



**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**ELEKTROS IR ELEKTRONIKOS FAKULTETAS**

**Mantas Tamaliūnas**

**ĮRENGINIŲ GEDIMŲ PROGNOZAVIMO TYRIMAS**

Baigiamasis magistro projektas

**Vadovas**

Dr. Kęstas Rimkus

**KAUNAS, 2016**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**ELEKTROS IR ELEKTRONIKOS FAKULTETAS**

**AUTOMATIKOS KATEDRA**

**ĮRENGINIŲ GEDIMŲ PROGNOZAVIMO TYRIMAS**

Baigiamasis magistro projektas  
**Valdymo technologijos (kodas 621H66001)**

**Vadovas**

Dr. Kęstas Rimkus

**Recenzentas**

Doc. dr. Vygandas Vaitkus

**Projektą atliko**

Mantas Tamaliūnas

**KAUNAS, 2016**



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Elektros ir elektronikos fakultetas

(Fakultetas)

Mantas Tamaliūnas

(Studento vardas, pavardė)

Valdymo technologijos, 621H66001

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Įrenginių gedimų prognozavimo tyrimas“

### AKADEMINIO SAŽININGUMO DEKLARACIJA

20 16 m. birželio 2 d.  
Kaunas

Patvirtinu, kad mano **Manto Tamaliūno** baigiamasis projektas tema „Įrenginių gedimų prognozavimo tyrimas“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

\_\_\_\_\_  
(vardą ir pavardę įrašyti ranka)

\_\_\_\_\_  
(parašas)

Tamaliūnas, Mantas. Įrenginių gedimų prognozavimo tyrimas. Valdymo technologijų magistro baigiamasis projektas / vadovas dr. Kęstas Rimkus; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas, automatikos katedra.

Kaunas, 2016. 67 psl.

## SANTRAUKA

Pagrindinė šiame darbe nagrinėjama problema – AB „Volfas Engelman“ įrenginių gedimo tikimybės įvertinimas pasitelkiant dirbtinio intelekto įrankius. Iškeltas projekto tikslas – atlikti įrenginių gedimų prognozės tyrimą pasitelkiant dirbtinį intelektą, nustatyti pasirinktų prognozavimo metodų galimybes bei įvertinti atliktų prognoziavimų tikslumą.

Analitinėje dalyje apžvelgiama įrenginių gedimų svarba vykdant gamybos procesą bei dirbtinio intelekto galimybes. Parenkami dirbtinio neuroninio tinklo, artimiausio kaimyno, atraminio vektoriaus mašinos ir sprendimo medžio klasifikavimo metodai. Išanalizuotos parinktų klasifikavimo metodų charakteristikos, realizavimas, tikslumo įvertinimas.

Eksperimentui pasirinktoje gėrimų gamybos įmonėje apdorojami surinkti objektų požymiai (matuojami prietaisais) bei paruošiamos klasės gedimo įvykiui, gedimo tipui ir įrenginio gedimui atpažinti.

Įvertinus realizuotų klasifikavimo metodų tikslumą (pagal 2016 m. duomenis), aukščiausias rezultatas gautas k-NN klasifikatoriumi prognozuojant gedimo įvykį visai gamybos linijai – 79,4 proc. Atraminio vektoriaus mašinos ir sprendimo medžio klasifikatoriais prognozuojant įrenginio gedimą gauti nepatikimi rezultatai: remiantis SVM metodu – 53,9 proc., remiantis „Bagged tree“ metodu – 54,9 proc. tikslumas. Pasitelkus tyrimo metu sudarytus klasifikavimo požymius nepavyko realizuoti klasifikatorius taip, kad jie nustatytų gedimo tipą.

*Reikšminiai žodžiai: dirbtinis intelektas, klasifikatoriai, gedimo tipas, įrenginio gedimas, prognozavimas.*

Tamaliūnas, Mantas. Machines fault prediction research. Control technologies final project of master degree / supervisor dr. Kęstas Rimkus; Kaunas University of Technology, Faculty of Electrical and Electronics Engineering, department of automation.

Kaunas, 2016. 67 psl.

## SUMMARY

The main problem to solve in this project – SC "Volfas Engelman" machines failure prediction estimation using artificial intelligence tools. The raised goal of the project – to carry out equipment failure prediction studies using selected artificial intelligence prediction methods and to assess the possibilities of the prediction accuracy.

The analytical part provides an overview of equipment failures importance within the context of the production process and artificial intelligence capabilities. Selected artificial neural network, nearest neighbor, support vector machine and decision tree classification methods. The analysis of selected methods, characteristics of classification methods, implementation, accuracy rating is made.

For experimental part beverage production facility objects characteristics (measured by devices) are selected and prepared failure event, type of failure, machine failure classes to recognize.

Evaluated realized classification accuracy of methods (by 2016 year data), the best result obtained by the k-NN classification in predicting failure event for the whole production line – 79.4 percent. Support vector machine and decision tree classification is predicting device failure within unreliable results: based on SVM method – 53.9 percent., based on "Bagged tree" method – 54.9 percent. accuracy. Through realized classification methods and used features in this project can not predict the type of failure.

*Keywords (up to 8 words): artificial intelligence, classifiers, type of failure, device failure, prediction*

## TURINYS

SANTRUMPŲ IR ŽENKLŲ AIŠKINIMO ŽODYNAS .....	8
ĮVADAS .....	10
1. ĮRENGINIŲ GEDIMŲ SVARBA VYKDANT GAMYBOS PROCESĄ .....	11
2. DIRBTINIS INTELEKTAS.....	12
3. KLASIFIKAVIMAS .....	14
3.1. Klasifikavimo metodų parinkimas ir taikymas .....	14
3.2. Objektų klasifikavimo tikslumo įvertinimas.....	16
4. KLASIFIKAVIMO METODŲ ANALIZĖ.....	19
4.1. Neuroninio tinklo klasifikatorius .....	19
4.1.1. Neuroninio tinklo modelis .....	20
4.1.2. Neuroninio tinklo mokymas .....	21
4.2. Artimiausio kaimyno klasifikavimo metodas .....	23
4.3. Atraminų vektorių mašinos.....	25
4.4. Sprendimo medžių – hierarchinis klasifikavimas .....	27
4.4.1. Gaubiamasis sprendimo medžio klasifikavimo metodas.....	28
5. KLASIFIKAVIMO METODŲ REALIZAVIMAS .....	30
5.1. Naudojami duomenys klasifikavimo realizavimui .....	30
5.2. Klasifikavimo metodų mokymas .....	32
5.2.1. Neuroninio tinklo klasifikavimo metodas.....	32
5.2.2. Artimiausio kaimyno klasifikavimo metodas .....	37
5.2.3. Atraminio vektoriaus mašinos klasifikavimo metodas.....	40
5.2.4. Sprendimo medžio klasifikavimo metodas.....	43
5.2.5. Klasifikavimo rezultatų apibendrinimas .....	45
5.3. Klasifikavimo metodų mokymas su papildomais požymių rinkiniais.....	46
5.4. Klasifikavimo metodų tikslumo įvertinimas su mokyme nedalyvaujančiais požymiais .....	50

REZULTATAI IR IŠVADOS .....	55
INFORMACIJOS ŠALTINIŲ SĄRAŠAS.....	56
PRIEDAI .....	59
Priedas 1. „Matlab“ programinis kodas neuroninio tinklo klasifikavimo mokymui, testavimui.....	59
Priedas 2. „Matlab“ programinis kodas k-NN, SVM, „Bagged tree“ klasifikavimo mokymui, testavimui .....	63
Priedas 3. „Matlab“ funkcijos k-NN, SVM, „Bagged tree“ klasifikavimo mokymui ...	65
Priedas 4. „Matlab“ realizuotų k-NN, SVM, „Bagged tree“ klasifikatorių modeliai ....	67

## SANTRUMPŲ IR ŽENKLŲ AIŠKINIMO ŽODYNAS

ANN	Dirbtinis neuroninis tinklas ( <i>angl. Artificial Neural Network</i> )
k-NN	Artimiausio kaimyno metodas ( <i>angl. k – Nearest Neighbor</i> )
SVM	Atraminio vektoriaus mašinos ( <i>angl. Support Vector Machines</i> )
TPR	Tiksliūs teigiami rezultatai ( <i>angl. True Positive Rates</i> )
FNR	Netikslūs neigiami rezultatai ( <i>angl. False Negative Rates</i> )
PPV	Teigiamos prognozuojamos reikšmės ( <i>angl. Positive Predictive Values</i> )
FDR	Netikslūs atpažinti rezultatai ( <i>angl. False Discovery Rates</i> )
FPR	Netikslūs teigiami rezultatai ( <i>angl. False Positive Rates</i> )
CM	Klaidingumo matrica ( <i>angl. Confusion Matrix</i> )
ROC	Sprendimų priėmimo charakteristika ( <i>angl. Receiver Operating Characteristic</i> )
NPRtool	Neuroninio tinklo modelių atpažinimo įrankis ( <i>angl. Neural Pattern Recognition tool</i> )
RBF	Radialinių bazių funkcija ( <i>angl. Radial Basis Function</i> )



## ĮVADAS

Gamyboje reikalingas patikimas įrenginių darbas su kuo mažesniais nuostoliais bei be nenumatytų gamybos proceso sustojimų, taigi būtina tinkama priežiūra, kad būtų to išvengta. Nuo tinkamo priežiūros planavimo priklauso įrenginių nusidėvėjimas vykstant gamybiniam procesams bei pastarųjų stabilumas, efektyvumas, nenutrūkstamumas (nepertraukiamumas). Tai ypač svarbu vykdant linijinės gamybos procesus, kuriuose nors vieno segmento gedimas daro didelę įtaką visam procesui.

Gamybos įrenginių nusidėvėjimo įvertinimas yra sudėtingas procesas, kuris labiausiai priklauso nuo įrenginių sudarančių segmentų kiekio bei sudėtingumo. Nusidėvėjimo, gedimo tikimybės įvertinimui pasitelkiami įvairūs metodai – vienas iš jų – dirbtinis intelektas, kuris nagrinėjamas šiame projekte. Dirbtinio intelekto prognozavimo rezultatas (gaunama aibė) yra paremtas turimais istorijos duomenimis. Pagal procesus vykusius praeityje darome prielaidą, kad ateities procesai pasikartos arba bus paremti istorijos duomenimis. Realybėje tobulas proceso prognozavimas yra neįmanomas, tačiau galima kuo tiksliau optimizuoti (sumažinti) netikslias prognozes.

Pagrindinė šiame darbe nagrinėjama problema: kaip nustatyti įrenginio gedimą pagal įrenginio efektyvumo rodiklius, darbo režimą, kitą informaciją ir sumažinti nenumatytų sustojimų skaičių bei laiką dėl techninių gedimų?

Darbo tikslas – atlikti įrenginių gedimų prognozės tyrimą, nustatyti pasirinktų prognozavimo metodų galimybes bei įvertinti atliktų prognozavimų tikslumą.

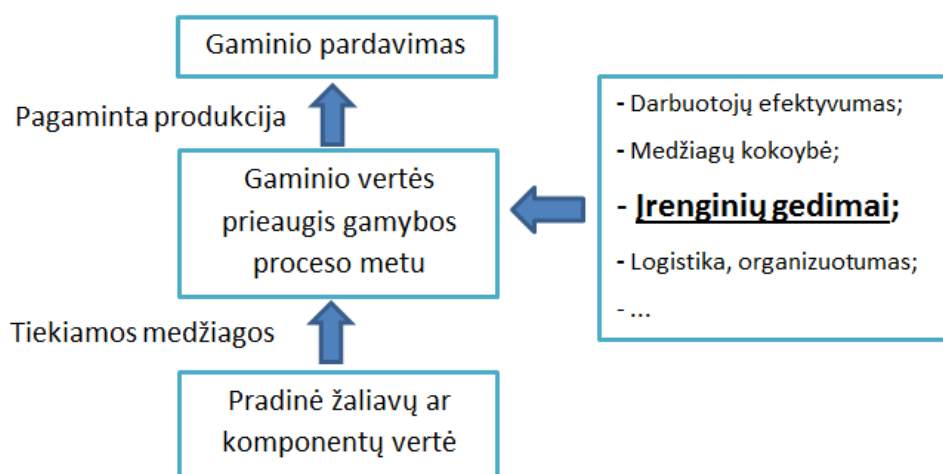
Tyrimui atlikti naudojami duomenys iš AB „Volfas Engelman“ įrenginių gedimų ir priežiūros duomenų bazės „Arrow Novi“ bei monitoringo sistemos „Arrow Machine Track“, kuriais bus realizuoti naudojami metodai įrenginių gedimo prognozavimui.

Iškeliami magistro projekto uždaviniai:

1. Surinkti tyrimo duomenis ir sudaryti gedimų klasių rinkinius;
2. Parinkti klasifikavimo metodus įrenginių gedimų prognozavimo tyrimui;
3. Parinkti klasifikavimo metodų parametrus pagal mokymo duomenis;
4. Ištirti ir palyginti realizuotų klasifikavimo metodų prognozavimo tikslumą;
5. Nustatyti sudarytų klasių atpažinimo galimybes naudojant pasirinktus požymius ir klasifikavimo metodus.

## 1. ĮRENGINIŲ GEDIMŲ SVARBA VYKDANT GAMYBOS PROCESĄ

Gaminio vertės prieaugis gamybos proceso metu priklauso nuo daug veiksnių: darbuotojų efektyvumas, medžiagų kokybė, organizuotumas, įrenginių gedimai ir kt. Supaprastinta gaminio vertės schema pavaizduota 1.1 paveiksle. Siekiant užtikrinti kuo mažesį gaminio vertės prieaugį dėl įrenginių gedimų gamybos proceso metu, reikalingas tinkamas techninės būklės įvertinimas ir atnaujinimas tam tikru periodiškumu arba remiantis nustatytais požymiais, matavimais ir kt [1].



1.1 pav. Gaminio vertės didėjimas gamybos proceso metu

Pats primityviausias įrenginių priežiūros būdas – atliekami remonto darbai, kai įrenginių būklė tampa kritinė ir pastebimai netinkama atlikti tinkamo gamybos proceso. Tai pats neefektyviausias būdas, kadangi nežinoma įrenginio darbo laiko prognozė, nežinomas remonto laikas, apimtis bei galima gamybos prastova.

Remonto organizavimas pagal įrenginių techninę būklę, taip pat nėra labai efektyvus, kadangi susidaro prastovos, kurios yra netolygios. Šis būdas ypač netinkamas gamyboje, kurioje turi būti užtikrintas nepertraukiamas gamybos procesas bei nestabdomas kitų įrenginių darbas. Taip pat remonto darbų apimtis ir periodiškumas yra labai sunkiai prognozuojamas.

Kitas įrenginių remonto organizavimas – planinis. Planuojami periodiniai įrenginių remontai pagal įrenginių veikimo laiką. Toks techninės priežiūros būdas yra gana efektyvus, kadangi išvengiama nenumatyto įrenginio ar jo dalies susidėvėjimo [2]. Pats didžiausias trūkumas – neišvengiama nepagrįstai didelių remontų, kurie reikalauja didelių kaštų.

Pats efektyviausias ir moderniausias – prognozuoti įrenginių nusidėvėjimą pasitelkiant patikimumo skaičiavimą, istorinę statistiką, įtraukiant dirbtinį intelektą stebint vykstantį įrenginio darbo procesą, efektyvumą ir kitus rodiklius [2].

## 2. DIRBTINIS INTELEKTAS

Dirbtinis intelektas – tai techninių, programinių modelių visuma. Intelektu modeliai yra sudaryti iš techninių programinių arba matematinių metodų. Dirbtinis intelektas egzistuoja tik kartu su natūraliu intelektu bei yra papildomas. Galima teigti, kad dirbtinis intelektas neegzistuoja, kadangi net ir galingiausios sistemos savęs „nesuvokia“ – reikalingas operatorius arba sistemos mokymas ir nauji duomenys.

Pagrindinės dirbtinio intelekto savybės:[3]

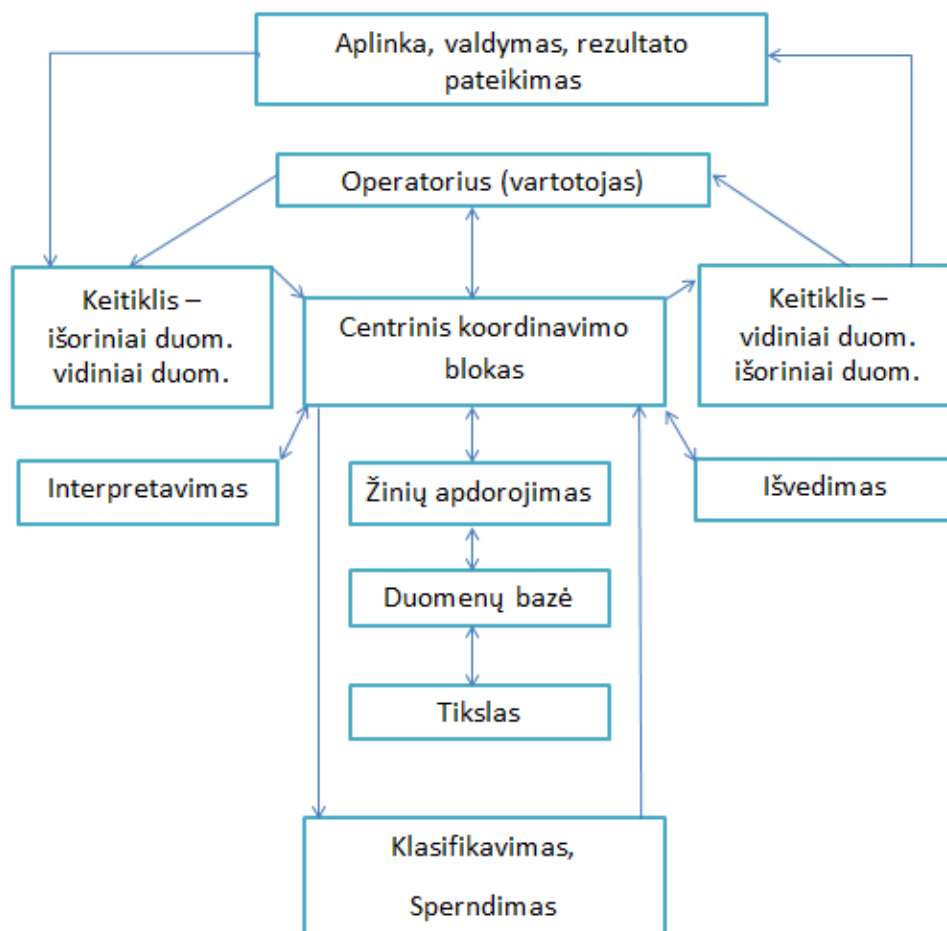
- Natūralių, formalių savokų operavimas – tai įvairios procedūros pagal dinamiškas duomenų bazines, kurios yra atnaujinamos;
- Sistemos veikimas kai sąlygos yra neapibrėžtos bei tikslumas sprendžiant sudėtingas matematinės logikos taisykles ir kitas sudėtingas sistemas;
- Kaupiama patirtis – faktų registravimas, statistinių duomenų saugojimas, kitaip vadinama proceso „atspindėjimu“.

Patirties kaupimo savybė dirbtiniame intelekto yra labai svarbi, kai sistema veikia pagal visas tris paminėtas savybes. Neapibrėžtumo savybė susideda iš:[3]

- Neaiškaus sistemos funkcionavimo tikslo;
- Neaiškus sprendimo pasirinkimo metodo;
- Klaidos (žmogiškasis faktorius);
- Dinamiškumo;
- Neaiškių kriterijų, matavimų, matavimų ribų;
- Nepilnai, neaiškiai aprašyto modelio;
- Skirtingo sistemų interpretavimo „suvokimo“.

Šiame darbe pagrindinis dėmesys skiriamas klasifikavimo metodų analizei, klasifikavimo modelių realizavimui, mokymui bei jų tikslumo įvertinimui. Skiriami penki dirbtinio intelekto modeliai:[1]

- Žinių bazių modeliai;
- Neroniniai tinklų modeliai;
- Klasifikavimo metodai;
- Neraiškios aibės ir sistemos (fuzzy);
- Genetiniai algoritmai.



2.1 pav. Dirbtinio intelekto struktūra

Dirbtinio intelekto veikimą galima atvaizduoti funkciniais blokais, sujungtais į algoritmą (2.1 pav.). Pagrindiniai dirbtinio intelekto funkciniai blokai:[3]

- Žinių apdorojimas, duomenų bazė, tikslas – duomenų kaupimas, „mokymasis“;
- Centrinis blokas – koordinuoja kitų funkcininių blokų darbą;
- Klasifikavimas, sprendimas – pagal priskirtą metodą randamas tinkamiausias sprendimas.

Visos dirbtinio intelekto sukurtos taisyklės bei veikimas remiasi tik sukauptais duomenimis, didžiausias DI trūkumas – duomenų nustatymas, kurias remsis sistema negalimas pačios sistemos (reikalingas žmogaus ar išorinės įrangos suformuotas teisingas atsakymas).

Taigi neapibrėžtumas, kokio tipo, laikotarpio, formos, išraiškos, klaidos duomenys bus naudojami daro DI priklausomą nuo natūralaus žmogaus intelekto. Mokslinėje literatūroje teigiama, jog dirbtinis intelektas gali būti artimas tik gyvosios būtybės instinktamas ar suvokimui, bet ne intelektui, bet koku atveju DI sprendimas yra apspręstas žmogaus ir jo sukurtų įrankių pagalba [4].

### 3. KLASIFIKAVIMAS

Klasifikavimas – tai proceso ar objekto priskirimas tam tikrai iš anksto nustatytai klasei, pagal tam tikrus požymius arba kitaip –  $x$  vektoriaus priskyrimas vienai iš klasių. Klasės nustatomos naudojantis tam tikromis klasifikavimo taisyklėmis – mokymo imtimis. Klasifikavimas priskiriamas mokymui su mokytoju. Dažniausiai klasifikavimo analizė naudojama prognozavimui, objektų, procesų atpažinimui pagal turimus jų požymius.

Klasifikatoriai – tai programinės sistemos, kurios skirtos procesams arba objektams atpažinti – konkreti savybė ar veiksmas, būseną, priskiriama objekto procesui ar situacijai. Klasifikavimo režimai skirstomi:[5]

- Infomacijos kaupimas – formuojama žinių bazė, parenkami požymių kintamieji;
- Atpažinimas – klasifikavimo taisyklių formavimas, proceso ar objekto priskirimas klasei;
- Kompleksinis veikimas – klasifikavimo kokybės įvertinimas, atnaujinama žinių bazė bei procesų, objektų ir kt., atpažinimas pagal klasifikavimo kokybę.

Mokslinėje literatūroje nurodoma labai daug klasifikavimo metodų (apie 200) iš įvairių klasifikavimo „šeimų“: diskriminantinė analizė, Bajeso klasifikatorius, atraminio vektoriaus mašinos, sprendimo medžiai, atsitiktiniai medžiai, neuroninių tinklų klasifikavimas, artimiausio kaimyno klasifikatorius, taisyklėmis pagrįsti klasifikatoriai, apibendrinti tiesiniai modeliai ir kt. Taip pat klasifikatoriai pagrįsti skirtingoms mokslo šakoms gali būti skirstomi į: statistinius, geometrinius, lingvistinius. Taigi įvairūs klasifikavimo metodai gali būti skirstomi pagal naudojamus skaičiavimo algoritmus, funkcijas, mokslo stitį, greitaveiką, ir pan[9]. Renkantis klasifikavimo metodą reikia atsižvelgti į minėtas charakteristikas, bei įvertinti turimus duomenų masyvus.

#### 3.1. Klasifikavimo metodų parinkimas ir taikymas

Klasifikatorių tikslumui įvertinti neužtenka realizavimo su vienu duomenų masyvu ar panašių charakteristikų duomenų masyvu. Deivido Volperto teorema teigia: „Geriausias klasifikatorius nebus toks pats geras visiems duomenų rinkiniams“. Remiantis Deivido Volperto teorema galima teigti, kad daryti prielaidą, jog kažkuris metodas yra daug pranašesnis už kitą yra neteisinga, kadangi pasirinktas metodas gali sąlygoti tenkinamus rezultatus analizuojant tam tikros srities duomenis, tačiau būti visiškai netinkamas analizuojant kitą sritį. Taip pat į klasifikatorių tikslumo vertinimą reikia įtraukti gaunamus rezultatus su praplėtais duomenų masyvais, kadangi tai gali padidinti arba sumažinti gaunamo klasifikavimo tikslumą.[6]

Joaquin Vanschoren, Hendrik Blockeel, Bernhard Pfahringer, and Georey Holmes straipsnyje „A new way to share, organize and learn from experiments. Machine Learning.“ atliktame tyrime keičiant duomenų masyvo dydį bei testuojant klasifikatorius su įvairiais klasių požymiais daroma išvada, kad yra neįmanoma nustatyti pasiekiamą maksimalų tikslumą pasirinktam duomenų rinkiniui, kaip ir sunku įvertinti tikrąjį kiekvieno klasifikatoriaus tikslumą.[7]

Priešingai nei daromos išvados minėtame straipsnyje 2014 metais atliktas Manuel Fernandez-Delgado, Eva Cernadas, Senen Barro, Dinani Amorim autorių tyrimas. Tyrime ištirti 179 klasifikavimo metodai iš 17 skirtingų klasifikatorių „šeimų“ – diskriminantinė analizė, Bajeso klasifikatorius, neuroniniai tinklai, sprendimo medžiai, taisyklių klasifikatoriai, atsitinkinio medžio klasifikatoriai, artimiausio vektoriaus metodai, kurie realizuoti „R“, „Matlab“, „Weka“ programine įranga. Atliktas tyrimas naudojant įvairius duomenų rinkinius iš sistemų mokymo duomenų bazės „UCI Machine learning“ [8]. Ištyrus 121 duomenų rinkinį, pastebėta, kad geriausią rezultatą pasiekė atsitiktinio medžio sprendimo klasifikatorius (*angl. Random trees classifier*) – 94,1 proc. maksimalų tikslumą ir iš 84,3 proc. visų naudotų duomenų rinkinių viršijo 90 proc. tikslumą. Antroje vietoje – Atraminio vektoriaus mašinos klasifikatorius naudojant Gauso bronduolio funkciją, kuris pasiekė 92.3 proc. tikslumą. Toliau seka neuroninio tinklo klasifikatorius (daugiasluoksnis neuroninis tinklas).[9] Taigi daroma išvada, kad klasifikavimo metodo parinkimas priklauso nuo naudojamo duomenų rinkinio. Tačiau pagal atliktą itin plačią analizę ir klasifikatorių tikslumo lyginimą, galimas klasifikatoriaus su geriausia tikimybe gauti aukščiausią tikslumą nustatymas.

Atliekant tyrimą su vieno pobūdžio duomenų rinkiniu tiksliausias klasifikatorius gali visiškai skirtis nuo gautų rezultatų su įvairiais duomenų rinkiniais. Taip pat gaunamas klasifikavimo tikslumas labai priklauso nuo turimo duomenų rinkinio padalinimo mokymui, validavimui bei testavimui. Naudojant įvairių duomenų rinkinių paskirstymą – gausime labai skirtingus tikslumo rezultatus. Labai svarbu tinkamai padalinti mokymo bei testavimo procesui duomenis – tikslinga mokymo imtyje naudoti didesnę klasių imčių skaičių nei testavimo imtyje. Hyonho Chun and Sunduz Keles atliktame tyrime naudojamas skirtingas duomenų padalinimas – daroma išvada, kad neįmanoma lyginti eksperimentų, kai naudojamos duomenų imtys yra nevienodai padalinamos skirtingiems klasifikavimo metodams. Taigi lyginant skirtingus klasifikavimo metodus būtina naudoti tuos pačius duomenų rinkinius bei tokį patį arba labai artimą duomenų dalinimą.[10][11]

Klasifikavimo metodo rezultatų testavimo kriterijus yra labai svarbus norint kuo tiksliau įvertinti klaidingo klasifikavimo tikimybę, sikirtingiems klasifikavimo metodams. Didesnės imties klasių rinkiniams testavimo kriterijaus parinkimas tikslumui turi mažesnę įtaką nei

mažesnės imties (nuo 6000 objektų imčių). Taigi lyginant skirtingus klasifikatorius tikslinga naudoti vienodą klasifikavimo tikimybės vertinimo metodą. [12]

### 3.2. Objektų klasifikavimo tikslumo įverinimas

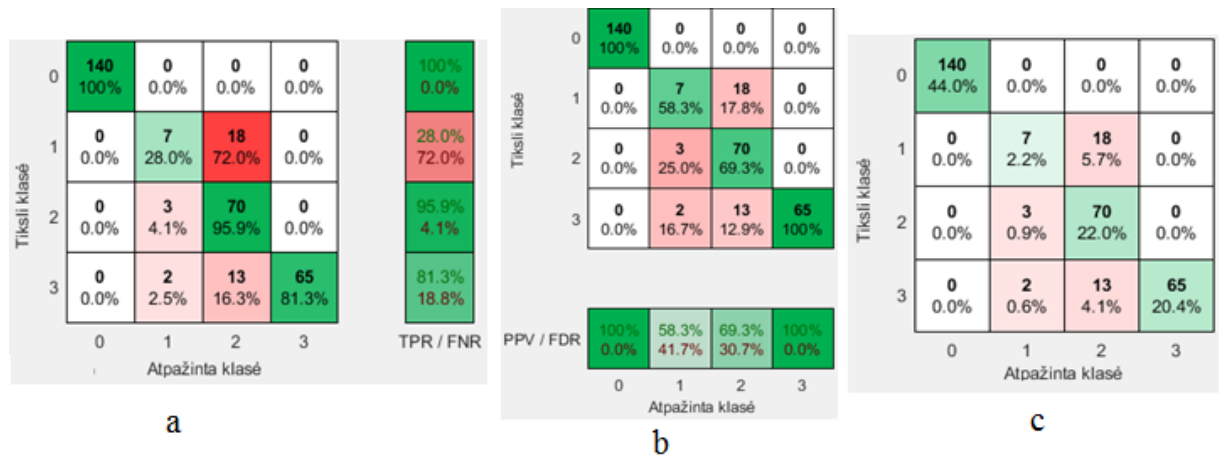
Kiekvienos klasės ir bendro klasifikatoriaus klasifikavimo tikslumui įvertinti dažniausiai naudojami klaidingo klasifikavimo įverčiai – tikslumo matricos (*angl. Confusion Matrix*) taip pat sprendimų priėmimo charakteristikos kreivės ROC (*angl. Receiver Operating Characteristic*).

Klasifikavimo tikslumo matricos susideda iš tikslų klasių bei atpažintų klasių duomenų (matricoje pateikiama horizontaliai arba vertikalčiai). Tikslumo matricomis atvaizduojamas realizuoto klasifikatoriaus tikslumas kiekvienos klasės atžvilgiu skaitinė ir (arba) procentine reikšme – tikimybe. Skiriamos trys klasifikavimo matricų grupės[13]:

- Sumuojamas tikslų klasių rezultatas įtraukiant tikslus teigiamus rezultatus, kai objektai priskiriami tam pačiam klasteriui ir klasei – TPR (*angl. True Positive Rates*), bei netikslus neigiamus rezultatus, kai objektai priskiriami tai pačiai klasei, bet skirtingiems klasteriams. FNR (*angl. False Negative Rates*) (3.1 pav. a).
- Sumuojamas atpažintų klasių rezultatas įtraukiant teigiamas atpažintas reikšmes, kai objektai priklauso tam pačiam klasteriui, bet skirtingoms klasėms – PPV (*angl. Positive Predictive Values*) bei netikslus atpažintus rezultatus, kai objektai priklauso skirtingoms klasėms ir klasterims – FDR (*angl. False Discovery Rates*) (3.1 pav. b).
- Bendri tikslumo matricos rezultatai – pateikiamas kiekvienos klasės tikslūs bei netikslūs atpažinimo rezultatai bei kokią dalį šie rezultatai sudaro iš visų duomenų (3.1 pav. c).

Bendros tikslumo matricos rezultatas – bendras tikslumas  $R$  (Teisingų rezultatų procentinė išraiška)

$$R = \frac{TPR + FNR}{TPR + PPV + FNR + FDR} \quad (3.1)$$



3.1 pav. Klaidingo klasifikavimo įverčių matricių pavyzdžiai

Klaidingo klasifikavimo tikimybės vertinimas išreikštas formulėmis:

$$a_i = \frac{b_i}{n_i} \quad (3.2)$$

$$a = \frac{\sum_i b_i}{m} \quad (3.3)$$

$a_i$  – klasės klaidingo klasifikavimo įvertis;

$b_i$  – klasės klaidingai klasifikuoti objektai;

$n_i$  – klasės objektų imties skaičius;

$m$  – objektų testavimo imties skaičius.

Klasifikatorių klaidingo klasifikavimo vertinimas gali būti atliekamas skirtingais metodais požymių duomenų imties atžvilgiu. Naudojami metodai, kurių pagrindinis skirtumas – visos duomenų imties padalinimas į mokymo ir testavimo imtis. Skiriami pagrindiniai metodai [11]:

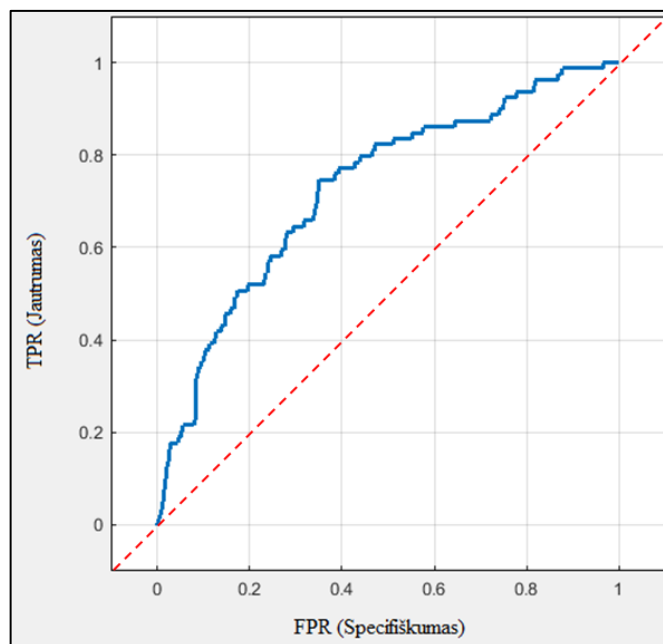
- Nustatytos tiesinės imties arba nustatyto dydžio imtys kiekvienai klasei.
- Kryžminis validavimo (angl. *cross validation*) – atliekat klasifikavimo mokymą, vienas iš suklasifikuotų objektų priskiriamas atliktai testavimo imčiai, o likusieji mokymo imčiai. Toks skirstymas atliekamas kol visi objektai priskiriami klasėms. Naudojamos 3.1 ir 3.2 išraiškos.
- Monte Karlo kryžminis validavimo (angl. *Monte Carl cross validation*) – mokymų ir testavimų imtys padalinamos atstiktine tvarka ir atliekamas klasifikavimas. Klaidingo klasifikavimo tikimybė gaunama pagal 3.2 formulę, nustatytą  $n$  kartų. Randamas klaidingo klasifikavimo tikimybės vidurkis (analogiškai skaičiuojama klasėms) – Monte Karlo validavimo klaidingo klasifikavimo tikimybės įvertis.



- Pakartotinių imčių – iš mokymo imties sudaroma imtis su  $n$  kartų pasikartojančia imtimi tokio paties dydžio kaip ir mokymo imtis. Likę objektai iš mokymo imties priskiriami testavimo imčiais. Kartojant šį paskirstymą sudaroma tikslumo matrica pagal gautos klaidingo klasifikavimo tikimybės įverčių imties vidurkį, naudojant 3.2 išraišką.

Klasifikavimo tikslumo verinimui pasirenkamas nustatyto dydžio ir kryžminis validavimo metodas, kadangi yra tinkamesnis mažesnių duomenų rinkinių analizei (iki 6000 požymių įrašų [14]). Siekiant gauti kuo tikslesnį klasifikatorių palyginimą, kitais validavimo metodais tikslumas nėra vertinamas.

Sprendimų priėmimo charakteristikos kreivės ROC parodo klasifikatoriaus jautrumo ir specifiškumo priklausomybę, kitaip – tyrimo specifiškumo su vieneto skirtumo sąryšį [15]. Jautrumas parodo tiksliai klasifikuotos pirmos klasės ir visų pirmos klasės objektų santykį, specifiškumas tiksliai klasifikuotų antros klasės ir visų antros klasės objektų santykį. ROC kreivės horizontalioje ašyje (FPR) rezultatų santykis, vertikalioje ašyje – (TPR) rezultatų santykis, taigi visą ROC kreivę sudaro tik klasifikatoriaus prognozuojami teigiami dviejų klasių rezultatai (3.2 pav.)[16].



3.2 pav. ROC kreivės pavyzdys

Klasifikatoriaus prognozavimo savybės yra geresnės tuo atveju, jeigu plotas po ROC kreive yra artimesnis vienetai. Priimtina laikyti, jog tyrimo prognozavimo savybės yra visiškai nepatikimos arba atsitiktinės, jeigu plotas po ROC kreive yra mažesnis negu pusė viso sprendimų priėmimo charakteristikos kreivės ploto, 3.2 paveiksle šis plotas pažymėtas punktyrine linija [16]. Jeigu dviejų ar daugiau klasifikatorių, bendros klaidingo klasifikavimo

tikimybės įverčiai yra vienodi arba labai artimi – tikslesnis klasifikatorius, kurio jautumas yra didesnis. Klasifikuojant dvi klases – klasių ROC kreivių plotas yra lygus, taigi ROC kreivės atskirai nėra vertinamos.

## 4. KLASIFIKAVIMO METODŲ ANALIZĖ

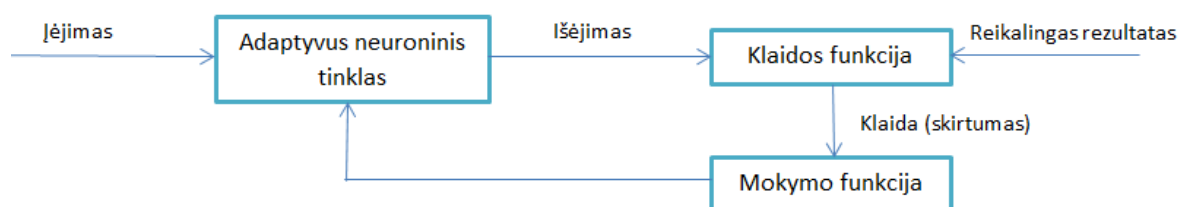
### 4.1. Neuroninio tinklo klasifikatorius

Dirbtinis neuroninis tinklas ANN (*angl. Artificial Neural Network*) – tai sistema, kuria yra apdorojama informacija, sukuriant biologinės nervų sistemos analogą. Tinkle dominuojantys neuronai – tai komponentai, kurie sudaromi iš daug tarpusavyje susijusių skaičiavimų.[17]

Pagal esamus pavyzdžius neuroninis tinklas yra mokomas – pavyzdžiai parenkami kuo tiksliau, kadangi tai apsprendžia kaip neuroninis tinklas mokysis ir tinkamai funkcionuos. Dirbtiniai neuroniniai tinklai yra sudaryti iš daug neuronų, kuriuos galima vadinti skaičiuojamaisiais elementais. Neuronai vienas su kitu yra sujungti nustatyto stiprumo jungtimis – dar vadinamomis sinapsių jungtimis. Analogiškai, kaip biologinių neuronų sinapsės yra keičiamos smegenyse mokantis, taip dirbtiniame neuroniniame tinkle tarp neuronų esančios jungtys yra keičiamos (jungčių stiprumo koeficientai arba svoriai) – vykdomas tinklo mokymas.

Neuroniniai tinklai naudojami duomenų klasifikavimui, įvairių sistemų modeliavimui, signalų apdorojimui, identifikavimui, procesų prognozavimui[17]. Jau dabar neuroniniais tinklais pagrįstas daugelio produktų veikimas: vaizdų apdorojimo sistemos, kalbos atpažinimo sistemos, medicinos įranga ir daugelis kt.

Neuroniniai tinklai yra adaptyvūs, kadangi sistema išorinius duomenis naudoja vidinių parametrų nustatymų ir keitimui. Taip vyksta tinklo mokymas – žinomos įėjimo ir išėjimo reikšmės. Neuroniniame tinkle naudojama klaidos funkcija, kuri nusako skirtumą tarp išėjimo ir užduoties reikšmės. Klaidos reikšmė yra mažinama keičiant svorių reikšmes [18]. Adaptyvaus neuroninio tinklo blokinė schema pavaizduota 4.1 pav.

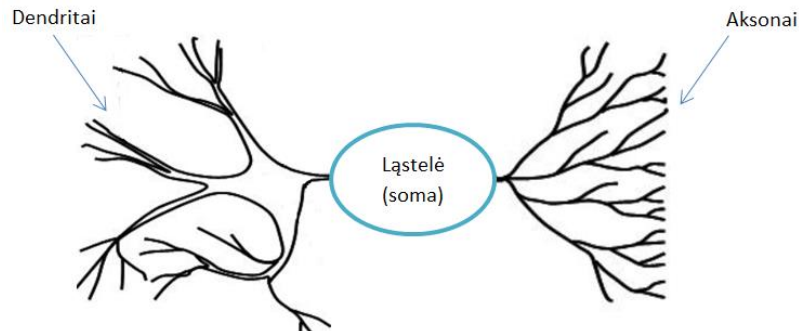


4.1 pav. Adaptyvaus dirbtinio neuroninio tinklo blokinė schema

### 4.1.1. Neuroninio tinklo modelis

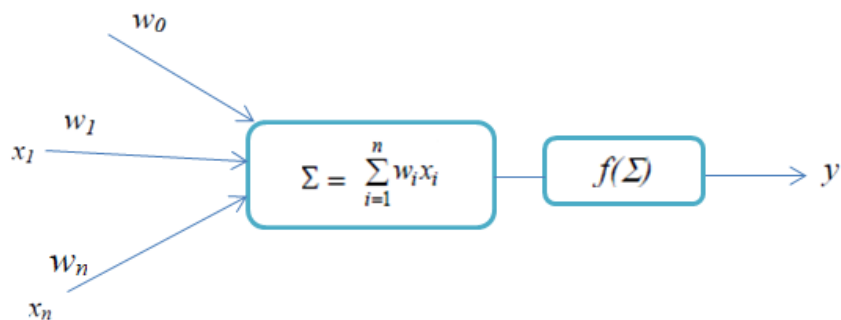
Neuroninio tinklo modeliavimą pradėdame nuo neurono modelio. Dirbtinio neuroninio tinklo neuronas sudarytas išlaikant biologinio neurono sistemą (4.2 pav.)

Biologiniame neurone įėjimai – dendritai kurie jungiasi su ląstele (soma) ir išėjimo aksonais (išėjimais). Toliau aksonai sinapsėmis jungiami su kitu biologiniu neuronu.



4.2 pav. Biologinis neuronas

Dirbtiniame neurone – įėjimo reikšmių jungtys turi atitinkamus koeficientus, kurie apsprendžia svorius (atitinka sinapsių funkciją).  $X_1 \dots X_n$  – įėjimo reikšmės,  $W_1 \dots W_n$  – atitinkamų svorių koef.,  $F()$  – perdavimo funkcija,  $Y$  – dirbtinio neurono išėjimas (4.3 pav).



4.3 pav. Dirbtinis neuronas

Kiekvienas neuronas turi perdavimo funkciją ir slenkščio reikšmę. Neurono sužadinimo reikšmė gaunama įėjimo svorinę reikšmę atėmus iš slenkščio reikšmės. Pagal neurono perdavimo funkciją ir slenkščio reikšmę gaunamas neurono išėjimas[1].

Neurono žymėjimas:  $i$  – neurono indeksas tinkle,  $w_{ij}$  – neurono  $j$  įėjimos svorinė vertė (nustatoma tinklo mokymo etape). Taigi įėjimos funkcijos reikšmė:

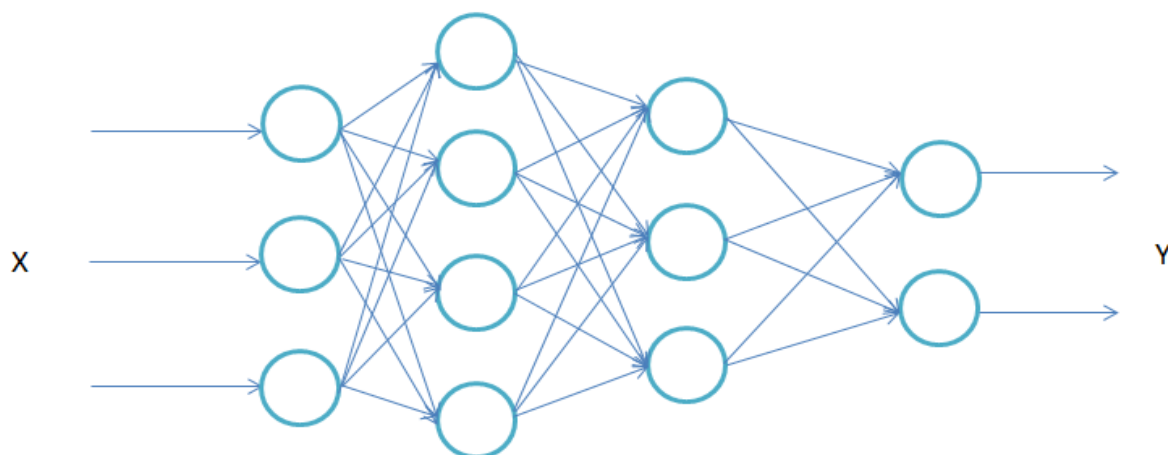
$$in_i = \sum_{j=1}^n w_{ij} x_{ij} \tag{4.1}$$

Aktyvavimo funkcijos reikšmė  $a_i(t + \Delta t)$  priklauso nuo įėjimo funkcijos reikšmės bei slenkščio  $Q_i$ . Aktyvavimo – išėjimo funkcijos išreiškiamos:

$$a_i(in_i) = \frac{1}{1 + e^{-inp+Q}} \quad (4.2)$$

Visi neuronai yra jungiami vienas su kitu taip sudarant tinklą. Paprasčiausios dinamikos yra tiesioginio sklidimo neuroninio tinklo struktūra dažniausiai naudojama spręsti realią problemą ar objektų lyginimą. Skiriami tiesioginio sklidimo neuroniniai tinklai:[19]

- Vienasluoksniai neuroniniai tinklai – tinklai kurie nėra grupuoti į atskirus sluoksnius;
- Daugiasluoksniai neuroniniai tinklai (4.4 pav.) – paeiliui skaičiuojamos kiekvieno sluoksnio neuroų išėjimo reikšmės bei svorinė prieš tai buvusio neurono reikšmė, kuri atimama iš slenkščio reikšmės ir gaunams sužadinimas pagal kurį keičiama pedavimo funkcija. Perdavimo funkcija skaičiuoja išėjimo reikšmę.



4.4 pav. Daugiasluoksnis tiesioginio sklidimo neuroninis tinklas

#### 4.1.2. Neuroninio tinklo mokymas

Neuroninių tinklų viena pagrindinių savybių – nėra žinoma priklausomybė tarp įėjimo ir išėjimo reikšmių. Neuroninio tinklo mokymo tikslas yra išgauti priklausomybę tarp išėjimo ir įėjimo reikšmių. Jeigu tokia priklausomybė neegzistuoja (užduotos reikšmės kinta pagal atsitiktinumą, arba įėjimo duomenys nesuteikia informacijos apie išėjimo reikšmes) neuroninis tinklas negali gauti teisingų rezultatų.

Skiriami du neuroninio tinklo mokymo tipai:

- Su mokytoju – pateikiamos įėjimo reikšmės kartu su išėjimo reikšmėmis (užduotimis);
- Be mokytojo – remiamasi užduotais vidiniais tinklo kriterijais, nekeičiamos svorių reikšmės.

Dažniau naudojamas neuroninis tinklas su mokytoju, kadangi taip keičiama priklausomybė tarp įėjimo ir išėjimo reikšmių. Mokymo metu gaunama reikšmė, kuri įvertina neuronio tinklo tikslumą – skirtumas tarp tinklo išėjimo ir užduotos (teisingos) reikšmės. Pagal gautą tikslumo reikšmę keičiamos svorių ir slenkščių reikšmės, kad tinklas turėtų mažesnę klaidą arba tikslenį kriterijų.[18]

Mokymo tikslumas apibūdinamas sumine kvadratine paklaida  $e_s$  (4.3) – vidutinis skirtumas tarp gautų dirbtinio neuroninio tinklo rezultatų ir tikrųjų reikšmių.

$$e_s = \sqrt{\sum_{i=1}^k (y_i - y_{ai})^2} \quad (4.3)$$

$e_s$  – suminė kvadratinė paklaida;

$y_i$  –  $i$ -ojo pavyzdžio išėjimo duomenys;

$y_{ai}$  –  $i$ -ojo pavyzdžio mokomo neuroninio tinklo išėjimo duomenys gauti iš duomenų  $x_i$ ;

$k$  – duomenų pavyzdžių skaičius.

Atliekant mokymo procesą svorių reikšmės keičiamos taip, kad suminės kvadratinės paklaidos reikšmė mažėtų. Realizuotas ANN testuojamas pagal sudarytą atskirą duomenų paketą. Jeigu testavimo suminė kvadratinė paklaida apytiksliai lygi mokymo suminei kvadratinei paklaidai (naudojamas vienodas testavimo ir mokymo duomenų dydis) – galime teigti, kad tinklas išmokytas teisingai [19].

Vienas iš neuroninio tinklo mokymo uždavinių – nustatyti racionalų paslėptojo sluoksnio neuronų skaičių. Naudojant per mažą neuronų skaičių padidėja galimybė atsirasti dideliai modeliavimo klaidai, naudojant per didelį neuronų skaičių – mažinamas tinklo generalizavimas. Tinklo parametrų skaičiaus nustatymui naudojama Akaike išraiška, kuria įtraukiama tinklo parametrų skaičius į tinklo paklaidos išraišką:[18]

$$E_g = E \cdot \left( \frac{1 + \frac{Q}{N}}{1 - \frac{Q}{N}} \right) \quad (4.4)$$

- E – tinklo paklaida;
- Q – modelio parametrų skaičius;
- N – eksperimento taškų skaičius.

#### 4.2. Artimiausio kaimyno klasifikavimo metodas

Artimiausio kaimyno klasifikavimo metodas k-NN (*angl. k – Nearest Neighbor*) priskiriamas prie vektorinių klasifikavimo metodų – galimybė atvaizduoti duomenų požymių aibes vektorinėje erdvėje. Klasifikuojamam objektui atliekamas artimiausio kaimyno algoritmas ir pagal klasės raktinius požymius (mokymo duomenis) priskiriama iš anksto apibrėžtomis kategorijoms. Kasifikatoriaus k-NN pagrindinis tikslas – klases apibūdinančių požymių skirtumo įvertinimas (mato nustatymas). Kuo didesnė skirtumo reikšmė, tuo klasės labiau skirtingos.[20]

Pagrindinės k-NN klasifikatoriaus mokymo algoritmo dalys:

- Apskaičiuojama klasės  $k$  artimiausių iš mokymo duomenų, pagal metrinį atstumo matą – intervalus arba santykių skalę.
- Nustatoma kuriai kategoriai priklauso daugiausiai  $k$  artimiausių kaimynų;
- Klasei priskiriama nustatyta kategorija.

Dažniausiai naudojami atstumo matavimai [21]:

- Euklido atstumas – trumpiausias atstumas tarp taškų  $x$  ir  $y$  – kitaip trumpiausia tiesė jungianti šiuos taškus.

$$d_{Euklidinis}(x, y) = \sqrt{\sum_{i=1}^n (x_i + y_i)^2}$$

(4.5)

- Euklido atstumo kvadratas – trumpiausio atstumo tarp taškų  $x$  ir  $y$  kvadratas. Dažniausiai naudojamas palyginti atstumams.

$$d_{Euklidinis\ kvadrat.}(x, y) = \sum_{i=1}^n (x_i + y_i)^2$$

(4.6)

- Čebyšovo atstumas – didžiausia absoliuti skirtumo reikšmė tarp stebėjimo porų.

$$d_{\text{Čebyšovo}}(x, y) = \max_i |x_i - y_i| \quad (4.7)$$

- Minkowski atstumas naudojamas  $p$  skirtingų artstumo skaičiavimų įvertinimui.

$$d_{\text{Minkowski } p}(x, y) = \left( \sum_{i=1}^n (x_i + y_i)^p \right)^{\frac{1}{p}} \quad (4.8)$$

- Mahanalobi atstumas – stebimų porų reikšmių absoliučiuųjų skirtumų suma.

$$d_{\text{Mahanalobi}}(x, y) = \sum_{i=1}^n |x_i - y_i|^2 \quad (4.9)$$

Artimiausio kaimyno metodo realizavimą galima išreikšti kaip klasės  $z$  priskyrimą kategorijai  $c$ , kai  $c$  turi aukščiausią panašumo rodiklį iš visų kategorijų (4.11).

$$s(z, c_j) = \sum_{d_i \in k-NN} \text{sim}(z, z_i) y(z_i, c_j) \quad (4.10)$$

$\text{sim}(z, z_i)$  – panašumas tarp tikrosios ir mokomosios klasės (atstumas tarp  $z$  ir  $z_i$ );

$y(z_i, c_j)$  – funkcija, kuri lygi 1 jei klasė  $z_i$  priklauso kategorijai  $c_j$ .

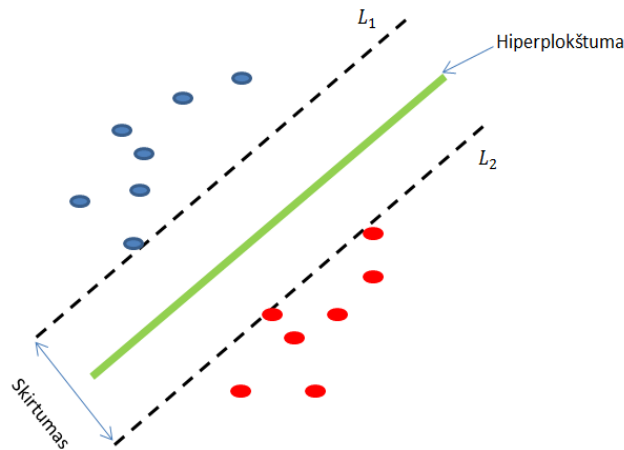
Kategorijos priskyrimas:

$$\text{arg max}_{j=1..m} (s(z, c_j)) \quad (4.11)$$

$c_1 \dots c_m$  – apibrėžtos kategorijos.

### 4.3. Atraminų vektorių mašinos

Atraminio vektoriaus mašinos klasifikavimo metodas SVM (*angl. Support Vector Machines*) dažniausiai naudojamas dviejų klasių atpažinimui. Atraminio vektoriaus metodu randama plokštuma (*angl. Separating hyperplane*), kuri geriausiai atskiria vienos klasės duomenų rinkinius nuo kitos klasės duomenų rinkinių – randamas didžiausias skirtumas (maksimalus atstumas) tarp dviejų klasių (taškų) ir plokštumos. Vietoje sudėtingos kreivės funkcijos naudojama hiperplokštuma ir dvi vienodai nutolusios plokštumos  $L_1, L_2$ , šios plokštumos 4.5 paveiksle pažymėtos žalia ir punktyrine linija. Dvi vienodai nutolusios plokštumos nuo hiperploštumos turi išlaikyti sąlygą, kad tarp jų nėra taškų – taip atskiriami viiškai sirtingi duomenys. Taškai, kurie yra ant plokštumos  $L_1, L_2$  vadinami atraminiais vektoriais.



4.5 pav. Duomenų atskyrimas hiperplokštuma

Atraminio vektoriaus mašinos mokymu gaunama pati optimaliausia plokštuma, kuri atspindi klasifikavimo pasirinkimo ribą tarp dviejų klasių. Tai galima išreikšti kaip reikšmių  $\alpha_i$  ir  $b$  gavimą atlikus 4.12 uždavinį [23].

$$\left\{ \begin{array}{l} \max \left( \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j K(x_i, x_j) \right); \\ 0 \leq \alpha_i \leq C \\ \sum_{i,j=1}^l \alpha_i y_i = 0 \end{array} \right\}$$

(4.12)



Plokštumos funkcija:

$$f(x) = x'\beta + b = 0 \quad (4.13)$$

$(x_i, y_i)$  – artimiausio vektoriaus mašinos mokymo duomenys,  $y_i \in \{1, -1\}$ ;

$b \in \mathbb{R}$  – slenkstis;

$K$  – branduolio funkcija,  $K(x_i, x_j) = x_i \cdot x_j$ ;

$\alpha_i$  – svoriai;

$C$  – neteisingo klasifikavimo kaina

Jeigu duomenys yra neatskiriami, įvedamas netikslumo rodiklis  $C$  – leidimas duomenis atskirti neteisingai. Kuo didesnė reikšmė  $C$ , tuo klasifikavimo tikslumas yra svarbesnis (siauresnė duomenų atskyrimo riba).

Tuo atveju, kai dviejų klasių požymius galima atskirti plokštuma dažniausiai naudojama tiesinė SVM forma, jeigu ne – naudojama netiesinė forma. Kai duomenys neatskiriami tiesiškai, naudojama transformacija, kuri yra išreikiama SVM branduolyje.

Skiriamos pagrindinės SVM branduolio funkcijos[24]:

- Tiesinė:

$$K(x_i, x_j) = (x_i \cdot x_j)^T \quad (4.14)$$

- Polinominė kvadratinė, kubinė:

$$K(x_i, x_j) = (x_i \cdot x_j + C)^2 \quad (4.15)$$

$$K(x_i, x_j) = (x_i \cdot x_j + C)^3 \quad (4.16)$$

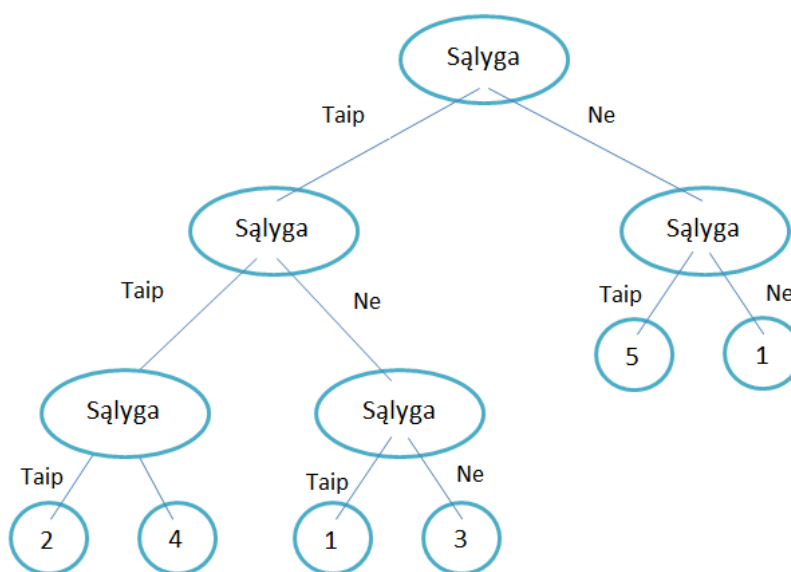
- Radialinių bazių (RBF) Gauso skirstinys:

$$K(x_i, x_j) = e^{-|x_i - x_j|^2 / 2\sigma^2} \quad (4.17)$$

SVM mašinos yra pritaikomos klasifikuoti daugiau negu dvejoms klasėms – atraminio vektoriaus mašinai pritaikomas metodas „vienas prieš vieną klasifikatorius“. Sugeneruojama lokaliųjų dvejų klasių klasifikatorių, kurie vienas nuo kito nepriklauso ir atrenkamas tiksliausias klasifikatorius. Jeigu turime  $N$  klasių, tai reikia realizuoti  $N(N - 1)/2$  dvejų klasifikatorių atraminių vektorių mašinas.[24]

#### 4.4. Sprendimo medžių – hierarchinis klasifikavimas

Sprendimo medžio klasifikavimas – duomenų priskyrimas atitinkamai medžio šakai. Nuo medžio „viršūnės“ pagal naudojamus filtrus, slenkščius duomenys (rekursyviai) keliauja atitinkamomis šakomis kol yra priskiriami vienai iš nustatytų klasių (8.1 pav.). Duomenys, kurie būna išreikšti daugiamačiais vektoriais gali būti kategoriniai arba tolydieji kintamieji. Kiekvieną medžio šaką galime pavadinti klasifikavimo sąlyga ar uždaviniu, visą klasifikavimo medį – dendograma [25].



8.1 pav. Sprendimo medžio schemas pavyzdys

Viena iš pagrindinių sprendimo medžių savybių – skirstant duomenis neprarandama informacija, kadangi bendras medžio duomenų skaičius yra lygus jo mazguose esančių duomenų skaičiui[20]. Viename medžio segmente dažniausiai naudojamas vienas įrašas arba įrašai su vienodomis charakteristikomis.

Mokymosi procese turi būti nustatyta taisyklė tinkama klasifikuoti ne tik mokyme naudojamus duomenis, bet ir naujus tas pačias klases apibūdinančius vektorius (požymius). Kol duomenų aibėje yra daug vektorių, sukuriamas sprendimo taikymo mazgas, kuris yra parenkamas geriausiai atitikusio kriterijaus testu, bei turi ne mažiau kaip dvi baigtis. Remiantis testo baigtimis pradinė aibė padalinama į poaibius, kuriems sukuriami nauji pomedžiai. Vis gilesniame lygmenyje mokymo vektorių poaibis mažėja, kol priskiriami vienai klasei.[26]

Sprendimo medžio daromos klaidos įvertinimas:

$$R_{SK}(T) = \frac{1}{N} \sum_n \frac{|y_n - y(t)|}{y_n} \quad (4.18)$$

$$R_{KK}(T) = \frac{1}{N} \sum_n (y_n - y(t))^2 \quad (4.19)$$

$R_{SK}(T)$  – daroma klaida pagal santykinės klaidos kriterijų;  $R_{KK}(T)$  – daroma klaida pagal vidutinės kvadratinės klaidos kriterijų;  $y(t)$  – atitinkamo mazgo  $t$  prognozuojama reikšmė. Gaunamas klasifikavimo įvertinimas pagal santykinės klaidos kriterijų parodo visų prognozuojamų reikšmių medianą mazge  $t$ , pagal kvadratinės klaidos kriterijų – geriausias prognozės vidurkis.

Sprendimo medžių optimizavimas, dar kitaip vadinamas genėjimu – procesas, kurio metu siekiama pagerinti klasifikavimo tikslumą naujiems duomenims, mažinant medžio mazgų skaičių, sudėtingumą. Genėjimo procesas reikalingas tokiems sprendimo medžiams, kurių klasifikavimo klaida stipriai padidėja naudojant naujus duomenis – sprendimo medis per daug prisitaikęs prie mokymo duomenų. Efektyvus optimizavimo metodas – prieš pridėdant naują medžio mazgą atlikti testą su naujais požymių duomenimis, kuris parodytų ar klasifikavimo tikslumas pagerėjo [25]. Šis metodas vadinamas išankstiniu genėjimu.

Minimalios klaidos sudėtingumo genėjimo algoritmas (4.20) – tai iteracinis procesas, kuriuo mažinamos šakos turinčios mažiausią santykį. Tokiu būdu sudaroma medžių seka su vis mažesniu mazgų skaičiumi (pagal mokymo imtį).

$$\frac{R(t) - R(T_t)}{|T_t| - 1} \quad (4.20)$$

$R(t)$  – mazgo  $t$  daroma klaida;  $R(T_t)$  – medžio šakos nuo mazgo  $t$  klaida;  $T_t$  – šakos lapų skaičius.

#### 4.4.1. Gaubiamasis sprendimo medžio klasifikavimo metodas

Gaubiamasis metodas (angl. *Bagging method*) – veikimas pagrįstas vienodo pobūdžio klasifikatorių konstravimu iš skirtingų pradinės mokymo duomenų aibės poaibių, kurių rezultatai yra sugrupuojami. Gaubiamasis sprendimo medis yra orientuojamas į klaidų mažinimą –

algoritmas klasifikatorių nustato jautrų klaidų kainai, vykdo kainos minimizavimo procedūrą. Mokymo vektorių klasės yra pakeičiamos į tos klasės mažiausios kainos klases ir gautos duomenų aibės naudojamos sprendimo medžio sudarymui. Šis sprendimo medžio klasifikavimas orientuotas į lokaliųjų minimumų problemos sprendimą [24].

Vektorių tikimybės klasės priklausymo įvertinimui iš duomenų aibės suformuojamas tam tikras imčių skaičius, kuris naudojamas sprendimo medžiui sudaryti. Kiekvienas medis turi tikimybę, kad atitinkamas vektorius  $x$  priklauso nustatytai klasei  $y$ , taip pat kiekvienai klasei išvedamas medžių rezultatų vidurkis, kitaip vadinamas klasės priklausymo tikimybe  $p_i$ .

Klasifikavimo klaidos kainos minimizuojamos naudojant klasės priklausymo tikimybes (4.16) – mokymo metu perklasifikuojami vektoriai [24].

$$x = \arg \min \sum_i p_i C(y, c) \tag{4.21}$$

$x$  – atitinkama klasė;

$p_i$  – tikimybė klasės priklausymui;

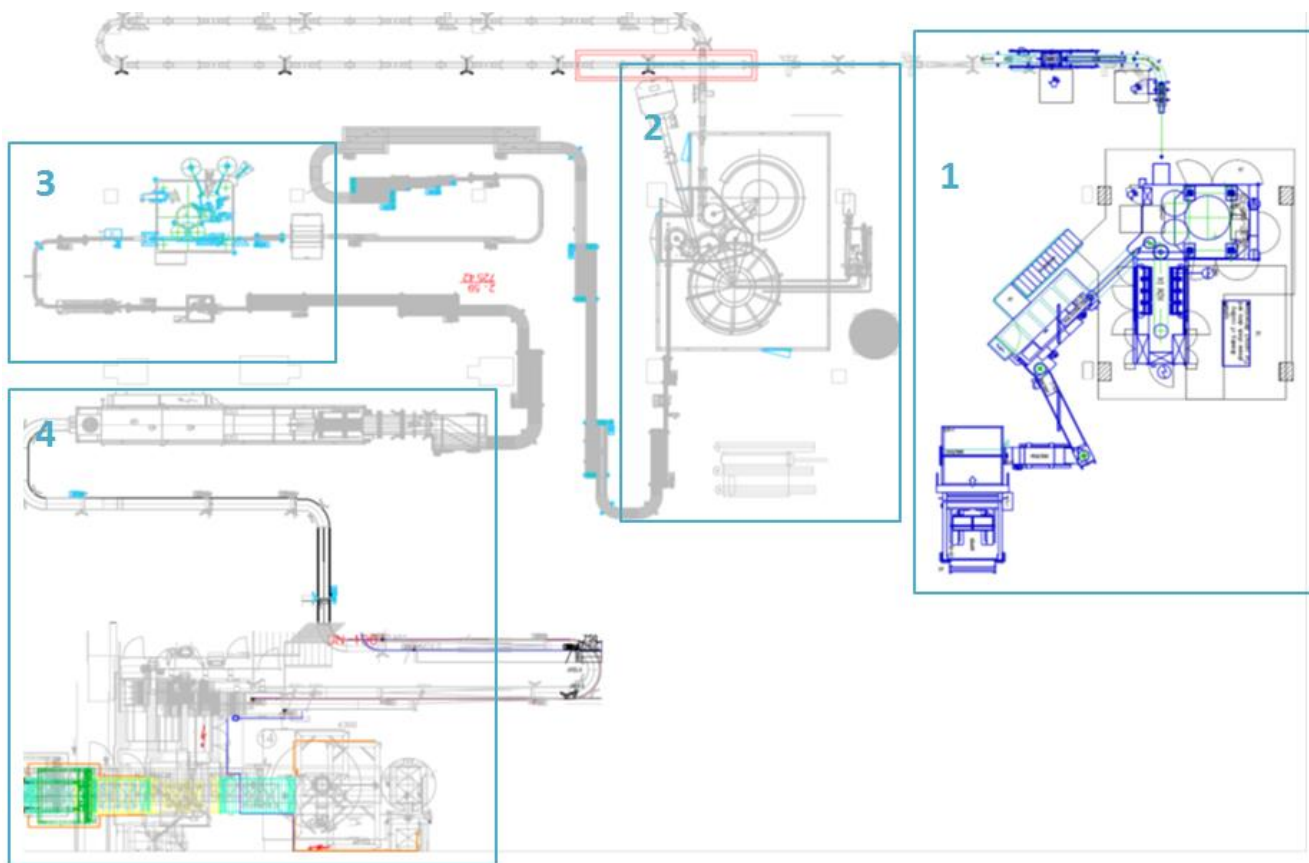
$C$  – klaidos kaina.

## 5. KLASIFIKAVIMO METODŲ REALIZAVIMAS

### 5.1. Naudojami duomenys klasifikavimo realizavimui

Tyrimui atlikti pasirinkta AB „Volfas Engelman“ gamyklos gėrimų pilstymo PET linija (5.1 pav.). Duomenys susiję su gamybos linijos informacija yra padalinami į keturis pagrindinius segmentus (įrenginius):

1. PET butelių formavimo įrenginys;
2. Pilstymo įrenginys;
3. Etikečių klijavimo, žymėjimo įrenginys;
4. Pakuočių – palečių formavimo įrenginys.



5.1 pav. Gėrimų pilstymo PET linija

Duomenys apie šiuos gamybos linijos segmentus surenkami naudojant „Arrow machine track“ monitoringo sistemą, energetinių resursų sunaudojimo bazę ir įrenginių priežiūros duomenų bazę „Arrow Novi“. Pasirinkti duomenys, kurie matuojami automatiniais skaitikliais, be rankinio skaitiklių stebėjimo – priimama, jog automatinės monitoringo sistemos klaidos tikimybė yra mažesnė, bei įvertinus rankinio duomenų įrašymo bazę, pastebėtas duomenų netikslumas.

Turimi duomenys apdorojami ir paruošiami klasifikavimui – gauti požymiai ir klasės parodo atitinkamą informaciją apie gamybą per vieną parą. Duomenys susiję su suvartojamais energijos ištekliais, paverčiami į santykinį dydį su pagaminta produkcija – taip gaunami linijos efektyvumo rodikliai pagal atitinkamą energijos suvartojimą. Likusių duomenų dydžiai nekeičiami. Gaunami požymiai (įvestys) klasifikatorių mokymui:

1. Elektros energija, [kW/h] / pagaminta produkcija, [hl];
2. Garų kiekis, [kg] / pagaminta produkcija, [hl];
3. Vandens kiekis, [l] / pagaminta produkcija, [hl];
4. CO<sub>2</sub> kiekis, [l] / pagaminta produkcija, [hl];
5. PET linijos vidutinis greitis, [vnt/h];
6. Likęs laikas iki periodinės profilaktinės priežiūros (visiems linijos įrenginiams);
7. Pagamintas produkcijos kiekis nuo profilaktinės priežiūros atlikimo, [hl];
8. Įrenginių prastovos laikas dėl elektrinio gedimo per paraėjusias 24 h, [min];
9. Įrenginių prastovos laikas dėl mechaninio gedimo per paraėjusias 24 h, [min];
10. Įrenginių prastovos laikas dėl derinimo gedimo per paraėjusias 24 h, [min];

Pagal surinktus duomenis suformuojami trys galimi klasių rinkiniai (išvestys):

- I. Klasės pagal gedimo įvykį:
  1. Gedimas;
  2. Nėra gedimo.
- II. Klasės pagal gedimo tipą (stabdantis gamybos procesą):
  1. Mechaninis;
  2. Elektrinis;
  3. Derinimo gedimas;
  4. Nėra gedimo.
- III. Klasės pagal įrenginio gedimą (stabdantis gamybos procesą):
  1. Nėra gedimo;
  2. PET butelių formavimo įrenginys;
  3. Pilstymo įrenginys;
  4. Etikečių klijavimo, žymėjimo įrenginys;
  5. Pakuočių – palečių formavimo įrenginys.

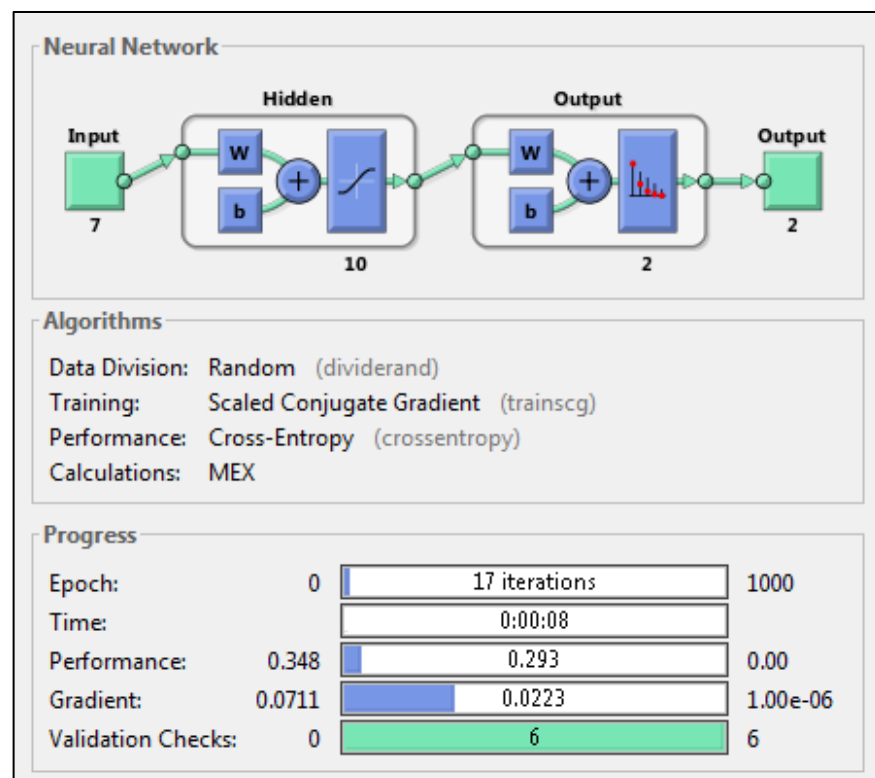
Įvertinama tai, jog per 24 valandas įvyksta daugiau nei vienas gedimas. Tokiais atvejais visiems per parą įvykusiems gedimams (klasėms) taikomi analogiškų reikšmių požymiai.

## 5.2. Klasifikavimo metodų mokymas

### 5.2.1. Neuroninio tinklo klasifikavimo metodas

Neuroninis tinklas realizuojamas „Matlab“ programine įranga, naudojant „Neural Net Pattern recognition“ įrankį (5.2 pav.), kuris skirtas neuroninių tinklų imitavimui, algoritmų ir funkcijų generavimui, bei mokymo vizualizavimui [27]. Neuroninio tinklo modelis įrenginių gedimo atpažinimo problemai spręsti kuriamas pagal „Nprtool“ įrankyje numatytuosius tinklo parametrus.

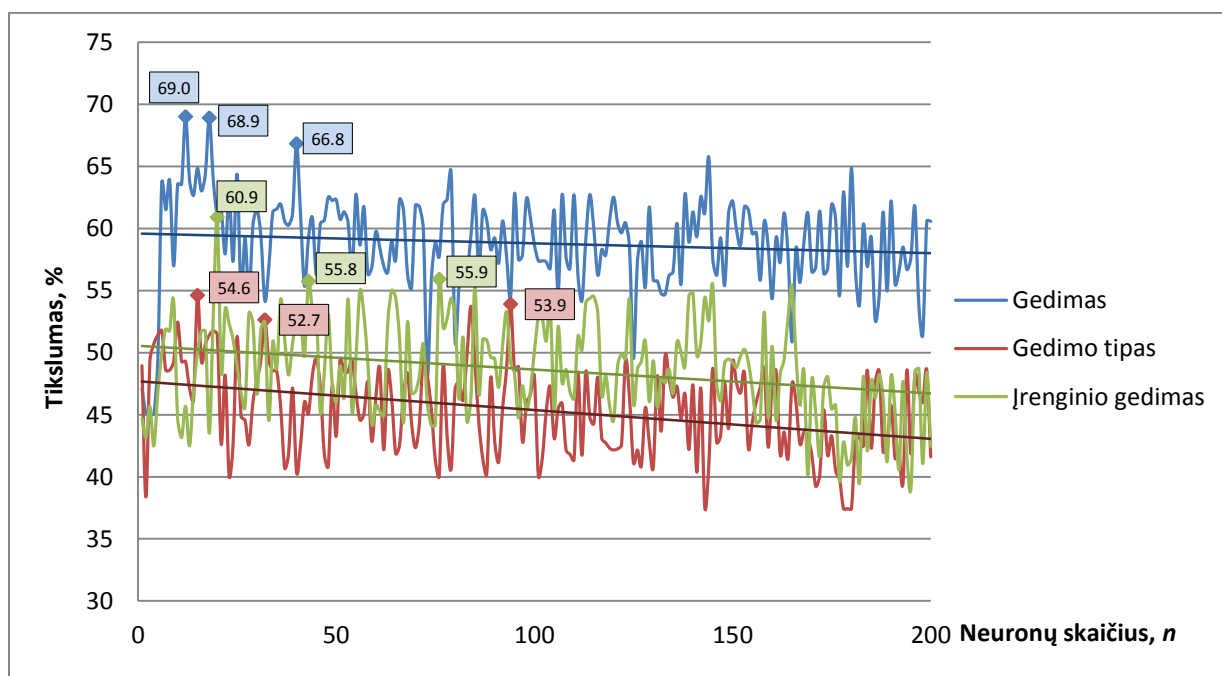
Duomenys padalinami į tris dalis: mokymui – tinklas nustatomas pagal gaunamą klaidą; validavimui – matuojamas tinklo sudėtingumas ir sustabdomas tinklo mokymas jeigu tinklo klaida didėja, kad būtų išvengta tinklo perkrovimo sudėtingomis funkcijomis; testavimui – patikrinamas NN tinklo tikslumas su nepriklausomomis reišmėmis tinklo mokymo metu ir po tinklo mokymo. Požymiai ir klasės išrenkami atsitiktine tvarka – neuroninio tinklo mokymui naudojami 70 proc. duomenų, validavimui – 15 proc., testavimui – 15 proc., atitinkamai kiekvienai klasei. Tiesioginio sklidimo daugiasluoksnio neuroninio tinklo realizavimui, naudojamas Levenberg – Marquardt algoritmas pagal, kurį priskiriamos neuronų svorių ir nuokrypių reikšmės.



5.2 pav. „Matlab“ programinės įrangos „Neural Net Pattern recognition“ įrankio neuroninio tinklo mokymo vizualizacijos pavyzdys

Pagal turimus 2015 m. duomenis suformuojamos trys klasių matricos: 2x284 dyžio matrica, kurioje nurodoma ar įvyko gedimas, 4x284 matrica nurodomas gedimo tipas, ir 5x284 duomenų matrica, kurioje nurodomas sugedęs įrenginys. Kiekvienoje matricoje 1 nurodo jog duomenų rinkinys priklauso atitinkamai klasei, 0 jog nepriklauso. Taip pat suformuojama 10x284 dydžio požymių matrica.

Sudaromas programinis kodas (1 priedas), kuriuo pagal kiekvieną suformuotą klasių rinkinių matricą atliekamas neuroninio tinklo mokymas keičiant neuronų skaičių paslėptajame sluoksnyje. Pastebimas klasifikavimo tikslumo mažėjimas didinant neuronų skaičių  $n$ , taigi nutraukiamas neuroninio tinklo mokymas ties 200 neuronų bei išrenkamos trys didžiausią tikslumą pasiekusios  $n$  reikšmės. Apskaičiuotas realizuoto neuroninio tinklo tikslumas pagal naudojamų neuronų skaičių ( $n$  kinta nuo 1 iki 200) pavaizduojamas 5.3 paveikslėlyje.

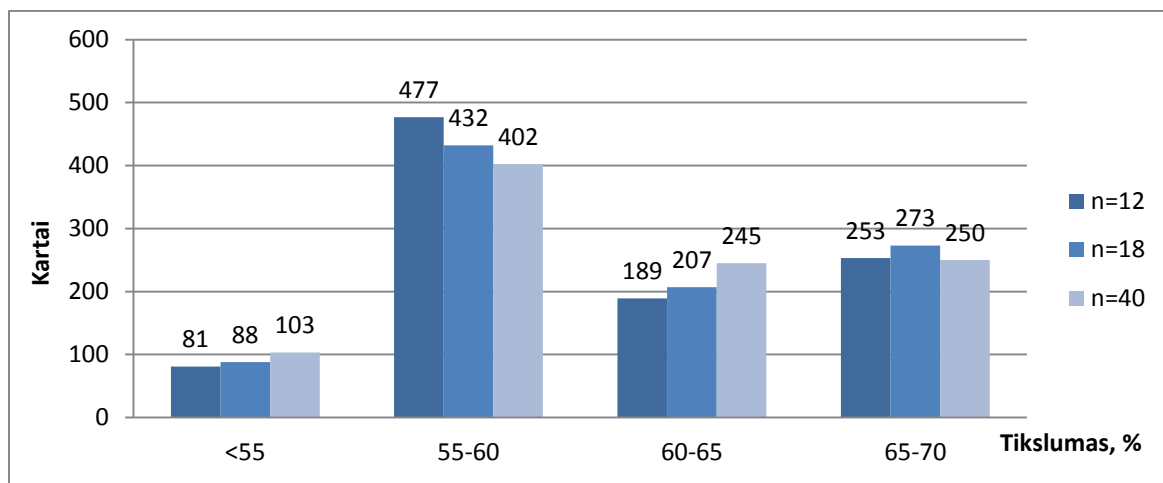


5.3 pav. Neuroninio tinklo tikslumas pagal naudojamų neuronų skaičių  $n$  paslėptajame sluoksnyje

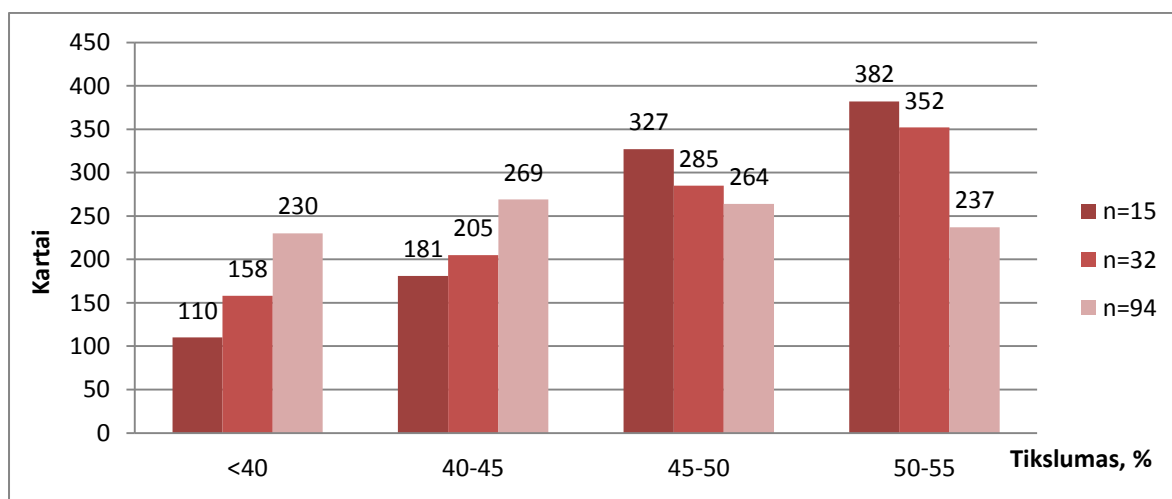
Pagal gautus pradinius klasifikavimo rezultatus pastebima, kad klasifikavimo tikslumas kinta plačiose ribose. Geriausi rezultatai gauti naudojant iki 20 neuronų paslėptajame sluoksnyje, tačiau didinant neuronų skaičių, taip pat kreivėje matomi šuoliai artimi maksimaliam pasiektam tikslumui. Tam didelę įtaką daro savybių padalinimas mokymui, testavimui bei validavimui atsitiktine tvarka. Taigi, siekiant įvertinti gaunamo klasifikavimo tikslumo pasikartojamumą atliekamas DNT mokymas 1000 kartų naudojant išskirtas tris  $n$  reikšmes (5.3 pav.) su kuriomis gautas maksimalus bendras tikslumas (1 priedas).



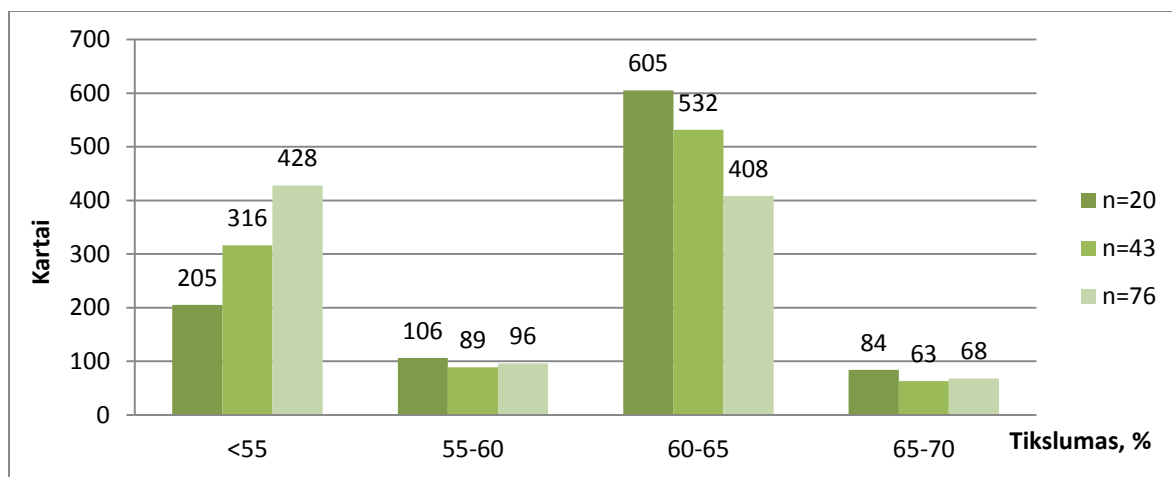
Naudojant gautus rezultatus, sudaromos histogramos kiekvienam klasių rinkiniui pagal atitinkamus neuronų skaičius.



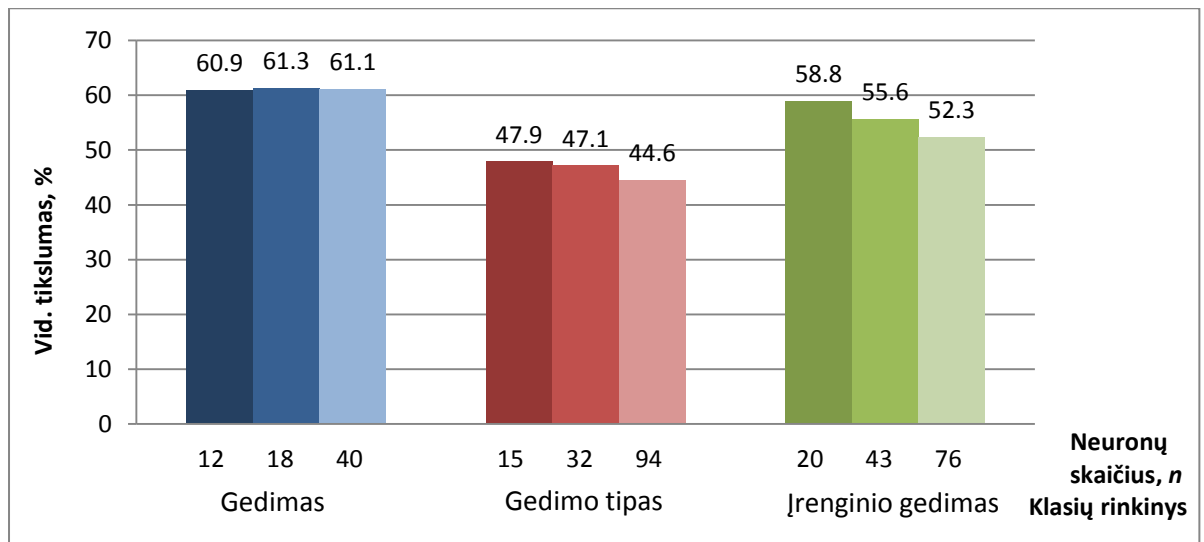
5.4 pav. Klasifikavimo tikslumo histograma, klasifikuojant gedimo įvykį



5.5 pav. Klasifikavimo tikslumo histograma, klasifikuojant gedimo tipą



5.6 pav. Klasifikavimo tikslumo histograma, klasifikuojant įrenginių gedimus



5.7 pav. Vidutinis klasifikavimo tikslumas pagal neuronų skaičių paslėptajame sluoksnyje atitinkamam klasių rinkiniui

Klasifikuojant gedimo įvykį didžiausias pasikartojamumas matomas ties 55 – 60 proc. klasifikavimo tikslumu, klasifikuojant gedimo tipus didžiausias pasikartojamumas yra ties 50 – 55 proc. tikslumu, klasifikuojant įrenginių gedimus ties 60 – 65 proc. tikslumu. Pagal histogramas apskaičiuojamas klasifikavimo vidurkis kiekvienam klasių rinkiniu su atitinkamu neuronų skaičiumi (5.7 pav.). Pastebima, kad vidutinis klasifikavimo tikslumo skirtumas gana didelis lyginant su maksimaliais rezultatais.

Klasifikuojant gedimo įvykį tikslinga naudoti 18 neuronų paslėptajame neuroninio tinklo sluoksnyje, o klasifikuojant gedimo tipą tikslinga naudoti 15 neuronų – aukščiausias klasifikavimo vidurkis bei didžiausias pasikartojamumas ties 65-75 proc. tikslumu (gedimo įvykis) ir ties 50-55 proc. tikslumu (gedimo tipas). Klasifikuojant įrenginių gedimus vidutinis klasifikavimo tikslumas (aliekant neuroninio tinklo mokymą 1000 kartų) daug skiriasi nuo maksimalaus tikslumo, 5.6 paveiksle pateiktoje histogramoje matome, kad tikslinga naudoti 20 neuronų – didžiausias pasikartojamumas ties 60 – 70 proc. tikslumu.

Alpažinta klasė	1. Nėra gedimo	118 41.5%	48 16.9%	71.1% 28.9%
	2. Gedimas	40 14.1%	78 27.5%	66.1% 33.9%
		74.7% 25.3%	61.9% 38.1%	69.0% 31.0%
		1. Nėra gedimo	2. Gedimas	
		Tiksli klasė		

5.8 pav. ANN tikslumo matrica, kai  $n=12$ , klasifikuojant gedimo įvykius

Atpažinta klasė	1.Nėra gedimo	114 40.1%	11 3.9%	40 14.1%	44 15.5%	54.5%
	2.Elektrinis gedimas	0 0.0%	1 0.4%	0 0.0%	0 0.0%	100%
	3.Mechaninis gedimas	3 1.1%	2 0.7%	12 4.2%	5 1.8%	54.5%
	4.Derinimo gedimas	9 3.2%	1 0.4%	14 4.9%	28 9.9%	53.8%
		90.5% 9.5%	6.7% 93.3%	18.2% 81.8%	36.4% 63.6%	54.6% 45.4%
		1.Nėra gedimo	2.Elektrinis gedimas	3.Mechaninis gedimas	4.Derinimo gedimas	
		Tiksli klasė				

5.9 pav. ANN tikslumo matrica, kai  $n=15$ , klasifikuojant gedimų tipus

Atpažinta klasė	1.Nėra gedimo	124 43.7%	9 3.2%	17 6.0%	11 3.9%	19 6.7%	68.9%
	2. Butelių formavimo įreng.	0 0.0%	9 3.2%	5 1.8%	3 1.1%	7 2.5%	37.5%
	3. Butelių išpilstymo įreng.	0 0.0%	1 0.4%	2 0.7%	1 0.4%	1 0.4%	40.0%
	4. Etiketavimo įreng.	1 0.4%	1 0.4%	3 1.1%	8 2.8%	12 4.2%	32.0%
	5. Pakavimo įreng.	1 0.4%	6 2.1%	9 3.2%	4 1.4%	30 10.6%	60.0%
		98.4% 1.6%	34.6% 65.4%	5.6% 94.4%	29.6% 70.4%	43.5% 56.5%	60.9% 39.1%
		1.Nėra gedimo	2. Butelių formavimo įreng.	3. Butelių išpilstymo įreng.	4. Etiketavimo įreng.	5. Pakavimo įreng.	
		Tiksli klasė					

5.10 pav. ANN tikslumo matrica, kai  $n=20$ , klasifikuojant įrenginių gedimus

Tiksliausias neuroninio tinklo rezultatas klafikuojant, ar gedimas įvyko, gautas naudojant 12 neuronų paslėptajame sluoknyje – 69 proc., (tikslumo matrica pateikta 5.8 paveiksle), naudojant 18 neuronų – 68,9 proc., 40 neuronų – 66,8 proc. 75 proc. (118 tikslų iš 158) atpažįstama klasė „Gedimas“, 62 proc. atpažįstama klasė „Gedimo nėra“ (5.8 pav.).

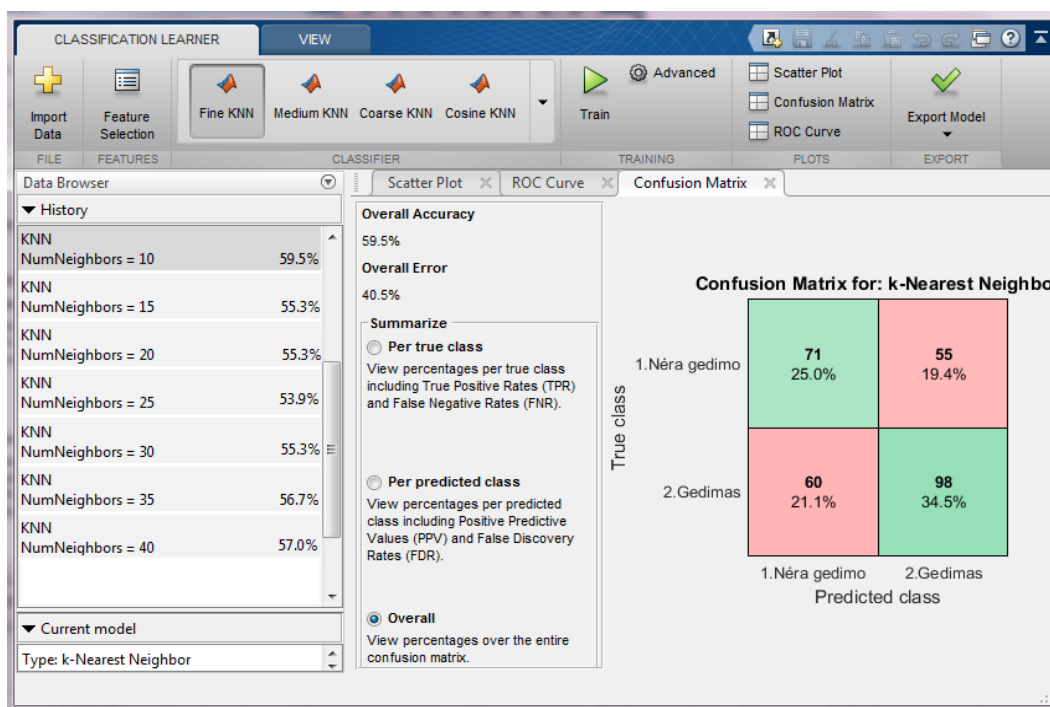
Klasifikuojant gedimo tipus gautas maksimalus tikslumas tik 54,6 proc., su 15 neuronų (5.9 pav), naudojant 93 neuronus – 53,9 proc., 32 neuronus – 52,7 proc. Tikslumo matrica (5.9 pav.) rodo, jog šis ANN tinklas nėra tinkamas klasifikuoti „Elektrinis gedimas” ir „Mechaninis gedimas” klasių – vidutinis šių klasių TPR tikslumas tik 16 proc.

Įrenginių gedimų prognozavimo maksimalus bendras rezultatas – 60,9 proc. su 20 neuronų, 55,9 proc. su 76 neuronais, 55,8 proc. – 43 neuronais. Tiksliausio rezultato tikslumo matrica (5.10 pav.) rodo jog, gedimo nebūvimas klasifikuojamas net 98 proc. tikslumu, tačiau butelių formavimo, etiketavimo, pakavimo įrenginiai atpažįstami tik 39 proc. vidutiniu tikslumu, o butelių išpilstymo įrenginys tik 5.6 proc. tikslumu.

Gautos realizuoto neuroninio tinklo tikslumo matricos rodo, jog vertinant tikslumą pagal kiekvieną klasę, trijų klasių iš gedimo tipo ir įrenginių gedimų klasifikuoti nepavyko, taigi prognozuojami rezultatai nepatikimi. NN klasifikatorius 69 proc. tikslumu prognozuoja, jog visoje gamybos linijoje įvyks (TPR –74,7 proc.) arba neįvyks gedimas (TPR –61,9 proc.), tačiau tinkamesnis klasifikatorius su tikslesniu „Gedimas“ klasės atpažinimu.

### 5.2.2. Artimiausio kaimyno klasifikavimo metodas

Artimiausio kaimyno klasifikavimo metodas realizuojamas „Matlab” programine įranga, naudojant „Classification learner” įrankį (5.11 pav.). Šiuo įrankiu mokomi klasifikavimo modeliai duomenų klasifikavimui, taip pat išvedami gauti rezultatai bei sugeneruojamas programinis kodas ir funkcija klasifikatorių tikslumui įvertinti su nepriklausomais nuo klasifikavimo mokymo duomenimis [28].



5.11 pav. „Matlab” programinės įrangos „Classification learner” įrankio langas

Suformuojama analogiška dirbtiniam neuroniniam tinklo klasifikavimui  $10 \times 284$  dydžio požymių matrica pagal turimus 2015 metų duomenis. Taip pat suformuojamos trys  $1 \times 284$  dydžio klasių matricos – 1 nurodo, kad nėra gedimo, toliau klasių matricose seka atitinkamos klasės numeris pagal gedimo įvykį, gedimo tipą, įrenginį.

Klasifikavimo tikslumo verinimui pasirenkamas kryžminis validavimo (angl. *cross validation*) metodas, kadangi yra tinkamesnis mažesnių duomenų rinkinių analizei (iki 6000 požymių įrašų) [28]. Pagal nutylėjimą nustatomos 5 duomenų divizijos (nesusikertantys duomenų rinkiniai), kuriose požymiai atitinkamai klasei padalinami į vienodas dalis. Šios požymių ir klasių divizijos apsaugo nuo duomenų „perkrovimo“ – labai aukšto tikslumo klasifikuojant pagal vieną duomenų rinkinį. Tokiu būdu klasifikatorius generalizuojamas nepriklausomiems nuo mokymo požymiams klasifikuoti.

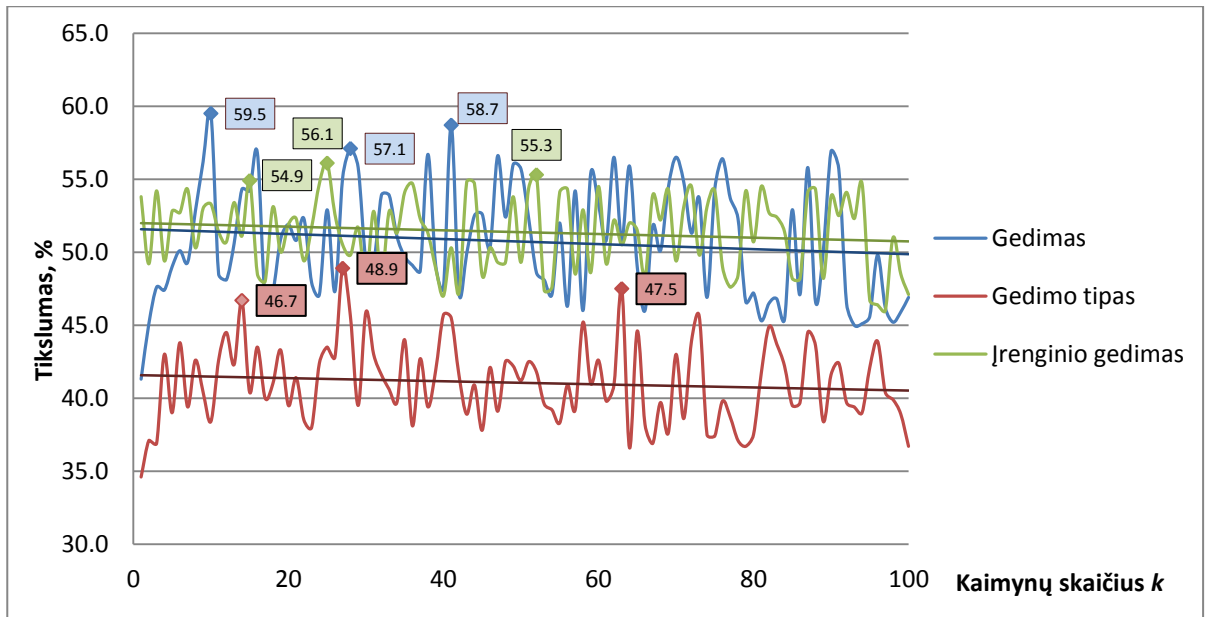
Vertinant klasifikavimo tikslumą kryžminiu validavimo metodu:

- Duomenų rinkinys padalinamas į 5 nesusikertančius rinkinius su vienodu požymių ir klasių santykiu;
- Klasifikavimo modelis mokomas su duomenimis, kurie nepriklauso nustatytoms rinkinių divizijoms;
- Įvertinamas klasifikavimo tikslumas naudojant duomenis, kurie priklauso nustatytoms rinkinių divizijoms;
- Apskaičiuojamas vidutinis nustatytomis divizijomis gautas tikslumas.

„Classification learner„ įrankis artimiausio kaimyno klasifikavimo mokymui naudoja *fitcknn* funkciją. Atstumo matavimas pagal kurį nustatomi atstumo svoriai – Euklido (gautas atstumas lygus svoriui). Sudaromas programinis kodas (2 priedas), funkcija (3 priedas) ir atliekamas artimiausio kaimyno klasifikavimo mokymas keičiant naudojamų kaimynų skaičių  $k$  nuo 1 iki 100. Naudojamas duomenų standartizavimas, kadangi turimos požymių reikšmės išreikštos skirtingose skalėse – požymių reikšmių koordinatės naudojamos mokyme sutraukiamos į vieną skalę.

Remiantis gautais tyrimo klasifikavimo rezultatais pavaizduojamas tikslumas pagal naudojamų artimiausių kaimynų skaičių  $k$ , realizuojamas klasifikatorius didinant  $k$  reikšmę (5.12 pav.). Pastebėta, kad didinant  $k$  kaimynų skaičių mažėja gaunamas klasifikavimo tikslumas, aitinkamai kiekvienam klasifikavimo rinkiniui, taigi klasifikatorių realizavimas nutraukiamas ties  $k = 100$ .

Sudaromi klasifikatorių modeliai, kurių rezultatai buvo tiksliausi. Sudaryti modeliai naudojami klasifikatoriaus tikslumui įvertinti su nepriklausomais „naujais“ duomenimis.



5.12 Artimiausio kaimyno klasifikavimo tikslumas pagal naudojamų kaimynų skaičių  $k$

Tiksli klasė	1. Nėra gedimo	78 27.5%	48 16.9%
	2. Gedimas	67 23.6%	91 32.0%
		1. Nėra gedimo	2. Gedimas
Atpažinta klasė			

5.13 pav.  $k$ -NN klasifikavimo tikslumo matrica, kai  $k=10$ , klasifikuojant gedimo įvykius

Tiksli klasė	1. Nėra gedimo	108 38.0%	0 0.0%	6 2.1%	12 4.2%
	2. Elektrinis gedimas	9 3.2%	0 0.0%	2 0.7%	4 1.4%
	3. Mechaninis gedimas	43 15.1%	0 0.0%	11 3.9%	12 4.2%
	4. Derinimo gedimas	48 16.9%	0 0.0%	9 3.2%	20 7.0%
		1. Nėra gedimo	2. Elektrinis gedimas	3. Mechaninis gedimas	4. Derinimo gedimas
Atpažinta klasė					

5.14 pav.  $k$ -NN klasifikavimo tikslumo matrica, kai  $k=27$ , klasifikuojant gedimo tipus

Tiksli klasė	1.Nėra gedimo	121 42.6%	1 0.4%	0 0.0%	2 0.7%	2 0.7%
	2.Butelių formavimo įreng.	10 3.5%	2 0.7%	3 1.1%	2 0.7%	9 3.2%
	3.Butelių išpilstymo įreng.	18 6.3%	2 0.7%	1 0.4%	1 0.4%	14 4.9%
	4.Etiketavimo įreng.	12 4.2%	2 0.7%	0 0.0%	3 1.1%	10 3.5%
	5.Pakavimo įreng.	22 7.7%	6 2.1%	5 1.8%	4 1.4%	32 11.3%
Atpažinta klasė		1.Nėra gedimo	2.Butelių formavimo įreng.	3.Butelių išpilstymo įreng.	4.Etiketavimo įreng.	5.Pakavimo įreng.

5.15 pav. k-NN klasifikavimo tikslumo matrica, kai  $k=15$ , klasifikuojant įrenginių gedimus

Klasifikuojant ar įvyko gedimas, gaunamas tiksliausias rezultatas 59.5 proc., naudojant  $k=10$ . Pagal tikslumo matricą (5.13 pav.) matyti, kad klases „Nėra gedimo“ ir „Gedimas“ klasifikuojama panašiu tikslumu, atitinkamai 61proc. (78 teisingi iš 126) ir 58 proc. (91 teisingi iš 158) FPR tikslumu.

Klasifikuojant gedimo tipus bendras tiksliausias rezultatas pasiektas, tik 48,9 proc. naudojant  $k=27$ . Tikslumo matrica (5.14 pav.) parodo, kad k-NN metodu nepavyko klasifikuoti „Elektrinis gedimas“ klasės, tiksliausiai klasifikuojama klasė – „Nėra gedimo“ 86 proc. (108 teisingi iš 126), kiti gedimai klasifikuojami tik apie 22 proc. tikslumu.

Prognozuojant įrenginių gedimus gautas tik 56,1 proc. maksimalus tikslumas su 25 artimiausiais kaimynais. Klasių „Butelių formavimo įrenginys“, „Butelių išpilstymo įrenginys“, „Etiketavimo įrenginys“ vidutinis atpažinimo tikslumas neviršija net 7 proc. (5.15 pav.) . Tiksliausiai klasifikuojama klasė – „Nėra gedimo“ 96 proc. (121 teisingi iš 126), klasė „Pakavimo įrenginys,, – 46 proc. tikslumas (32 teisingi iš 69).

k-NN klasifikavimo tikslumo bendras klasių rinkinių rezultatas kinta ne itin plačiose ribose, tačiau vertinant atskirų klasių atpažinimą – tikslumo nuokrypis yra didelis (labai žemas rezultatas arba aukštas rezultatas atitinkamai klasei). Realizuoto k-NN metodo prognozavimo rezultatai nėra tenkinami – pavienės klasės neatpažįstamos ir atpažinimo rezultatas itin prastas.

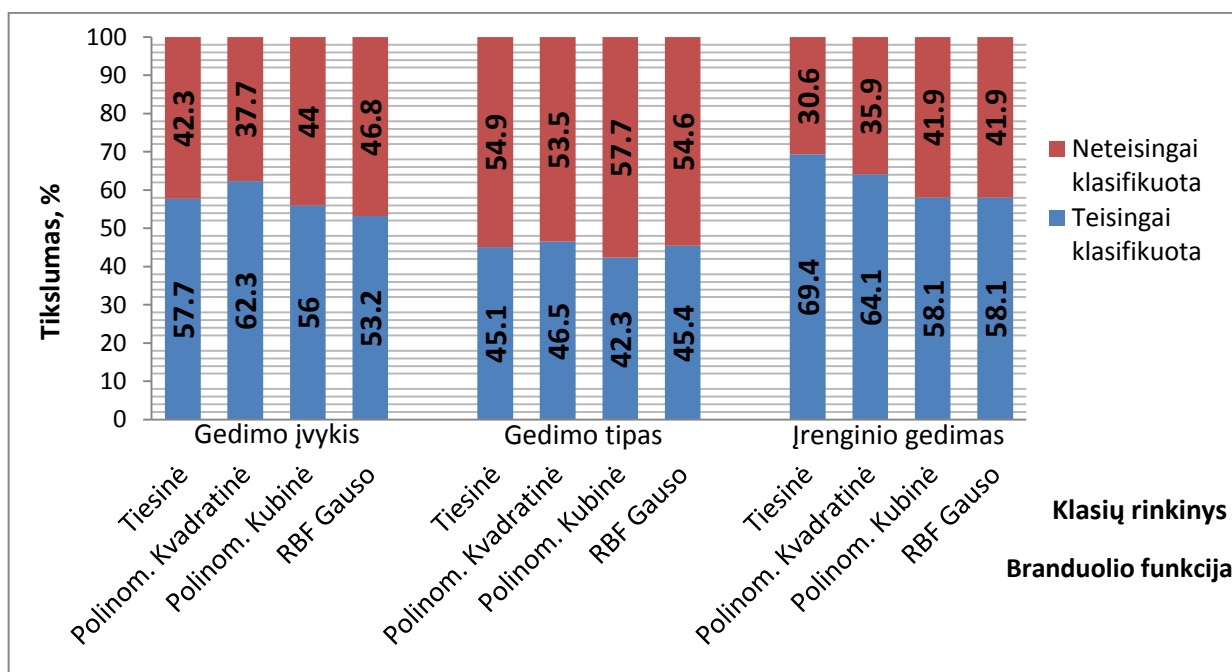
### 5.2.3. Atraminio vektoriaus mašinos klasifikavimo metodas

Atraminio vektoriaus klasifikavimo metodas realizuojamas „Matlab“ programine įranga, naudojant „Classification learner“ įrankį (5.11 pav). Klaisifikavimo mokymas, testavimas, tikslumo

vertinamas atliekamas analogiškai kaip aprašyta 5.2.2 dalyje. Naudojamos analogiškos požymių bei klasių matricos, kaip ir klasifikuojant k-NN metodu.

Realizuojamas atraminio vektoriaus klasifikavimo metodas naudojant „Matlab“ *fitcsvm* funkciją. SVM mokomas naudojant tiesinę (*angl. Linear*), kvadratinę (*angl. Quadratic*), kubinę (*angl. Cubic*), Gauso (*angl. Gaussian*), branduolio funkcijas kiekvienam klasifikavimo rinkiniui (3 priedas). Apskaičiuojamas klasifikavimo tikslumas pagal naudojamas funkcijas (5.16 pav.).

Klasifikuojant ar įvyko gedimas pasiektas tiksliausias rezultatas su polinominė kvadratinė branduolio funkcija – 62,3 proc. Gedimo tipo atpažinimo rezultatas itin žemas – tikslumas pagal naudojamas branduolio funkcijas kinta nuo 42,3 iki 46,5 proc. Geriausias rezultatas klasifikuojant įrenginių gedimus pasiektas naudojant tiesinę branduolio funkciją – 69,4 proc. tikslumas, pagal skirtingas branduolio funkcijas tikslumas kinta 10 proc. intervale.



5.16 pav. SVM klasifikavimo tikslumas pagal naudojamas branduolio funkcijas

Tiksliai klasė	1. Nėra gedimo	67 23.6%	59 20.8%
	2. Gedimas	48 16.9%	110 38.7%
		1. Nėra gedimo	2. Gedimas
		Atpažinta klasė	

5.17 pav. SVM klasifikavimo tikslumo matrica, branduolio funkcija – polinominė kvadratinė, klasifikuojant gedimo įvykius



Tiksli klasė	1.Nėra gedimo	95 33.5%	0 0.0%	15 5.3%	16 5.6%
	2.Elektrinis gedimas	12 4.2%	0 0.0%	2 0.7%	1 0.4%
	3.Mechaninis gedimas	31 10.9%	0 0.0%	13 4.6%	22 7.7%
	4.Derinimo gedimas	38 13.4%	0 0.0%	15 5.3%	24 8.5%
Atpažinta klasė		1.Nėra gedimo	2.Elektrinis gedimas	3.Mechaninis gedimas	4.Derinimo gedimas

5.18 pav. SVM klasifikavimo tikslumo matrica, branduolio funkcija – polinominė kvadratinė, klasifikuojant gedimo tipus

Tiksli klasė	1.Nėra gedimo	126 44.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%
	2.Butelių formavimo įreng.	0 0.0%	4 1.4%	0 0.0%	2 0.7%	20 7.0%
	3.Butelių išpilstymo įreng.	2 0.7%	2 0.7%	0 0.0%	2 0.7%	30 10.6%
	4.Etiketavimo įreng.	2 0.7%	0 0.0%	0 0.0%	5 1.8%	20 7.0%
	5.Pakavimo įreng.	0 0.0%	2 0.7%	0 0.0%	5 1.8%	62 21.8%
Atpažinta klasė		1.Nėra gedimo	2.Butelių formavimo įreng.	3.Butelių išpilstymo įreng.	4.Etiketavimo įreng.	5.Pakavimo įreng.

15.19 pav. SVM klasifikavimo tikslumo matrica, branduolio funkcija – tiesinė, klasifikuojant įrenginių gedimus

Pagal gautas tikslumo matricas kiekvienam rinkiniui, pastebima, kad klasifikuojant ar gedimas įvyko „Gedimas“ klasė atpažįstama tiksliau – 69 proc. tikslų pasirinkimų iš klasės „Gedimas“, 53 proc. tikslų iš klasės „Nėra gedimo“ (5.17 pav.). Gedimo tipo atpažinime klasė „Nėra gedimo“ atpažįstama 75 proc. (95 teisingi iš 126) tikslumu, tačiau klasė „Elektrinis gedimas“, neapažįstama bei kiti gedimo tipai atpažįstami tik apie 26 proc. tikslumu (5.18 pav.). Klasifikuojant įrenginių gedimus be klaidų atpažįstama jog gedimas neįvyko, taip pat 90 proc. tikslumu atpažįstamas pakavimo įrenginio gedimas, tačiau klasė „Butelių išpilstymo įrenginys“ neatpažįstama bei kitų įrenginių gedimų klasifikavimas siekia vos 17 proc. tikslumą (5.19 pav.). Taigi SVM metodo prognozavimas tinkamas tik gedimo įvykio nustatymui (apie 30 proc. paklaida).

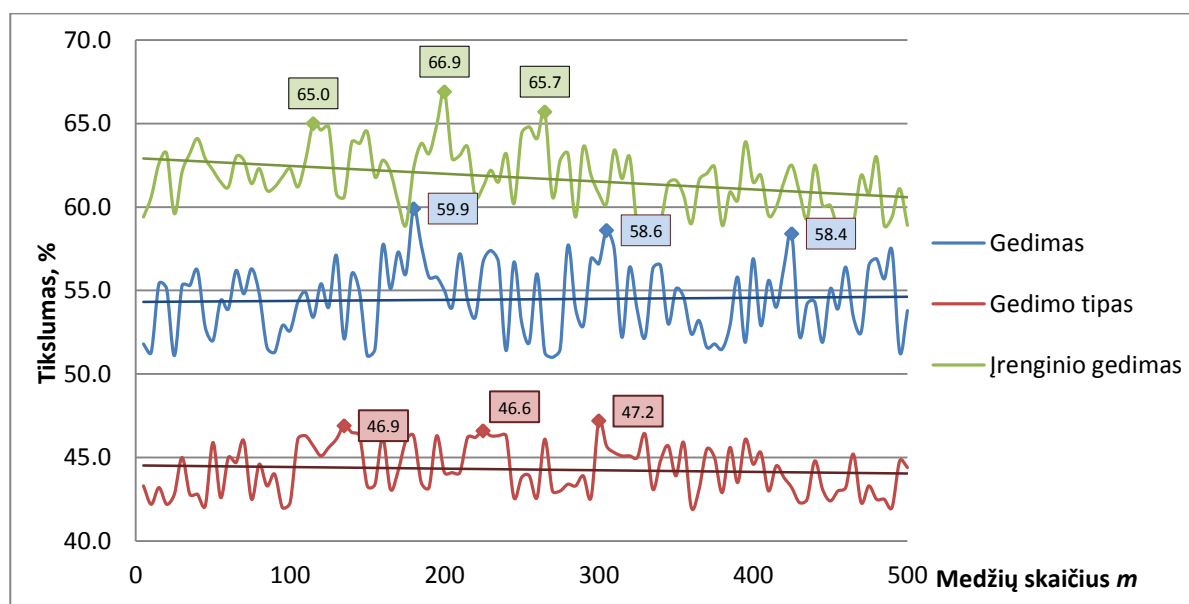
#### 5.2.4. Sprendimo medžio klasifikavimo metodas

Sprendimo medžio klasifikatoriaus gaubiamasis metodas realizuojamas „Matlab“ programine įranga, naudojant „Classification learner“ įrankį (5.11 pav), funkcija – *fitensemble bag*. Klaisifikavimo mokymas, testavimas, tikslumo vertinamas atliekamas analogiškai kaip aprašyta 5.2.2 dalyje. Sudaromas programinis kodas (3 priedas). Naudojamos 5.2.2 dalyje suformuotos požymių bei klasių matricos.

Atliekamas „Bagged tree“ medžio klasifikatoriaus mokymas keičiant naudojamų medžių skaičių  $m$  nuo 5 iki 500, kiti parametrai paliekami numatytieji – klaidos kaina  $C=0,1$ . Šio klasifikatoriaus mokymas reikalauja daug skaičiavimo resursų, taigi medžių skaičius  $m$  dininamas intervalais po 5 – taip gaunamas optimalus 10 sekundžių sprendimo medžio mokymo laikas.

Pavaizduojamas klasifikavimo tikslumas kiekvienam klasių rinkiniui pagal naudojamų medžių skaičių (15.20 pav.). Sprendimo medžio klasifikatorius tiksliausiai atpažįsta sugedusius įrenginius – tikslumas kinta siaurame intervale nuo 58,8 proc. iki 66,9 proc., tiksliausias rezultatas gautas naudojant  $m=200$ . Ar įvyko gedimas klasifikuojama vidutiniškai apie 54,5 proc., aukščiausias rezultatas – 59,9 proc. su 180 medžių. Prognozuojant gedimo tipą pasiekiamas tik 47,2 proc. maksimalus tikslumas.

Pastebimas klasifikavimo tikslumo mažėjimas didinat medžių skaičių klasifikuojant ar įvyko gedimas bei sugedusius įrenginius, tačiau klasifikuojant gedimo tipus vidutinis klasifikavimo tikslumas išlieka stabiliai žemas.



15.20 pav. Sprendimo medžio klasifikavimo tikslumas pagal naudojamų medžių skaičių  $m$

Tiksli klasė	1.Nėra gedimo	57 20.1%	69 24.3%
	2.Gedimas	48 16.9%	110 38.7%
Atpažinta klasė		1.Nėra gedimo	2.Gedimas

5.21 pav. Sprendimo medžio klasifikavimo tikslumo matrica, 59,9 proc. bendras tikslumas,  $m=180$ , klasifikuojant gedimo įvykius

Tiksli klasė	1.Nėra gedimo	95 33.5%	2 0.7%	11 3.9%	18 6.3%
	2.Elektrinis gedimas	10 3.5%	0 0.0%	3 1.1%	2 0.7%
	3.Mechaninis gedimas	34 12.0%	1 0.4%	12 4.2%	19 6.7%
	4.Derinimo gedimas	37 13.0%	0 0.0%	13 4.6%	27 9.5%
Atpažinta klasė		1.Nėra gedimo	2.Elektrinis gedimas	3.Mechaninis gedimas	4.Derinimo gedimas

5.22 pav. Sprendimo medžio klasifikavimo tikslumo matrica, 47,2 proc. bendras tikslumas,  $m=300$ , klasifikuojant gedimo tipus

Tiksli klasė	1.Nėra gedimo	126 44.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%
	2. Butelių formavimo įreng.	0 0.0%	1 0.4%	8 2.8%	3 1.1%	14 4.9%
	3. Butelių išpilstymo įreng.	1 0.4%	2 0.7%	6 2.1%	2 0.7%	25 8.8%
	4. Etiketavimo įreng.	0 0.0%	2 0.7%	6 2.1%	7 2.5%	12 4.2%
	5. Pakavimo įreng.	0 0.0%	6 2.1%	8 2.8%	5 1.8%	50 17.6%
		1.Nėra gedimo	2. Butelių formavimo įreng.	3. Butelių išpilstymo įreng.	4. Etiketavimo įreng.	5. Pakavimo įreng.
		Atpažinta klasė				

5.23 pav. Sprendimo medžio klasifikavimo tikslumo matrica, 66,9 proc. bendras tikslumas,  $m=200$ , klasifikuojant įrenginių gedimus

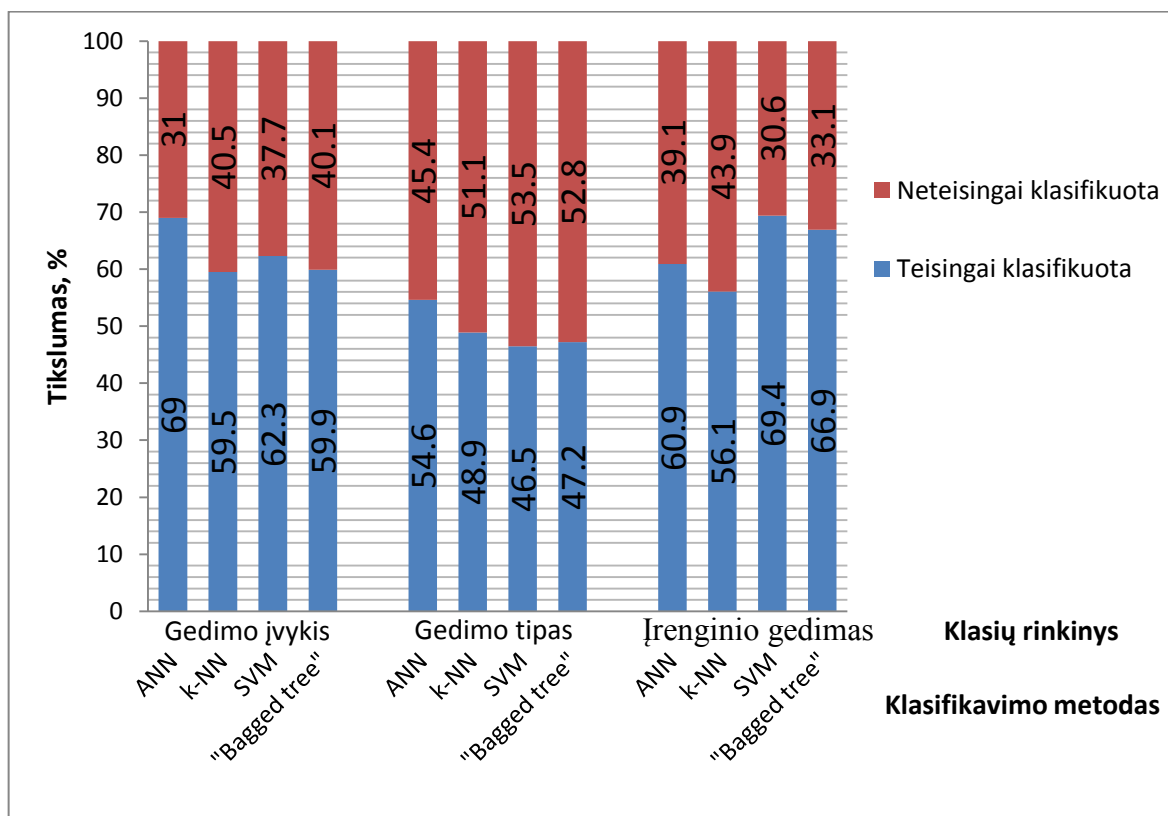
Gedimo įvykio atpažinimo tikslumo matrica (5.21 pav.) rodo, kad sprendimo medžio metodu tiksliau klasifikuojama jog gedimas įvyks – 70 proc. tikslumas, jog gedimo nėra – tik 45 proc. tikslumas. Klasifikuojant gedimo tipus pagal tikslumo matricą (5.22 pav.) pastebima, jog elektrinis gedimas yra neatpažįstamas, net 75 proc. klasės „Nėra gedimo“ objektų klasifikuojama tiksliai, kiti gedimo tipai klasifikuojami tik apie 27 proc. tikslumu. Klasifikuojant gedimus pagal įrenginį, gedimo nebūvimas atpažįstamas be klaidų bei pakavimo įrenginio gedimas atpažįstamas su 50 teisingų nustatymų iš 69 atvejų, tačiau kitos klasės atpažįstamos itin prastai – dauguma atvejų (51 iš 89) priskiriama klasei „Pakavimo įrenginys“ (5.23 pav.). Realizuotas „Bagged tree“ klasifikavimo metodo prognozavimo tikslumas nėra tinkamas gedimo tipui, įrenginio gedimui atpažinti – nepavyko klasifikuoti pavienių klasių,

### 5.2.5. Klasifikavimo rezultatų apibendrinimas

Realizavus klasifikavimo metodus pagal 2015 metų turimus duomenis pastebima, jog gedimo tipo ir įrenginio gedimo klasių nepavyko prognozuoti nei su vienu klasifikavimo metodu. Gedimo įvykio atpažinime tiksliausi rezultatai gauti naudojant neuroninio tinklo klasifikatorių – 69 proc. Taip pat neuroninis tinklas parodė geriausią rezultatą lyginant klasifikatorius pagal atskirą klasių atpažinimą gedimo tipo ir įrenginių gedimo klasifikavimui, tačiau NN klasifikatoriumi šių klasių rinkinių

prognozavimas nera tinkamas, kadangi atpažįstamos klasės, kurių mokymo ir testavimo imtys didžiausios.

Klasifikatoriais k-NN, SVM, „Bagged tree“ gauti rezultatai panašūs – taip pat, tiksliausiai atpažįstamos klasės, kurių požymių matricos sudarė daugiausiai duomenų. Bendras gautų klasifikavimo metodų tikslumo palyginimas pateiktas 5.24 paveiksle.



5.24 pav. Realizuotų klasifikavimo metodų tikslumo palyginimas

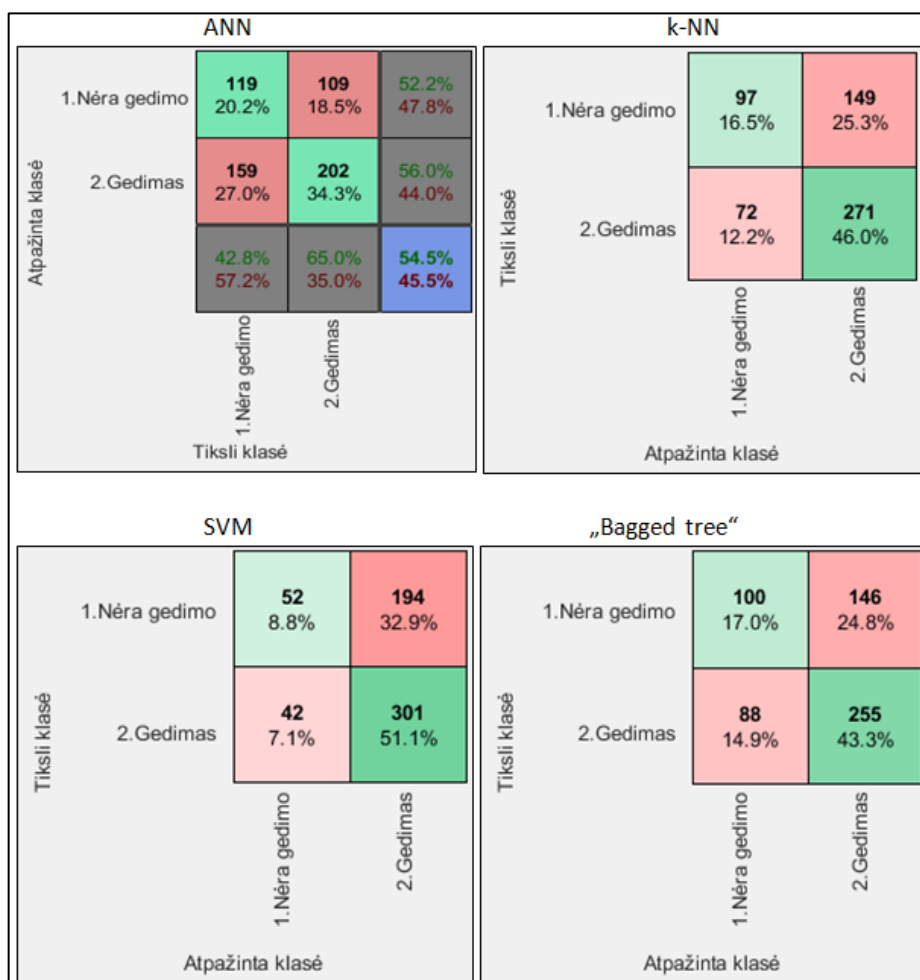
### 5.3. Klasifikavimo metodų mokymas su papildomais požymių rinkiniais

Klasifikavimo metodams mokyti 5.2 dalyje naudoti požymių ir klasių rinkiniai papildomi duomenimis iš tiriamojo objekto pasitelkiant 2014 m. duomenų bazės sukauptą informaciją. Siekiant gauti tikslesnius klasifikavimo rezultatus, proporcingai visiems požymių rinkiniams papildomos gedimo tipo bei įrenginių gedimų klasės, kurių 5.2 dalyje gautas klasifikavimo rezultatas buvo itin žemas – taip sudaromos panašaus dydžio klasių imtys. Klasių su papildomais požymių rinkiniais pasisiskirstymas pateiktas 5.1 lentelėje.

5.1 lentelė. Klasių pasiskirstymas su papildomas požymiais

Klasių rinkinys	Klasė	Kiekis
Gedimo įvykis	Nėra gedimo	246
	Gedimas	343
Gedimo tipas	Elektrinis gedimas	104
	Mechaninis gedimas	118
	Derinimo gedimas	121
Įrenginio gedimas	PET butelių formavimo įrenginys;	86
	Pilstymo įrenginys;	88
	Etikečių klijavimo, žymėjimo įreng.	79
	Pakuočių – palečių formavimo įreng.	100

Atliekamas klasifikavimo metodų realizavimas analogiškai 5.2 dalyje atliktam klasifikatorių mokymui naudojant papildomus klasių požymius. Išrenkami tiksliausi rezultatai – kiekvieno klasifikatoriaus tiksumo matricos atitinkamam klasių rinkiniui pateiktos 5.25 – 5.27 paveiksluose.



5.25 pav. Klasifikavimo tikslumo matricos, klasifikuojant gedimo įvykius



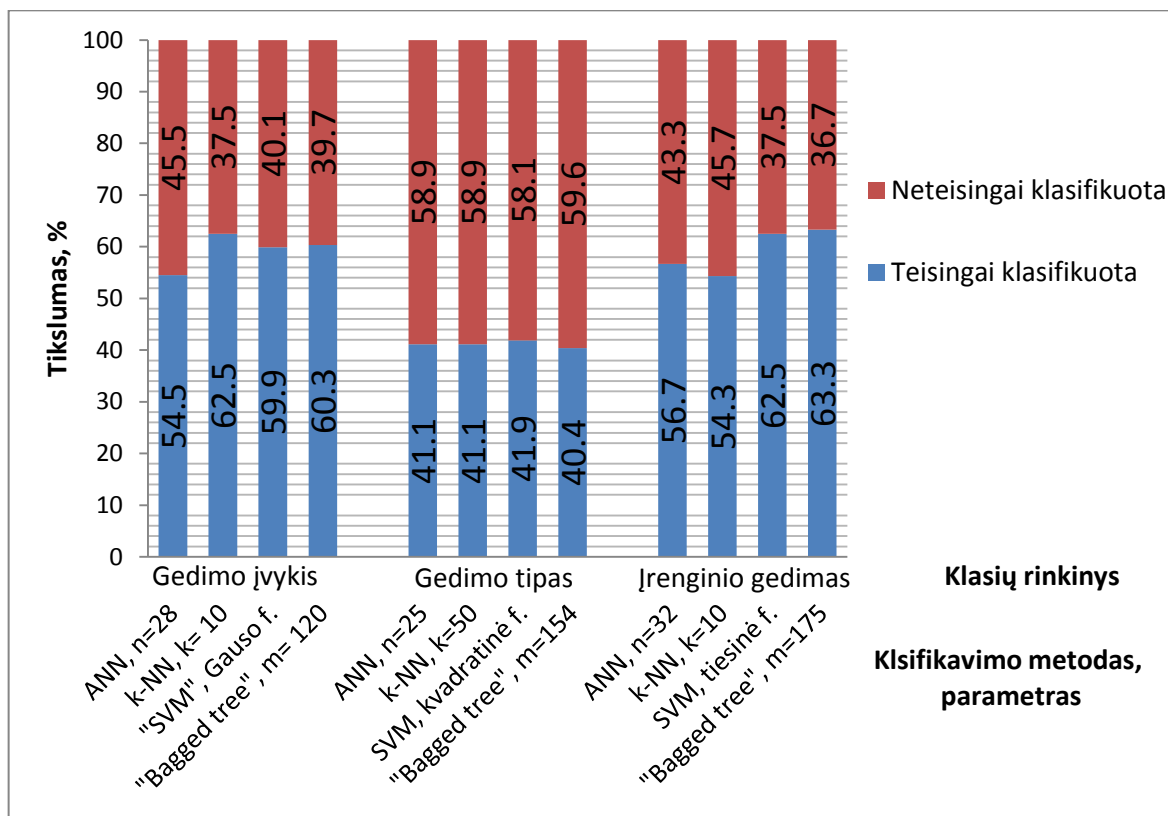
		ANN					k-NN						
Atpažinta klasė	1.Nėra gedimo	245 41.6%	22 3.7%	32 5.4%	27 4.6%	20 3.4%	70.8% 29.2%	1.Nėra gedimo	244 41.4%	1 0.2%	1 0.2%	0 0.0%	0 0.0%
	2. Butelių formavimo įreng.	1 0.2%	34 5.8%	24 4.1%	20 3.4%	18 3.1%	35.1% 64.9%	2. Butelių formavimo įreng.	35 5.9%	18 3.1%	10 1.7%	7 1.2%	16 2.7%
	3. Butelių išpilstymo įreng.	0 0.0%	11 1.9%	15 2.5%	10 1.7%	17 2.9%	28.3% 71.7%	3. Butelių išpilstymo įreng.	42 7.1%	12 2.0%	7 1.2%	10 1.7%	17 2.9%
	4. Etiketavimo įreng.	0 0.0%	2 0.3%	1 0.2%	7 1.2%	2 0.3%	58.3% 41.7%	4. Etiketavimo įreng.	36 6.1%	5 0.8%	6 1.0%	23 3.9%	9 1.5%
	5. Pakavimo įreng.	0 0.0%	17 2.9%	16 2.7%	15 2.5%	33 5.6%	40.7% 59.3%	5. Pakavimo įreng.	29 4.9%	14 2.4%	12 2.0%	7 1.2%	28 4.8%
			99.6% 0.4%	39.5% 60.5%	17.0% 83.0%	8.9% 91.1%	36.7% 63.3%	56.7% 43.3%					
		1.Nėra gedimo	2. Butelių formavimo įreng.	3. Butelių išpilstymo įreng.	4. Etiketavimo įreng.	5. Pakavimo įreng.		1.Nėra gedimo	2. Butelių formavimo įreng.	3. Butelių išpilstymo įreng.	4. Etiketavimo įreng.	5. Pakavimo įreng.	
		Tiksli klasė						Atpažinta klasė					
		SVM					„Bagged tree“						
Tiksli klasė	1.Nėra gedimo	246 41.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1.Nėra gedimo	246 41.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	
	2. Butelių formavimo įreng.	0 0.0%	13 2.2%	20 3.4%	24 4.1%	29 4.9%	2. Butelių formavimo įreng.	0 0.0%	24 4.1%	24 4.1%	20 3.4%	18 3.1%	
	3. Butelių išpilstymo įreng.	3 0.5%	9 1.5%	19 3.2%	30 5.1%	27 4.6%	3. Butelių išpilstymo įreng.	1 0.2%	24 4.1%	23 3.9%	18 3.1%	22 3.7%	
	4. Etiketavimo įreng.	0 0.0%	5 0.8%	6 1.0%	50 8.5%	18 3.1%	4. Etiketavimo įreng.	0 0.0%	15 2.5%	8 1.4%	37 6.3%	19 3.2%	
	5. Pakavimo įreng.	1 0.2%	9 1.5%	20 3.4%	20 3.4%	40 6.8%	5. Pakavimo įreng.	1 0.2%	15 2.5%	17 2.9%	14 2.4%	43 7.3%	
			1.Nėra gedimo	2. Butelių formavimo įreng.	3. Butelių išpilstymo įreng.	4. Etiketavimo įreng.	5. Pakavimo įreng.		1.Nėra gedimo	2. Butelių formavimo įreng.	3. Butelių išpilstymo įreng.	4. Etiketavimo įreng.	5. Pakavimo įreng.
		Atpažinta klasė						Atpažinta klasė					

5.27 pav. Klasifikavimo tikslumo matricos, klasifikuojant įrenginių gedimus

Gautos įrenginių gedimų klasifikavimo matricos rodo, jog pavyko tiksliau atpažinti atskiras klases nei 5.2 dalyje atliktame tyrime. Priešingai, nei 5.2 dalyje dirbtinio neuroninio tinklo klasifikatorius parodė žemiausią rezultatą 54,5 proc. bei iš kitų klasifikatorių išsiskyrė prastesniu „Nėra gedimo“ klasės atpažinimu – 70,8 proc., kiti klasifikatoriai šią klasę atpažino 99 – 100 proc. tikslumu. Tiksliausias rezultatas gautas naudojant „Bagged tree“ metodą – 63.3 proc., tačiau kiekvienas įrenginio gedimas atpažįstamas tik 37 proc. vidutiniu tikslumu (127 tikslūs iš 343). SVM ir k-NN klasifikatoriais gauti žemesni rezultatai – 54 ir 62 proc. tikslumas.



Pastebima, jog visais klasifikatoriais pavyko pasiekti proporcingesnę kiekvienos įrenginio gedimo klasės nustatymą – mažesnę klaidingo klasifikavimo įverčio skirtumą (lyginant su 5.2 dalies tyrimu), tačiau atpažįstama mažiau nei pusę šių klasių. Bendras visų realizuotų klasifikatorių tiksliausias rezultatas ir naudojamas parametras pateikias 5.28 paveiksle.



5.28 pav. Realizuotų klasifikavimo metodų tikslumo palyginimas

#### 5.4. Klasifikavimo metodų tikslumo įvertinimas su mokyme nedalyvaujančiais požymiais

Papildomas realizuotų klasifikatorių įvertinimas atliekamas naudojant 2016 metais sukauptus požymių bei klasių duomenų rinkinius. Duomenų rinkiniai apdorojami analogiškai kaip 5.2 dalyje. Klasių su 2016 metų požymių rinkiniais pasisiskirstymas pateiktas 5.2 lentelėje.

Tikslumo įvertinimui pasirenkami realizuoti klasifikatoriai, kuriais gautas geriausias rezultatas – pasiekė arba viršijo 60 proc. tikslumą bei mažiausiu tikslumo skirtumu atpažino kiekvieną klasę:

- k-NN klasifikatorius, gedimo įvykis – 62.5 proc. tikslumas;
- ANN klasifikatorius, gedimo įvykis – 69 proc. (realizuota pagal 2015 m. duomenis);
- SVM klasifikatorius, įrenginio gedimas – 62.5 proc. tikslumas;
- Sprendimo medžio klasifikatorius „Bagged tree“, įrenginio gedimas – 63.3 proc. tikslumas.

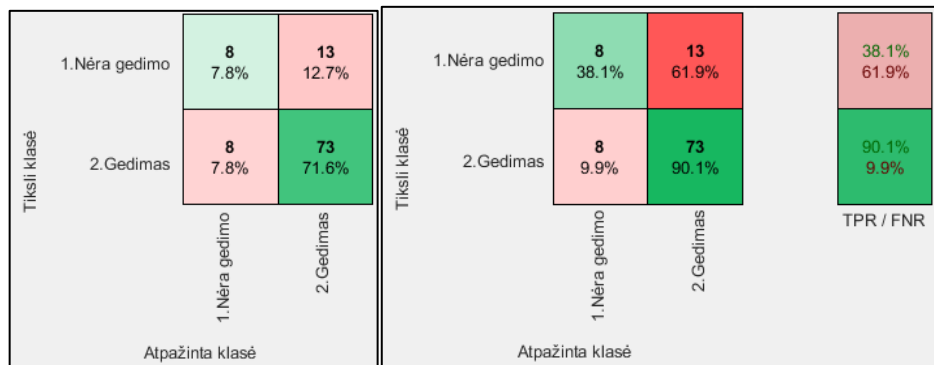
5.2 lentelė. Klasių pasiskirstymas su 2016 m. sukauptais požymiais

Klasių rinkinys	Klasė	Kiekis
Gedimo įvykis	Nėra gedimo	21
	Gedimas	81
Įrenginio gedimas	PET butelių formavimo įrenginys;	12
	Pilstymo įrenginys;	23
	Etikečių klijavimo, žymėjimo įreng.	27
	Pakuočių – palečių formavimo įreng.	13

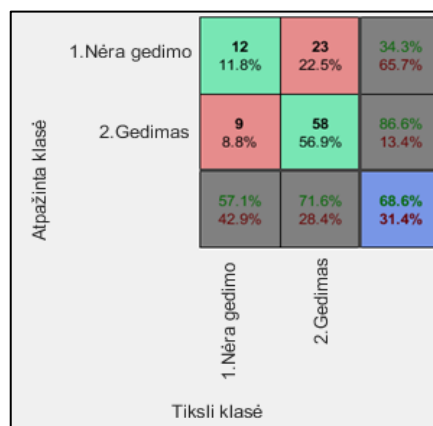
Naudojami realizuotų k-NN, SVM, „Bagged tree“ klasifikatorių modeliai (4 priedas) suformuoti 5.3 dalyje ir ANN klasifikatoriaus funkcija (1 priedas) suformuota 5.2 dalyje, kuriais gaunamos klasifikavimo tikslumo matricos bei ROC kreivės.

Klasifikuojant gedimo įvykį ANN modeliu bendras tikslumas – 68,6 proc. (5.30 pav.), k-NN modeliu – 79,4 proc (5.29 pav.). Tikslumo matricos rodo, jog tiksliau atpažįstama klasė „Gedimas“.

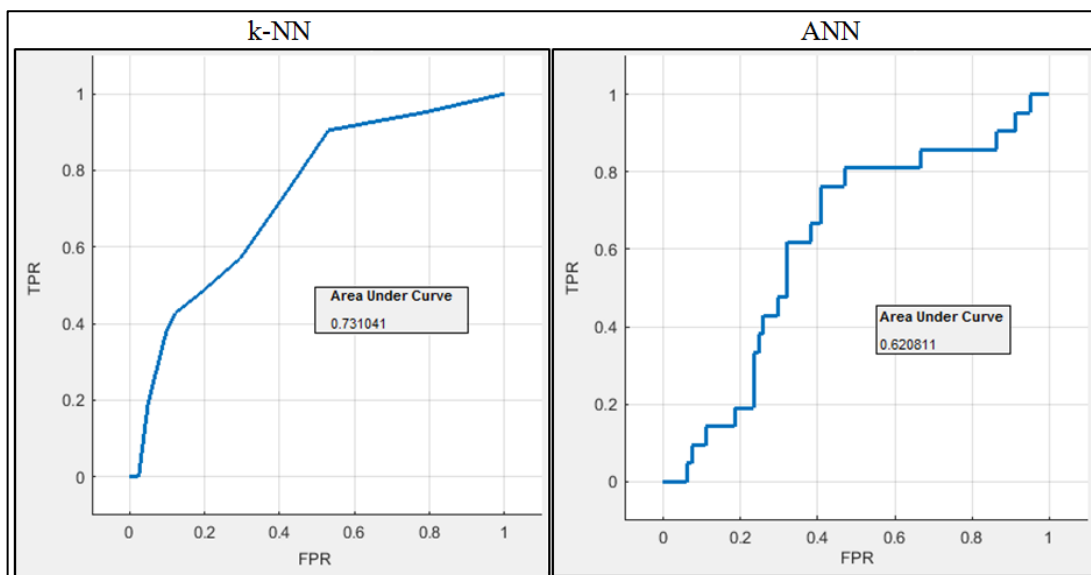
Artimiausio kaimyno klasifikavimo metodu pranašesnis kiekvienos klasės atpažinimo tikslumas bei jautrumo rezultatas (5.31 pav.). Plotas po ROC krivėmis: k-NN – 0,73; ANN – 0,62. Klasifikatoriaus k-NN tikslumo matricos rodo, jog tiksliau atpažįstama klasė „Gedimas“ – 90,1 proc. tikslumu, nei klasė „Nėra gedimo“ – 38,1 proc.



5.29 pav. k-NN gedimo įvykio klasifikavimo tikslumo matricos



5.30 pav. ANN gedimo įvykio klasifikavimo tikslumo matricos



5.31 pav. k-NN ir ANN gedimo įvykio klasifikavimo ROC kreivės

Klasifikuojant įrenginių gedimus atraminio vektoriaus mašinos ir sprendimo medžio klasifikatoriais gauti panašūs bendri tikslumo bei kiekvienos klasės tikslumo atpažinimo rezultatai. Bendro tikslumo skirtumas tik 1 proc.: SVM – 53,9 proc.; „Bagged tree“ – 54,9 proc. (5.32 pav.). Tiksliausiai atpažįstama klasė „Nėra gedimo“ – apie 90 proc. tikslumas, praščiausiai atpažįstamas butelių formavimo įrenginio gedimas – 25 proc. (5.33 pav.). Šiam tyrimui pasirinktas gamybos objektas, kuriam reikalingas didesnės tikimybės „Gedimas“ klasės prognozavimas, taigi tiksinga naudoti k-NN klasifikatorių gedimo įvykui atpažinti (5.29 pav.).

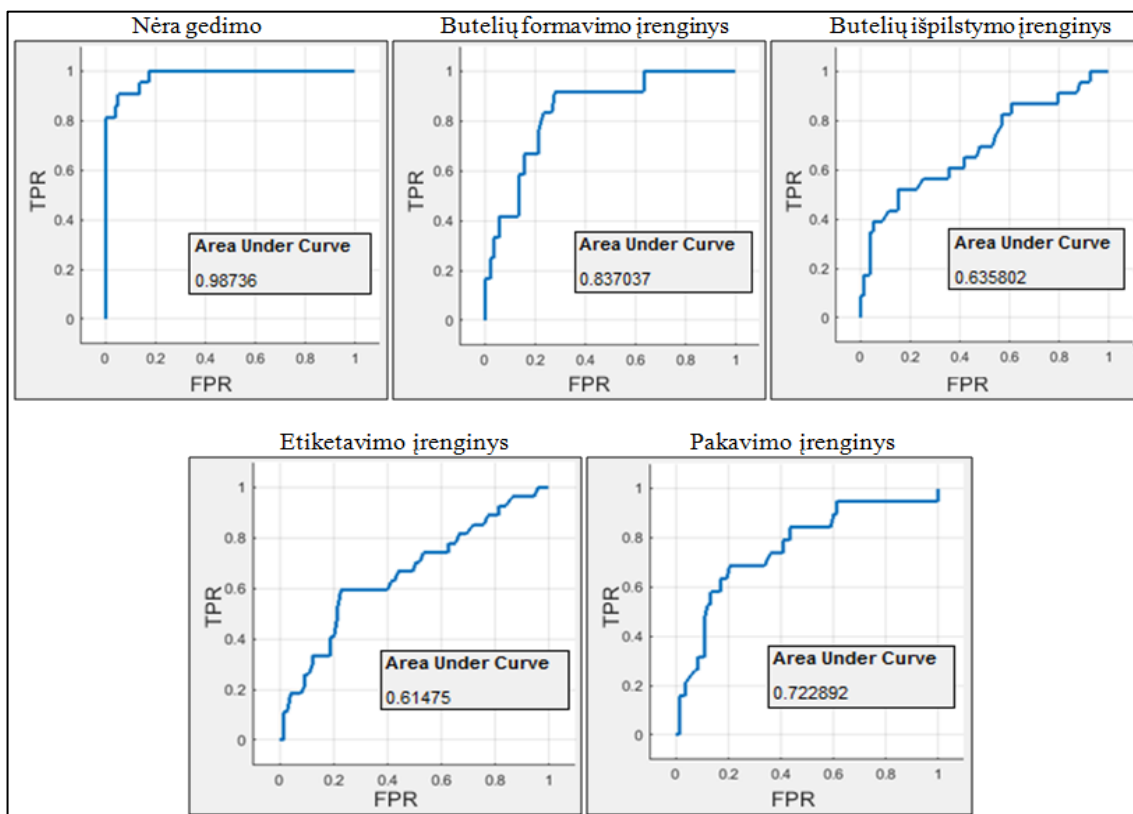
		SVM					„Bagged tree“				
		18 17.6%	0 0.0%	2 2.0%	1 1.0%	0 0.0%	19 18.6%	0 0.0%	1 1.0%	1 1.0%	0 0.0%
Tiksli klasė	1. Nėra gedimo	1 1.0%	3 2.9%	4 3.9%	2 2.0%	2 2.0%	1 1.0%	3 2.9%	5 4.9%	2 2.0%	1 1.0%
	2. Butelių formavimo įreng.	1 1.0%	1 1.0%	12 11.8%	6 5.9%	3 2.9%	1 1.0%	1 1.0%	12 11.8%	6 5.9%	3 2.9%
	3. Butelių išpilstymo įreng.	2 2.0%	1 1.0%	4 3.9%	14 13.7%	6 5.9%	2 2.0%	1 1.0%	6 5.9%	13 12.7%	5 4.9%
	4. Etiketavimo įreng.	3 2.9%	1 1.0%	1 1.0%	6 5.9%	8 7.8%	2 2.0%	1 1.0%	1 1.0%	6 5.9%	9 8.8%
	5. Pakavimo įreng.										
		1. Nėra gedimo	2. Butelių formavimo įreng.	3. Butelių išpilstymo įreng.	4. Etiketavimo įreng.	5. Pakavimo įreng.	1. Nėra gedimo	2. Butelių formavimo įreng.	3. Butelių išpilstymo įreng.	4. Etiketavimo įreng.	5. Pakavimo įreng.
		Atpažinta klasė					Atpažinta klasė				

5.32 pav. SVM ir „Bagged tree“ įrenginių gedimų klasifikavimo bendra tikslumo matrica

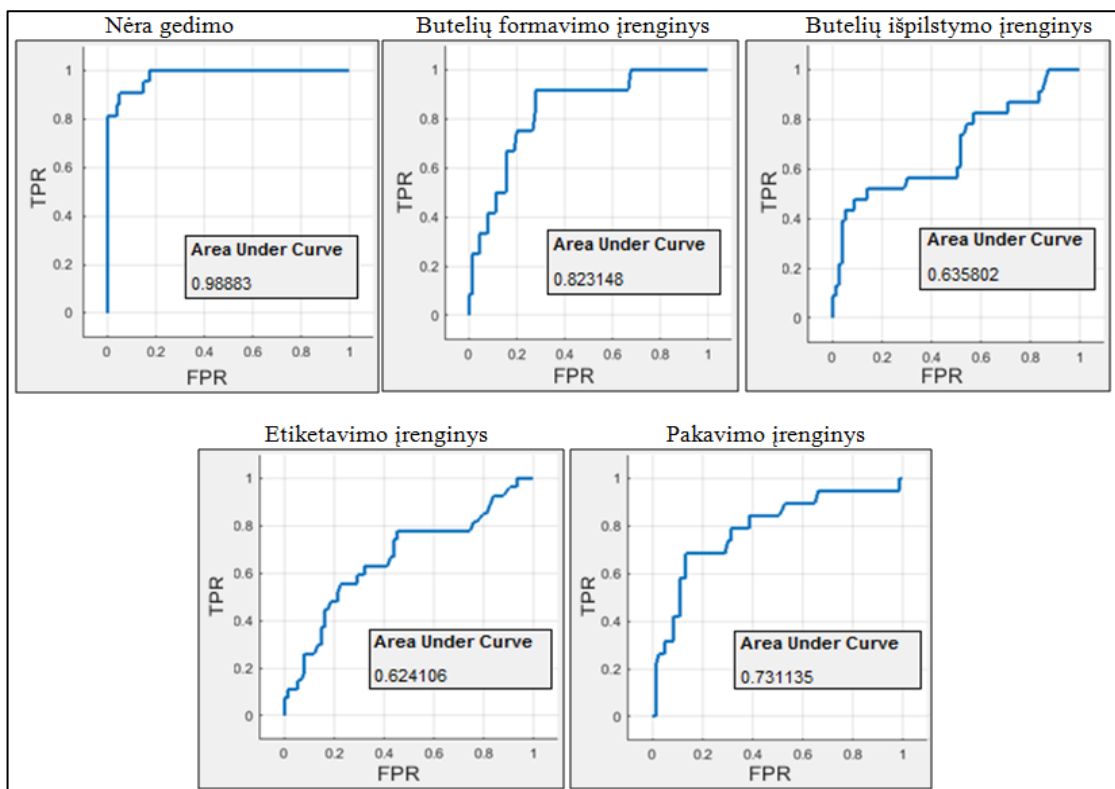
		SVM					„Bagged tree“								
Tiksli klasė	1. Nėra gedimo	18 85.7%	0 0.0%	2 9.5%	1 4.8%	0 0.0%	85.7%	14.3%	19 90.5%	0 0.0%	1 4.8%	1 4.8%	0 0.0%	90.5%	9.5%
	2. Butelių formavimo įreng.	1 8.3%	3 25.0%	4 33.3%	2 16.7%	2 16.7%	25.0%	75.0%	1 8.3%	3 25.0%	5 41.7%	2 16.7%	1 8.3%	25.0%	75.0%
	3. Butelių išpilstymo įreng.	1 4.3%	1 4.3%	12 52.2%	6 26.1%	3 13.0%	52.2%	47.8%	1 4.3%	1 4.3%	12 52.2%	6 26.1%	3 13.0%	52.2%	47.8%
	4. Etiketavimo įreng.	2 7.4%	1 3.7%	4 14.8%	14 51.9%	6 22.2%	51.9%	48.1%	2 7.4%	1 3.7%	6 22.2%	13 48.1%	5 18.5%	48.1%	51.9%
	5. Pakavimo įreng.	3 15.8%	1 5.3%	1 5.3%	6 31.6%	8 42.1%	42.1%	57.9%	2 10.5%	1 5.3%	1 5.3%	6 31.6%	9 47.4%	47.4%	52.6%
		1. Nėra gedimo	2. Butelių formavimo įreng.	3. Butelių išpilstymo įreng.	4. Etiketavimo įreng.	5. Pakavimo įreng.	TPR / FNR		1. Nėra gedimo	2. Butelių formavimo įreng.	3. Butelių išpilstymo įreng.	4. Etiketavimo įreng.	5. Pakavimo įreng.	TPR / FNR	
		Atpažinta klasė					Atpažinta klasė								

5.33 pav. SVM ir „Bagged tree“ įrenginių gedimų klasifikavimo TPR/FNR tikslumo matrica

Atraminio vektoriaus mašinos ir sprendimo medžio „Bagged tree“ klasifikatorių ROC kreivės (5.34 pav. ir 5.35 pav.) rodo jog, „jautriausios“ klasės „Etiketavimo įrenginys“ ir „Butelių išpilstymo įrenginys“, kadangi gaunamas mažiausias plotas po ROC kreive – nuo 0,61 iki 0,64. Mažiausiai „jautrus“ šių klasifikatorių prognozavimas, jog gedimas neįvyks – 0,99. Taip pat žemas prognozavimo jautrumas, kad įvyks butelių formavimo įrenginio gedimas – 0,82 ir 0,84, tačiau prognozavimo tikslumas mažiausias (5.33 pav.).

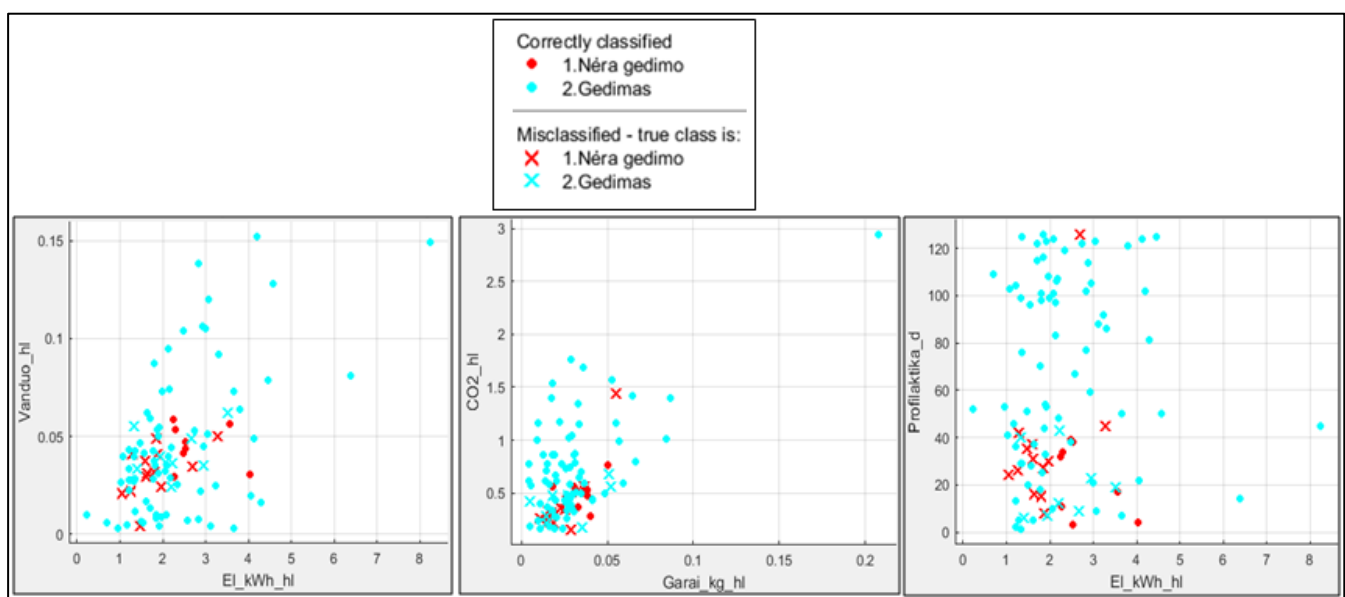


5.34 pav. SVM įrenginių gedimų klasifikavimo ROC kreivės



5.35 pav. „Bagged tree“ įrenginių gedimų klasifikavimo ROC kreivės

5.2 ir 5.3 dalyje atliktus klasifikatorių realizavimą, pastebima jog, klasifikavimo mokymui naudoti požymiai neturi „ryškių“ tarpusavio skirtumų klasifikuojant gedimo tipus, bei įrenginių gedimus. Tačiau gedimo įvykiui atpažinti labiausiai pastebimas klasių skirtumas pagal elektros, garų, anglies dioksido, vandens ir likusio laiko iki profilaktinės priežiūros požymius. Atvaizduojami k-NN klasifikatoriaus gauti rezultatai (įvertininti su 2016 m. duomenimis) pagal požymių išidėstymą – tiksliai ir klaidingai klasifikuotos klasės pagal požymius(5.36 pav.).



5.36 pav. k-NN tiksliai ir klaidingai klasifikuotos gedimo įvykio klasės pagal požymius

## REZULTATAI IR IŠVADOS

1. Išanalizavus įmonėje matuojamus energetinius, įrenginių priežiūros, efektyvumo rodiklius – parinkti duomenys matuojami prietaisais, siekiant išvengti klaidingų duomenų (iš 2014 – 2016 m. sukauptų duomenų): elektros, garų, anglies dioksido, vandens sunaudojimas, vidutinis pilstymo greitis, profilaktikos atlikimo laikas, prastovų dėl gedimų laikas.
2. Parenkami ANN, k-NN, SVM, „Bagged tree“ klasifikatoriai remiantis atlikta [6] analize. Klasifikavimo metodų parametrai parenkami pagal gaunamą didžiausią prognozavimo tikslumą ir mažiausią kiekvienos klasės (gedimo, gedimo tipo, įrenginio gedimo klaidingo klasifikavimo įvertį) atpažinimo skirtumą.
3. Tiksliausi rezultatai gauti naudojant:  $k=10$  prognozuojant gedimo įvykį KNN metodu,  $n=12$  prognozuojant gedimo įvykį ANN metodu, tiesine branduolio funkcija prognozuojant įrenginio gedimą SVM metodu,  $m=175$  prognozuojant įrenginio gedimą „Bagged tree“ metodu.
4. Naudojant 2015 m. požymius realizuotiems klasifikatoriams (ANN, k-NN, SVM, „Bagged tree“) nepavyko atpažinti pavienių klasių, kurių imtis mokymo proceso metu buvo mažiausia. Realizuojant klasifikatorius gedimų nustatymui tikslinga naudoti proporcingą klasių imčių skaičių. Tiksliausiai įrenginių gedimai prognozuojami sprendimo medžio „Bagged tree“ klasifikatoriumi – 63,3 proc., panašus rezultatas gaunamas SVM klasifikatoriumi – 62,5 proc., (pagal 2014 – 2015 m. duomenis). Gedimo tipo prognozavimo atlikti nepavyko – gaunamas tikslumas iki 50 proc.
5. Įvertintos prognozavimo galimybės pagal 2016 metais sukauptus duomenis:
  - Gedimo įvykis visai linijai prognozuojamas 79,4 proc. tikslumu naudojant artimiausio kaimyno klasifikavimo metodą. Atsižvelgiant į tai, jog analizuojamas linijinės gamybos objektas, gedimo įvykio prognozavimas leidžia apie 20 proc. paklaidos tikslumu numatyti gamybos linijos stabdymą dėl pasikartojančių gedimų bei remiantis prognozės rezultatais koreguoti profilaktinės priežiūros grafiką.
  - Įrenginių gedimas prognozuojamas nepatikimai – 53,9 proc. tikslumu naudojant SVM klasifikavimo metodą ir 54,9 proc. „Bagged tree“ klasifikavimo metodu.
6. Tolimesniam gedimų prognozavimo tyrimui reikia keisti galimas nepatikimo prognozavimo priežastis – per maža (naudoti 589 įrašai) duomenų imtis klasifikatorių realizavimui, per didelis duomenų fiksavimo intervalas (24 h.), netinkamai apdorotos požymių reikšmės.

## INFORMACIJOS ŠALTINIŲ SĄRAŠAS

1. Algirdas Bargelis, Kazimieras Jurėnas Rasa Mankutė, Marius Rimašauskas, Gamybos procesų modeliavimas. Mokomoji knyga, Kaunas 2012.-157 p., ISBN 978-609-433-095-7.
2. Jolanta Janutienė, Inžinerinių sistemų patikimumas. Mokomoji knyga Vilnius 2008.-134 p., ISBN 978-9955-20-257-8.
3. Bronius Tamulynas, Intelektualiųjų sistemų modeliavimas. Kaunas, Technologija 2008.- 275 p., ISBN 978-995-525-599-4.
4. Rolandas Bartkus, Arūnas Bingelis, Technikos antropologija: technologiai žmogaus tęsiniai kaip medijos. Kaunas, Technologija 2012. – 48 p., ISBN 978-609-02-559-4.
5. Šarūnas Raudys, Žinių išgavimas iš duomenų. Klaipėdos universitetas, 2009.-172 psl.
6. Nuria Macia and Ester Bernado-Mansilla, 2014. Straipsnis – Towards UCI A mindful repository design. [Žiūrėta 2015-12-02]. Prieiga per internetą:  
<<http://www.sciencedirect.com/science/article/pii/S0020025513006336>>
7. Joaquin Vanschoren, Hendrik Blockeel, Bernhard Pfahringer, and Georey Holmes, 2014. Straipsnis – A new way to share, organize and learn from experiments. Machine Learning. [Žiūrėta 2015-12-07]. Prieiga per internetą:  
<<http://link.springer.com/article/10.1007%2Fs10994-011-5277-0>>
8. UCI Machine learning – duomenų archyvas klasifikavimo mokymui. [Žiūrėta 2015-12-02]. Prieiga per internetą: <<http://archive.ics.uci.edu/ml/index.html>>
9. Manuel Fernandez-Delgado, Eva Cernadas, Senen Barro, Dinani Amorim, 2014. Straipsnis – Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?, Journal of Machine Learning Research 16, [Žiūrėta 2015-12-15]. Prieiga per internetą:  
<<http://jmlr.csail.mit.edu/papers/volume15/delgado14a/delgado14a.pdf>>
10. Hyonho Chun and Sunduz Keles, 2010. Straipsnis – Sparse partial least squares for simultaneous dimension reduction and variable selection, [Žiūrėta 2015-12-20]. Prieiga per internetą:  
<<http://onlinelibrary.wiley.com/doi/10.1111/j.1467-9868.2009.00723.x/abstract>>
11. Gordon S. Linoff, Michael J. A. Berry, Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management 2011.-225p. ISBN 976-047-65093.6
12. Tammo Krueger, Danny Panknin, Mikio BraunManuel Fernandez-Delgado, Eva Cernadas, Senen Barro, Dinani Amorim, 2015. Straipsnis – Fast Cross-Validation via Sequential Testing, Journal of Machine Learning Research 16, [Žiūrėta 2015-11-18]. Prieiga per internetą:  
<[www.jmlr.org/papers/volume16/krueger15a/krueger15a.pdf](http://www.jmlr.org/papers/volume16/krueger15a/krueger15a.pdf)>



13. V. Mohan, Manas Rajan, 2015. Straipsnis – A Novel Approach to Compute Confusion Matrix for Classification of n-Class Attributes. [Žiūrėta 2015-11-05]. Prieiga per internetą: <[www.scholarpublishing.org/index.php/TMLAI/article/download/1108/pdf\\_28](http://www.scholarpublishing.org/index.php/TMLAI/article/download/1108/pdf_28)>
14. Šimkevičius, Simonas, 2016. Klasifikavimo su mokytoju lyginamoji analizė, Magistro darbas. 101 p. [Žiūrėta 2016-02-25]. Prieiga per internetą:<[http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2006~D\\_20060605\\_092832-93331/DS.005.0.02.ETD.](http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2006~D_20060605_092832-93331/DS.005.0.02.ETD.)>
15. Jese Davis, Mark Goadrich, University of Wisconsin-Madison, 2006. Straipsnis – The relationship between Precision and ROC curves. [Žiūrėta 2016-01-29]. Prieiga per internetą: <<http://dl.acm.org/citation.cfm?id=1143874>>
16. Karimollah Hajian-Tilaki, Department of Social Medicine and Health, Babol University of Medical Sciences 2013. Straipsnis – Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation. [Žiūrėta 2016-01-22]. Prieiga per internetą: <[http://www.caspjim.com/files/site1/user\\_files\\_6942f6/archive-A-10-62-133-d55696e.pdf](http://www.caspjim.com/files/site1/user_files_6942f6/archive-A-10-62-133-d55696e.pdf)>
17. Verikas A., Gelžinis A., Neuroniniai tinklai ir neuroniniai skaičiavimai. Kaunas, 2008, ISBN 9789955-591-53-5.
18. Martin Anthony, Peter L. Bartlett, Neural Network Learning: Theoretical Foundations. Cambridge University Press 2009 – 389 p., ISBN 978-0-521-57353-5.
19. Riccardo Taormina Kwok-wing Chau Rajandrea Sethi, Engineering Applications of Artificial Intelligence, 2012, Straipsnis – Artificial Neural Network simulation of hourly groundwater levels in a coastal aquifer system of the Venice lagoon. [Žiūrėta 2016-01-15]. Prieiga per internetą: <[repository.lib.polyu.edu.hk/jspui/bitstream/10397/5259/1/EAAI7.pdf](http://repository.lib.polyu.edu.hk/jspui/bitstream/10397/5259/1/EAAI7.pdf)>
20. Alex Berson, Stephen Smith, and Kurt Thearling, An Overview of Data Mining Techniques [Žiūrėta 2016-01-16]. Prieiga per internetą: <<http://www.thearling.com/text/dmtechniques/dmtechniques.htm>>
21. Kilian Q. Weinberger, Lawrence K. Saul, Journal of Machine Learning Research 10, 2009. Straipsnis – Distance Metric Learning for Large Margin Nearest Neighbor Classification. [Žiūrėta 2016-01-21]. Prieiga per internetą: <[machinelearning.wustl.edu/mlpapers/paper\\_files/jmlr10\\_weinberger09a.pdf](http://machinelearning.wustl.edu/mlpapers/paper_files/jmlr10_weinberger09a.pdf)>
22. David Meyer, FH Technikum Wien, Austria, 2015. Straipsnis – Support vector machines. [Žiūrėta 2016-01-29]. Prieiga per internetą: <[ftp://ftp.cse.yzu.edu.tw/CRAN/web/packages/e1071/vignettes/svmdoc.pdf](http://ftp.cse.yzu.edu.tw/CRAN/web/packages/e1071/vignettes/svmdoc.pdf)>
23. Marc Claesen Journal, Frank De Smet, Johan A.K. Suykens, Bart De Moor Journal of Machine Learning Research 15, 2014. Straipsnis – A Library for Ensemble Learning Using Support Vector Machines. [Žiūrėta 2016-03-06]. Prieiga per internetą: <[www.jmlr.org/papers/volume15/claesen14a/claesen14a.pdf](http://www.jmlr.org/papers/volume15/claesen14a/claesen14a.pdf)>



24. Varoneckas, Audrius. 2011. Hibridinis RR intervalų sekų modelis miego stadijoms atpažinti. Daktaro disertacija. 125 p. [Žiūrėta 2016-01-25]. Prieiga per internetą:  
<[http://elaba.lvb.lt/primo\\_library/libweb/action/search.do?vid=ELABA](http://elaba.lvb.lt/primo_library/libweb/action/search.do?vid=ELABA)>
25. Norberto A., Goussies Sebastian Ubalde, Marta Mejail Transfer, Journal of Machine Learning Research 15, 2014. Straipsnis – Transfer Learning Decision Forests for Gesture Recognition. [Žiūrėta 2015-12-01]. Prieiga per internetą:  
< [www.jmlr.org/papers/volume15/claesen14a/claesen14a.pdf](http://www.jmlr.org/papers/volume15/claesen14a/claesen14a.pdf)>
26. Albertas Gimbutas Gražina Laurinavičiute, Klaidų kainoms jautrūs klasifikavimo algoritmai. Magistro baigiamasis darbas, Vilniaus universitetas 2013. [Žiūrėta 2016-02-29]. Prieiga per internetą:  
<[http://elaba.lvb.lt/primo\\_library/libweb/action/search.do?vid=ELABA](http://elaba.lvb.lt/primo_library/libweb/action/search.do?vid=ELABA)>
27. Mark Hudson Beale , Martin T. Hagan , Howard B. Demuth, 2012. Neural Network Toolbox. [Žiūrėta 2015-03-15]. Prieiga per internetą:  
< <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.699.4831>>
28. Matlab Classification Learner, [Žiūrėta 2015-03-10]. Prieiga per internetą:  
< <http://se.mathworks.com/help/stats/classificationlearner-app.html>>

## PRIEDAI

### Priedas 1. „Matlab“ programinis kodas neuroninio tinklo klasifikavimo mokymui, testavimui

```
% Duomenų įkėlimas neuroninio tinklo klasifikavimo apmokymui
% Požymiai (nurodomas aplankas)
InputM =
xlsread('C:\Users\Mantas\Desktop\Machine\InputM.csv', 'Sheet1');
InputM = reshape([raw{:}], size(raw));
% Išvalomos laikinos reikšmės
clearvars raw;
% Klases (nurodomas aplankas į klasių rinkinius TargF, TargT, TargM)
TargM =
xlsread('C:\Users\Mantas\Desktop\Machine\TargM.csv', 'Sheet1');
TargM = reshape([raw{:}], size(raw));
clearvars raw;

%%
=====
% Klasifikavimas neuroniniu tinklu
fileID = fopen('tikslumas.txt', 'w');

% Tikslumas pagal neuronų skaičių pasleptajame sluoksnyje n
for n = 10:5:200

x = InputM';
t = TargM';

% Tinklo apmokymo funkcijos pasirinkimas
trainFcn = 'trainlm';

% Sukuriamas atpazinimo tinklas.
hiddenLayerSize = n;
net = patternnet(hiddenLayerSize);

% Parengiami duomenys tinklo mokymui help nnprocess
net.input.processFcns = {'removeconstantrows', 'mapminmax'};
net.output.processFcns = {'removeconstantrows', 'mapminmax'};

% Padalinami duomenys mokymui, patvirtinimui, testavimui, help
nndivide

net.divideFcn = 'divideint'; % Dalinama pagal tenkanciu klasiu
kieki
net.divideFcn = 'dividerand'; % Dalinama atsitiktine tvarka

net.divideMode = 'sample'; % Dalinama kiekvienu paleidimu
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
```

```

% Psirenkama veikimo funkcija, help nnperformance
net.performFcn = 'crossentropy'; % Cross-Entropy
net.performFcn = 'mse'; % Mean squared error
% Pasirenkamos garafines funkcijos, help nnplot
net.plotFcns = {'plotperform', 'plottrainstate', 'ploterrhist', ...
    'plotconfusion', 'plotroc'};

% Tinklo apmokymas
[net,tr] = train(net,x,t);

% Tinklo testavimas
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);

%Perskaiciujamas mokymas, Patvirtinimas Testavimo atlikimas
trainTargets = t .* tr.trainMask{1};
valTargets = t .* tr.valMask{1};
testTargets = t .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

%view(net)
%Braizomi atitinkami grafikai
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotconfusion(t,y)
%figure, plotroc(t,y)

%Klasifikavimo tikslumas

[c,cm] = confusion(t,y)
fprintf('Percentage Correct Classification : %f%%\n', 100*(1-c));
fprintf('Percentage Incorrect Classification : %f%%\n', 100*c);

%irasomas neuronu skaicius ir gaunamas tikslumas
tiks = 100*(1-c);
A = [n,tiks];
fprintf(fileID, '%6.2f %12.8f\n',A);

end
fclose(fileID);

%=====

```

```

% Funkcijos generavimas nauju duomenu testavimui. False = True

if (false)

    genFunction(net, 'myNeuralNetworkFunction');
    y = myNeuralNetworkFunction(x);
end
% matrix funkcijos generavimas
if (false)
    % matrix funkcijos generavimas
    genFunction(net, 'myNeuralNetworkFunction', 'MatrixOnly', 'yes');
    y = myNeuralNetworkFunction(x);
end

%%
=====

%%
=====
% neroroninio tinklo apmokymas atitinkamas naudojant maksimalius n
(histogramos duomenys
)
fileID = fopen('tikslumasN.txt', 'w');

% Tikslumas pagal neuronu skaiciu pasleptajame sluoksnyje n
for k = 0:1:1000

k = k+1;

x = InputM';
t = TargM';

% Tinklo apmokymo funkcijos pasirinkimas
trainFcn = 'trainlm';

% Sukuriamas atpazinimo tinklas.
%Naudojama n reiksme atitinkamam klasiu rinkiniui

hiddenLayerSize = 12;
net = patternnet(hiddenLayerSize);

% Parengiami duomenys tinklo mokymui help nnprocess
net.input.processFcns = {'removeconstantrows', 'mapminmax'};
net.output.processFcns = {'removeconstantrows', 'mapminmax'};

% Padalinami duomenys mokymui, patvirtinimui, testavimui, help
nndivide

net.divideFcn = 'divideint'; % Dalinama pagal tenkanciu klasiu
kieki
net.divideFcn = 'dividerand'; % Dalinama atsitiktine tvarka

```

```

net.divideMode = 'sample'; % Dalinama kiekvienu paleidimu
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Psirenkama veikimo funkcija, help nnperformance
net.performFcn = 'crossentropy'; % Cross-Entropy
net.performFcn = 'mse'; % Mean squared error
% Pasirenkamos garafines funkcijos, help nnplot
net.plotFcns = {'plotperform', 'plottrainstate', 'ploterrhist', ...
               'plotconfusion', 'plotroc'};

% Tinklo apmokymas
[net,tr] = train(net,x,t);

% Tinklo testavimas
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);

%Perskaiciujamas mokymas, Patvirtinimas Testavimo atlikimas
trainTargets = t .* tr.trainMask{1};
valTargets = t .* tr.valMask{1};
testTargets = t .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

%Klasifikavimo tikslumas

[c,cm] = confusion(t,y)
fprintf('Percentage Correct Classification : %f%%\n', 100*(1-c));
fprintf('Percentage Incorrect Classification : %f%%\n', 100*c);

%irasomas apmokymo numeris ir gaunamas tikslumas
tiksn = 100*(1-c);
B = [k,tiksn];
fprintf(fileID, '%6.2f %12.8f\n',B);

end
fclose(fileID);
%%
=====

```

## Priedas 2. „Matlab“ programinis kodas k-NN, SVM, „Bagged tree“ klasifikavimo mokymui, testavimui

```
% Duomenu ikelimas klasifikavimo apmokymui
% Sudaromas duomenu lenteles dydis
[~, ~, raw] =
xlsread('C:\Users\Mantas\Desktop\MatlabM\InputTargMTF.csv', 'Sheet1')
;
raw(cellfun(@(x) ~isempty(x) && isnumeric(x) && isnan(x), raw)) =
{' '};
cellVectors = raw(:, [12, 14, 16]);
raw = raw(:, [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 15]);

%% Sukuriama isvedami kintamieji
data = reshape([raw{:}], size(raw));

%% Sukuriama lentele
InputTargMTF = table;

%% Priskiriammi ikeltu stulpeliu pvadinimai
InputTargMTF.El_kWh_hl = data(:, 1);
InputTargMTF.Garai_kg_hl = data(:, 2);
InputTargMTF.Vanduo_hl = data(:, 3);
InputTargMTF.CO2_hl = data(:, 4);
InputTargMTF.Greitis_but_h = data(:, 5);
InputTargMTF.Profilaktika_d = data(:, 6);
InputTargMTF.Ispilta_hl = data(:, 7);
InputTargMTF.Elektr_min = data(:, 8);
InputTargMTF.Mech_min = data(:, 9);
InputTargMTF.Dernim_min = data(:, 10);
InputTargMTF.IrenginioGedimasNr = data(:, 11);
InputTargMTF.IrenginioGedimas = cellVectors(:, 1);
InputTargMTF.GedimasNr = data(:, 12);
InputTargMTF.Gedimas = cellVectors(:, 2);
InputTargMTF.GedimoTipasNr = data(:, 13);
InputTargMTF.GedimoTipas = cellVectors(:, 3);

%% Isvalomos laikinos reiksmes
clearvars data raw cellVectors;

%%
=====

% Klasifikavimas Artimiausio kaimymo metodu (KNN)

% KNN apmokymas trainKNN
[trainedClassifier, validationAccuracy] = trainKNN(InputTargMTF)

% Tikslumo ivertinimas naudojant naujus domenis apmokytam KNN
klasifikatoriui, kreipiamasi i klasifikatoriaus modeli
```

```

yfit = predict(trainedKNN,
InputTargMTF2016{:,trainedKNN.PredictorNames})

%%
=====

% Klasifikavimas Atraminio vektoriaus metodu (SVM)

% SVM apmokymas trainSSM
[trainedClassifier, validationAccuracy] = trainSVM(InputTargMTF)

% Tikslumo ivertinimas naudojant naujus domenis apmokytam SVM
klasifikatoriui, kreipiamasi i klasifikatoriaus modeli
yfit = predict(trainedSVM,
InputTargMTF2016{:,trainedSVM.PredictorNames})

%%
=====

% Klasifikavimas Bagged tree metodu
[trainedClassifier, validationAccuracy] =
BaggedTreetrain(InputTargMTF)

% Tikslumo ivertinimas naudojant naujus domenis apmokytam Bagged
tree klasifikatoriui, kreipiamasi i klasifikatoriaus modeli
yfit = predict(trainedBaggedTree,
InputTargMTF2016{:,trainedBaggedTree.PredictorNames})

%%
=====

```

### Priedas 3. „Matlab“ funkcijos k-NN, SVM, „Bagged tree“ klasifikavimo mokymui

Funkcija k-NN klasifikatraus mokymui:

```
function [trainedClassifier, validationAccuracy] =
trainClassifier(datasetTable)
% Isskleidziami pozymiai ir klases
predictorNames = {'El_kWh_hl', 'Garai_kg_hl', 'Vanduo_hl', 'CO2_hl',
'Greitis_but_h', 'Profilaktika_d', 'Ispilta_hl', 'Elektr_min',
'Mech_min', 'Dernim_min'};
predictors = datasetTable(:,predictorNames);
predictors = table2array(varfun(@double, predictors));
response = datasetTable.Gedimas;

% Apmokomas klasifikatorius
trainedClassifier = fitcknn(predictors, response, 'PredictorNames',
{'El_kWh_hl' 'Garai_kg_hl' 'Vanduo_hl' 'CO2_hl' 'Greitis_but_h'
'Profilaktika_d' 'Ispilta_hl' 'Elektr_min' 'Mech_min' 'Dernim_min'},
'ResponseName', 'Gedimas', 'ClassNames', {'1.N?ra gedimo'
'2.Gedimas'}, 'Distance', 'Euclidean', 'Exponent', '',
'NumNeighbors', 10, 'DistanceWeight', 'Equal', 'StandardizeData',
1);

% Atliekamas kryzminis validavimas
partitionedModel = crossval(trainedClassifier, 'Kfold', 5);

% Skaiciuojamas validavimo tikslumas
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun',
'ClassifError');

%% Skaiciuojami rezultatai
[validationPredictions, validationScores] =
kfoldPredict(partitionedModel);

%=====
```

Funkcija SVM klasifikatraus mokymui:

```
function [trainedClassifier, validationAccuracy] =
trainSVM_F(datasetTable)
% Isskleidziami pozymiai ir klases
predictorNames = {'El_kWh_hl', 'Garai_kg_hl', 'Vanduo_hl', 'CO2_hl',
'Greitis_but_h', 'Profilaktika_d', 'Ispilta_hl', 'Elektr_min',
'Mech_min', 'Dernim_min'};
predictors = datasetTable(:,predictorNames);
predictors = table2array(varfun(@double, predictors));
response = datasetTable.Gedimas;
% Apmokomas klasifikatorius
trainedClassifier = fitcsvm(predictors, response, 'KernelFunction',
'polynomial', 'PolynomialOrder', 2, 'KernelScale', 'auto',
'BoxConstraint', 1, 'Standardize', 1, 'PredictorNames', {'El_kWh_hl'
'Garai_kg_hl' 'Vanduo_hl' 'CO2_hl' 'Greitis_but_h' 'Profilaktika_d'
```



```

'Ispilta_hl' 'Elektr_min' 'Mech_min' 'Dernim_min'}, 'ResponseName',
'Gedimas', 'ClassNames', {'1.N?ra gedimo' '2.Gedimas'}));

% Atliekamas kryzminis validavimas
partitionedModel = crossval(trainedClassifier, 'Kfold', 5);

% Skaiciuojamas validavimo tikslumas
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun',
'ClassifError');

%% Skaiciuojami rezultatai
[validationPredictions, validationScores] =
kfoldPredict(partitionedModel);

```

=====

Funkcija „Bagged tree” klasifiktraus mokymui:

```

function [trainedClassifier, validationAccuracy] =
trainClassifier(datasetTable)
% Isskleidziami pozymiai ir klases
predictorNames = {'El_kWh_hl', 'Garai_kg_hl', 'Vanduo_hl', 'CO2_hl',
'Greitis_but_h', 'Profilaktika_d', 'Ispilta_hl', 'Elektr_min',
'Mech_min', 'Dernim_min'};
predictors = datasetTable(:, predictorNames);
predictors = table2array(varfun(@double, predictors));
response = datasetTable.Gedimas;
% Apmokomas klasifikatorius
trainedClassifier = fitensemble(predictors, response, 'Bag', 180,
'Tree', 'Type', 'Classification', 'PredictorNames', {'El_kWh_hl'
'Garai_kg_hl' 'Vanduo_hl' 'CO2_hl' 'Greitis_but_h' 'Profilaktika_d'
'Ispilta_hl' 'Elektr_min' 'Mech_min' 'Dernim_min'}, 'ResponseName',
'Gedimas', 'ClassNames', {'1.N?ra gedimo' '2.Gedimas'}));

% Atliekamas kryzminis validavimas
partitionedModel = crossval(trainedClassifier, 'Kfold', 5);

% Skaiciuojamas validavimo tikslumas
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun',
'ClassifError');

%% Skaiciuojami rezultatai
[validationPredictions, validationScores] =
kfoldPredict(partitionedModel);

```

#### Priedas 4. „Matlab“ realizuotų k-NN, SVM, „Bagged tree“ klasifikatorių modeliai

Paveiksle 6.1 pav. pateikti k-NN, SVM, „Bagged tree“ klasifikatorių modeliai.

1x1 ClassificationKNN		1x1 ClassificationSVM		1x1 ClassificationBaggedEnsemble	
Property ^	Value	Property ^	Value	Property ^	Value
NumNeighbors	10	BoxConstraints	284x1 double	Y	284x1 cell
Distance	'euclidean'	CacheInfo	1x1 struct	X	284x10 double
DistParameter	[]	ConvergenceInfo	1x1 struct	W	284x1 double
IncludeTies	0	Gradient	284x1 double	ModelParameters	1x1 EnsembleParams
DistanceWeight	'equal'	IsSupportVector	284x1 logical	NumObservations	284
BreakTies	'smallest'	Nu	[]	PredictorNames	1x10 cell
NSMethod	'kdtree'	NumIterations	23131	CategoricalPredict...	[]
Mu	[2.1955,0.0321,0.0447,...	OutlierFraction	0	ResponseName	'Gedimas'
Sigma	[0.9773,0.0301,0.0752,...	ShrinkagePeriod	0	ClassNames	2x1 cell
Y	284x1 cell	Solver	'SMO'	Prior	[0.4437,0.5563]
X	284x10 double	Y	284x1 cell	Cost	[0,1;1,0]
W	284x1 double	X	284x10 double	ScoreTransform	'none'
ModelParameters	1x1 KNNParams	W	284x1 double	Method	'Bag'
NumObservations	284	ModelParameters	1x1 SVMParams	LearnerNames	1x1 cell
PredictorNames	1x10 cell	NumObservations	284	ReasonForTermin...	'Terminated normall...
CategoricalPredict...	[]	PredictorNames	1x10 cell	FitInfo	[]
ResponseName	'Gedimas'	CategoricalPredict...	[]	FitInfoDescription	'None'
ClassNames	2x1 cell	ResponseName	'Gedimas'	UsePredForLearner	[]
Prior	[0.4437,0.5563]	ClassNames	2x1 cell	NumTrained	180
Cost	[0,1;1,0]	Prior	[0.4437,0.5563]	Trained	180x1 cell
ScoreTransform	'none'	Cost	[0,1;1,0]	TrainedWeights	180x1 double
		ScoreTransform	'none'	CombineWeights	'WeightedAverage'
		Alpha	80x1 double	FResample	1
		Beta	[]	Replace	1
		Bias	2.6098	UseObsForLearner	284x180 logical
		KernelParameters	1x1 struct		
		Mu	[2.1955,0.0321,0.0447,...		
		Sigma	[0.9773,0.0301,0.0752,...		
		SupportVectors	80x10 double		
		SupportVectorLab...	80x1 double		

6.1 pav. Realizuotų klasifikatorių modeliai