



Kauno technologijos universitetas

Informatikos fakultetas

Giliojo mašininio mokymosi metodų trumpalaikiam kritulių prognozavimui iš radaro duomenų tyrimas

Baigiamasis magistro studijų projektas

Marius Oliandra

Projekto autorius

doc. Mantas Lukoševičius

Vadovas

Kaunas, 2024



Kauno technologijos universitetas

Informatikos fakultetas

Giliojo mašininio mokymosi metodų trumpalaikiam kritulių prognozavimui iš radaro duomenų tyrimas

Baigiamasis magistro studijų projektas

Programų sistemos (612I30002)

Marius Oliandra

Projekto autorius

doc. Mantas Lukoševičius

Vadovas

prof. Rytis Maskeliūnas

Recenzentas

Kaunas, 2024



Kauno technologijos universitetas

Informatikos fakultetas

Marius Oliandra

Giliojo mašininio mokymosi metodų trumpalaikiam kritulių prognozavimui iš radaro duomenų tyrimas

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Mariaus Oliandros, baigiamasis projektas tema „Giliojo mašininio mokymosi metodų trumpalaikiam kritulių prognozavimui iš radaro duomenų tyrimas“ yra parašytas visiškai savarankiškai ir 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)

Oliandra, Marius. Giliojo mašininio mokymosi metodų trumpalaikiam kritulių prognozavimui iš radaro duomenų tyrimas. Magistro studijų baigiamasis projektas / vadovas doc. Mantas Lukoševičius; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos mokslai, dirbtinio intelekto informatika.

Reikšminiai žodžiai: mašininis mokymasis, gilusis mašininis mokymasis, kritulių prognozavimas, trumpalaikis kritulių prognozavimas.

Kaunas, 2024. 61 p.p.

Santrauka

Šio magistro darbo tikslas yra ištirti giliojo mašininio mokymosi metodų pritaikymo galimybes trumpalaikiame kritulių prognozavime, naudojant radaro duomenis. Tyrimo metu buvo realizuota 15 skirtingų giliojo mašininio mokymosi modelių, įskaitant konvoliucinius ir rekurentinius neuroninius tinklus bei jų variacijas. Modelių apmokymui ir prognozavimui buvo naudojamos skirtingos duomenų aibės iš darbo metu sudaryto Lietuvos regiono kritulių radaro paveikslėlių duomenų rinkinio, kuris apima periodą nuo 2022 m. kovo 1 d. iki 2023 m. gruodžio 3 d. Rezultatai parodė, kad gilusis mašininis mokymasis gali būti efektyviai pritaikytas trumpalaikių kritulių prognozavimo srityje. Geriausiais prognozavimo rezultatais pasižymėjo UNet ir ConvLSTM architektūrų modeliai, kurių prognozių F1 tikslumo įverčiai viršijo 0,85, o vidutinės kvadratinės paklaidos buvo mažesnės nei 0,04. Taip pat tyrimo metu buvo pastebėta, jog pritaikius konvoliucinių neuroninių tinklų lokalizacijos metodus, galima reikšmingai pagerinti modelių efektyvumą prognozuojant lokalius meteorologinius reiškinius. Tai yra ypač svarbu analizuojant regionines kritulių tendencijas. Tyrimo rezultatų skyriuje atlikta sudarytų modelių palyginamoji analizė, detalai aprašomos jų galimybės ir apribojimai prognozuojant įvairius meteorologinius reiškinius, bei įvertinama lokalizacijos metodų nauda analizuojamoje srityje.

Oliandra, Marius. Analysis of deep learning methods for short-term precipitation forecasting from radar data. Master's Final Degree Project / supervisor lect. Mantas Lukoševičius; Informatics Faculty, Kaunas University of Technology.

Study field and area (study field group): Computer science, artificial intelligence informatics.

Keywords: machine learning, deep learning, precipitation forecasting, precipitation nowcasting.

Kaunas, 2024. 61 p.p.

Summary

The aim of this Master's thesis is to investigate the applicability of deep machine learning methods to short-term precipitation nowcasting using radar data. During the study, 15 different deep machine learning models were implemented, including convolutional and recurrent neural networks and their variations. Different data subsets were used to train and predict the models from a dataset of radar precipitation images of the Lithuanian region, which covers the period from 1 March 2022 to 3 December 2023. The results show that deep machine learning models can be efficiently applied to short-term precipitation nowcasting. The best forecasting performance was achieved by the UNet and ConvLSTM architectures, with F1-accuracy estimates above 0.85 and root mean square errors below 0.04. The study also showed that the application of convolutional neural network localisation techniques can significantly improve the performance of the models in forecasting localised meteorological events. This is particularly important for the analysis of regional precipitation trends. The results section of the study provides a comparative analysis of the developed models, detailing their capabilities and limitations in forecasting various meteorological phenomena, and assesses the usefulness of localisation methods in the field of analysis.

Turinys

Lentelių sąrašas	7
Paveikslų sąrašas	8
Santrumpų ir terminų žodynas	9
Įvadas	10
1. Orų prognozavimo literatūros apžvalga	11
1.1. Skaitmeniniai orų prognozavimo modeliai	11
1.1.1. Skaičiavimo išteklių problemos	11
1.1.2. Prisitaikymas prie klimato kaitos padarinių	12
1.1.3. Trumpalaikis oro sąlygų prognozavimas	12
1.2. Gilusis mašininis mokymasis kritulių prognozavime	13
1.2.1. Konvoliuciniai ir rekurentiniai neuroniniai tinklai	13
1.2.2. Pažangesnės neuroninių tinklų architektūros	14
1.2.3. Konvoliuciniai ilgos trumpalaikės atminties modeliai	15
1.2.4. Lokalizuoti konvoliuciniai neuroniniai tinklai	15
2. Trumpalaikio kritulių prognozavimo metodika	18
2.1. Technologijų analizė	18
2.1.1. Python galimybės ir apribojimai giliojo mokymosi srityje	18
2.1.2. Java programavimo kalbos aspektai	19
2.1.3. C++ efektyvumas ir trūkumai	19
2.1.4. R privalumai ir iššūkiai	19
2.1.5. Pasirinktos technologijos	20
2.2. Duomenų rinkinys ir jo sudarymas	20
2.3. Kritulių prognozavimo eksperimentų struktūra	23
2.3.1. Apmokymo duomenų rinkiniai	23
2.3.2. Testavimo etape atliekamos modelių prognozės	24
2.3.3. Trumpalaikis kritulių prognozavimas	26
2.3.4. Prognozių vertinimo metrikos	27
2.4. Realizuoti giliojo mašininio mokymosi modeliai	28
2.4.1. Eksperimentinių failų struktūra ir realizacija	28
2.4.2. CNN modeliai	30
2.4.3. RNN tipo prognozavimo modelis	30
2.4.4. Konvoliucinis LSTM modelis ir jo realizacija	31
2.4.5. U-Net architektūros variacijos	31
2.4.6. Lokalizuoti neuroniniai tinklai	32
2.4.7. UNet ir ConvLSTM modelių lokalizacijos	33
3. Trumpalaikio kritulių prognozavimo rezultatų analizė	35
3.1. Tiriamų modelių rezultatų apžvalga	35
3.2. Lokalizuotų neuroninių tinklų rezultatai	41
4. Išvados	46
Literatūros sąrašas	48
Priedai	52

Lentelių sąrašas

2.1 lentelė.	Realizuotų CNN modelių palyginimas	30
2.2 lentelė.	Realizuoto RNN_v1.0 modelio architektūra	30
2.3 lentelė.	ConvLSTM modelio sluoksnių ir parametrų informacija	31
2.4 lentelė.	UNet_v1.0 ir UNet_v1.1 modelių sluoksnių parametrų palyginimas	32
2.5 lentelė.	LocalizedCNN modelių parametrų palyginimas	33
2.6 lentelė.	LocalizedUNet_v1.2_loc ir ConvLSTM_v2.2_loc modelių parametrų palyginimas	34

Paveikslų sąrašas

1.1 pav.	Analizuotame straipsnyje naudotų modelių architektūrų palyginimas [37]	14
1.2 pav.	Išmokstamos įvesties (angl. <i>learnable input</i>) konvoliucinio sluoksnio lokalizavimo metodas [42]	16
1.3 pav.	Lokalių svorių (angl. <i>local weights</i>) konvoliucinio sluoksnio lokalizavimo metodas [42].	16
1.4 pav.	Kombinuotas konvoliucinio sluoksnio lokalizavimo metodas [42]	17
2.1 pav.	Lietuvos hidrometeorologijos faktinių kritulių radarų pavyzdinė nuotrauka	21
2.2 pav.	„Tesseract OCR“ veikimo principas [53].	22
2.3 pav.	Apdorotų kritulių radarų paveikslėlių pavyzdžiai	23
2.4 pav.	Prognozuojami krituliai, naudojant 6 mėn. apmokymo rinkinį, 2 val. prognozės	24
2.5 pav.	Prognozuojami krituliai, naudojant 2023 m. vasaros mėnesių apmokymo rinkinį, 2 val. prognozės	25
2.6 pav.	Prognozuojami krituliai, naudojant 2023 m. žiemos mėnesių apmokymo rinkinį, 2 val. prognozės	25
2.7 pav.	Python aplinkoje atvaizduotų validavimui skirtų nuotraukų pavyzdžiai	27
2.8 pav.	Skirtingų modelių eksperimentinių „Jupyter notebook“ failų struktūra.	29
3.1 pav.	Pusės metų duomenų rinkinio, modelių testavimo metu atliktų prognozių tikslumo įverčių diagrama.	35
3.2 pav.	UNet_1.2_loc modelio kritulių prognozių pavyzdžiai ir atitinkamų faktinių duomenų palyginimai	36
3.3 pav.	RNN modelio kritulių prognozių pavyzdžiai ir atitinkamų faktinių duomenų palyginimai	36
3.4 pav.	Skirtingų scenarijų aspektais efektyviausi modeliai	37
3.5 pav.	MSE įverčio sklaidos diagrama	38
3.6 pav.	UNet_v1.2_loc modelio 2 val. prognozės, gautos naudojant Rugsjūčio mėnesio duomenų rinkinį	38
3.7 pav.	UNet modelių palyginimas skirtingų eksperimentų atžvilgiu	39
3.8 pav.	Efektyviausių modelių įverčių palyginimas naudojant 2023 m. žiemos validacijos duomenų rinkinį	40
3.9 pav.	Modelių RMSE įverčiai visais analizuojamais validacijos scenarijais	40
3.10 pav.	ConvLSTM_v2.2_loc modelio Balandžio mėnesio kritulių pavyzdinės prognozės	41
3.11 pav.	localizedCNN_v2.0 modelio išmoktos lokalizacijos matricos ir Lietuvos reljefo palyginimas	42
3.12 pav.	LocalizedCNN_4.2_two_loc_layers modelio lokalizacijos matricos	42
3.13 pav.	ConvLSTM modelio lokalizacijų matricos	43
3.14 pav.	UNet modelio lokalizacijos matrica	44
3.15 pav.	ConvLSTM modelio lokalizuotos versijos rezultatų palyginimas su įprastos architektūros modeliu	44
3.16 pav.	UNet modelių skirtingų versijų rezultatų palyginimasis grafikas	45

Santrumpų ir terminų žodynas

Santrumpos:

CNN – konvoliucinis neuroninis tinklas
ConvLSTM – konvoliucinis ilgos trumpalaikės atminties modelis
CSIM – kritinės sėkmės indeksas
FSS – trupmenų įgūdžių balas
GAN – generatyvinis priešiškas tinklas
LocCNN – lokalizuotas konvoliucinis neuroninis tinklas
LSTM – ilgos trumpalaikės atminties modelis
MAE – vidutinė absoliutinė paklaida
MSE – vidutinė kvadratinė klaida
NWP – skaitmeninis orų prognozavimas
RMSE – šaknies vidurkio kvadratinė paklaida
RNN – rekurentinis neuroninis tinklas
SSIM – struktūrinio panašumo indeksas
UNet – U formos konvoliucinis neuroninis tinklas

Terminai:

Gilusis mokymasis – Mašininio mokymosi sritis, kurioje naudojami daugiasluoksniai neuroniniai tinklai, didelio sudėtingumo duomenų apdorojimui ir prognozavimui.

Konvoliucinis neuroninis tinklas – Neuroninių tinklų tipas, naudojamas vaizdinių duomenų analizavimui ir klasifikavimui, pasitelkiant konvoliucinius sluoksnius.

Rekurentinis neuroninis tinklas – Neuroninių tinklų architektūra, skirta nuoseklių duomenų analizei ir prognozavimui, kurioje neuronai susiję ciklais.

Generatyvinis priešiškas tinklas – Neuroninių tinklų struktūra, susidedanti iš dviejų konkuruojančių tinklų: generatoriaus ir diskriminatoriaus, plačiausiai naudojama duomenų generavimui.

Lokaluoti konvoliuciniai neuroniniai tinklai – Konvoliuciniai neuroniniai tinklai, kurie yra pritaikyti atsižvelgti į lokaliai svarbų informaciją, taip siekiant pagerinti prognozių tikslumą.

Ivadas

Šiais laikais, globalinis klimato atšilimas yra viena iš aktualiausių problemų, nes jo sukelti padariniai yra itin reikšmingi ekonomikos, ekosistemos ir visuomenės stabilumui. Tirpstantys ledynai, kylanti vidutinė vandenynų temperatūra, trumpėjantys šaltųjų sezonų laikotarpiai – tai pastoviai vykstantys veiksniai, kurie kyla dėl klimato kaitos padarinių [1]. Dėl šių padarinių, per pastaruosius 50 m. ekstremalių meteorologinių reiškinių metinis skaičius išaugo penkis kartus [2]. Remiantis 2021 m. statistiniais duomenimis, bent vienas nelaimingas įvykis, susijęs su oru ar klimatu, pasaulyje įvyksta kasdien. Tačiau šių nelaimių sukeltus nuostolius ir prarandamų žmonių gyvybių skaičių galima sumažinti laiku aptinkant ir informuojant visuomenę apie meteorologines anomalijas [2].

Nors skaitmeninių orų prognozių tikslumas pastaraisiais metais ženkliai padidėjo, dėl klimato kaitos sukeltų nenusėjamų meteorologinių reiškinių, klasikiais metodais gaunamos prognozės dažnai būna klaidingos [3]. Siekiant spręsti šiuos iššūkius ir sukurti tokius prognozavimo modelius, kurie sugebėtų prisitaikyti prie klimato kaitos sukeltų pokyčių, yra bandoma pritaikyti giliojo mašininio mokymosi suteikiamas naujas galimybes. Mašininio mokymosi sritis šiais laikais yra viena iš populiariausių ir inovatyviausių mokslo sričių [4]. Todėl jau dabar yra atlikta nemažai tyrimų, kuriuose pritaikyti mašininio mokymosi modelius atliekamos meteorologinių reiškinių prognozės, o jų tikslumas pranoksta skaitmeninių orų prognozavimo metodais gautus rezultatus [5]. Tačiau nors ir orų prognozavimas gali būti efektyvesnis pasitelkiant mašininį mokymąsi, naudojant mašininio mokymosi modelius vis tiek susiduriama su sunkumais siekiant laiku ir tiksliai prognozuoti tam tikrus ekstremalius meteorologinius reiškinius.

Būtent dėl to tarptautinė meteorologijos organizacija 2019 m. apibrėžė trumpalaikių orų prognozavimo metodą. Šio metodo tikslas yra atlikti labai tiksliai ir koncentruotas meteorologinių sąlygų prognozes nuo 0 iki 6 valandų intervale [6]. Atliekant tokias prognozes yra daug lengviau tiksliai nustatyti meteorologinių reiškinių intensyvumą ir laiką, o tai leidžia sumažinti ekstremalių meteorologinių reiškinių sukeltą žalą. Dėl šių priežasčių, magistro darbe analizuojama skirtingų mašininio mokymosi bei giliojo mokymosi modelių pritaikymas trumpalaikiam kritulių prognozavimui.

Pagrindinis magistro darbo tikslas – ištirti giliojo mašininio mokymosi modelių pritaikymo galimybes trumpalaikio kritulių prognozavimo srityje, naudojant radaro kritulių duomenis.

Darbo tikslui pasiekti buvo suformuluoti pagrindiniai uždaviniai:

1. Surinkti ir apdoroti Lietuvos regiono kritulių radaro nuotraukų rinkinį, kuris būtų tinkamas trumpalaikiam kritulių prognozavimui naudojant giliojo mašininio mokymosi modelius.
2. Pritaikyti skirtingus, plačiai meteorologijos srityje analizuojamus, giliojo mašininio mokymosi modelius trumpalaikiam kritulių prognozavimui.
3. Tarpusavyje palyginti realizuotus giliojo mašininio mokymosi modelius ir nustatyti geriausiai Lietuvos regiono kritulių prognozavimui tinkamus metodus.
4. Ištirti lokalizuotų konvoliucinių neuroninių tinklų sluoksnių galimybes trumpalaikiame kritulių prognozavime ir nustatyti jų efektyvumą egzistuojančiuose sprendimuose.

1. Orų prognozavimo literatūros apžvalga

Kritulių prognozavimas yra svarbi ir iššūkių kupina meteorologijos mokslo sritis, turinti tiesioginį poveikį žemės ūkiui, stichinių nelaimių prevencijai ir kasdieniniam žmonių gyvenimui. Tradiciniai meteorologiniai modeliai, tokie kaip skaitmeninis orų prognozavimas, teikia vertingos informacijos, tačiau juos pasitelkiant dažnai susiduriama su ribotu prognozavimo tikslumu ir operaciniu greičiu [7, 8]. Skaitmeninių prognozavimo modelių trūkumai dažnai pasireiškia atliekant trumpalaikes orų prognozes, dėl to mokslinėje literatūroje vis dažniau taikomi giliojo mašininio mokymosi modeliai trumpalaikiam meteorologinių duomenų prognozavimui, kurie pradėjo revoliucionizuoti kritulių prognozavimo sritį. Šiame skyriuje yra apžvelgiama mokslinė literatūra, kurioje analizuojamas kritulių prognozavimas pasitelkiant skaitinius orų prognozavimo modelius ir giliojo mašininio mokymosi modelius, tokius kaip konvoliuciniai neuroniniai tinklai [9] (angl. *convolutional neural network* (CNN)), rekurentiniai neuroniniai tinklai [10] (angl. *recurrent neural network* (RNN)), generatyviniai priešiški tinklai [11] (angl. *generative adversarial network* (GAN)) ir hibridiniai modeliai [12].

1.1. Skaitmeniniai orų prognozavimo modeliai

Šiais laikais didžioji dalis meteorologinių prognozių yra gaunamos pasitelkiant skaitinius orų prognozavimo (angl. *numerical weather prediction* (NWP)) modelius [13]. V. Spiridonov ir M. Čurić [14] aprašo orų prognozavimą kaip svarbiausią taikomosios meteorologijos aspektą ir detaliau aptaria meteorologinių sąlygų, tokių kaip kritulių, debesuotumo, vėjo bei temperatūros prognozavimą. Knygoje išsamiai pristatoma, kaip meteorologai prognozuoja šiuos veiksnius rinkdami kiekybinius duomenis apie esamą atmosferos būklę (temperatūrą, slėgį, drėgmę ir vėjo greitį) ir naudodami skaitinius modeliavimo metodus vaizduoja atmosferą kaip judantį skystį, pagrįstą moksliniu atmosferos procesų supratimu. Tai yra atliekama remiantis fizikiniais dėsniais, tokiais kaip skysčių dinamika ir termodinamika. Kita vertus, knygoje taip pat pabrėžiama, jog norint gauti tikslias ir patikimas orų prognozes, dėl chaotiško atmosferos pobūdžio, yra reikalingi dideli skaičiavimo resursai.

1.1.1. Skaičiavimo išteklių problemos

Mokslinėje literatūroje plačiai kalbama apie didelių skaičiavimo išteklių poreikio problemą atliekant skirtingų tipų meteorologines prognozes. A. Rojas-Campos ir kt. [15] teigia, jog dėl didelio kintamųjų kiekio skaičiavimuose, jų kompleksiško ir sąveikos tarpusavyje išauga skaitinių orų prognozavimo modelių išteklių poreikis. Taip pat, darbe rašoma apie tokių modelių problemas bandant prognozuoti greitai besikeičiančius meteorologinius reiškinius, tokius kaip krituliai ar vėjas. Naudojant skaitmeninius prognozavimo modelius yra susiduriama su šališkumo (angl. *bias*) koregavimo, bei erdvinių (angl. *spatial*) ir laikinių (angl. *temporal*) prognozavimo aspektų derinimo tarpusavyje problemomis, kas turi didelę įtaką tiksliai fiksuojant greitai besikeičiančias oro sąlygas [15, 16, 17]. Siekiant spręsti šias problemas, darbe pristatoma, kaip iš skaitmeniniuose prognozavimo modeliuose naudojamų kintamųjų reikšmių sudaryti kritulių rodmenų žemėlapiai gali būti panaudoti giliojo mašininio mokymosi algoritmuose trumpalaikiam kritulių prognozavimui, pralenkiant klasikiais metodais pasiekiamus tikslumo ir greitimeikos įverčius.

1.1.2. Pristatymas prie klimato kaitos padarinių

Dėl aptartų priežasčių, taikant skaitmeninius orų prognozavimo modelius yra sudėtinga gauti tikslias prognozes, ypač prognozuojant dėl klimato kaitos kylančius ekstremalius meteorologinius reiškinius. Klaidingų ekstremalių kritulių prognozavimo pavyzdžiai ir siūlomas sprendimas yra aptariami V. B. Morozov ir kt. straipsnyje apie pristatymą prie globalinio atšilimo sukeliama klimato pokyčių [18]. Straipsnyje yra pristatomi pavyzdžiai, kaip taikant skaitmeninius orų prognozavimo modelius buvo gautos netikslios kritulių ir temperatūros prognozės tropinėje klimato zonoje. Siekiant pristatyti prie dinamiškų tropinio klimato oro sąlygų autoriai prognozes atliko pasitelkiant trumpalaikio orų prognozavimo (angl. *nowcasting*) metodus. Tyrimo rezultatai parodė, jog ekstremalių kritulių ar temperatūros reiškiniai gali būti tiksliau prognozuojami naudojant ne ilgesnį nei kelių dienų prognozavimo periodą. Darbe taip pat pabrėžiama, jog dėl klimato kaitos kylanti oro temperatūra, kritulių ar kitų meteorologinių reiškinių intensyvumas ir dažnumas ateityje tik didės, todėl yra svarbu plačiau taikyti ir tobulinti tokių reiškinių prognozavimo metodus.

1.1.3. Trumpalaikis oro sąlygų prognozavimas

Atliekant mokslinės literatūros analizę, taip pat buvo išsiginčijama į mokslinius tyrimus, kuriuose analizuojami trumpalaikio kritulių ir kitų oro sąlygų prognozavimo privalumai bei aktualumas ir nauda [19, 20]. E. De Coning ir V. Nietosvaara (angl. *European Conference on Severe Storms*) konferencijoje pristatė trumpalaikių orų prognozavimo metodų naudą Afrikos žemės ūkyje [21]. Tyrimo metu buvo pastebėta, jog pritaikant trumpalaikių orų prognozavimą ir palydovinius meteorologinius duomenis, Afrikos regione galima įdiegti efektyvias ekstremalių oro pokyčių aptikimo bei prognozavimo sistemas. Pasak mokslininkų, naudojant šias sistemas ankstyviems pavojaus įspėjimams gauti, galima nuo potencialios žalos apsaugoti vietinių gyventojų pasėlius, galvijų ūkius, vandens tiekimo sistemas bei transporto sektorių. Dėl klimato kaitos ekstremaliūs meteorologiniai reiškiniai skirtingose klimato zonose tampa vis dažnesni [22, 23, 24]. Šiuo metu mokslinėje literatūroje yra aktyviai ieškoma būdų, kaip optimizuoti šią prognozavimo metodiką maksimizuojant sutaupomus finansinius bei kitus išteklius [25, 26, 27]. Minėtos konferencijos metu, N. Rombeek ir kt. [28] taip pat pristatė trumpalaikių audrų su perkūnija prognozių efektyvumą naudojant pažangius meteorologinius radarus bei giliojo mašininio mokymosi modelius. Šveicarijos mokslininkai šiame tyrime taip pat pabrėžia, jog trumpalaikės krušos, žaibų ar stiprių kritulių prognozės gali apsaugoti nuo didelės ekonominės žalos. Šio tyrimo metu gautos prognozės buvo atliktos pasitelkiant RNN ir CNN modelius, kuriuose naudojami skirtingų pažangių meteorologinių radarų duomenys. Šių modelių pagalba taip pat buvo gautos tikslios ir patikimos žaibavimo prognozės Šveicarijos regione.

Taigi, atlikus skaitmeninių orų prognozavimo modelių taikymo analizę, buvo pastebėta, jog mokslinėje literatūroje yra plačiai rašoma apie šių modelių efektyvumą, tobulėjimo galimybes bei trūkumus šių dienų kontekste. Dėl globalinio atšilimo keliamų klimato pokyčių, ekstremalių oro reiškinių vis dažnėja, o juos teisingai nuspėti naudojant iki šiol priimtais būdais tampa sudėtinga. Todėl meteorologijos srityje aktyviai ieškoma alternatyvių orų prognozavimų metodų, pritaikant naujausius technologinius sprendimus. Trumpalaikės orų prognozės, kurias atliekant naudojami giliojo mašininio mokymosi metodai bei įvairūs moderniųjų meteorologinių radarų duomenys yra

vienas iš efektyviausių šios problemos sprendimo būdų. Remiantis naujausių tyrimų rezultatais, galima teigti, jog pasitelkiant tokį prognozavimo būdą galima tiksliai nustatyti ekstremalius oro reiškinius, laiku pateikti įspėjimus ir taip apsaugoti įvairius sektorius nuo ekonominės žalos.

1.2. Gilusis mašininis mokymasis kritulių prognozavime

Giliojo mašininio mokymosi algoritmų taikymas įvairiose srityse yra viena iš plačiausiai tiriama mokslinės literatūros sričių [29]. Viešasis saugumas, transporto sektorius, finansai, medicina, robotika – tai keli pavyzdžiai sričių, kuriose plačiai tiriamos giliojo mašininio mokymosi modelių pritaikymo galimybės. Kaip buvo aptarta skaitinių kritulių literatūros apžvalgos dalyje, meteorologijoje mašininio ir giliojo mašininio mokymosi modeliai yra naudojami skirtingoms orų prognozėms gauti, o jų gaunamos paklaidos ir tikslumo įverčiai gali pralenkti skaitmeninių prognozavimo modelių gaunamus rezultatus [30, 31]. Todėl tolimesnėje mokslinės literatūros apžvalgoje analizuojami dažniausiai taikomi giliojo mašininio mokymosi modeliai ir jų panaudojimo galimybės skirtingų oro reiškinių prognozavime.

1.2.1. Konvoliuciniai ir rekurentiniai neuroniniai tinklai

Konvoliuciniai neuroniniai tinklai dažniausiai naudojami klasifikuojant ar analizuojant vaizdinius duomenis (nuotraukas), tačiau juos pritaikyti ir gerus rezultatus pasiekti galima ir su laiko erdvėje atliekamomis vaizdinėmis prognozėmis [32, 33, 34]

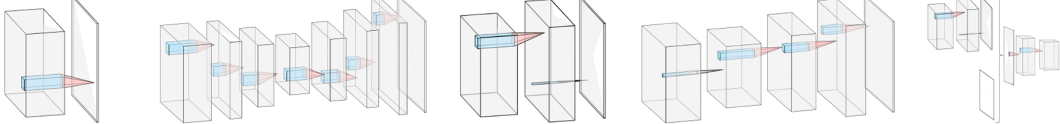
J. Raz ir kt. atliko tyrimą [35], kurio metu buvo lyginami konvoliuciniai bei rekurentiniai neuroniniai tinklai prognozuojant atogrąžų ciklonų (angl. *typhoon*) reiškinius Filipinų valstybės regione. Tyrimui atlikti buvo pasitelkti skirtingų meteorologinių stočių rodmenys bei NASA radarų teikiami duomenys. Analizuojamų modelių prognozės parodė, jog CNN ir RNN modelių panaudojimas gali būti veiksmingas sprendžiant ekstremalių meteorologinių reiškinių klasifikavimo problemas tropiniuose regionuose. Abu modeliai pasižymėjo skirtingomis prognozavimo galimybėmis – didžioji dalis klasifikuojamų kategorijų tiksliau buvo prognozuojamos taikant rekurentinį neuroninį tinklą, tačiau paprastų, ne cikloninių audrų, kritulių tikslesnės prognozės buvo gautos naudojant konvoliucinius tinklus. Dėl šios priežasties tyrėjai pabrėžia, jog atliekant meteorologinių reiškinių prognozavimą su giliojo mašininio mokymosi modeliais, svarbu naudoti tam teisingai pritaikytą algoritmą.

Dar vieną mokslinį tyrimą, kuriame realizuoti giliojo mašininio mokymosi modeliai, sprendžiantys trumpalaikių kritulių prognozavimo uždavinį, atliko H. Hachemi ir kt. [36]. Šio tyrimo metu buvo išbandytas hibridinės, konvoliucinio ir rekurentinio neuroninių tinklų architektūros modelis. Atliktos prognozės buvo lyginamos su abiejų neuroninių tinklų atskirais modeliais ir rezultatai parodė, jog tikslesnės trumpalaikių kritulių prognozės šiaurės Alžyro regione yra gaunamos naudojant hibridinį modelį. Remiantis mokslininkų išvadomis, galima teigti, kad konvoliucinis neuroninis tinklas yra efektyvesnis erdvinės analizės, o rekurentinis – laiko prognozių aspektais. Darbe taip pat pabrėžiama, jog trumpalaikis kritulių prognozavimas – vienas iš perspektyviausių ekstremalių meteorologinių reiškinių prognozavimo būdų, o tyrimo rezultatai įrodo giliojo mokymosi pranašumą šioje srityje.

1.2.2. Pažangesnės neuroninių tinklų architektūros

Tačiau tolimesnėje mokslinės literatūros analizėje buvo pastebėta, jog moderniausiuose meteorologinių reiškinių prognozavimo tyrimuose pastaraisiais metais didesnio dėmesio susilaukia RNN ir CNN modelių variacijos, pavyzdžiui, „U-Net“ ar „LSTM“ neuroniniai tinklai [37, 38, 39].

A. Rojas-Campos ir kt. moksliniame straipsnyje [37] analizuojami penki skirtingi giliojo mašininio mokymosi modeliai skirti trumpalaikiam kritulių prognozavimui Vokietijos vakariniame regione. Tyrimui atlikti buvo naudojamas radaro nuotraukų rinkinys, kuris buvo sudarytas apjungiant skaitinių prognozavimo modelių naudojamus meteorologinius rodmenis erdvėje, taip gaunant detalius ir tikslius kritulių paveikslėlius. Prognozavimas atliktas naudojant bazinį modelį (angl. *baseline*), „U-Net“, du dekonvoliucinius ir vieną sąlyginį generatyvinį priešaringąjį tinklą (angl. *conditional generative adversarial network* (CGAN)). Straipsnyje aprašytų modelių architektūra, parametrų skaičius ir naudotų hiperparametrų palyginamoji iliustracija pateikta 1.1 pav.



Name	Baseline	U-Net	Deconv1L v1	Deconv3L v1	CGAN v1
Layers	2dDeconv(1)	See Appendix A	2dDeconv(32) BatchNorm Conv(1)	Max pooling(2) 2dDeconv(32) BatchNorm 2dDeconv(16) Conv(1)	Deconv3L + discriminator
Kernel size	7×7		7×7	5×5	5×5
Stride	2		2	2	2
Parameters	7008	153 849	224 673	127 841	127 841

1.1 pav. Analizuotame straipsnyje naudotų modelių architektūrų palyginimas [37]

Šių modelių tikslumai buvo lyginami tarpusavyje, naudojant vidutinės absoliutinės paklaidos (angl. *mean absolute error* (MAE)), tiesinės paklaidos tikimybių erdvėje (angl. *linear error in probability space* (LEPS)), teisingos grėsmės (angl. *equitable threat score* (ETC)) ir kritinės sėkmės indekso įverčius (angl. *critical success index* (CSI)). Tyrimo metu pastebėta, jog „U-Net“ modelis pasižymėjo tikslesnėmis vidutinės trukmės kritulių prognozėmis, dekonvoliuciniai modeliai pasiekė geriausius tikslumo įverčius atliekant silpnų kritulių reiškinių prognozavimą, o bendrai geriausius rezultatus mokslininkams pavyko pasiekti naudojant sąlyginį generatyvinį priešišką tinklą. Visų modelių prognozės buvo lyginamos su skaitmeninio orų prognozavimo modelio teikiamomis prognozėmis ir tyrimo išvadose yra teigiama, jog naudojant pasirinktus giliojo mašininio mokymosi modelius trumpalaikiam kritulių prognozavime galima padvigubinti kritulių prognozių detalumą bei tikslumą.

Panašios išvados pateikiamos ir N. K. Sabat ir kt. moksliniame straipsnyje, kuriame rašoma apie

LSTM ir GRU modelių bei skirtingų jų variacijų galimybes trumpalaikiame temperatūros prognozavime [39]. Tyrimui atlikti buvo naudojami skirtingi meteorologiniai rodmenys laiko erdvėje, gauti iš Bubanesvaro miesto hidrometeorologinių stočių radarų Indijoje. Šio tyrimo išvadose pabrėžiama trumpalaikių orų prognozių svarba žmonių gyvenimams, galimybė prisitaikyti prie klimato atšilimo keliamų problemų bei paantrinama, jog gilusis mašininis mokymasis gali tapti pagrindiniu įrankiu atliekant meteorologines prognozes.

1.2.3. Konvoliuciniai ilgos trumpalaikės atminties modeliai

Dar viena mokslinėje literatūroje dažnai tiriama sritis yra hibridinių konvoliucinių ir ilgos trumpalaikės atminties arba ConvLSTM [40] modelių taikymas trumpalaikiame oro sąlygų prognozavime.

D. Schertzer ir E. M. C. Guerrero atliko tyrimą [38], kurio metu panaudojo hibridinį LSTM neuroninį tinklą trumpalaikėms kritulių prognozėms gauti. Prognozės buvo atliekamos naudojant Prancūzijos regiono kritulių radaro nuotraukas, prognozuojant 0-6 val. į ateitį. Darbo išvadose teigiama, jog duomenimis grindžiami giliojo mašininio mokymosi modeliai, tokie kaip ConvLSTM, suteikia perspektyvią alternatyvą trumpalaikiame kritulių prognozavime, nes sugeba nustatyti ryšius tarp įvesties ir išvesties duomenų nesiremiant aiškiais fizikiniais procesais. Tyrėjai pabrėžia, kad atliekant tokio tipo prognozavimą yra svarbu atsižvelgti į poreikius – sumažinus prognozuojamo laiko trukmę, galima ženkliai padidinti prognozių tikslumą.

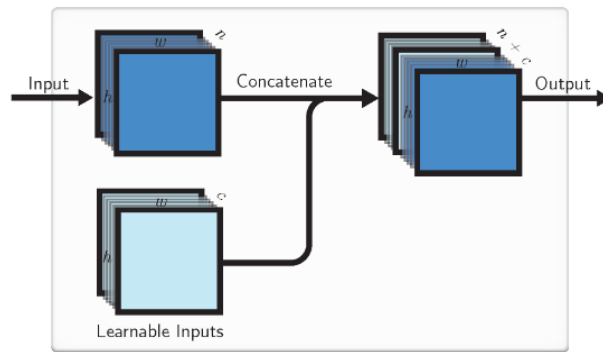
D. Kong ir kt. atliko išsamų tyrimą [41], kurio metu prognozavo iki 2 val. į priekį. Šio tyrimo metu buvo realizuoti du giliojo mašininio mokymosi modeliai – ConvLSTM ir PredRNN, prognozes teikiantys 6 min. intervalais. Remiantis šio bei ankščiau aptartų mokslinių darbų išvadamis galima teigti, jog yra skirtingų pažangių neuroninių tinklų tipų, kuriuos galima sėkmingai pritaikyti sprendžiant trumpalaikių kritulių prognozavimo uždavinius.

1.2.4. Lokalizuoti konvoliuciniai neuroniniai tinklai

Mokslinės literatūros apžvalgos metu taip pat buvo įsigilinta į lokalizuotų konvoliucinių neuroninių tinklų architektūrinius sprendimus (angl. *localised CNNs*). Pasitelkiant CNN modelius galima efektyviai atlikti užduotis su erdviniais duomenimis, tačiau kai kuriose srityse, pavyzdžiui orų prognozavime, naudojant geoerdvinius duomenis, yra susiduriama su tam tikrais iššūkiais. Naudojant lokalizuotus konvoliucinius neuroninius tinklus galima prognozuoti ne tik globaliųjų, bet ir lokalių stebinių savybes, taip praplečiant CNN modelių pritaikymo galimybes [42, 43, 44].

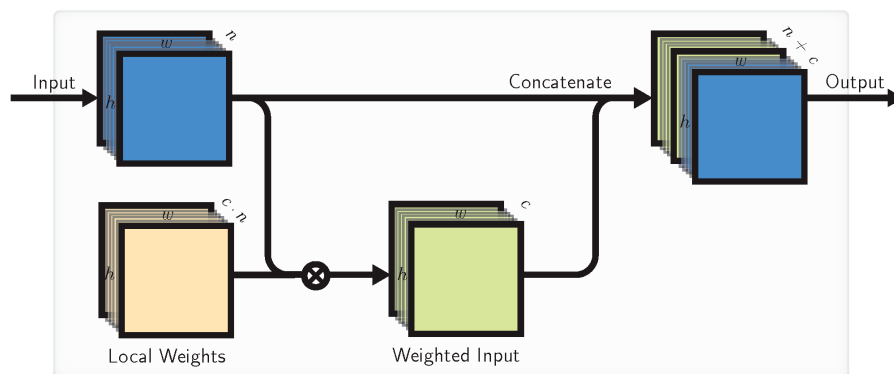
Tarp analizuotų šios mokslo srities darbų buvo ir A. Uselio, M. Lukoševičiaus ir L. Stasyčio parengtas straipsnis, kuriame siūlomi nauji būdai, kaip pasitelkiant lokalizuotų konvoliucinių tinklų metodus trumpalaikiam vėjo reiškinių prognozavimui, galima spręsti problemas, išskylančias naudojant geoerdvinius duomenis [42]. Šiame straipsnyje išsamiai pristatomi keli skirtingi neuroninių tinklų lokalizacijos būdai, universaliai tinkantys bet kokiam CNN tipo modeliui. Pasak autorių, konvoliuciniai neuroniniai tinklai iš principo nėra pritaikyti absoliučiosios branduolio padėties (angl. *absolute position of the kernel*) identifikacijai. Vienas būdas, kurio pagalba galima spręsti šią problemą, yra laisvųjų, modelio išmokstamų įvesčių, arba „LI“ (angl. *learnable*

inputs) sluoksnių pritaikymas. Šie laisvieji parametrai gali būti išmokstami atgalinės sklaidos (angl. *back propagation*) pagalba, kartu su likusiais neuroninio tinklo svoriais ir nereikalauja išankstinės informacijos apie modelio sprendžiamą užduotį. Tai leidžia modeliui prisitaikyti prie lokaliai svarbios informacijos, pavyzdžiui, analizuojamo regiono reljefo, ir efektyviau prognozuoti meteorologinius reiškinius. Pasiūlytame sprendime, išmokstamų įvesčių sluoksnis atitinka pradinės įvesties matmenis ir gali būti integruojamas bet kuriame CNN sluoksnyje (žr. 1.2 pav.). „LI“ sluoksniai padeda konvoliuciniams tinklams lengviau įsisavinti reikšmingą informaciją ir slopinti klaidinančias ypatybes [45, 42].



1.2 pav. Išmokstamos įvesties (angl. *learnable input*) konvoliucinio sluoksnio lokalizavimo metodas [42]

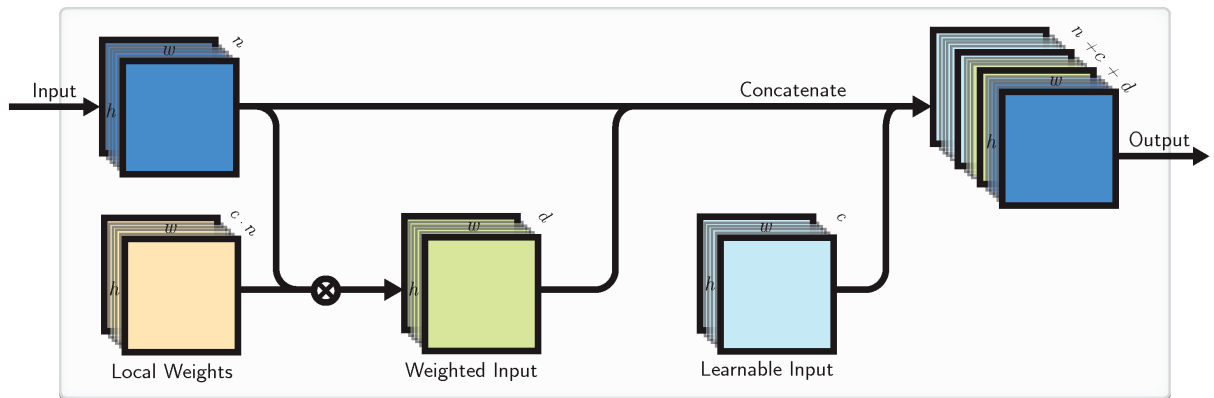
Tačiau sprendžiant uždavinius su erdviniais duomenimis, ne visada skirtingų įvesties dalių informacija yra vienodai svarbi. Prie regioninių meteorologinių reiškinių prognozių tikslumo prisidėti gali lokalių svorių (angl. *local weights* (LW)) lokalizacijos metodas [42, 44]. Naudojant lokalių svorių sluoksnius, CNN modeliai įgauna galimybę lengviau atskirti svarbią informaciją nuo triukšmo duomenų. Kaip šis metodas realizuojamas konvoliucinių tinklų architektūroje yra pavaizduota 1.3 pav.



1.3 pav. Lokalių svorių (angl. *local weights*) konvoliucinio sluoksnio lokalizavimo metodas [42]

Taip pat, realizuojant neuroninių tinklų lokalizacijas, galima naudoti hibridinį būdą – pritaikyti abu – išmokstamų įvesčių ir lokaliųjų svorių sluoksnius. Tokiu būdu yra pasiekama abiejų metodų teikiama nauda, o kadangi šie sluoksniai yra prijungiami prie egzistuojančių neuroninių tinklų, jų parametrų, įvesčių bei išvesčių kiekis gali būti patogiai pritaikomas prie sprendžiamos problemos reikalavimų [42]. Šis metodas yra vizualiai pateikiamas 1.4 pav.

Remiantis analizuotų straipsnių išvadomis, galima teigti jog lokalizuoti CNN leidžia modeliams



1.4 pav. Kombinuotas konvoliucinio sluoksnio lokalizavimo metodas [42]

efektyviau atpažinti ir reaguoti į svarbius, bet lokalizuotus įvykius duomenyse, pavyzdžiui, specifinius orų reiškinius tam tikrose geografinėse vietovėse. Tai yra ypač svarbu meteorologijos taikymo srityse, kur vietos specifikos suvokimas gali ženkliai pagerinti prognozių tikslumą. Taikant lokalizuotus CNN modelius galima efektyviau apdoroti ir analizuoti vietinių duomenų ypatybes, todėl jie tampa nepakeičiami sprendžiant sudėtingas erdvinės užduotis, kur reikalingas aukštas lokalizacijos lygis.

Taigi, remiantis literatūros apžvalgos rezultatais galima teigti, jog trumpalaikis oro reiškinių prognozavimas yra vienas iš perspektyviausių prognozavimo būdų, padedančių prisitaikyti prie klimato kaitos padarinių, o giliojo mašininio mokymosi taikymas šioje srityje, leidžia atlikti itin aukšto tikslumo ir greičio prognozes. Analizės metu apžvelgtos skirtingos neuroninių tinklų architektūros ir jų taikymo būdai, magistro darbo tyrimui atlikti buvo pasirinkti šie giliojo mašininio mokymosi modeliai: CNN, RNN, U-Net ir ConvLSTM. Taip pat, remiantis aptarta lokalizuotų konvoliucinių neuroninių tinklų nauda meteorologinių reiškinių prognozavime, šiame tyrime buvo nutarta išbandyti siūlomus architektūrinius sprendimus trumpalaikiame kritulių prognozavime.

2. Trumpalaiko kritulių prognozavimo metodika

Šiame skyriuje aprašoma metodologija, naudota magistro darbo tyrime. Pirmiausia, atliekama technologijų analizė, siekiant nustatyti tinkamiausias programavimo kalbas ir įrankius giliojo mašininio mokymosi modelių kūrimui ir eksperimentams. Analizuojamos Python, Java, C++ ir R kalbų galimybės ir jų trūkumai. Vėliau aprašomas kritulių radaro duomenų rinkinio surinkimo ir apdorojimo procesas, įskaitant automatizuoto duomenų rinkimo įrankio kūrimą bei duomenų paruošimo veiksmus. Galiausiai pateikiama eksperimentų struktūra, aptariami apmokymo ir testavimo duomenų rinkiniai, bei prognozių vertinimo metrikos.

2.1. Technologijų analizė

Magistro darbo pagrindinis tikslas – realizuoti skirtingų tipų neuroninius tinklus ir pritaikyti sukurtus modelius trumpalaikiam kritulių prognozavimui naudojant kritulių radaro nuotraukų duomenis. Siekiant užtikrinti modelių ir tyrimo patikimumą, greitaveiką bei sklandų darbą, buvo atlikta analizuojamosios srities technologinė analizė. Šios analizės tikslas – ištirti, kokios programavimo kalbos yra dažniausiai naudojamos kuriant ir atliekant eksperimentus su giliojo mašininio mokymosi modeliais. Taip pat, pasidomėti apie pasirinktos technologijos išorines bibliotekas, tinkamas analizuojamoje srityje – kokie jų privalumai, trūkumai ir kurios geriausiai tinka magistro darbo išskeltiems tikslams pasiekti.

2.1.1. Python galimybės ir apribojimai giliojo mokymosi srityje

Apart nuotolinių mokymų platformos „Springboard“ ekspertų, nėra vienos geriausios programavimo kalbos skirtos mašininio mokymosi modelių kūrimui, kiekviena naudojama kalba turi savus privalumus bei minusus [46]. Dauguma duomenų analitikos mokslininkų ir mašininio mokymosi specialistų renkasi programavimo kalbą pagal tai, kokio tipo problemą sprendžia – natūralios kalbos apdorojimo (angl. *natural language processing* (NLP)) problemoms spręsti dažnai pasirinkama Python programavimo kalba, sentimentų analizei atlikti naudojama tiek R, tiek Python programavimo kalbos. Programuotojai, kurie turi patirties su C++, Java ar PHP programavimo kalbomis, dažnai naudoja tas pačias kalbas sprendžiant mašininio mokymosi problemas. Remiantis metiniais statistiniais „Stack Overflow“ duomenimis (2009–2020 m.), detalesnei programavimo kalbų analizei buvo atsirinktos keturios 2020 m. daugiausia populiarumo mašininio mokymosi srityje susilaukusios programavimo kalbos [46].

Remiantis daugeliu šaltinių, Python programavimo kalba yra populiariausias pasirinkimas sprendžiant mašininio mokymosi srities problemas ir kuriant giliojo mokymosi modelius. Virš 60 % mašininio mokymosi programuotojų naudoja šią programavimo kalbą, nes ji yra labai lengvai išmokstama ir paprasta, tačiau suteikianti plačias galimybes [47].

Python pasižymi plačiu panaudojimo spektru, programinis kodas dažniausiai yra ženkliai trumpesnis, nei tokių programavimo kalbų kaip Java ar C++. Didelis bibliotekų pasirinkimas ir integracija – nuo duomenų analizės bei vizualizacijos iki mašininio mokymosi modelių repozitorijų [48]. Taip pat, Python programavimo kalba turi aktyvią bendruomenę, kurioje dalinamasi patirtimi, pavyzdžiais ir reikalinga pagalba. Galiausiai, Python yra atviro kodo programavimo kalba, o tai reiškia, jog visi gali prisidėti prie naujų plėtinių ar bibliotekų kūrimo.

Kita vertus, vienas iš didžiausių Python programavimo kalbos trūkumų yra tai, jog ši programavimo kalba ribotai palaiko žemo lygio programavimą. Tai reiškia, jog ši programavimo kalba nėra geras pasirinkimas įterptinių sistemų programavimui, taip pat apribojamas daugiagijis ir lygiagretus programos vykdymas. Šis aspektas yra svarbus sprendžiant vaizdų atpažinimo problemas, kai yra atliekami eksperimentai su labai didelės raiškos ir dydžio duomenų rinkiniu. Taip pat, lyginant Python programavimą su C++ ar Java, programinis kodas yra vykdomas daug ilgiau, nes tai yra interpretuojama programavimo kalba, o pastarosios – kompiliuojamos [48].

2.1.2. Java programavimo kalbos aspektai

Java programavimo kalba nenusileidžia likusioms dominuojančioms mašininio mokymosi algoritmų kūrimo programavimo kalboms šioje srityje. Ši kalba suteikia patogias aplinkas, skirtas mašininio mokymosi užduočių sprendimams atlikti, tokias kaip "Weka", "Knime" ar "Elka", naudojant grafines naudotojo sąsajas [47]. Pasitelkiant Java programavimo kalbą yra nesunku realizuoti daugiagijį programos vykdymą, šios kalbos kompiliatorius ir interpretatorius pasižymi itin aukštu saugumo lygiu bei patikimumu. Sparčiai kuriami ir tobulinami įrankiai, bibliotekos bei API palengvina darbą bei suteikia plačias galimybes įvairiose srityse [49]. Tačiau programos, sukurtos pasitelkiant Java dažnai būna sunkiai skaitomos, jas realizuojantis programinis kodas ilgas ir nepatogus. Taip pat, lyginant su kitomis kompiliuojamomis programavimo kalbomis, tokiomis kaip C++ ar R, Java reikalauja daugiau atminties ir skaičiavimo resursų [49].

2.1.3. C++ efektyvumas ir trūkumai

Dar viena plačiai mašininio mokymosi srityje naudojama programavimo kalba yra C++. Ši programavimo kalba pasižymi itin sparčiu veikimu ir plačių mašininio mokymosi platformų bei bibliotekų pasirinkimu [47]. C++ programinis kodas yra rašomas statiniu būdu, todėl išvengiama vykdymo metu gaunamų klaidų. Atminties valdymo mechanizmai, aktyvios bendruomenės, patogus plečiamumas – keletas iš daugelio šios kalbos privalumų, kurie labai praverčia kuriant ir naudojant mašininio mokymosi modelius [50]. Kita vertus, C++ programavimo kalba neturi integruotų gijų valdymo mechanizmų, egzistuoja saugumo trūkumai, resursai valdomi pasitelkiant atminties rodykles (angl. *pointers*) ir nėra automatinio šiukšlių surinkėjo (angl. *garbage collector*). Dėl šių priežasčių rašyti programinį kodą yra sudėtinga, reikalauja plačių žinių, kodas nėra lengvai skaitomas ir reikalinga daug programinio kodo eilučių realizuojant elementarius veiksmus [50].

2.1.4. R privalumai ir iššūkiai

Galiausiai, buvo pastebėta, jog didelio populiarumo dėl statistinio skaičiavimo, analizės ir vizualizavimo mašininio mokymosi srityje susilaukia R programavimo kalba [47]. Ši programavimo kalba yra plačiai naudojama duomenų mokslininkų įmonėse, tokiose kaip "Facebook" ar "Google". Nors ir R programavimo kalba dažniausiai pasirenkama bioinžinerijos ir biomedicinos statistinėms problemoms spręsti, ji taip pat yra plačiai naudojama mašininio mokymosi srityse, tokiose kaip tiesinė regresija, duomenų klasifikacija ar sprendimų medžiai, skirtingoms problemoms spręsti [47]. Pagrindinis šios programavimo kalbos privalumas yra tai, jog R yra specializuota duomenų mokslo sričiai – patogios vektorių, matricų ir daugiadimensinių masyvų

valdymo operacijos bei platus grafinės vizualizacijos paketų pasirinkimas. Taip pat, egzistuoja daug skirtingų įrankių ir bibliotekų pritaikytų mašininio mokymosi problemoms spręsti [51]. Deja, R taip pat yra žinoma kaip lėta ir daug atminties reikalaujanti programavimo kalba. Dar vienas šios programavimo kalbos trūkumas yra sunkiai suprantamos arba nepakankamai aprašytos išorinių bibliotekų ir įrankių dokumentacijos [51].

2.1.5. Pasirinktos technologijos

Taigi, apžvelgus skirtingų programavimo kalbų privalumus ir trūkumus bei įvertinus sprendžiamos užduoties keliamus reikalavimus, magistro darbo užduočiai išspręsti buvo pasirinkta Python programavimo kalba. Technologijos pasirinkimas buvo paremtas sintaksės paprastumu, intuityvumu, plačiu bibliotekų pasirinkimu ir daug žadančiomis giliojo mašininio mokymosi palaikymo galimybėmis. Taip pat, šios programavimo kalbos didelė ir aktyvi bendruomenė aktyviai dalinasi naujaisiais sprendimais, išspręstomis problemomis ir pasiektais rezultatais.

2.2. Duomenų rinkinys ir jo sudarymas

Magistro darbo tyrimo objektas – Lietuvos teritorijos klimato zona. Pasirinktas regionas priklauso vėsiojo vidutinio klimato juostai su vidutinio šilto vasaromis bei vidutinio šaltumo žiemomis. Nors šiame regione ekstremalių kritulių reiškiniai pasireiškia retai, geografinės padėties, vyraujančių vakarų ir pietvakarių vėjų, kurie atneša drėgną orą iš Atlanto vandenyno, ir sezoninių pokyčių dėka, Lietuva pasižymi dideliu metiniu kritulių kiekiu. Taip pat, atliekant literatūros analizę buvo pastebėta, jog mokslinių darbų, tiriančių Baltijos regiono meteorologinius ypatumus nėra daug. Dėl šių priežasčių buvo nuspręsta tyrimo metu atliekamiems eksperimentams naudoti Lietuvos Hidrometeorologijos tarnybos teikiamų faktinių kritulių radaro nuotraukų duomenų rinkinį. Viso darbo metu buvo vykdomas automatizuotas šių nuotraukų rinkimas ir periodinis apdorojimas. Šiame poskyryje plačiau aptariama, kokiais žingsniais ir kokių technologijų pagalba buvo paruoštas kritulių radaro nuotraukų duomenų rinkinys.

Vienas iš pirmųjų darbų, atliktų magistro tyrimo metu, buvo automatinio duomenų rinkimo įrankio sukūrimas. Lietuvos Hidrometeorologijos tarnybos puslapyje yra teikiamos viešai prieinamos, kelių pastarųjų valandų kritulių radaro nuotraukos. Kiekviena nuotrauka atvaizduoja Lietuvos regione užfiksuotus faktinius kritulių pasiskirstymo ir intensyvumo rodmenis. Nuotraukose matomos kaimyninių valstybių sienos, Lietuvos didieji miestai ir Baltijos jūros kontūrai. Krituliai žymimi skirtingomis spalvomis, kur mėlyna spalva vaizduoja silpnus kritulius, žalia vidutinius, o raudona - stipriausius. Pavyzdinė nuotrauka pateikta 2.1 pav., vaizduojanti kritulius Lietuvoje, užfiksuotus 2023 m. lapkričio 15 d. 11:40 val.



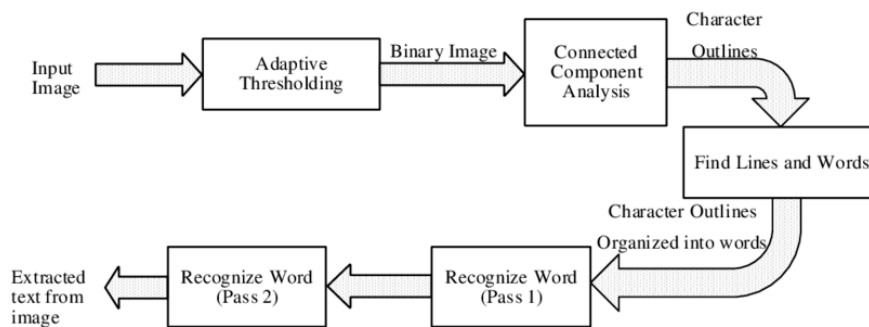
2.1 pav. Lietuvos hidrometeorologijos faktinių kritulių radarų pavyzdinė nuotrauka

Nuotraukos apatiniame kairiajame kampe vaizduojami pavojingų oro sąlygų, tokių kaip šlapdriba ar plikledis, išpėjimai, o dešinėje – užfiksuotų faktinių kritulių data ir laikas. Šie rodmenys fiksuojami 15 min. intervalais, tačiau internetinėje svetainėje jų patalpinimo laikas nėra pastovus – kartais nuotraukos pateikiamos iškart, kartais užtrunka net iki 25 min., kol atitinkama nuotrauka yra patalpinama. Siekiant sukurti patikimą duomenų rinkimo ir saugojimo įrankį, gebanti prisi- taikyti prie skirtingų nuotraukų talpinimo momentų, buvo pasirinktas .NET karkasas ir sukurtas „worker“ tipo servisas.

Šio serviso veikimo principas yra ganėtinai paprastas – atlikti periodines užklausas į viešai prieinamą Lietuvos hidrometeorologijos tarnybos aplikacijų programavimo sąsają (API), jei nuotrauka gauta sėkmingai ir tokia nuotrauka dar neegzistuoja išsaugotame duomenų rinkinyje, tuomet nuotrauka yra išsaugojama ir po 15 min. veiksmai kartojami. Kartais pateikus užklausas buvo gaunami klaidos pranešimai, jog serverio pusėje įvyko klaida, tokiu atveju programa darbą sustabdo ir po 2 min. mėgina dar kartą. Taip pat, jei gauta nuotrauka yra išsaugoto duomenų rinkinio nuotraukos dublikatas, užklausa yra nutraukiama ir laukiama tris minutes kol veiksmai bus vykdomi iš naujo. Tokio veikimo principo pagalba per pastaruosius 3 metus buvo surinktas kokybiškas duomenų rinkinys, apimantis periodą nuo 2022 m. kovo 1 d. iki 2023 m. gruodžio 3 d. Žinoma, dėl techninių nesklandumų, radarų ar programinės įrangos atnaujinimų meteorologinės internetinės sistemos pusėje, duomenų rinkinyje atsirado skirtingos trukmės praleistų nuotraukų periodų, tačiau tolimesniuose duomenų apdorojimo žingsniuose šios spragos buvo ištaisytos, duomenis suskirstant į testinius segmentus. Analizuojamą rinkinį sudaro daugiau nei 50 tūkst. faktinių kritulių radarų nuotraukų. Analizuojamajame periode, iš viso nuotraukų turėjo būti beveik 62 tūkst. tačiau dėl minėtų problemų hidrometeorologijos tarnybos nuotraukų tiekime, buvo

prarasta apie 12 tūkst nuotraukų. Identifikavus reikšmingus prarastų reikšmių periodus, buvo rankiniu pašalinami nutrūkstančių kritulių reiškiniai duomenys, tam, kad modeliai neišmokyti klaidinančių tendencijų.

Tam, kad surinktos kritulių radaro nuotraukos būtų tinkamos giliojo mokymosi modelių apmokymui ir prognozavimui, kiekvieną išsaugotą nuotrauką reikia apdoroti tam tikra veiksmų seka. Šiam procesui atlikti buvo sukurti keli skirtingi Python programavimo kalba parašyti programiniai kodai. Pirmiausia, siekiant užtikrinti duomenų rinkinio vientisumą laiko erdvėje, buvo sukurtas programinis kodas, kuris iš visų pateiktame duomenų rinkinyje esančių paveikslėlių išgauna juose atvaizduotą datą bei laiką. Naudojant „Tesseract Open Source OCR Engine“ [52] įrankį Python aplinkoje, buvo išgauta kiekviename paveikslėlyje vaizduojama tekstinė informacija. Šis įrankis realizuoja keletą skirtingų mašininio mokymosi modelių, apmokytų naudojant įvairaus teksto nuotraukas. Įrankis panaudotas kaip išorinė Python biblioteka, todėl į detalesnį veikimo principą nebuvo gilinamasi, tačiau 2.2 pav. pateikiamas abstraktus teksto iš radaro nuotraukų išgavimo procesas .



2.2 pav. „Tesseract OCR“ veikimo principas [53]

Realizavus šį procesą, visų šio darbo metu surinktų kritulių radaro nuotraukų failų pavadinimai buvo pervadinti taip, kad teisingai atspindėtų vaizduojamos datos ir laiko reikšmes, tokiu būdu buvo gautas tęstinumą išlaikantis duomenų rinkinys. Tam tikrose situacijose naudojant šį įrankį nepavyko išgauti teisingo teksto iš paveikslėlių, tačiau visi tokie failai buvo pervadinti rankiniu būdu, atfiltravus nesėkmingai apdorotų failų pavadinimus.

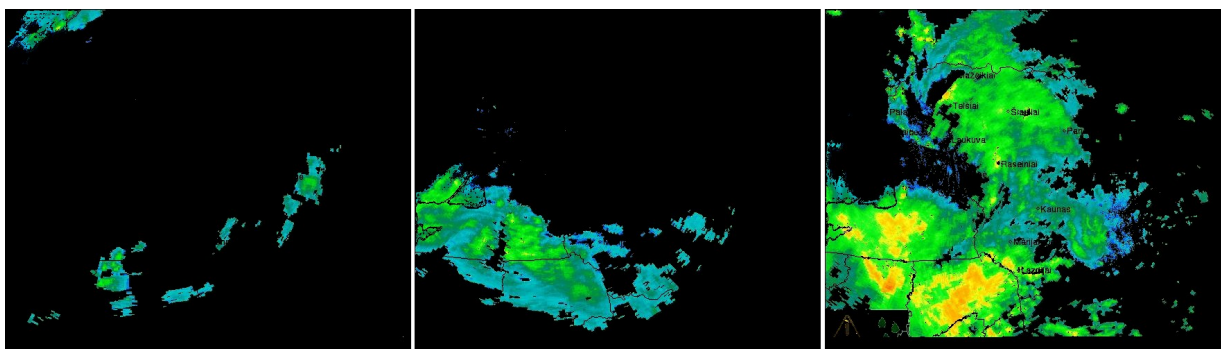
Sutvarkius duomenų rinkinio failų pavadinimus ir tęstinumą, iš nuotraukų buvo pašalinama visa, apart kritulių reiškinį, vaizdinė informacija. Paveikslėlių rinkinys buvo apdorotas atlikus šiuos žingsnius:

1. Konvertuojami visų duomenų rinkinio nuotraukų failų tipai. Hidrometeorologijos tarnybos teikiami paveikslėliai yra „gif“ formato, todėl visi surinkti duomenys buvo konvertuojami į „png“ tipą. Šis nuotraukų tipas buvo pasirinktas dėl vaizdinių duomenų nepraradimo bruožo (angl. *lossless compression*), tam kad paveikslėliuose būtų išsaugoti nepakitę kritulių intensyvumo duomenys.
2. Papildomai buvo atlikti visų surinktų kritulių radaro paveikslėlių dimensijų analizė, kurios metu identifikuota beveik šimtas išskirčių – paveikslėlių, kurių dydžiai neatitiko teikiamų

duomenų standartų. Šios išskirtys buvo pašalintos, jas pakeičiant prieš jas einančių nuotraukų duomenimis.

3. Pakeitus nuotraukų tipus ir pašalinus išskirtis duomenų rinkinyje, buvo atliktas svarbiausias duomenų paruošimo žingsnis – kritulių informacijos išgavimas. Pirmiausia, buvo atrinkta viena iš nuotraukų, kurioje nėra jokių kritulių. Tuomet pikselių perspektyva ši nuotrauka buvo naudojama surasti skirtumams tarp visų kitų nuotraukų rinkinyje.
4. Kadangi nereikalingos informacijos šalinimo metu buvo dirbama su RGBA paveikslėlių formatu, paskutinis duomenų apdorojimo žingsnis buvo konvertuoti peršviečiamas (angl. *transparent*) pikselių reikšmes į juoda spalvą ir paveikslėlius perkonvertuoti iš RGBA formato į RGB. Tokiu būdu buvo sumažinamas duomenų rinkinio dydis ir palengvinamas jo naudojimas.

Tokiu būdu buvo siekta sumažinti duomenų rinkinio dydį, o tai ženkliai sutaupo atliekamų eksperimentų vykdymo laiką bei giliojo mokymosi modelių veikimas yra optimizuojamas. Keli apdorotų kritulių radarų paveikslėlių pavyzdžiai yra pateikti 2.3



2.3 pav. Apdorotų kritulių radarų paveikslėlių pavyzdžiai

Programinis kodas (žr. 1 priede), realizuojantis šiuos duomenų apdorojimo žingsnius buvo pritaikytas visam surinktų radaro nuotraukų rinkiniui ir tokiu būdu gautas vientisas, optimalus ir analizuojamam tyrimui pritaikytas duomenų rinkinys. Sudarytas duomenų rinkinys atspindi kritulių radaro rodmenis Lietuvos regione nuo 2021 m. kovo 1 d. iki 2023 m. gruodžio 3 d.

2.3. Kritulių prognozavimo eksperimentų struktūra

Siekiant nustatyti skirtingų tipų kritulių prognozavimo galimybes ir apmokymo duomenų imties įtaką giliojo mašininio mokymosi modelių tikslumui, iš paruošto kritulių radaro nuotraukų rinkinio buvo parengti mažesni duomenų poaibiai. Buvo naudojami trys skirtingi duomenų rinkiniai modelių apmokymui bei paruošti penki skirtingi duomenų rinkiniai skirti modelių validacijai. Šioje dalyje aptariama kiekvieno naudoto duomenų rinkinio struktūra ir sandara bei argumentuojama parinktų duomenų rinkinių svarba eksperimentų sėkmingumui.

2.3.1. Apmokymo duomenų rinkiniai

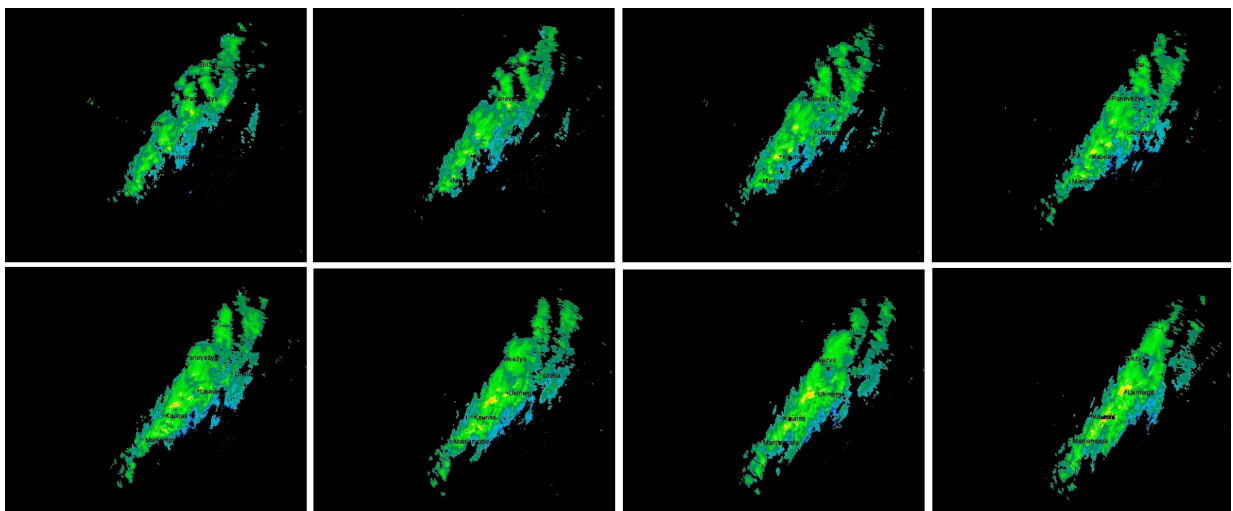
Vykdomi eksperimentai buvo atliekami naudojant skirtingus apmokymo duomenų rinkinius. Atrenkant duomenis, buvo siekiama išskirti tokius paruoštų nuotraukų rinkinio poaibius, kuriuose

vaizduojama kokybiškiausia ir vertingiausia, skirtingų metų sezonų metu užfiksuota kritulių informacija. Buvo sudaryti tokie duomenų poaibiai:

1. 2022 m. pusmečio trukmės duomenų rinkinys, prasidedantis nuo lapkričio mėn. pradžios. Šis duomenų rinkinys buvo sudarytas iš daugiau nei 13 tūkst. kritulių radaro paveikslėlių, kai apytiksliai 36 % paveikslėlių atsispindi krituliai skirtinguose Lietuvos regionuose. Šis rinkinys pasižymėjo didžiausiu praleistų reikšmių skaičiumi, tačiau atlikus detalesnę analizę buvo nustatyta, jog trūkstamos datos yra iš periodo, kuomet buvo atliekami techniniai darbai meteorologijos stotyje. Trūkstami stebiniai buvo pakeičiami paveikslėliais be kritulių.
2. 2023 m. žiemos kritulių radaro nuotraukos – tai rinkinys, sudarytas iš beveik 7 tūkstančių kritulių radaro nuotraukų užfiksuotų žiemos sezono metu. Šiame rinkinyje yra įvairių kritulių reiškinių – regioninių arba valstybės masto pūgų, šlapdrių, audrų. Duomenų poaibyje buvo nustatyta nemažai trūkstamų stebinių, todėl nutrūkstamųjų kritulių reiškiniai buvo pašalinti ir juos vaizduojantys paveikslėliai pakeisti paveikslėliais be kritulių.
3. 2023 m. vasaros kritulių radaro nuotraukų duomenų rinkinys. Ši duomenų poaibį sudaro 8 tūkstančiai paveikslėlių, vaizduojančių vasaros sezono kritulių tendencijas Lietuvoje. Beveik pusė (3792 vnt.) stebinių vaizduoja skirtingų intensyvumų kritulių reiškinius. Šis duomenų poaibis yra kokybiškiausias – praleistos vos kelių dienų kritulių radaro nuotraukos.

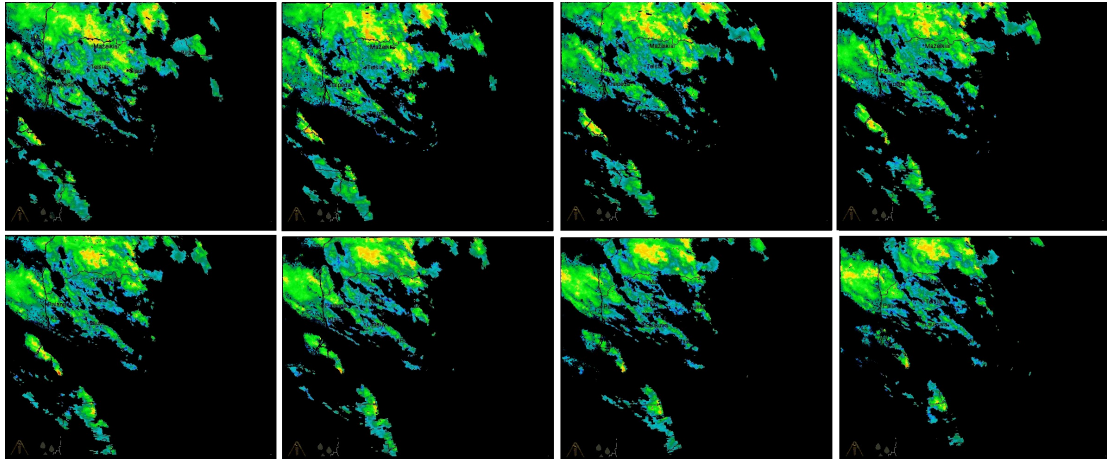
2.3.2. Testavimo etape atliekamos modelių prognozės

Kiekvieno eksperimento metu, atlikus modelių apmokymą, buvo atliekamos testavimui skirtos prognozės. Šios prognozės yra skirtos patikrinti, ar modelis teisingai prisitaiko prie sprendžiamos užduoties. Kadangi tai yra tik verifikacinis žingsnis, prognozavimui naudojamos 8-ios kritulių radaro nuotraukos, pratęsiančios apmokymo duomenų rinkinio kritulių tendenciją 2 val. prognoze. Nuotraukos yra atskiriamos iš duomenų rinkinio prieš apmokant modelį, todėl prognozuojamos reikšmės nėra naudojamos mokymo metu. Žemiau pateiktoje iliustracijoje yra pateiktos 2 val. kritulių prognozės, atliekamos naudojant nuo 2022-ųjų m. lapkričio mėnesio prasidedantį 6 mėn. apmokymo duomenų rinkinį:



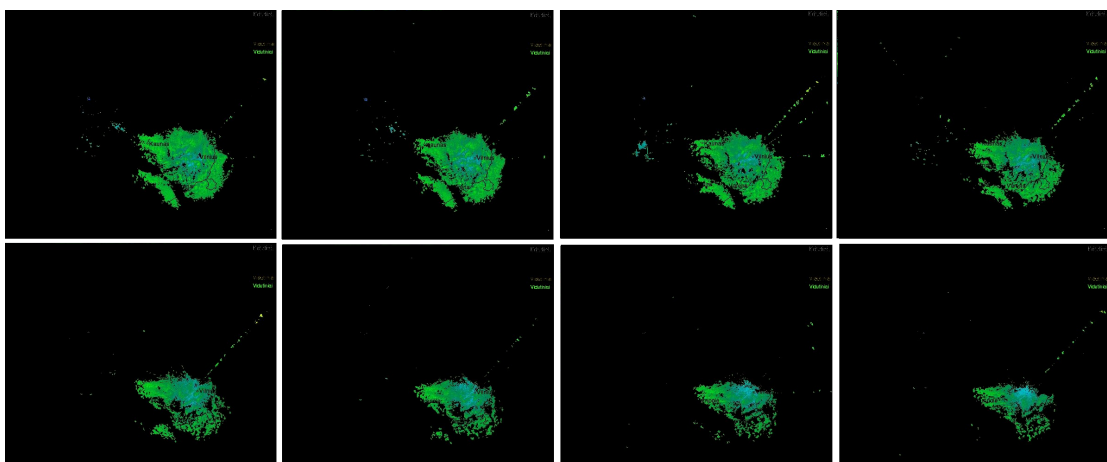
2.4 pav. Prognozuojami krituliai, naudojant 6 mėn. apmokymo rinkinį, 2 val. prognozės

Prognozuojami 2 val. intensyvių kritulių reiškiniai, apimantys Marijampolės, Kauno ir Ukmergės regionus. Kaip buvo minėta duomenų rinkinio aprašymo skyriuje 2.2, kiekviena radaro nuotrauka vaizduoja 15 min. intervalais užfiksuotus kritulių rodmenis, todėl 2 val. yra atvaizduojamos 8-iomis nuotraukomis. Atliekant skirtingų eksperimentų duomenų poaibių atrinkimą buvo siekiama duomenų rinkinius išskirstyti taip, kad skirtinguose eksperimentuose nebūtų naudojamos pasikartojančios kritulių nuotraukos. Tai galima pastebėti palyginus 2.4 pav. vaizduojamus testavimo duomenis su 2.5 pav. pateiktomis 2023 m. vasaros reikšmėmis.



2.5 pav. Prognozuojami krituliai, naudojant 2023 m. vasaros mėnesių apmokymo rinkinį, 2 val. prognozės

Antrojo eksperimento metu buvo atliktos didelės audros prognozės, Lietuvos šiaurinėje dalyje. Šie krituliai yra netolygiai pasiskirstę per visą vakarinę regiono dalį. Lyginant su pusės metų rinkinio prognozėmis, yra matomas aiškus kritulių intensyvumo ir pasiskirstymo skirtumas. Trečiajame eksperimente naudojamo rinkinio prognozuojamos reikšmės yra panašios į 2.4 pav. pateiktus kritulių rodmenis, tačiau šiose prognozėse matomi kritulių radaro trikdžiai, dar vadinami „artefaktais“. Šis duomenų rinkinys buvo pasirinktas siekiant ištirti, kaip analizuojami modeliai reaguoja į klaidingus duomenis, ne natūralias situacijas. Šios nuotraukos yra pateiktos 2.6 pav.



2.6 pav. Prognozuojami krituliai, naudojant 2023 m. žiemos mėnesių apmokymo rinkinį, 2 val. prognozės

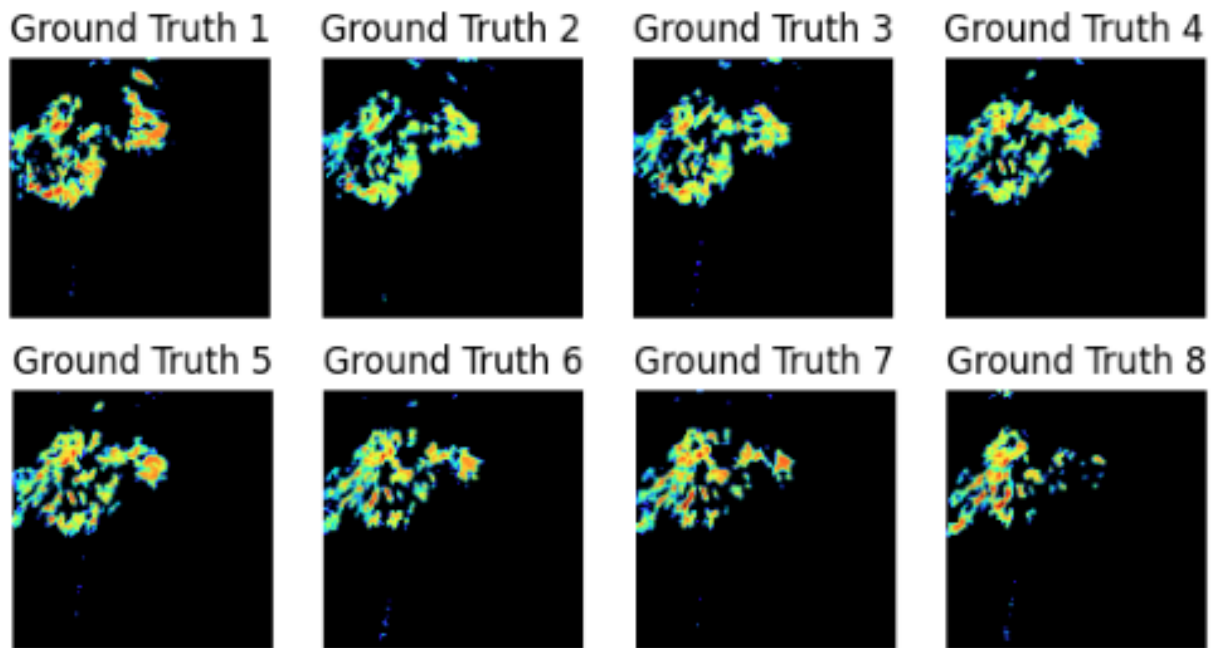
Apibendrinant eksperimentams skirtų duomenų rinkinių analizę, galima teigti, kad visiems trimis eksperimentams buvo naudojami skirtingomis kritulių tendencijomis pasižymintys duomenų rinkiniai. Testavimui parinktos tokios prognozuojamos reikšmės, kurių vaizduojami kritulių reiškiniai yra skirtingi intensyvumo bei pasiskirstymo aspektais. Tokiu būdu siekiama eksperimentų rezultatuose gauti daugiau vertingų įžvalgų ir įvertinti skirtingų modelių efektyvumą.

2.3.3. Trumpalaikis kritulių prognozavimas

Atlikus skirtingus eksperimentus su apmokytais modeliais, buvo atliekamos pagrindinės trumpalaikių kritulių prognozės, įvertinant vidutines prognozavimo paklaidas ir modelių prognozavimo tikslumą. Šiam žingsniui atlikti buvo naudojamos penkios skirtingos, modelių apmokymui nenaudotų duomenų aibės, kurios sudarytos panašiu būdu, kaip aprašyta 2.3.2 skyrelyje. Kad būtų galima korektiškai palyginti gautus rezultatus, kiekvienai duomenų aibei buvo parinktas vienodas periodas, apimantis vienos paros kritulių radaro duomenis. Naudojant sudarytus duomenų rinkinius buvo atliktos slenkančios dviejų valandų prognozės, t. y. naudojama 1 val. (4 kritulių radaro paveikslėliai) istorinių duomenų prognozuoti 2 val. (8 kritulių radaro paveikslėlius) ateities duomenų ir procesas kartojamas, kol prognozuojami visi kritulių paveikslėliai analizuojamą parą. Žemiau pateikti šioms prognozėms sudarytų duomenų rinkinių aprašymai.

- 2022 m. gruodžio 23 d. duomenų rinkinys. Naudojant šį duomenų rinkinį buvo prognozuojami vidutinių kritulių reiškiniai, netolygiai pasiskirstę Lietuvos vakarinėje dalyje. Šią dieną buvo užfiksuotas kritulių pragiedrėjimo reiškinys, kai po tam tikro laiko lietus išblėsta ir liūtis nustoja.
- 2022 m. balandžio 5 d. duomenų rinkinys. Netolygiai pasiskirsčiusių kritulių prognozavimo užduoties rinkinys, sudarytas iš 93 nuotraukų, 61-oje iš jų yra užfiksuoti kritulių reiškiniai, padengiantys bent 4 % Lietuvos regiono. Vaizduojama vidutinio stiprumo liūtis rytinėje ir centrinėje šalies dalyse.
- 2023 m. rugpjūčio 1 d. duomenų rinkinys. Šis duomenų rinkinys pasižymi smarkia audra. Visose 65-iose nuotraukose vaizduojami intensyvūs krituliai, padengiantys didžiąją analizuojamo regiono dalį.
- 2023 m. birželio 11 d. duomenų rinkinys. Kaip ir Gruodžio 23 d. rinkinyje, šioje validavimo duomenų imtyje vaizduojami vidutinio intensyvumo krituliai, ne vienodai pasiskirstę po Lietuvą, o analizuojamos paros pabaigoje – išblėstantys. Toks pasirinkimas buvo atliktas todėl, nes buvo siekiama iširti panašių kritulių judėjimo tendencijų prognozavimą žiemos ir vasaros laikotarpiais.
- 2023 m. gruodžio 1 d. duomenų rinkinys. Šis duomenų rinkinys pasižymi stipria pūga, padengiančia šiaurės ir vakarų regionus. Iš 88-ių nuotraukų, 52-ose yra vaizduojami intensyvūs ir plačioje dalyje susitelkę krituliai.

2.7 pav. yra pateiktas prognozuojamų 2 val. nuotraukų pavyzdys. Tokios prognozės atliekamos vidutiniškai 10 kartų, naudojant visus 5 aptartus validavimo duomenų rinkinius. Atliekant kiekvieną 2 val. prognozę, yra pasirenkamos kelios iš paskutiniųjų turimų duomenų reikšmių (priklausomai nuo modelio konfigūracijos) ir kritulių judėjimo tendencija yra pratęsiama remiantis įvestimi. Šio slenkančio prognozavimo būdu galima apskaičiuoti vidutinį prognozių tikslumą, gaunamą analizuojant pasirinktą laiko periodą.



2.7 pav. Python aplinkoje atvaizduotų validavimui skirtų nuotraukų pavyzdžiai

Taigi, įvertinus modelių validacijos žingsnius ir atrinktų duomenų rinkinių sandarą, galima teigti, jog atliekamų eksperimentų metu gauti modelių įverčiai padės analizuoti skirtingų meteorologinių reiškinių tendencijas. Tokiu būdu validuojant sudarytus modelius, siekiama užtikrinti, jog darbe pasiektos išvados yra vertingos ir teisingai pagrįstos. Taip pat, eksperimentus struktūrizuojant aprašytu būdu, projektinė dalis gali būti lengvai modifikuojama, įtraukiant skirtingo dydžio bei turinio duomenų rinkinius, skirtus apmokymo, testavimo ar validacijos žingsniams atlikti.

2.3.4. Prognozių vertinimo metrikos

Tam, kad būtų galima eksperimentų metu gautas analizuojamų modelių prognozes palyginti tarpusavyje, turi būti apskaičiuojami skaitiniai paklaidų ir tikslumų įverčiai. Kaip buvo pastebėta literatūros apžvalgų metu, nėra universalios metrikų pasirinkimo, kai yra atliekami giliojo mašininio mokymosi eksperimentai su vaizdiniais duomenimis. Todėl šiame darbe buvo pasirinktos tokios įverčių metrikos, kurios yra plačiausiai naudojamos analizuojamos problemos sprendimuose, siekiant užtikrinti, jog bus įvertinami prognozuojamų kritulių išsidėstymo ir intensyvumo aspektai.

Paklaidų skaičiavimams buvo naudojami vidutinės kvadratinės paklaidos (angl. *mean squared error* (MSE)), šaknies vidurkio kvadratinės paklaidos (angl. *root mean squared error* (RMSE)) ir vidutinės absoliutinės paklaidos (angl. *mean absolute error* (MAE)) įverčiai. Mažesnės pasirinktų klaidų vertinimo metrikų reikšmės reiškia, jog prognozė yra tikslesnė. Nors visos trys metrikos vaizduoja prognozių paklaidas, jų analizuojamos savybės padeda atsižvelgti į skirtingus prognozių efektyvumo aspektus.

Vidutinė kvadratinė klaida (MSE) yra viena iš dažniausiai sutinkamų paklaidų skaičiavimo metrikų giliojo mašininio mokymosi srityje. Ištraukus kvadratinę šaknį iš MSE reikšmės, gaunama

šaknies vidurkio kvadratinė paklaida (RMSE). RMSE paklaidos yra lengviau interpretuojamos nei MSE, todėl yra naudingesnis rodiklis vertinant bendrą kritulių prognozių paklaidų dydį. Dar viena dažnai naudojama paklaidų metrika yra vidutinė absoliuti paklaida (MAE), kuri matuoja vidutinius absoliučius skirtumus tarp numatytų ir tikrų verčių. Lyginant su MSE ir RMSE, ši metrika yra ne tiek jautri nuokrypiams, o kritulių prognozavimo iš radaro duomenų užduotyje, tai yra labai svarbus aspektas. Tačiau paklaidų metrikų nepakanka, kad būtų galima priimti pagrįstas išvadas apie modelių prognozavimo tikslumą. Tam, kad būtų galima identifikuoti medelių tinkamumą, reikalinga įvertinti ne tik kritulių intensyvumo paklaidų aspektą, bet ir išsidėstymo erdvėje bei struktūros požymius. Dėl to buvo pasirinktos trys papildomos metrikos, kurios padeda geriau įvertinti modelių tikslumą.

Struktūrinio panašumo indeksas (angl. *structural similarity index measure* (SSIM)) yra lengvai interpretuojama ir plačiai literatūroje naudojama metrika, vertinanti dviejų vaizdų panašumą. Kritulių prognozavimo srityje SSIM yra svarbi, nes ji įvertina prognozuojamų vaizdų kokybę taip, kad ji atitiktų žmogaus regos suvokimą. Taip pat panašių modelių vertinime dažnai sutinkama ir trupmenų įgūdžių balo (angl. *Fractions Skill Score* (FSS)) metrika. Ji gali būti naudojama efektyviai skaičiuojant kritulių prognozių erdvinį tikslumą, įvertinant atitikimus tarp prognozuojamų ir stebimų rodmenų. Galiausiai, į pasirinktų tikslumo įverčių sąrašą buvo įtrauktas ir F1 rodiklis, kuris pasižymi kokybišku vaizdinių duomenų aptikimo tikslumo įvertinimu. Apibendrinant pasirinkimus, galima sakyti jog F1 įvertis orientuojasi į klasifikavimo tikslumą, SSIM – į vaizdo panašumą, o FSS – į erdvinį tikslumą.

2.4. Realizuoti giliojo mašininio mokymosi modeliai

Magistro darbe buvo sudaryti ir realizuoti 15 skirtingi giliojo mašininio mokymosi modeliai, pritaikant juos darbe analizuojamai kritulių prognozavimo problemai spręsti. Šiame skyriuje detaliai aptariamos realizuotų modelių architektūros, jų tarpusavio skirtumai, aprašoma implementacinė projekto dalis. Renkantis modelius buvo siekiama išbandyti skirtingus, šiais laikais plačiausiai tiriamus, neuroninių tinklų tipus, naudojamus trumpalaikių orų reiškinį prognozavimo srityje.

2.4.1. Eksperimentinių failų struktūra ir realizacija

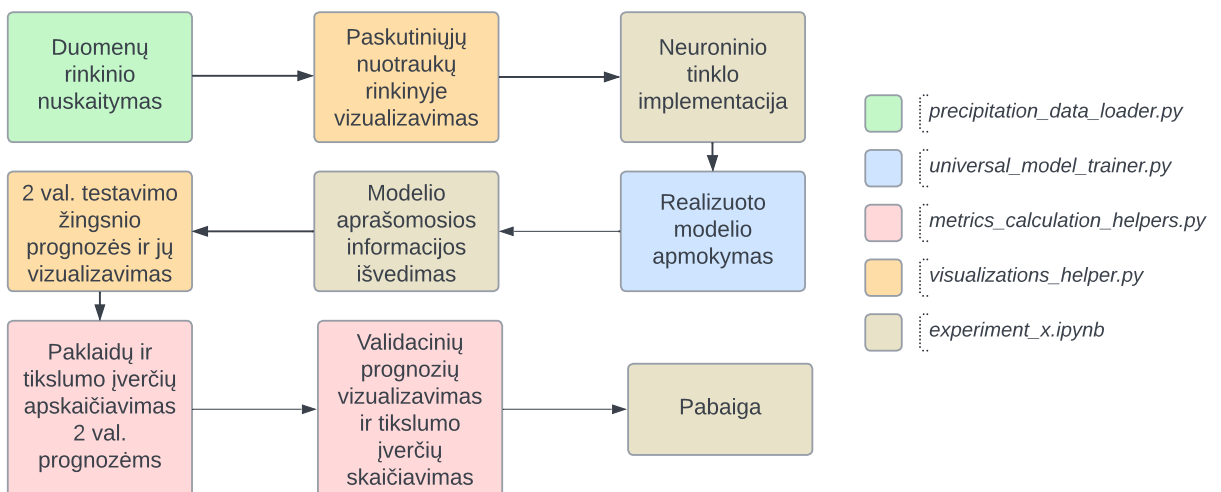
Prieš atliekant modelių implementacinę dalį, buvo realizuoti eksperimentinėje dalyje aptarti žingsniai – modelio apmokymo, testavimo ir validacijos dalys bei kitos reikalingos pagalbinės funkcijos. Kiekvienas modelis yra realizuojamas atskiroje „Jupyter notebook“ aplinkoje. Šiuose atskiruose failuose yra naudojamos sukurtos Python objektų, pagalbinių funkcijų bei klasių bibliotekos. Darbo metu buvo realizuotos 4-ios skirtingos pagalbinės bibliotekos:

- „precipitation_data_loader“ – tai sukurta biblioteka, aprašanti duomenų rinkinio klasės objektą, pavadintą `RadarPrecipitationDataset`. Ši klasė praplečia `Pytorch` bibliotekos `Dataset` objektą, paveldėdama visus jo metodus ir parametrus. Dėl to buvo modifikuotas tik pagrindinis duomenų rinkinio klasės metodas, atsakingas už stebinių atrinkimą iš duomenų rinkinio. Sprendžiamoje prognozavimo užduotyje yra reikalinga, kad teikiant prognozes modeliai panaudotų paskutines ateities duomenų kritulių radaro nuotraukas. Todėl šis metodas buvo perrašytas taip, jog būtų gražinama nurodytos imties įvesties seka

bei sekanti sekos nuotrauka vaizduojanti prognozės tikslą.

- „universal_model_trainer“ – tai modelių apmokymo žingsnį atliekanti biblioteka, kurioje yra aprašomi du, skirtingą mokymo būdą apibūrinantys, pagalbiniai metodai. Pirmoje apmokymo metodo versijoje yra naudojama tik viena įvesties nuotrauka, o antroje – visa įvesties nuotraukų seka. Pagrindiniams darbo eksperimentams atlikti buvo taikoma antroji modelių apmokymo funkcijos versija. Naudojant šias funkcijas, parametruose reikalinga pateikti modelio ir duomenų nuskaitymo objektus, nurodyti naudojamą vykdymo aplinką (GPU arba CPU įrenginys) bei tokius, su mašininio mokymusi susijusius objektus, kaip optimizavimo ir tikslo funkcija, apmokymo epochų skaičių (angl. *number of epochs*).
- „metrics_calculation_helpers“ – biblioteka, kurioje yra aprašytas visas programinis kodas, kurio pagalba atliekamas prognozių tikslumo ir paklaidų įverčių skaičiavimas bei gautų rezultatų (įskaitant ir kritulių nuotraukas) atvaizdavimas. Realizuojant kiekvieną iš skirtingų modelių eksperimentinių failų, šios bibliotekos funkcijos buvo naudojamos testavimo ir validacijos žingsniuose su jau apmokytu modeliu.
- „visualizations_helpers“ – tai su vizualizavimo dalimi susijusių pagalbinių funkcijų pake-tas. Čia yra aprašytos funkcijos, kurių pagalba atliekamos prognozės su sudarytais mo-deliais ir vizualizuojami rezultatai. Taip pat šioje bibliotekoje yra aprašytas programinis kodas, skirtas lokalizacijos sluoksnių atvaizdavimui.

Pasitelkiant aprašytas pagalbines bibliotekas, eksperimentų realizavimas tampa ženkliai papras-tesnis. Nors kiekvieno žingsnio metu atliekami veiksmai yra gan sudėtingi, „Jupyter notebook“ aplinkoje realizuotų funkcijų ir objektų panaudojimas reikalauja vos kelių eilučių programinio kodo. Realizuojant skirtingus giliojo mašininio mokymosi modelius buvo naudotos tos pačios pagalbinės bibliotekos, taip užtikrinant jog atliekamus eksperimentus būtų galima lyginti tarpu-savyje esant toms pačioms sąlygoms. Kiekvieno eksperimento metu atliekami žingsniai ir jų naudojamų sąsajų paaiškinamoji diagrama yra pateikiami 2.8 pav.



2.8 pav. Skirtingų modelių eksperimentinių „Jupyter notebook“ failų struktūra.

2.4.2. CNN modeliai

Šiame projekte buvo sukurti ir pritaikyti du konvoliuciniai neuroniniai tinklai (CNN) – CNN_v2.1 ir CNN_v2.2, kurių architektūros pateiktos 2.1 lentelėje. CNN_v2.1 modelis turi du konvoliucinius sluoksnius su 16 ir 32 filtrų, tarp kurių yra maksimalaus telkimo (angl. *max pooling*) sluoksnis, o po jų seka du pilnai sujungti sluoksniai (angl. *linear*), kurie padaroja išgautus bruožus tinklo pabaigoje. Šiuo modeliu buvo siekiama patikrinti elementaraus konvoliucinio neuroninio tinklo architektūros efektyvumą analizuojamoje srityje.

Kita vertus, CNN_v2.2 modelis sukurtas naudojant iš eilės einančius tris konvoliucinius sluoksnius. Šia versija siekiama iširti maksimalaus kaupimo ir pilnai susietųjų sluoksnių įtaką prognozių tikslumui. Dėl to, abiejų modelių versijų parametru skaičius ženkliai skiriasi – CNN_v2.1 turi 25 187 520 treniruojamų parametru, o CNN_v2.2 1 185. Šių skirtumų pagalba galima iširti modelio parametru skaičiaus tiesioginę įtaką reikalaujamų resursu kiekiui ir teikiamu prognozių tikslumui. Lentelėje 2.1 pateikiama detali šių dviejų modelių parametru ir jų architektūros palyginimo analizė.

Sluoksnis (gylis)	CNN_v2.1 Parametru skaičius	CNN_v2.2 Parametru skaičius
Conv2d (1)	160	80
MaxPool2d (2)	–	–
Conv2d (3)	4 640	1 040
Linear (4)	16 777 728	–
Linear (5)	8 404 992	–
Viso	25 187 520	1 185

2.1 lentelė. Realizuotų CNN modelių palyginimas

2.4.3. RNN tipo prognozavimo modelis

Be konvoliuciniu, šiame projekte buvo išbandytas ir rekurentinis neuroninis tinklas, siekiant palyginti skirtingu modelių architektūras ir jų pranašumus trumpalaikėse kritulių prognozėse. RNN modelis, kurio architektūra pateikta 2.2 lentelėje, buvo suprojektuotas taip, kad apdorotų sekų duomenis, gautus iš radarų vaizdu, ir užtikrintų laikinės informacijos analizę. Pagrindiniai šio modelio elementai yra vienas rekurentinis sluoksnis su 8 651 776 parametrais ir vienas pilnai susietas sluoksnis su 8 404 992 parametrais. Bendras parametru skaičius buvo pasirinktas panašus į CNN_v2.1 modelio tam, kad būtų galima iširti, kuris neuroninio tinklo tipas yra labiau tinkamas atliekant trumpalaikį kritulių prognozavimą naudojant radaro nuotraukų rinkinį.

Sluoksnis (gylis)	Parametru skaičius
RNN (1)	8 651 776
Linear (2)	8 404 992
Viso	17 056 768

2.2 lentelė. Realizuoto RNN_v1.0 modelio architektūra

2.4.4. Konvoliucinis LSTM modelis ir jo realizacija

Taip pat, darbe buvo realizuoti ir modernių architektūrų neuroniniai tinklai, plačiai naudojami analizuojamoje srityje, pavyzdžiui, konvoliuciniai ilgos trumpalaikės atminties (ConvLSTM). Kaip buvo aptarta anksčiau, šis neuroninių tinklų tipas realizuoja konvoliucinių ir rekurentinių architektūrų geriausius aspektus sprendžiant laiko ir erdvės uždavinius. Todėl šiame projekte buvo sukurtas ir išbandytas ConvLSTM modelis, kurio architektūra pateikta 2.3 lentelėje. Šis modelis sujungia konvoliucinių neuroninių tinklų gebėjimą analizuoti erdvinę informaciją ir LSTM tinklo laiko erdvės analizės galimybes.

Realizuojant šį modelį buvo sukurtos konvoliucinės LSTM ląstelės (angl. *ConvLSTM cell*), kurios dinamiškai pridedamos neuroninio tinklo pradžioje. Kiekviena iš šių ląstelių turi skirtingą kiekį išmokstamųjų parametrų ir yra atsakinga už LSTM architektūrinių principų realizavimą, kartu pritaikant konvoliucinius veiksmus. Toks sluoksnių išdėstymas padeda modeliui išlaikyti ir analizuoti sudėtingas erdvines ir laiko priklausomybes, būdingas meteorologiniams reiškiniams. Kadangi eksperimento programiniame kode tinklo sudedamieji komponentai aprašomi kaip atskiros klasės, tinklo kompleksiskumas gali būti lengvai modifikuojamas. Lentelėje 2.3 pateikiama šio modelio sluoksnių parametrų analizė.

Sluoksnis (tipas-gylis)	Parametrų skaičius
ConvLSTM (1-1)	–
ModuleList (2-1)	–
ConvLSTMCell (3-1)	9 856
ConvLSTMCell (3-2)	55 424
ConvLSTMCell (3-3)	221 440
Conv2d (1-2)	65
Viso	286 785

2.3 lentelė. ConvLSTM modelio sluoksnių ir parametrų informacija

2.4.5. U-Net architektūros variacijos

Taip pat aukšti tikslumo įverčiai kritulių prognozavimo srityje buvo pasiekti naudojant UNet architektūros modelius. Šiame projekte sukurti ir išbandyti dvi skirtingos UNet modelių versijos. UNet modeliai pasižymi simetrine kodavimo ir dekodavimo architektūra, kai kodavimo dalis yra skiriama įvesties vaizdo imties mažinimui, o dekodavimas – imties didinimui tam, kad būtų tiksliai nustatomos lokaliai vaizdų savybės. Tai yra ypač svarbu meteorologinių reiškinių prognozavimui naudojant vaizdinius duomenis. Todėl šiame projekte buvo įgyvendintos dvi U-Net versijos: UNet_v1.0 ir UNet_v1.1, kurių architektūros pateiktos 2.4 lentelėje.

Pirmasis modelis, UNet_v1.0, sudarytas iš 8-ių iš eilės einančių sudedamųjų blokų, realizuojančių UNet architektūrą. Pirmiausia, įvestys apdorojamos konvoliuciniame bloke, kuris yra sudarytas iš trijų konvoliucinių, normalizacijos ir ReLU aktyvavimo sluoksnių. Po to, seka trys sumažinimo (angl. *downsampling*) ir trys atstatymo (angl. *upsampling*) sluoksniai. Paskutiniame modelio bloke yra realizuojamas konvoliucinis sluoksnis. UNet_v1.1 modelis yra šiek tiek supaprastinta

anksčiau pristatyto modelio versija, kurioje buvo pašalinti keli sluoksniai iš pirmojo konvoliucinio bloko. Toks pakeitimas buvo pasirinktas eksperimentiniam palyginimui, siekiant nustatyti, ar didesnis sluoksnių skaičius gali pagerinti prognozių tikslumą. 2.4 lentelėje pateikiama detali šių dviejų modelių parametrų analizė.

Blokas	Sluoksnis (tipas-gylis)	UNet_v1.0 Parametrų skaičius	UNet_v1.1 Parametrų skaičius
DoubleConv (1-1)	Sequential (2-1)	–	–
DoubleConv (1-1)	Conv2d (3-1)	640	640
DoubleConv (1-1)	BatchNorm2d (3-2)	128	128
DoubleConv (1-1)	ReLU (3-3)	–	–
DoubleConv (1-1)	Conv2d (3-4)	36 928	36 928
DoubleConv (1-1)	BatchNorm2d (3-5)	128	128
DoubleConv (1-1)	ReLU (3-6)	–	–
DoubleConv (1-1)	Conv2d (3-7)	–	36 928
DoubleConv (1-1)	BatchNorm2d (3-8)	–	128
DoubleConv (1-1)	ReLU (3-9)	–	–
Down (1-2)	Sequential (2-2)	–	–
Down (1-2)	MaxPool2d (3-10)	–	–
Down (1-2)	DoubleConv (3-11)	221 952	369 792
Down (1-3)	Sequential (2-3)	–	–
Down (1-3)	MaxPool2d (3-12)	–	–
Down (1-3)	DoubleConv (3-13)	886 272	1 476 864
Down (1-4)	Sequential (2-4)	–	–
Down (1-4)	MaxPool2d (3-14)	–	–
Down (1-4)	DoubleConv (3-15)	3 542 016	5 902 848
Up (1-5)	ConvTranspose2d (2-5)	524 544	524 544
Up (1-5)	DoubleConv (2-6)	–	–
Up (1-5)	Sequential (3-13)	1 771 008	2 361 600
Up (1-6)	ConvTranspose2d (2-7)	131 200	131 200
Up (1-6)	DoubleConv (2-8)	–	–
Up (1-6)	Sequential (3-14)	443 136	590 976
Up (1-7)	ConvTranspose2d (2-9)	32 832	32 832
Up (1-7)	DoubleConv (2-10)	–	–
Up (1-7)	Sequential (3-15)	110 976	590 976
OutConv (1-8)	Conv2d (2-11)	65	65
Viso	-	7 701 825	11 613 633

2.4 lentelė. UNet_v1.0 ir UNet_v1.1 modelių sluoksnių parametrų palyginimas

2.4.6. Lokalizuoti neuroniniai tinklai

Konvoliucinių neuroninių tinklų lokalizacijos metodo taikymo galimybėms iširti buvo realizuoti septyni skirtingi lokalizuoti konvoliuciniai neuroniniai tinklai (LocalizedCNN). Šie modeliai sujungia standartinių konvoliucinių sluoksnių ir lokalizacijos sluoksnių privalumus, leidžiančius efektyviau apdoroti lokaliųjų ir globaliųjų savybių informaciją erdviniuose duomenyse, kas yra svarbu siekiant geriau prognozuoti kritulius iš radarų vaizdų.

Pirmoji modelio versija, LocalizedCNN_v1.0, sudaryta iš keturių sluoksnių – lokalizuoto konvoliucinio, įprasto konvoliucinio, maksimalaus telkimo (angl. *max pooling*) ir dar vieno įprasto konvoliucinio sluoksnio pabaigoje. Lokalizacijos sluoksnyje yra naudojamas Pytorch bibliotekos, Tensor struktūros filtras. Pirmiausia, šis filtras buvo inicijuojamas užpildant saugomą informaciją vienetais, tačiau modelio mokymo eigoje reikšmės buvo atitinkamai atnaujinamos, kas padėjo sėkmingai nustatyti lokaliai svarbios informacijos reiškiniai duomenyse. Pirmojoje LocalizedCNN_v1.0 versijoje lokalizacija buvo pritaikyta atliekant elementarią įvesties ir Tensor lokaliojo filtro elementų daugybą.

LocalizedCNN_v2.0 modelis buvo patobulintas, įtraukiant pilną lokalizacijos matricą, kuri padidina modelio gebėjimą apdoroti duomenis skirtingose erdvės vietose. Taip pat šio modelio pakeitimais siekiama iširti lokalizacijos tipo įtaką kritulių prognozavimo efektyvumui. Kita vertus, LocalizedCNN_v3.0 versija buvo sukurta modifikuojant anksčiau aptartą modelį, pakeičiant lokalizavimo metodą iš elementų daugybos į apjungimą (angl. *concatination*). Šis veiksmas atliekamas pasitelkiant Pytorch bibliotekos `torch.cat()` funkciją. Lokalizacijos tipas išlaikomas matriciniu pavidalu, atitinkančiu įvesties nuotraukos dydį. Taip pat, siekiant patikrinti, ar lokalizacijos įtraukimas į šios struktūros konvoliucinį neuroninį tinklą yra naudingas, buvo sukurta LocalizedCNN_v3.0_no_loc modelio versija, kurioje lokalizuotas konvoliucinis sluoksnis pakeičiamas paprastu konvoliuciniu tinklu, išlaikant tą pačią modelio struktūrą.

Taip pat, buvo sukurti trys papildomi ketvirtosios kartos LocalizedCNN modeliai. LocalizedCNN_v4.0 modelyje buvo išlaikoma LocalizedCNN_v3.0 modelio lokalizacijos dalis, tačiau buvo pašalintas maksimalaus telkimo sluoksnis. Siekiant patikrinti, ar lokalizacijos sluoksnio padėtis tinkle daro įtaką rezultatams, sukurta LocalizedCNN_v4.1_second_layer_loc versija, kurioje šis sluoksnis apkeičiamas su antruoju konvoliuciniu sluoksniu. Paskutinėje, LocalizedCNN_v4.2_two_loc_layers, versijoje realizuojami du lokalizuoti konvoliuciniai sluoksniai, pakeičiant įprastą konvoliucinį sluoksnį lokalizuotu. Šiuo modeliu buvo tikrinama kelių lokalizuotų sluoksnių nauda kritulių prognozavimo efektingumui. 2.5 lentelėje pateikiama visų aptartų modelių palyginamoji informacija.

Modelis	Lokalizuoti sluoksniai	Konvoliuciniai sluoksniai	Lokalizacijos matrica	Parametrų skaičius
LocalizedCNN_v1.0	1	2	Ne	26 529
LocalizedCNN_v2.0	1	2	Taip	42 913
LocalizedCNN_v3.0	1	2	Taip	26 529
LocalizedCNN_v3.0_noLoc	0	3	Ne	22 857
LocalizedCNN_v4.0	2	2	Taip	26 529
LocalizedCNN_v4.1_second_layer_loc	1	2	Taip	26 529
LocalizedCNN_v4.2_two_loc_layers	2	2	Taip	71 201

2.5 lentelė. LocalizedCNN modelių parametrų palyginimas

2.4.7. UNet ir ConvLSTM modelių lokalizacijos

Galiausiai buvo realizuotos dvi papildomos eksperimentinės modelių versijos, kuriose lokalizacijos metodais siekiama optimizuoti anksčiau pristatytus UNet ir ConvLSTM modelius. Šie modeliai skirti iširti lokalizacijos sluoksnių įtaką tikslumui ir efektyvumui pažangesniuose giliojo mokymosi modeliuose, prognozuojant kritulius iš radarų vaizdų.

Pirmasis eksperimentas buvo atliktas su UNet modeliu, sukuriant LocalizedUNet_v1.2_loc versiją. Šiame modelyje buvo įtrauktas lokalizacijos sluoksnis, kuris naudojamas pradiniam konvoliuciniame sluoksnyje, siekiant tiksliau išgauti erdvinę informaciją. LocalizedUNet_v1.2_loc modelyje naudojamas LocalizedConv2d sluoksnis, kuriame yra taikoma matricos tipo lokalizacija apjungimo metodu, taip, kaip anksčiau aptartame LocalizedCNN_v2.0 modelyje. Šio modelio kompleksškumas ženkliai skiriasi nuo LocalizedCNN versijų, tačiau šio eksperimento tikslas buvo palyginti paprastą UNet modelį su lokalizuotu modeliu.

Panašiu būdu buvo sukurta ir lokalizuota ConvLSTM modelio versija ConvLSTM_v2.2_loc. Šio modelio tikslas buvo integruoti lokalizacijos sluoksnį į ConvLSTM ląstelę, siekiant pagerinti modelio gebėjimą nustatyti lokalaus svarbumo informaciją analizuojamame uždavinyje. Abiejų aptartų modelių versijų palyginamoji lentelė yra pateikiama apačioje.

Modelis	Lokaluoti sluoksniai	Konvoliuciniai sluoksniai	Lokalizacijos matrica	Parametrų skaičius
UNet_v1.2_loc	1	10	Taip	7 681 729
ConvLSTM_v2.2_loc	1	3	Taip	290 820

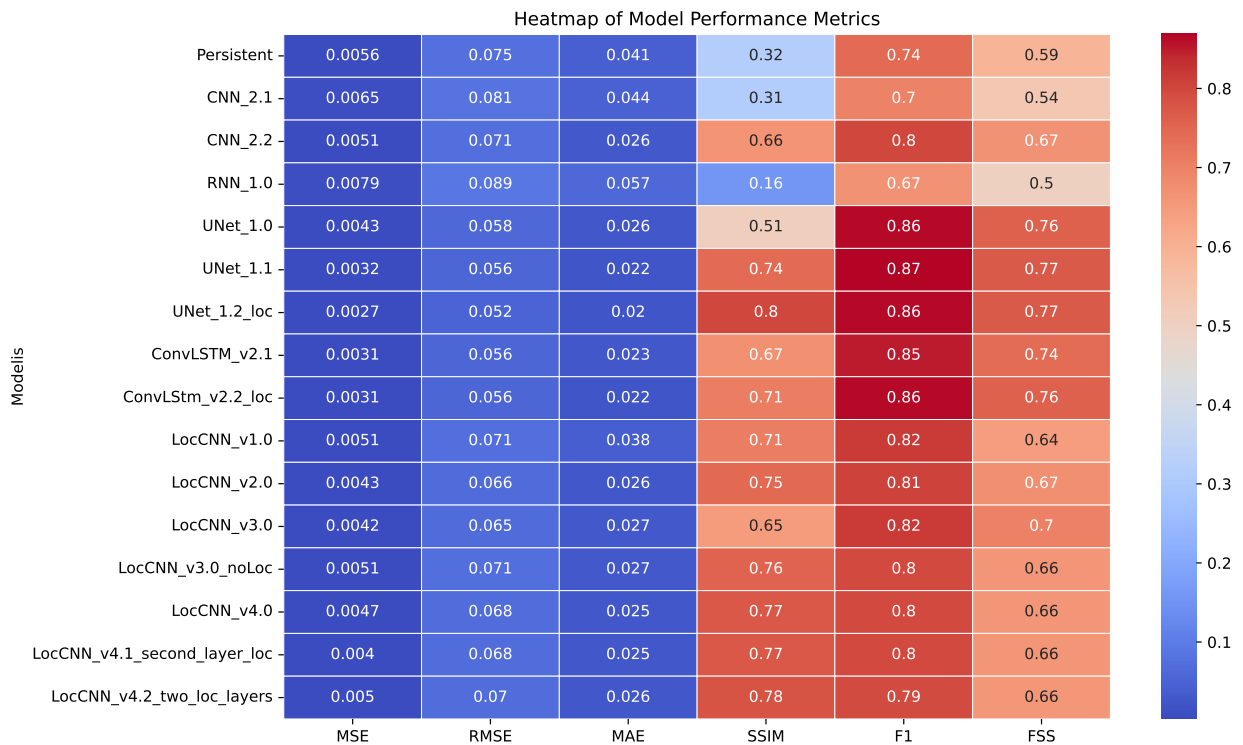
2.6 lentelė. LocalizedUNet_v1.2_loc ir ConvLSTM_v2.2_loc modelių parametrų palyginimas

3. Trumpalaikio kritulių prognozavimo rezultatų analizė

Magistro darbe buvo atliktas trumpalaikis kritulių prognozavimas pasitelkiant realizuotus CNN, RNN, UNet ir ConvLSTM modelius. Modelių apmokymui ir prognozavimui buvo naudojami 2.3.1 ir 2.3.3 skyriuose aptarti duomenų rinkiniai. Buvo išmėgintos skirtingos apmokymo metu naudojamų kriterijų ir optimizavimo funkcijų kombinacijos, tačiau pirminiai rezultatai, kad analizuojamam uždaviniui tinkamiausi yra MSELoss criterion ir Adam optimizer metodai. Šiame skyrelyje detaliam aptariami trumpalaikio kritulių prognozavimo Lietuvoje rezultatai ir lokalizuotų neuroninių tinklų efektyvumas trumpalaikio kritulių prognozavimo srityje.

3.1. Tiriamų modelių rezultatų apžvalga

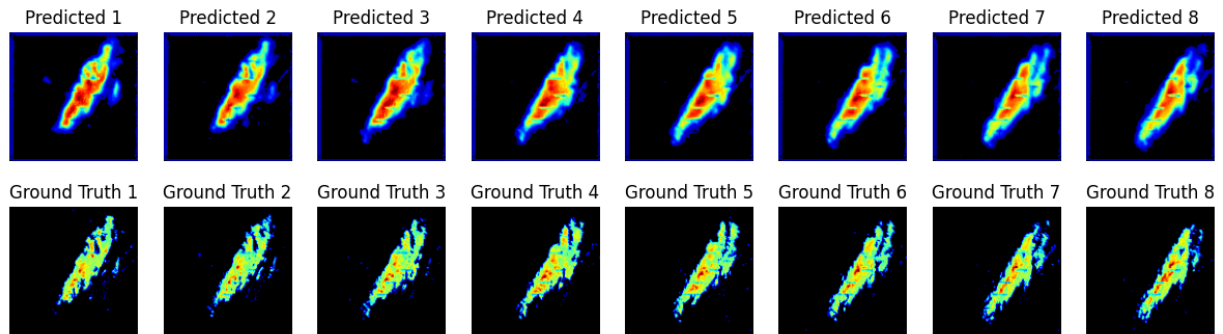
Pirmiausia, dviejų valandų kritulių reiškinį prognozavimui buvo pritaikyti 15-a skirtingų giliojo mašininio mokymosi modelių, naudojant pusės metų apmokymo duomenų rinkinį. Tokiu būdu buvo siekiama nustatyti, kurie modeliai efektyviausiai sprendžia analizuojamą problemą. Tam, kad būtų galima tarpusavyje palyginti modelių rezultatus, buvo sudaryta 3.1 diagrama, kurioje pateikiama visų analizuojamų modelių tikslumo įverčių informacija gauta atliekant 2 val. prognozavimą.



3.1 pav. Pusės metų duomenų rinkinio, modelių testavimo metu atliktų prognozių tikslumo įverčių diagrama

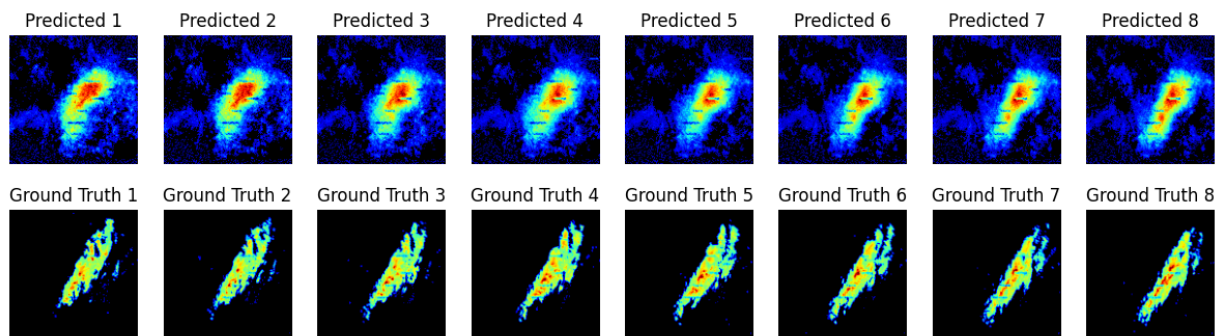
Diagramoje mėlyna spalva yra žymimos reikšmės, kurios yra mažesnės nei vidutinė visų įverčių reikšmė (apie 0,5), o raudona spalva – didesnės. Pirmi trys stulpeliai vaizduoja MSE, RMSE ir MAE paklaidas, kur mažesnė reikšmė nusako tikslesnes prognozes. Likusiuose stulpeliuose pateikiami SSIM, F1 ir FSS tikslumo įverčiai, kur priešingai nei paklaidų metrikų stulpeliuose – geresnės prognozės yra apibūdinamos aukštesnėmis reikšmėmis. Iš diagramos galima pastebėti, jog tiksliausios prognozės buvo gautos naudojant UNet ir ConvLSTM modelius. Lokalizuoto

UNet neuroninio tinklo UNet_1.2_loc modelio prognozių SSIM ir F1 tikslumo įverčiai buvo didesni nei 0,8, o paklaidų įverčiai mažiausi iš visų analizuojamų modelių. Prognozės, su kuriomis buvo pasiektos geriausios reikšmės yra pateikiamos 3.2 pav.



3.2 pav. UNet_1.2_loc modelio kritulių prognozių pavyzdžiai ir atitinkamų faktinių duomenų palyginimai

Kita vertus, prasčiausios prognozės buvo gautos naudojant RNN_v1.0 modelį. Rekurentinis neuroninis tinklas pasižymi aukščiausiais paklaidų įverčiais ir mažiausiais tikslumo matais. Modelio prognozuojami kritulių rodmenys yra pateikti 3.3 pav., kuriuos palyginus su anksčiau minėtomis UNet_v1.2_loc modelio prognozėmis, galima pamatyti reikšmingus prognozuojamų kritulių paveikslėlių skirtumus. Iš gautų rezultatų galima teigti, jog apmokytas rekurentinis neuroninis tinklas nesėkmingai nustato kritulių pasiskirstymo bei intensyvumo informaciją laiko erdvėje.

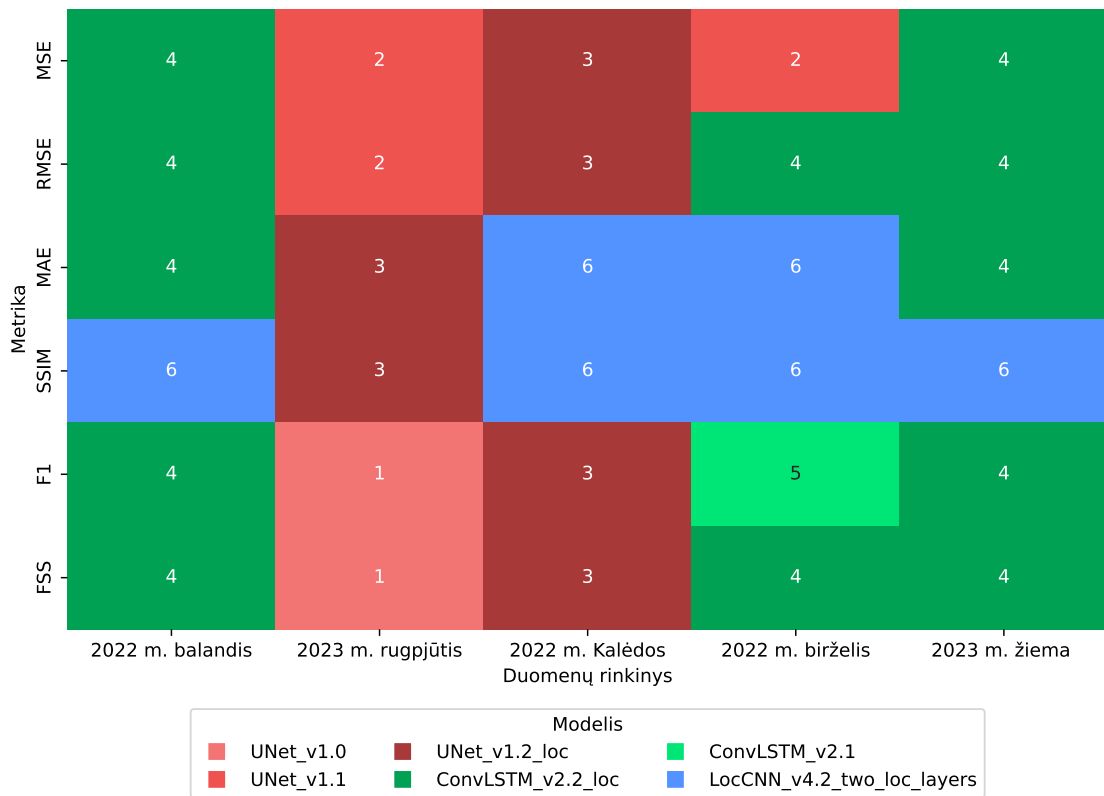


3.3 pav. RNN modelio kritulių prognozių pavyzdžiai ir atitinkamų faktinių duomenų palyginimai

Tam, kad modeliai būtų teisingai įvertinami, reikalinga atsižvelgti į skirtingų kritulių sąlygų prognozavimo galimybes. Kaip buvo aptarta ankstesniuose skyriuose, modelių efektyvumo vertinimo metu buvo naudojami penki skirtingi duomenų rinkiniai, vaizduojantys vienos paros kritulių reiškinius skirtingais metų laikais. Kiekvienas skirtingu duomenų rinkiniu apmokytas modelis buvo naudojamas visų penkių validacijų metu, todėl iš viso tyrimo metu buvo atlikti 225 eksperimentai.

Visuose validacijos duomenų rinkiniuose yra vaizduojami skirtingi kritulių reiškiniai ir jų tendencijos. Atlikus validaciją su modeliais, apmokytais naudojant pusės metų duomenų rinkinį, buvo pastebėta, jog kiekvieno eksperimento metu geriausius įverčius ne visada pasiekdavo tie patys modeliai. Tačiau atrinkus skirtingų eksperimentu metu geriausius rezultatus pasiekusių modelių

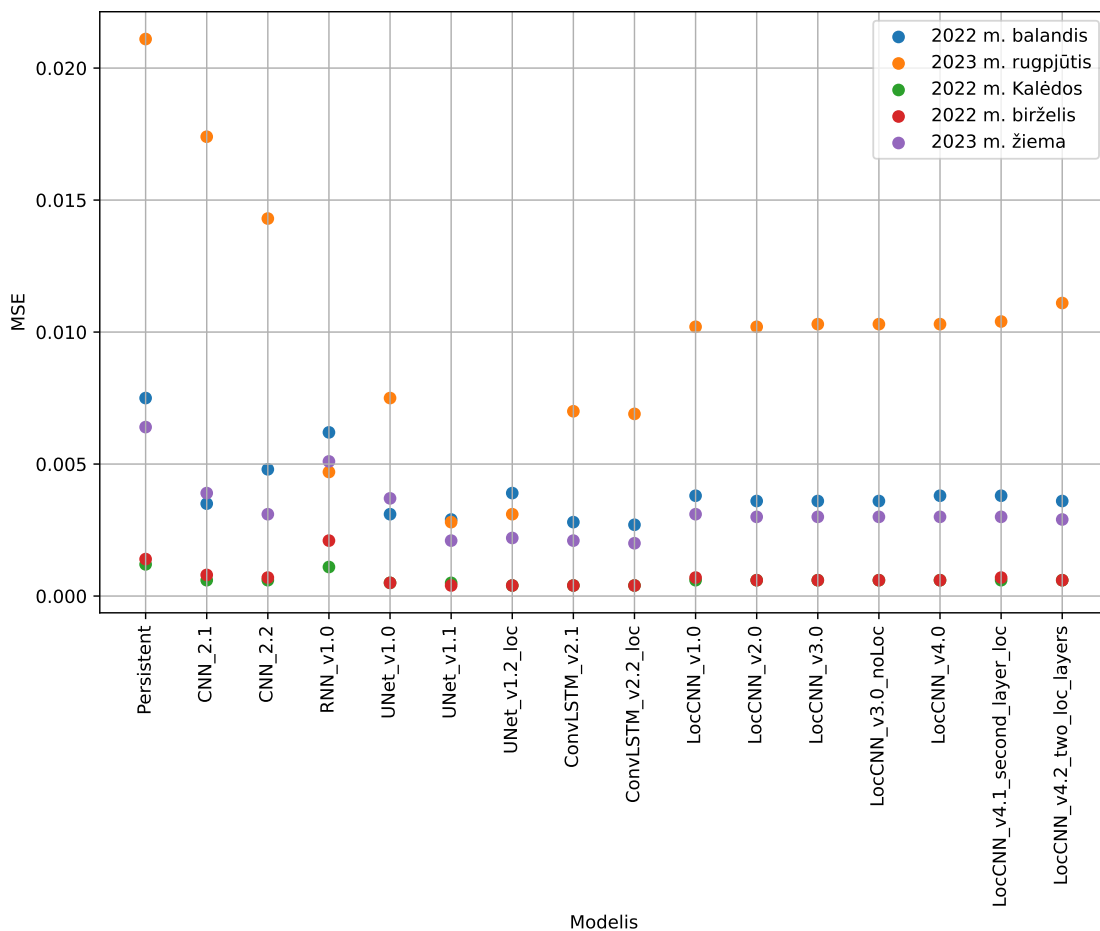
pavadinimus, buvo pastebėta, jog efektyviausios prognozės yra atliekamos kelių skirtingų architektūrų modelių. Skirtingais aspektais geriausius rezultatus pasiekę modeliai yra atvaizduoti 3.4 pav. pateiktame žemiau:



3.4 pav. Skirtingų scenarijų aspektais efektyviausi modeliai

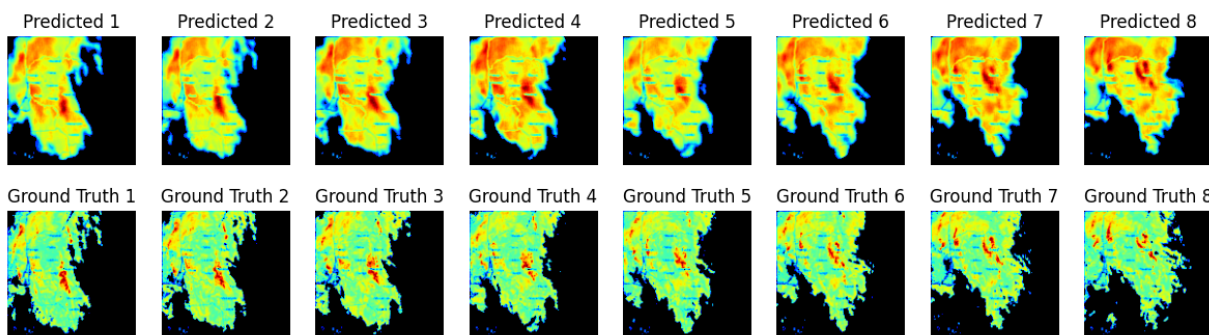
3.4 pav. pateiktame grafike galima pastebėti, jog daugiausia aukščiausių įverčių buvo pasiekta naudojant ConvLSTM_v2.2_loc modelį. Šio modelio paklaidų metrikos balandžio ir gruodžio rinkiniuose buvo mažiausios, o F1 ir FSS geriausi tikslumo įverčiai buvo pasiekti net su trimis iš penkių skirtingų analizuotų duomenų rinkinių. LocCNN_v4.2_two_loc_layers modelis, kuriame yra naudojami du lokalizacijos sluoksniai, iš penkių analizuojamų scenarijų, net 4-iouose pasiekė geriausias SSIM įverčio reikšmes. Prognozuojant rugpjūčio mėnesio kritulius, tiksliausi rezultatai buvo gauti naudojant UNet_v1.1 ir UNet_v1.2_loc modelius. Šio validavimo metu buvo prognozuojami didelės audros reiškiniai vakariniame Lietuvos regione, todėl galima daryti prielaidą, jog sukurti UNet architektūros modeliai yra labiau tinkami ekstremalių orų prognozavimui nei kiti. Dviejų valandų pavyzdinės prognozės, gautos naudojant UNet_v1.2_loc modelį yra pateiktos 3.6 pav.

Sudarius analizuojamų tikslumo ir paklaidų įverčių sklaidos diagramas, vaizduojančias gautas reikšmes su skirtingais duomenų rinkiniais, buvo pastebėtos kelios rezultatų tendencijos. Remiantis žemiau pateiktoje diagramoje (žr. 3.5 pav.) vaizduojama informacija, galima pastebėti, jog 2022 m. rugpjūčio mėn. duomenų rinkinys sukėlė daugiausia iššūkių prognozuojant kritulių reiškinius – šiuo atveju buvo gautos didžiausios MSE paklaidų reikšmės. Mažiausios paklaidų reikšmės atliekant kritulių prognozes buvo gautos naudojant 2023 m. žiemos ir 2022 m. Kalėdų duomenų rinkinius.



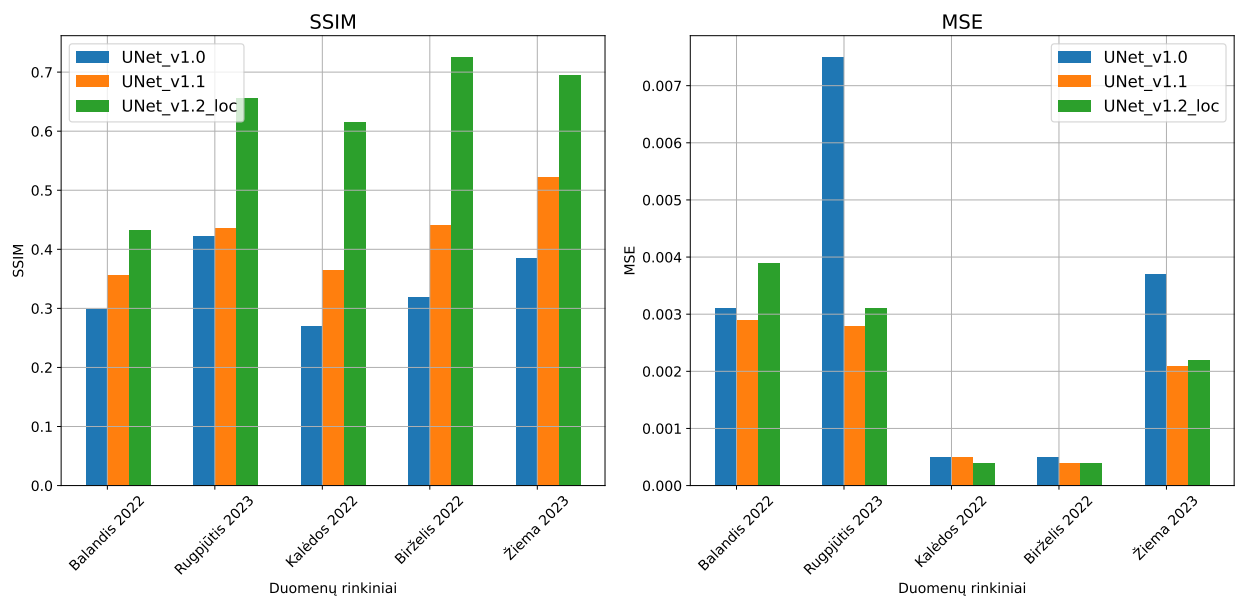
3.5 pav. MSE įverčio sklaidos diagrama

Palyginus skirtingų modelių prognozavimo rezultatus, galima pastebėti, kad UNet_v1.2_loc ir ConvLSTM_v2.1 modelių paklaidos analizuojamais atvejais buvo mažiausios. Tuo tarpu CNN_2.2 ir RNN_v1.0 modeliams buvo gautos didesnės paklaidos, ypač lyginant rezultatus, gautus su 2022 m. rugpjūčio duomenimis. Skirtingi duomenų rinkiniai daro reikšmingą įtaką modelių prognozavimo efektyvumui ir tai pabrėžia duomenų rinkinių sudėtingumo, sezoninių bei klimato sąlygų įtaką kritulių prognozavimo tikslumui. Tolimesnėje analizuojamų rezultatų apžvalgoje yra detaliau aptariamas skirtingų modelių efektyvumas atliekant kritulių prognozes. Žemiau pateikiamas grafikas (žr. 3.6 pav.), kuriame palyginami skirtingų modelių SSIM ir MSE metrikų įverčiai, naudoti įvairiems duomenų rinkiniams, siekiant detaliau analizuoti modelių prognozavimo tikslumą.



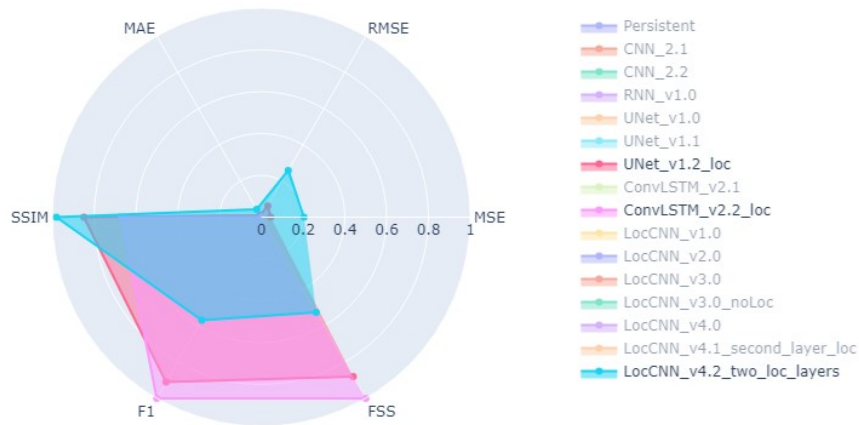
3.6 pav. UNet_v1.2_loc modelio 2 val. prognozės, gautos naudojant Rugpjūčio mėnesio duomenų rinkinį

Pateiktoje iliustracijoje, kaip ir ankstesniuose pavyzdžiuose, pirmojoje eilutėje yra pavaizduoti prognozuojami kritulių paveikslėliai, o apačioje – faktinė kritulių informacija. Kiekviena nuotrauka atitinka 15 min. intervalais fiksuojamų kritulių radaro rodmenis. Lyginant UNet_v1.2_loc modelio rezultatus, galima pastebėti jog kritulių išsidėstymas, intensyvių reiškinių pozicijos ir judėjimo tendencijos atitinka faktinius kritulių rodmenis. Tam tikri neatitikimai yra matomi vaizduojamos audros pakraščiuose, bei bendriniame intensyvumo ženklime. Lyginant tarpusavyje UNet architektūros modelių versijas, buvo pastebėta, kad panaši efektyvumo tendencija buvo išlaikoma visų atliktų eksperimentų metu. 3.4 pav. parodyta, kad visi trys UNet modeliai efektyviausio modelio vietą užima tik tam tikrais scenarijais. 3.7 pav. pateiktoje stulpelinėje diagramoje yra atvaizduoti UNet modelių SSIM ir MSE metrikų palyginimai, gauti naudojant skirtingus kritulių prognozavimo duomenų rinkinius.



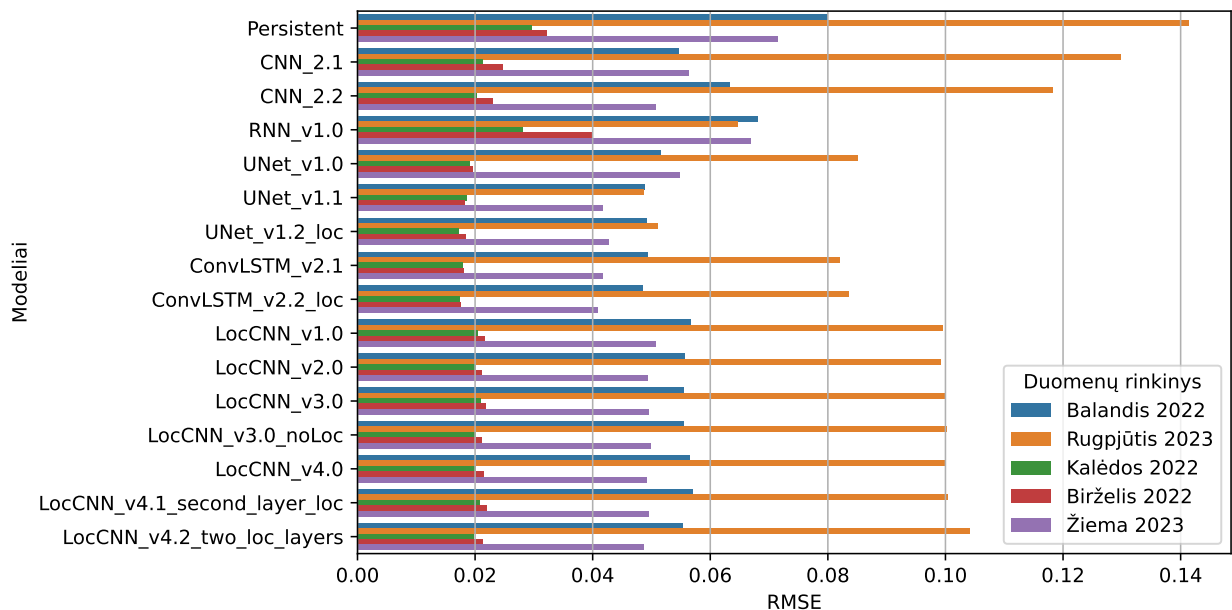
3.7 pav. UNet modelių palyginimas skirtingų eksperimentų atžvilgiu

Iš 3.7 pav. pateiktos diagramos galima pastebėti itin gerus MSE paklaidos įverčius, gautus naudojant 2022 m. Kalėdų, 2022 m. birželio ir 2023 m. žiemos duomenų rinkinius. Žemiausios MSE įverčių reikšmės pasiektos su 3.4 pav. pateiktomis geriausių modelių versijomis. Palyginus modelius, kuriems buvom gautos didžiausios tikslumo įverčių reikšmės, buvo pastebėta, jog ConvLSTM ir UNet modeliams gauti panašūs paklaidų ir tikslumo įverčiai visais analizuojamais scenarijais, tačiau LocCNN modeliai pasižymėjo geresniais SSIM metrikos rodmenimis. Tai leidžia daryti prielaidą, jog LocCNN modeliai yra tinkamesni kritulių pasiskirstymo, bet ne intensyvumo prognozavime. Aptarti metrikų skirtumai yra matomi 3.8 pav. diagramoje pateiktoje apačioje.



3.8 pav. Efektyviausių modelių įverčių palyginimas naudojant 2023 m. žiemos validacijos duomenų rinkinį

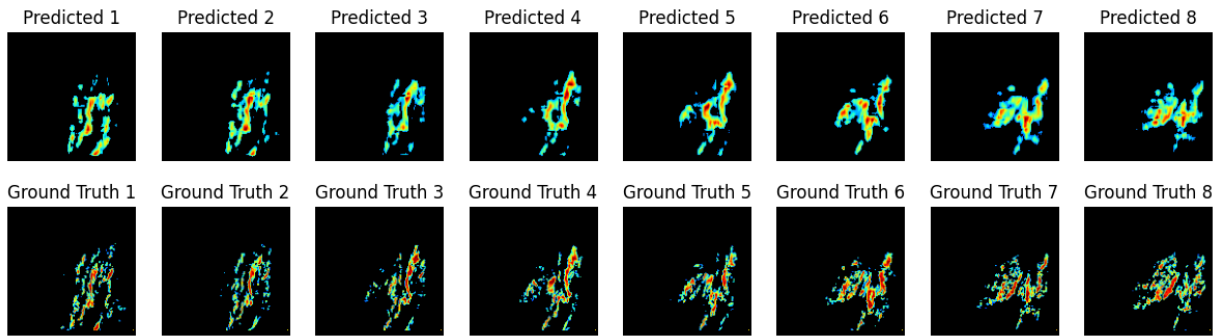
Nors geriausių modelių tendencijos skirtingais analizuojamais scenarijais buvo išlaikomos panašios, lyginant gautus tikslumo įverčius tarp skirtingų duomenų rinkinių tyrimų buvo pastebėta, kai kurių duomenų rinkinių teikiamos prognozės yra prastesnės nei kitų. To priežastis yra skirtingas duomenų rinkinių sudėtingumo lygis, ši tendencija yra pavaizduota 3.9 pav. pateiktoje diagramoje.



3.9 pav. Modelių RMSE įverčiai visais analizuojamais validacijos scenarijais

Prasčiausios modelių prognozės validavimo etape buvo gautos naudojant 2023 m. žiemos ir rugpjūčio mėnesio bei 2022 m. balandžio mėnesio duomenų rinkinius. Rezultatai yra tokie, dėl to jog šiuose rinkiniuose yra vaizduojami netipiniai, į apmokymo rinkinius nepanašūs kritulių reiškiniai. Modeliai sunkiausiai prognozavo rugpjūčio mėnesio kritulių reiškinį, tačiau kaip buvo minėta anksčiau, UNet modeliai šiuose scenarijuose pasiekė geresnius rezultatus nei kiti modeliai. Tai matosi iš 3.9 lentelėje pateiktų paklaidų skirtumų tarp kiekvieno modelio scenarijų rezultatų.

Tačiau nors paklaidų įverčiai tam tikrais eksperimentais buvo žemi, atsižvelgiant iš struktūrinių tikslumo metrikų perspektyvos, efektyviausi modeliai vis tiek sugebėjo pateikti patikimas prognozes, net ir sudėtingų kritulių prognozavimo scenarijais. Tai galima pamatyti vieno iš efektyviausių, ConvLSTM_v2.2_loc, modelio balandžio mėnesio prognozių pavyzdžiuose, kurie yra pateikti 3.10 pav. Šie rezultatai parodo, jog realizuoti modeliai sugeba prognozuoti net ir išsisklaidžiusius, intensyvius kritulių reiškinius.



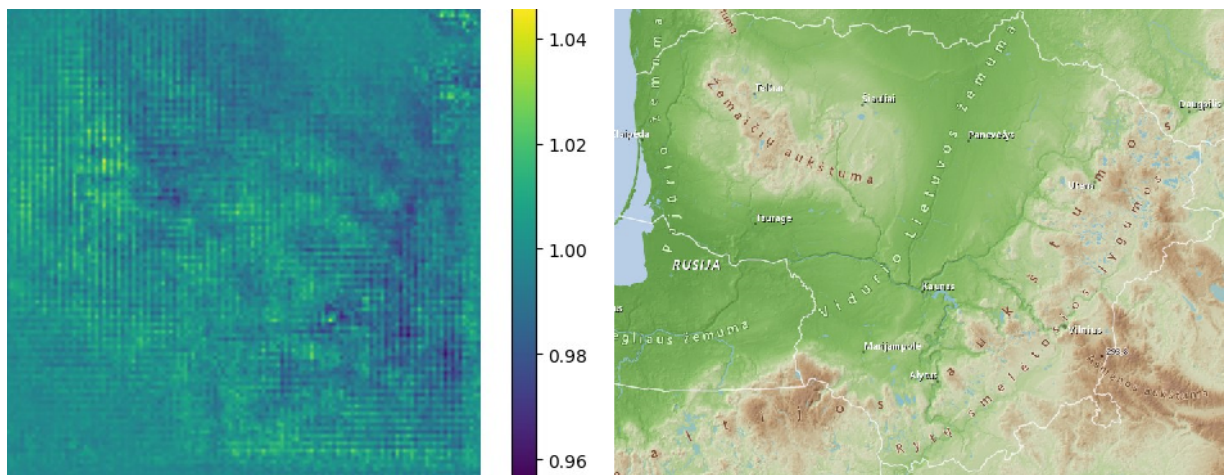
3.10 pav. ConvLSTM_v2.2_loc modelio Balandžio mėnesio kritulių pavyzdinės prognozės

Taigi, apibendrinant validacijos metu atliktus tyrimus, galima teigti jog iš realizuotų modelių geriausios prognozės buvo gautos naudojant UNet ir ConvLSTM modelių versijas su lokalizacijos sluoksniais. Beveik visais atvejais lokalizacijos sluoksnio integracija pagerino šių architektūrų modelių efektyvumą. Efektyviausi modeliai pasižymėjo prisitaikymu prie tam tikrų kritulių reiškinių specifikų: UNet modeliai efektyviau prognozavo didelių audrų, o ConvLSTM pasiskirsčiusių ir intensyvių kritulių reiškinius.

3.2. Lokalizuotų neuroninių tinklų rezultatai

Kaip buvo pabrėžta ankstesnių rezultatų apžvalgoje, modelių lokalizacijos daugeliu analizuotų atvejų pagerino kritulių prognozavimo efektyvumą konvoliuciniuose, UNet ir ConvLSTM modeliuose. Tačiau kiekvieno lokalizuoto modelio išmoktos lokalizacijos sluoksnio reikšmės buvo skirtingos. Atliekant eksperimentus su lokalizuotais neuroniniais tinklais, buvo vizualiai pateikiamos gautos lokalizacijos matricos, kurios plačiau aptariamos šiame skyriuje.

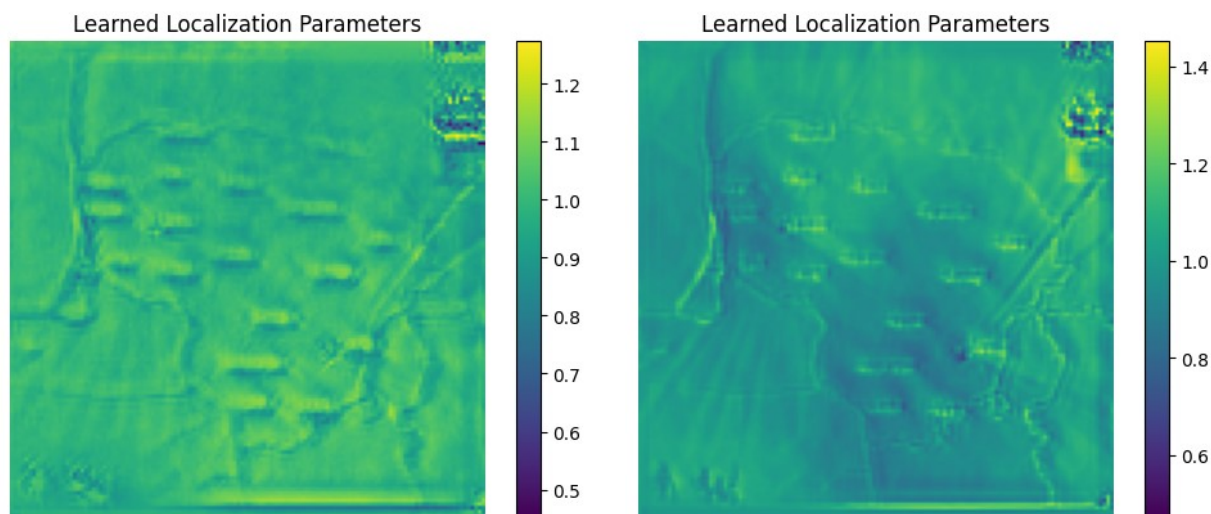
Pirmiausia, buvo atlikta LocalizedCNN_v2.0 modelio lokalizacijos matricos analizė. Kadangi šiame modelyje naudojama matrica, atitinkanti naudojamų radaro nuotraukų dimensijas, rezultatai atitinka Lietuvos regiono geografiją. Šio modelio galutinė lokalizacijos matrica, gauta apmokius modelį su pusės metų duomenų rinkiniu, yra pateikta 3.11 pav.



3.11 pav. localizedCNN_v2.0 modelio išmoktos lokalizacijos matricos ir Lietuvos reljefo palyginimas

Gautoje matricoje galima išvelgti keletą savybių: apatiniame dešiniajame kampe ir viršutinėje kairioje dalyje matomi žemesnių reikšmių rodmenys, o aplink jas yra didesnės reikšmės. Palyginus šį gautą rezultatą su Lietuvos regiono aukščių žemėlapiu, kurie yra pateikti 3.11 pav., galima pastebėti, jog šis modelis sėkmingai identifikavo analizuojamo regiono reljefą. Taip pat gautoje lokalizacijos matricoje galima pastebėti keletą regionų, kur pažymėtos reikšmės yra didesnės (šviesiai žalia spalva pažymėti taškai). Šiose vietose, pasak Lietuvos hidrometeorologijos tarnybos teikiamų duomenų, statistškai per metus iškrenta daugiau kritulių nei likusioje Lietuvos dalyje, todėl galima daryti išvadą, jog išmokta lokalizacijos matrica šiuo atveju atitinka Lietuvos regiono lokalių meteorologinių reiškinių tendencijas.

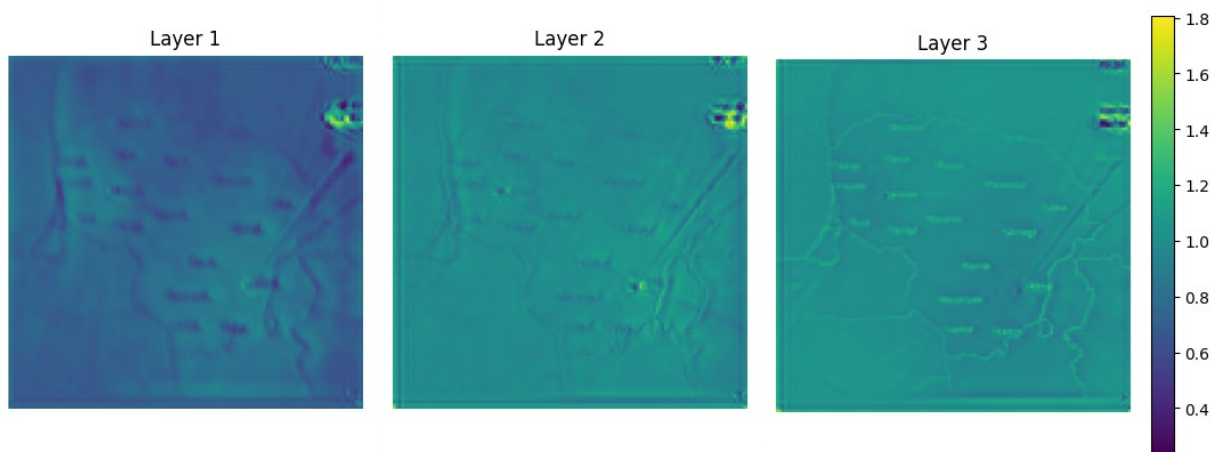
Kitą vertus, atliekant tyrimą su likusiais lokalizuotais modeliais, gautų matricų informacija skyrėsi nuo 3.11 pav. pateiktų rezultatų. Lietuvos regiono reljefo nustatymas buvo pastebimas visose gautose lokalizacijų matricose, tačiau didžioji dalis modelių taip pat išsaugojo valstybinių sienų kontūrus ir tam tikrų artefaktų pozicijas. Šie rezultatai yra pateikiami 3.12 pav., kur yra pavaizduoti konvoliucinio neuroninio tinklo, su dviem lokalizacijos sluoksniais, modelio rezultatai.



3.12 pav. LocalizedCNN_4.2_two_loc_layers modelio lokalizacijos matricos

Šioje iliustracijoje, galima aiškiai matyti Lietuvos valstybinių sienų kontūrus ir tam tikrus artefaktus dešiniuosiuose paveikslėlių kampuose. Analizuojamame duomenų rinkinyje valstybinės sienos yra vaizduojamos virš vaizduojamų kritulių. Kritulių radaro nuotraukų fono šalinimo metu buvo pašalinta visa informacija vietose, kur nėra kritulių, tačiau situacijose kai kritulių informacija vaizduojama virš valstybinių sienų arba virš didžiųjų Lietuvos miestų, ši informacija nebuvo šalinama. Dėl to, lokalizuoti modeliai analizuodami skirtingų pozicijų kritulių informaciją aptiko pasikartojimus ir statinės pozicijos informaciją.

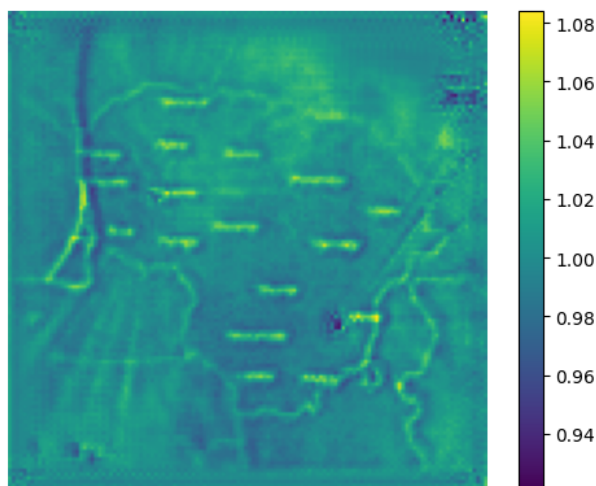
ConvLSMT modelių lokalizacija buvo atlikta įterpiant lokalizacijos matricas į visus tris modelio konvoliucinius blokus. To pasekoje, šis modelis apmokymo metu suformuoja tris skirtingas lokalizacijos matricas, kurios yra pateikiamos 3.13 pav.:



3.13 pav. ConvLSTM modelio lokalizacijų matricos

Palyginus visus tris paveikslėlius galima pastebėti, jog kiekviename sluoksnyje vaizduojamos matricos yra skirtingos, tačiau tam tikra informacija išlaikoma panaši. Pirmajame sluoksnyje ryškiausiai matomos reljefo ir vyraujančių kritulių savybės, antrasis ir trečiasis sluoksniai vaizduoja panašią informaciją, tačiau trečiajame ryškiausiai matomos valstybinės sienos bei miestų pavadinimai.

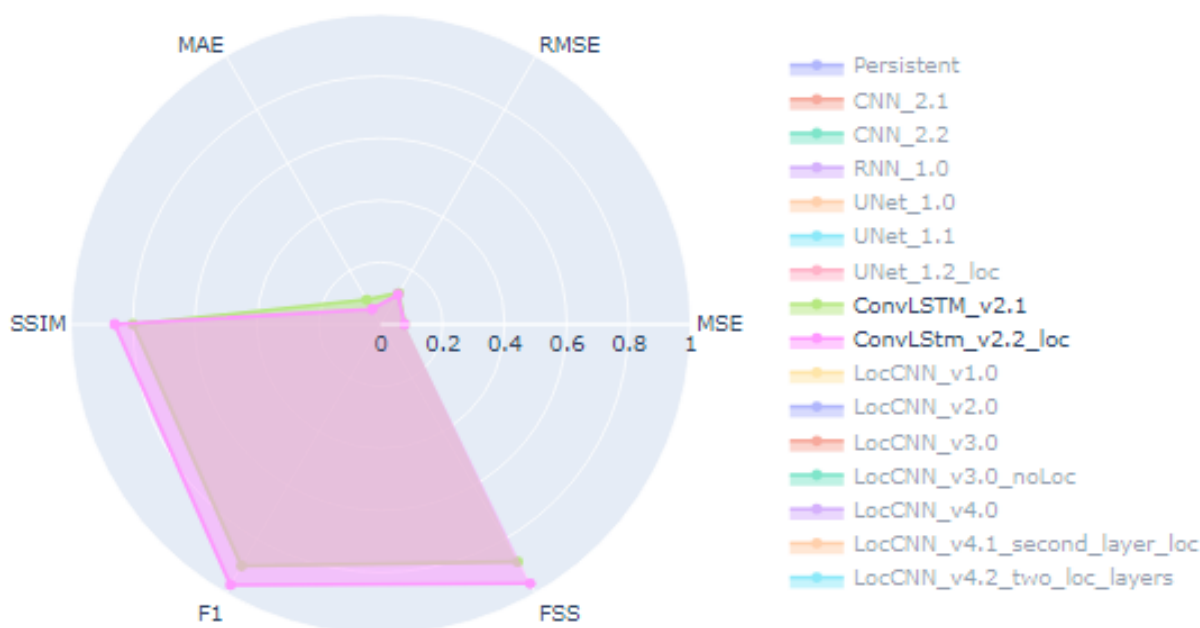
UNet modelio lokalizacija buvo atlikta į UNet_v1.1 neuroninio tinklo pirmąjį sluoksnį įterpiant lokalizuotąjį sluoksnį. Šiuo atveju buvo realizuota tik viena lokalizacijos matrica, tačiau šiuo modelio patobulinimo būdu kritulių prognozavimo rezultatai, beveik visais analizuojamais atvejais, buvo tikslesni. Taip pat pastebėta, nors šioje architektūroje lokalizacija buvo pritaikoma tik viename sluoksnyje, rezultatų matrica atitinka visas, anksčiau aptartas, vaizduojamos informacijos tendencijas. 3.14 pav. pateiktoje rezultatų matricoje galima pastebėti analizuojamo regiono reljefo, valstybinių sienų, miestų pavadinimų ir vyraujančių kritulių lokalią informaciją.



3.14 pav. UNet modelio lokalizacijos matrica

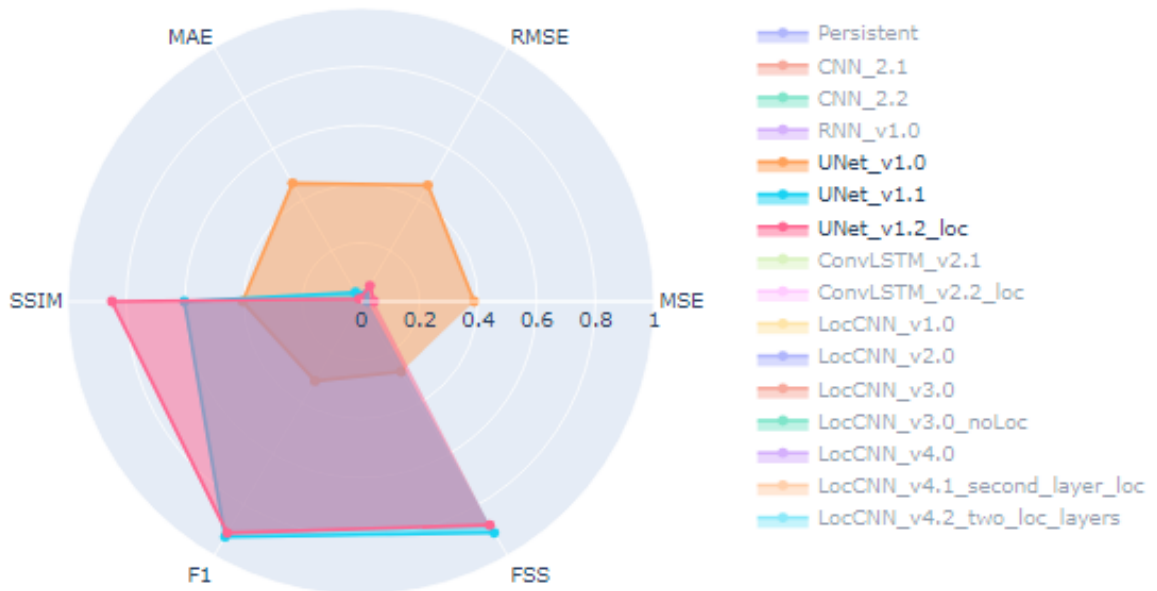
Tarpusavyje palyginus skirtingų lokalizuotų neuroninių tinklų gautus lokalizacijos matricos rezultatus galima teigti, jog šis metodas padeda nustatyti lokalią svarbą informaciją, aktualią laiko ir erdvės prognozavimo aspektais. Efektyviausių UNet ir ConvLSTM modelių gautose rezultatų matricose ši lokali informacija buvo matoma ryškiausiai. Remiantis šia bei validacijos rezultatų metu padarytomis išvadomis, galima teigti, jog lokalizuoti sluoksniai atlieką svarbią rolę trumpalaikių kritulių prognozavimo uždavinyje ir jų naudojimas gali pagerinti modelių efektyvumą.

Siekiant išsiaiškinti lokalizacijos sluoksnių naudą modelių prognozių efektyvumui, buvo sudaryti skirtingų modelių rezultatus lyginantys grafikai. 3.15 pav. pateiktoje diagramoje yra pateikiami ConvLSTM modelio ir jo lokalizuotos versijos skirtingų įverčių palyginimai.



3.15 pav. ConvLSTM modelio lokalizuotos versijos rezultatų palyginimas su įprastos architektūros modeliu

Diagramoje galima pastebėti, jog lokalizuota modelio versija visais vertinamais aspektais atliko tikslesnes kritulių prognozes. Palyginus šiuos rezultatus galima daryti išvadą, jog modelio išmoktų lokaliųjų sluoksnių informacija yra reikšminga ir prisideda prie teikiamų prognozių gerinimo. Tokiu pat būdu buvo palyginti ir UNet architektūros modelių rezultatai, kurie yra pateikiami 3.16



3.16 pav. UNet modelių skirtingų versijų rezultatų palyginimasis grafikas

Šiuo atveju galima pastebėti, jog lokalizuota modelio versija pralenkė UNet_v1.0 modelio prognozių tikslumą visais vertinamais kriterijais, tačiau UNet_v1.1 modelis išliko efektyvesnis FSS ir F1 tikslumo įverčių atžvilgiu. Kitą vertus, lokalizacijos metodo pagalba modelio pasiekta SSIM reikšmė yra ženkliai geresnė nei UNet_v1.1 modelio, o tai reiškia jog lokalizacijos sluoksnio pagalba modelis sugebėjo tiksliau nustatyti kritulių struktūrinę pasiskirstymą analizuojamame regione.

Ištyrus efektyviausių modelių rezultatus, gautus su lokalizuotomis ir įprastomis modelių versijomis buvo pastebėta, jog daugeliu aspektų teikiamų prognozių tikslumas buvo geresnis naudojant modelius su lokalizuotais sluoksniais. Tai įrodo, jog šiame skyriuje aptarta, lokalizuotų sluoksnių išmokta informacija yra vertinga ir padeda gerinti trumpalaikių kritulių prognozių efektyvumą.

4. Išvados

Magistro darbe buvo sėkmingai sukurti ir pritaikyti skirtingų tipų neuroninių tinklų modeliai skirti trumpalaikiam kritulių prognozavimui. Geriausius rezultatus pasiekė UNet ir ConvLSTM architektūrų modeliai, o konvoliucinių sluoksnių lokalizacijos metodai pagerino vidutinius tikslumo įverčius visuose modeliuose su konvoliuciniais sluoksniais. Žemiau pateiktos pagrindinės duomenų apdorojimo, modelių rezultatų ir modelių lokalizacijos išvados.

1. Kritulių radaro nuotraukų duomenų rinkinys
 - 1.1 Sėkmingai sukurta automatizuota duomenų rinkimo ir apdorojimo sistema, kurios pagalba sudarytas daugiau nei 50 tūkst. kritulių radaro nuotraukų duomenų rinkinys, padengiantis laikotarpį nuo 2022 m. kovo 1 d. iki 2023 m. gruodžio 3 d.
 - 1.2 Realizuoti Python programiniai kodai, kurių pagalba duomenų rinkinys buvo paruoštas giliojo mašininio mokymosi uždaviniams išspręsti. Šio proceso metu sėkmingai pašalinta nereikalinga vaizdinė informacija iš visų paveikslėlių, paliekant tik kritulių duomenis, aktualiausius mašininio mokymosi modeliams.
 - 1.3 Didžiausius iššūkius sukėlė dėl kritulių radaro veikimo trikdžių praleistos reikšmės bei techninių radaro sutrikimų užfiksuoti vaizdiniai artefaktai. Šios problemos buvo išspręstos pasitelkiant realizuotus Python programinius kodus bei duomenų rinkinį apdorojant rankiniu būdu.
 - 1.4 Sunkiausia buvo prognozuoti 2022 m. rugpjūčio ir balandžio mėnesių duomenų rinkiniuose vaizduojamus kritulių reiškinius. Šiais laikotarpiais buvo analizuojamas didžiausias meteorologinių reiškinių intensyvumas ir variacija, todėl tiksliai prognozuoti kritulius modeliams buvo sudėtingiausia.
2. Nelokaluotų modelių apmokymo ir prognozavimo rezultatų palyginamosios analizės išvados
 - 2.1 Atlikus eksperimentų rezultatų analizę, buvo nustatyta, kad tiksliausios kritulių prognozės beveik visais analizuojamais scenarijais gautos pasitelkiant UNet ir ConvLSTM architektūrų modelius.
 - 2.2 UNet modelių prognozių vidutiniai SSIM įverčiai buvo 36,7 % didesni, o MSE rodikliai 54,8 % mažesni, lyginant su visų likusių realizuotų modelių vidutiniais tikslumo įverčiais. UNet modeliai pasižymėjo ypač gerais rezultatais prognozuojant didelių audrų ir visame analizuojamame regione pasiskirsčiusių intensyvių kritulių reiškinius.
 - 2.3 ConvLSTM modeliai pasiekė vidutiniškai 31,9 % geresnius SSIM įverčius ir 40,4 % mažesnius MSE rodiklius nei kiti modeliai. Šios architektūros modeliai efektyviausiai prognozavo netolygius, vidutinio intensyvumo kritulių reiškinius.
 - 2.4 Atlikus skirtingų duomenų rinkinių prognozių analizę, buvo nustatyta, kad efektyviausias buvo lokalizuotas ConvLSTM_v2.2_loc modelis, o prasčiausias – rekurentinis RNN_1.0 tinklas. Šiam ConvLSTM architektūros modeliui gautas vidutinis MSE įvertis lygus 0,00256, o rekurentiniam – 59,1 % mažesnis (0,00626). ConvLSTM modeliui gauta vidutinė SSIM siekia 0,7367, o tai yra net 315,6 % daugiau nei RNN_1.0 modeliui gauta reikšmė (0,1772).

3. Rezultatų pagerinimas pritaikant lokalizuotus konvoliucinius sluoksnius
 - 3.1 Analizuojant lokalizuotų neuroninių tinklų architektūras, buvo pastebėta, kad naudojant lokalizacijos sluoksnius buvo sėkmingai nustatyta lokaliai svarbi informacija, tokia kaip Lietuvos regiono reljefas ar vyraujančių kritulių pasiskirstymas.
 - 3.2 Vidutinė lokalizuoto ConvLSTM modelio MSE reikšmė lygi 0,00236, o tai yra 24,3 % mažiau, nei įprastos architektūros ConvLSTM modelio pasiektas įvertis (0,00312).
 - 3.3 Ištyrus lokalizuoto ir įprasto UNet modelių rezultatus taip pat pastebėta lokalizacijos nauda modelio efektyvumui. Lokalizuoto UNet modeliui gauta vidutinė MSE reikšmė lygi 0,00277, o tai yra 22,6 % mažiau, nei MSE paklaida, gauta įprastos architektūros tinklui (0,00358). SSIM vidurkis lokalizuotame UNet_1.2_loc modelyje siekė 0,6467, tai yra 42,9 % daugiau nei įprastos architektūros modelio SSIM – 0,4526.
 - 3.4 Pritaikius lokalizacijos metodą CNN architektūros modeliuose taip pat buvo pastebėta reikšminga nauda. Lokalizuoto CNN vidutinė MSE reikšmė buvo 0,00296, kas yra 27,5 % mažiau nei įprastos architektūros CNN modelio vidutinė MSE reikšmė (0,00408). Be to, lokalizuotų CNN modelių SSIM vidurkis buvo 0,6212, tai yra 34,3 % daugiau nei įprastos architektūros CNN modelių SSIM vidurkis (0,4626).
 - 3.5 Šie rezultatai rodo, kad visose architektūrose, kuriose yra naudojami konvoliuciniai sluoksniai, lokalizacija pagerina vidutinius rezultatus.

Magistro darbe pristatyti neuroninių tinklų modeliai ir lokalizacijos metodai gali tapti esminiu įrankiu, padedančiu tiksliau prognozuoti trumpalaikius kritulius. Tyrimo rezultatai rodo, kad lokalizuotos architektūros, tokios kaip ConvLSTM_v2.2_loc, yra ypač efektyvios, lyginant su tradiciniais modeliais. Tačiau šiame tyrime analizuotos tik ribotos meteorologinių duomenų sąlygos, todėl ateityje būtų galima tyrimą praplėsti, įtraukiant kitus meteorologinius reiškinius, tokius kaip vėjo greitis ar temperatūros svyravimai. Be to, būtų naudinga iširti kitų lokalizacijos metodų potencialą, kurie galėtų dar labiau pagerinti prognozių tikslumą. Tai ne tik leistų sukurti tikslesnius ir patikimesnius kritulių prognozavimo modelius, bet ir atskleistų įvairių neuroninių tinklų architektūrų pritaikymo galimybes meteorologijos srityje. Ateities tyrimuose taip pat būtų prasminga išbandyti šią metodiką kitų regionų meteorologiniams duomenims, siekiant įvertinti modelių universalumą ir pritaikomumą platesniu mastu. Tai galėtų padėti meteorologijos tarnyboms ir kitiems suinteresuotiems subjektams geriau pasirengti ir reaguoti į ekstremalius oro reiškinius.

Literatūros sąrašas

1. NASA. *The Effects of Climate Change*. 2022-01-05. Prieiga per: <https://climate.nasa.gov/effects/>.
2. UNFCCC. *Climate Change Leads to More Extreme Weather, but Early Warnings Save Lives* [interaktyvus]. 2022-01-05. [žiūrėta 2021-09-01]. Pasielkiamas per: <https://unfccc.int/news/climate-change-leads-to-more-extreme-weather-but-early-warnings-save-lives>.
3. BBC. *Climate change: Science failed to predict flood and heat intensity* [interaktyvus]. 2022-01-05. [žiūrėta 2021-06-16]. Pasielkiamas per: <https://www.bbc.com/news/science-environment-57863205>.
4. LEARN, Simpli. *Top 9 New Technology Trends for 2022* [interaktyvus]. 2021-01-06. [žiūrėta 2021-12-07]. Pasielkiamas per: <https://www.simplilearn.com/top-technology-trends-and-jobs-article>.
5. WASHINGTON, University of. *A.I. model shows promise to generate faster, more accurate weather forecasts* [interaktyvus]. [žiūrėta 2020-12-15]. Pasielkiamas per: <https://www.washington.edu/news/2020/12/15/a-i-model-shows-promise-to-generate-faster-more-accurate-weather-forecasts/>.
6. DIRECT, Science. *Nowcasting* [interaktyvus]. 2021-01-06. [žiūrėta 2019]. Pasielkiamas per: <https://www.sciencedirect.com/topics/earth-and-planetary-sciences/nowcasting>.
7. Dynamical cores for NWP: An uncertain landscape. 2021. Prieiga per: <https://www.sciencedirect.com/science/article/abs/pii/B978012815491500001X?via%3Dihub>.
8. The use of 100m scale NWP models to understand differences between different measures of mixing height in a morning growing clear convective boundary layer over London. 2021. Prieiga per: <https://rmets.onlinelibrary.wiley.com/doi/10.1002/qj.4291>.
9. MISHRA, M. *Convolutional Neural Networks, Explained*. 2020-08-26. Prieiga per: <https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939>.
10. NABI, J. *Recurrent Neural Networks (RNNs)*. 2019-07-12. Prieiga per: <https://towardsdatascience.com/recurrent-neural-networks-rnns-3f06d7653a85>.
11. PRA, M. D. *Generative Adversarial Networks*. 2023-10-30. Prieiga per: <https://medium.com/@marcodelpira/generative-adversarial-networks-dba10e1b4424>.
12. A Hybrid CNN and RNN Variant Model for Music Classification. 2023. Prieiga per: <https://www.mdpi.com/2076-3417/13/3/1476>.
13. DODLA, Venkata Bhaskar Rao. *Numerical Weather Prediction*. CRC Press, 2022. ISBN 9781003354017.
14. SPIRIDONOV V., Ćurić M. *Weather Forecast and NWP*. Springer, Cham, 2020. ISBN 978-3-030-52654-2.
15. A. ROJAS - CAMPOS M. Langguth, M. Wittenbrink ir G. Pipa. Deep learning models for generation of precipitation maps based on Numerical Weather Prediction. *EGUsphere*. 2022, p. 20. Prieiga per: <https://doi.org/10.5194/egusphere-2022-648>.

16. Forecasting of monsoon heavy rains: challenges in NWP. 2016. Prieiga per: <https://www.doi.org/10.1117/12.2223646>.
17. Space–time calibration of wind speed forecasts from regional climate models. 2021. Prieiga per: <https://link.springer.com/article/10.1007/s10651-021-00509-0>.
18. AL., V. B. Morozov et. Adapting to climate change through anticipatory action: The potential use of weather-based early warnings. *Weather and Climate Extremes*. 2022, p. 8. Prieiga per: <https://doi.org/10.1016/j.wace.2022.100508>.
19. Improvements in deep learning-based precipitation nowcasting using major atmospheric factors with radar rain rate. 2024. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S0098300424000128>.
20. Characterization and nowcasting of severe weather events over Milano Malpensa. 2022. Prieiga per: <https://meetingorganizer.copernicus.org/EGU22/EGU22-9357.html>.
21. V. NIETOSVAARA, E. De Coning ir. Guidelines for Satellite-based Nowcasting in Africa warnings. *ESSL*. 2023. Prieiga per: <https://doi.org/10.5194/ecss2023-95>.
22. Extreme weather impacts of climate change: an attribution perspective. 2022. Prieiga per: <https://iopscience.iop.org/article/10.1088/2752-5295/ac6e7d>.
23. Trends of extreme US weather events in the changing climate. 2022. Prieiga per: <https://www.pnas.org/doi/10.1073/pnas.2207536119>.
24. *Climate Change and Extreme Weather Events*. CRC Press, 2022. Prieiga per: <https://www.taylorfrancis.com/chapters/edit/10.1201/9781003364689-5/climate-change-extreme-weather-events-sandeep-pattnaik>.
25. Early Warning Weather Hazard System for Power System Control. 2022. Prieiga per: <https://www.mdpi.com/1996-1073/15/6/2085>.
26. A Civil Protection Early Warning System to Improve the Resilience of Adriatic-Ionian Territories to Natural and Man-made Risk. 2022. Prieiga per: <https://ieeexplore.ieee.org/document/9921697>.
27. Towards the development of an AI-based early warning system: a deep learning approach to bias correct and downscale seasonal climate forecasts. 2023. Prieiga per: <https://meetingorganizer.copernicus.org/EGU23/EGU23-9555.html>.
28. N. ROMBEEK, Jj Leinonen ir U. Hamann. Nowcasting thunderstorm hazards with radar polarimetry using deep learning. *ESSL*. 2023. Prieiga per: <https://doi.org/10.5194/ecss2023-59>.
29. Y. CHEN, Y. Liu ir. Overview of Deep Learning Research. *Atlantis press*. 2019. Prieiga per: <https://www.atlantis-press.com/proceedings/sschd-19/125925738>.
30. Deep Learning for Day Forecasts from Sparse Observations. 2023. Prieiga per: <https://arxiv.org/abs/2306.06079>.
31. Deep learning approach for one-hour ahead forecasting of weather data. 2023. Prieiga per: <https://www.tandfonline.com/doi/full/10.1080/15567036.2023.2222690>.
32. Diving deep in Deep Convolutional Neural Network. 2020. Prieiga per: <https://ieeexplore.ieee.org/document/9362907>.
33. CNN study of convolutional neural networks in classification and feature extraction applications. 2020. Prieiga per: <https://doi.org/10.1117/12.2560372>.

34. An Overview and Application of Deep Convolutional Neural Networks for Medical Image Segmentation. 2023. Prieiga per: <https://ieeexplore.ieee.org/document/10073857>.
35. J. RAZ, J. V. Abayon ir C. Peña. A performance evaluation of convolutional and recurrent neural network on Philippine typhoon data. *AIP Publishing*. 2021. Prieiga per: <https://pubs.aip.org/aip/acp/article-abstract/2508/1/020013/2878820/A-performance-evaluation-of-convolutional-and?redirectedFrom=fulltext>.
36. H. HAACHEMI, L. Hamad ir O. Ddouba. Precipitation Nowcasting using CNN-RNN - Use case: North of Algeria. *Research square*. 2023. Prieiga per: <https://www.researchsquare.com/article/rs-2677640/v1>.
37. A. ROJAS-CAMPOS M. Langguth, M. Wittenbrink ir G. Pipa. Deep learning models for generation of precipitation maps based on numerical weather prediction. *European geosciences union*. 2023. Prieiga per: <https://gmd.copernicus.org/articles/16/1467/2023/>.
38. H. ZHOU, D. Schertzer ir I. Tchiguirinskaia. Deep Learning and Universal Multifractal for Nowcasting Precipitation in Urban Geosciences. *EGU General Assembly 2023*. 2023. Prieiga per: <https://meetingorganizer.copernicus.org/EGU23/EGU23-768.html>.
39. N. K. SABAT R. Nayak, U. C. Pati ir S. K. Das. A Comparative Analysis of Univariate Deep Learning-based Time-series Models for Temperature Forecasting of the Bhubaneswar. *IEEE*. 2022. Prieiga per: <https://ieeexplore.ieee.org/document/10051494>.
40. XAVIER, A. *An introduction to ConvLSTM*. 2019-03-25. Prieiga per: <https://medium.com/neuronio/an-introduction-to-convlstm-55c9025563a7>.
41. AL., D. Kong et. Precipitation Nowcasting Based on Deep Learning over Guizhou, China. *MDPI*. 2023. Prieiga per: <https://www.mdpi.com/2073-4433/14/5/807>.
42. A. USELIS, M. Lukoševičius ir L. Stasytis. Localized Convolutional Neural Networks for Geospatial Wind Forecasting. *MDPI*. 2020. Prieiga per: <https://doi.org/10.3390/en13133440>.
43. H. T. KUNG, M. Comiter ir. Localization Convolutional Neural Networks Using Angle of Arrival Images. *IEEE*. 2018. Prieiga per: <https://ieeexplore.ieee.org/document/8647687>.
44. J. GAO, B. Peng ir Y. Yang. A Localized Neural Network with Dependent Data: Estimation and Inference. *arXiv*. 2023. Prieiga per: <https://doi.org/10.48550/arXiv.2306.05593>.
45. Learn To Pay Attention. 2018. Prieiga per: <https://arxiv.org/abs/1804.02391>.
46. SPRINGBOARD. *What Is the Best Language for Machine Learning?* [interaktyvus]. 2022-03-21. [žiūrėta 2021]. Pasiiekiamas per: <https://www.springboard.com/blog/data-science/best-language-for-machine-learning/>.
47. GEEKS, Geeks for. *Top 5 Programming Languages and their Libraries for Machine Learning in 2020* [interaktyvus]. 2022-03-21. [žiūrėta 2021-11-06]. Pasiiekiamas per: <https://www.geeksforgeeks.org/top-5-programming-languages-and-their-libraries-for-machine-learning-in-2020/>.

48. GEEKS, Geeks for. *Python Programming Language* [interaktyvus]. 2022-03-22. [žiūrėta 2022-06-23]. Pasielkiamas per: <https://www.geeksforgeeks.org/python-programming-language/>.
49. SPRINGBOARD. *Which is Better for AI: Java or Python*. 2022-03-24. Prieiga per: <https://www.springboard.com/blog/data-science/which-is-better-for-ai-java-or-python/>.
50. DATAFLAIR. *Advantages and disadvantages of C++* [interaktyvus]. 2022-03-21. [žiūrėta 2021-11-06]. Pasielkiamas per: <https://data-flair.training/blogs/advantages-and-disadvantages-of-cpp/>.
51. DIGITAL, DAC. *Advantages and Disadvantages of C++ | Make your Next Move* [interaktyvus]. 2022-03-24. [žiūrėta 2020]. Pasielkiamas per: <https://dac.digital/which-is-better-for-ai-python-or-r/>.
52. PAGES, GitHub. *Tesseract documentation*. 2021-12-20. Prieiga per: <https://tesseract-ocr.github.io/>.
53. GATE, Research. *Tesseract architecture*. 2024-05-09. Prieiga per: https://www.researchgate.net/publication/328997563_From_Videos_to_URLs_A_Multi-Browser_Guide_To_Extract_User's_Behavior_with_Optical_Character_Recognition.

Priedai

1 PRIEDAS. Realizuotų modelių architektūros

```
=====  
Layer (type:depth-idx)          Param #  
=====  
PrecipitationCNN                --  
├─Conv2d: 1-1                   160  
├─MaxPool2d: 1-2                --  
├─Conv2d: 1-3                   4,640  
├─Linear: 1-4                   16,777,728  
├─Linear: 1-5                   8,404,992  
=====  
Total params: 25,187,520  
Trainable params: 25,187,520  
Non-trainable params: 0  
=====
```

CNN_v2.1 modelio struktūra

```
=====  
Layer (type:depth-idx)          Param #  
=====  
PrecipitationCNN                --  
├─Conv2d: 1-1                   160  
├─Conv2d: 1-2                   2,320  
├─Conv2d: 1-3                   145  
=====  
Total params: 2,625  
Trainable params: 2,625  
Non-trainable params: 0  
=====
```

CNN_v2.2 modelio struktūra

```
=====  
Layer (type:depth-idx)          Param #  
=====  
PrecipitationRNN                --  
├─RNN: 1-1                      8,651,776  
├─Linear: 1-2                   8,404,992  
=====  
Total params: 17,056,768  
Trainable params: 17,056,768  
Non-trainable params: 0  
=====
```

RNN_v1.0 modelio struktūra

```

=====
Layer (type:depth-idx)                Param #
=====
PrecipitationNowcastingNet             --
├─ConvLSTM: 1-1                       --
│   └─ModuleList: 2-1                 --
│       └─ConvLSTMCell: 3-1           9,856
│           └─ConvLSTMCell: 3-2       55,424
│               └─ConvLSTMCell: 3-3   221,440
├─Conv2d: 1-2                          65
=====
Total params: 286,785
Trainable params: 286,785
Non-trainable params: 0
=====

```

ConvLSTM_v2.1 modelio struktūra

```

=====
Layer (type:depth-idx)                Param #
=====
PrecipitationNowcastingNet             --
├─ConvLSTM: 1-1                       --
│   └─ModuleList: 2-1                 --
│       └─ConvLSTMCell: 3-1           26,816
│           └─ConvLSTMCell: 3-2       72,960
│               └─ConvLSTMCell: 3-3   240,128
├─Conv2d: 1-2                          65
=====
Total params: 339,969
Trainable params: 339,969
Non-trainable params: 0
=====

```

Localizuoto ConvLSTM_v2.2 modelio struktūra

```

=====
Layer (type:depth-idx)                Param #
=====
PrecipitationRNN                       --
├─RNN: 1-1                             8,651,776
├─Linear: 1-2                          8,404,992
=====
Total params: 17,056,768
Trainable params: 17,056,768
Non-trainable params: 0
=====

```

UNet_v1.0 modelio struktūra

Layer (type:depth-idx)	Param #
PrecipitationUNet	--
└DoubleConv: 1-1	--
└Sequential: 2-1	--
└Conv2d: 3-1	640
└BatchNorm2d: 3-2	128
└ReLU: 3-3	--
└Conv2d: 3-4	36,928
└BatchNorm2d: 3-5	128
└ReLU: 3-6	--
└Conv2d: 3-7	36,928
└BatchNorm2d: 3-8	128
└ReLU: 3-9	--
└Down: 1-2	--
└Sequential: 2-2	--
└MaxPool2d: 3-10	--
└DoubleConv: 3-11	369,792
└Down: 1-3	--
└Sequential: 2-3	--
└MaxPool2d: 3-12	--
└DoubleConv: 3-13	1,476,864
└Down: 1-4	--
└Sequential: 2-4	--
└MaxPool2d: 3-14	--
└DoubleConv: 3-15	5,902,848
└Up: 1-5	--
└ConvTranspose2d: 2-5	524,544
└DoubleConv: 2-6	--
└Sequential: 3-16	2,361,600
└Up: 1-6	--
└ConvTranspose2d: 2-7	131,200
└DoubleConv: 2-8	--
└Sequential: 3-17	590,976
└Up: 1-7	--
└ConvTranspose2d: 2-9	32,832
└DoubleConv: 2-10	--
└Sequential: 3-18	148,032
└OutConv: 1-8	--
└Conv2d: 2-11	65

Total params: 11,613,633	
Trainable params: 11,613,633	
Non-trainable params: 0	

UNet_v1.1 modelio struktūra

```

=====
Layer (type:depth-idx)                               Param #
=====
PrecipitationUNet                                     --
├─DoubleConv: 1-1                                     --
│   └─Sequential: 2-1                                 --
│       └─LocalizedConv2d: 3-1                       17,600
│           └─BatchNorm2d: 3-2                       128
│               └─ReLU: 3-3                           --
├─Down: 1-2                                           --
│   └─Sequential: 2-2                                 --
│       └─MaxPool2d: 3-4                             --
│           └─DoubleConv: 3-5                       221,952
├─Down: 1-3                                           --
│   └─Sequential: 2-3                                 --
│       └─MaxPool2d: 3-6                             --
│           └─DoubleConv: 3-7                       886,272
├─Down: 1-4                                           --
│   └─Sequential: 2-4                                 --
│       └─MaxPool2d: 3-8                             --
│           └─DoubleConv: 3-9                       3,542,016
├─Up: 1-5                                              --
│   └─ConvTranspose2d: 2-5                           524,544
│       └─DoubleConv: 2-6                             --
│           └─Sequential: 3-10                       1,771,008
├─Up: 1-6                                              --
│   └─ConvTranspose2d: 2-7                           131,200
│       └─DoubleConv: 2-8                             --
│           └─Sequential: 3-11                       443,136
├─Up: 1-7                                              --
│   └─ConvTranspose2d: 2-9                           32,832
│       └─DoubleConv: 2-10                           --
│           └─Sequential: 3-12                       110,976
├─OutConv: 1-8                                        --
│   └─Conv2d: 2-11                                   65
=====
Total params: 7,681,729
Trainable params: 7,681,729
Non-trainable params: 0
=====

```

Localizuoto UNet_v1.2 modelio struktūra

```

=====
Layer (type:depth-idx)                               Param #
=====
LocalizedCNN                                          --
├─LocalizedConv2d: 1-1                               32
│   └─Conv2d: 2-1                                    320
├─Conv2d: 1-2                                         9,248
├─MaxPool2d: 1-3                                     --
├─Conv2d: 1-4                                         289
=====
Total params: 9,889
Trainable params: 9,889
Non-trainable params: 0
=====

```

Localized_CNN_v1.0 modelio struktūra

```

=====
Layer (type:depth-idx)          Param #
=====
LocalizedCNN                    --
├─LocalizedConv2d: 1-1         524,288
│   └─Conv2d: 2-1             320
├─Conv2d: 1-2                 9,248
├─MaxPool2d: 1-3              --
└─Conv2d: 1-4                 289
=====
Total params: 534,145
Trainable params: 534,145
Non-trainable params: 0
=====

```

Localized_CNN_v2.0 modelio struktūra

```

=====
Layer (type:depth-idx)          Param #
=====
LocalizedCNN                    --
├─Conv2d: 1-1                 320
├─Conv2d: 1-2                 9,248
├─MaxPool2d: 1-3              --
└─Conv2d: 1-4                 289
=====
Total params: 9,857
Trainable params: 9,857
Non-trainable params: 0
=====

```

Localized_CNN_v3.0 su pašalintu lokalizacijos sluoksniu modelio struktūra

```

=====
Layer (type:depth-idx)          Param #
=====
LocalizedCNN                    --
├─LocalizedConv2d: 1-1         16,384
│   └─Conv2d: 2-1             608
├─Conv2d: 1-2                 9,248
├─MaxPool2d: 1-3              --
└─Conv2d: 1-4                 289
=====
Total params: 26,529
Trainable params: 26,529
Non-trainable params: 0
=====

```

Localized_CNN_v3.0 modelio struktūra


```

=====
Layer (type:depth-idx)                Param #
=====
LocalizedCNN                          --
├─LocalizedConv2d: 1-1                 16,384
│   └─Conv2d: 2-1                     608
├─Conv2d: 1-2                         9,248
└─Conv2d: 1-3                         289
=====
Total params: 26,529
Trainable params: 26,529
Non-trainable params: 0
=====

```

Localized_CNN_v4.0 modelio struktūra

```

=====
Layer (type:depth-idx)                Param #
=====
LocalizedCNN                          --
├─Conv2d: 1-1                         320
├─LocalizedConv2d: 1-2                 16,384
│   └─Conv2d: 2-1                     9,536
└─Conv2d: 1-3                         289
=====
Total params: 26,529
Trainable params: 26,529
Non-trainable params: 0
=====

```

Localized_CNN_v4.1 modelio struktūra

```

=====
Layer (type:depth-idx)                Param #
=====
LocalizedCNN                          --
├─LocalizedConv2d: 1-1                 16,384
│   └─Conv2d: 2-1                     608
├─LocalizedConv2d: 1-2                 16,384
│   └─Conv2d: 2-2                     19,072
├─Conv2d: 1-3                         18,464
└─Conv2d: 1-4                         289
=====
Total params: 71,201
Trainable params: 71,201
Non-trainable params: 0
=====

```

Localized_CNN_v4.2 modelio struktūra

2 PRIEDAS. Modelių prognozių rezultatai

Modelis	MSE	RMSE	MAE	SSIM	F1	FSS
Persistent	0.0056	0.0746	0.041	0.3173	0.7401	0.5877
CNN_2.1	0.0065	0.0805	0.0438	0.3113	0.6976	0.536
CNN_2.2	0.0051	0.0715	0.0264	0.6635	0.799	0.6655
RNN_1.0	0.0079	0.0886	0.0569	0.1555	0.6671	0.5009
UNet_1.0	0.0034	0.0582	0.0258	0.5096	0.8627	0.7587
UNet_1.1	0.0032	0.0564	0.0224	0.7424	0.8682	0.7673
UNet_1.2_loc	0.0027	0.0517	0.0192	0.7994	0.8682	0.7672
ConvLSTM_v2.1	0.0031	0.056	0.0231	0.8122	0.859	0.7529
ConvLstm_v2.2_loc	0.0031	0.0558	0.0218	0.7085	0.8622	0.7579
LocCNN_v1.0	0.0051	0.0715	0.0291	0.7078	0.7839	0.6448
LocCNN_v2.0	0.0043	0.0655	0.0258	0.7453	0.8056	0.6747
LocCNN_v3.0	0.0042	0.0648	0.0271	0.6474	0.8163	0.6897
LocCNN_v3.0_noLoc	0.0051	0.0712	0.0271	0.7565	0.795	0.6598
LocCNN_v4.0	0.0047	0.0682	0.0254	0.7728	0.7963	0.6617
LocCNN_v4.1_second_layer_loc	0.0047	0.0682	0.0254	0.7728	0.7963	0.6617
LocCNN_v4.2_two_loc_layers	0.005	0.0703	0.0261	0.779	0.7938	0.6583

Modelių prognozių tikslumo įverčiai gauti atliekant 2 valandų prognozavimą

Modelis	MSE	RMSE	MAE	SSIM	F1	FSS
Persistent	0.0075	0.08	0.025	0.7115	0.2221	0.1301
CNN_2.1	0.0035	0.0546	0.0254	0.4215	0.3259	0.2123
CNN_2.2	0.0048	0.0634	0.0239	0.4803	0.2912	0.1805
RNN_v1.0	0.0062	0.0734	0.0622	0.234	0.256	0.1438
UNet_v1.0	0.0031	0.0516	0.0221	0.298	0.407	0.2716
UNet_v1.1	0.0029	0.0489	0.0198	0.356	0.421	0.321
UNet_v1.2_loc	0.0039	0.0492	0.0189	0.432	0.451	0.321
ConvLSTM_v2.1	0.0028	0.0494	0.0207	0.48	0.4637	0.3202
ConvLSTM_v2.2_loc	0.0027	0.0485	0.0184	0.5337	0.475	0.331
LocCNN_v1.0	0.0038	0.0567	0.0228	0.6739	0.3171	0.202
LocCNN_v2.0	0.0036	0.0557	0.0215	0.7007	0.324	0.2086
LocCNN_v3.0	0.0036	0.0555	0.0229	0.549	0.3235	0.2081
LocCNN_v3.0_noLoc	0.0036	0.0555	0.0214	0.6931	0.3144	0.2009
LocCNN_v4.0	0.0038	0.0566	0.021	0.707	0.3066	0.1938
LocCNN_v4.1_second_layer_loc	0.0038	0.057	0.0219	0.6662	0.3401	0.214
LocCNN_v4.2_two_loc_layers	0.0036	0.0554	0.0195	0.7162	0.2742	0.1718

Modelių prognozių tikslumo įverčiai gauti naudojant 2022 m. balandžio dienos duomenų rinkinį

Modelis	MSE	RMSE	MAE	SSIM	F1	FSS
Persistent	0.0211	0.1415	0.0667	0.5878	0.7024	0.5763
CNN_2.1	0.0174	0.1299	0.0831	0.1990	0.7051	0.5765
CNN_2.2	0.0143	0.1183	0.0579	0.4854	0.7711	0.6524
RNN_v1.0	0.0047	0.0648	0.0406	0.1074	0.3050	0.1921
UNet_v1.0	0.0075	0.0851	0.0477	0.4222	0.8730	0.7903
UNet_v1.1	0.0028	0.0488	0.0192	0.4363	0.4662	0.3234
UNet_v1.2_loc	0.0031	0.0511	0.0185	0.6568	0.4224	0.2875
ConvLSTM_v2.1	0.0070	0.0820	0.0423	0.5372	0.8636	0.7764
ConvLSTM_v2.2_loc	0.0069	0.0836	0.0426	0.5627	0.8689	0.7846
LocCNN_v1.0	0.0102	0.0996	0.0530	0.5402	0.8097	0.7068
LocCNN_v2.0	0.0102	0.0993	0.0531	0.5535	0.8127	0.7094
LocCNN_v3.0	0.0103	0.0999	0.0542	0.5061	0.8126	0.7095
LocCNN_v3.0_noLoc	0.0103	0.1003	0.0533	0.5486	0.8084	0.7057
LocCNN_v4.0	0.0103	0.1001	0.0506	0.6183	0.8110	0.7084
LocCNN_v4.1_second_layer_loc	0.0104	0.1005	0.0512	0.5900	0.8041	0.6997
LocCNN_v4.2_two_loc_layers	0.0111	0.1042	0.0547	0.6154	0.8055	0.7020

Modelių prognozių tikslumo įverčiai gauti naudojant 2023 m. rugpjūčio dienos duomenų rinkinį

Modelis	MSE	RMSE	MAE	SSIM	F1	FSS
Persistent	0.0012	0.0296	0.0051	0.8878	0.1758	0.1054
CNN_2.1	0.0006	0.0213	0.0077	0.5349	0.2163	0.1413
CNN_2.2	0.0006	0.0203	0.0070	0.4604	0.3232	0.2192
RNN_v1.0	0.0011	0.0282	0.0182	0.2027	0.2268	0.1448
UNet_v1.0	0.0005	0.0192	0.0093	0.2697	0.3489	0.2482
UNet_v1.1	0.0005	0.0187	0.0076	0.3643	0.3381	0.2400
UNet_v1.2_loc	0.0004	0.0173	0.0054	0.6154	0.3752	0.2711
ConvLSTM_v2.1	0.0004	0.0180	0.0071	0.4440	0.3561	0.2540
ConvLSTM_v2.2_loc	0.0004	0.0174	0.0062	0.4723	0.3620	0.2622
LocCNN_v1.0	0.0006	0.0205	0.0049	0.8750	0.2616	0.1739
LocCNN_v2.0	0.0006	0.0202	0.0046	0.8839	0.2682	0.1794
LocCNN_v3.0	0.0006	0.0210	0.0066	0.6413	0.2727	0.1826
LocCNN_v3.0_noLoc	0.0006	0.0201	0.0045	0.8971	0.2558	0.1698
LocCNN_v4.0	0.0006	0.0202	0.0044	0.8798	0.2871	0.1911
LocCNN_v4.1_second_layer_loc	0.0006	0.0208	0.0051	0.7769	0.2839	0.1886
LocCNN_v4.2_two_loc_layers	0.0006	0.0198	0.0039	0.9061	0.2457	0.1613

Modelių prognozių tikslumo įverčiai gauti naudojant 2022 m. Kalėdų duomenų rinkinį

Modelis	MSE	RMSE	MAE	SSIM	F1	FSS
Persistent	0.0014	0.0323	0.0048	0.9146	0.1280	0.0775
CNN_2.1	0.0008	0.0248	0.0086	0.5571	0.0659	0.0422
CNN_2.2	0.0007	0.0231	0.0072	0.5474	0.2967	0.1964
RNN_v1.0	0.0021	0.0398	0.0282	0.1413	0.1059	0.0620
UNet_v1.0	0.0005	0.0197	0.0091	0.3185	0.3685	0.2666
UNet_v1.1	0.0004	0.0183	0.0072	0.4412	0.4011	0.2921
UNet_v1.2_loc	0.0004	0.0184	0.0052	0.7254	0.3651	0.2651
ConvLSTM_v2.1	0.0004	0.0181	0.0067	0.5334	0.4221	0.3057
ConvLSTM_v2.2_loc	0.0004	0.0176	0.0060	0.5694	0.4194	0.3094
LocCNN_v1.0	0.0007	0.0217	0.0046	0.8886	0.2237	0.1463
LocCNN_v2.0	0.0006	0.0212	0.0041	0.9047	0.2371	0.1587
LocCNN_v3.0	0.0006	0.0218	0.0062	0.6684	0.2401	0.1622
LocCNN_v3.0_noLoc	0.0006	0.0212	0.0041	0.9062	0.2281	0.1510
LocCNN_v4.0	0.0006	0.0216	0.0039	0.9076	0.2319	0.1511
LocCNN_v4.1_second_layer_loc	0.0007	0.0220	0.0047	0.8216	0.2355	0.1529
LocCNN_v4.2_two_loc_layers	0.0006	0.0214	0.0036	0.9223	0.1976	0.1287

Modelių prognozių tikslumo įverčiai gauti naudojant 2022 m. birželio dienos duomenų rinkinį

Modelis	MSE	RMSE	MAE	SSIM	F1	FSS
Persistent	0.0064	0.0715	0.0212	0.8002	0.3275	0.2212
CNN_2.1	0.0039	0.0563	0.0284	0.3761	0.3472	0.2361
CNN_2.2	0.0031	0.0507	0.0182	0.5393	0.4223	0.3113
RNN_v1.0	0.0051	0.0669	0.0427	0.1039	0.3080	0.2010
UNet_v1.0	0.0037	0.0549	0.0279	0.3846	0.3677	0.2554
UNet_v1.1	0.0021	0.0417	0.0160	0.5231	0.4847	0.3804
UNet_v1.2_loc	0.0022	0.0428	0.0153	0.6949	0.4818	0.3744
ConvLSTM_v2.1	0.0021	0.0417	0.0166	0.5430	0.4862	0.3807
ConvLSTM_v2.2_loc	0.0020	0.0409	0.0150	0.5701	0.4988	0.3981
LocCNN_v1.0	0.0031	0.0507	0.0193	0.7337	0.3997	0.2916
LocCNN_v2.0	0.0030	0.0494	0.0176	0.7674	0.4069	0.2986
LocCNN_v3.0	0.0030	0.0496	0.0183	0.7251	0.4202	0.3048
LocCNN_v3.0_noLoc	0.0030	0.0500	0.0190	0.7032	0.4277	0.3089
LocCNN_v4.0	0.0030	0.0493	0.0169	0.7768	0.4313	0.3127
LocCNN_v4.1_second_layer_loc	0.0030	0.0496	0.0177	0.7109	0.4259	0.3085
LocCNN_v4.2_two_loc_layers	0.0029	0.0488	0.0162	0.7865	0.4165	0.3047

Modelių prognozių tikslumo įverčiai gauti naudojant 2023 m. gruodžio dienos duomenų rinkinį