	**0.369**	0.024
Heart-statlog	**0.363**	0.005	0.270	0.055	0.263	0.058	0.339	0.008	0.308	0.007
Iono	0.125	0.000	0.305	0.052	0.299	0.024	-	-	-	-
Iris	0.657	0.006	**0.890**	0.04	0.751	0.011	0.841	0.007	0.763	0.008
Wine	**0.876**	0.000	0.856	0.055	0.926	0.054	0.822	0.001	0.799	0.003
Thyroid	0.559	0.000	**0.783**	0.059	0.661	0.051	0.382	0.009	0.390	0.008
Generated clusters with outliers										
2 clusters (0.5% outliers)	0.976	0.000	0.976	0.000	0.976	0.000	0.977	0.000	**1.000**	0.000
2 clusters (1% outliers)	0.947	0.000	0.957	0.000	0.957	0.000	0.958	0.000	**0.974**	0.000
2 clusters (2% outliers)	0.916	0.000	0.925	0.000	0.925	0.000	0.928	0.000	**0.976**	0.000
2 clusters (4% outliers)	0.867	0.000	0.876	0.000	0.876	0.000	0.886	0.000	**0.972**	0.000
3 clusters (0.5% outliers)	0.978	0.000	0.978	0.000	0.978	0.000	0.978	0.000	**0.993**	0.000
3 clusters (1% outliers)	0.964	0.000	0.964	0.000	0.964	0.000	0.964	0.000	**0.986**	0.000
3 clusters (2% outliers)	0.943	0.000	0.943	0.000	0.943	0.000	0.945	0.000	**0.985**	0.000
3 clusters (4% outliers)	0.907	0.000	0.901	0.000	0.898	0.000	0.911	0.000	**0.982**	0.000

Bold underlined values indicate best results for each dataset.

Table A3. *Cont.*

Dataset	K-Means		GMM		BGMM		MIDEv1		MIDEv2	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
Real										
Breast	0.833	0.004	0.887	0.001	0.874	0.002	-	-	-	-
CPU	0.656	0.013	0.489	0.058	0.500	0.077	0.733	0.011	**0.751**	0.010
Dermatology	0.719	0.038	0.699	0.079	0.730	0.106	-	-	-	-
Diabetes	0.252	0.004	0.283	0.033	0.299	0.028	0.275	0.008	**0.307**	0.004
Ecoli	0.557	0.009	0.655	0.018	0.663	0.006	0.606	0.008	**0.663**	0.007
Glass	0.340	0.010	0.362	0.036	0.365	0.032	0.397	0.012	**0.412**	0.009
Heart-statlog	0.720	0.003	0.663	0.043	0.659	0.045	0.714	0.005	**0.727**	0.004
Iono	0.549	0.000	0.686	0.018	0.673	0.031	-	-	-	-
Iris	0.730	0.008	**0.923**	0.064	0.752	0.029	0.889	0.012	0.905	0.009
Wine	0.935	0.000	0.917	0.052	**0.958**	0.046	0.904	0.012	0.917	0.011
Thyroid	0.787	0.000	**0.914**	0.035	0.856	0.038	0.639	0.007	0.675	0.008
Generated blobs with outliers										
2 clusters (0.5% outliers)	0.993	0.000	0.993	0.000	0.993	0.000	0.993	0.000	**1.000**	0.000
2 clusters (1% outliers)	0.983	0.000	0.985	0.000	0.985	0.000	0.985	0.000	**0.991**	0.000
2 clusters (2% outliers)	0.969	0.000	0.971	0.000	0.971	0.000	0.972	0.000	**0.994**	0.000
2 clusters (4% outliers)	0.942	0.000	0.944	0.000	0.944	0.000	0.946	0.000	**0.996**	0.000
3 clusters (0.5% outliers)	0.991	0.000	0.991	0.000	0.991	0.000	0.993	0.000	**0.998**	0.000
3 clusters (1% outliers)	0.983	0.000	0.983	0.000	0.983	0.000	0.985	0.000	**0.995**	0.000
3 clusters (2% outliers)	0.969	0.000	0.969	0.000	0.969	0.000	0.972	0.000	**0.994**	0.000
3 clusters (4% outliers)	0.941	0.000	0.932	0.000	0.927	0.000	0.945	0.000	**0.993**	0.000

Bold underlined values indicate best results for each dataset.

Table A4. Different models were compared (means and standard deviation) based on the Fowlkes–Mallows index (FMI) for 10,000 runs.

Dataset	K-Means		GMM		BGMM		MIDEv1		MIDEv2	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
Synthetic										
Aggregation	0.785	0.006	0.840	0.055	**0.891**	0.070	0.875	0.011	**0.867**	0.015
Atom	0.654	0.001	0.653	0.006	0.649	0.003	0.659	0.002	**0.669**	0.003
D31	**0.951**	0.015	0.906	0.025	0.681	0.012	0.645	0.011	0.689	0.016
R15	**0.993**	0.000	0.977	0.033	0.682	0.016	0.779	0.011	0.817	0.009
Gaussians1	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	**1.000**	0.000
Threenorm	0.518	0.000	0.535	0.030	0.514	0.002	0.552	0.002	**0.559**	0.003
Twenty	1.000	0.000	0.987	0.026	0.857	0.075	**1.000**	0.000	0.984	0.004
Wingnut	0.835	0.000	0.931	0.001	**0.932**	0.000	0.792	0.001	0.764	0.001
Real										
Breast	0.847	0.004	0.893	0.001	0.881	0.001	-	-	-	-
CPU	0.771	0.006	0.619	0.052	0.633	0.065	0.802	0.012	**0.871**	0.009
Dermatology	0.769	0.030	0.760	0.074	0.784	0.087	-	-	-	-
Diabetes	0.326	0.002	0.382	0.017	0.378	0.028	0.375	0.008	**0.389**	0.007
Ecoli	0.625	0.006	0.740	0.008	**0.762**	0.009	0.678	0.006	0.698	0.006
Glass	0.393	0.012	0.435	0.058	0.437	0.048	**0.540**	0.021	0.519	0.015
Heart-statlog	0.734	0.002	0.683	0.026	0.679	0.028	0.724	0.011	**0.737**	0.009
Iono	0.601	0.000	0.711	0.004	0.698	0.023	-	-	-	-
Iris	0.743	0.006	**0.927**	0.041	0.781	0.011	0.899	0.005	0.877	0.005
Wine	0.932	0.000	0.914	0.042	0.955	0.038	0.895	0.011	0.886	0.008
Thyroid	0.841	0.000	0.931	0.023	0.888	0.022	0.705	0.013	0.736	0.009

Table A4. *Cont.*

Dataset	K-Means		GMM		BGMM		MIDEv1		MIDEv2	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
Generated blobs with outliers										
2 clusters (0.5% outliers)	0.995	0.000	0.995	0.000	0.995	0.000	0.996	0.000	1.000	0.000
2 clusters (1% outliers)	0.988	0.000	0.990	0.000	0.990	0.000	0.990	0.000	0.994	0.000
2 clusters (2% outliers)	0.978	0.000	0.980	0.000	0.980	0.000	0.981	0.000	0.996	0.000
2 clusters (4% outliers)	0.960	0.000	0.961	0.000	0.951	0.000	0.963	0.000	0.995	0.000
3 clusters (0.5% outliers)	0.993	0.000	0.993	0.000	0.993	0.000	0.993	0.000	0.998	0.000
3 clusters (1% outliers)	0.988	0.000	0.988	0.000	0.988	0.000	0.991	0.000	0.996	0.000
3 clusters (2% outliers)	0.978	0.000	0.978	0.000	0.978	0.000	0.981	0.000	0.996	0.000
3 clusters (4% outliers)	0.959	0.000	0.951	0.000	0.948	0.000	0.964	0.000	0.995	0.000

Bold underlined values indicate best results for each dataset.

References

1. Ding, S.; Jia, H.; Du, M.; Xue, Y. A semi-supervised approximate spectral clustering algorithm based on HMRF model. *Inf. Sci.* **2018**, *429*, 215–228. [CrossRef]
2. Liu, A.-A.; Nie, W.-Z.; Gao, Y.; Su, Y.-T. View-based 3-D model retrieval: A benchmark. *IEEE Trans. Cybern.* **2017**, *48*, 916–928. [CrossRef] [PubMed]
3. Nie, W.; Cheng, H.; Su, Y. Modeling temporal information of mitotic for mitotic event detection. *IEEE Trans. Big Data* **2017**, *3*, 458–469. [CrossRef]
4. Karim, M.R.; Beyan, O.; Zappa, A.; Costa, I.G.; Rebholz-Schuhmann, D.; Cochez, M.; Decker, S. Deep learning-based clustering approaches for bioinformatics. *Brief. Bioinform.* **2021**, *22*, 393–415. [CrossRef]
5. Kim, T.; Chen, I.R.; Lin, Y.; Wang, A.Y.Y.; Yang, J.Y.H.; Yang, P. Impact of similarity metrics on single-cell RNA-seq data clustering. *Brief. Bioinform.* **2019**, *20*, 2316–2326. [CrossRef] [PubMed]
6. Govender, P.; Sivakumar, V. Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019). *Atmos. Pollut. Res.* **2020**, *11*, 40–56. [CrossRef]
7. Xu, S.; Yang, X.; Yu, H.; Yu, D.-J.; Yang, J.; Tsang, E.C. Multi-label learning with label-specific feature reduction. *Knowl. -Based Syst.* **2016**, *104*, 52–61. [CrossRef]
8. Liu, K.; Yang, X.; Yu, H.; Mi, J.; Wang, P.; Chen, X. Rough set based semi-supervised feature selection via ensemble selector. *Knowl. -Based Syst.* **2019**, *165*, 282–296. [CrossRef]
9. Wiwie, C.; Baumbach, J.; Röttger, R. Comparing the performance of biomedical clustering methods. *Nat. Methods* **2015**, *12*, 1033–1038. [CrossRef] [PubMed]
10. Chen, C.-H. A hybrid intelligent model of analyzing clinical breast cancer data using clustering techniques with feature selection. *Appl. Soft Comput.* **2014**, *20*, 4–14. [CrossRef]
11. Polat, K. Classification of Parkinson's disease using feature weighting method on the basis of fuzzy C-means clustering. *Int. J. Syst. Sci.* **2012**, *43*, 597–609. [CrossRef]
12. Nilashi, M.; Ibrahim, O.; Ahani, A. Accuracy improvement for predicting Parkinson's disease progression. *Sci. Rep.* **2016**, *6*, 1–18. [CrossRef] [PubMed]
13. Trevithick, L.; Painter, J.; Keown, P. Mental health clustering and diagnosis in psychiatric in-patients. *BJPsych Bull.* **2015**, *39*, 119–123. [CrossRef] [PubMed]
14. Yilmaz, N.; Inan, O.; Uzer, M.S. A new data preparation method based on clustering algorithms for diagnosis systems of heart and diabetes diseases. *J. Med. Syst.* **2014**, *38*, 48–59. [CrossRef]
15. Alashwal, H.; El Halaby, M.; Crouse, J.J.; Abdalla, A.; Moustafa, A.A. The application of unsupervised clustering methods to Alzheimer's disease. *Front. Comput. Neurosci.* **2019**, *13*, 31. [CrossRef]
16. Farouk, Y.; Rady, S. Early diagnosis of alzheimer's disease using unsupervised clustering. *Int. J. Intell. Comput. Inf. Sci.* **2020**, *20*, 112–124. [CrossRef]
17. Li, D.; Yang, K.; Wong, W.H. Density estimation via discrepancy based adaptive sequential partition. *Adv. Neural Inf. Process. Syst.* **2016**, *29*.
18. Rothfuss, J.; Ferreira, F.; Walther, S.; Ulrich, M. Conditional density estimation with neural networks: Best practices and benchmarks. *arXiv* **2019**, arXiv:1903.00954.
19. Trentin, E.; Lusnig, L.; Cavalli, F. Parzen neural networks: Fundamentals, properties, and an application to forensic anthropology. *Neural Netw.* **2018**, *97*, 137–151. [CrossRef]
20. Trentin, E. Soft-constrained neural networks for nonparametric density estimation. *Neural Process. Lett.* **2018**, *48*, 915–932. [CrossRef]
21. Huynh, H.T.; Nguyen, L. Nonparametric maximum likelihood estimation using neural networks. *Pattern Recognit. Lett.* **2020**, *138*, 580–586. [CrossRef]

22. Ruzgas, T.; Lukauskas, M.; Čepkauskas, G. Nonparametric Multivariate Density Estimation: Case Study of Cauchy Mixture Model. *Mathematics* **2021**, *9*, 2717. [CrossRef]
23. Biernacki, C.; Celeux, G.; Govaert, G. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Comput. Stat. Data Anal.* **2003**, *41*, 561–575. [CrossRef]
24. Xu, Q.; Yuan, S.; Huang, T. Multi-dimensional uniform initialization Gaussian mixture model for spar crack quantification under uncertainty. *Sensors* **2021**, *21*, 1283. [CrossRef]
25. Fraley, C. Algorithms for model-based Gaussian hierarchical clustering. *SIAM J. Sci. Comput.* **1998**, *20*, 270–281. [CrossRef]
26. Maitra, R. Initializing partition-optimization algorithms. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2009**, *6*, 144–157. [CrossRef]
27. Meila, M.; Heckerman, D. An experimental comparison of several clustering and initialization methods. *arXiv* **2013**, arXiv:1301.7401.
28. Hasselblad, V. Estimation of parameters for a mixture of normal distributions. *Technometrics* **1966**, *8*, 431–444. [CrossRef]
29. Behboodian, J. On a mixture of normal distributions. *Biometrika* **1970**, *57*, 215–217. [CrossRef]
30. Ćwik, J.; Koronacki, J. Multivariate density estimation: A comparative study. *Neural Comput. Appl.* **1997**, *6*, 173–185. [CrossRef]
31. Tsuda, K.; Akaho, S.; Asai, K. The em algorithm for kernel matrix completion with auxiliary data. *J. Mach. Learn. Res.* **2003**, *4*, 67–81.
32. Lartigue, T.; Durrleman, S.; Allassonnière, S. Deterministic approximate EM algorithm; Application to the Riemann approximation EM and the tempered EM. *Algorithms* **2022**, *15*, 78. [CrossRef]
33. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B* **1977**, *39*, 1–22.
34. Everitt, B. *Finite Mixture Distributions*; Springer: Berlin/Heidelberg, Germany, 2013.
35. Redner, R.A.; Walker, H.F. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Rev.* **1984**, *26*, 195–239. [CrossRef]
36. Xie, C.-H.; Chang, J.-Y.; Liu, Y.-J. Estimating the number of components in Gaussian mixture models adaptively for medical image. *Optik* **2013**, *124*, 6216–6221. [CrossRef]
37. Ahmadinejad, N.; Liu, L. J-Score: A Robust Measure of Clustering Accuracy. *arXiv* **2021**, arXiv:2109.01306.
38. Zhong, S.; Ghosh, J. Generative model-based document clustering: A comparative study. *Knowl. Inf. Syst.* **2005**, *8*, 374–384. [CrossRef]
39. Lawrence, H.; Phipps, A. Comparing partitions. *J. Classif.* **1985**, *2*, 193–218.
40. Wang, P.; Shi, H.; Yang, X.; Mi, J. Three-way k-means: Integrating k-means and three-way decision. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 2767–2777. [CrossRef]
41. Fowlkes, E.B.; Mallows, C.L. A method for comparing two hierarchical clusterings. *J. Am. Stat. Assoc.* **1983**, *78*, 553–569. [CrossRef]
42. Caliński, T.; Harabasz, J. A dendrite method for cluster analysis. *Commun. Stat. -Theory Methods* **1974**, *3*, 1–27. [CrossRef]
43. Davies, D.L.; Bouldin, D.W. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, 224–227. [CrossRef]
44. Sun, Y.; Wang, Y.; Wang, J.; Du, W.; Zhou, C. A novel SVC method based on K-means. In Proceedings of the 2008 Second International Conference on Future Generation Communication and Networking, Hainan, China, 13–15 December 2008; pp. 55–58.
45. Hyde, R.; Angelov, P. Data density based clustering. In Proceedings of the 2014 14th UK Workshop on Computational Intelligence (UKCI), Bradford, UK, 8–10 September 2014; pp. 1–7.

mathematics

MDPI

Article

Reduced Clustering Method Based on the Inversion Formula Density Estimation

Mantas Lukauskas * and Tomas Ruzgas

Department of Applied Mathematics, Faculty of Mathematics and Natural Sciences, Kaunas University of Technology, 44249 Kaunas, Lithuania
* Correspondence: mantas.lukauskas@ktu.lt

Abstract: Unsupervised learning is one type of machine learning with an exceptionally high number of applications in various fields. The most popular and best-known group of unsupervised machine learning methods is clustering methods. The main goal of clustering is to find hidden relationships between individual observations. There is great interest in different density estimation methods, especially when there are outliers in the data. Density estimation also can be applied to data clustering methods. This paper presents the extension to the clustering method based on the modified inversion formula density estimation to solve previous method limitations. This new method's extension works within higher dimensions (d > 15) cases, which was the limitation of the previous method. More than 20 data sets are used in comparative data analysis to prove the effectiveness of the developed method improvement. The results showed that the new method extension positively affects the data clustering results. The new reduced clustering method, based on the modified inversion formula density estimation, outperforms popular data clustering methods on test data sets. In cases when the accuracy is not the best, the data clustering accuracy is close to the best models' obtained accuracies. Lower dimensionality data were used to compare the standard clustering based on the inversion formula density estimation method with the extended method. The new modification method has better results than the standard method in all cases, which confirmed the hypothesis about the new method's positive impact on clustering results.

Keywords: nonparametric density estimation; unsupervised machine learning; clustering; inversion formula; dimensions reduction

MSC: 62G05; 62G07; 62G30

Citation: Lukauskas, M.; Ruzgas, T. Reduced Clustering Method Based on the Inversion Formula Density Estimation. *Mathematics* **2023**, *11*, 661. https://doi.org/10.3390/math11030661

Academic Editor: José Antonio Roldán-Nofuentes

Received: 29 December 2022
Revised: 23 January 2023
Accepted: 25 January 2023
Published: 28 January 2023

1. Introduction

Scientists first mentioned artificial intelligence long ago, but in the last decade, it has gained immense popularity, and it can now be used in all areas of life. Machine learning is the most widely used group of mathematical methods, often called artificial intelligence methods, to gain more popularity. Researchers classify machine learning into various types, such as supervised, semi-supervised, reinforcement, and unsupervised learning. This paper focuses explicitly on unsupervised learning methods. Unsupervised machine learning is one type of machine learning with an exceptionally high application in various fields. The most popular and best-known group of unsupervised machine learning methods is clustering methods used in cluster analysis. In 1932, Driver and Kroeber first mentioned cluster analysis, which noticeably was mentioned earlier than artificial intelligence. The main goal of cluster analysis is to find hidden relationships between individual observations. These observations can be the company's customers, products, and goods, as well as doctors' patients and other data. The main aim is to divide observations into groups that the researcher does not immediately know. Data clustering has applications in different fields: healthcare for detection of breast cancer [1], Alzheimer's

disease [2,3], and others. Clustering can also be applied in pattern recognition [4,5] and big data text clustering [6,7]. Clustering methods often use different distance measures to determine how close individual observations are to each other and determine the best clustering. In current studies, the k-means clustering method is the most used, but there are also many different clustering methods. Various clustering methods are used to solve various practical problems. However, although the choice of these methods is wide, this problem remains challenging and new methods that can solve clustering problems are constantly being sought. It can also be noted that much attention is paid to developing various density estimation methods that can be used to solve the problem of data clustering. There is also an ongoing search for ways to estimate density if there are outliers in the data. Various methods based on neural targets have been proposed to solve this problem: soft-constrained neural networks [8] and Parzen neural networks [8]. This work is also based on data clustering based on density estimation. This research paper is a follow-up work of a more extensive study on developing a new data clustering method [9]. A previous research paper observed that the newly developed clustering method based on the modified inversion formula density estimation (CBMIDE) performs poorly when the data dimensions are higher ($d > 10$). In this paper, as a novelty, we suggest modification of the CBMIDE clustering method for higher-dimension data to allow the method to work in higher dimensions; moreover, to increase the previous method's accuracy in the lower-dimensionality case. For this reason, this research paper examines the impact of data dimensionality reduction modification on the method. Data dimensions are essential for the accuracy of clustering methods, calculation time, and using different computing resources. In a typical case, it is observed that with increasing dimension, the time of data clustering also increases significantly and can even increase exponentially. In such a case, one solution is to reduce the dimensionality of the data. It can be done using different dimensionality reduction methods. Combining data dimensionality reduction with clustering is quite common because it saves many resources. Data dimensionality reduction and the application of reduced dimensions in clustering are also discussed in the scientific literature, where the simplest k-means and principal component analysis (PCA) methods were first combined [10]. These studies are popular even now. The main reason for this is the increasing amount of data every year. A relatively popular field of combining these methods is gene analysis. Data dimensionalities reduction methods such as principal component analysis (PCA) [11], non-negative matrix factorization (NMF) [12], independent component analysis (ICA) [13] and clustering methods such as k-means, density-based spatial clustering of applications with noise (DBSCAN), and others used to study gene sequences [14]. Furthermore, more complex methods are noticed to reduce the dimensions of the data, such as t-distributed stochastic neighbour embedding (t-SNE) [15], uniform manifold approximation and projection (UMAP) [16], and different combinations of methods [17–22].

In this paper, we modify the CBMIDE method and test our hypothesis about the impact of dimensionality reduction on clustering results. The paper hypothesizes that the accuracy of data clustering can be maintained by reducing data dimensions using data dimensionality reduction methods and that the new RCBMIDE method has the advantage compared with an earlier method and other popular methods. This paper uses a much larger number of data dimensionality reduction methods and clustering methods to compare results.

This paper is organized as follows. In the second section of this paper, we present clustering based on the modified inversion formula density estimation and a reduced version of this method. The third section of the paper contains information about the methodology, methods used in the comparative analysis, data sets, and others. The fourth section of the paper reviews the results obtained from the research study. Finally, the last section of this paper presents a discussion of research and research results, conclusions, and future work.

2. Clustering Based on the Density of the Modified Inversion Formula

Data clustering is possible based on density estimation functions. In statistical modelling, estimating probability density functions (pdf) is one of the most critical parts. Probability density functions make it possible to describe random variables as hidden functions of other variables and simultaneously identify hidden relationships, which can be used to create data groups. For example, if we claim that the random vector $X \in \mathbf{R}^d$ satisfies the distribution mixture model if the distribution density $f(x)$ satisfies Equation (1).

$$f(x) = \sum_{k=1}^{q} p_k f_k(x) = f(x, \theta). \tag{1}$$

In the given formula (1), $f_k(x)$—distribution density, θ—multivariate model parameter, and p_k—the probability of the k-th element. It is worth noting that when evaluating a mixture of distributions, the number of clusters q and a priori probability p_k is used, and the following conditions must be met:

$$p_k > 0, \sum_{k=1}^{q} p_k = 1. \tag{2}$$

Suppose X is a d-dimensional random vector with a distribution density f(x), and there is a sample of independent copies of X, where $X = (X(1), \ldots, X(n))$. We will say that the sample satisfies the mixture model if X(t) satisfies (1). We will call size n the sample size (volume). When examining the approximations of parametric methods, it should be emphasized that as the data dimension increases, the number of model parameters grows rapidly, which makes it more difficult to find accurate parameter estimates. One-dimensional data projections $X_\tau = \tau' X$ density f_τ is much easier to find than multidimensional data density f because a mutually unambiguous correspondence exists, $f \leftrightarrow \left\{ f_\tau,\ \tau \in R^d \right\}$. The projection density f_τ of one-dimensional data is much easier to find than the density f of multidimensional data. If the distributions used are Gaussian, then the one-dimensional Gaussian mixture model can be described by the following formula.

$$f_\tau(x) = \sum_{k=1}^{q} p_{k,\tau} \varphi_{k,\tau}(x) = f_\tau(x, \theta_\tau), \tag{3}$$

here $\varphi_{k,\tau}(x) = \varphi\left(x;\ m_{k,\tau},\ \sigma^2_{k,\tau}\right)$—Gaussian density and multidimensional parameter depends on the given parameters $\theta_\tau = \left(p_{k,\tau},\ m_{k,\tau},\ \sigma^2_{k,\tau}\right)$, $k = 1, \ldots, q$. The following equations relate these parameters $p_{j,\tau} = p_j$, $m_{j,\tau} = \tau' M_j$, and $\sigma^2_{j,\,\tau} = \tau' R_j \tau$. Then we can use the inversion formula (4) given below.

$$f(x) = \frac{1}{(2\pi)^d} \int_{\mathbf{R}^d} e^{-it'x} \psi(t) dt, \tag{4}$$

where $\psi(t) = \mathrm{E} e^{it'x}$ denotes the characteristic function of the random variable X. To use the inversion formula, first of all, we select the number of design directions, which must be evenly distributed on the unit sphere, the set T, and by changing the characteristic function by its estimate, we therefore obtain the estimating formula for the density calculation [23,24]:

$$\hat{f}(x) = \frac{A(d)}{\#T} \sum_{\tau \in T} \int_0^\infty e^{-iu\tau' x} \hat{\psi}_\tau(u) u^{d-1} e^{-hu^2} du, \tag{5}$$

where # denotes the number of elements in the array. It is possible to calculate the constant A(d), which depends on the dimension of the data, using the formula for the volume of a d-dimensional sphere.

$$A(d) = \frac{(V_d(1))_R^{'}}{(2\pi)^d} = \frac{d2^{-d}\pi^{-\frac{d}{2}}}{\Gamma\left(\frac{d}{2}+1\right)}. \quad (6)$$

Formula (5) can be used for various estimates of the characteristic function of the projected data. One of the disadvantages of the method using the inversion formula defined in (5) is that the mixture model of Gaussian distributions described by this estimate (when $f_k = \varphi_k$) only estimates the density well of observations of the distribution close to it. The density estimation using a mixture of Gaussian distributions can become complicated when approximating the density under study due to many components with low a priori probabilities. To overcome this, we can introduce a noise cluster and use a modified algorithm based on a multidimensional Gaussian distribution mixture model. One such algorithm is the inversion formula (4). Using this formula, we can calculate the parametric estimate of the characteristic function of uniform distribution density.

$$\hat{\psi}(u) = \frac{2}{(b-a)u}\sin\frac{(b-a)u}{2}\cdot e^{\frac{iu(a+b)}{2}}. \quad (7)$$

The uniform distribution density function (7) defines b as the maximum value and a as the minimum value. In the density estimate calculation formula (5), we construct the estimation of the characteristic function by combining the characteristic functions of a mixture of Gaussian distributions and a uniform distribution, using the corresponding a priori probabilities. Then the calculated continuous characteristic function presented earlier and the characteristic function of Gaussian distributions are used to form the density estimate. A priori probabilities are used to calculate the density estimate, which allows for controlling the influence of outliers on the density estimate. In formula (7), the second component of the density estimate is evaluated as a noise cluster whose probability/weight is $\hat{p}_0$.

$$\hat{\psi}_\tau(u) = \sum_{k=1}^{\hat{q}_\tau} \hat{p}_{k,\tau} e^{iu\hat{m}_{k,\tau} - u^2\hat{\sigma}^2_{k,\tau}/2} + \hat{p}_{0,\tau}\frac{2}{(b-a)u}\sin\frac{(b-a)u}{2}\cdot e^{\frac{iu(a+b)}{2}}, \quad (8)$$

The EM algorithm is used to apply this density estimate to data clustering. The expectation-maximization (EM) algorithm can be used to estimate the parameters of a mixture model, which is a probabilistic model for representing a dataset as a mixture of multiple distributions. A mixture model is defined by a set of parameters, including the individual components' means, variances, and mixing proportions. E-step (expectation step): For each data point, the algorithm calculates the probability that the point belongs to each component in the mixture based on the current estimates of the model parameters. Model parameters estimation is done by computing the likelihood of the data point given each component and multiplying it by the mixing proportion of that component. M-step (maximization step): In this step, the algorithm updates the model parameters to maximize the expected likelihood of the data based on the probabilities computed in the E-step. Specifically, the components' means, variances, and mixing proportions are updated to maximize the likelihood of the data under the current assignment of data points to components. The algorithm alternates between the E-step and the M-step until convergence is reached, at which point the estimated parameters are considered the maximum likelihood estimates. In the first step of the clustering algorithm, initial parameters selection using the initialization of k-means is performed. In the scientific literature, the random selection of parameters is used quite often [25,26]. However, this parameter selection showed that, in this case, the clustering method has worse stability. Furthermore, another way to choose the initial parameters is to use hierarchical clustering to combine the data [27]. However, this parameter initialization also failed in the case of the method being created.

For this reason, it was decided to use a more stable initialization of k-means parameters, which is used for other density-based clustering methods, such as the Gaussian mixture model and the Bayesian Gaussian mixture model (in package scikit-learn). The EM algorithm is famous in scientific works [28–30]. Based on the EM algorithm, the parameters of the data clustering method calculated after the number of r cycles are $\hat{\pi}_k = \hat{\pi}_k^{(r)}$. Then the new estimate of the multivariate parameter is $\hat{\theta} = \hat{\theta}^{(r+1)}$, whose individual components (probability, mean matrix, and covariance matrix, respectively) are calculated according to the formulas below.

$$\hat{p}_k = \frac{1}{n}\sum_{t=1}^{n} \hat{\pi}_k(X(t)) \tag{9}$$

$$\hat{M}(k) = \frac{1}{n\hat{p}_k}\sum_{t=1}^{n} \hat{\pi}_k(X(t)) \cdot X(t) \tag{10}$$

$$\hat{R}(k) = \frac{1}{n\hat{p}_k}\sum_{t=1}^{n} \hat{\pi}_k(X(t)) \left[X(t) - \hat{M}(k)\right] \cdot \left[X(t) - \hat{M}(k)\right]' \tag{11}$$

where $k = 1, \ldots, q$. Using the estimate of θ, the estimates of the probabilities π_k are obtained by replacing the unknown parameters with their statistical estimates from the formula

$$\pi_k(x) = \frac{p_k f_k(x)}{f(x,\theta)} \text{ and } k = \overline{1,q} \tag{12}$$

Suppose that the distribution of X depends on a random variable v that takes on values $1, \ldots, q$ with corresponding probabilities $p_1, \ldots, p_q$. In classification theory, v is interpreted as the number of the class to which the observed object belongs. Thus, observations $X(t)$ would correspond to $v(t)$, $t = 1, \ldots, n$. The functions f_k are treated as the conditional distribution density of X under the condition $v = k$. According to this approach, loose clustering of the sample is understood as posterior probabilities

$$\pi_k(x) = \mathbf{P}\{v = k \mid X = x\} \tag{13}$$

evaluation when all $x \in \{X(1), \ldots, X(n)\}$. Strict sample clustering would be the evaluation of random variables $v(1), \ldots, v(n)$, i.e., the sample is divided into subsets based on equality

$$\hat{v}(t) = arg\max_{k=1,\ldots,q} \hat{\pi}_k(X(t)) \tag{14}$$

The estimates $\hat{\pi}_k$ are obtained by approximating the unknown components of the distribution density with density estimates from the inversion formula and using the EM (expectation maximization) algorithm.

The following Algorithm 1 can describe the generalized clustering of data based on modified inversion formula density estimation (CBMIDE).

Algorithm 1: CBMIDE clustering algorithm

Input: Data set $X = [X_1, X_2, \ldots, X_n]$, cluster number K
Output: $C_1, C_2, \_\_, C_t$
Initiation of the mean vector using the k-means method
Generate a T matrix. The set T calculation where design directions are evenly spaced on the sphere.
1 For i = 1: t do
2 Density estimation for each point and cluster with the formula (8)
Update $\hat{M}$, $\hat{p}_k$, $\hat{R}$
3 End
4 Return $C_1, C_2, \_\_, C_t$ and $\hat{M}$, $\hat{p}_k$, $\hat{R}$

The biggest drawback of the earlier proposed method is its use for large-dimensional data. To use the inversion formula, first of all, we select the number of the design direction, which must be evenly distributed on the unit sphere, the set T. As the dimensions increase, finding design directions on a unit sphere becomes difficult. Therefore, it is not easy to generate the required matrix T. For this reason, an extension of the CBMIDE method is needed, allowing the use of this method for large-dimensional data. The data dimensionality reduction applied in the first clustering step is an extension of the method proposed in this paper.

$$Z = f(X,\ d) \tag{15}$$

where Z—reduced dimensions data in latent space, X—initial data in the original space, and d—number of dimensions. Further clustering of the data is then performed using the reduced dimensionality data. Dimensional data reduction allows for the separation of redundant characteristics of the observations, reducing the problem of multicollinearity in the data. Likewise, data dimensionality reduction allows easier data visualization when dimensions are reduced to two or three dimensions that can be represented visually. Data dimensionality reduction can be performed using feature selection or methods that form new dimensions that best represent the original data. Multiple data reduction techniques were implemented into the new algorithm to extend the original clustering method based on the inversion formula algorithm. To analyse and compare the results of these methods: principal component analysis (PCA), independent component analysis (ICA), factor analysis (FA), T-distributed stochastic neighbour embedding (t-SNE), TriMap, uniform manifold approximation and projection (UMAP), ISOMAP, multidimensional scaling (MDS), and locally linear embedding (LLE) were used. Different variants are used to determine which of the modifications has the best clustering results by modifying the general data clustering method based on the modified density estimation of the inversion formula. In the first modification of the algorithm, we use principal component analysis to reduce data dimensions into desired latent space. One of the most widely used data dimensionality reduction methods is principal component analysis (PCA) [31].

Let us say we have a multidimensional data matrix X, then we aim to calculate the correlation matrix R, and then the covariance matrix C is also created. If the individual variables are not covariate, then their covariance coefficient equals 0. To find the main components, the covariance matrix's real vectors E_k and real values λ_k are found, which are in the following equation

$$CE_k = \lambda_k E_k \tag{16}$$

solutions. In the above equation, E_k is the column, C is the previously defined covariance matrix, and the real vector λ_k is found in the characteristic equation $|C - \lambda_k I| = 0$, where I is the identity matrix.

After sorting the eigenvectors E_k according to the values of the corresponding true values in descending order, a matrix of principal components is formed as $A = (E_1, E_2, \ldots, E_n)$. Then the transformation of the data into the latent space is performed according to the formula:

$$Z_i = \left(X_i - \overline{X}\right) A_d \tag{17}$$

where Z_i—column of latent space, X_i—column of the original space, d—selected number of dimensions, and i—column number.

The second modification to improve the algorithm is the application of independent components for dimensionality reduction. The main difference between principal component analysis (PCA) and independent component analysis (ICA) is that PCA relies on uncorrelated factors while ICA relies on independent [32].

The third modification used in this paper is factor analysis. Factor analysis is a method that considers the correlations of variables, and the aim is to find latent variables that describe the original variables [33]. Suppose that there are k variables $X_1 \ldots X_{k,}$ and we seek to determine the factors that describe these variables, the number of which is m, then, the mathematical model of factor analysis can be summarized as

$$X_k = \lambda_{k1}F_1 + \lambda_{k2}F_2 + \ldots + \lambda_{km}F_m + \varepsilon_k \tag{18}$$

Multipliers λ_i are called factor weights. The factor analysis problem is the inverse of the linear regression problem, i.e., we know X_k values, and we want to find out what can be said about the common factors F_m.

The fourth modification of the CBMIDE method is to reduce the dimensionality of the t-SNE data by constructing latent dimensions. T-distributed stochastic neighbour embedding (t-SNE) is a method that is often intended for the visualization of multidimensional data after reducing this data to a more convenient two-dimensional or three-dimensional space [34]. This method is performed using two main steps. In the first step, a probabilistic calculation is performed. If we have X as our dataset, we try to compute probabilities p_{ij} based on the formula

$$p_{i|j} = \frac{\exp(-||x_i - x_j||_2/2\sigma_i^2)}{\sum_{k \neq i}\exp(-||x_i - x_k||_2/2\sigma_i^2)} \tag{19}$$

One of the most important things is that $p_{i|i} = 0$ and $\sum\limits_{i,j} p_{i,j} = 1$. Then, in the second step, Kullback–Leibler error (KL) optimization is performed, thus aiming to determine the exact locations of the observations in a smaller space. T-SNE aims to create d-dimensional data, so the similarity between two points located in the reduced dimension z_i and z_j, is calculated, which can be written in the given formula

$$q_{i,j} = \frac{\left(1 + ||z_i - z_j||^2\right)^{-1}}{\sum_k \sum_{l \neq k}\left(1 + ||z_k - z_l||^2\right)^{-1}} \tag{20}$$

Here, the Cauchy distribution is used to determine the similarity between observations. Then, to find the most suitable reduced data matrix, the previously mentioned Kullback–Leibler error (KL) is used, which can be defined as:

$$KL(P||Q) = \sum_{i \neq j} p_{i,j} log \frac{p_{i,j}}{q_{i,j}} \tag{21}$$

The gradient descent is used to do this. The Euclidean distance is mainly used in this method; therefore, this distance is also used in this paper. The T-SNE data reduction method is used quite often, so the inclusion of this method in the study is strongly justified [35–37]. The perplexity parameter of the t-SNE method used in the study is 30, the distance is Euclidean, and the method used is Barnes–Hut.

The fifth, sixth, and seventh modifications of the method are similar to the previously discussed modification of the t-SNE method since methods of the same family perform data dimensionality reduction. UMAP (uniform manifold approximation and projection) is a data dimensionality reduction technique such as t-SNE. Embedding is found by finding a low-dimensional data projection with the closest possible equivalent fuzzy topological structure. TriMap is a dimensionality reduction method that uses a triple constraint to form an embedding of a low-dimensional set of points. TriMap provides a significantly better global representation of data than other dimensionality reduction methods such as t-SNE, LargeVis, and UMAP [38]. The ISOMAP method calculates a quasi-isometric, low-dimensional embedding that best represents the original data set.

The eighth and ninth modifications of the method are performed using the MDS and LLE methods. The MDS method uses similarity or dissimilarity matrices to identify the nearest neighbours. Given a proximity matrix with proximities between each pair of objects in a set and a selected number of dimensions N, the MDS algorithm places each object in N-dimensional space (lower-dimensional representation) so that the distances between objects are preserved as best as possible. Locally linear embedding (LLE) tries to reduce these dimensions by trying to preserve the geometric features of the original non-linear structure [39]. The LLE first finds the k-nearest neighbours of points. Then it approximates each data vector as a weighted linear combination of k nearest neighbours. Finally, it calculates the weights that best restore the vectors from the neighbours and then produces the low-dimensional vectors best restored by these weights. One of the advantages of the LLE algorithm is that only one parameter needs to be tuned, which is the value of k or the number of nearest neighbours considered part of a cluster. Furthermore, the following methods are used as further relevant modifications: Spectral embedding and kernel PCA. Spectral embedding is a technique used in machine learning and data analysis to represent data points in a high-dimensional space in a lower-dimensional space, typically for visualization or dimensionality reduction. Representation is achieved by constructing a graph representation of the data points and using the eigenvectors of the graph's Laplacian matrix to define the embedding. The resulting embedding preserves the pairwise distances between data points up to a scaling factor and can be used for various tasks such as clustering, classification, and visualization. Spectral embedding has been widely applied in various fields, including computer vision, natural language processing, and social network analysis, due to its ability to reveal the underlying structure of the data and facilitate downstream tasks. Kernel principal component analysis (kernel PCA) is a non-linear dimensionality reduction technique that extends the traditional PCA by projecting the data points onto a higher-dimensional feature space, where the data is linearly separable. Linear separation is achieved using a kernel function to compute the dot product of the data points in the feature space. The resulting embedding preserves the pairwise distances between data points up to a scaling factor and can be used for various tasks such as clustering, classification, and visualization. Kernel PCA has been widely applied in various fields, including computer vision, natural language processing, and genomics, due to its ability to capture non-linear patterns in the data and facilitate downstream tasks. However, kernel PCA suffers from high computational complexity, as it requires the computation of the kernel matrix, which has a quadratic time complexity concerning the number of data points. Complexity makes kernel PCA impractical for large datasets.

One possible metric for successful data dimensionality reduction is trustworthiness. The trustworthiness metric is used to evaluate the success of the data dimensionality reduction methods used in the study [40]. This metric is calculated using the following formula.

$$T(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{j \in N_i^k} \max(0,\ (r(i,j) - k)) \tag{22}$$

where for each observation i, N_i^k are its k nearest neighbours (the output space obtained after applying the methods), and for each observation j, $r(i,j)$ is its original data space. If an unintended observation is observed in the output space, it is penalized. The number of neighbours used in the study is 5. The generalized algorithm of the extended CBMIDE is shown in the Algorithm 2.

Algorithm 2: RCBMIDE clustering algorithm

Input: Data set X= [X_1, X_2, . . . , X_n], cluster number K, the smoothness parameter h, and the percentage ratio of outliers p_0, number of iterations t, number of dimensions d, data reduction method.
Output: C_1, C_2, _ _, C
Input data reduction:
For j= 2: d < 15 do
Reduce dimension with the dimensionality reduction method
Calculate the trustworthiness of the reduced dimensions with (13)
Choose the best-reduced dimensions data based on trustworthiness.
Initiation of the mean vector using the k-means and k-means++ method
Generate a T matrix. The set T calculation where directions are evenly spaced on the sphere.
1 For i = 1: t do
2 Density estimation for each point and cluster
Update $\hat{M}$, $\hat{p}_k$, $\hat{R}$ values
3 End
4 Return C_1, C_2, _, C_t and $\hat{M}$, $\hat{p}_k$, $\hat{R}$

3. Materials and Methods

This section provides information on materials and methods. The first part of the section discusses the clustering evaluation metrics used in the research. Then, research data sets and data preparation for the research. Finally, the experimental setup was used in the study.

3.1. Clustering Evaluation Metrics

Clustering results can be evaluated differently depending on whether the true values are known. In the real-world case, the true values are unknown, so evaluation metrics such as Calinski–Harabasz [41], Davies–Bouldin [42], or the silhouette coefficient must be used. Meanwhile, in this study, the true values of the clusters are known, whereas synthetic data sets are used. For this reason, other performance evaluation metrics can be used: J score [43], normalized mutual information (NMI) [44], adjusted rand index (ARI) [45], accuracy (ACC) [46], and Fowlkes–Mallows index (FMI) [47]. In this study, the primary metric for evaluating data clustering is ACC. Furthermore, if the accuracy is close to or equal, the clustering results are evaluated by NMI, ARI, and other metrics mentioned previously. With the metric values remaining the same, the created database also stores more metrics to evaluate the clustering results, such as completeness and homogeneity scores. Additionally, visual representation of two-dimensional and three-dimensional data to visually evaluate the clustering results.

Data clustering accuracy is evaluated based on the sum of the diagonal elements of the confusion matrix, divided by the number of samples to obtain a value between 0 and 1. This work considers this accuracy the only primary metric by which it is evaluated for final clustering success.

$$ACC = \frac{1}{N}\sum_{i=1}^{k} n_i \tag{23}$$

The N is the total number of observations, n_i is the number of data points correctly divided into the corresponding cluster i, and k is the cluster number.

3.2. Research Datasets

This subsection provides basic information about the data used in the study. This study uses more than 20 popular data sets from other scientific studies to evaluate clustering. Table 1 provides basic information about the data. The following methods are used for data preparation/normalization: Raw, MinMax, Standard, Robust, Max-Abs, QuantileNormal, QuantileUniform, PowerTransformer, and Normalizer. It is worth noting that raw data carries a high risk of possible differences between variables. However, the inclusion of

these data allows evaluation of the influence of different data preparation methods on the results of data clustering. For example, the MinMax scaler's data is compressed to a (0,1) scale based on the maximum and minimum values.

$$X_{std} = \frac{(X - X_{min})}{(X_{max} - X_{min})} \tag{24}$$

In the case of the standard scaler, the data is standardized by removing the mean and scaling to unit variance.

$$X_{std} = \frac{X - \overline{X}}{s} \tag{25}$$

The robust scaler removes the mean and scales the data by the quantile range (default IQR: interquartile range). The IQR is the interval between the first quartile (25th quantile) and the third (75th) quartile).

$$X_{std} = \frac{X - Q_1(X)}{Q_3(X) - Q_1(X)} \tag{26}$$

Table 1. A description of the data set used.

ID	Data Sets	Sample Size (N)	Dimensions (D)	Classes
1	Balance-scale	625	4	3
2	Arrhythmia	452	262	13
3	atom	800	3	2
4	Breast	570	30	2
5	Coil20	1440	1024	20
6	CPU	209	6	4
7	Dermatology	366	17	6
8	Diabetes	442	10	4
9	Ecoli	336	7	8
10	German	1000	60	2
11	Glass	214	9	6
12	Haberman	306	3	2
13	Heart-statlog	270	13	2
14	Iono	351	34	2
15	Iris	150	4	3
16	pmf	649	3	5
17	segment	2310	19	7
18	spambase	4601	57	2
19	Thyroid	215	5	3
20	wdbc	569	30	2
21	Wine	178	13	3

3.3. Experimental Setup

This subsection reviews the experimental setup of this research. Considering that the research uses many different data dimension reduction methods, scaler methods, and clustering methods, an appropriate experimental setup is required. First, data preparation and dimension reduction are performed on a Linux server (30 CPUs, A30 24 GB GPU). Then, with the help of DVC, the data is versioned and transferred to other machines used in the study. All research results are stored in a PostgreSQL database, which allows nonrepetitive calculations to be performed on multiple machines simultaneously. In addition, Grafana is available for data visualization and Prometheus for error logging. Calculations are performed using five separate machines: three servers and two computers. The parameters of these machines: 1S—30 CPUs, 64 GB RAM; 2S—8 CPUs 128 GB RAM, 3S—16 CPUs, 64 GB RAM; 1C—Intel Core i7-8750H (6 cores, 12 logical processors), 32 GB RAM, 2060 6 GB GPU; and 2C—Intel Core i7-12700 K (12 cores, 20 threads), 32 GB RAM, NVIDIA 3080 Ti 12 GB GPU. The calculations of combinations of methods can be performed in parallel,

and then the results are stored in a database table. Different clustering methods are used to provide different parameters, so the parameters of each method are selected separately. A predefined rule is used for parameter selection: Each parameter has at least 20 steps. More than 65 million models have been created and saved in the current database.

4. Results

This section presents the main results of the research and compares reduced clustering based on the modified inversion formula density estimation (RCBMIDE) with other clustering methods. The notation used in the following results is as follows: Agg—agglomerative clustering, GMM—Gaussian mixture model, and BGMM—Bayesian Gaussian mixture; the full names of other methods are given. A rich set of different parameters is used for data clustering. Using the Agg method, the affinity values used were Euclidean, Manhattan, and cosine, and the linkage values used were ward, complete, average, and single. In the case of the BIRCH method, two main parameters are changed: Threshold and branching factor. The threshold was changed from 0.05 to 1 and branching factor from 2 to 10. The k-means method initializes the GMM, BGMM, CBMIDE, and RCBMIDE centres. The CBMIDE and RCBMIDE methods also changed the smoothness parameter h from 0 to 0.5 and the size of the generated matrix T from 10 to 1000 observations. The DBSCAN and HDBSCAN methods used Euclidean, Manhattan, and Chebyshev distances and changed the values of eps and min samples. The minimum and maximum limits of the eps values were determined by calculating the closest and farthest distance between points in the data set: the Min sample value changes from 1 to $N/2$, where N is the number of data samples. The database currently contains more than 60 million models (including models using dimensionality reduction), and the current best-performing models are shown in the subsections of this section. In the case of each model, different scalers are also used, which allows one to find the most optimal model. The best clustering methods differ for different datasets, and there is no single best method. It is also worth noting that the CBMIDE method cannot work for some datasets due to many dimensions.

4.1. Performances of Reduced Clustering Based on the Modified Inversion Density Estimation for Lower Dimensions Datasets

This subsection reviews the results obtained with lower-dimensional data sets. This section is designed to compare the original CBMIDE method and the modified RCBMIDE method. Given that the original CBMIDE method suffers in high dimensions, for this reason, the comparison can only be made for lower dimensional datasets. Based on the results obtained (see Table 2), it can be observed that the accuracy of the data clustering using the newly modified reduced CBMIDE method (RCBMIDE) provides better clustering results than the original CBMIDE method. These results prove our hypothesis about reduced method application for data clustering and result improvement compared with the original method. It can also be noticed that for five data sets, the best results are obtained using the RCBMIDE method compared with other popular methods. In other cases, the accuracy results are close to the best clustering method for the corresponding data set, which proves the idea of reduced/modified method application in data clustering. Talking about modifications, best results were achieved mainly with the sixth modification (TriMap) and fifth modification (UMAP). For complete best modifications in each dataset, see Table A2.

Table 2. Different clustering methods results for lower dimensions datasets.

Dataset	Agg	BIRCH	GMM	BGMM	DBSCAN	K-Means	HDBSCAN	CBMIDE	RCBMIDE
1balance-scale	0.624	0.658	0.568	0.586	0.464	0.603	0.597	0.576	**0.693**
atom	**1.000**	0.868	0.883	0.960	**1.000**	0.719	**1.000**	0.891	**1.000**
cpu	0.823	0.828	0.746	0.641	0.833	0.761	0.813	0.815	**0.858**
diabetes	0.507	**0.514**	0.459	0.455	0.482	0.428	0.477	0.502	0.512
ecoli	0.804	**0.845**	0.762	0.747	0.682	0.688	0.646	0.754	0.817
glass	0.514	0.565	0.509	0.528	0.514	0.547	0.528	0.527	**0.607**
Haberman	0.748	**0.761**	0.667	0.716	0.758	0.748	0.739	0.735	0.742
iris	0.967	0.973	0.967	0.893	0.94	0.967	0.700	0.975	**0.983**
pmf	0.977	0.977	0.92	0.977	**0.983**	0.844	0.978	0.934	0.981
thyroid	0.93	0.949	**0.963**	0.949	0.874	0.944	0.823	0.778	0.834
Wine	0.978	**0.994**	0.972	0.983	0.949	0.978	0.876	0.953	0.963

Values in bold and underlined indicate the best results for each dataset.

4.2. Performances of Reduced Clustering Based on the Modified Inversion Density Estimation for Higher-Dimensional Datasets

This subsection reviews the results obtained using datasets with more dimensions. It is important to note that CBMIDE does not work for large data dimensions, so it is impossible to include it in this comparison. Based on these results (see Table 3), it can be said that the proposed extension of the method is valid since, in this case, it is possible to apply the clustering method based on the density estimation of the inversion formula. The results showed that the extended RCBMIDE method for specific data sets has the highest clustering accuracy compared to other existing clustering methods. For example, the best results were achieved on german and segment datasets. For five more datasets, the achieved results are second best in the comparison. For the german dataset, the best method modification is the eighth modification (multidimensional scale (MDS)), and for the segment dataset, the best results were achieved using the sixth modification (TriMap). For complete best modifications in each dataset, see Table A1.

Table 3. Different clustering methods results for higher dimensions datasets.

Dataset	Agg	BIRCH	GMM	BGMM	DBSCAN	K-Means	HDBSCAN	RCBMIDE
arrhythmia	**0.600**	0.571	0.485	0.431	0.573	0.438	0.582	0.597
Breast	0.942	**0.954**	0.951	0.953	0.903	0.928	0.743	0.909
Coil20	0.738	0.738	0.638	0.675	0.867	0.733	**0.884**	0.882
dermatology	0.956	**0.978**	0.91	0.855	0.694	0.962	0.809	0.867
german	0.705	0.713	0.696	0.638	0.712	0.673	0.704	**0.716**
heart-statlog	0.807	0.811	0.819	0.826	0.815	**0.848**	0.626	0.831
iono	0.729	0.795	0.849	0.809	**0.929**	0.712	0.906	0.899
segment	0.708	0.732	0.631	0.612	0.529	0.665	0.530	**0.789**
spambase	0.878	**0.918**	0.856	0.857	0.693	0.854	0.690	0.905
wdbc	0.942	0.954	0.951	**0.958**	0.903	0.928	0.743	0.951

Values in bold and underlined indicate the best results for each dataset.

5. Discussion

This paper reviews a data clustering method based on modified inversion formula density estimation (CBMIDE) and its extension. It uses data dimensionality reduction—reduced clustering based on the modified inversion formula density estimation (RCBMIDE). This extension of the data clustering method is carried out because it was observed that the CBMIDE method does not work for higher dimensions cases. For this reason, this paper proposes an extension of the method by including data dimensionality reduction in the data clustering algorithm. The application of data dimensionality reduction makes it possible to apply the improved method to higher dimensions (d > 15). For the extension, data reduction methods, such as principal component analysis, independent component analysis, factor analysis, and ISOMAP, were used to reduce data dimensions. In addition, various methods

of reducing data dimensions and dimensions were used. It allowed us to evaluate the possible data impact of compression on the clustering accuracy. The results showed that the new method extension positively affects the data clustering results. Compared to other popular data clustering methods, the newly constructed RCBMIDE method works well and achieves the best accuracy in many cases. When the accuracy is not the best, the data clustering results are close to those obtained by the best models. Furthermore, an experimental evaluation was made of the data of lower dimensions. This comparison aims to compare this with the CBMIDE method without the extension. The results showed that the new modification method has better results than the usual CBMIDE method in all cases. Therefore, we can say that the hypothesis about method modification's impact on better clustering results was proved. It is also worth noting that the results showed that the data dimensionality reduction methods used in all cases are different, so it is not easy to decide which method works best. The trustworthiness metric allowed us to compare how well the dimensionality reduction was performed and the effectiveness of the clustering after that. The results showed that the UMAP and TriMap methods, which are methods of the same family, usually have the best results for lower dimensional data. It can be said that this paper proved the hypothesis about the extension of the developed method to large-dimensional data applications in data clustering. Further research will be conducted based on this work. The future direction of these studies and additional studies will be carried out using a more significant number of data clustering methods, more data sets, and dimensionality reduction methods. Further studies also aim to evaluate the influence of individual method parameters on the results of data clustering, not only by isolating the best models but by evaluating these models in more detail. Furthermore, this study is the beginning of further studies in which the main focus will be deep clustering. The development of the deep clustering method will be based on the density estimation of the modified inversion formula.

Author Contributions: Conceptualization, T.R. and M.L.; methodology, T.R.; software, T.R. and M.L.; formal analysis, T.R. and M.L.; investigation, T.R. and M.L.; writing—original draft preparation, T.R. and M.L.; writing—review and editing, M.L.; supervision, T.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank the area editor and the reviewers for valuable comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. RCBMIDE method extension (dimensionality reduction) for each higher-dimensions dataset.

Dataset	RCBMIDE
arrhythmia	TriMap (5)
breast	UMAP (3)
Coil20	LLE (3)
dermatology	NMF (2)
german	MDS (3)
heart-statlog	CosinePCA (4)
iono	LLE (2)
segment	TriMap6th (5)
spambase	PCA (10)
wdbc	MDS (5)

The value in the brackets represents the number of dimensions.

Table A2. RCBMIDE method extension (dimensionality reduction) for each lower-dimensions dataset.

Dataset	RCBMIDE
1balance-scale	UMAP (2)
atom	UMAP (2)
cpu	TriMap (5)
diabetes	TriMap (5)
ecoli	UMAP (4)
glass	LLE (5)
Haberman	TSVD (2)
iris	UMAP (6)
pmf	LLE (2)
thyroid	TSVD (4)
Wine	PCA (4)

The value in the brackets represents the number of dimensions.

References

1. Chen, C.-H. A hybrid intelligent model of analyzing clinical breast cancer data using clustering techniques with feature selection. *Appl. Soft Comput.* **2014**, *20*, 4–14. [CrossRef]
2. Alashwal, H.; El Halaby, M.; Crouse, J.J.; Abdalla, A.; Moustafa, A.A. The application of unsupervised clustering methods to Alzheimer's disease. *Front. Comput. Neurosci.* **2019**, *13*, 31. [PubMed]
3. Farouk, Y.; Rady, S. Early diagnosis of alzheimer's disease using unsupervised clustering. *Int. J. Intell. Comput. Inf. Sci.* **2020**, *20*, 112–124. [CrossRef]
4. Liu, A.-A.; Nie, W.-Z.; Gao, Y.; Su, Y.-T. View-based 3-D model retrieval: A benchmark. *IEEE Trans. Cybern.* **2017**, *48*, 916–928. [CrossRef] [PubMed]
5. Nie, W.; Cheng, H.; Su, Y. Modeling temporal information of mitotic for mitotic event detection. *IEEE Trans. Big Data* **2017**, *3*, 458–469. [CrossRef]
6. Abualigah, L.; Gandomi, A.H.; Elaziz, M.A.; Hamad, H.A.; Omari, M.; Alshinwan, M.; Khasawneh, A.M. Advances in meta-heuristic optimization algorithms in big data text clustering. *Electronics* **2021**, *10*, 101.
7. Lukauskas, M.; Pilinkienė, V.; Bruneckienė, J.; Stundžienė, A.; Grybauskas, A.; Ruzgas, T. Economic Activity Forecasting Based on the Sentiment Analysis of News. *Mathematics* **2022**, *10*, 3461. [CrossRef]
8. Trentin, E.; Lusnig, L.; Cavalli, F. Parzen neural networks: Fundamentals, properties, and an application to forensic anthropology. *Neural Netw.* **2018**, *97*, 137–151. [CrossRef] [PubMed]
9. Lukauskas, M.; Ruzgas, T. A New Clustering Method Based on the Inversion Formula. *Mathematics* **2022**, *10*, 2559. [CrossRef]
10. Ding, C.; He, X. K-means clustering via principal component analysis. In Proceedings of the 21st International Conference on Machine Learning, Banf, AL, Canada, 4–8 July 2004; p. 29.
11. Yang, L.; Liu, J.; Lu, Q.; Riggs, A.D.; Wu, X. SAIC: An iterative clustering approach for analysis of single cell RNA-seq data. *BMC Genom.* **2017**, *18*, 689.
12. Kakushadze, Z.; Yu, W. * K-means and cluster models for cancer signatures. *Biomol. Detect. Quantif.* **2017**, *13*, 7–31. [CrossRef] [PubMed]
13. Shin, J.; Berg, D.A.; Zhu, Y.; Shin, J.Y.; Song, J.; Bonaguidi, M.A.; Enikolopov, G.; Nauen, D.W.; Christian, K.M.; Ming, G.-L. Single-cell RNA-seq with waterfall reveals molecular cascades underlying adult neurogenesis. *Cell Stem Cell* **2015**, *17*, 360–372. [CrossRef] [PubMed]
14. Feng, C.; Liu, S.; Zhang, H.; Guan, R.; Li, D.; Zhou, F.; Liang, Y.; Feng, X. Dimension reduction and clustering models for single-cell RNA sequencing data: A comparative study. *Int. J. Mol. Sci.* **2020**, *21*, 2181. [CrossRef] [PubMed]
15. Melit Devassy, B.; George, S.; Nussbaum, P. Unsupervised clustering of hyperspectral paper data using t-SNE. *J. Imaging* **2020**, *6*, 29. [CrossRef] [PubMed]
16. Bollon, J.; Assale, M.; Cina, A.; Marangoni, S.; Calabrese, M.; Salvemini, C.B.; Christille, J.M.; Gustincich, S.; Cavalli, A. Investigating How Reproducibility and Geometrical Representation in UMAP Dimensionality Reduction Impact the Stratification of Breast Cancer Tumors. *Appl. Sci.* **2022**, *12*, 4247. [CrossRef]
17. Li, H.; Liu, J.; Liu, R.W.; Xiong, N.; Wu, K.; Kim, T.-h. A dimensionality reduction-based multi-step clustering method for robust vessel trajectory analysis. *Sensors* **2017**, *17*, 1792. [CrossRef]
18. Wenskovitch, J.; Crandell, I.; Ramakrishnan, N.; House, L.; North, C. Towards a systematic combination of dimension reduction and clustering in visual analytics. *IEEE Trans. Vis. Comput. Graph.* **2017**, *24*, 131–141. [CrossRef]
19. Tang, B.; Shepherd, M.; Milios, E.; Heywood, M.I. Comparing and combining dimension reduction techniques for efficient text clustering. In Proceedings of the SIAM International Conference on Data Mining, Newport Beach, CA, USA, 23 April 2005; pp. 17–26.
20. Wang, X.-D.; Chen, R.-C.; Zeng, Z.-Q.; Hong, C.-Q.; Yan, F. Robust dimension reduction for clustering with local adaptive learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *30*, 657–669. [CrossRef]

21. Markos, A.; D'Enza, A.I.; van de Velden, M. Beyond tandem analysis: Joint dimension reduction and clustering in R. *J. Stat. Softw.* **2019**, *91*, 1–24. [CrossRef]
22. Wenskovitch, J.; Dowling, M.; North, C. With respect to what? simultaneous interaction with dimension reduction and clustering projections. In Proceedings of the 25th International Conference on Intelligent User Interfaces, Cagliari, Italy, 17–20 March 2020; pp. 177–188.
23. Ruzgas, T.; Lukauskas, M.; Čepkauskas, G. Nonparametric Multivariate Density Estimation: Case Study of Cauchy Mixture Model. *Mathematics* **2021**, *9*, 2717. [CrossRef]
24. Kavaliauskas, M.; Rudzkis, R.; Ruzgas, T. The projection-based multivariate density estimation. *Acta Comment. Univ. Tartu. Math.* **2004**, *8*, 135–141. [CrossRef]
25. Biernacki, C.; Celeux, G.; Govaert, G. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Comput. Stat. Data Anal.* **2003**, *41*, 561–575. [CrossRef]
26. Xu, Q.; Yuan, S.; Huang, T. Multidimensional uniform initialization Gaussian mixture model for spar crack quantification under uncertainty. *Sensors* **2021**, *21*, 1283. [CrossRef] [PubMed]
27. Fraley, C. Algorithms for model-based Gaussian hierarchical clustering. *SIAM J. Sci. Comput.* **1998**, *20*, 270–281. [CrossRef]
28. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B (Methodol.)* **1977**, *39*, 1–22.
29. Everitt, B. *Finite Mixture Distributions*; Springer Science & Business Media: New York, NY, USA, 2013.
30. Redner, R.A.; Walker, H.F. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Rev.* **1984**, *26*, 195–239. [CrossRef]
31. Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52. [CrossRef]
32. Comon, P. Independent component analysis, a new concept? *Signal Process.* **1994**, *36*, 287–314. [CrossRef]
33. Jöreskog, K.G. Factor analysis as an error-in-variables model. In *Principals of Modern Psychological Measurement*; Routledge: Abingdon-on-Thames, UK, 1983; pp. 185–196.
34. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2580–2605.
35. Van Der Maaten, L. Accelerating t-SNE using tree-based algorithms. *J. Mach. Learn. Res.* **2014**, *15*, 3221–3245.
36. Li, W.; Cerise, J.E.; Yang, Y.; Han, H. Application of t-SNE to human genetic data. *J. Bioinform. Comput. Biol.* **2017**, *15*, 1750017. [CrossRef] [PubMed]
37. Kobak, D.; Berens, P. The art of using t-SNE for single-cell transcriptomics. *Nat. Commun.* **2019**, *10*, 1–14. [CrossRef] [PubMed]
38. Amid, E.; Warmuth, M.K. TriMap: Large-scale dimensionality reduction using triplets. *arXiv* **2019**, arXiv:1910.00204.
39. Ghojogh, B.; Ghodsi, A.; Karray, F.; Crowley, M. Locally linear embedding and its variants: Tutorial and survey. *arXiv* **2020**, arXiv:2011.10925.
40. Venna, J.; Kaski, S. Neighborhood Preservation in Non-linear Projection Methods: An Experimental Study. In Proceedings of the Artificial Neural Networks—ICANN, Berlin/Heidelberg, Germany, 21–25 August 2001; pp. 485–491.
41. Caliński, T.; Harabasz, J. A dendrite method for cluster analysis. *Commun. Stat. -Theory Methods* **1974**, *3*, 1–27. [CrossRef]
42. Davies, D.L.; Bouldin, D.W. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *PAMI-1*, 224–227. [CrossRef]
43. Ahmadinejad, N.; Liu, L. J-Score: A Robust Measure of Clustering Accuracy. *arXiv* **2021**, arXiv:2109.01306.
44. Zhong, S.; Ghosh, J. Generative model-based document clustering: A comparative study. *Knowl. Inf. Syst.* **2005**, *8*, 374–384. [CrossRef]
45. Lawrence, H.; Phipps, A. Comparing partitions. *J. Classif.* **1985**, *2*, 193–218.
46. Wang, P.; Shi, H.; Yang, X.; Mi, J. Three-way k-means: Integrating k-means and three-way decision. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 2767–2777. [CrossRef]
47. Fowlkes, E.B.; Mallows, C.L. A method for comparing two hierarchical clusterings. *J. Am. Stat. Assoc.* **1983**, *78*, 553–569. [CrossRef]

mathematics

MDPI

Article

Economic Activity Forecasting Based on the Sentiment Analysis of News

Mantas Lukauskas [1,*], Vaida Pilinkienė [2], Jurgita Bruneckienė [2], Alina Stundžienė [2], Andrius Grybauskas [2] and Tomas Ruzgas [1]

[1] Department of Applied Mathematics, Faculty of Mathematics and Natural Sciences, Kaunas University of Technology, 44249 Kaunas, Lithuania
[2] School of Economics and Business, Kaunas University of Technology, 44249 Kaunas, Lithuania
* Correspondence: mantas.lukauskas@ktu.lt

Abstract: The outbreak of war and the earlier and ongoing COVID-19 pandemic determined the need for real-time monitoring of economic activity. The economic activity of a country can be defined in different ways. Most often, the country's economic activity is characterized by various indicators such as the gross domestic product, the level of employment or unemployment of the population, the price level in the country, inflation, and other frequently used economic indicators. The most popular were the gross domestic product (GDP) and industrial production. However, such traditional tools have started to decline in modern times (as the timely knowledge of information becomes a critical factor in decision making in a rapidly changing environment) as they are published with significant delays. This work aims to use the information in the Lithuanian mass media and machine learning methods to assess whether these data can be used to assess economic activity. The aim of using these data is to determine the correlation between the usual indicators of economic activity assessment and media sentiments and to forecast traditional indicators. When evaluating consumer confidence, it is observed that the forecasting of this economic activity indicator is better based on the general index of negative sentiment (comparisons with univariate time series). In this case, the average absolute percentage error is 1.3% lower. However, if all sentiments are included in the forecasting instead of the best one, the forecasting is worse and in this case the MAPE is 5.9% higher. It is noticeable that forecasting the monthly and annual inflation rate is thus best when the overall negative sentiment is used. The MAPE of the monthly inflation rate is as much as8.5% lower, while the MAPE of the annual inflation rate is 1.5% lower.

Keywords: clustering; economic activity; natural language processing; NLP; transformers; BERT; forecasting; nowcasting; economic sentiment

MSC: 68T50; 91B84; 62H30

Citation: Lukauskas, M.; Pilinkienė, V.; Bruneckienė, J.; Stundžienė, A.; Grybauskas, A.; Ruzgas, T. Economic Activity Forecasting Based on the Sentiment Analysis of News. *Mathematics* **2022**, *10*, 3461. https://doi.org/10.3390/math10193461

Academic Editor: Gheorghe Săvoiu

Received: 31 August 2022
Accepted: 19 September 2022
Published: 22 September 2022

1. Introduction

Currently, artificial intelligence is subject to more and more different applications in practice. One of the areas of artificial intelligence that has seen significant improvement in recent years is natural language processing. Natural language processing is a discipline with characteristics of linguistics and computer science. This field applies various mathematical and computational methods to natural language processing. The application areas can be diverse and include text reading and voicing [1], automatic translation [2] (which everyone often uses), automatic text correction [3], information search [4], and many other areas. Natural language processing is widely used in the activities of companies, both for the previously mentioned tasks and for various others. One such task is sentiment analysis. Sentiment analysis uses mathematical methods and textual information to determine whether the presented text is positive or negative [5,6]. Furthermore, the text can be

analyzed in another way: deciding whether the text is positive or negative and deciding the mood of the text itself. This application of natural language processing allows one to automate various processes and increase the speed of data analysis. Even with enormous amounts of data, it is possible to determine whether the analyzed texts are positive or negative. This application can not only be used in the activities of companies, and specifically user feedback on companies, but also to evaluate economic processes. Currently, to link information in the press with economic, political, and other phenomena, indices are usually used and calculated based on the number of certain words in the text [7,8]. The authors calculated words such as economy, uncertainty, industry, politics, regulation, deficit, etc. Repetition in the texts is made up of an index based on them [9,10]. A compilation of the index allows the researcher to link the obtained index with other indicators to assess the relationship of certain events, such as wars, representing the crisis with the index. Newer research shows that the relevance of this topic is indeed great, and the authors' attention is currently explicitly directed to the application of machine learning in sentiment analysis. One such study is Shapiro et al. (2022), which applied both word-based computing and machine learning techniques [11]. It is also important to mention that many studies rely on more straightforward methods, such as bag-of-words (BoW) and global vectors for word representation (GloVE). However, it is also noticeable that more and more diverse machine learning methods are being used, providing faster and often better results [12,13]. Moreover, a smaller dataset is often used due to possible computational problems or limited data availability. It is also noticeable that news sentiment analysis is often associated with stock markets, as they are susceptible to various events and people's moods [12,14]. However, this use is not limited to predicting these indicators and can be applied to predicting various other indicators. This article aims to use natural language-processing technologies to analyze the news portal's news and determine whether the news sentiment index influences different indicators of economic activity. The article hypothesizes that the index of negative Lithuanian news is inversely related to indicators of economic activity. For example, as the negative Lithuanian news index increases, the unemployment rate increases and GDP decreases. This hypothesis is based on the fact that possible negative consequences for the economy are discussed before specific economic processes occur, so the use of natural language processing for this purpose can help predict economic indicators faster and even more accurately. This research paper may contribute to the development of natural language, and by establishing that news sentiment analysis can be used to predict economic activity, it may help to establish further economic guidance.

This article is organized as follows. Section 2 introduces the concept of economic activity, methods of economic activity assessment, and indicators used. The third chapter of this article introduces natural language processing methods and models used in natural language sentiment analysis. The fourth chapter of this article discusses the data, methods, and metrics used in the research. The fifth chapter discusses the results obtained during the research and compares different methods of sentiment analysis and forecasting indicators based on the sentiment index. Finally, the conclusions and future work are discussed in the sixth chapter.

2. Economic Activity

The outbreak of war and the earlier and ongoing COVID-19 pandemic determined the need for real-time monitoring of economic activity. The economic activity of a country can be defined in different ways. Most often, the country's economic activity is characterized by various indicators such as the gross domestic product, the level of employment or unemployment of the population, the price level in the country, inflation, and other frequently used economic indicators. The most natural way was to use the gross domestic product (GDP) and industrial production. However, such traditional tools have started to decline in modern times (when the timely knowledge of information becomes a critical factor in decision making in a rapidly changing environment) as they are published with significant delays. The most common indicators of economic activity cover the economy according to

different dimensions: private household consumption, production activity, labour market, domestic and international trade, prices, environment (conventional pollution), transport, and logistics. States and investors seek to assess economic activity as soon as possible to make timely decisions. Data delay challenges are particularly painful during periods of various shocks (pandemic, war) when countries' governments have to make urgent decisions. Economic shocks significantly distort macroeconomic forecasts due to the lag effect of traditional macroeconomic indicators and their nature [15,16].

When assessing the country's economic activity, it is usually associated with the gross domestic product or changes in industrial production, which allow one to assess the actions taking place in the country's industry/production [17–19]. However, as mentioned earlier, various sudden economic changes, such as war or pandemics, suggest that the usual indicators for monitoring economic activity are no longer sufficient. For this reason, the number of monitored indicators is expanded, and the frequency of their monitoring is increased to assess the situation in time [20–22]. Examples of such new data can be Google's mobile movement data, satellite data, and other data related to people's mobility during the pandemic [20,23]. These data were previously used very rarely, but now the conditions are set for broader use of such data. It is also worth noting that, for example, Google data can often be used in real time. In recent years, real-time/high-frequency data have received substantial attention. Although most methods are still based on historical data, which are characterized by a relatively significant lag (often a lag of one month), such a delay is significant for the accuracy of forecasts and the real assessment of the situation. This problem has been studied by several researchers [24,25], who unanimously agree that the lack of data is the main problem when making timely decisions.

New economic modelling capabilities are being sought to help address this issue. For this reason, machine learning methods and their use are essential in economic modelling. Applying artificial intelligence methods (to analyse and interpret data, as well as provide more accurate forecasts) [26] and processing large amounts of data (Big Data) are both essential. Compared to previously used methods, machine learning methods can help to assess the situation better, as they often perform better than traditional methods. Some authors integrate machine learning techniques in their work in order to process large amounts of data, including various alternative indicators that have not been evaluated before [27–30]. The possibilities of processing large amounts of data make it possible to use data such as:

- social media information (search keywords, comments);
- business company data (prices of real estate and goods on online portals, the volume of transactions);
- mobility data (fixed and mobile sensor data, satellite images, pollution data);
- Energy consumption data;
- Financial market data, credit card transactions;

Forecasting becomes much simpler and can be carried out with extremely low latency with such data. It is all the more important to mention that the amount of data generated is increasing yearly. The high frequency of data generation makes it possible to have high-frequency data; if data were only previously available once a year, it is now possible to have weekly, daily, or even hourly data [31–33]. Some authors use a combination of traditional and non-traditional indicators to obtain the best result [34,35], combining high-frequency indicators with conventional and low-frequency macroeconomic variables. More and more researchers are using these indicators, indicating that these new indicators will become more and more important for economic monitoring in the future [26].

For this reason, as mentioned earlier, the aim of this work is to use the information in the Lithuanian mass media and machine learning methods to assess whether these data can be used for assessing economic activity. Furthermore, the aim of using these data is to determine the correlation between the usual indicators of economic activity assessment and media sentiments and to forecast traditional indicators. Despite the growing number of scientific articles [30,34,36], confirming the contribution of high-frequency information

means providing an accurate forecast of economic indicators. Research [37] is still refuting or requires further attention. However, various results and active discussions among scientists only confirm the relevance and novelty of the problem.

3. Natural Language and Transformers

Natural language processing is the computer analysis and processing of natural language (which can be both written and audio information) using various mathematical methods for linguistic application. Natural language processing can be used for a variety of tasks. Natural language processing was introduced in the mid-20th century, but only rule-based systems could be developed at that time. Later, neural networks, or rather recurrent neural networks (RNNs), were introduced. These neural networks made it possible to perform various tasks in which static values, and the dynamics of these values, are essential. Due to the shortcoming of these methods, which is related to their memory, another model of neural networks developed from them: the long–short-term memory neural network. After such great discoveries and their application in natural language processing, it seemed that the best result was achieved, but in 2017, a new structure of transformers was created [38]. Moreover, most natural language processing tasks are currently being solved using these structure models. Transformers can be said to have fundamentally changed the direction of natural language processing and allowed the development of many different applications. The basic structure of transformer models is presented in the figure below (see Figure 1).

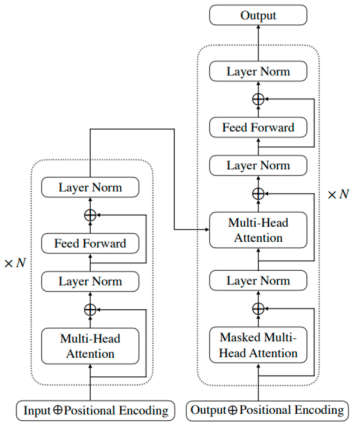


Figure 1. Schematic structure of the transformer's architecture.

It can be said that the central element in the architecture of transformers is multi-head attention, which is calculated using the following formulas [38]:

$$MultiHead(Q,\ K,\ V) = Concat(head_1, \ldots, head_h)W^0 \tag{1}$$

$$head_i = Attention(QW_i^Q,\ KW_i^K, VW_i^V) \tag{2}$$

$$Attention(Q,\ K,\ V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

where Q is the query, K is the keys, and V is the values. Concat refers to the concatenation of layers and variable h describes the number of heads. $W_i^0 \in R^{d_{model} x d_{model}}$ is a matrix of weights of the i-th head and d_{model} is the size of the input embeddings and $d_k = d_{model}/h$.

Attention $(\cdot)$ is called scaled dot–product attention because their weight values are based on key and dot–product queries. The difference between multi-head attention and masked multi-head attention is that the former allows the model to see the future context. At the same time, the latter does not, so they are used in the encoder and decoder structures. The feed-forward component transforms the output from the last transformer decoder block into a probability distribution using FC layers with a softmax activation function. A position encoding is added to each input insertion to include the order of the input sequence. Currently, there are many models based on the structure of transformers, and most of the ones used in practice by various language researchers are based on the structure of transformers. Around four years ago, OpenAI released its first generative pre-training transformer (GPT) model. This model was already a huge revolution in natural language processing, but two years later, OpenAI released a second version of the model which was even more powerful. The GPT-2 1.5 billion parameter model was trained with web texts [39]. The second version of the model was even ten times larger than the previously released version, so even better results characterized it. The latest GPT model is currently in its third version [40]. This model is trained with as many as 175 billion parameters. However, GPT models are only one of the structure models of transformers, one of the widely used models in BERT. Bidirectional encoder representations from transformers (BERTs) can be described as a pre-training technique based on work on contextual representations [41,42]. BERT models have many different model variants developed over the years. One of the more minor mods created for simple tasks is DistilBERT [43]. The main difference between this model and the usual BERT models is the distillation in the model, which reduces the model's volume in an extreme way, while even maintaining about 97 percent of the model's accuracy. There are also many other technical improvements to BERT models such as ALBERT [44], BART [45], DocBERT [46], or Facebook's RoBERTa [47]. Information on these models, as well as many other models, is provided in the Methods and Materials section. XLNet builds on the BERT and GPT models and aims to address their shortcomings. XLNet's core architecture is based on the Transformer-XL model [48]. However, the problem with these models is that they predict tokens in a random order rather than a sequential order [49].

Natural language processing is increasingly applied in different scientific and practical fields, as it can be applied to solve various problems. These natural language processing tasks can be information extraction from unstructured data [50], automated text generation [51,52], text translation into other languages [53], and also (for the main purpose of this research) sentiment or feeling analysis using text [54–58]. Different architectures of transformers are also used in this study, which are presented in the Materials and Methods section below.

4. Materials and Methods

This section describes how the data used in the study were obtained, how these data were processed, and the main characteristics of the data. The following subsections of this chapter describe the main methods used to perform different research tasks (natural language processing sentiment analysis, clustering, and prediction) and evaluation metrics for different research tasks (clustering and prediction). The general scheme of the study is presented in the figure below (see Figure 2); this scheme provides a general outline of the study, the individual elements of which are discussed in the subsections below.

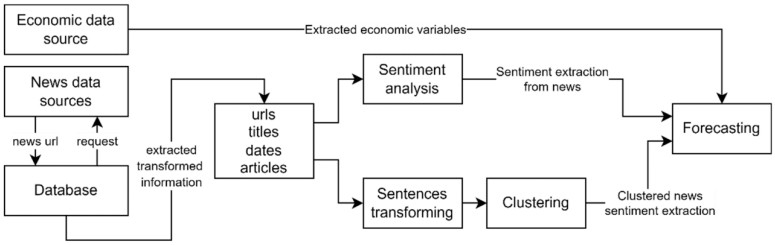


Figure 2. Basic simplified scheme of research.

4.1. Data Gathering, Processing, and Analysis

In the course of this study, articles on news portals were collected. *Python* packages *Playwright*, *Selenium*, and *BeautifulSoup* were used to collect this information during the research. The structure of the articles is presented in the figure below (see Figure 3). When collecting all the information from the articles, each part of the article was used as a separate piece of information. In addition, the publication time of the article (date variable), article category (categorical variable), article title, main article information (lead), and article text were collected as textual variables. All this information was collected using separate computer systems and stored in the *PostgreSQL* database to collect it faster.

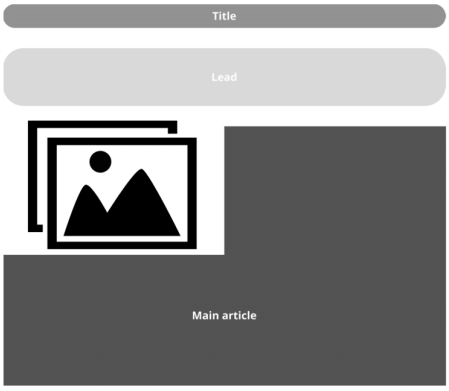


Figure 3. Example of the news structure and data used in the research.

The dataset used in the study was collected from the two largest news portals in Lithuania; the studied period was January 2000–July 2022. The total number of news articles used in the study was 2,570,815 (1,552,947 articles from the first source and 1,017,868 from the second source). In the graph below (see Figure 4), it can be seen that the amount of information on news portals increased every year. A reasonably significant increase in the news was observed in the post-crisis period, and a significant jump could also be seen after the start of the COVID-19 pandemic and the war in Ukraine.

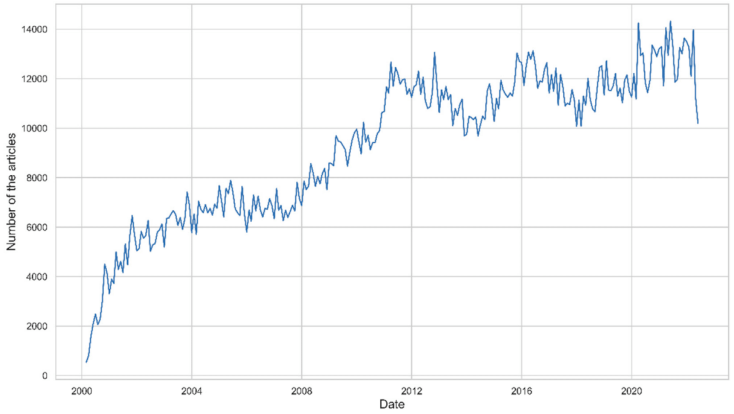


Figure 4. Monthly number of articles over time.

Economic activity data (dependent variables) were obtained using the database of the Lithuanian Statistics Department. These indicators of economic activity were selected based on the literature analysis presented above, during which it was determined which underground indicators were used by authors describing the economic activity. Additionally, when choosing the indicators, it was necessary to consider that in most cases, only annual data were provided. A significant amount of information was lost when examining annual data, with the expectation to find more frequent data. Therefore, only data with a monthly frequency were selected. This makes it possible to have a fairly large time series to forecast these indicators.

4.2. NLP Models Used in the Research

In this study, textual data were analysed; therefore, the previously discussed transformers were used to analyse these data. Transformers provided better results compared to conventional methods used before their appearance. There are quite a few sentiment analysis models, but it is worth noting that there are almost no such models in the Lithuanian language; therefore, for this reason, the articles in Lithuanian had to be translated first. Some random translations were checked, and the quality of these translations was evaluated. It is noticeable that Lithuanian–English translations were performed with high quality. These translations were performed using the Python package deep-translator. This package includes different tools, including the Google translator and the DeepL translator. Google Translate was used in this study to evaluate the translation quality.

Next, another text analysis task was performed. These tasks were performed using HuggingFace models. In the first case, the text was transformed into points in space, as this was necessary for text clustering using the sentence transformer model all-MiniLM-L6-v2 (All-MiniLM-L6-v2 model link: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2 (accessed on 20 August 2022)). It is a model of sentence transformers and maps sentences and paragraphs into a 384-dimensional dense vector space and can be used for tasks such as clustering or semantic search.

When evaluating the sentiments of different textual data, the dataset with which the used models were trained can be of considerable importance. For this reason, it was decided that a combination of different models would be used during the sentiment analysis, as opposed to one specific model. This study used 4 different pre-trained models

for text sentiment detection: DistilBERT-base-uncased, FinBERT, Twitter-roBERTa-base, and FinBERT-tone. These models were trained with different data, thus avoiding the larger influence of the training data.

The DistilBERT-base-uncased model (Distil-BERT-uncased modelio nuoroda: https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english accessed on 22 August 2022) is a reduced version of the Bert-base-uncased model but exerts extremely high performance. This model was trained on the SST-2 dataset and has an accuracy of 91.3 percent. FinBERT is a model developed by Prosus, specifically designed to analyse financial texts [59]. This model is a BERT model, but it was explicitly trained on financial textual data, which allowed this model to identify better sentiments in texts related to financial information. Data from Financial PhraseBank were used to train the model [60]. Another model used in the study was the FinBERT-tone model (FinBERT-tone modelio nuoroda: https://huggingface.co/yiyanghkust/finbert-tone accessed on 18 August 2022). The textual data of financial information were also used to train this model [61]. This model was trained using as many as three different sets of financial information. The total case size was 4.9 B tokens. Companies report 10-K and 10-Q with USD 2.5 billion tokens, earnings call transcripts with USD 1.3 billion tokens, and analyst reports with USD 1.1 billion tokens. In this tone, the model was trained with manually labelled data. This model achieves better performance in the financial tone analysis task. The final model in this study was the Twit-ter-roBERTa-base model, specifically used for sentiment analysis [62]. This model was trained using as many as 124 million Twitter messages collected over three years. It can also evaluate sentiments, not only for financial data but also for general texts.

4.3. Clustering Methods Used in the Research

The purpose of this study was to classify all news texts into groups to determine these groups' sentiments. For this purpose, cluster analysis was used; the models used are described further in this subsection. Cluster analysis is a type of unsupervised learning, the main goal of which is to classify the observations into certain unknown groups based on the similarity of the observations. In this case, observations in one cluster are as similar as possible to each other, while observations in separate clusters are different from each other. This analysis helps to discover clusters that may not usually be discernible in the original data. When analysing cluster analysis, it can be noticed that distance-based cluster analysis is usually mentioned. This type of cluster analysis is based on the distance between observations. One of the most popular k-means clustering methods was used in this study. This method is convenient to use due to its simple operation and the small number of required parameters. The k-means method divides the available data into k groups, where each observation belongs to exactly one group. In the first cycle, the data are divided into k groups. Then, during the iterations, an attempt is made to find the most suitable partition of the data so that the elements in the cluster are similar (the distance between them is the smallest). At the same time, the observations between individual clusters are different (the distance between them is the largest). The essence of the k-means method is the division of observations into k-specified clusters, but using these methods and randomly initializing the cluster centres, the clusters may be different. This method can be described in 5 main steps (see Figure 5):

1. The observations are randomly divided into k clusters, and the initial centres of these clusters are selected.
2. Cluster centres are recalculated.
3. The distance of each observation to the clusters is calculated based on distance measures.
4. The observations are assigned to the nearest cluster according to the distance to the cluster centres.
5. Steps 2–4 are repeated until the cluster centres do not change or whenchange is less than the specified tolerance limit.

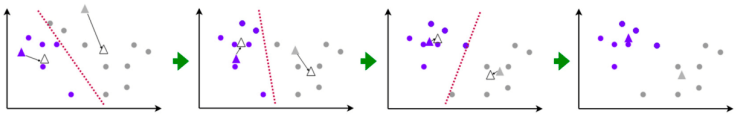


Figure 5. Visualization of the k-means method.

The new *modified inversion formula density estimation* (*MIDE*) clustering method was also used in this study [63]. This method is based on a modified inversion formula, and the obtained empirical research results show that this method performs qualitative clustering. Moreover, in order to determine the most suitable clustering methods for the analysed data, other clustering methods were used in this study: Gaussian mixture models, Bayesian Gaussian mixture models, density-based spatial clustering of applications with noise (DBSCAN) [64], balanced iterative reducing and clustering using hierarchies (BIRCH) [65], and ordering points to identify the clustering structure (OPTICS) [66]. These different models were trained by changing their parameters; thus, to determine the best clustering models, the parameters of each model were selected from its parameter set. For example, the MIDE parameters set the percentage of exceptions, which can be changed from 0 to 10%, and the DBSCAN method sets the minimum distance between points or the minimum number of points in a cluster.

4.4. Clustering Evaluation Metrics

This subsection presents the main metrics used to evaluate clustering results in the study. In this case, clustering was performed without prior knowledge of the true classes, so metrics such as accuracy and NMI (or other metrics that require true classes) cannot be used. This work used several different metrics that do not require actual classes. One of the metrics is the Calinski and Harabasz metric [67], which is also often called the variance ratio criterion. The score is the ratio of the within-cluster variance to the sum of the within-cluster variance. Another metric used was the Davies–Bouldin metric [68], which evaluates cluster similarities. This metric was calculated as the similarity between within-cluster distances and between-cluster distances. The lowest possible value for this metric was zero, and the lower the value, the better the clustering results. Finally, the last metric used was the silhouette coefficient [69]. However, it is important to emphasize that this coefficient is more difficult to calculate when such a large amount of data is used in this paper. A large amount of data makes it difficult to calculate distances for each observation. The observed silhouette coefficient is (b-a)/max (a, b). For clarity, b is the distance between the sample and the nearest cluster of which the observation is not a part. The best value is one and the worst is -1. Values near 0 indicate overlapping groups.

4.5. Forecasting Methods Used in the Research

Many different econometric models are used in scientific research to forecast economic activity and different economic variables. These models include models such as the dynamic factorial model [70,71], Bayesian vector autoregression (BVAR) models [72], and factor-augmented VAR (FAVAR) models [73]. Richardson et al. (2021) [74] demonstrated that machine learning algorithms allow central banks to assess the current state of the economy in more detail and can be more accurate than conventional econometric models. For this reason, this study did not use traditional econometric models for forecasting, but rather machine learning methods. Such methods allow the influence of sentiment analysis on different indicators of economic activity and the significance of the use of machine learning in economic forecasting to be evaluated. Considerable attention in data science is paid specifically to neural networks. Feed-forward neural networks are commonly used to solve problems, but it is important to mention that these neural networks cannot capture data variation. This makes it difficult to use these neural networks to predict economic indicators. In order to predict dynamic indicators, recurrent neural networks were created,

which allow both the current state and also the past state to be recorded, as well as data from different periods [75]. However, RNNs suffer from the problem of vanishing gradients, which hinders the learning of long data sequences. For this reason, newer long–short-term memory (LSTM) neural networks have been developed, a type of recurrent neural network that not only captures past data when the gap between input information and output is small, but also when this gap is much larger [76]. Another modification of recurrent neural networks is the gated recurrent unit (GRU). One of the main differences between LSTMs and GRUs is that GRUs do not have memory cells [77]. This type of neural network does not separate forget gate and input gate but combines them into one update gate. Moreover, this type of neural network combines the cell's state and the hidden state.

4.6. Forecasting Evaluation Metrics

An essential factor in developing machine learning models is the accuracy of these models, so functions that can evaluate the accuracy of the models are needed. Error functions perform this function by comparing the values predicted by the models and the actual values. Depending on the problem being solved, different error functions were applied. The following table shows the error functions of the regression models (see Table 1). It is essential to mention that, considering the task that is solved in this work, not all the metrics presented in the table were used, but these metrics are still discussed in the paper. The root mean square error (RMSE) [61] is the standard deviation of the errors. This metric is one of the most commonly used metrics for solving problems involving regression models. The RMSE metric describes how widely the errors are spread. The RMSE is used in climatology, forecasting, and regression analysis to verify experimental results. Another metric used in regression problems is the mean squared error [78], which can essentially be said to be the same RMSE metric, except that the root is not used in its calculation. The mean absolute error [79] is the absolute mean error of the errors, which allows us to precisely estimate the absolute error. The coefficient of determination [80] is an evaluation function whose best value is unity; the closer this value is to unity, the better the trained model.

Table 1. Most common evaluation metrics for forecasting/regression methods.

Metric Name	Formula
R^2—*coefficient of determination*	$R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$
R^2_{adj}—*adjusted coefficient of determination*	$R^2_{adj} = 1 - \left[\frac{(1-R^2)(n-1)}{n-k-1}\right]$
MAE—mean absolute error	$MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$
MSE—mean square error	$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
RMSE—rooted mean square error	$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$
MPE—mean percentage error	$MPE = \frac{100\ \%}{n}\sum_{i=1}^{n}\frac{y_i - \hat{y}_i}{y_i}$
MAPE—mean absolute percentage error	$MAPE = \frac{100\ \%}{n}\sum_{i=1}^{n}\left\lvert\frac{y_i - \hat{y}_i}{y_i}\right\rvert$
MASE—mean absolute scaled error	$MASE = \frac{1}{n}\sum_{i=1}^{n}\frac{\lvert y_i - \hat{y}_i\rvert}{\frac{1}{n-1}\sum_{i=2}^{n}\lvert y_{i-1} - y_i\rvert}$

5. Results

This section presents the main results of the study. The first subsection of this chapter (see Section 5.1) provides information on the results of news clustering using different clustering methods. These results were evaluated using the clustering performance evaluation metrics described in the previous section. The second subsection of this chapter (see Section 5.2) provides information on news sentiment analysis. The results are also

presented separately because the sentiment analysis was conducted in different directions. Sentiment analysis was performed for all news in general, individual news categories, and clusters obtained during clustering. Finally, the third subsection of this chapter (see Section 5.3) provides information on forecasting different economic indicators describing the economic activity. The forecasting of different indicators was based on the sentiment analysis results obtained in the second subsection and the clustering results presented in the first subsection.

5.1. News Clustering Results

This subsection provides information on the different clustering methods used during the study and the obtained results. In the first step of news clustering, all textual information was transformed into numerical information using sentence transformers. Using sentence transformers, textual data are transformed into 384-dimensional data. Each text corresponds to a certain point in this space, according to the words in the sentence, their meanings, and their semantic meaning. These points are then clustered based on different clustering methods, and the obtained results are compared based on the metrics discussed in the previous section. The methods with the best results are used in further research. The table below (see Table 2) shows the clustering results. More clustering methods were used in the study in the Methods section, but some problems were observed with these methods. Due to the huge amount of data, the BIRCH clustering method required as much as 4 TB of RAM, which made it hard to implement at this step of the problem. Furthermore, the DBSCAN and OPTICS methods, due to their matrix calculation, cope with the presented tasks in a difficult way. These methods take a very long time, making it difficult to discover suitable parameter sets. The table below shows the results of the four clustering methods. It can be seen that the best clustering results were obtained using the K-means method. Moreover, the MIDE method showed quite good clustering results. During the clustering, data dimensionality reduction methods were additionally applied (PCA, t-SNE, and SMACOF), but no positive influence on the clustering results was observed.

Table 2. Different models (means and standard deviation) were compared based on the Calinski and Harabasz score and the Davies–Bouldin score for 100 runs.

	Calinski and Harabasz Score	**Davies–Bouldin Score**
GMM	11,648	5.841
BGMM	13,547	6.148
K-means	13,387	4.627
MIDE	12,542	5.314

5.2. Sentiment Index of the News

This subsection presents the results of the sentiment analysis. Sentiment analysis was performed using different cuts of the datasets. In the first case, sentiment analysis was performed using the entire available dataset. In the second case, sentiment analysis was performed using news categories extracted from news articles (business, health, in Lithuania, abroad). In the last case, sentiment analysis was performed based on the clustering results. In order to perform such sentiment analysis, first, all data were clustered according to the best model determined in the previous section. Sentiment analysis was then performed using separate clusters, and the sentiment time series was thus created, which is used in the following section. Four different models were used for sentiment analysis to avoid the possible influence of individual sentiment analysis models, which were previously trained on different datasets. These models are discussed in the Materials

and Methods section. The general sentiment index (*SI*) for time t is calculated according to the formula below:

$$SI_t = \frac{1}{N_t} \sum_{j=1}^{4} \sum_{i=1}^{N_t} T_j(A_{it}) \quad (4)$$

where SI_t is the sentiment index at a point in time t; T_j, a sentiment analysis model (transformer), is used since the sum of the four models used in total is up to 4; $T_j(A_{it})$, the output, is given in the interval from 0 to 1; and A_{it} is the ith news article at time t, where i is in the interval from 1 to N_t and N_t is the number of news articles at a time t.

Below is a graphical representation of negative sentiment analysis for the business news category using only news article titles (see Figure 6). Based on the presented results, it can be observed that the negative sentiment toward knowledge increased, particularly during the period of economic crisis. A big jump is also observed at the beginning of the COVID-19 pandemic and the beginning of the war in Ukraine. These economic shocks can explain these changes in negative sentiment in business news. When a crisis, war, or pandemic starts, or when these events are anticipated, a higher number of negative news is observed in the information of business news. There are also discussions of various possible options, so negative sentiment can indicate upcoming shocks in economic activity as well. It is also important to mention the fact that this compiled index has a fairly high correlation with the indices previously compiled by other authors. For example, Baker et al. (2016) compiled the economic policy uncertainty index (EPU) [9]. Using the data available in this study, it was found that the correlation between the EPU index and the SI index obtained in the study is statistically significant. However, it is important to emphasize that the EPU index uses pre-defined words, whereas this work does not require this to calculate the index.

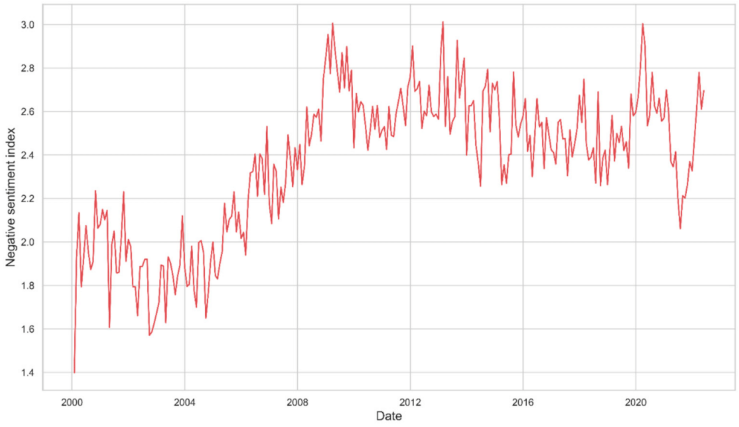


Figure 6. Graphical representation of negative sentiment analysis for the business news category using only news article titles.

Below is a graphic representation of negative sentiment analysis for the business news category using news titles and article lead information (see Figure 7). These results provide similar interpretations as the previous graphical representation. However, in this case, it can be observed that after the shocks, the negative sentiment decreases more. The most negative sentiment changes are seen in the same periods discussed earlier. Numerically, it is

observed that the negative sentiment is higher than when only using the textual information of the titles.

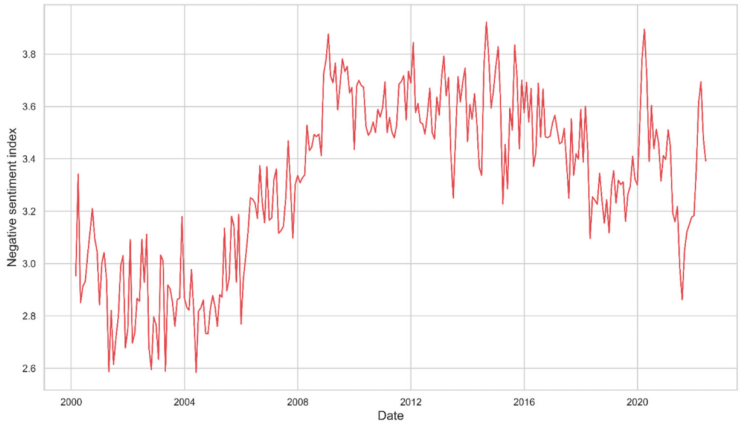


Figure 7. Graphic representation of negative sentiment analysis for the business news category using news article titles and article lead information.

As can be seen, only the negative sentiments of business news were presented, but during the study, the analysis was carried out with different categories. Therefore, the sentiment analysis results for these categories are presented in the graphs in the Appendix A (see Figures A1–A6).

5.3. Economic Activity Forecasting

This subsection presents the forecasting results of different indicators of economic activity. Conventional correlation analysis can be performed in the first forecasting stage. The table below (see Table 3) shows the results of the correlation analysis between the negative sentiments of different news categories and economic variables. Abbreviations in the table below (see Table 3) are as follows: UNY—youth unemployment rate, UNA—total unemployment rate, CS—consumer satisfaction, MI—monthly inflation rate, YI—annual inflation rate, and PI—output index. It can be noted that not all variables have statistically significant correlations in the presented table. It is observed that the youth unemployment rate decreases as the negative sentiment of foreign news, health news, and cultural news increases (more bad news). The same conclusions are also observed when adding the sentiments of Lithuanian news and science news and evaluating the overall unemployment level. These results can be interpreted so that when the number of negative news on news portals increases, employees are less inclined to leave their jobs and are more inclined to look for work. As we can see, there is no significant correlation with business news, so it can be assumed that these sentiments are perhaps not so crucial for business. It is observed that consumer confidence is negatively related to negative sentiment across categories. Arguably, the more negative news in the press, the less trust consumers have in companies. This can be related to various price increases in negative news about companies. The results show a positive and statistically significant relationship between news sentiment and monthly and annual inflation rates.

Table 3. Correlation of different categories of negative sentiment with the assessed indicators of economic activity.

Title 1	UNY	UNA	CS	MI	YI	PI
Overall sentiment	0.021	0.050	−0.200 ***	−0.009	−0.013	−0.012
Business sentiment	0.024	−0.067	−0.357 ***	0.045	0.148 *	−0.128 *
Lithuania sentiment	−0.119	−0.213 ***	−0.137 *	0.148 *	0.164 **	−0.087
Foreign sentiment	−0.212 ***	−0.259 ***	−0.650 ***	0.052	−0.113	0.068
Health sentiment	−0.184 **	−0.206 ***	0.114	0.071	0.096	−0.060
Science sentiment	−0.139 *	−0.192 **	0.014	0.148 *	0.327 ***	−0.012
Culture sentiment	−0.301 ***	−0.272 ***	−0.300 ***	−0.039	0.002	−0.075

*** correlation is significant at the 0.001 level; ** correlation is significant at the 0.01 level; * correlation is significant at the 0.05 level.

Interestingly, the negative sentiment of business news has a statistically significant relationship, but only with the annual inflation rate and not the monthly inflation rate. It is also noticeable that both the monthly and annual inflation rates have a statistically significant relationship with the negative sentiment of the Lithuanian news category. The production index has a statistically significant relationship with the negative sentiment of the business news category; as the negative sentiment increases, the production index decreases.

In the second stage of economic activity forecasting, different machine learning methods were applied to predict the obtained time series. The following table (see Table 4) presents the results obtained during the study. During the study, different neural networks were used for prediction: the simplest RNN, LSTM, and GRU. In order to find optimal prediction models, different parameters of the neural network were changed: the number of hidden layers of the neural network (h), the number of nodes of the neural network (n), and the learning rate of the neural network (lr). The number of hidden layers of the neural network changed from 1 to 10, the number of nodes of the neural network from 8 to 512, and the learning rate of the neural network from 0.001 to 0.1. Moreover, to generalize the model as much as possible, k-fold cross-validation was used, and the table below shows the average values of the metrics and their standard deviation. K-fold cross-validation for time series was carried out, like rolling estimation. For example, model training used 80 percent of the data (from the period beginning to the 80th percentile). Then, the model was tested for the next three months of the data. In the second cycle, 84 percent of data were used for training and the next three months for testing. Different metrics were calculated based on the testing data, and averages and standard deviations were calculated. Model tests like this one verify whether models are generalized for "good" periods and for different trends and seasonality periods. Different datasets have been used to forecast economic activity:

1. Univariate. In the univariate time series, only forecasted past values of the indicator are used.
2. Best sentiment. A univariate time series of the economic activity indicator and a time series of negative sentiments for individual categories are used. Here, the sentiment time series of the categories are used separately.
3. All sentiments. A univariate time series of the economic activity indicator and immediately the time series of all categories of negative sentiments are used.
4. Biggest cluster. The one-dimensional time series of the economic activity indicator and the time series of the negative sentiment of the largest cluster is used.

Table 4. Results of economic activity forecasting based on univariate time series and multivariate time series.

Data/Model	RMSE		MAE		MAPE	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**
Youth unemployment rate						
Univariate	**0.038**	0.010	**0.027**	0.006	**0.042**	0.009
Business sentiment	0.119	0.017	0.105	0.018	0.336	0.052
All sentiments	0.207	0.041	0.186	0.040	0.435	0.061
Biggest cluster	0.221	0.045	0.196	0.043	0.458	0.068
Overall unemployment rate						
Univariate	**0.045**	0.009	**0.035**	0.006	**0.047**	0.008
Business sentiment	0.121	0.023	0.110	0.025	0.338	0.061
All sentiments	0.264	0.023	0.256	0.025	0.533	0.040
Biggest cluster	0.267	0.035	0.278	0.037	0.576	0.045
Consumer satisfaction						
Univariate	0.079	0.010	0.057	0.007	0.079	0.009
Overall sentiment	**0.048**	0.005	**0.040**	0.003	**0.066**	0.006
All sentiments	0.119	0.022	0.095	0.021	0.138	0.036
Biggest cluster	0.125	0.023	0.098	0.023	0.147	0.037
Monthly inflation rate						
Univariate	0.187	0.010	0.143	0.008	0.340	0.015
Overall sentiment	**0.123**	0.003	**0.093**	0.002	**0.255**	0.008
All sentiments	0.209	0.010	0.162	0.006	0.365	0.032
Biggest cluster	0.208	0.013	0.158	0.007	0.356	0.045
Annual inflation rate						
Univariate	0.087	0.018	0.065	0.013	0.298	0.030
Overall sentiment	**0.062**	0.007	**0.053**	0.006	**0.283**	0.035
All sentiments	0.106	0.059	0.081	0.043	0.285	0.117
Biggest cluster	0.156	0.068	0.098	0.056	0.305	0.158
Production index						
Univariate	0.214	0.011	0.177	0.014	0.402	0.031
Lithuania sentiment	**0.099**	0.003	**0.076**	0.004	**0.166**	0.010
All sentiments	0.112	0.011	0.086	0.010	0.171	0.014
Biggest cluster	0.156	0.023	0.105	0.021	0.205	0.026

Bolded underlined values indicate the best obtained results.

Time series forecasting uses a time series lag of the economic indicator and negative sentiment from 1 to 12. The table below shows the univariate time series for each indicator of economic activity, the best negative sentiment for one category (the name of the best predictor category is given), the negative sentiment for all categories, and the highest cluster negative sentiment prediction results. This table shows only the results of the best models. A total of more than 20,000 different models were created during the study with different parameters and datasets. It can be seen that both the youth unemployment rate and the overall unemployment rate are best predicted with univariate time series. Although these variables have previously been correlated with category negative sentiment, time elutes do not provide such an advantage in predicting sentiment. It can be seen that negative sentiment-based forecasting outperforms one-dimensional forecasting across all metrics. In the case of clustering, only the most significant cluster was used, so the results obtained are worse than using single-category sentiment. When evaluating consumer confidence, it is observed that the forecasting of this economic activity indicator is better based on the general index of negative sentiment (comparisons with univariate time series). In this case, the average absolute percentage error is 1.3% lower. However, if all sentiments are included in the forecasting, instead of the best one, the forecasting deterioration is noticeable, and in

this case, the MAPE is 5.9% higher. It is noticeable that forecasting the monthly and annual inflation rate is thus best when the overall negative sentiment is used. The MAPE of the monthly inflation rate is as much as 8.5% lower, while the MAPE of the annual inflation rate is 1.5% lower. The output index shows the largest change in the forecast between the univariate time series and sentiment forecasting.

6. Discussion

Several main goals were set and implemented during the research, which were discussed in this paper. In the first phase of the study, a large amount of data was collected. This work collected information from two leading Lithuanian news portals (about 2.5 million articles). It is important to note that there are many more news portals in Lithuania, and this project's further development envisages more excellent information collection.

Further in this work, data clustering was performed, and it can be observed that data clustering with such a large amount of data does not work as well as expected at the beginning of the work. Only part of the expected models for clustering could be used in this research, but these are the most used models in practice. This allowed us to evaluate clustering's impact on news sentiment analysis and forecasting. Another important factor and limitation of this work is that the titles and leads of the articles were used in the work, but not the entire article's structure. Nevertheless, we could approve our sentiment impact on the economic activity hypothesis even with the title and lead sentiment analysis. A Lithuanian sentiment analysis model is also currently being developed, which would no longer require the additional translation of texts, and pure texts could be used to extract negative sentiments. In summary, the other results obtained during the study were expected, which supports the hypothesis that negative news sentiment is related to economic activity.

Furthermore, it was observed that negative news sentiment (in individual categories) increases when the economic situation worsens, e.g., with crises, the COVID-19 pandemic, or war. The determined correlation coefficients only further confirmed a statistically significant linear relationship between individual indicators of economic activity and individual categories of negative sentiments. Moreover, after applying the machine learning model to forecasting different economic activity indicators, it is observed that negative sentiment essentially helps to forecast economic activity better. Such results confirm the hypothesis raised during the work about the influence of negative news sentiments on economic activity. Additionally, in a future project, the more extensive use of different machine learning methods in forecasting is planned. Finally, it is essential to mention that low-frequency traditional data are mainly used for forecasting Lithuanian economic activity, and currently, alternative or Big Data are not so often used. Therefore, this study is an excellent start to better use the alternative data available in Lithuania which, as the study confirmed, can be applied to forecasting and refine forecasting compared to traditional data.

7. Policies Implications

The results obtained during the study confirmed that negative news sentiment, extracted using machine learning methods, has a significant relationship with different indicators of economic activity. The gained results may be helpful for government institutions in making timely policy decisions and evaluating policy implementation effectiveness, as the sentiment analysis by different categories provides more detailed information on different areas of the state, such as economy, business, health, and others. The gained results may be helpful for business companies as well, as negative news sentiment can also indicate further economic directions, which allows them to prepare for possible economic shocks, assess the market situation, and create a backup business model. For analysts and experts in the field, this research helps to evaluate the application of machine learning methods in natural language processing and economics. It helps to assess the difficulties of collecting a large amount of data, the need for processing, and the further possibilities of developing new methods. Further cooperation between the academic and business

community is possible based on the research results. It has also been observed that large amounts of freely available data create a significant number of new alternative economic variables for national banks and other institutions.

8. Conclusions and Future Research

This study proved the hypothesis that negative news sentiment is related to economic activity. Furthermore, negative news sentiment can be determined based on artificial intelligence methods or transformer structure models. Using negative sentiment in economic activation forecasting reduces model errors and makes more accurate forecasts.

However, this research is further expanded in several different directions: (1) the improvement of the dataset; (2) the application and comparison of different methods for evaluating news sentiment; (3) the development of different structured artificial intelligence models (transformers). Firstly, it is essential to note that many more news portals exist in Lithuania, and this project's further development envisages greater information collection. This would provide more data and more diverse categories. Furthermore, when evaluating data extraction and its quality and use, in the further stage of this project, it is expected to apply both textual information and visual information of articles. In order to solve this, in the further stages of the research, a comparison of various data dimensionality reduction methods is expected, which would allow clustering to be performed much more simply and without losing a large amount of information. Secondly, this research was based on transformer structure and did not use other authors' methodologies for comparison purposes. One of the future research fields refers to the different approaches comparison for the same task and mixed sentiment index creations based on the different approaches. Last but not least, the information in the full article was limited to the models used, subject to a maximum text length. However, further work aims to solve this limitation by dividing the text into parts and evaluating the negative sentiment of individual sentences/paragraphs or other parts of the sentence.

Author Contributions: Conceptualization, M.L., V.P., J.B., A.S., A.G. and T.R.; methodology, M.L., V.P., A.G. and T.R.; software, M.L. and T.R.; validation, V.P., J.B., A.S. and A.G.; formal analysis, M.L.; investigation, M.L., V.P. and A.G.; resources, M.L., V.P., J.B., A.S., A.G. and T.R.; data curation, M.L.; writing—original draft preparation, M.L. and A.G.; writing—review and editing, M.L., V.P., J.B., A.S., A.G. and T.R.; visualization, M.L.; supervision, V.P., J.B., A.S. and T.R.; project administration, V.P., J.B. and A.S.; funding acquisition, V.P., J.B. and A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This project has received funding from European Regional Development Fund (project No. 13.1.1-LMT-K-718-05-0012) under a grant agreement with the Research Council of Lithuania (LMTLT). Funded as European Union's measure in response to the COVID-19 pandemic.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank the area editor and the reviewers for giving valuable comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

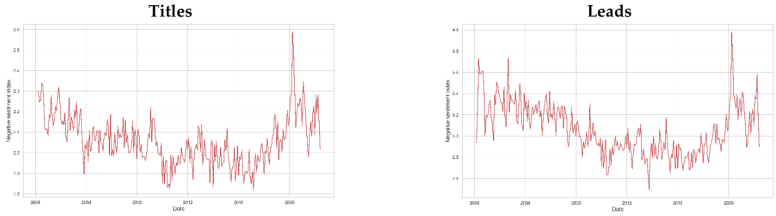


Figure A1. Sentiment analysis of news titles and leads over time for overall news.

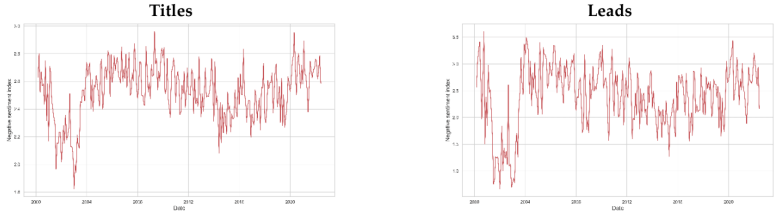


Figure A2. Sentiment analysis of news titles and leads over time for category "Lithuania" news.

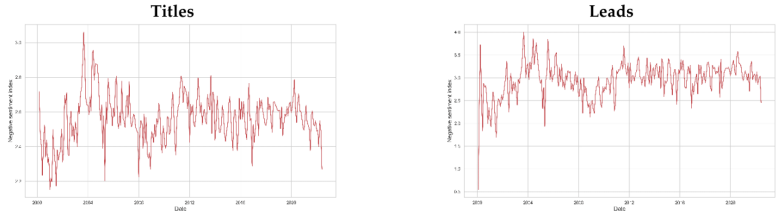


Figure A3. Sentiment analysis of news titles and leads over time for category "Foreign" news.

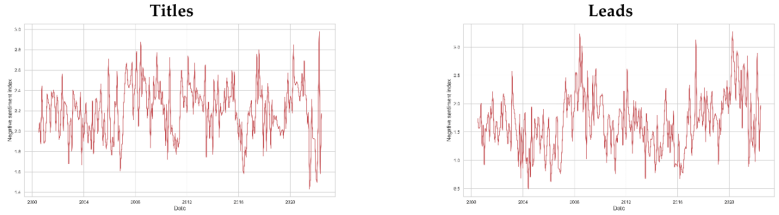


Figure A4. Sentiment analysis of news titles and leads over time for category "Science" news.

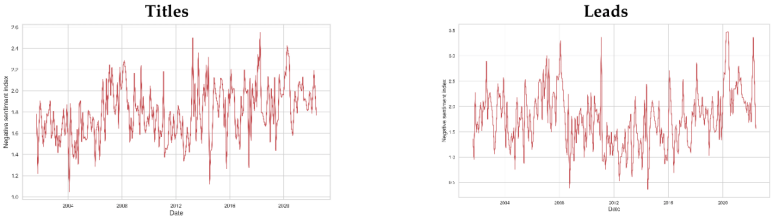


Figure A5. Sentiment analysis of news titles and leads over time for category "Culture" news.

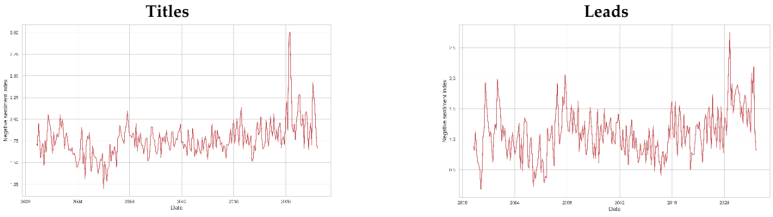


Figure A6. Sentiment analysis of news titles and leads over time for category "Health" news.

References

1. Alexakis, G.; Panagiotakis, S.; Fragkakis, A.; Markakis, E.; Vassilakis, K. Control of smart home operations using natural language processing, voice recognition and IoT technologies in a multi-tier architecture. *Designs* **2019**, *3*, 32. [CrossRef]
2. Ren, H.; Mao, X.; Ma, W.; Wang, J.; Wang, L. An English-Chinese machine translation and evaluation method for geographical names. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 139. [CrossRef]
3. Neto, A.F.d.S.; Bezerra, B.L.D.; Toselli, A.H. Towards the natural language processing as spelling correction for offline handwritten text recognition systems. *Appl. Sci.* **2020**, *10*, 7711. [CrossRef]
4. de Oliveira, N.R.; Pisa, P.S.; Lopez, M.A.; de Medeiros, D.S.V.; Mattos, D.M. Identifying fake news on social networks based on natural language processing: Trends and challenges. *Information* **2021**, *12*, 38. [CrossRef]
5. Zhang, L.; Wang, S.; Liu, B. Deep learning for sentiment analysis: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1253. [CrossRef]
6. Hussein, D.M.E.-D.M. A survey on sentiment analysis challenges. *J. King Saud Univ. Eng. Sci.* **2018**, *30*, 330–338. [CrossRef]
7. Taj, S.; Shaikh, B.B.; Meghji, A.F. Sentiment analysis of news articles: A lexicon based approach. In Proceedings of the 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Online, 30–31 January 2019; pp. 1–5.
8. Buckman, S.R.; Shapiro, A.H.; Sudhof, M.; Wilson, D.J. News sentiment in the time of COVID-19. *FRBSF Econ. Lett.* **2020**, *8*, 5.
9. Baker, S.R.; Bloom, N.; Davis, S.J. Measuring economic policy uncertainty. *Q. J. Econ.* **2016**, *131*, 1593–1636. [CrossRef]
10. Caldara, D.; Iacoviello, M. Measuring geopolitical risk. *Am. Econ. Rev.* **2022**, *112*, 1194–1225. [CrossRef]
11. Shapiro, A.H.; Sudhof, M.; Wilson, D.J. Measuring news sentiment. *J. Econom.* **2020**, *228*, 221–243. [CrossRef]
12. Sousa, M.G.; Sakiyama, K.; de Souza Rodrigues, L.; Moraes, P.H.; Fernandes, E.R.; Matsubara, E.T. BERT for stock market sentiment analysis. In Proceedings of the 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), Portland, OR, USA, 4–6 November 2019; pp. 1597–1601.
13. Jang, E.; Choi, H.; Lee, H. Stock prediction using combination of BERT sentiment Analysis and Macro economy index. *J. Korea Soc. Comput. Inf.* **2020**, *25*, 47–56.
14. Gite, S.; Khatavkar, H.; Kotecha, K.; Srivastava, S.; Maheshwari, P.; Pandey, N. Explainable stock prices prediction from financial news articles using sentiment analysis. *PeerJ Comput. Sci.* **2021**, *7*, e340. [CrossRef] [PubMed]
15. Galbraith, J.W.; Tkacz, G. *Nowcasting GDP with Electronic Payments Data*; 928991906X; ECB Statistics Paper; European Central Bank: Frankfurt am Main, Germany, 2015.
16. Bok, B.; Caratelli, D.; Giannone, D.; Sbordone, A.M.; Tambalotti, A. Macroeconomic nowcasting and forecasting with big data. *Annu. Rev. Econ.* **2018**, *10*, 615–643. [CrossRef]
17. Cooper, I.; Priestley, R. The world business cycle and expected returns. *Rev. Financ.* **2013**, *17*, 1029–1064. [CrossRef]

18. Baumeister, C.; Hamilton, J.D. Structural interpretation of vector autoregressions with incomplete identification: Revisiting the role of oil supply and demand shocks. *Am. Econ. Rev.* **2019**, *109*, 1873–1910. [CrossRef]
19. Herrera, A.M.; Rangaraju, S.K. The effect of oil supply shocks on US economic activity: What have we learned? *J. Appl. Econom.* **2020**, *35*, 141–159. [CrossRef]
20. Sampi Bravo, J.R.E.; Jooste, C. *Nowcasting Economic Activity in Times of COVID-19: An Approximation from the Google Community Mobility Report*; World Bank Policy Research Working Paper; The World Bank: Washington, DC, USA, 2020.
21. Diaz, E.M.; Perez-Quiros, G. GEA tracker: A daily indicator of global economic activity. *J. Int. Money Financ.* **2021**, *115*, 102400. [CrossRef]
22. Angelov, N.; Waldenström, D. The Impact of COVID-19 on Economic Activity: Evidence from Administrative Tax Registers. 2021. Available online: https://ssrn.com/abstract=3886818 (accessed on 20 August 2022).
23. Bricongne, J.-C.; Meunier, B.; Pical, T. Can Satellite Data on Air Pollution Predict Industrial Production? 2021. Available online: https://ssrn.com/abstract=3967146 (accessed on 20 August 2022).
24. Baldwin, R.; Di Mauro, B.W. Economics in the time of COVID-19: A new eBook. *VOX CEPR Policy Portal* **2020**, 2–3. Available online: https://fondazionecerm.it/wp-content/uploads/2020/03/CEPR-Economics-in-the-time-of-COVID-19_-A-new-eBook.pdf (accessed on 20 August 2022).
25. Chernis, T.; Cheung, C.; Velasco, G. A three-frequency dynamic factor model for nowcasting Canadian provincial GDP growth. *Int. J. Forecast.* **2020**, *36*, 851–872. [CrossRef]
26. Lourenço, N.; Rua, A. The Daily Economic Indicator: Tracking economic activity daily during the lockdown. *Econ. Model.* **2021**, *100*, 105500. [CrossRef]
27. Cavallo, A.; Diewert, W.E.; Feenstra, R.C.; Inklaar, R.; Timmer, M.P. Using online prices for measuring real consumption across countries. In *AEA Papers and Proceedings*; American Economic Association: Nashville, TN, USA, 2018; pp. 483–487. [CrossRef]
28. Mellander, C.; Lobo, J.; Stolarick, K.; Matheson, Z. Night-time light data: A good proxy measure for economic activity? *PLoS ONE* **2015**, *10*, e0139779. [CrossRef] [PubMed]
29. Kapetanios, G.; Papailias, F. *Big Data & Macroeconomic Nowcasting: Methodological Review*; Economic Statistics Centre of Excellence, National Institute of Economic and Social Research: London, UK, 2018. Available online: http://escoe-website.s3.amazonaws.com/wp-content/uploads/2020/07/13161005/ESCoE-DP-2018-12.pdf (accessed on 20 August 2022).
30. Fenz, G.; Stix, H. Monitoring the economy in real time with the weekly OeNB GDP indicator: Background, experience and outlook. *Monet. Policy Econ.* **2021**, *Q4/20–Q1/21*, 17–40.
31. Orihuel, E.; Sapena, J.; Navarro-Ortiz, J. An empirical algorithm for COVID-19 nowcasting and short-term forecast in Spain: A kinematic approach. *Appl. Syst. Innov.* **2021**, *4*, 2. [CrossRef]
32. Xin, M.; Shalaby, A.; Feng, S.; Zhao, H. Impacts of COVID-19 on urban rail transit ridership using the Synthetic Control Method. *Transp. Policy* **2021**, *111*, 1–16. [CrossRef]
33. Li, B.; Ma, L. Migration, transportation infrastructure, and the spatial transmission of COVID-19 in China. *J. Urban. Econ.* **2020**, *15*, 103351. [CrossRef]
34. Eraslan, S.; Götz, T. An unconventional weekly economic activity index for Germany. *Econ. Lett.* **2021**, *204*, 109881. [CrossRef]
35. Eckert, F.; Kronenberg, P.; Mikosch, H.; Neuwirth, S. *Tracking Economic Activity with Alternative High-Frequency Data*; KOF Working Papers; KOF Swiss Economic Institute, ETH Zurich: Zürich, Switzerland, 2020; Volume 488. [CrossRef]
36. Lewis, D.J.; Mertens, K.; Stock, J.H.; Trivedi, M. Measuring real activity using a weekly economic index 1. *J. Appl. Econom.* **2022**, *37*, 667–687. [CrossRef]
37. Fornaro, P.; Luomaranta, H. Aggregate fluctuations and the effect of large corporations: Evidence from Finnish monthly data. *Econ. Model.* **2018**, *70*, 245–258. [CrossRef]
38. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30. Available online: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf (accessed on 20 August 2022).
39. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI blog* **2019**, *1*, 9.
40. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
41. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
42. Pires, T.; Schlinger, E.; Garrette, D. How multilingual is multilingual BERT? *arXiv* **2019**, arXiv:1906.01502.
43. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
44. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. Albert: A lite bert for self-supervised learning of language representations. *arXiv* **2019**, arXiv:1909.11942.
45. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv* **2019**, arXiv:1910.13461.
46. Adhikari, A.; Ram, A.; Tang, R.; Lin, J. Docbert: Bert for document classification. *arXiv* **2019**, arXiv:1904.08398.

47. Liu, X.; He, P.; Chen, W.; Gao, J. Multi-task deep neural networks for natural language understanding. *arXiv* **2019**, arXiv:1901.11504.
48. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; Volume 32. Available online: https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html (accessed on 20 August 2022).
49. Gautam, A.; Venktesh, V.; Masud, S. Fake news detection system using xlnet model with topic distributions: Constraint@ aaai2021 shared task. In *International Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situation*; Springer: Cham, Switzerland, 2021; pp. 189–200.
50. Merchant, K.; Pande, Y. Nlp based latent semantic analysis for legal text summarization. In Proceedings of the 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 19–22 September 2018; pp. 1803–1807.
51. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M. Huggingface's transformers: State-of-the-art natural language processing. *arXiv* **2019**, arXiv:1910.03771.
52. Topal, M.O.; Bas, A.; van Heerden, I. Exploring transformers in natural language generation: Gpt, bert, and xlnet. *arXiv* **2021**, arXiv:2102.08036.
53. Gao, F.; Zhu, J.; Wu, L.; Xia, Y.; Qin, T.; Cheng, X.; Zhou, W.; Liu, T.-Y. Soft contextual data augmentation for neural machine translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5539–5544.
54. Li, X.; Fu, X.; Xu, G.; Yang, Y.; Wang, J.; Jin, L.; Liu, Q.; Xiang, T. Enhancing BERT representation with context-aware embedding for aspect-based sentiment analysis. *IEEE Access* **2020**, *8*, 46868–46876. [CrossRef]
55. Dang, N.C.; Moreno-García, M.N.; De la Prieta, F. Sentiment analysis based on deep learning: A comparative study. *Electronics* **2020**, *9*, 483. [CrossRef]
56. Khan, I.U.; Khan, A.; Khan, W.; Su'ud, M.M.; Alam, M.M.; Subhan, F.; Asghar, M.Z. A review of Urdu sentiment analysis with multilingual perspective: A case of Urdu and roman Urdu language. *Computers* **2021**, *11*, 3. [CrossRef]
57. Iglesias, C.A.; Moreno, A. Sentiment analysis for social media. *Appl. Sci.* **2019**, *9*, 5037. [CrossRef]
58. Hasan, A.; Moin, S.; Karim, A.; Shamshirband, S. Machine learning-based sentiment analysis for twitter accounts. *Math. Comput. Appl.* **2018**, *23*, 11. [CrossRef]
59. Araci, D. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv* **2019**, arXiv:1908.10063.
60. Malo, P.; Sinha, A.; Korhonen, P.; Wallenius, J.; Takala, P. Good debt or bad debt: Detecting semantic orientations in economic texts. *J. Assoc. Inf. Sci. Technol.* **2014**, *65*, 782–796. [CrossRef]
61. Huang, A.; Wang, H.; Yang, Y. FinBERT—A Deep Learning Approach to Extracting Textual Information. 2020. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3910214 (accessed on 20 August 2022).
62. Rosenthal, S.; Farra, N.; Nakov, P. SemEval-2017 task 4: Sentiment analysis in Twitter. *arXiv* **2019**, arXiv:1912.00741.
63. Lukauskas, M.; Ruzgas, T. A New Clustering Method Based on the Inversion Formula. *Mathematics* **2022**, *10*, 2559. [CrossRef]
64. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996; pp. 226–231.
65. Zhang, T.; Ramakrishnan, R.; Livny, M. BIRCH: An efficient data clustering method for very large databases. *ACM Sigmod Rec.* **1996**, *25*, 103–114. [CrossRef]
66. Ankerst, M.; Breunig, M.M.; Kriegel, H.-P.; Sander, J. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod Rec.* **1999**, *28*, 49–60. [CrossRef]
67. Caliński, T.; Harabasz, J. A dendrite method for cluster analysis. *Commun. Stat. Theory Methods* **1974**, *3*, 1–27. [CrossRef]
68. Davies, D.L.; Bouldin, D.W. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, 2, 224–227. [CrossRef]
69. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [CrossRef]
70. Aruoba, S.B.; Diebold, F.X.; Scotti, C. Real-time measurement of business conditions. *J. Bus. Econ. Stat.* **2009**, *27*, 417–427. [CrossRef]
71. Matheson, M.T. *Taxing Financial Transactions: Issues and Evidence*; IMF: Washington, DC, USA, 2011.
72. Brave, S.A.; Butters, R.A.; Justiniano, A. Forecasting economic activity with mixed frequency BVARs. *Int. J. Forecast.* **2019**, *35*, 1692–1707. [CrossRef]
73. Bai, J.; Li, K.; Lu, L. Estimation and inference of FAVAR models. *J. Bus. Econ. Stat.* **2016**, *34*, 620–641. [CrossRef]
74. Richardson, A.; van Florenstein Mulder, T.; Vehbi, T. Nowcasting GDP using machine-learning algorithms: A real-time assessment. *Int. J. Forecast.* **2021**, *37*, 941–948. [CrossRef]
75. Graves, A.; Mohamed, A.-r.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 6645–6649.
76. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
77. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.

78. Wang, Z.; Bovik, A.C. Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE Signal Process. Mag.* **2009**, *26*, 98–117. [CrossRef]
79. Willmott, C.J.; Matsuura, K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Clim. Res.* **2005**, *30*, 79–82. [CrossRef]
80. Glantz, S.A.; Slinker, B.K. *Primer of Applied Regression & Analysis of Variance*, 3rd ed.; McGraw-Hill, Inc.: New York, NY, USA, 2001; Volume 654.

applied sciences

MDPI

Article

Enhancing Skills Demand Understanding through Job Ad Segmentation Using NLP and Clustering Techniques

Mantas Lukauskas [1,*], Viktorija Šarkauskaitė [2], Vaida Pilinkienė [3], Alina Stundžienė [3], Andrius Grybauskas [3] and Jurgita Bruneckienė [3]

[1] Department of Applied Mathematics, Faculty of Mathematics and Natural Sciences, Kaunas University of Technology, 44249 Kaunas, Lithuania
[2] Independent Researcher, 51297 Kaunas, Lithuania
[3] School of Economics and Business, Kaunas University of Technology, 44249 Kaunas, Lithuania
* Correspondence: mantas.lukauskas@ktu.lt

Abstract: The labor market has been significantly impacted by the rapidly evolving global landscape, characterized by increased competition, globalization, demographic shifts, and digitization, leading to a demand for new skills and professions. The rapid pace of technological advancements, economic transformations, and changes in workplace practices necessitate that employees continuously adapt to new skill requirements. A quick assessment of these changes enables the identification of skill profiles and the activities of economic fields. This paper aims to utilize natural language processing technologies and data clustering methods to analyze the skill needs of Lithuanian employees, perform a cluster analysis of these skills, and create automated job profiles. The hypothesis that applying natural language processing and clustering in job profile analyzes can allow the real-time assessment of job skill demand changes was investigated. Over five hundred thousand job postings were analyzed to build job/position profiles for further decision-making. In the first stage, data were extracted from the job requirements of entire job advertisement texts. The regex procedure was found to have demonstrated the best results. Data vectorization for initial feature extraction was performed using BERT structure transformers (sentence transformers). Five dimensionality reduction methods were compared, with the UMAP technique producing the best results. The HDBSCAN method proved to be the most effective for clustering, though RCBMIDE also demonstrated a robust performance. Finally, job profile descriptions were generated using generative artificial intelligence based on the compiled job profile skills. Upon expert assessment of the created job profiles and their descriptions, it was concluded that the automated job advertisement analysis algorithm had shown successful results and could therefore be applied in practice.

Keywords: clustering; natural language processing; NLP; jobs requirements; machine learning; generative AI; GPT

Citation: Lukauskas, M.; Šarkauskaitė, V.; Pilinkienė, V.; Stundžienė, A.; Grybauskas, A.; Bruneckienė, J. Enhancing Skills Demand Understanding through Job Ad Segmentation Using NLP and Clustering Techniques. *Appl. Sci.* **2023**, *13*, 6119. https://doi.org/10.3390/app13106119

Academic Editors: Jerry Chun-Wei Lin, Gautam Srivastava and Stefania Tomasiello

Received: 1 April 2023
Revised: 13 May 2023
Accepted: 14 May 2023
Published: 16 May 2023

1. Introduction

Economic and geopolitical changes lead to increasing competition, globalization, demographic challenges, and digitization in almost all labor market areas. Digitization poses many challenges to businesses and employees regarding tasks previously performed by people, such as data entry, accounting, and work on the conveyor belt, which are now supported by digital technologies and artificial intelligence [1,2]. Digitization is predicted to only increase from here, meaning the challenges will also increase. For example, in the US, 47% of jobs are predicted to be automated in the coming decades [3]. In Lithuania, it is also noticeable that automation is increasing, and 40% of jobs already face significant changes after its introduction [4]. From an economic perspective, automation reduces company costs. It increases production, efficiency, and productivity, resulting in fewer people needed to achieve company goals.

However, due to the increasing digitization, another challenge arises—the demand for new professions and skills [5]. Automating a significant part of the employees' work creates a problem, as companies lack qualified employees with the appropriate skills to implement and maintain the latest technologies in the organization. There has been a significant shortage of skilled cyber security or data analysis workers. With the popularity and development of artificial intelligence, companies are increasingly choosing and using this tool in their business, meaning therefore more new jobs are emerging in IT, robotics, and industry. Technological changes already make it possible to see that some workers in the market have the necessary skills and can adapt more easily to the labor market.

In contrast, others lack the necessary skills [6]. We call skills the knowledge, qualities, and abilities of a person that can be learned. Learned knowledge, abilities, and acquired qualities enable people to successfully and systematically perform assigned tasks or activities [7]. Skills and their development are particularly important in enabling a country and its people to adapt and function successfully in an ever-changing world and labor market. Those who acquire strong, necessary skills are more innovative, efficient, confident, and have a higher quality of life.

In most cases, skill analysis is conducted through regular surveys. However, in this case, they can only be conducted during a certain time period and observing the dynamic changes in skills is difficult. One solution to better understand these required skills is the application of artificial intelligence methods in analyzing these skills. Developing artificial intelligence methods allow for extraction, process, and interpreting dynamic skill needs, whilst also enabling these processes' extremely fast and automated performance.

In recent years, the analysis of job requirements has received much attention—machine learning methods are increasingly used to extract and classify valuable information from job advertisements. Identifying key job requirements such as education, experience, skills, certifications, language proficiency, physical requirements, background checks, availability, personal characteristics, and legal eligibility can provide valuable insights for job seekers, employers, and policymakers [8]. Natural language processing techniques and machine learning algorithms can effectively classify and analyze the various requirements presented in job postings, thereby offering a data-driven approach to understanding the changing needs of the labor market [9]. For example, text mining and machine learning techniques have been used to study the prevalence of specific skills and qualifications in job postings, revealing trends in employer preferences and skill gaps across industries [10]. In addition, job requirement analyzes can help match job seekers with suitable employment opportunities, leading to more effective labor market outcomes and a better workforce development [11].

Despite the growing interest in analyzing job requirements using machine learning and natural language processing techniques, there remains a research gap in understanding the real-time assessment of job skill demand changes in the Lithuanian labor market. Our study aims to use natural language processing technologies and data clustering to analyze the skill needs of Lithuanian employees, perform a cluster analysis of these skills, and create new employee profiles. The primary objectives of this research are to explore the dynamics of specific skills, compare the different clustering methods, and identify the main profiles of employees in the Lithuanian labor market. The contributions of this study include providing valuable insights into the changing needs of the Lithuanian job market and offering a data-driven approach for assessing job skill demand changes in real time.

The remainder of this paper is organized as follows: Section 2 introduces the concept of employee requirements, definitions of different requirements, and their importance for employers, as well as the application of natural language processing in analyzing employee requirements. Section 3 presents the data used in the study, the methods, and the metrics for evaluating the results, including data processing methods, data dimensionality reduction methods, data clustering methods, and metrics for evaluating their results. Section 4 discusses the research findings, including the dynamics of specific skills, data dimensionality reduction outcomes, a comparative analysis of the different clustering

methods, and the main profiles of the employees. Finally, Section 5 concludes the paper, summarizing the main findings and outlining the potential avenues for future research.

2. Job Requirements and Natural Language Processing Application

Significant changes in the labor market due to economic, geopolitical, and digitization effects, as well as the COVID-19 pandemic, have lead to a greater need for work skills and work requirements [12]. Digitization has already changed several areas of the labor market, and it has been noticed that there is an increased demand for employees who have the necessary competencies to work with the latest technologies and specific programming languages. As digitization is rapidly increasing, there is already a perceived need for specialists with technical skills such as cyber security and data analysis. However, soft skills are no less important: critical thinking, communication, and emotional intelligence are all still valuable [13]. COVID-19 has led to a massive shift in the job market and accelerated remote work opportunities, with organizations allowing employees to work from home or abroad to adapt to this change. However, with the possibility of remote work, the need for strong digital literacy skills and effective virtual communication skills rose, and the need for more individual work grew [14]. As the experience of other countries in the world shows, skills and other job requirements are an integral component of a country's success [15]. Therefore, it is important to recognize these changes and conduct research accordingly so that educators, policymakers, and others can appropriately prepare the workforce. Research has shown that countries with a soft skills gap in the labor market are more efficient and innovative, have a better quality of life, attract more foreign investment, and build greater confidence [16]. It has also been noticeable that in countries where the analysis of workers' skills has not been conducted, there is a risk of a mismatch between the available workers' skills and the employers' needs [17]. There are several risks to the country's economy at this point. First, without clear information and understanding of what the exact skills the country's labor market needs are, this could lead to the investment of money and time in education and training programs that will not bring the desired result, as they do not meet the market's needs. Secondly, without clear information about what skills are needed today, employers cannot find suitable employees, which thereby reduces the company's productivity and economic growth [18]. In the last period, it was observed that employers often fill vacant job positions with foreign talent, but foreign talent is hard to come by, and is usually more expensive.

Employers require skills, abilities, and other requirements when looking for specialists for various positions. The labor market is changing rapidly. For example, a few years ago, soft skills such as communication, teamwork, and creativity were emphasized more [19]. However, soft skills and abilities are easier to learn and do not require specific training. In most cases, people have had such skills since childhood, so employers refer to soft skills as abilities in job advertisements [20]. In the labor market, certain skills and requirements are categorized as technical. These skills are acquired through specific learning and can be continually trained, improved, and expanded [21]. One challenge with technical skills is that employees are often reluctant to learn new abilities, leading to a shortage of specialists with up-to-date expertise. In the recruitment process, the requirements and skills mentioned in these job advertisements play a crucial role in determining whether a candidate possesses the necessary qualifications and qualities to be selected as the most suitable applicant. The specifications outlined in job advertisements hold significant importance throughout the selection and recruitment process, impacting both the job seekers and the employers.

Job seekers typically search for employment opportunities on various job posting portals, which often provide guidelines for employers regarding the information they must include about the position and the desired qualifications of potential candidates. However, these guidelines are not strictly enforced, leading to employers presenting job advertisements creatively, sometimes without listing the essential skills. Additionally, employers may not differentiate between the required skills, presenting them as part of the position's responsibilities. These factors all contribute to the job seeker's difficulty in

discerning the skills needed for a particular position. Conversely, employers might also struggle to prepare an effective job advertisement that accurately captures the requirements for a new position. In job advertisements, employers often specify certain requirements for potential candidates. Many employers mandate a minimum level of education, such as a bachelor's or master's degree in a relevant field of study [22].

There has been a growing demand for candidates with PhD degrees, particularly in sectors such as Fintech, artificial intelligence, and related fields. A candidate's educational background allows the employer to better understand the applicant's profile, and possessing a degree implies that the candidate has acquired the foundational knowledge, and is therefore likely capable of achieving good results in the relevant field [10]. Many employers seek candidates with specific technical skills related to the job, such as laboratory techniques, software expertise, or proficiency in various programming languages and tools. Evaluating technical skills enables the employer to determine whether the candidate can perform the assigned tasks and gauge their ability to learn new programming languages or tools quickly. Experience is often another requirement or advantage for gaining employment in a particular field, such as project management or data analysis projects. A candidate's work experience allows the employer to assess their existing skills and specialized knowledge. Typically, candidates with more work experience better understand the work environment [23]. Job postings often indicate the minimum years of experience required in a specific field. Although technical skills and experience are crucial, soft skills such as communication, collaboration, problem-solving, leadership, cultural fit, and a strong work ethic are also highly valued and sought after across all job fields. Employers actively search for candidates possessing suitable qualities and a cultural and value alignment with the organization, as these factors contribute significantly to the overall success of the company [24]. The above skills enable employers to evaluate a candidate's potential performance in their assigned tasks. While certificates and licenses may not be common requirements in all job advertisements, they are crucial for specific fields. For instance, medical professionals must possess a medical license and the necessary certificates to demonstrate their competencies and eligibility to work in a particular position. Certificates and licenses indicate a candidate's skills, experience, work quality, productivity, and suitability for a specific field [10].

The requirement for publications and research experience is not popular in job advertisements. However, this requirement is mandatory for candidates applying for academic and research-related positions. In order to occupy academic positions, a certain number of scientific research articles are often required. Employers often require knowledge of one or more languages besides their native language, especially if the work involves international clients, or the work environment is multilingual. Knowing multiple languages is a necessary skill in this era of globalization, as it facilitates communication between colleagues, partners, and clients. Knowledge of languages makes it possible to assess employees' ability to find information in another language, making work much easier [25]. One essential requirement for employees looking for work in the industry, construction, or production field is physical capacity, meaning in this case, the physical endurance needed to perform certain tasks, such as lifting weights, standing for prolonged periods, operating various machinery, and performing tasks properly and safely in compliance with all requirements. Some positions also require a background check. These are usually areas where employees can access confidential information or finances, such as public administration or finance. To get a job, a person must pass various background checks, which is important to maintain a safe working environment. Many jobs require the employee to be able to work nights, weekends, shifts, and holidays. Employers value the flexibility of candidates, which is important for the productivity and efficiency of the organization [26], and is especially valuable for areas such as industry, where work takes place around the clock. Moreover, legal eligibility is another important requirement, especially now that there are many workers from other countries in our country. This requirement indicates that the person must have a valid work visa and other necessary documents, according to which the individual has

the official permission to work in the country. It is an important requirement that ensures that employers do not face legal problems after hiring a candidate.

Job requirement analysis is becoming increasingly relevant in the current labor market. Understanding the required qualifications, skills, and attributes is crucial for job seekers, employers, and policymakers [8,27]. These requirements, which include factors such as education, experience, skills, certifications, language skills, physical requirements, background checks, availability, personal qualities, and legal eligibility, all play a critical role in determining the suitability of candidates for specific positions and ensuring the success of organizations [10]. An accurate assessment of job requirements is essential for job seekers to focus on acquiring their relevant qualifications and skills, ultimately enhancing their employment and career prospects [28]. For employers, clearly defined job requirements facilitate efficient recruitment procedures and select candidates with the right skills, thereby reducing turnover and improving overall workforce productivity. In addition, a deeper understanding of the job requirements can help policymakers design targeted education and training programs that address these skills gaps and promote workforce development, contributing to a more efficient and competitive labor market. Advances in machine learning, clustering, and NLP have improved our ability to analyze job postings and identify trends in the job requirements. Machine learning algorithms can be trained on large datasets of job postings to identify job requirements and their patterns. Clustering algorithms can group similar job postings according to their requirements, allowing for a more detailed analysis of the skills and qualifications required in a specific field. NLP techniques can extract information from unstructured text in job postings, such as required skills, education, and years of experience. By combining these methods, students can better understand the labor market and identify new trends and opportunities. Natural language processing (NLP) is a branch of artificial intelligence that develops algorithms and techniques to enable computers to understand, interpret, and generate human speech [29]. NLP has revolutionized text analysis by offering advantages over previous methods, such as increased efficiency, scalability, and the ability to uncover hidden patterns in enormous amounts of unstructured text data. Compared to manual text analyzes, NLP techniques can quickly process enormous quantities of data, making them ideal for sentiment analysis, topic modeling, machine translation, and information extraction applications. In addition, NLP techniques can capture complex linguistic structures and semantic relationships, allowing for a more accurate and detailed analysis of textual data. Despite its advantages, NLP also faces several challenges, including the inherent ambiguity and variability of natural languages, making it difficult for algorithms to accurately interpret the meaning and context of words and phrases [30]. Additionally, training NLP systems often require copious amounts of labeled data, which can be time consuming and expensive. Additionally, NLP models may struggle to accommodate the dynamic nature of language, which evolves and varies across domains, cultures, and communities [31].

3. Materials and Methods

This article subsection provides information about the data used in the study, their acquisition and processing, and the main characteristics of this data. This subsection also provides information about the main methods used in the research to solve the tasks of different stages of the research. Basic data processing methods, data clustering and evaluation metrics, natural language processing methods used in this study, and other possible methods were discussed. The main scheme of the study is shown in the figure below (see Figure 1), which helps to understand the main idea of this study. The following is a detailed description of the stages of the study.

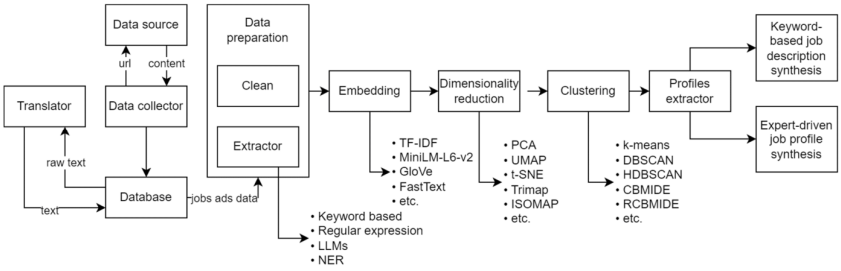


Figure 1. The main research scheme.

3.1. Data Gathering, Processing, Extraction, and Analysis

In order to collect the data required for this study, freely available Lithuanian job advertisements on the largest Lithuanian portals were used. Different Python libraries, including Playwright, BeautifulSoup, and Selenium, were used to collect the data. It is important to understand that the data collected was unstructured, requiring more processing than structured data. Moreover, due to the highly unstructured nature of the data, various data/advertisement structures were also possible, which only further complicates the work of data processing. The figure below shows an example of one of the possible data structures that was collected. The title section describes the name of the job position, and in rare cases, the exact job position was not written. However, a few words typically describe a certain activity, e.g., "Driver required", where it is not specified what driver is required. The location and company section contains information about the company and its location/city. The statistics section provides information on how many people have viewed the advertisement under analysis since it was posted. The nature of the work section describes the activities the employee will perform while working in this position in the company that submitted the advertisement. The requirements section contains the main requirements for an employee seeking to work in the analyzed position. The requirements are the main part of the analysis of this study, as the aim is to determine which skills are the most in demand, to create profiles of the advertisements presented, and to determine opportunities for employees to retrain for other positions. The offer section provides information about the employer's offer to the employee. Depending on the employer, additional benefits may be provided in this section, and a team description may also be provided. The last section would contain information regarding the salary and benefits that the employee will receive during additional work at this workplace. All collected data were stored in the created and protected PostgreSQL database, allowing for data collection and analysis in parallel.

Upon initial observation, the data seems to have a clear structure; however, in this example (see Figure 2), only a perfectly written job advertisement was displayed. In reality, the number of such well-structured job advertisements is quite limited. All components of the job advertisement are free form, meaning they may not always be present, and employers might label these components differently, such as "Requirements", "Required Skills", or "Competencies". Furthermore, these components can be placed in various sections of the job advertisement, potentially describing the required skills while only providing information about the job tasks. Consequently, significant uncertainties in data acquisition makes data extraction particularly challenging. Several different methods for extracting data from such texts are explored below.

Keyword-based search: this method searches for specific words or phrases in the analyzed text. This method can easily be used in common search tools. In this case, the big problem is that an initial set of keywords are required. Only specific words are searched

for, meaning when new requirements appear, adding the dictionary of the searched words is also required, which is unacceptable for a real-working system. One solution, in this case, is a synonyms search using WordNet [32], WordHoard (WordHoard Github repository: https://github.com/johnbumgarner/wordhoard, accessed on 25 March 20233), and others. A similar method is the rule-based method. In this case, the aim is to create rules based on certain linguistic patterns or parts of speech (POS) that reflect certain job posting requirements. For example, it is possible to find noun phrases that follow specific verbs (e.g., "require", "seek", and "need") or adjectives (e.g., "strong", "excellent", and "demonstrated").

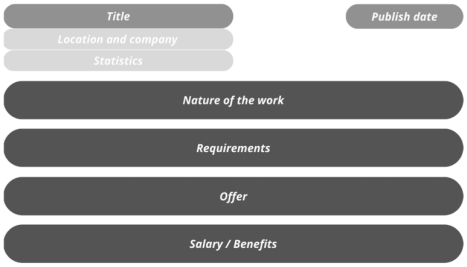


Figure 2. The ideal structure of the unstructured job advertisement text.

Another way to extract the required data from the text is regex. Regular expressions, abbreviated as regex, are powerful for pattern matching and text processing. A regex is a string of characters defining a specific search pattern. Regular expressions provide a concise and flexible way to extract information from unstructured data, such as log files, emails, and web pages. They can be implemented in various applications, including data validation, text extraction, and cleaning. While regular expressions can be complex and challenging to master, they are an essential tool for data scientists and software developers who need to work with copious amounts of text data. In this case, using regex makes it possible to find the requirements in job advertisements using patterns such as "Requirements" and the ending pattern "Company offers". For example:

"This is the job description. Requirements: experience, English etc. Company offers: health insurance, flexible work".

After using regex and the initial pattern "Requirements" and the final pattern "Company offers", the text "experience, English etc." is then extracted.

Another particularly developing way to extract the necessary information, systematize substantial amounts of textual information, or even generate additional textual data is large language models. LLMs are advanced machine learning models that are capable of processing and generating human-like language. These models are trained on massive amounts of textual data, such as books, articles, and websites, using complex algorithms that enable them to identify and learn patterns in language. Some examples of LLMs include OpenAI's GPT-4 (OpenAI GPT-4 information: https://openai.com/product/gpt-4, accessed on 25 March 2023), GPT-3.5-turbo (OpenAI GPT-3.5-turbo information: https://platform.openai.com/docs/models/gpt-3-5, accessed on 25 March 2023), GPTNeoX [33], and Google's BERT [31]. LLMs have many applications, including in natural language processing, machine translation, and chatbot development. They are becoming increasingly popular in artificial intelligence and have the potential to revolutionize how humans interact with computers and machines. Using the above example and extracting the necessary data using this methodology, the request looks like this:

"Extract job requirements from the text provided:

This is the job description requirements: experience, English etc. The company offers health insurance and flexible work".

The results obtained after using OpenAI's GPT-3.5-turbo are:

"Based on the text provided, the job requirements are experience (specifics not mentioned) English proficiency (specifics not mentioned). It is important to note that without further context, it is difficult to determine the level of experience or proficiency required for the job".

At the time of research, the price of GPT-3.5-turbo was $0.002 per 1000 tokens, where 1000 tokens are about 750 words. The results obtained after using Open-AI's GPT-4 look like this: "*From the provided text, the job requirements are 1. Experience 2. English proficiency. Additionally, the company offers: 1. Health insurance, 2. Flexible work*". At the time of research, GPT-4 was $0.03 per 1000 prompt tokens and $0.06 per 1000 completion tokens, making it even about 15 times more expensive.

Given that the collected data contained about 120 million words and the answers' size would also be similar, the required number of tokens would be 320 million. For instant research, it is a more expensive method than the one mentioned above. A considerable amount of prompt engineering was also required for the model to answer what was being asked and to provide the results in the desired format.

Finally, the last method discussed to identify the required information from the text was named entity recognition (NER). NER is a fundamental task NLP that involves identifying and classifying named entities, such as persons, organizations, locations, and other specific terms in the unstructured text [34]. In the context of job advertisement requirement analysis, NER can play a crucial role in extracting the relevant information regarding the desired qualifications, skills, and other attributes employers seek in potential candidates. By automatically recognizing and categorizing the key entities mentioned in the job advertisements, NER can help streamline the process of analyzing large numbers of job postings, thereby enabling researchers, job seekers, and employers to gain insights into the dynamics of the job market and the most in-demand skills and qualifications. Furthermore, NER techniques can be combined with other NLP and machine learning methods, such as topic modeling and clustering, to group job advertisements based on the extracted requirements, as well as identify the common patterns, trends, and skill gaps across various industries and job roles.

In summary, applying named entity recognition to job advertisement requirement analysis can provide valuable information for job seekers, employers, and educational institutions, thereby helping them to make informed decisions and adapt to the ever-changing job market. However, in order to use this methodology in this research, a specially trained model was needed to determine the job advertisement's requirements. To train the model, a large amount of data was required, which was available in this work, but these data were not labeled. For this reason, the application of NER in this study was postponed in further project plans, which are discussed in the subsection future directions.

This work relies on keywords-based (for the visualization of the specific skills) and regex methodology, which was already presented and mentioned earlier, and during the research, these data were prepared for the training of the NER model, which can be applied in further research.

3.2. Text Vectorization

After cleaning the data from unnecessary information, the textual data must be vectorized for analysis, clustering, and other research tasks. Text vectorization converts the text data into numerical data for later utilization in machine-learning techniques [35]. Text vectorization is a particularly crucial step in natural language processing in classification, clustering, information extraction, and sentiment analysis. It allows machine learning algorithms to operate on the text data in a numerical format. The main idea of text vectorization is that each text document (a sentence, paragraph, or entire document) is recorded with

numerical data while preserving the main information of these documents. An essential characteristic of text vectorization is preserving information as much as possible after this process. Various methods were used for text vectorization, such as:

Term frequency-inverse document frequency (TF-IDF) is a popular text vectorization method. TF-IDF computes the importance of each term in a document relative to a collection of documents [36]. It is based on the idea that terms frequently appear in a specific document. However, it is rare to carry more discriminative information across the entire document collection. The resulting term-weighted document vectors can be used for tasks such as document classification and information retrieval. Although TF-IDF might not capture semantic meaning as effectively as sentence transformers, it is computationally efficient. It has been proven useful for various text analytical applications:

$$w_{i,j} = tf_{i,j} \times log\left(\frac{N}{df_i}\right) \quad (1)$$

where $tf_{i,j}$—frequency of the i in the j, df_i—number of the documents containing i, and N—total number of the documents.

Another important method that has recently received a lot of attention is sentence transformers. Sentence transformers, introduced by Reimers and Gurevych (2019) [37], employ pre-trained transformer models such as BERT [31] to generate dense vector embeddings of sentences. These embeddings capture the semantic meaning of the input text. They can be used for various tasks, including text classification, semantic similarity, and clustering. The advantage of sentence transformers is that they can capture complex linguistic structures and relationships within the text, thereby improving the performance on various NLP tasks. BERT is a pre-trained transformer model that utilizes self-attention mechanisms to capture contextual information in both directions (left-to-right and right-to-left) [31]. Fine-tuning BERT on specific tasks can generate high-quality sentence or document embeddings, which are effective for various NLP tasks such as sentiment analysis, text classification, and semantic similarity. Advantages of these pre-trained transformer models are as follows: captures context and word order at the sentence or document level, can be fine-tuned on specific tasks, yielding high-quality embeddings, and that it can outperform traditional methods such as TF-IDF and Word2Vec in numerous NLP tasks. However, disadvantages of these pre-trained models include that it requires substantial computational resources for training and fine-tuning, generates high-dimensional embeddings, which may necessitate dimensionality reduction techniques, and that these models are characterized with a complex architecture and larger model size compared to other methods. Another method that can be compared with the sentence transformers is OpenAI Ada embedding.

However, although the previously mentioned methods are among the main methods for vectorizing textual data, other methods can be used to implement these tasks. Among the alternative methods, we can mention other, more frequently used methods:

- Word2Vec, proposed by Mikolov et al. (2013) [38], is a popular word embedding technique that generates dense vector representations of words by predicting the context of a given the word using an external neural network. These embeddings capture the semantic and syntactic relationships between words. The word vectors can be averaged, summed, or combined for a sentence or document-level representation using more sophisticated techniques including weighted averages or the smooth inverse frequency method [39]. Word2Vec consists of two primary architectures: Continuous bag of words (CBOW) and Skip-Gram. CBOW predicts a target word based on its surrounding context, while Skip-Gram predicts the context given a target word. There are several limitations to the Word2Vec method: It requires substantial computational resources for training on large corpora. It focuses on word-level embeddings, which may not capture sentence- or document-level semantics. Despite these limitations, Word2Vec does possess several benefits: It captures semantic and syntactic relationships between words. Generates dense continuous vectors, reducing dimensionality

compared to sparse methods such as TF-IDF. Pre-trained models are available for various languages and domains.

- GloVe (Global Vectors for Word Representation) is another word embedding method introduced by Pennington et al. (2014) [40]. It generates word embeddings based on the global co-occurrence statistics of words in a corpus. Similar to Word2Vec, GloVe embeddings can be aggregated to create a sentence or document-level representations similarly.
- The bag-of-words (BoW) model is a simple text vectorization method that represents documents as fixed-size vectors based on the frequency of words they contain [41]. While BoW does not capture the order of words or semantic relationships, it is computationally efficient. It can be effective for certain text analytical tasks.
- Developed by Bojanowski et al. (2017) [42], FastText is an extension of the Word2Vec model that generates embeddings for sub-word units (such as N-grams) instead of entire words. This approach enables capturing of morphological information and better handling of the rare and out-of-vocabulary words. The sentence or document-level embeddings can be obtained by aggregating the sub-word embeddings.
- Doc2Vec, also known as paragraph vectors, is an extension of Word2Vec introduced by Le and Mikolov (2014) [43]. It generates dense vector representations for entire documents by considering both the words and the document as an input during the training process. This method can capture the overall semantic meaning of a document and can be directly used for document-level tasks.

In summary, various text vectorization methods can serve as alternatives to sentence transformers and TF-IDF, each with their strengths and limitations. The choice of the appropriate method depends on factors such as the specific NLP task, data characteristics, and computational resources. In this work, we used sentence transformers models for the initial data vectorization and feature extraction.

3.3. *Dimensionality Reduction Methods*

Vectorization of textual data often results in extremely large matrices. In the case of sentence transformers, the resulting matrix is Nx364. In contrast, the TF-IDF matrix depends on how many different words are in the data and which N-gram is used. Data dimensionality reduction plays a particularly significant role in data analysis and various machine learning techniques by reducing high-dimensional datasets to low-dimensional datasets that contain only the most valuable information and relationships. Large datasets can be challenging to work with as they often contain noise, redundancy, and sparsity, making it difficult to identify patterns and relationships in the data. When performing data clustering, a particularly enormous number of dimensions is not desirable if these dimensions do not provide additional information. Data dimensionality reduction is an important preprocessing step prior to data clustering, as it helps overcome the challenges many dimensions face. This phenomenon is often known as the "curse of dimensionality" [44]. As sparsity in high-dimensional spaces increases, many dimensions can lead to the overfitting and poor clustering of the results. With the utilization of data dimensionality reduction, it has been noticeable that better data clustering results are obtained, and such results are much easier to interpret [45]. Data dimensionality reduction techniques have also been well used to reduce noise in data [46], or detect outliers in the data [47], and improve the accuracy and generalization of the models being developed [48]. Data dimensionalities reduction methods such as principal component analysis (PCA), t-distributed stochastic neighbor embedding (t-SNE), and uniform manifold approximation and projection (UMAP) transform the original high-dimensional data into smaller-dimensional data, while also retaining as much as possible, more similar data structures and relationships between the individual data points [49]. These data dimensionality reduction techniques allow clustering methods such as k-means and DBSCAN to identify clusters in the data more efficiently. Moreover, data dimensionality reduction methods significantly reduce computational complexity and memory requirements for clustering methods, thus enabling

them to perform actions with larger datasets. However, it is important to note that data dimensionality reduction methods also have their disadvantages. One is that a certain amount of information can be lost when the dimensionality is reduced [50].

The main data dimensionality reduction techniques used in this study are briefly discussed below. One of the most widely used and widely known data dimensionality reduction techniques is principal component analysis (PCA). The principal component analysis is a linear method that projects data into a space of smaller dimensions to maximize variance and, at the same time, minimize information loss. PCA identifies and extracts the principal components, which are orthogonal linear combinations of the original variables, capturing the maximum variance in the data while minimizing information loss [51]. The principal components are ranked according to their variance. Hence, the first principal component captures the largest variance, the second principal component the second largest variance, and so on. By retaining only a few principal components with a higher variance, PCA can effectively reduce the dimensionality of the data while retaining most of the information. PCA has been successfully applied in various domains, including image processing, bioinformatics, and finance, for feature extraction, data compression, visualization, and noise reduction [52,53]. Recent advances in PCA include incorporating regularization techniques, such as sparse PCA and elastic net PCA, which add constraints to improve the interpretability and robustness of the model, particularly in high-dimensional data with multicollinearity issues [54]. Despite its advantages, PCA has certain limitations, such that it assumes linearity in the data, which may not always be appropriate, especially when the data exhibits complex nonlinear structures.

Another data dimensionality reduction method that, in contrast to principal component analysis, is not linear, is t-distributed stochastic neighbor embedding (t-SNE). This method has been widely used due to its ability to maintain local and global relationships in high-dimensionality reductions, thus maintaining exceptionally good data visualization and interpretation [55]. This method has gained much attention recently due to its ability to effectively visualize large-dimensional data by representing it in smaller dimensions [56]. The t-SNE algorithm achieves this by converting high-dimensional Euclidean distances into conditional probabilities, representing similarities between data points. Then, a gradient descent optimization method minimizes the divergence between these probabilities across dimensions [57]. Researchers have proposed efficient approximations, such as the Barnes-Hut t-SNE, which allows the algorithm to manage large datasets while maintaining its effectiveness in preserving local structures. T-SNE has been widely applied across various scientific domains, including single-cell analysis, bioinformatics, and computer vision. Despite its success, certain limitations of t-SNE have been identified, such as sensitivity to initial conditions and hyperparameters, slow convergence, and the presence of local optima [56].

Uniform manifold approximation and projection (UMAP) is a newer data dimensionality reduction method. Similar to t-SNE, it is nonlinear, and can maintain local and global relationships. Data dimensionality reduction improves data visualization and interpretation [49]. UMAP is built on the foundations of multivariate learning and topological data analysis, using concepts such as simple fuzzy sets to model the underlying geometry of the data [58]. UMAP creates fuzzy topological representations of high-dimensional data and projects them into a lower-dimensional space. The resulting projection preserves the local structure of the data, meaning that nearby points in high-dimensional space are also close to each other in the low-dimensional projection. UMAP is similar to other dimensionality reduction methods, such as t-SNE and PCA, but has several advantages over these methods. For example, UMAP is generally faster and more scalable than t-SNE and can handle larger datasets. In addition, UMAP can be used with various distance metrics, including non-Euclidean metrics that allow for capturing complex data relationships. UMAP's preservation of local and global structures facilitates data visualization and interpretation. It is suitable for various applications, including single-cell RNA-seq data analysis [58], image classification, and natural language processing tasks [59]. In addition,

UMAP has shown faster execution times than t-SNE, making it more suitable for large-scale datasets [49]. Recent advances in UMAP include the development of supervised and semi-supervised variants that incorporate label information into the dimensionality reduction process, resulting in improved class separation and more meaningful embeddings. Despite its advantages, UMAP can exhibit some limitations, especially in cases where assumptions about the underlying data collector do not hold, or when the data show high noise levels. In such situations, alternative dimensionality reduction methods may be better.

In this work, we assessed different dimensionality reduction methods listed in this section and different method performances presented in the results sections.

3.4. Clustering Methods Used in the Research

The following section provides an in-depth overview of the clustering algorithms that were employed to analyze the dataset under investigation. Clustering is a fundamental technique in unsupervised machine learning. It offers valuable insights into data's inherent structure and relationships by grouping similar objects into clusters based on their features. Selecting appropriate clustering algorithms is crucial for obtaining accurate and meaningful results. Each method has its strengths and weaknesses depending on the data's specific characteristics and the analysis's objectives.

In this research, we have carefully chosen a diverse set of clustering algorithms, including hierarchical, partitioning, density-based, and model-based methods, to provide a comprehensive understanding of the underlying patterns in the data. Each clustering method offers a unique perspective on the data organization. By comparing their results, we aimed to derive robust conclusions and minimize potential biases associated with any algorithm [60]. This section will outline the principles and rationale behind each clustering method used in the study, discuss their respective strengths and limitations, and provide a detailed explanation of their implementation within the context of this research. Furthermore, we will describe the process of selecting the optimal number of clusters and present the unsupervised evaluation metrics employed to assess the quality of the clustering results obtained from each algorithm.

The main data clustering techniques used in this study are discussed below. K-means clustering is a widely used unsupervised learning technique that aims to partition unlabeled data into distinct clusters based on their inherent features [61]. K-means clustering operates by iteratively assigning data points to a predefined number of cluster centroids based on the minimization of within-cluster distances, typically using Euclidean distance as the similarity measure. The algorithm initializes by selecting random or strategically placed centroids. It then iteratively refines their positions until convergence, resulting in compact and well-separated clusters. One of the main advantages of K-means is its simplicity and ease of implementation, which makes it computationally efficient and suitable for large datasets. However, the algorithm has limitations, such as sensitivity to the initial centroid positions and the requirement to specify the number of clusters a priori, which may not always be known or easily determined [62].

Additionally, K-means is predominantly suited for detecting spherical and equally sized clusters and may struggle with more complex data. Recent advancements in this field have expanded its applicability to various domains, including text analytics. For instance, K-means has been employed in document clustering, grouping similar textual information, and enhancing information retrieval efficiency.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm that identifies clusters by grouping data points that are tightly packed based on a specified distance metric and density threshold [63] (Ester et al., 1996). This approach allows DBSCAN to effectively manage clusters of arbitrary shapes and noise in the data, which is a significant advantage over traditional partitioning methods such as K-means. However, DBSCAN is sensitive to its hyperparameters, namely the neighborhood radius (Eps), and the minimum number of points required to form a dense region (MinPts), which can be challenging to determine a priori [63]. HDBSCAN (hierarchical density-based

spatial clustering of applications with noise) is an extension of DBSCAN that addresses some of its limitations by incorporating a hierarchical clustering approach [64]. HDBSCAN does not require the specification of a global density threshold; instead, it automatically identifies clusters at varying densities by constructing a dendrogram and applying a cluster extraction method based on the stability of clusters over different density levels. This results in a more robust and flexible clustering algorithm adapting to varying data distributions and densities.

Nevertheless, both DBSCAN and HDBSCAN may suffer from high computational complexity, especially for large datasets, and are sensitive to the choice of a distance metric, which can significantly impact clustering performance. DBSCAN and HDBSCAN have been applied to various text analytical tasks, leveraging their ability to detect clusters of arbitrary shapes and varying densities. For instance, DBSCAN has been employed in topic modeling. It can identify coherent thematic groups within large collections of documents, improving the organization and retrieval of textual information. Similarly, HDBSCAN has been proven valuable in text summarization, enabling the extraction of representative sentences from a given document while maintaining the diversity of information and covering various aspects of the content. These applications demonstrate the versatility of density-based clustering methods in text analysis, offering unique advantages in handling complex data distributions and capturing nuanced relationships within textual data.

BIRCH (balanced iterative reducing and clustering using hierarchies) is a hierarchical clustering method designed to efficiently process large datasets by constructing a tree structure called the clustering feature tree (CF-Tree) that captures the essential attributes of data points, such as their linear sum and squared sum [65]. The algorithm can manage large datasets by incrementally processing data points and adjusting the tree structure dynamically, significantly reducing computational complexity and memory requirements compared to the traditional hierarchical clustering methods. Recent applications of BIRCH in text analysis include document clustering. The method can group similar texts based on their feature representations, such as term frequency-inverse document frequency (TF-IDF) vectors. BIRCH has also been employed in analyzing customer reviews, enabling the identification of patterns and trends in customers' opinions, which can inform businesses about their strengths and areas for improvement.

Furthermore, BIRCH has been utilized in social media analytics for event detection. The algorithm can cluster textual data from social media platforms to identify significant events or emerging discussion topics. These applications demonstrate the potential of BIRCH as an efficient and scalable clustering method for text analysis tasks, particularly in handling large-scale textual data.

Affinity propagation is a clustering algorithm based on message passing that identifies a set of exemplar data points, which best represent the clusters, and groups similar data points around them [66]. Unlike many other clustering methods, affinity propagation does not require the number of clusters to be specified a priori, making it more adaptive to various data distributions. The algorithm works by iteratively exchanging real-valued messages between data points until their convergence, reflecting the suitability of each point to serve as an exemplar, and the preference of each point to select a specific exemplar. In the context of text analysis, affinity propagation has been applied to various tasks, such as document clustering, where it can effectively group similar documents based on their content or feature representations (e.g., TF-IDF vectors) [67]. Additionally, affinity propagation has been employed in analyzing social media data, such as community detection in social networks based on user-generated content, enabling the discovery of meaningful user relationships [68]. Despite its adaptability, affinity propagation may suffer from high computational complexity, particularly for large datasets, and sensitivity to the choice of preference parameter, which influences the number of exemplars. Nevertheless, the algorithm's ability to automatically determine the number of clusters and its robustness to noise make it a valuable method for various text analysis applications.

Spectral clustering is a technique that leverages the spectral properties of the data's similarity matrix to perform clustering in a lower-dimensional space, thus enabling the detection of complex structures and non-linear relationships within the data [69]. The algorithm first constructs a similarity graph, where nodes represent data points, and edges represent pairwise similarities, typically using Gaussian kernel or k-nearest neighbors. It then computes the eigenvalue decomposition of the graph's Laplacian matrix, clustering the eigenvectors corresponding to the k smallest eigenvalues using traditional clustering methods, such as K-means. Spectral clustering has been applied in text analysis tasks to various scenarios, including document clustering. It can efficiently group similar documents based on their feature representations, such as term frequency-inverse document frequency (TF-IDF) vectors [70]. However, Spectral clustering has some limitations, such as sensitivity to the choice of similarity measure and the requirement to specify the number of clusters. The method can also be computationally expensive, especially for large datasets, due to the eigenvalue decomposition step.

CBMIDE (clustering based on the modified inversion formula density estimation) [71] and RCBMIDE (reduced clustering based on the modified inversion formula density estimation) [72] are novel clustering methods that leverage the modified inversion density estimation to effectively capture the underlying structure of the data. In contrast to the traditional methods such as the Gaussian mixtures models, CBMIDE focuses on the reciprocal distance between the data points and the cluster centers, thereby providing a more robust estimation of the density structure. By utilizing this alternative approach to density estimation, CBMIDE has the potential to reveal complex relationships and structures within the data that conventional methods may overlook. CBMIDE can be applied in text analysis to various tasks, such as document clustering. It can efficiently group similar documents based on their feature representations, such as the term frequency-inverse document frequency (TF-IDF) vectors. It is also worth noting that this clustering method exhibits robustness, ensuring that only legitimate data clusters are identified and analyzed, thereby minimizing the influence of noise and outliers on the overall results, and enhancing the accuracy and reliability of the findings.

In this work, we evaluated all the abovementioned clustering methods and compared these methods' performances based on the different metrics. All results are presented in the results sections.

4. Results

This subsection presents the main results obtained during the study. First, the data collected during the research and the dynamics of the data collected during the research were reviewed. The second subsection of this chapter presents the data preparation and feature extraction used in this research. The third subsection of this chapter presents the results of different methods of data dimensionality reduction and data clustering, allowing us to decide on further methods in this study. The fourth subsection of this chapter presents the data clustering results and the interpretation of these results, which are the main insights.

4.1. Dynamic of the Specific Requirements/Keywords in Lithuania

The rapid evolution of technology and industries has resulted in the job market's continuous shift of skill requirements. It is therefore essential to understand these changing dynamics to help job seekers, employers, and educational institutions adapt to the evolving landscape. In this section, we present an exploratory data analysis (EDA) of job advertisements over time, focusing on the prevalence of multiple keywords that represent specific job requirements. By extracting these keywords from job adverts and tracking their frequency over time, we aimed to uncover the trends and patterns that reflect the growing or declining importance of various skills and qualifications across different industries and job roles.

The analysis includes a series of graphs for each keyword, illustrating its frequency in job adverts over time. These visualizations enabled us to identify emerging trends, such as

the increasing demand for certain programming languages, data analysis tools, soft skills, and shifts in industry-specific requirements. Comparing these trends across multiple keywords also revealed the relative importance of various skills and qualifications in the job market, thereby providing valuable insights for job seekers, employers, and educational institutions.

By understanding the dynamic nature of job requirements, job seekers can make informed decisions regarding the skills they should acquire or further enhance. At the same time, employers can design more targeted job adverts and recruitment strategies. Moreover, educational institutions can adjust curricula to better prepare students for the evolving job market, ensuring that graduates have the most relevant and in-demand skills.

$$F_{k,t} = \frac{\sum_{i=1}^{n} D_{k,t,i}^{T}}{n} \quad (2)$$

where $F_{k,t}$—frequency of the specific keyword k in the time moment t, n—total number of the documents, and $D_{k,t,i}^{T}$—documents that contain specific keyword k at the moment t.

The graph below (see Figure 3) provides information about the demand for selected programming languages in job advertisements by analyzing the requirements of job advertisements. Based on the presented graph, the SQL programming language is in the greatest demand. However, at the same time, it was also noticeable that in the last period, the demand for this programming language, similar to the other programming languages, has decreased quite strongly.

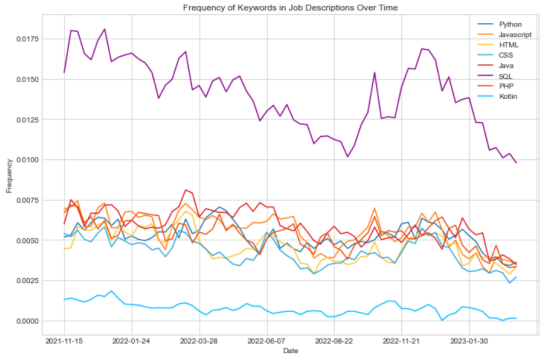


Figure 3. Different selected programming language demand changes over time.

4.2. Feature Extraction of the Job Adverts

After collecting the information, the relevant information was extracted from these job adverts—the job requirements, the publication date necessary to determine dynamic changes, and the name. After comparing the methods discussed earlier, the best results and the simplest implementation were observed using the regex procedure and distinguishing certain job requirements section names. It is well known that using method-generated synonyms makes this implementation simpler, as it reduces possible different combinations. After extracting the information significant for the study from the data, feature extraction was performed. The previously mentioned sentence transformers (BERT structure) method, whose dimensions are 384, was found to have performed this function best.

4.3. Comparative Analysis of Dimensionality Reduction Results

After performing feature selection, the size of the data dimensions obtained, as mentioned earlier, was 384. With such large data dimensions, data clustering required much

larger resources. Several methods, such as CBMIDE, are not adapted to such dimensions. The following work uses data dimensionality reduction methods based on these insights. Different data dimensionality reduction methods have different properties and perform differently depending on the dataset. In order to properly evaluate data dimensionality reduction, metrics were needed to enable this. Estimating the dimensions of the data allow for determining which new dataset was the best, and how many new dimensions were needed to retain the maximum amount of information possible. One of the possible evaluation metrics is the trustworthiness metric [73]. This metric assesses how well the information and relationships between observations in the original and reduced dimensions are preserved. The computation of this metric was accomplished by utilizing the following formula:

$$T(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{j \in N_i^k} \max(0, (r(i,j) - k)) \quad (3)$$

where for each observation i, N_i^k are its k nearest neighbors (the output space obtained after applying the methods), and for each observation j, $r(i,j)$ was its original data space. In the event of a random observation appearing in the output space, it incurred a penalty. At the same time, the study employed a set of five neighbors.

During data dimensionality reduction in this work, different data dimensionality reduction methods were used, and their hyperparameters were changed. For all data dimensionality reduction methods, the number of new dimensions varied from two to fifty. In the case of PCA, only the number of dimensions was varied. In the case of UMAP, the parameters and their values were changed: n neighbors hyperparameter set {5, 10, 20, 30, 50, 100}, minimum distance set {0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 0.75, 1}, and distance metrics {euclidean, cosine, and manhattan}. ISOMAP parameters and their values: n neighbors hyperparameter set {5, 10, 20, 30, 50, 100}, and the distance metric hyperparameter set {manhattan, cosine, manhattan, and minkowski}. T-SNE parameters and their values: learning rate {10, 50, 100, 500, and 1000}, and perplexi-ty {5, 10, 20, 30, 40, 50}. Trimap parameters and their values: number of inliers hyperparameter set {5, 10, 12, 20, 30, 50}, and the number of outliers hyperparameter set {2, 3, 4, 5, 10}. Below is a table of the best model results (see Table 1). More than 11,000 models were created to evaluate the best dimensionality reduction method for the current task.

Table 1. Trustworthiness metric values for the best dimensional reduction model of all the tested methods in specific dimensions.

Components	PCA	UMAP	Trimap	t-SNE	ISOMAP
2	0.756	0.933	0.831	0.871	0.748
3	0.805	0.950	0.882	0.881	0.821
4	0.844	0.955	0.908	0.883	0.859
5	0.876	0.961	0.931	0.886	0.896
6	0.898	0.964	0.942	0.902	0.908
7	0.915	0.966	0.950	0.912	0.920
8	0.928	0.969	0.955	0.928	0.938
9	0.940	0.971	0.961	0.937	0.949
10	0.948	0.973	0.964	0.954	0.965
15	0.968	0.974	0.971	0.966	0.977
20	0.980	0.975	0.974	0.975	0.989
25	0.986	0.978	0.976	0.978	0.989
30	0.990	0.980	0.977	0.979	0.990
35	0.992	0.983	0.978	0.981	0.991
40	0.993	0.984	0.978	0.982	0.991
50	0.996	**0.986**	0.978	0.984	0.992

Bold underlined value indicates the selected model used in the further research.

Based on the results of data reduction, it can be observed that all the methods performed quite similarly for a larger number of components. All studied methods became fairly stable from 15 dimensions. A higher number of dimensions was found to not improve the results of the models. Given that the results are important to consider the training time of the models, it was noticeable that the t-SNE models were trained longer compared to others as the Barnes-Hut algorithm was not used. The ISOMAP method also had quite a higher computational time compared with the other methods. It is important to note that the PCA method has a high trustworthiness value. However, after further analysis, it was noticed that the work profiles created using the PCA method were more complicated to interpret, and therefore this method was abandoned in further work. Based on these results, the UMAP dimensionality reduction method with fifty new output dimensions was further used in this work due to a faster computation time.

4.4. Clustering Results

This subsection provides information about the different clustering methods used during the study and the obtained results. A key point to evaluate clustering, in this case, is that data clustering was performed without prior knowledge of the actual data classes/clusters. It was impossible to use metrics such as accuracy, NMI, or other metrics requiring real clusters. Therefore, this paper used only the metrics that do not require real clusters and can be applied to solve real problems. Commonly used evaluation metrics for clustering without prior knowledge of labels include the silhouette coefficient, Davies–Bouldin index, and the Calinski–Harabasz index (CH Index). The silhouette coefficient measures the cohesion and separation of the clusters by comparing the average distance between data points within the same cluster to the average distance between data points in the nearest different cluster [74]. Higher silhouette coefficient values indicate better clustering quality, ranging from -1 to 1. The Davies–Bouldin index evaluates the clustering quality by combining intra-cluster similarity and inter-cluster dissimilarity [75]. Lower Davies–Bouldin index values signify better clustering with compact and well-separated clusters. The Calinski–Harabasz index (CH Index) assesses clustering quality by comparing the ratio of the between-cluster dispersion to the within-cluster dispersion [76]. A higher CH index value indicates better clustering, as it signifies a greater separation between clusters relative to the dispersion within clusters.

During data dimension reduction in this work, different data clustering methods were used, and their hyperparameters were changed, thus determining the most suitable data clustering method for the data. K-means variable parameter k is the number of clusters whose parameter set is {2, 3, 4, 5, 10, 20, 30}. DBSCAN changeable parameters: eps with set {0.1, 0.2, 0.3, 0.4, 0.5, 0.75}, and minimum samples {5, 10, 20, 30, 50, 100}. HDBSCAN changeable parameters: minimum cluster size set {10, 20, 30, 40, 50, 100}, minimum samples set {5, 10, 20, 50}, and epsilon set {0.1, 0.2, 0.3, 0.4, 0.5, 1}. BIRCH changeable parameters: number of clusters set {2, 3, 4, 5, 10, 20, 50}, and branching factor set {10, 20, 50, 100}. Affinity propagation damping factor set {0.5, 0.6, 0.7, 0.8, 0.9}. CBMIDE and RCBMIDE parameters: number of the cluster as earlier methods, number of projections directions T set {5, 10, 20, 50, 100}, smoothing parameter h {0.05, 0.1, 0.2, 0.3, 0.4, 0.5}, and probability of noise cluster set {0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5}. Below is a table of the best model results (see Table 2). More than 3000 models were created to evaluate the best available model for this task.

As mentioned earlier, more than 3000 data clustering models were created based on the reduced dataset. The reduced dataset made it possible to perform calculations faster. The results presented in the previous subsection prove that it is possible to do this with almost no loss of information. Data clustering assessment was performed based on Davies–Bouldin and Calinski–Harabasz metrics. In this case, the main metric was time Davies–Bouldin. In the case of extremely similar values, the Calinski–Harabasz metric was considered. Based on the obtained results, it was found that the HDBSCAN method displayed the best results, with a Davies–Bouldin value of 0.4475. HDBSCAN has emerged as the best-performing method in this scenario due to its unique ability to manage clusters of varying densities and

shapes, and its robustness in identifying noise points. Unlike other clustering algorithms that require a predefined number of clusters, HDBSCAN automatically detects the optimal cluster structure based on the data's underlying density distribution. This adaptability makes it particularly suitable for complex datasets with uneven densities and non-spherical clusters. Furthermore, it is important to note that good results were also obtained using the RCBMIDE clustering method. This robust method eliminates the outliers' impact. Based on these results, the HDBSCAN method was chosen to be used in further research.

Table 2. Davies–Boulding and Calinski–Harabasz metrics values for the top three models of each clustering model evaluated in the research.

Method	Parameters	Davies–Bouldin	Calinski–Harabasz
K-means	{'n_clusters': 5}	0.9143	**3386**
	{'n_clusters': 10}	1.0041	2995
	{'n_clusters': 20}	1.0487	2551
DBSCAN	{'eps': 0.2, 'min_samples': 30}	1.1245	1352
	{'eps': 0.3, 'min_samples': 20}	1.1568	1458
	{'eps': 0.3, 'min_samples': 50}	1.2658	1589
HDBSCAN	{'cluster_selection_epsilon': 0.3, 'min_cluster_size': 50, 'min_samples': 20}	**0.4475**	2698
	{'cluster_selection_epsilon': 0.2, 'min_cluster_size': 20, 'min_samples': 5}	0.7968	1398
	{'cluster_selection_epsilon': 0.3, 'min_cluster_size': 30, 'min_samples': 5}	0.9033	1548
BIRCH	{'branching_factor': 100, 'n_clusters': 5, 'threshold': 0.4}	1.1823	3216
	{'branching_factor': 10, 'n_clusters': 4, 'threshold': 0.3}	1.2641	2515
	{'branching_factor': 20, 'n_clusters': 30, 'threshold': 0.4}	1.2951	1927
Affinity propagation	{'damping': 0.5}	1.1374	1011
	{'damping': 0.8}	1.2493	1265
	{'damping': 0.7}	1.2623	1255
CBMIDE	{'n_components': 2}	1.1731	1689
	{'n_components': 30}	1.1875	1456
	{'n_components': 20}	1.2041	1265
RCBMIDE	{'n_components': 4}	1.0931	2035
	{'n_components': 20}	1.1175	1689
	{'n_components': 10}	1.1540	1356

Bold underlined: Best overall model metrics values.

4.5. Jobs Advertisement Requirements Cluster Analysis for Demand Understanding

In this section, we present the results of our comprehensive analysis of natural language processing (NLP) and clustering-based job profile extraction from a large dataset of over half a million job advertisements. The primary objective of this study was to synthesize meaningful and coherent job profile descriptions by employing advanced clustering techniques and NLP algorithms, thereby facilitating a deeper understanding of the labor market dynamics and the evolving nature of these job requirements. The results of our analysis not only provide valuable insights into the underlying structure and patterns of job profiles, but also highlight the frequency changes of extracted job profiles over time. In doing so, we aimed to contribute to this scientific field and human resource management by offering a data-driven, robust, and scalable approach to identifying the key trends and shifts in the job market. In the visualization presented below (see Figure 4), the automatically extracted job profiles have been depicted, which were generated utilizing clustering techniques and natural language processing methodologies. The results obtained through these methods demonstrated the successful creation of distinct job profiles. A closer examination of the skills identified in the first job profile reveals competencies such as "accounting", "finance",

"economics", and other related areas, which can be reasonably interpreted as describing a job profile within the finance/economics domain.

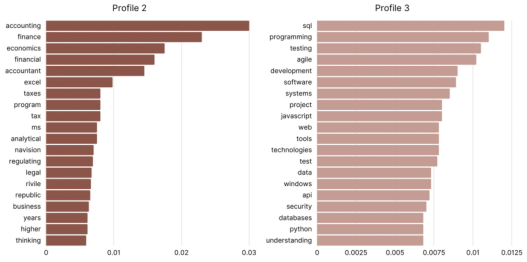


Figure 4. Requirements for extracted profile 2 (finance/accounting specialist) and profile 3 (programmer).

In contrast, the second job profile illustrated skills such as "SQL", "programming", "testing", and "agile", which are indicative of a programmer's profile. Further elucidation of these job profiles, including additional examples, can be found in Appendix A Figure A1. Based on the compiled profiles, it can be seen that several profiles included skills that were more difficult to understand, such as German, cat, and others (see Profile 8 in Appendix A). Such results show that this method does have limitations when the obtained skills are more difficult to interpret. In this case, it was noticeable that we are talking about knowing the German language in the case of the "German" skill. At the same time, "Cat" describes job advertisements related to animal care.

Applying these advanced techniques in clustering and natural language processing has allowed the efficient and accurate extraction of salient job-related skill sets, which can be used to better understand and categorize various roles within different industries, and which could thereby facilitate improved job matching and skill development opportunities for both employers and job seekers.

Two distinct approaches can be employed to automatically generate job profiles using the data acquired through clustering and natural language processing techniques: expert evaluation and profiling, or automated profiling with the assistance of generative artificial intelligence (AI). The following table (see Table 3) offers detailed information regarding the synthesized job profile descriptions based on the previously extracted keywords. These synthesized job descriptions were created using the GPT-4 API version that used the prompt "Generate job profile based on the keywords listed: {profile keywords}". Upon examination, it was apparent that the job profiles were accurately and comprehensively portrayed, capturing not only the primary attributes of the job profiles but also elaborating on the associated information. Utilizing generative AI, these job descriptions can be further expanded and refined, providing a more in-depth understanding of the roles and responsibilities involved in each profile. This approach of combining expert evaluation and profiling, along with generative AI, has the potential to yield highly accurate and detailed job profiles that can better facilitate matching candidates to relevant job opportunities and inform career development and training programs. Additionally, these synthesized job profile descriptions can serve as a valuable resource for human resources professionals, recruiters, and job seekers, enabling them to better understand the various positions available across different industries and the corresponding skill sets required for success in those roles. For full results of the generated profiles see Appendix B Table A1.

The analysis of job profiles and the identification of relevant skills not only involves the instantaneous recognition of specific profiles, but also entails examining the temporal dynamics of these profiles. The graph below illustrates the fluctuations in demand for profile 3 over time (see Figure 5). A significant surge in demand for workers possessing

these particular skills was observed at the beginning of 2022. However, following the outbreak of the war in Ukraine in March 2022, the demand for these specialists sharply declined. By November 2022, there was no noticeable increase in the worker requirement within this profile. A modest uptick in demand for these professionals has been observed since November 2022, and the need for workers with these skills has been anticipated to grow, with the recent upward trend in the latest available data supporting this projection. The utility of this data lies in its capacity to inform various stakeholders, such as employers, job seekers, and policymakers, about the shifting demand for particular skill sets in the job market. By understanding these fluctuations, stakeholders can make informed decisions about workforce development, investment in training programs, and recruitment strategies that align with the evolving needs of the industry. Furthermore, this temporal analysis provides valuable insights into the potential impact of external factors, such as geopolitical events and on-demand for specific job profiles, enabling more adaptive and resilient planning for the future.

Table 3. Synthesized keyword-based job profiles for two job profiles requirements.

Profile	Synthesized Keyword-Based Job Profile
2	This profile is about a finance and accounting professional with a strong background in economics and financial management. They have experience working as an accountant and are skilled in using Excel and other MS Office programs, as well as accounting software such as Navision and Rivilė. Their expertise includes tax preparation, regulation compliance, and legal aspects related to financial operations. They possess analytical and critical thinking abilities, which contribute to their effectiveness in the financial field. The individual has a higher level of education, potentially a degree in business or economics and has several years of experience in industry. They may also have knowledge of the financial regulations specific to a certain republic or region.
3	This profile is about a software development and testing professional with expertise in various programming languages, tools, and technologies. They have experience working with databases, APIs, security, and data management in both web and Windows environments. They are also knowledgeable in Agile project management methodologies and have a strong understanding of software systems and development processes. Their skills include SQL, Python, and JavaScript programming, as well as using various testing and development tools.

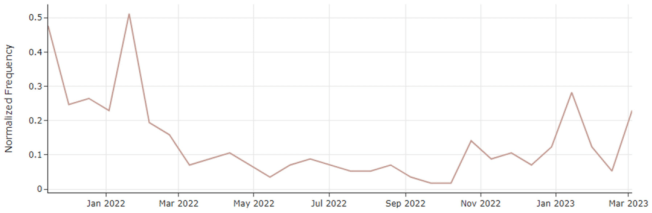


Figure 5. Job requirements profile 3 (programmer) over time.

5. Discussion

The primary objective of this article was to explore the application of natural language processing, data clustering, and other machine learning techniques in determining job profile requirements within the Lithuanian context. The discussion focuses on the key methods to extract data from publicly available unstructured sources. Moreover, the paper delves into the subsequent processing of these data using various vectorization techniques, examining the advantages and disadvantages of each method.

During the research, it was observed that the regex method yielded the most favorable results for data extraction in the case of the Lithuanian job advertisements. Regarding data vectorization and feature extraction, the study utilized two primary techniques—TF-IDF

and sentence transformers (BERT model). Due to the high dimensionality of the feature extracted data (384 dimensions), dimensionality reduction methods were employed, with the ISOMAP approach achieving the best results. A total of 11,000 models for dimensionality reduction were created to assess the different dimensionality reduction methods and their parameters, as well as the optimal parameters set for each method. Nonetheless, other methods, such as UMAP, also demonstrated a satisfactory performance. Speed was the main parameter in the automated job advertisement analyzes, as the UMAP method was selected as the research method. The UMAP method, compared with the other methods such as t-SNE or ISOMAP, demonstrated a better speed, and as mentioned earlier, is important in the daily job advertisement analysis. To have the best speed, the PCA method could have been used, but its results compared with the UMAP would have been of lower quality. Notably, increasing the latent dimensions did not yield improved outcomes in the context of data dimensionality reduction. From 20 latent dimensions, the applied data dimensionality reduction methods did not show better results.

For this reason, 20 latent dimensions could also be used in case of larger system limitations. This study used the 50 latent dimensions obtained with the previously mentioned UMAP method for dimensionality reduction. It was also observed that dimensionality reduction for this specific data, even with an extremely small number of latent dimensions, e.g., 2, 5, that the value of the trustworthiness metric was more than 0.9.

The paper reviewed many different data clustering methods. More than 3000 different models were created for data clustering to optimize the parameters of different methods. The HDBSCAN method proved to be the most effective in data clustering, largely attributable to its hierarchical structure. Additionally, the RCBMIDE method also exhibited relatively strong results, as per the metrics employed in the study, by enabling the elimination of the outliers and the inclusion of data that best represented the information. The obtained job profiles based on both methods were quite similar. However, judging by the metrics, HDBSCAN was used as the main data clustering method. It was also noticeable that when using the most used k-means method, the best clustering result with this method was obtained when the number of clusters was only 5, which was deemed to be extremely small when compiling the number of job profiles of the country. The number of formed profiles must therefore be interpreted.

Upon evaluating the job profiles, it was evident that the extraction quality was high, thereby supporting the utilization of these methods for automated profile research. Based on the compiled profiles, it was observed that some profiles included skills that were more difficult to understand, such as German, cat, and others. Such results demonstrate that this method has limitations when the obtained skills are more difficult to interpret. In this case, it was noticeable that they are talking about knowing the German language in the case of the "German" skill. At the same time, "Cat" describes job advertisements related to animal care. Based on the received job profiles, these job profiles were described using an automated generative artificial intelligence algorithm. Different models of generative artificial intelligence were used to describe these job profiles. The best results were seen with the current latest GPT-4 language model, although the GPT-3.5 model also performed well. Meanwhile, older models of generative artificial intelligence did not have particularly good results, and the created job profile descriptions could be only applied in practice through their additional analysis and improvement.

5.1. Extrapolating this Study to Other Countries

Several considerations should be considered regarding the extrapolation of this study to other countries. Firstly, differences in the language and terminologies used in job advertisements may require adaptations to natural language processing techniques and data extraction methods. For instance, cultural nuances and terminologies specific to a particular country may necessitate adjustments to the regex method to maintain its effectiveness. Secondly, variations in labor market structure and industry composition across countries could affect the applicability of the chosen clustering and dimensionality reduction methods. It would therefore be essential to assess the effectiveness of these

methods in the context of the target country's labor market characteristics to ensure that the extracted job profiles are both accurate and reliable. Lastly, legal, and ethical considerations, such as data privacy regulations and consent requirements for using publicly available data, may differ across countries. Researchers must consider these factors when applying the methodology used in this study in other contexts. In conclusion, although this study successfully extracted job profiles within the Lithuanian context, researchers should be aware of the potential challenges and limitations when attempting to extrapolate the findings to other countries. This methodology may provide valuable insights into job profile requirements across various international contexts by addressing the language, labor market, and legal considerations.

5.2. Future Directions

As we look towards future research, several promising avenues exist for extending the current study on job advertisements and job profiles within a multi-country European context. The following areas should be investigated in future work, allowing for a more comprehensive understanding of the European labor market dynamics. Standardization and harmonization: Developing a standardized framework for job titles and classifications across European countries will facilitate comparative analyzes and enhance our results' generalizability. Adopting an existing system, such as the International Standard Classification of Occupations (ISCO), can provide a foundation for harmonizing job roles and enable more efficient cross-country comparisons. Adaptation to diverse contexts: With the diversity in economic conditions, labor regulations, and employment practices across the European countries, future research should aim to adapt data extraction and processing techniques accordingly. It may involve employing supervised and unsupervised machine learning algorithms to capture the nuances specific to each country and ensure the accuracy of the extracted job profiles. Cross-cultural analysis: Investigating cultural differences in job advertisement language and presentation can provide valuable insights into the variation in job requirements and expectations across the European countries. By examining these cultural aspects, we can better understand the dynamics of the European labor market and develop more tailored strategies for workforce development and talent acquisition. Longitudinal analysis: Conducting a longitudinal analysis of the job advertisements and job profiles can help to identify the trends and shifts in labor market demands over time. This temporal perspective would enable policymakers, employers, and job seekers to anticipate and respond to emerging needs and opportunities more effectively. Integration with labor market indicators: Combining the analysis of job advertisements and job profiles with other labor market indicators, such as unemployment rates, wage levels, and skill shortages, can provide a more holistic view of the European labor market landscape. This integrated approach can support evidence-based decision-making for education, training, and employment policies at both national and regional levels. By pursuing these future research directions, we can further advance the understanding of job profile requirements across various European contexts and contribute to more informed policymaking and workforce development strategies.

Author Contributions: Conceptualization: M.L., V.Š., V.P., J.B., A.S. and A.G; methodology: M.L. and V.Š.; software: M.L.; validation: V.P., J.B., A.S. and A.G.; formal analysis: M.L. and V.Š.; investigation: M.L., V.Š., V.P. and A.G.; resources: M.L., V.Š., V.P., J.B., A.S. and A.G.; data curation: M.L.; writing—original draft preparation: M.L. and V.Š.; writing—review and editing: M.L., V.Š., V.P., J.B., A.S. and A.G.; visualization: M.L. and V.Š.; supervision: V.P., J.B. and A.S.; project administration: V.P., J.B. and A.S.; funding acquisition: V.P., J.B. and A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This project has received funding from the European Regional Development Fund (project No 13.1.1-LMT-K-718-05-0012) under a grant agreement with the Research Council of Lithuania (LMTLT). Funded as the European Union's measure in response to the Cov-19 pandemic.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank the area editor and the reviewers for their valuable comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

This appendix contains information about the skills prevalent in different profiles, i.e., what skills are specific to a certain profile. It is important to note that the top 20 skills and their repetition are presented in the job postings.

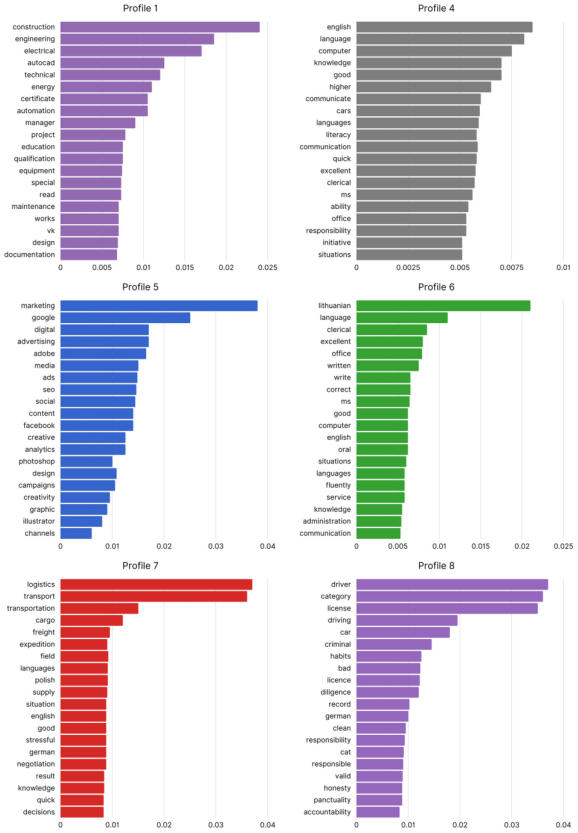


Figure A1. *Cont.*

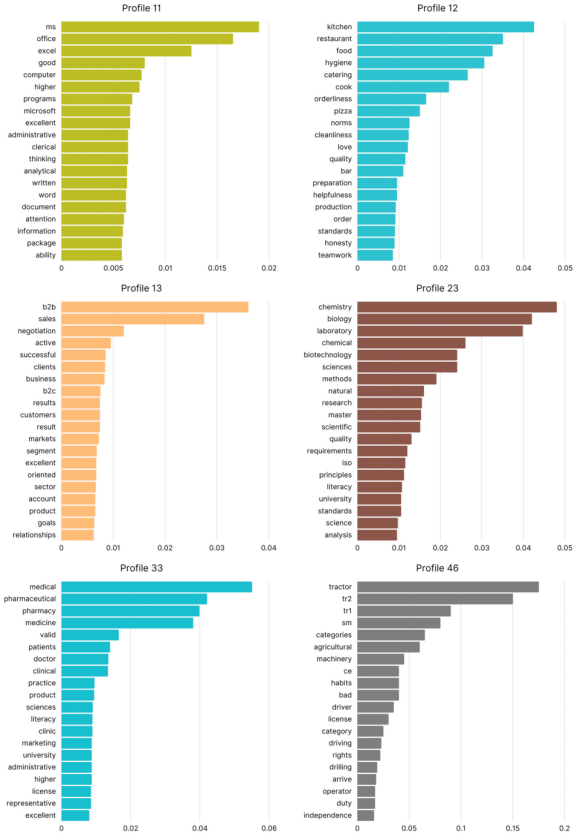


Figure A1. Requirements for extracted profiles.

Appendix B

This appendix describes the job profiles that were created using generative artificial intelligence based on the individual profile skills discussed earlier.

Table A1. Syntesized job profiles descriptions using generative artificial intelligence.

Profile	Synthesized Keyword-Based Job Profile
1	This profile is about a construction and engineering professional with expertise in electrical systems, automation, and energy management. They have experience in project management and are skilled in using AutoCAD for technical design and documentation. This individual holds the relevant education and qualifications, including a certificate in a specialized area. They are knowledgeable in equipment maintenance, handling works related to electrical engineering, and can read and understand technical documentation. Their background also includes managing and designing various construction and engineering projects.

Table A1. *Cont.*

Profile	Synthesized Keyword-Based Job Profile
4	This profile is about a professional with strong English language and communication skills, proficiency in computer literacy, and experienced in clerical and office-related tasks. They have a good knowledge of MS Office tools and can effectively communicate in various situations. The individual is also responsible, takes initiative, and can quickly adapt to new environments. They have a higher level of education and might have an interest in or experience with cars. Their language skills may extend to other languages as well, highlighting their overall linguistic abilities.
5	This profile is about a digital marketing and advertising professional with expertise in various aspects of online marketing, such as Google Ads, SEO, and social media management. They have experience in content creation, campaign management, and analytics, utilizing tools such as Adobe Photoshop and Illustrator for graphic design and creative purposes. Their skills include managing and optimizing advertising campaigns on platforms such as Facebook and other digital channels. They possess a strong creativity and are proficient in using marketing analytics tools to measure the success of their campaigns. Overall, this individual is well-versed in the digital media landscape, and has a deep understanding of how to leverage various platforms and tools to achieve marketing objectives.
6	This profile is about a bilingual professional with fluency in both the Lithuanian and English languages, possessing excellent clerical and administrative skills. They have experience in office administration and are proficient in using MS Office tools and other computer applications. Their strong written and oral communication abilities allow them to excel in various situations, providing excellent service in both languages. They can write and speak fluently and correctly in Lithuanian and English, demonstrating their adaptability in diverse environments. This individual's background includes a good knowledge of administrative tasks and effective communication in multiple languages, making them a valuable asset in any organization requiring multilingual support.
7	This profile is about a logistics and transportation professional with experience in cargo and freight expeditions. They have a strong background in the field of transport and supply chain management, with the ability to handle various situations, including stressful ones. They possess excellent negotiation skills and can make quick, result-oriented decisions. This individual is also proficient in multiple languages, including English, Polish, and German, which allows them to effectively communicate and coordinate in diverse environments. Their knowledge of the logistics sector and expertise in transport make them a valuable asset to any organization involved in the movement of goods and freight.
8	This profile is about a responsible and diligent driver with a valid license for a specific category of vehicles. They have experience driving cars and maintaining a clean criminal record, as well as a good driving record without any bad habits. This individual demonstrates responsibility, honesty, punctuality, and accountability in their work. They may also have knowledge of the German language, which could be beneficial in certain driving situations or locations. Their strong sense of diligence and commitment to safe driving practices make them a reliable and trustworthy candidate for any driving-related job.
12	This profile is about a professional in the kitchen, restaurant, and catering industry who is enthusiastic about food and its preparation. They have experience as a cook, possibly specializing in pizza and other culinary delights. They are committed to maintaining high standards of hygiene, cleanliness, and orderliness in their work environment, adhering to established norms and regulations. This individual values quality in food production and preparation while also demonstrating helpfulness, honesty, and teamwork. Their love for the culinary arts, combined with their dedication to maintaining high standards in the kitchen, makes them an excellent candidate for roles in the food and restaurant industry.
23	This profile is about a professional in the fields of chemistry and biology, with a strong background in laboratory work, biotechnology, and natural sciences research. They have a master's degree from a university, showcasing their expertise in scientific principles and methods. Their experience includes working with chemical analysis, quality control, and adhering to ISO standards and other requirements. This individual is knowledgeable in various scientific techniques and possesses strong literacy in the sciences. Their dedication to maintaining high-quality research and understanding of both chemistry and biology make them an excellent candidate for roles in the scientific and biotechnology industries.
32	This profile is about a professional in the medical and pharmaceutical fields, with experience in pharmacy, clinical practice, and medicine. They have a valid license and a higher degree in the sciences from a university, demonstrating their expertise in the field. This individual possesses excellent literacy in medical and pharmaceutical topics, and has experience working with patients, doctors, and other healthcare professionals. They may also have experience in product marketing and administrative tasks within a clinic or healthcare setting. Their strong background in medicine and pharmaceuticals, along with their dedication to patient care and professionalism, make them an ideal candidate for roles in the healthcare and pharmaceutical industries.

References

1. Nielsen, P.; Holm, J.R.; Lorenz, E. Work policy and automation in the fourth industrial revolution. In *Globalisation, New and Emerging Technologies, and Sustainable Development*; Routledge: Abingdon, UK, 2021; pp. 189–207.
2. Lloyd, C.; Payne, J. Rethinking country effects: Robotics, AI and work futures in Norway and the UK. *New Technol. Work. Employ.* **2019**, *34*, 208–225. [CrossRef]
3. Frey, C.B.; Osborne, M.A. The future of employment: How susceptible are jobs to computerisation? *Technol. Forecast. Soc. Chang.* **2017**, *114*, 254–280. [CrossRef]
4. Quintini, G. *Automation, Skills Use and Training*; Technical Report; OECD Publishing: Paris, France, 2018.
5. Bacher, J.; Tamesberger, D. The Corona Generation: (Not) Finding Employment during the Pandemic. *CESifo Forum* **2021**, *22*, 3–7.
6. Arntz, M.; Gregory, T.; Zierahn, U. *The Risk of Automation for Jobs in OECD Countries: A Comparative Analysis*; OECD Publishing: Paris, France, 2016.
7. OECD. *OECD Skills Studies OECD Skills Strategy Lithuania Assessment and Recommendations*; OECD Publishing: Paris, France, 2021.
8. Hershbein, B.; Kahn, L.B. Do recessions accelerate routine-biased technological change? Evidence from vacancy postings. *Am. Econ. Rev.* **2018**, *108*, 1737–1772. [CrossRef]
9. Verma, A.; Lamsal, K.; Verma, P. An investigation of skill requirements in artificial intelligence and machine learning job advertisements. *Ind. High. Educ.* **2022**, *36*, 63–73. [CrossRef]
10. Deming, D.; Kahn, L.B. Skill requirements across firms and labor markets: Evidence from job postings for professionals. *J. Labor Econ.* **2018**, *36*, S337–S369. [CrossRef]
11. Boselli, R.; Cesarini, M.; Mercorio, F.; Mezzanzanica, M. Using machine learning for labour market intelligence. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, 18–22 September 2017; pp. 330–342.
12. Brynjolfsson, E.; Horton, J.J.; Ozimek, A.; Rock, D.; Sharma, G.; Tuye, H.-Y. *COVID-19 and Remote Work: An Early Look at US Data*; National Bureau of Economic Research: Cambridge, MA, USA, 2020.
13. Autor, D.; Reynolds, E. *The Nature of Work after the COVID Crisis: Too Few Low-Wage Jobs*; Brookings Institution: Washington, DC, USA, 2020.
14. Kramer, A.; Kramer, K.Z. The potential impact of the COVID-19 pandemic on occupational status, work from home, and occupational mobility. *J. Vocat. Behav.* **2020**, *119*, 103442. [CrossRef]
15. Fabo, B. The Corona-Inducted Shift Towards Intermediate Digital Skills Across Occupations in Slovakia. In *Digital Labour Markets in Central and Eastern European Countries*; Routledge: Abingdon, UK, 2023; pp. 37–48.
16. Rebele, J.E.; Pierre, E.K.S. A commentary on learning objectives for accounting education programs: The importance of soft skills and technical knowledge. *J. Account. Educ.* **2019**, *48*, 71–79. [CrossRef]
17. Brunello, G.; Wruuck, P. Skill shortages and skill mismatch: A review of the literature. *J. Econ. Surv.* **2021**, *35*, 1145–1167. [CrossRef]
18. Wagner, J.A.; Hollenbeck, J.R. *Organizational Behavior: Securing Competitive Advantage*; Routledge: Abingdon, UK, 2020.
19. Ibrahim, R.; Boerhannoeddin, A.; Bakare, K.K. The effect of soft skills and training methodology on employee performance. *Eur. J. Train. Dev.* **2017**, *41*, 388–406. [CrossRef]
20. Heckman, J.J.; Kautz, T. Hard evidence on soft skills. *Labour Econ.* **2012**, *19*, 451–464. [CrossRef] [PubMed]
21. Asbari, M.; Purwanto, A.; Ong, F.; Mustikasiwi, A.; Maesaroh, S.; Mustofa, M.; Hutagalung, D.; Andriyani, Y. Impact of hard skills, soft skills and organizational culture: Lecturer innovation competencies as mediating. *EduPsyCouns J. Educ. Psychol. Couns.* **2020**, *2*, 101–121.
22. De Mauro, A.; Greco, M.; Grimaldi, M.; Ritala, P. Human resources for Big Data professions: A systematic classification of job roles and required skill sets. *Inf. Process. Manag.* **2018**, *54*, 807–817. [CrossRef]
23. Autor, D.H. Work of the Past, Work of the Future. *AEA Pap. Proc.* **2019**, *109*, 1–32. [CrossRef]
24. Groysberg, B.; Lee, J.; Price, J.; Cheng, J. The leader's guide to corporate culture. *Harv. Bus. Rev.* **2018**, *96*, 44–52.
25. Isphording, I.E. Language and labor market success. In *International Encyclopedia of the Social & Behavioral Sciences*; Institute of Labor Economics: Bonn, Germany, 2014.
26. Berg, P.; Kossek, E.E.; Misra, K.; Belman, D. Work-life flexibility policies: Do unions affect employee access and use? *ILR Rev.* **2014**, *67*, 111–137. [CrossRef]
27. Bilal, M.; Malik, N.; Khalid, M.; Lali, M.I.U. Exploring industrial demand trend's in Pakistan software industry using online job portal data. *Univ. Sindh J. Inf. Commun. Technol.* **2017**, *1*, 17–24.
28. Clarke, M. Rethinking graduate employability: The role of capital, individual attributes and context. *Stud. High. Educ.* **2018**, *43*, 1923–1937. [CrossRef]
29. Mahany, A.; Khaled, H.; Elmitwally, N.S.; Aljohani, N.; Ghoniemy, S. Negation and Speculation in NLP: A Survey, Corpora, Methods, and Applications. *Appl. Sci.* **2022**, *12*, 5209. [CrossRef]
30. Kalyan, K.S.; Rajasekharan, A.; Sangeetha, S. Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv* **2021**, arXiv:2108.05542.
31. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.

32. Fellbaum, C. WordNet. In *Theory and Applications of Ontology: Computer Applications*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 231–243.
33. Black, S.; Biderman, S.; Hallahan, E.; Anthony, Q.; Gao, L.; Golding, L.; He, H.; Leahy, C.; McDonell, K.; Phang, J. Gpt-neox-20b: An open-source autoregressive language model. *arXiv* **2022**, arXiv:2204.06745.
34. Li, J.; Sun, A.; Han, J.; Li, C. A survey on deep learning for named entity recognition. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 50–70. [CrossRef]
35. Joulin, A.; Grave, E.; Bojanowski, P.; Douze, M.; Jégou, H.; Mikolov, T. Fasttext. zip: Compressing text classification models. *arXiv* **2016**, arXiv:1612.03651.
36. Salton, G. *Introduction to Modern Information Retrieval*; McGraw-Hill: New York, NY, USA, 1983.
37. Reimers, N.; Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv* **2019**, arXiv:1908.10084.
38. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013.
39. Arora, S.; Liang, Y.; Ma, T. A simple but tough-to-beat baseline for sentence embeddings. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
40. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
41. Harris, Z.S. Distributional structure. *Word* **1954**, *10*, 146–162. [CrossRef]
42. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
43. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 1188–1196.
44. Bellman, R.; Kalaba, R. On adaptive control processes. *IRE Trans. Autom. Control* **1959**, *4*, 1–9. [CrossRef]
45. Wang, Y.; Yao, H.; Zhao, S. Auto-encoder based dimensionality reduction. *Neurocomputing* **2016**, *184*, 232–242. [CrossRef]
46. Dong, Y.; Du, B.; Zhang, L.; Zhang, L. Dimensionality reduction and classification of hyperspectral images using ensemble discriminative local metric learning. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2509–2524. [CrossRef]
47. Thomas, R.; Judith, J. Hybrid dimensionality reduction for outlier detection in high dimensional data. *Int. J.* **2020**, *8*, 5883–5888.
48. Li, M.; Wang, H.; Yang, L.; Liang, Y.; Shang, Z.; Wan, H. Fast hybrid dimensionality reduction method for classification based on feature selection and grouped feature extraction. *Expert Syst. Appl.* **2020**, *150*, 113277. [CrossRef]
49. McInnes, L.; Healy, J.; Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv* **2018**, arXiv:1802.03426.
50. Sumithra, V.; Surendran, S. A review of various linear and non linear dimensionality reduction techniques. *Int. J. Comput. Sci. Inf. Technol.* **2015**, *6*, 2354–2360.
51. Abdi, H.; Williams, L.J. Principal component analysis. *Wiley Interdiscip. Rev. Comput. Stat.* **2010**, *2*, 433–459. [CrossRef]
52. Du, X.; Zhu, F. A novel principal components analysis (PCA) method for energy absorbing structural design enhanced by data mining. *Adv. Eng. Softw.* **2019**, *127*, 17–27. [CrossRef]
53. Iannucci, L. Chemometrics for data interpretation: Application of principal components analysis (PCA) to multivariate spectroscopic measurements. *IEEE Instrum. Meas. Mag.* **2021**, *24*, 42–48. [CrossRef]
54. Fan, C.; Sun, Y.; Zhao, Y.; Song, M.; Wang, J. Deep learning-based feature engineering methods for improved building energy prediction. *Appl. Energy* **2019**, *240*, 35–45. [CrossRef]
55. Van Der Maaten, L. t-SNE. 2019. Available online: https://lvdmaaten.github.io/tsne (accessed on 25 March 2023).
56. Linderman, G.C.; Steinerberger, S. Clustering with t-SNE, provably. *SIAM J. Math. Data Sci.* **2019**, *1*, 313–332. [CrossRef]
57. Kobak, D.; Linderman, G.C. Initialization is critical for preserving global data structure in both t-SNE and UMAP. *Nat. Biotechnol.* **2021**, *39*, 156–157. [CrossRef]
58. Becht, E.; McInnes, L.; Healy, J.; Dutertre, C.-A.; Kwok, I.W.; Ng, L.G.; Ginhoux, F.; Newell, E.W. Dimensionality reduction for visualizing single-cell data using UMAP. *Nat. Biotechnol.* **2019**, *37*, 38–44. [CrossRef] [PubMed]
59. Böhm, J.N.; Berens, P.; Kobak, D. A unifying perspective on neighbor embeddings along the attraction-repulsion spectrum. *arXiv* **2020**, arXiv:2007.08902.
60. Arunkumar, N.; Mohammed, M.A.; Abd Ghani, M.K.; Ibrahim, D.A.; Abdulhay, E.; Ramirez-Gonzalez, G.; de Albuquerque, V.H.C. K-means clustering and neural network for object detecting and identifying abnormality of brain tumor. *Soft Comput.* **2019**, *23*, 9083–9096. [CrossRef]
61. Jain, A.K. Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666. [CrossRef]
62. Singh, A.; Yadav, A.; Rana, A. K-means with Three different Distance Metrics. *Int. J. Comput. Appl.* **2013**, *67*, 13–17. [CrossRef]
63. Schubert, E.; Sander, J.; Ester, M.; Kriegel, H.P.; Xu, X. DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Trans. Database Syst.* **2017**, *42*, 1–21. [CrossRef]
64. Campello, R.J.; Moulavi, D.; Zimek, A.; Sander, J. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Trans. Knowl. Discov. Data* **2015**, *10*, 1–51. [CrossRef]
65. Zhang, T.; Ramakrishnan, R.; Livny, M. BIRCH: An efficient data clustering method for very large databases. *ACM SIGMOD Rec.* **1996**, *25*, 103–114. [CrossRef]

66. Dueck, D.; Frey, B.J. Non-metric affinity propagation for unsupervised image categorization. In Proceedings of the 2007 IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 14–21 October 2007; pp. 1–8.
67. Guan, R.; Shi, X.; Marchese, M.; Yang, C.; Liang, Y. Text clustering with seeds affinity propagation. *IEEE Trans. Knowl. Data Eng.* **2010**, *23*, 627–637. [CrossRef]
68. Fang, Q.; Sang, J.; Xu, C.; Rui, Y. Topic-sensitive influencer mining in interest-based social media networks via hypergraph learning. *IEEE Trans. Multimed.* **2014**, *16*, 796–812. [CrossRef]
69. Ng, A.; Jordan, M.; Weiss, Y. On spectral clustering: Analysis and an algorithm. *Adv. Neural Inf. Process. Syst.* **2001**, *14*, 849–856.
70. Janani, R.; Vijayarani, S. Text document clustering using spectral clustering algorithm with particle swarm optimization. *Expert Syst. Appl.* **2019**, *134*, 192–200. [CrossRef]
71. Lukauskas, M.; Ruzgas, T. A New Clustering Method Based on the Inversion Formula. *Mathematics* **2022**, *10*, 2559. [CrossRef]
72. Lukauskas, M.; Ruzgas, T. Reduced Clustering Method Based on the Inversion Formula Density Estimation. *Mathematics* **2023**, *11*, 661. [CrossRef]
73. Venna, J.; Kaski, S. Neighborhood preservation in nonlinear projection methods: An experimental study. In Proceedings of the Artificial Neural Networks—ICANN 2001: International Conference, Vienna, Austria, 21–25 August 2001; pp. 485–491.
74. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [CrossRef]
75. Davies, D.L.; Bouldin, D.W. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *PAMI-1*, 224–227. [CrossRef]
76. Caliński, T.; Harabasz, J. A dendrite method for cluster analysis. *Commun. Stat.-Theory Methods* **1974**, *3*, 1–27. [CrossRef]

PADĖKA

Dėkoju disertacijos moksliniam vadovui doc. dr. Tomui Ruzgui, kuris dalinosi sukauptomis žiniomis, siūlė naujas idėjas ir visada rado papildomo darbo. Dėkoju ir Matematikos ir gamtos mokslų fakultetui, kuriame praleidau 10 metų. Ateidamas į jį nežinojau, ko tikėtis, ir likau jame tokiam ilgam laikui.

Dėkoju projekto vykdymo komandai prof. dr. Vaidai Pilinkienei, prof. dr. Alinai Stundžienei, prof. dr. Jurgitai Bruneckienei, dr. Andriui Grybauskui, kurie suteikė galimybes įgyvendinti idėjas, rašyti ir finansuoti straipsnius, prisidėti prie vykdomų projektų, planuoti ir rengti naujus projektus.

Be abejo, didelę padėką reiškiu įmonei UAB „Hostinger“, kuri prisidėjo prie mano tobulėjimo, leido duomenų mokslo srityje atrasti daug įdomių dalykų, vienu metu suteikė galimybes ir dirbti, ir studijuoti.

Labiausiai dėkoju Viktorijai Šarkauskaitei, kuri tiek daug metų pakentė darbą po 16 valandų per parą, savaitgaliais, be atostogų. Dėkoju už visą palaikymą, išklausymą, pagalbą ir darbus kartu.

Nuoširdžiai dėkoju ir gerbiamiems recenzentams.

Pagarbiai
Mantas Lukauskas

PRIEDAI

1 priedas. Duomenų rinkinių informacija

ID	Duomenų rinkinys	Imties dydis (N)	Dimensijos (D)	Klasteriai
1	1balance-scale	625	4	3
2	3-spiral	312	2	3
3	CinC_1000_10_15_20	1750	2	2
4	CinC_1000_15_15_15	1750	2	2
5	CinC_1000_15_15_20	1750	2	2
6	CinC_1000_15_15_30	1750	2	2
7	CinC_100_10_15_20	175	2	2
8	CinC_100_15_15_20	175	2	2
9	CirclesWithOutliers_1000_10_10_005	2050	2	3
10	CirclesWithOutliers_1000_10_10_01	2100	2	3
11	CirclesWithOutliers_1000_10_10_02	2200	2	3
12	CirclesWithOutliers_100_10_10_01	210	2	3
13	CirclesWithOutliers_100_10_10_02	220	2	3
14	Circles_1000_100_15_20	750	2	2
15	Circles_1000_100_50_20	750	2	2
16	Circles_1000_100_5_20	750	2	2
17	Circles_1000_10_15_20	750	2	2
18	Circles_1000_15_15_20	750	2	2
19	Circles_100_15_15_20	75	2	2
20	Coil100	7200	1024	100
21	Coil20	1440	1024	20
22	Corners_1000_05_10_5	2000	2	4
23	Corners_1000_0_10_5	2000	2	4
24	Corners_1000_1_10_5	2000	2	4
25	Corners_1000_1_5_5	2000	2	4
26	Corners_100_1_5_5	200	2	4
27	CrescentFullMoon_1000_10_15_20	1000	2	2
28	CrescentFullMoon_1000_15_15_15	1000	2	2
29	CrescentFullMoon_1000_15_15_20	1000	2	2
30	CrescentFullMoon_1000_25_15_15	1000	2	2
31	CrescentFullMoon_100_10_15_20	100	2	2
32	D31	3100	2	31
33	MNIST	60000	784	10
34	Outliers_1000_15_10_005_5	1000	2	3
35	Outliers_1000_15_15_005_2	1000	2	3
36	Outliers_1000_15_15_005_5	1000	2	3
37	Pendigits	10992	16	10
38	R15	600	2	15
39	S1	5000	2	15
40	Shuttle	58000	9	7
41	Spirals_1000_1440_2	1000	2	2
42	Spirals_1000_720_0	1000	2	2
43	Spirals_1000_720_1	1000	2	2
44	Spirals_1000_720_2	1000	2	2
45	Spirals_1000_720_3	1000	2	2
46	Spirals_1000_720_4	1000	2	2
47	Spirals_100_720_0	100	2	2
48	TwoHalfMoon_1000_10_15_1	1000	2	2
49	TwoHalfMoon_1000_10_15_2	1000	2	2

50	TwoHalfMoon_1000_15_15_1	1000	2	2
51	TwoHalfMoon_1000_15_15_2	1000	2	2
52	TwoHalfMoon_100_10_15_2	100	2	2
53	TwoHalfMoon_100_15_15_2	100	2	2
54	aggregation	788	2	7
55	aml28	804	2	5
56	arrhythmia	452	262	13
57	atom	800	3	2
58	banana	4811	2	2
59	breast	569	30	2
60	chainlink	1000	3	2
61	compound	399	2	6
62	cpu	209	6	4
63	cure-t0-2000n-2D	2000	2	3
64	cure-t2-4k	4200	2	7
65	curves1	1000	2	2
66	curves2	1000	2	2
67	dermatology	366	34	6
68	diabetes	442	10	4
69	ecoli	336	7	8
70	flame	240	2	2
71	gaussians1	100	2	2
72	german	1000	60	2
73	glass	214	9	6
74	haberman	306	3	2
75	heart-statlog	270	13	2
76	iono	351	34	2
77	iris	150	4	3
78	jain	373	2	2
79	letter_new	20000	16	26
80	lsun	400	2	3
81	pathbased	300	2	3
82	pmf	649	3	5
83	rings	1000	2	3
84	s-set1	5000	2	15
85	segment	2310	19	7
86	shapes	1000	2	4
87	simplex	500	3	4
88	smile1	1000	2	4
89	sonar	208	60	2
90	spambase	4601	57	2
91	spherical_4_3	400	3	4
92	spiral	1000	2	2
93	spiralsquare	1500	2	6
94	square1	1000	2	4
95	st900	900	2	9
96	tae	151	5	3
97	target	770	2	6
98	tetra	400	3	4
99	threenorm	1000	2	2
100	thy	215	5	3
101	triangle1	1000	2	4
102	twenty	1000	2	20
103	twodiamonds	800	2	2
104	usps	9298	256	10

105	vehicle	846	18	4
106	vowel	990	10	6
107	wdbc	569	30	2
108	wine	178	13	3
109	wingnut	1016	2	2
110	wisc	699	9	2
111	xclara	3000	2	3
112	xor	1000	3	4
113	zoo	101	16	7

2 priedas. Duomenų klasterizavimo rezultatai (klasterizavimo tikslumas)

Duomenų rinkinys	Agg	BGM	BIRCH	CBMIDE	DBSCAN	FCM	GMM	HDBSCAN	K-means	MIDE	OPTICS	ST-DBSCAN
1balance-scale	0,742	0,670	0,790	0,834	0,746	0,906	0,682	0,770	0,693	0,600	0,714	0,758
3-spiral	1,000	0,394	0,478	0,590	1,000	0,359	0,429	1,000	0,990	0,340	1,000	0,474
aggregation	1,000	0,948	0,984	0,901	0,944	0,562	0,971	0,827	0,863	0,346	0,865	0,588
aml28	0,999	0,989	1,000	0,989	0,998	0,808	0,999	0,999	1,000	0,682	0,975	0,976
arrhythmia	0,613	0,571	0,617	0,597	0,591	0,569	0,542	0,593	0,542	0,546	0,573	0,588
atom	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
banana	1,000	0,811	0,961	0,920	1,000	0,831	0,825	1,000	0,829	0,524	1,000	0,866
breast	0,942	0,968	0,954	0,949	0,938	0,946	0,960	0,923	0,958	0,909	0,914	0,942
chainlink	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	0,988
CinC_100_10_15_20	1,000	0,674	0,857	0,983	1,000	0,537	0,606	1,000	0,554	0,857	0,983	0,857
CinC_100_15_15_20	0,891	0,606	0,857	0,863	0,926	0,543	0,589	0,891	0,583	0,857	0,891	0,857
CinC_1000_10_15_20	1,000	0,654	0,857	1,000	1,000	0,509	0,517	1,000	0,512	0,857	0,999	0,857
CinC_1000_15_15_15	0,858	0,648	0,857	0,885	0,938	0,508	0,505	0,886	0,507	0,857	0,857	0,857
CinC_1000_15_15_20	0,859	0,633	0,857	0,877	0,951	0,525	0,643	0,885	0,527	0,857	0,857	0,857
CinC_1000_15_15_30	0,857	0,639	0,857	0,901	0,991	0,519	0,524	0,872	0,522	0,857	0,857	0,857
Circles_100_15_15_20	1,000	0,947	1,000	0,973	1,000	0,920	0,933	0,827	0,973	0,667	0,827	0,920
Circles_1000_10_15_20	1,000	0,667	0,856	0,937	1,000	0,840	1,000	0,937	0,685	0,667	0,937	0,983
Circles_1000_100_15_20	0,853	0,968	0,879	0,967	0,967	0,803	0,968	0,772	0,803	0,667	0,676	0,881
Circles_1000_100_5_20	0,829	0,987	0,859	0,901	0,992	0,775	0,763	0,768	0,787	0,667	0,993	0,855
Circles_1000_100_50_20	0,892	0,851	0,897	0,825	0,876	0,884	0,852	0,784	0,873	0,667	0,876	0,871
Circles_1000_15_15_20	1,000	0,988	0,989	1,000	1,000	0,717	0,899	0,877	0,732	0,667	0,899	0,877
CirclesWithOutliers_100_10_10_01	0,967	0,967	0,967	0,952	0,971	0,952	0,914	0,967	0,786	0,476	0,967	0,952
CirclesWithOutliers_100_10_10_02	0,936	0,936	0,932	0,936	0,955	0,909	0,927	0,950	0,823	0,455	0,936	0,909
CirclesWithOutliers_1000_10_10_005	0,982	0,839	0,983	0,980	0,990	0,976	0,787	0,989	0,815	0,488	0,988	0,976
CirclesWithOutliers_1000_10_10_01	0,968	0,784	0,966	0,966	0,979	0,952	0,752	0,978	0,791	0,476	0,952	0,952
CirclesWithOutliers_1000_10_10_02	0,928	0,894	0,929	0,923	0,954	0,909	0,727	0,953	0,763	0,455	0,909	0,909
Coil20	0,863	0,857	0,881	0,882	0,967	0,100	0,882	0,919	0,855	0,242	0,317	0,935
compound	0,872	0,647	0,865	0,872	0,845	0,627	0,752	0,925	0,727	0,396	0,727	0,642
Corners_100_1_5_5	1,000	1,000	1,000	1,000	1,000	0,500	1,000	1,000	1,000	0,330	0,250	0,510

Corners_1000_0_10_5	0,894	0,922	0,910	0,992	0,959	0,500	0,941	0,667	0,992	0,480	0,894	0,501
Corners_1000_05_10_5	1,000	1,000	1,000	1,000	1,000	0,500	1,000	0,823	1,000	0,250	1,000	0,500
Corners_1000_1_10_5	1,000	1,000	1,000	1,000	1,000	0,500	1,000	1,000	1,000	0,499	0,980	0,501
Corners_1000_1_5_5	1,000	1,000	1,000	1,000	1,000	0,500	1,000	1,000	1,000	0,356	1,000	0,500
cpu	0,837	0,837	0,842	0,866	0,871	0,842	0,828	0,852	0,852	0,828	0,833	0,833
CrescentFullMoon_100_10_15_20	1,000	0,800	0,900	0,950	1,000	0,770	0,800	1,000	0,800	0,750	1,000	0,890
CrescentFullMoon_1000_10_15_20	1,000	0,607	0,909	0,878	1,000	0,771	0,631	1,000	0,718	0,750	1,000	0,870
CrescentFullMoon_1000_15_15_15	0,926	0,628	0,889	0,815	0,923	0,761	0,621	0,882	0,762	0,750	0,815	0,866
CrescentFullMoon_1000_15_15_20	0,910	0,631	0,913	0,838	0,954	0,752	0,699	0,921	0,750	0,750	0,750	0,872
CrescentFullMoon_1000_25_15_15	0,876	0,631	0,884	0,905	0,972	0,743	0,637	0,818	0,772	0,750	0,637	0,890
cure-t0-2000n-2D	1,000	0,687	0,800	0,923	1,000	0,737	0,507	1,000	0,478	0,800	0,923	0,800
curves1	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
curves2	1,000	1,000	1,000	1,000	1,000	0,823	1,000	1,000	0,823	0,682	1,000	0,702
dermatology	0,956	0,970	0,978	0,943	0,852	0,503	0,967	0,858	0,967	0,503	0,784	0,855
diabetes	0,507	0,536	0,525	0,520	0,514	0,486	0,493	0,493	0,471	0,434	0,561	0,477
ecoli	0,827	0,848	0,857	0,833	0,821	0,643	0,807	0,821	0,795	0,628	0,821	0,815
flame	0,975	0,813	0,988	0,879	0,983	0,854	0,825	0,983	0,842	0,888	0,983	0,796
gaussians1	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
german	0,715	0,709	0,717	0,720	0,724	0,711	0,710	0,710	0,715	0,706	0,700	0,718
glass	0,626	0,565	0,626	0,626	0,621	0,481	0,607	0,584	0,589	0,449	0,607	0,612
haberman	0,765	0,752	0,775	0,768	0,778	0,758	0,748	0,761	0,758	0,742	0,755	0,778
heart-statlog	0,807	0,826	0,811	0,831	0,815	0,811	0,819	0,763	0,848	0,648	0,804	0,807
iono	0,920	0,917	0,934	0,899	0,957	0,915	0,897	0,946	0,866	0,849	0,872	0,954
iris	0,973	0,967	0,980	0,983	0,973	0,667	0,980	0,920	0,980	0,687	0,973	0,947
jain	0,946	0,879	0,995	0,922	1,000	0,887	0,855	0,997	0,887	0,740	0,922	0,925
Outliers_1000_15_10_005_5	0,697	0,592	0,719	0,695	0,569	0,717	0,598	0,612	0,585	0,497	0,577	0,478
Outliers_1000_15_15_005_2	0,946	0,825	0,877	0,991	0,954	0,900	0,748	0,991	0,738	0,817	0,954	0,503
Outliers_1000_15_15_005_5	0,948	0,809	0,920	0,950	0,961	0,900	0,743	0,949	0,732	0,463	0,946	0,500
pathbased	0,760	0,737	0,777	0,833	0,967	0,633	0,717	0,947	0,760	0,367	0,947	0,583
pmf	0,983	0,982	0,980	0,981	0,988	0,732	0,982	0,988	0,978	0,732	0,982	0,978
rings	0,607	0,632	0,618	0,686	0,665	0,445	0,606	0,586	0,553	0,343	0,553	0,531
segment	0,805	0,790	0,803	0,789	0,780	0,286	0,783	0,724	0,781	0,267	0,784	0,715
shapes	1,000	1,000	1,000	1,000	1,000	0,500	1,000	1,000	1,000	0,250	0,999	0,538
simplex	1,000	1,000	1,000	1,000	1,000	0,500	1,000	1,000	1,000	0,500	0,994	1,000

smile1	1,000	0,986	0,808	0,950	1,000	0,500	0,884	1,000	0,755	0,250	1,000	0,626
sonar	0,755	0,712	0,788	0,740	0,745	0,712	0,707	0,654	0,721	0,649	0,760	0,745
spambase	0,878	0,857	0,918	0,905	0,793	0,848	0,856	0,749	0,859	0,651	0,761	0,793
spherical_4_3	1,000	1,000	1,000	1,000	1,000	0,500	1,000	1,000	1,000	0,500	1,000	1,000
Spirals_100_720_0	0,760	0,630	0,770	0,630	0,600	0,560	0,560	0,650	0,560	0,560	0,560	0,640
Spirals_1000_1440_2	0,540	0,538	0,560	0,605	0,539	0,544	0,532	0,521	0,537	0,534	0,528	0,523
Spirals_1000_720_0	0,607	0,585	0,627	0,661	0,507	0,516	0,543	0,672	0,536	0,613	0,614	0,504
Spirals_1000_720_1	0,603	0,541	0,625	0,526	0,571	0,532	0,526	0,794	0,524	0,590	0,500	0,512
Spirals_1000_720_2	0,599	0,573	0,636	0,633	0,549	0,539	0,539	0,549	0,540	0,545	0,517	0,526
Spirals_1000_720_3	0,599	0,587	0,617	0,596	0,559	0,528	0,545	0,532	0,528	0,517	0,532	0,571
Spirals_1000_720_4	0,598	0,572	0,604	0,584	0,575	0,579	0,565	0,534	0,592	0,593	0,565	0,578
square1	0,979	0,981	0,982	0,981	0,976	0,500	0,980	0,843	0,980	0,512	0,950	0,522
s-set1	0,997	0,998	0,998	0,997	0,974	0,141	0,999	0,375	0,998	0,070	0,403	0,141
st900	0,890	0,918	0,924	0,781	0,490	0,252	0,908	0,518	0,922	0,129	0,908	0,363
tae	0,563	0,556	0,576	0,570	0,536	0,470	0,543	0,483	0,570	0,464	0,517	0,530
target	1,000	0,675	0,809	0,894	1,000	0,534	0,678	0,988	0,639	0,513	0,984	0,764
tetra	1,000	1,000	1,000	1,000	1,000	0,500	1,000	1,000	1,000	0,500	1,000	0,750
threenorm	0,849	0,705	0,884	0,816	0,914	0,646	0,691	0,846	0,672	0,637	0,846	0,695
thy	0,949	0,953	0,949	0,949	0,907	0,837	0,972	0,856	0,944	0,791	0,907	0,944
triangle1	1,000	1,000	1,000	1,000	1,000	0,500	1,000	1,000	0,998	0,250	0,998	0,668
twenty	1,000	0,900	1,000	1,000	1,000	0,100	1,000	0,932	1,000	0,050	0,231	0,200
twodiamonds	1,000	1,000	1,000	1,000	1,000	1,000	1,000	0,994	1,000	0,959	1,000	0,769
TwoHalfMoon_100_10_15_2	0,880	0,860	0,910	0,900	0,880	0,880	0,860	0,820	0,870	0,740	0,870	0,750
TwoHalfMoon_100_15_15_2	0,840	0,770	0,840	0,790	0,840	0,780	0,770	0,820	0,780	0,710	0,760	0,760
TwoHalfMoon_1000_10_15_1	1,000	0,798	0,894	0,894	1,000	0,822	0,798	1,000	0,809	0,685	0,787	0,701
TwoHalfMoon_1000_10_15_2	1,000	0,783	0,900	0,869	1,000	0,807	0,779	1,000	0,795	0,684	0,827	0,696
TwoHalfMoon_1000_15_15_1	0,836	0,777	0,839	0,818	0,776	0,773	0,766	0,699	0,774	0,685	0,774	0,737
TwoHalfMoon_1000_15_15_2	0,861	0,794	0,862	0,806	0,868	0,808	0,797	0,885	0,805	0,722	0,885	0,750
vehicle	0,654	0,559	0,648	0,567	0,505	0,424	0,550	0,527	0,591	0,402	0,479	0,494
vowel	0,560	0,536	0,573	0,542	0,503	0,364	0,530	0,489	0,538	0,316	0,506	0,440
wdbc	0,942	0,968	0,954	0,951	0,937	0,946	0,958	0,923	0,958	0,884	0,933	0,935
wine	0,978	0,983	0,994	0,978	0,949	0,702	0,983	0,876	0,989	0,697	0,848	0,916
wingnut	1,000	0,996	1,000	1,000	1,000	0,984	0,996	1,000	0,988	0,995	0,996	0,681
wisc	0,974	0,974	0,977	0,976	0,980	0,974	0,974	0,971	0,974	0,976	0,973	0,971

xclara	1,000	0,998	1,000	0,999	0,993	0,703	0,998	0,892	0,999	0,703	0,998	0,737
xor	0,495	0,480	0,520	0,580	0,546	0,380	0,477	0,498	0,475	0,349	0,534	0,479
zoo	0,931	0,911	0,941	0,941	0,921	0,604	0,911	0,950	0,950	0,604	0,881	0,921

3 priedas. Duomenų klasterizavimo rezultatai („JScore“ metrika)

Duomenų rinkinys	Agg	BGM	BIRCH	CBMIDE	DBSCAN	FCM	GMM	HDBSCAN	K-means	MIDE	OPTICS	ST-DBSCAN
1balance-scale	0,628	0,533	0,680	0,745	0,611	0,859	0,538	0,644	0,606	0,440	0,571	0,630
3-spiral	1,000	0,274	0,315	0,418	1,000	0,270	0,281	1,000	0,981	0,337	1,000	0,313
aggregation	1,000	0,941	0,969	0,834	0,917	0,484	0,948	0,802	0,786	0,267	0,799	0,500
aml28	0,998	0,980	1,000	0,980	0,996	0,796	0,998	0,998	1,000	0,575	0,968	0,970
arrhythmia	0,476	0,444	0,482	0,465	0,460	0,452	0,406	0,463	0,406	0,409	0,456	0,457
atom	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
banana	1,000	0,682	0,924	0,852	1,000	0,712	0,702	1,000	0,707	0,512	1,000	0,764
breast	0,891	0,939	0,913	0,903	0,885	0,897	0,923	0,863	0,919	0,833	0,843	0,891
chainlink	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	0,976
CinC_100_10_15_20	1,000	0,596	0,803	0,967	1,000	0,454	0,537	1,000	0,465	0,803	0,967	0,803
CinC_100_15_15_20	0,829	0,530	0,803	0,784	0,873	0,459	0,493	0,840	0,485	0,803	0,840	0,803
CinC_1000_10_15_20	1,000	0,550	0,803	1,000	1,000	0,441	0,449	1,000	0,442	0,803	0,998	0,803
CinC_1000_15_15_15	0,803	0,543	0,803	0,821	0,891	0,440	0,439	0,827	0,441	0,803	0,803	0,803
CinC_1000_15_15_20	0,804	0,531	0,803	0,808	0,911	0,450	0,541	0,823	0,451	0,803	0,803	0,803
CinC_1000_15_15_30	0,802	0,536	0,803	0,836	0,983	0,447	0,453	0,810	0,449	0,803	0,803	0,803
Circles_100_15_15_20	1,000	0,900	1,000	0,955	1,000	0,853	0,876	0,754	0,948	0,606	0,754	0,854
Circles_1000_10_15_20	1,000	0,500	0,755	0,886	1,000	0,725	1,000	0,886	0,521	0,606	0,886	0,966
Circles_1000_100_15_20	0,752	0,938	0,789	0,936	0,936	0,684	0,938	0,650	0,684	0,606	0,658	0,793
Circles_1000_100_5_20	0,719	0,974	0,759	0,824	0,984	0,650	0,637	0,643	0,665	0,606	0,987	0,754
Circles_1000_100_50_20	0,809	0,748	0,817	0,706	0,784	0,796	0,750	0,656	0,781	0,606	0,784	0,777
Circles_1000_15_15_20	1,000	0,976	0,979	1,000	1,000	0,559	0,817	0,797	0,578	0,606	0,817	0,787
CirclesWithOutliers_100_10_10_01	0,946	0,946	0,946	0,931	0,952	0,931	0,871	0,946	0,726	0,466	0,946	0,930
CirclesWithOutliers_100_10_10_02	0,897	0,897	0,891	0,897	0,921	0,872	0,886	0,915	0,747	0,437	0,897	0,871
CirclesWithOutliers_1000_10_10_005	0,971	0,795	0,973	0,969	0,982	0,964	0,755	0,981	0,776	0,482	0,979	0,963
CirclesWithOutliers_1000_10_10_01	0,947	0,737	0,945	0,945	0,962	0,930	0,713	0,961	0,744	0,466	0,930	0,930
CirclesWithOutliers_1000_10_10_02	0,886	0,841	0,888	0,881	0,921	0,871	0,676	0,919	0,700	0,437	0,871	0,869
Coil20	0,840	0,811	0,845	0,800	0,961	0,100	0,861	0,911	0,814	0,217	0,275	0,919
compound	0,823	0,593	0,809	0,823	0,769	0,547	0,705	0,887	0,651	0,306	0,651	0,551
Corners_100_1_5_5	1,000	1,000	1,000	1,000	1,000	0,500	1,000	1,000	1,000	0,273	0,250	0,477
Corners_1000_0_10_5	0,811	0,856	0,839	0,983	0,938	0,497	0,888	0,599	0,983	0,431	0,811	0,492

Corners_1000_05_10_5	1,000	1,000	1,000	1,000	1,000	0,500	1,000	0,758	1,000	0,250	1,000	0,500
Corners_1000_1_10_5	1,000	1,000	1,000	1,000	1,000	0,500	1,000	1,000	1,000	0,496	0,961	0,497
Corners_1000_1_5_5	1,000	1,000	1,000	1,000	1,000	0,500	1,000	1,000	1,000	0,296	1,000	0,500
cpu	0,755	0,756	0,768	0,799	0,794	0,801	0,760	0,776	0,774	0,760	0,782	0,789
CrescentFullMoon_100_10_15_20	1,000	0,671	0,821	0,907	1,000	0,630	0,671	1,000	0,671	0,682	1,000	0,811
CrescentFullMoon_1000_10_15_20	1,000	0,438	0,839	0,790	1,000	0,632	0,462	1,000	0,565	0,682	1,000	0,783
CrescentFullMoon_1000_15_15_15	0,866	0,459	0,809	0,705	0,861	0,626	0,452	0,803	0,627	0,682	0,705	0,777
CrescentFullMoon_1000_15_15_20	0,841	0,463	0,846	0,736	0,914	0,612	0,546	0,860	0,611	0,682	0,682	0,785
CrescentFullMoon_1000_25_15_15	0,791	0,464	0,802	0,833	0,946	0,607	0,487	0,720	0,645	0,682	0,487	0,811
cure-t0-2000n-2D	1,000	0,531	0,723	0,877	1,000	0,647	0,504	1,000	0,393	0,723	0,877	0,723
curves1	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
curves2	1,000	1,000	1,000	1,000	1,000	0,699	1,000	1,000	0,699	0,518	1,000	0,541
dermatology	0,924	0,945	0,959	0,897	0,840	0,481	0,940	0,848	0,940	0,460	0,755	0,844
diabetes	0,378	0,373	0,365	0,370	0,351	0,405	0,335	0,334	0,319	0,333	0,441	0,394
ecoli	0,735	0,762	0,777	0,734	0,755	0,556	0,711	0,755	0,699	0,534	0,739	0,746
flame	0,951	0,685	0,975	0,792	0,967	0,746	0,703	0,973	0,728	0,799	0,973	0,697
gaussians1	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
german	0,630	0,616	0,624	0,627	0,616	0,627	0,617	0,629	0,631	0,628	0,634	0,595
glass	0,483	0,420	0,499	0,482	0,478	0,363	0,463	0,445	0,447	0,318	0,462	0,472
haberman	0,644	0,629	0,658	0,662	0,663	0,638	0,621	0,640	0,638	0,668	0,632	0,660
heart-statlog	0,678	0,704	0,684	0,688	0,690	0,683	0,693	0,621	0,737	0,486	0,673	0,678
iono	0,853	0,848	0,878	0,774	0,918	0,844	0,815	0,905	0,768	0,742	0,777	0,913
iris	0,949	0,937	0,961	0,961	0,949	0,667	0,961	0,862	0,961	0,665	0,949	0,901
jain	0,900	0,788	0,989	0,859	1,000	0,801	0,750	0,996	0,801	0,672	0,859	0,864
Outliers_1000_15_10_005_5	0,543	0,442	0,589	0,536	0,413	0,583	0,472	0,456	0,455	0,428	0,412	0,329
Outliers_1000_15_15_005_2	0,909	0,738	0,811	0,982	0,920	0,861	0,690	0,982	0,684	0,733	0,920	0,479
Outliers_1000_15_15_005_5	0,911	0,723	0,871	0,915	0,928	0,866	0,673	0,912	0,685	0,430	0,909	0,476
pathbased	0,630	0,602	0,648	0,726	0,936	0,565	0,580	0,899	0,628	0,350	0,899	0,524
pmf	0,971	0,969	0,967	0,964	0,978	0,704	0,970	0,978	0,966	0,704	0,970	0,964
rings	0,451	0,499	0,456	0,554	0,641	0,349	0,462	0,477	0,405	0,338	0,405	0,444
segment	0,714	0,727	0,706	0,673	0,675	0,286	0,686	0,648	0,681	0,227	0,682	0,639
shapes	1,000	1,000	1,000	1,000	1,000	0,500	1,000	1,000	1,000	0,250	0,998	0,476
simplex	1,000	1,000	1,000	1,000	1,000	0,500	1,000	1,000	1,000	0,500	0,990	1,000
smile1	1,000	0,973	0,706	0,907	1,000	0,462	0,807	1,000	0,639	0,250	1,000	0,556

sonar	0,606	0,553	0,652	0,592	0,594	0,553	0,548	0,518	0,564	0,481	0,612	0,595
spambase	0,783	0,751	0,849	0,726	0,658	0,738	0,750	0,604	0,754	0,500	0,636	0,658
spherical_4_3	1,000	1,000	1,000	1,000	1,000	0,500	1,000	1,000	1,000	0,500	1,000	1,000
Spirals_100_720_0	0,613	0,463	0,627	0,463	0,431	0,389	0,389	0,490	0,389	0,389	0,389	0,476
Spirals_1000_1440_2	0,392	0,368	0,390	0,434	0,424	0,374	0,363	0,352	0,367	0,364	0,365	0,354
Spirals_1000_720_0	0,437	0,417	0,458	0,498	0,344	0,348	0,384	0,511	0,375	0,443	0,459	0,354
Spirals_1000_720_1	0,433	0,374	0,456	0,393	0,496	0,362	0,393	0,660	0,355	0,418	0,500	0,376
Spirals_1000_720_2	0,429	0,432	0,467	0,463	0,491	0,369	0,382	0,496	0,370	0,376	0,391	0,400
Spirals_1000_720_3	0,429	0,422	0,450	0,425	0,446	0,359	0,375	0,464	0,359	0,358	0,464	0,401
Spirals_1000_720_4	0,429	0,402	0,435	0,413	0,417	0,410	0,394	0,381	0,423	0,422	0,394	0,409
square1	0,959	0,963	0,965	0,963	0,956	0,421	0,961	0,738	0,961	0,488	0,908	0,468
s-set1	0,994	0,996	0,996	0,994	0,952	0,136	0,997	0,347	0,996	0,069	0,387	0,136
st900	0,805	0,849	0,861	0,646	0,430	0,229	0,833	0,445	0,857	0,120	0,833	0,316
tae	0,393	0,388	0,406	0,402	0,393	0,384	0,377	0,353	0,399	0,308	0,422	0,394
target	1,000	0,654	0,682	0,828	1,000	0,366	0,654	0,988	0,619	0,499	0,977	0,625
tetra	1,000	1,000	1,000	1,000	1,000	0,500	1,000	1,000	1,000	0,500	1,000	0,750
threenorm	0,738	0,549	0,792	0,690	0,842	0,478	0,530	0,756	0,508	0,493	0,756	0,537
thy	0,904	0,912	0,904	0,904	0,834	0,771	0,947	0,774	0,896	0,698	0,835	0,895
triangle1	1,000	1,000	1,000	1,000	1,000	0,463	1,000	1,000	0,996	0,250	0,996	0,615
twenty	1,000	0,899	1,000	0,722	1,000	0,100	1,000	0,922	1,000	0,050	0,202	0,200
twodiamonds	1,000	1,000	1,000	1,000	1,000	1,000	1,000	0,989	1,000	0,921	1,000	0,626
TwoHalfMoon_100_10_15_2	0,786	0,754	0,835	0,818	0,786	0,786	0,754	0,794	0,770	0,587	0,770	0,600
TwoHalfMoon_100_15_15_2	0,724	0,626	0,724	0,658	0,724	0,639	0,626	0,700	0,639	0,551	0,616	0,616
TwoHalfMoon_1000_10_15_1	1,000	0,664	0,808	0,808	1,000	0,698	0,664	1,000	0,679	0,521	0,649	0,540
TwoHalfMoon_1000_10_15_2	1,000	0,644	0,818	0,768	1,000	0,676	0,638	1,000	0,660	0,520	0,707	0,534
TwoHalfMoon_1000_15_15_1	0,718	0,636	0,723	0,692	0,634	0,630	0,621	0,552	0,631	0,521	0,631	0,588
TwoHalfMoon_1000_15_15_2	0,756	0,658	0,757	0,680	0,769	0,678	0,663	0,794	0,674	0,565	0,794	0,600
vehicle	0,525	0,404	0,568	0,454	0,346	0,349	0,419	0,473	0,507	0,346	0,374	0,389
vowel	0,393	0,379	0,406	0,418	0,422	0,324	0,408	0,409	0,378	0,232	0,429	0,364
wdbc	0,891	0,939	0,913	0,878	0,881	0,897	0,919	0,863	0,919	0,793	0,875	0,878
wine	0,956	0,967	0,989	0,956	0,904	0,663	0,967	0,829	0,978	0,637	0,737	0,845
wingnut	1,000	0,992	1,000	1,000	1,000	0,969	0,992	1,000	0,977	0,990	0,992	0,516
wisc	0,950	0,950	0,955	0,953	0,961	0,950	0,950	0,945	0,950	0,953	0,947	0,945
xclara	1,000	0,995	1,000	0,997	0,990	0,681	0,995	0,846	0,998	0,635	0,995	0,663

xor	0,331	0,316	0,355	0,446	0,377	0,316	0,339	0,344	0,319	0,284	0,370	0,446
zoo	0,897	0,881	0,928	0,922	0,903	0,567	0,884	0,927	0,934	0,515	0,839	0,884

4 priedas. Duomenų klasterizavimo rezultatai (Silueto koeficientas)

Duomenų rinkinys	Agg	BGM	BIRCH	CBMIDE	DBSCAN	FCM	GMM	HDBSCAN	K-means	MIDE	OPTICS	ST-DBSCAN
1balance-scale	0,249	0,121	0,254	0,272	0,374	0,777	0,184	0,300	0,293	0,354	0,148	0,431
3-spiral	0,008	0,235	0,259	0,045	0,008	0,332	0,184	0,008	0,576	-0,057	0,495	-0,027
aggregation	0,519	0,529	0,517	0,415	0,485	0,380	0,503	0,421	0,493	0,379	0,480	0,207
aml28	0,830	0,815	0,831	0,823	0,757	0,700	0,830	0,830	0,831	0,565	0,798	0,800
arrhythmia	0,178	0,136	-0,568	0,041	-0,203	0,631	-0,212	0,492	-0,197	-0,250	0,616	0,058
atom	0,818	0,675	0,830	0,971	0,830	0,971	0,971	0,830	0,818	0,818	0,971	0,830
banana	0,757	0,413	0,680	0,573	0,757	0,447	0,436	0,757	0,442	0,356	0,757	0,516
breast	0,525	0,135	0,546	0,782	0,738	0,560	0,122	0,385	0,192	0,479	0,586	0,745
chainlink	0,552	0,630	0,552	0,997	0,976	0,552	0,633	0,633	0,552	0,633	0,997	0,961
CinC_100_10_15_20	-0,091	0,069	-0,096	-0,104	-0,091	0,381	0,300	-0,091	0,383	-0,097	-0,104	-0,095
CinC_100_15_15_20	-0,183	0,084	-0,155	-0,045	-0,147	0,397	0,372	-0,292	0,390	-0,157	-0,292	-0,153
CinC_1000_10_15_20	-0,086	0,295	-0,090	-0,086	-0,086	0,382	0,376	-0,086	0,382	-0,091	-0,087	-0,089
CinC_1000_15_15_15	-0,083	0,329	-0,122	-0,149	-0,116	0,396	0,412	-0,299	0,412	-0,123	-0,120	-0,121
CinC_1000_15_15_20	-0,190	0,330	-0,122	-0,115	-0,116	0,371	0,317	-0,327	0,371	-0,123	-0,120	-0,121
CinC_1000_15_15_30	-0,146	0,311	-0,099	-0,100	-0,094	0,376	0,377	-0,372	0,377	-0,100	-0,097	-0,098
Circles_100_15_15_20	0,269	0,257	0,269	0,255	0,269	0,256	0,254	-0,060	0,265	-0,045	-0,054	0,249
Circles_1000_10_15_20	0,301	0,499	0,156	0,301	0,301	0,324	0,301	-0,010	0,474	0,124	-0,013	0,285
Circles_1000_100_15_20	0,482	0,365	0,454	0,425	0,368	0,530	0,365	0,527	0,530	0,485	0,319	0,412
Circles_1000_100_5_20	0,520	0,491	0,555	0,521	0,483	0,595	0,617	0,514	0,588	0,556	0,481	0,547
Circles_1000_100_50_20	0,305	0,326	0,300	0,275	0,313	0,298	0,326	0,016	0,318	0,051	0,325	0,320
Circles_1000_15_15_20	0,252	0,247	0,255	0,282	0,252	0,405	0,262	-0,270	0,376	0,185	0,315	0,213
CirclesWithOutliers_100_10_10_01	0,517	0,517	0,517	0,662	0,490	0,666	0,402	0,517	0,435	0,287	0,653	0,664
CirclesWithOutliers_100_10_10_02	0,537	0,584	0,558	0,537	0,532	0,632	0,585	0,545	0,464	0,306	0,604	0,623
CirclesWithOutliers_1000_10_10_005	0,594	0,488	0,582	0,582	0,599	0,666	0,499	0,597	0,486	0,321	0,594	0,500
CirclesWithOutliers_1000_10_10_01	0,527	0,455	0,533	0,523	0,547	0,659	0,476	0,557	0,486	0,321	0,632	0,649
CirclesWithOutliers_1000_10_10_02	0,524	0,432	0,521	0,521	0,525	0,644	0,465	0,529	0,474	0,313	0,621	0,632
Coil20	0,568	0,607	0,569	0,611	0,349	0,289	0,651	0,299	0,585	-0,612	-0,062	0,597
compound	0,480	0,384	0,477	0,490	0,507	0,449	0,376	0,159	0,416	0,352	0,422	0,267
Corners_100_1_5_5	0,599	0,599	0,599	0,599	0,599	0,404	0,599	0,599	0,599	0,035	0,056	0,248
Corners_1000_0_10_5	0,346	0,353	0,330	0,346	0,330	0,322	0,356	0,188	0,368	0,289	0,346	0,194
Corners_1000_05_10_5	0,422	0,422	0,422	0,422	0,422	0,345	0,422	0,321	0,422	0,186	0,422	0,345
Corners_1000_1_10_5	0,484	0,484	0,484	0,484	0,484	0,377	0,484	0,484	0,484	0,370	0,451	0,330
Corners_1000_1_5_5	0,607	0,607	0,607	0,607	0,607	0,418	0,607	0,607	0,607	0,064	0,596	0,414

cpu	0,800	0,781	0,601	0,473	0,481	0,568	0,803	0,395	0,568	0,297	0,725	0,674
CrescentFullMoon_100_10_15_20	0,215	0,386	0,293	0,104	0,215	0,403	0,386	0,215	0,386	0,151	0,215	0,229
CrescentFullMoon_1000_10_15_20	0,207	0,408	0,215	0,223	0,207	0,338	0,377	0,207	0,390	0,315	0,207	0,212
CrescentFullMoon_1000_15_15_15	0,278	0,399	0,309	0,336	0,247	0,318	0,387	0,155	0,321	0,306	0,346	0,344
CrescentFullMoon_1000_15_15_20	0,269	0,391	0,280	0,218	0,214	0,338	0,396	0,051	0,372	0,226	0,246	0,304
CrescentFullMoon_1000_25_15_15	0,378	0,338	0,379	0,305	0,260	0,327	0,436	0,211	0,340	0,336	0,346	0,360
cure-t0-2000n-2D	0,605	0,324	0,433	0,463	0,605	0,387	0,456	0,605	0,325	0,446	0,476	0,476
curves1	0,950	0,950	0,950	0,950	0,950	0,950	0,950	0,950	0,950	0,950	0,950	0,950
curves2	0,949	0,949	0,949	0,949	0,949	0,489	0,949	0,949	0,489	0,205	0,949	0,239
dermatology	-0,041	-0,023	-0,032	0,660	0,164	0,519	0,191	0,510	-0,033	0,429	0,437	0,230
diabetes	-0,117	0,115	0,000	0,056	0,033	0,175	0,004	0,160	0,135	0,025	0,149	0,195
ecoli	0,231	0,347	0,255	0,352	0,619	0,535	0,305	0,627	0,360	0,653	0,554	0,233
flame	0,344	0,371	0,343	0,252	0,324	0,372	0,376	0,286	0,380	0,371	0,365	0,236
gaussians1	0,920	0,920	0,920	0,920	0,920	0,920	0,920	0,920	0,920	0,920	0,920	0,920
german	0,724	0,845	0,318	0,280	0,265	0,924	0,845	0,244	0,914	0,734	0,745	0,766
glass	0,228	-0,036	0,123	0,032	0,198	0,145	0,252	0,445	0,230	-0,148	0,000	0,068
haberman	0,369	0,318	0,358	0,219	0,373	0,398	0,283	0,323	0,329	0,450	0,329	0,353
heart-statlog	0,328	0,049	0,075	0,115	0,090	0,048	0,057	0,237	0,078	0,142	0,315	0,328
iono	0,417	0,219	0,404	0,337	0,337	0,415	0,128	0,244	0,547	0,547	0,326	0,339
iris	0,595	0,586	0,848	0,848	0,675	0,685	0,595	0,718	0,595	0,417	0,676	0,567
jain	0,433	0,479	0,396	0,441	0,404	0,472	0,486	0,062	0,472	0,182	0,415	0,373
Outliers_1000_15_10_005_5	0,191	0,344	0,156	0,145	0,218	0,249	0,320	0,316	0,346	0,240	0,151	0,417
Outliers_1000_15_15_005_2	0,275	0,379	0,350	0,416	0,333	0,392	0,418	0,357	0,418	-0,036	0,345	0,173
Outliers_1000_15_15_005_5	0,360	0,401	0,358	0,360	0,334	0,412	0,374	0,349	0,420	-0,251	0,366	0,235
pathbased	0,524	0,537	0,519	0,327	0,275	0,512	0,531	0,260	0,537	0,185	0,326	0,241
pmf	0,810	0,771	0,809	0,841	0,809	0,539	0,807	0,809	0,901	0,539	0,746	0,924
rings	0,315	0,202	0,280	0,173	0,128	0,295	0,178	-0,028	0,374	0,054	0,356	0,283
segment	0,154	0,139	0,204	0,537	0,541	0,497	0,563	0,570	0,544	-0,172	0,535	0,559
shapes	0,764	0,764	0,764	0,764	0,764	0,466	0,764	0,764	0,764	0,764	0,762	0,273
simplex	0,909	0,837	0,880	0,874	1,000	0,533	0,909	1,000	0,909	0,732	0,850	1,000
smile1	0,807	0,782	0,506	0,697	0,807	0,554	0,666	0,807	0,456	0,315	0,807	0,452
sonar	0,183	0,228	0,160	0,132	0,217	0,144	0,161	-0,086	0,240	0,066	0,206	0,170
spambase	0,246	0,301	0,227	0,317	0,297	0,345	0,305	-0,868	0,358	-0,171	0,069	0,290
spherical_4_3	0,721	0,811	0,887	0,692	0,890	0,582	1,000	0,890	0,692	0,582	0,890	1,000
Spirals_100_720_0	0,261	0,371	0,266	0,371	0,311	0,351	0,359	0,299	0,359	0,359	0,359	0,080
Spirals_1000_1440_2	0,263	0,291	0,235	0,179	0,151	0,292	0,295	0,278	0,294	0,291	0,264	0,283
Spirals_1000_720_0	0,306	0,313	0,204	-0,033	0,289	0,297	0,310	0,192	0,287	0,244	-0,064	0,286
Spirals_1000_720_1	0,300	0,278	0,284	0,315	0,315	0,298	0,304	0,141	0,303	0,244	0,321	0,239

Spirals_1000_720_2	0,269	0,319	0,247	0,096	0,277	0,286	0,307	0,258	0,295	0,240	0,143	0,263
Spirals_1000_720_3	0,266	0,301	0,297	0,176	0,290	0,279	0,268	0,227	0,281	0,238	0,268	0,280
Spirals_1000_720_4	0,293	0,301	0,295	0,110	0,303	0,302	0,290	0,270	0,308	0,237	0,290	0,302
square1	0,598	0,603	0,600	0,598	0,597	0,316	0,603	0,401	0,603	0,116	0,568	0,259
s-set1	0,710	0,711	0,711	0,695	0,661	0,385	0,711	0,044	0,711	0,151	0,239	0,325
st900	0,446	0,485	0,486	0,301	0,255	0,364	0,474	0,211	0,492	0,196	0,484	0,296
tae	0,077	0,048	0,202	0,167	0,011	0,025	0,046	0,129	0,097	-0,015	0,049	0,079
target	0,368	0,589	0,421	0,290	0,368	0,217	0,590	0,351	0,611	0,315	0,343	0,323
tetra	0,587	0,587	0,587	0,876	0,876	0,575	0,876	0,889	0,876	0,603	0,876	0,538
threenorm	0,329	0,398	0,288	0,237	0,298	0,367	0,382	0,186	0,377	0,336	0,186	0,344
thy	0,408	0,404	0,376	0,386	0,235	0,322	0,851	0,431	0,391	0,243	0,683	0,353
triangle1	0,784	0,784	0,784	0,784	0,784	0,467	0,784	0,784	0,785	0,365	0,782	0,426
twenty	0,775	0,717	0,775	0,425	0,775	0,432	0,775	0,689	0,775	-0,024	0,040	0,291
twodiamonds	0,672	0,672	0,672	0,672	0,672	0,672	0,672	0,611	0,672	0,615	0,678	0,290
TwoHalfMoon_100_10_15_2	0,457	0,470	0,416	0,379	0,455	0,457	0,473	0,336	0,466	0,514	0,476	0,517
TwoHalfMoon_100_15_15_2	0,437	0,508	0,437	0,138	0,437	0,474	0,504	0,280	0,500	0,353	0,448	0,475
TwoHalfMoon_1000_10_15_1	0,256	0,477	0,390	0,402	0,256	0,463	0,482	0,256	0,474	0,523	0,285	0,523
TwoHalfMoon_1000_10_15_2	0,236	0,470	0,335	0,313	0,236	0,452	0,474	0,236	0,462	0,502	0,370	0,502
TwoHalfMoon_1000_15_15_1	0,434	0,460	0,426	0,241	0,488	0,493	0,498	0,202	0,498	0,527	0,498	0,451
TwoHalfMoon_1000_15_15_2	0,448	0,516	0,447	0,344	0,345	0,478	0,515	0,237	0,507	0,527	0,515	0,524
vehicle	0,006	0,043	0,313	-0,012	0,289	0,028	0,129	-0,056	0,138	0,192	0,309	0,255
vowel	0,146	0,116	0,158	0,120	0,227	0,401	0,184	0,273	0,184	-0,386	0,277	0,107
wdbc	0,525	0,135	0,546	0,312	0,478	0,560	0,126	0,385	0,192	0,554	0,745	0,558
wine	0,191	0,199	0,194	0,678	0,162	0,162	0,687	-0,053	0,203	0,630	0,461	0,160
wingnut	0,506	0,505	0,506	0,506	0,506	0,504	0,505	0,506	0,505	0,506	0,502	0,316
wisc	0,601	0,913	0,928	0,596	0,873	0,710	0,525	0,731	0,663	0,783	0,631	0,883
xclara	0,694	0,695	0,694	0,694	0,655	0,542	0,695	0,577	0,694	0,472	0,695	0,307
xor	-0,099	0,420	0,305	0,215	0,034	0,327	0,182	0,308	0,376	0,184	0,150	0,047
zoo	0,591	0,757	0,804	0,741	0,461	0,505	0,770	0,612	0,809	0,384	0,364	0,251

5 priedas. Duomenų klasterizavimo metodų testavimo realizacijos pavyzdys

```
from sklearn.cluster import KMeans, Birch
from sklearn.preprocessing import MinMaxScaler, StandardScaler,
RobustScaler, MaxAbsScaler, QuantileTransformer, PowerTransformer,
Normalizer
import time
import glob
import numpy as np
import pandas as pd
from datetime import datetime
from tqdm import tqdm
from pathlib import Path
import json
import random

from utils.get_machine import get_cpu_info, get_gpu_info
from utils.get_machine_id import get_machine_id
from utils.track_exist_in_DB import track_exists
from utils.track_fully_tested import track_finished
from utils.failed_to_DB import save_failed
from test_models_db import test_models

full_system_info = get_cpu_info()
full_system_info.update(get_gpu_info())
machine_info = json.dumps(full_system_info)
print(machine_info)

machine_id = get_machine_id(machine_info,
full_system_info["computer_network"])

file_process_map = {
    14: ('CBMIDEv2', 'Datasets/AllData/*/*.csv', True),
    23: ('CBMIDEv2', 'Datasets/AllData/*/*.csv', True),
    9: ('AgglomerativeClustering', 'Datasets/AllData/*/*.csv', True),
    11: ('ST-DBSCAN', 'Datasets/AllData/*/*.csv', True),
    13: ('BIRCH', 'Datasets/AllData/*/*.csv', False)
}

method, files_pattern, shuffle = file_process_map.get(machine_id,
(None, None, None))
files = glob.glob(files_pattern)
if shuffle:
    random.shuffle(files)
files = [f.replace('\\', '/') for f in files]

exclude_fully_tested = True

for filepath in tqdm(files):
    if exclude_fully_tested and track_finished(filepath, method):
        print(f"Skipping {filepath} because already in DB")
        continue

    name = filepath.split("/")[-1]
    if name in {'MNIST.csv', 'letter_new.csv', 'usps.csv',
'Coil100.csv', 'Pendigits.csv', 'Shuttle.csv'}:
```

```
        print(f"Skipping {name}")
        continue

    print(filepath)
    df = pd.read_csv(filepath, sep=",")
    full = df.to_numpy()

    if method in ["MIDEv1", "MIDEv2", "CBMIDEv2"] and df.shape[1] >
11:
        print("MIDE too many dimensions")
        continue

    for scale in tqdm(["Raw", "MinMax", "Standard", "Robust",
"MaxAbs", "QuantileNormal", "QuantileUniform",
                       "PowerTransformer", "Normalizer"],
desc="Scalers", leave=False):
        X, Y = full[:, :-1], full[:, -1]

        scalers = {
            "MinMax": MinMaxScaler(),
            "Standard": StandardScaler(),
            "Robust": RobustScaler(),
            "MaxAbs": MaxAbsScaler(),
            "QuantileNormal":
QuantileTransformer(output_distribution='normal'),
            "QuantileUniform":
QuantileTransformer(output_distribution='uniform'),
            "PowerTransformer": PowerTransformer(),
            "Normalizer": Normalizer()
        }

        if scale != "Raw":
            scaler = scalers[scale]
            X = scaler.fit_transform(X)

        clusters = len(np.unique(Y))
        data_folder = Path(filepath)
        filepath = str(data_folder).replace("\\", "/")

        # Generate dataset params based on condition
        dataset_params = {"Name": filepath.split("/")[-1],
                          "Number_of_obs": X.shape[0],
                          "Clusters": clusters,
                          "Scaler": scale,
                          "Dim": X.shape[1]}

        f_path = f'{filepath.split(".")[0]}.json'
        if Path(f_path).exists():
            with open(f_path, 'r') as file:
                data = json.load(file)
                dataset_params.update(data)

        # Example model testing
        if method == "K-means":
            package = "Scikit-learn"
            init_methods = ["k-means++", "random"]
```

```
            algorithms = ["lloyd", "elkan"]
            for init in init_methods:
                for algo in algorithms:
                    params = {"clusters": clusters, "init": init,
"algorithm": algo}
                    if not track_exists(filepath, dataset_params,
method, params, False, machine_id, package):
                        try:
                            start_time = time.time()
                            clustering = KMeans(n_clusters=clusters,
init=init, algorithm=algo).fit(X)
                            results = test_models(df, Y,
clustering.labels_, method, params, (time.time() - start_time),
filepath,
                                                  dataset_params,
machine_id, package)
                        except Exception as e:
                            print(e)
                            save_failed(filepath, method, params,
dataset_params, machine_id, e)
                            continue
```

6 priedas. „Jscore“ metrikos realizavimo kodas

```
import numpy as np
def JScore(truth, pred):
    if (len(truth) == len(pred)):
        A = np.empty([0, len(truth)], bool)
        test = list(set(pred))
        for i in test:
            A = np.vstack([A, (np.array(pred) == i)])
        suma = A.sum(axis=1)

        B = np.empty([0, len(truth)], bool)
        test = list(set(truth))
        for i in test:
            B = np.vstack([B, (np.array(truth) == i)])
        suma2 = B.sum(axis=1)

        C = np.empty([len(suma), len(suma2)], float)

        for i in range(0, len(suma)):
            for j in range(0, len(suma2)):
                C[i, j] = sum(A[i,] & B[j,]) / sum(A[i,] | B[j,])

        M1 = sum(np.amax(C, axis=1) * suma) / A.shape[1]
        M11 = sum(np.amax(C, axis=0) * suma2) / A.shape[1]

        M2 = 2 * M1 * M11 / (M1 + M11)

        return M2
    else:
        print('Truth and Pred have different lengths.')
```

7 priedas. „Jscore“ metrikos reikšmės esant nesutampantiems klasteriams

	Triukšmo lygis (%)					
Metodas	1 %	2 %	5 %	10 %	20 %	50 %
Affinity	0,543	0,502	0,532	0,444	0,465	0,499
Agglomerative	0,990	0,983	0,942	0,880	0,822	0,503
BGMM	1,000	0,998	0,988	0,957	0,971	0,668
Birch	0,985	0,970	0,927	0,858	0,732	0,500
CBMIDE	0,994	0,979	0,961	0,920	0,871	0,778
DBSCAN	1,000	0,998	0,994	0,986	0,986	0,990
FCM	0,985	0,970	0,927	0,849	0,762	0,678
GMM	1,000	0,970	0,996	0,973	0,982	0,609
HDBSCAN	0,996	0,986	0,969	0,924	0,886	0,781
k-means	0,985	0,977	0,942	0,885	0,779	0,451
MeanShift	0,991	0,980	0,948	0,889	0,791	0,497
OPTICS	0,988	0,977	0,944	0,889	0,826	0,969
RobustLinkage	0,951	0,951	0,865	0,460	0,758	0,573
Spectral	0,989	0,979	0,927	0,858	0,732	0,455
ST-DBSCAN	0,983	0,949	0,627	0,586	0,775	0,367

8 priedas. „Jscore“ metrikos reikšmės esant mažai sutampantiems klasteriams

	Triukšmo lygis (%)					
Metodas	1 %	2 %	5 %	10 %	20 %	50 %
Affinity	0,228	0,248	0,249	0,254	0,253	0,202
Agglomerative	0,985	0,975	0,938	0,869	0,773	0,436
BGMM	0,985	0,976	0,953	0,902	0,820	0,449
Birch	0,985	0,968	0,927	0,859	0,734	0,442
CBMIDE	0,985	0,975	0,954	0,915	0,841	0,710
DBSCAN	0,986	0,980	0,953	0,933	0,874	0,799
FCM	0,983	0,968	0,925	0,858	0,763	0,442
GMM	0,983	0,968	0,939	0,889	0,764	0,472
HDBSCAN	0,985	0,980	0,950	0,931	0,874	0,772
k-means	0,983	0,968	0,925	0,873	0,760	0,441
MeanShift	0,986	0,961	0,932	0,878	0,757	0,456
OPTICS	0,985	0,979	0,947	0,930	0,855	0,538
RobustLinkage	0,964	0,919	0,850	0,731	0,745	0,726
Spectral	0,983	0,968	0,925	0,864	0,736	0,440
ST-DBSCAN	0,632	0,613	0,595	0,557	0,482	0,369

9 priedas. „Jscore“ metrikos reikšmės esant stipriai sutampantiems klasteriams

	Triukšmo lygis (%)					
Metodas	1 %	2 %	5 %	10 %	20 %	50 %
Affinity	0,133	0,142	0,139	0,145	0,146	0,142
Agglomerative	0,817	0,792	0,766	0,694	0,617	0,363
BGMM	0,834	0,842	0,741	0,742	0,632	0,383
Birch	0,832	0,811	0,776	0,734	0,631	0,400
CBMIDE	0,821	0,807	0,761	0,735	0,654	0,581
DBSCAN	0,536	0,532	0,455	0,520	0,428	0,540
FCM	0,841	0,828	0,797	0,752	0,639	0,369
GMM	0,843	0,824	0,801	0,718	0,646	0,403
HDBSCAN	0,510	0,508	0,483	0,450	0,430	0,544
k-means	0,839	0,833	0,802	0,747	0,633	0,390
MeanShift	0,793	0,670	0,631	0,609	0,406	0,193
OPTICS	0,329	0,324	0,310	0,270	0,255	0,383
RobustLinkage	0,444	0,512	0,357	0,407	0,498	0,498
Spectral	0,841	0,828	0,582	0,557	0,658	0,392
ST-DBSCAN	0,488	0,473	0,454	0,320	0,383	0,373

UDK 004.8+ 004.93`14](043.3)

SL344. 20xx-xx-xx, xx leidyb. apsk. l. Tiražas 14 egz. Užsakymas xxx.
Išleido Kauno technologijos universitetas, K. Donelaičio g. 73, 44249 Kaunas
Spausdino leidyklos „Technologija“ spaustuvė, Studentų g. 54, 51424 Kaunas