



VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
KOMPIUTERINIO IR DUOMENŲ MODELIAVIMO KATEDRA

Kompiuterinio modeliavimo baigiamasis mokslinis darbas

**Saulės elektrinių dienos energijos prognozavimas pagal orų
prognozių duomenis skirtingose vietovėse**

Daily energy forecasting of solar power plants based on weather forecast data in different locations

Atliko:

Matas Amšiejus

parašas

Vadovas:

dr. Aistis Raudys

Vilnius
2026

Turinys

Santrauka	4
Summary	4
1. Įvadas	6
2. Literatūros apžvalga	8
2.1. Prognozės veiksniai ir duomenų svarba	8
2.2. Giliojo mokymosi metodų taikymas	8
2.3. Hiperparametrų optimizacija ir ansambliniai metodai	9
2.4. Sprendimų medžių metodai (XGBoost)	9
2.5. Erdvinis prognozavimas ir modelių generalizacija	10
3. Duomenys	11
3.1. Saulės elektrinių duomenys	11
3.2. Orų duomenys	12
3.3. Duomenų paruošimas	13
3.3.1. Išskirčių šalinimas ir tuščių reikšmių užpildymas	13
3.4. Naujų išvestinių duomenų kūrimas	14
3.4.1. Saulės aktyvumo laikas tiriamą valandą	14
3.4.2. Atstumas iki vidurdienio	14
3.4.3. Atstumas iki vasaros saulėgrįžos	15
3.4.4. Elektrinių polinkio ir azimuto nustatymas	16
3.5. Galutiniai duomenys	18
4. Modeliavimas	20
4.1. LSTM modelis	20
4.1.1. Vidinė ląstelės architektūra	20
4.1.2. Užmaršties vartai	20
4.1.3. Įvesties vartai ir kandidatų atnaujinimas	21
4.1.4. Išvesties vartai	22
4.1.5. LSTM taikymas saulės energetikoje	22
4.2. Sprendimų medžių ansambliai	22
4.2.1. Tikslų funkcija	22
4.2.2. Regularizacija ir modelio sudėtingumas	23
4.2.3. Optimizavimas naudojant antros eilės aproksimaciją	23
4.3. Duomenų paruošimas skirtingiems modeliams	23
4.4. Modelio hiperparametrų optimizavimas	24
4.4.1. Hiperjuostinis parametrų parinkimas	24
4.4.2. Vertinimui paruošti hiperparametrai	24
4.4.3. Atsitiktinis gardelės parametrų parinkimas	25
4.5. Keras	26
4.6. Dviejų etapų optimizavimas	26
4.7. Vertinimo metrikos	27

5. Rezultatai	28
5.1. Geriausias modelis	28
5.1.1. LSTM	28
5.1.2. XGBoost	29
5.2. Erdvinio prognozavimo tikslumo analizė	30
5.2.1. Tyrimo eiga ir metodika	31
5.2.2. Rezultatų analizė	31
5.3. Lokalaus ir bendro modelių palyginimas	32
5.3.1. Rezultatų apžvalga	32
5.3.2. Bendrojo modelio pranašumai	35
6. Išvados bei rekomendacijos	36
Literatūros šaltiniai	37
7. Priedai	39

Santrauka

Saulės elektrinės šiomis dienomis yra aktuali tema tiek visame pasaulyje, tiek Lietuvoje. Šiame darbe sprendžiama saulės elektrinių generacijos prognozavimo problema naujose vietovėse, pasitelkiant mašininio mokymosi metodus ir atvirus meteorologinius duomenis. Tyrimo metu lyginamas giliojo mokymosi (LSTM) ir sprendimų medžių (XGBoost) modelių efektyvumas. Eksperimentinis modelių vertinimas atliktas naudojant 10-ies skirtingų elektrinių istorinius duomenis, testuojant tikslumą su viena, mokymo procese nenaudota elektrine, kartojant ciklą 11 kartų. Rezultatai parodė, kad vidutiniškai XGBoost modelis pasižymėjo 15 % mažesne MAE nei LSTM, buvo pastovesnis bei turėjo mažesnius skaičiavimo kaštus. Nustatyta, kad sukurti globalūs modeliai yra pakankamai tikslūs ($MAE < 0.05$), tačiau nusileidžia lokaliems modeliams.

Summary

With the rapid expansion of renewable energy sources, particularly photovoltaics, their integration into national power grids presents significant challenges due to the intermittent nature of solar generation. Accurate forecasting is therefore critical for ensuring grid stability, efficient load balancing, and economic viability. The primary objective of thesis "Daily energy forecasting of solar power plants based on weather forecast data in different locations" is to address the challenge of spatial generalization—predicting solar power generation in new, previously unmonitored locations where historical generation data is unavailable, and to determine the impact of various meteorological variables on prediction accuracy.

To achieve this, the study investigates and compares two distinct machine learning approaches: the deep learning-based Long Short-Term Memory (LSTM) neural networks, designed to capture temporal dependencies, and the ensemble-based eXtreme Gradient Boosting (XGBoost) decision trees. The research utilizes a dataset comprising historical generation data from 11 distinct solar power plants alongside open-access meteorological reanalysis data from "Open-Meteo". To enhance model performance, extensive feature engineering was performed, including the calculation of panel variables (azimuth and tilt angles). Furthermore, the Hyperband optimization algorithm was used to tune hyperparameters and select the optimal model architectures.

The experimental phase involved training models on a dataset from 10 power plants and testing their performance on an 11th, "unseen" plant to simulate a real-world scenario of expanding to a new site. The results demonstrated that the XGBoost model significantly outperformed the LSTM network, achieving approximately 15 % lower MAE on average while requiring substantially lower computational resources for training.

The study highlights a trade-off between generalization and precision: comparative analysis showed that models trained specifically on local data from a single plant still yield higher accuracy

than generalized models. The thesis concludes with recommendations for further research, suggesting that integrating real-time satellite imagery or ground-based cloud observation data could further mitigate errors caused by cloud movement.

1. Įvadas

Dėl kylančių elektros kainų, klimato kaitos bei atsiribojimo nuo geopolitinės situacijos populiarėjanti atsinaujinanti energetika skatina vis dažnesnį saulės elektrinių įrengimą. Saulės energijos prognozavimas tampa itin aktualus siekiant efektyviau integruoti saulės elektrinių pagaminamą energiją į elektros tinklus, optimizuoti energijos tiekimą bei greitai identifikuoti ir spręsti veiklos sutrikimus. Šie tikslai yra ypač svarbūs energetikos sektoriuje, susiduriančiame su tokiomis problemomis kaip nepastovios meteorologinės sąlygos, kurios apsunkina energijos gamybos planavimą. Tai mažina viso tinklo efektyvumą ir potencialą išgauti maksimalią naudą iš saulės spinduliuotės. Tikslesnė prognozė gali padėti efektyviai paskirstyti elektrą po tinklus: esant jos pertekliui, nukreipti į baterijų sistemas, o stygiui – aktyvuoti alternatyvius, stabilesnius šaltinius, tokius kaip iškastinis arba branduolinis kuras. Norint įgyvendinti Lietuvos iškeltą tikslą – atsinaujinančių energijos išteklių dalį, palyginti su šalies bendruoju galutiniu energijos suvartojimu, padidinti iki 55 % (2030 m.) ir 95 % (2050 m.) [14], efektyvus saulės elektrinių sugeneruojamos energijos nukreipimas yra ypač svarbus. Šis darbas yra orientuotas į efektyvumo gerinimą prognozuojant sugeneruojamos elektros kiekį Lietuvos mastu.

Gausesnį kiekį duomenų turinčios orų prognozės ir didėjantis saulės elektrinių skaičius generuoja vis daugiau informacijos, leidžiančios kurti aukštos kokybės prognozavimo modelius. Tačiau dauguma jų kuriami tik vienai vietai, kurios parametrai kaip platuma ir ilguma, galingumas, kampas į saulę yra fiksuoti ir kinta tik laikas bei orai. Toks lokalizuotas požiūris riboja galimybę plačiau pritaikyti šiuos modelius įvairiose vietovėse, turinčiose skirtingus minėtus parametrus.

Šiame darbe bus atsižvelgiama į skirtingose vietovėse esančias elektrines, kuriant giliuoju mokymusi paremtus LSTM ir mašininį mokymu paremtus XGBoost modelius, mokant juos ne tik laike kintančiais duomenimis, bet ir elektrinių meta duomenimis, kaip elektrinės vietovė, individuali teorinė plokštumos apšvita.

Darbo tikslas: ištirti generalizuotą prognozavimo modelį, gebantį pagal orų prognozę, laiką, saulės bei elektrinių meta duomenis kaip galingumas, vietovė, tiksliai numatyti saulės elektrinių generaciją naujose Lietuvos vietovėse nenaudojant jų istorinių generacijos duomenų, bei įvertinti parametru įtaką modelio tikslumui.

Darbo uždaviniai:

1. Atlikti literatūros apžvalgą ir parinkti tiksliausią modelio architektūrą.
2. Surinkti ir apdoroti duomenis iš 11-os skirtingų Lietuvos saulės elektrinių bei „Open-Meteo“ reanalizės archyvų, pritaikant išskirčių šalinimą ir duomenų užpildymo algoritmus.
3. Sukurti metodiką nežinomų elektrinių geometrinių parametru (azimuto, pasvirimo kampo) nustatymui ir sugeneruoti inžinerinį kintamąjį – teorinę apšvitą į plokštumą (POA_{opt}).
4. Optimizuoti ir apmokyti du skirtingos architektūros modelius: ilgalaikės trumpalaikės atminties (LSTM) tinklą ir sprendimų medžių ansamblius (XGBoost).
5. Atlikti komponentų įtakos analizę, įvertinant POA_{opt} kintamojo naudą prognozavimo tikslumui.
6. Įvertinti modelių generalizacijos gebėjimus mokant modelį su 10 vietovių, testuojant su viena nauja.

7. Atlikti atvejo analizę, lyginant sukurto bendro modelio ir lokalaus modelio efektyvumą realiomis sąlygomis.

Mokslinis naujumas ir praktinė reikšmė: Darbe siūlomas bendras modelis leidžia prognozuoti generaciją bet kurioje Lietuvos vietoje be istorinių duomenų, o tai atveria galimybes kurti investicinio vertinimo įrankius bei efektyviau valdyti jėgaines.

Informacija iš mokslo tiriamojo darbo projekto (MTDP):

1. Naudojami tie patys saulės elektrinių duomenys. Imtis praplėsta nuo 7 iki 11 elektrinių.
2. Dalis literatūros apžvalgos sutampa, nes tema yra vienoda.
3. Dalis naudojamo kodo sutampa, pavyzdžiui elektrinių duomenų parsisiuntimas.
4. Naujų išvestinių duomenų kūrimo, Hyperband teorijos aprašymas yra panašūs.
5. Prieduose yra kelios iliustracijos naudotos MTDP.

Šiame darbe taip pat kartais bus paminimas „senesnis / ankstesnis darbas“, „smulkesnis tyrimas“, „pradinis tyrimas“. Tai atitinka mokslo tiriamojo darbo projektą.

Darbo struktūra: Literatūros apžvalgoje bus apžvelgiami panašūs darbai, susiję su generuojamos elektros prognozavimo uždaviniais. Duomenų skiltyje bus aprašomi duomenų šaltiniai, apdorojimas, analizė ir išvestinių duomenų kūrimas. Modeliavimo skyriuje matematiškai aprašyti naudojami modeliai, jų hiperparametrų optimizacija, vertinimo metrikos. Rezultatuose bus apžvelgiami rasti geriausi modeliai, kintamųjų svarba, atliekami palyginimai tarp skirtingų architektūrų su skirtingomis mokymo ir testavimo imtimis, lyginama vienos vietovės situacija.

2. Literatūros apžvalga

Šiomis dienomis pasaulyje yra atlikta daugybė tyrimų, susijusių su saulės elektrinių generuojamos energijos prognozavimu. Tačiau dauguma tyrimų apsiriboja viena vietoje, ir, dažniausiai, vienodų saulės kolektorių informacija. Kadangi šis darbas atsižvelgia į skirtingą vietovę bei elektrines, dauguma apžvelgtų darbų taip pat panaudoja šią informaciją. Visi straipsniai siejasi su pagaminamos energijos prognozavimu, kur naudojami tiek klasikiniai matematiniai modeliai kaip tiesinė regresija, tiek inovatyvūs neuroninių tinklų modeliai.

2.1. Prognozės veiksniai ir duomenų svarba

R. Ahmed ir kitų apžvalgoje [1] analizuojami skirtingi saulės elektrinių generacijos prognozavimo tipai, duomenų rinkiniai, optimizavimo metodai. Įvardinama, kad vienas esminių prognozavimo faktorių yra spinduliuotės kiekis, kuris smarkiai priklauso nuo debesuotumo judrumo kintamojo, kurį reikėtų įtraukti norint sudaryti tikslius modelius. Kaip vieną iš būdų tikslinti modelį autoriai įvardina nuo žemės daromas dangaus nuotraukas ir jų analizę. Šių vaizdų analizė padeda išvystyti tikslius minutinius modelius. Straipsnyje įvardinama, kad didžiausia koreliacija su pagaminamos elektros kiekiu yra saulės spektrinė spinduliuotė (tiesinės regresijos modelyje $R^2 \approx 0.99$), kuri priklauso nuo tokių faktorių kaip aerozolio sklaidos, vėjo greičio ir krypties, drėgmės ir debesuotumo. Priešingai, oro temperatūra neturėjo pastovios koreliacijos. Kaip svarbų žingsnį autoriai įvardino orų klasifikavimą. Vėjo greitis ir panelės temperatūra taip pat turėjo įtakos prognozėms, nes šie faktoriai daro įtaką kolektoriaus naudingumui. Pabrėžiama, kad praeities įvestys taip pat lemia modelių tikslumą. Prognozuojant dieną į priekį, paprastai užtekdamo 5 praėjusių dienų duomenų. Tačiau su kasvalandiniais duomenimis, daugiau neužtikrintumo atsirasdavo viduryje dienos, kai pastarosios įvestys būdavo nepastovesnės.

Dažnai panašiuose tyrimuose orams aprašyti yra naudojami orų reanalizės duomenys. Tačiau literatūroje minima, kad yra susiduriama su neatitikimais tarp fizinių matavimo stotelių ir globalių reanalizės modelių. A. Boilley ir L. Wald [6] savo tyrime lygino NASA (MERRA) ir Europos (ERA-Interim) reanalizės duomenis su tiksliais antžeminiiais matavimais. Autoriai nustatė, kad reanalizės modeliai turi tendenciją pervertinti saulės spinduliuotę, ypač debesuotomis dienomis, dažnai klaidingai prognozuodami giedrą dangų, kai realybėje yra debesuota. Nustatyta, kad reanalizės duomenys geriausiai tinka vertinti bendras tendencijas giedro dangaus sąlygomis, tačiau pasižymi didesne paklaida drėgno klimato zonose. Nepaisant šių ribotumų, reanalizės duomenys išlieka nepakeičiami sprendžiant geografiškai nutolusių ar nepakankamai matavimo įranga padengtų vietovių (kaip šio darbo atveju) prognozavimo problemas, nes užtikrina duomenų tolydumą ir prieinamumą bet kuriam taškui.

2.2. Giliojo mokymosi metodų taikymas

Giliojo mokymosi algoritmai tapo standartu laiko eilučių prognozavime. A. Ozbek, A. Yildirim ir M. Bilgili [17] darbe teigiama, kad saulės elektrinių pagaminamas elektros kiekis valanda į priekį gali būti pakankamai tiksliai prognozuojamas ir neturint tikslų meteorologinių duomenų. Tyrime apmokomi ANFIS modeliai (angl. *Adaptive neuro-fuzzy inference system*), naudojantys klasterizavimą bei sąlyginę logiką, kurie lyginami su LSTM modeliu. Pastarasis yra pranašesnis už visas ANFIS variacijas, su valandos į priekį pagaminamos elektros prognozavimo $R^2 = 0,977$.

Tuo tarpu Lietuvos kontekste atliktame A. Raudžio bei J. Gaidukevičiaus tyrime [18] prognozuojamas saulės elektrinių pagaminamos bei vienu namų suvartojamos elektros kiekiai kas valandą. Buvo pasirinktos dvi skirtingų maksimalių galingumų (5.5 kW ir 30 kW) saulės jėgainės. Darbe buvo išgauta informacija kaip saulėtekio, saulėlydžio bei vidurdienio metas. Šie laikai buvo panaudoti nustatant, ar saulė buvo matoma pilną valandą, ar tik jos dalį bei laiką iki/nuo vidurdienio. Darbe teigiama, kad nėra prasmės atsižvelgti į datą, todėl geriau vertinti laiką iki vasaros ir žiemos saulėgrįžų. Tyrime naudojamas ir saulės azimutas bei zenitas. Tiksliausia modelių architektūra buvo GRU, kuri tikslumu minimaliai aplenkė LSTM.

2.3. Hiperparametrų optimizacija ir ansambliniai metodai

Haoran Zhang ir kitų mokslininkų „Taikomosios energijos“ žurnalo straipsnyje [24] prognozuoja saulės elektrinių pagaminamos elektros kiekį skirtingose vietovėse naudojantis ilgalaikės trumpalaikės atminties LSTM tinklais bei dalelių spiečiaus optimizacijos algoritmu (angl. *Particle swarm optimization* ar *PSO*). Prognozuoti naudojamas elektrinių azimutas, kampas nuo žemės, orų sąlygos, elektrinių galingumas ir istorinis elektros generavimas. Vienas iš nepriklausomų kintamųjų yra saulės spinduliuotė, kuri yra puikus matas pagaminamos elektros prognozei. Tyrime buvo panaudoti 5 regionai esantys 12 km spindulyje, kur kiekvienam jų buvo priskiriamos kelios elektrinės. Darbe teigiama, kad nors ir yra tyrimų, naudojančių LSTM, tik maža dalis jų naudoja algoritmus, skirtus optimizuoti tinklų parametrus saulės elektrinių generavimo prognozavimui, nors pastarieji pagerina panašių sferų ar modelių tikslumus [22][23][25]. PSO optimizuoja parametrus kaip mokymosi greitis, laiko žingsnis, imties dydis ir neuroninių vienetų skaičius. Parinkta tinklo architektūra prognozavimo tikslumu pralenkia neoptimizuotus LSTM, neuroninius tinklus bei kitus modelius.

S. Al-Dahidi straipsnyje [2] yra tiriamas ansamblių ir dirbtinių neuroninių tinklų (angl. *Artificial Neural Networks* ar *ANN*) taikymas. Tyrime siekiama ne tik pagerinti prognozavimo tikslumą 24 valandoms į priekį, bet ir įvertinti reikšmės neapibrėžtumą. Darbe autoriai pasirinko naudoti „BAGGING“ agregavimo techniką, kurioje individualiems baziniams modeliams yra priskiriama unikali duomenų imtis. Prognozei naudojamas saulės spinduliuotės kiekis, oro temperatūra. Pirmą sukuriama didelis bazinių modelių skaičius, iš kurių gaunamos skirtingos prognozės. Pastarosios yra statistiškai agreguojamos ir išvedama galutinė reikšmė. Galiausiai išvedamas reikšmių neužtikrintumas naudojant „Bootstrap“ metodą. Naudojant validacijos imtis ir keičiant parametrus gardelėje buvo parinktas optimalus paslėptų neuronų skaičius 30 ir modelių skaičius 20. Gauta galutinio modelio MAE = 6.15 kWh. Gauti pasiklovimo intervalai (PI) naudojant BS metodą padengia 80 % prognozių.

2.4. Sprendimų medžių metodai (XGBoost)

Literatūroje pastebima, kad XGBoost modeliai nenusileidžia, o kartais ir lenkia LSTM modelius. Pavyzdžiui, 2023 m. „Energies“ žurnale publikuotame tyrime [8] buvo lyginami penki skirtingi modeliai (įskaitant XGBoost, LSTM ir MLP) prognozuojant fotovoltinę generaciją. Autoriai nustatė, kad XGBoost algoritmas ne tik pasižymėjo mažesne skaičiavimų kaina, bet ir tam tikrais atvejais (pvz., prognozuojant 1 valandą į priekį, kas bus daroma šiame darbe) demonstravo geresnį tikslumą nei LSTM. Tai paaiškinama tuo, kad XGBoost efektyviau tvarkosi su triukšmin-

gais duomenimis ir nereikalauja tokio didelio duomenų kiekio kaip giliojo mokymosi metodai, kad pasiektų konvergavimą. Šie rezultatai moksliskai pagrindžia XGBoost pasirinkimą kaip lygiavertį konkurentą LSTM architektūrai šiame magistriniame darbe.

XGBoost algoritmo efektyvumas laiko eilučių prognozavime nagrinėjamas ir R. Gupta bei A. K. Yadav darbe [10]. Autoriai lygino XGBoost modelio tikslumą su populiariu laiko eilučių prognozavimo įrankiu „Facebook Prophet“, prognozuojant saulės elektrinių generaciją. Tyrimo rezultatai parodė, kad XGBoost modelis pasižymėjo tikslesnėmis prognozėmis nei „Prophet“. Tai patvirtina XGBoost, kaip universalios ir galingo įrankio tinkamumą spręsti netiesinius saulės energijos generacijos uždavinius.

2.5. Erdvinis prognozavimas ir modelių generalizacija

J. M. Barrera ir kitų straipsnyje [3] kuriamas dirbtinių neuroninių tinklų modelis skirtas prognozuoti elektrinių generuojamą galią skirtingose Europos šalyse. Autoriai iš atvirų duomenų šaltinių pasirinko 4 šalis pretendentes, o iš jų - po kelias elektrines. Vietovei nusakyti jie naudojo UTM koordinates. Į duomenis buvo įtraukti ir skirtingi saulės kolektorių gamintojai bei modeliai ir jų konversijos koeficientai. Šiame darbe naudojami tie patys komponentai, todėl į šiuos parametrus nebus atsižvelgiama. Tyrime taip pat buvo panaudota fotovoltinės geografinės informacinės sistemos (angl. *Photovoltaic Geographical Information System* (PVGIS)), iš kurių buvo panaudota saulės apšvietos kiekio tam tikroje vietovėje informacija. Gautas modelis prognozavo pagaminamos elektros kiekį mokymo duomenyse neįtrauktose vietovėse su 16 % paklaida.

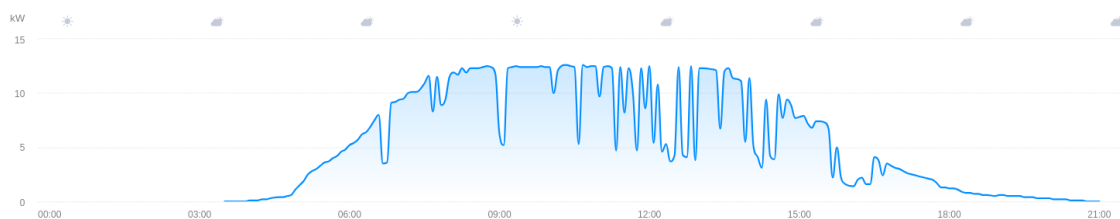
3. Duomenys

3.1. Saulės elektrinių duomenys

Duomenys buvo renkami iš vienos Lietuvos įmonės, įrengiančios saulės elektrines. Duomenyse apie pačias elektrines buvo saugoma jų maksimali elektros generavimo galia (kWp) (angl. *kilowatts peak (kWp)*), koordinatės, dienos nuo veikimo pradžios.

Nors atliekant tyrimą, pirma buvo tikimasi, kad saugoma tokia informacija kaip elektrinės azimutas ir kolektorių polinkio kampas, paaiškėjo, kad įmonė, teikianti duomenis, naudojo numatytas reikšmes (azimutas visada 180° , kampas - 0°); todėl ši informacija liko nepanaudota. Dėl to skyriuje 3.4.4 bus naudojama gardelės paieška vertinant šiuos dydžius.

Generuojamos elektros duomenys buvo renkami kas ≈ 15 min., tačiau buvo matuojamas tik momentinis generavimas (kW). Tai yra, jei matavimo metu plaukė debesis, gautas dydis yra kur kas mažesnis nei būtų praplaukus debesims. Dėl to dauguma grafikų yra nepastovūs ir turi daug svyravimų (1 pav.). Kiekvienai elektrinei surinktas duomenų kiekis skiriasi, tačiau visos jos turi bent vienus pilnus metus duomenų.

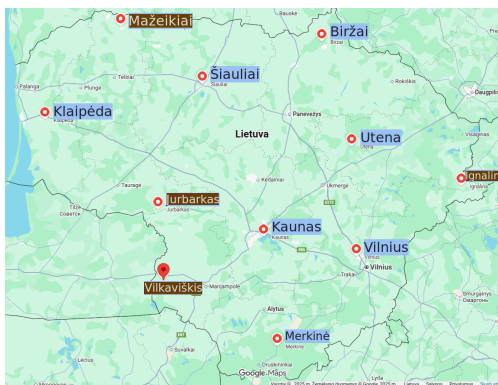


1 pav. Vienos dienos elektrinės galios grafikas (prieš kasvalandinių duomenų išvedimą)

Tam, kad suglodontume signalą ir gautume stabilesnes vertes bei prisitaikytume prie kasvalandinių orų duomenų, buvo pasirinkta išvesti kasvalandinę vidutinę sugeneruojamos elektros reikšmę, skaičiuojant vidurkį iš atitinkamos valandos.

Didžiosios dalies elektrinių stebėjimai buvo sekami UTC+2 laiko zonoje (Lietuvos žiemos laikas). Tačiau dalis elektrinių buvo sukonfigūruotos UTC+1 laiko zonai. Tokiu atveju prie jų laiko buvo pridėdama po 1 valandą. Toliau tyrime bus naudojama UTC+2 zona.

Pradinio tyrimo metu buvo pasirinktos 7 elektrinės. Darbą plečiant, pridėtos dar 4 elektrinės, bandant geriau padengti Lietuvos plotą. Elektrinės yra skirtinguose šalies regionuose, jas galima pamatyti 2 paveiksle.



2 pav. Tirtų saulės elektrinių išsidėstymas Lietuvoje

Mėlynai nuspaltintos pradinio tyrimo metu naudojamos elektrinės, rudai - pridėtos naujos. Elekt-

rinės vietai nurodytos koordinatės buvo konvertuotos į UTM koordinatės sistemą dėl paprasto interpretavimo (matuojama metrais).

3.2. Orų duomenys

Pirminiame tyrimo etape orų duomenys buvo gauti naudojant *meteo.lt* programų programavimo sąsają (angl. *API - application programming interface*). Nuo elektrinių rastos artimiausios meteorologinės stotys, kurios saugo istorinius duomenis, jų atstumai nuo tyrimo vietų gali būti pamatyti 1 lentelėje.

1 lentelė. Atstumai tarp elektrinių ir stočių

Elektrinė	Meteorologinė stotis	Atstumas (km)
Anykščių	Utenos st.	31
Biržų	Biržų st.	4.5
Kauno	Kauno st.	13
Klaipėdos	Klaipėdos st.	20
Merkinės	Varėnos st.	25
Šiaulių	Šiaulių st.	2
Vilniaus	Vilniaus st.	16

Nors šis metodas leido gauti oficialius matavimus, tyrimo eigoje išryškėjo esminiai trūkumai, ribojantys modelio tikslumą:

1. Atstumas tarp elektrinės ir meteorologinės stoties kai kuriais atvejais siekė virš 30 km (pvz., Anykščių elektrinė – Utenos stotis). Debesuotumas yra lokalus reiškinys – debesis, esantis virš stoties, nebūtinai yra virš elektrinės, ir atvirkščiai. Tai sukūrė nepastovumą duomenyse tarp debesuotumo ir generuojamos elektros.
2. Fizinės stotys susiduria su techniniais gedimais, dėl kurių atsiranda praleistos reikšmės. Didinant saulės elektrinių skaičių pastebėta, kad mažiau gyvenamose vietovėse meteorologinės stotys nefiksuodavo kai kurių raktinių parametrų kaip debesuotumas ar krituliai.
3. *Meteo.lt* nesaugo saulės apšvietos dydžių kaip GHI, kurie tiksliausiai nusako generuojamos elektros kiekį.

Siekiant eliminuoti šias problemas ir padidinti prognozavimo tikslumą, šiame darbe buvo pereita prie ERA5 ir ECMWF IFS reanalizės duomenų, naudojant *Open-Meteo*¹ istorinį archyvą. Tai palydoviniais stebėjimais ir globaliais atmosferos modeliais paremti duomenys, kurie leidžia gauti informaciją tikslioms elektrinės koordinatėms, eliminuojant atstumo paklaidą. Naujasis duomenų šaltinis leido įtraukti specifinius saulės apšvietos parametrus, kurių nebuvo pradiname etape:

1. GHI (angl. *Global Horizontal Irradiance*): Bendroji horizontali spinduliuotė (W/m^2).
2. DNI (angl. *Direct Normal Irradiance*): Tiesioginė normalioji spinduliuotė – ypač svarbi giedromis dienomis.

¹<https://open-meteo.com/en/docs/historical-weather-api>

3. DHI (angl. *Diffuse Horizontal Irradiance*): Difuzinė spinduliuotė – svarbi debesuotomis dienomis, kai tiesioginės saulės nėra.

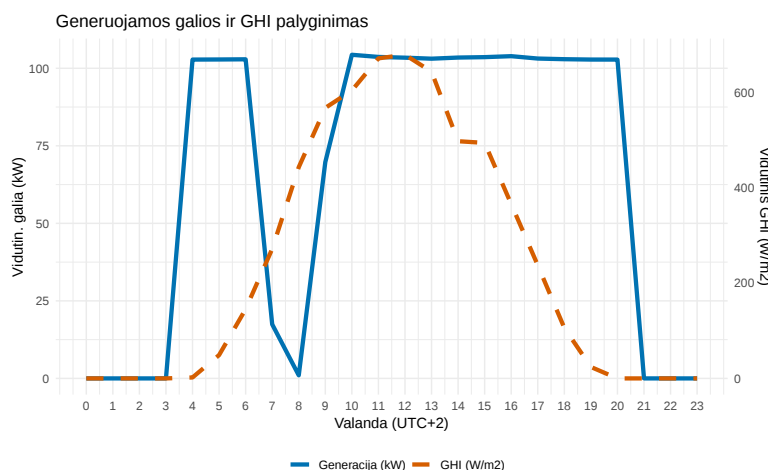
Šis perėjimas nuo stotelių duomenų prie lokalių reanalizės duomenų leido sukurti vientisą ir tikslesnį duomenų rinkinį, tiksliau koreliuojantį su elektrinių generacija.

Be minėtų parametrų, tyrime naudojama oro temperatūra 2 m. virš žemės ($^{\circ}\text{C}$), vėjo greitis (m/s), vėjo gūsis (m/s, maksimalus gūsis per valandą), vėjo kryptis ($^{\circ}$), debesuotumas (%; 0 - giedra, 100 - debesuota), slėgis jūros lygyje (hPa), santykinė oro drėgmė (%), lietaus kiekis (mm, suma per valandą), sniegas (cm). Orų laikui buvo naudojama UTC+2 laiko zona. Duomenys renkami kas valandą.

3.3. Duomenų paruošimas

3.3.1. Išskirčių šalinimas ir tuščių reikšmių užpildymas

Analizuojant surinktus istorinius duomenis, buvo identifikuotos dvi pagrindinės problemos: išskirtinai didelės ir trūkstamos reikšmės. Pastebėta, kad Ignalinos ir Vilkaviškio duomenyse 2024 08 mėn. 8 - 9 dienomis generacijos duomenys buvo drastiškai didesni nei įprastai, viršydami savo maksimalią elektros generavimo galią keliasdešimt kartų (3 pav.).



3 pav. Ignalinos elektrinės išsprogusi generacija

Kadangi LSTM modeliams, kuriuos toliau mokysime, svarbus laiko eilučių tolydumas, paprastas šio laikotarpio pašalinimas sukurtų laiko šuolį. Siekiant to išvengti, darbe pritaikytas tiesinės regresijos praleistų reikšmių užpildymas. Modelis buvo apmokytas naudojant tik kokybiškus tos pačios elektrinės vasaros duomenis. Generuojama galia (P) modeliuojama kaip funkcija nuo saulės spinduliuotės W/m^2 (GHI), lietaus mm (L) ir paros laiko (h): $P = f(GHI, K, h)$. Paros laikas įtrauktas kaip kategorinis kintamasis, siekiant, kad modelis geriau prisitaikytų prie specifinių tos elektrinės ypatumų kaip kampas ir šešėlis, kurie nepriklauso nuo bendros spinduliuotės. Duomenų tvarkymui buvo naudojama R programavimo kalba.

Naudojantis sukurtu modeliu, išsiskiriančio laikotarpio reikšmės buvo pakeistos modelio sugeneruotomis prognozėmis. Ištaisius išskirtis, Ignalinos vidutinės generuojamos elektros ir GHI Piersono koreliacija pakilo nuo 0,164 iki 0,87, nors buvo taisomos tik 2 dienos.

Analogiška logika pritaikyta ir trūkstamoms reikšmėms užpildyti kitose elektrinėse. Pildymui buvo taikoma taisyklė: jei toje vietovėje fiksuojama saulės spinduliuotė ($GHI > 10 \text{ W}/\text{m}^2$), bet

generacijos duomenų nėra, jie atkuriami naudojant minėtą regresijos modelį (apmokyta kiekvienai elektrinei atskirai). Kadangi žiemą irgi buvo praleistų reikšmių, šiam sezonui buvo sukuriamas atskiras modelis ir pridėdama sniego (cm) kovariantė. Nakties metu trūkstamos reikšmės buvo užpildytos nuliais.

3.4. Naujų išvestinių duomenų kūrimas

Remiantis kitų tyrėjų darbu [18], šiame tyrime pasirinkta išvesti kelis naujus kintamuosius, kurie padės įvertinti dienos bei metų sezoniškumus. Nauji dydžiai bus išvesti pagrįdę naudojant saulėtekio, vidurdienio ir saulėlydžio laikus. Pastarieji buvo suskaičiuoti naudojant *Python* programavimo kalbos biblioteką *Astral*, suvedus vietovių, kuriose yra saulės kolektoriai, koordinatas.

3.4.1. Saulės aktyvumo laikas tiriamą valandą

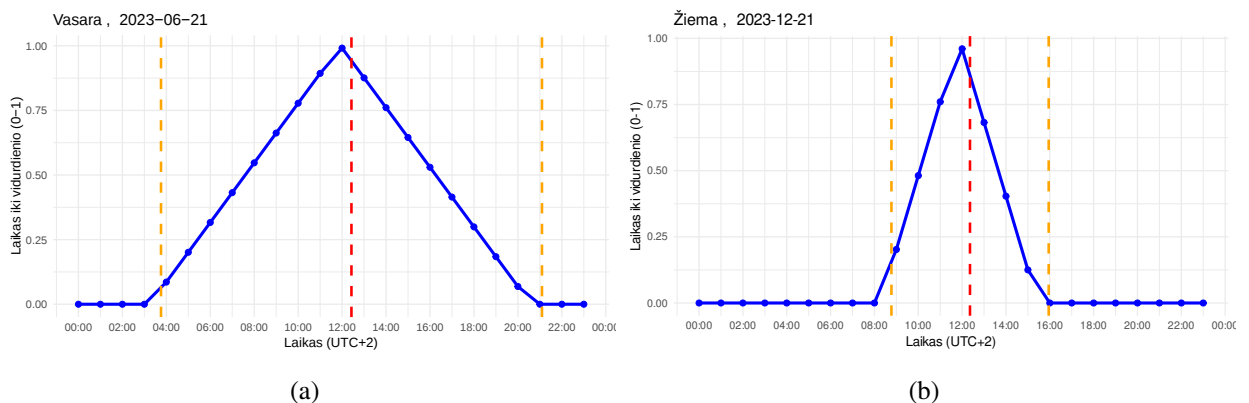
Kadangi turimi duomenys yra kasvalandiniai, per dieną pakliūva 2 valandos (ryte ir vakare) kai saulė (ne)šviečia pilną valandą. Kad įvertinti šį faktą, buvo išvestas saulės aktyvumo laikas tiriamą valandą naudojant saulėtekio ir saulėlydžio duomenis. Dydis parodo, kiek laiko valandos tarpe saulė buvo aktyvi (0-1). Saulės aktyvumo laikas tiriamą valandą paskaičiuotas pagal formulę:

$$f(d, s, p) = \begin{cases} 1, & \text{kai } s_h < d_h < p_h \\ \frac{60 - (\frac{s_s}{60} + s_m)}{60}, & \text{kai } s_h = d_h \\ \frac{\frac{p_s}{60} + p_m}{60}, & \text{kai } p_h = d_h \\ 0, & \text{kitaip atvejais} \end{cases} \quad (3.1)$$

kur d - tiriamo, s - saulėtekio, p - saulėlydžio laikas. s, m, h - atitinkamai sekundės, minutės ir valandos.

3.4.2. Atstumas iki vidurdienio

Reikšmė išvesta naudojant vidurdienio laiką. Šis dydis padeda įvertinti laiką iki teoriškai didžiausio elektrinės naudingumo atitinkamą dieną (4 pav.).



4 pav. Laikas iki vidurdienio Kauno elektrinėje vasarą (a) ir žiemą (b)

Atstumas iki vidurdienio buvo skaičiuojamas pagal formulę:

$$g(d, s, p, v) = \begin{cases} \frac{t(d)+1800-t(s)}{t(v)-t(s)}, & \text{kai } t(s) \leq t(d) \leq t(v) \\ 1 - \frac{t(v)-t(d)-1800}{t(v)-t(p)}, & \text{kai } t(v) \leq t(d) < t(p), \\ 0, & \text{kitais atvejais} \end{cases} \quad (3.2)$$

kur

$$t(x) = 60^2 x_h + 60 x_m + x_s. \quad (3.3)$$

Čia $v = (v_h, v_m, v_s)$ - vidurdienio laikas. Prie dabartinio laiko $t(d)$ buvo pridėta 1800 sekundžių (30 min.) tam, kad būtų vertinamas valandos vidurys, o ne tik pradžia ar pabaiga (pvz. iš 12:00 → 12:30).

3.4.3. Atstumas iki vasaros saulėgrįžos

Saulės elektrinių generacija pasižymi stipriu sezoniškumu, kurį lemia žemės pozicija aplink saulę ir sudaromas kampas tarp jos spindulių. Nors vienas būdas būtų naudoti metų dienos numerį (nuo 1 iki 365), šis metodas turi esminį trūkumą modeliuojant sezoninius procesus: metų pabaiga (gruodžio 31 d., diena = 365) ir metų pradžia (sausio 1 d., diena = 1) laiko atžvilgiu yra greta, tačiau skaitine prasme – toli. Toks šuolis apsunkina modelių mokymosi procesą.

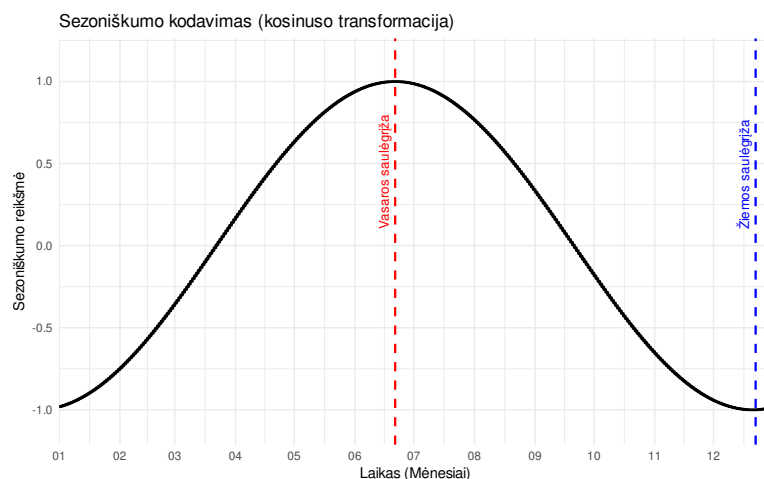
Siekiant išspręsti šią problemą ir užtikrinti tolydų sezoniškumo atvaizdavimą, šiame darbe pritaikyta kosinuso transformacija. Metinis ciklas buvo sudaromas pagal astronominę vasaros saulėgrįžą, o ne pagal kalendorinę dieną. Todėl išvestas dydis padės matematiškai įvertinti teorinį saulės potencialą per visus metus. Kiekvienam laiko momentui t sezoniškumo reikšmė $S_{cos}(t)$ apskaičiuojama pagal šią formulę:

$$S_{cos}(t) = \cos \left(\frac{2\pi \cdot (d(t) - d_{lygiad})}{D_{metai}(t)} \right), \quad (3.4)$$

kur:

- $d(t)$ – einamoji metų diena, kintanti nuo 1 iki 366;
- d_{lygiad} – vasaros saulėgrįžos diena tais metais (birželio 20 arba 21 d., priklausomai nuo keliamųjų metų). Tai yra atskaitos taškas, kurio reikšmė siekia maksimumą;
- $D_{metai}(t)$ – bendras dienų skaičius tuose metuose (365 įprastais, 366 keliamaisiais metais).

Ši transformacija sukuria tolydžią bangą intervale $[-1, 1]$, kuri atspindi astronominį metų ciklą be jokių šuolių (5 pav.). Reikšmė ≈ 1 atitinka vasaros saulėgrįžą (birželio mėn.), kai generacijos potencialas yra didžiausias, o ≈ -1 atitinka žiemos saulėgrįžą (gruodžio mėn.), kai potencialas yra mažiausias.

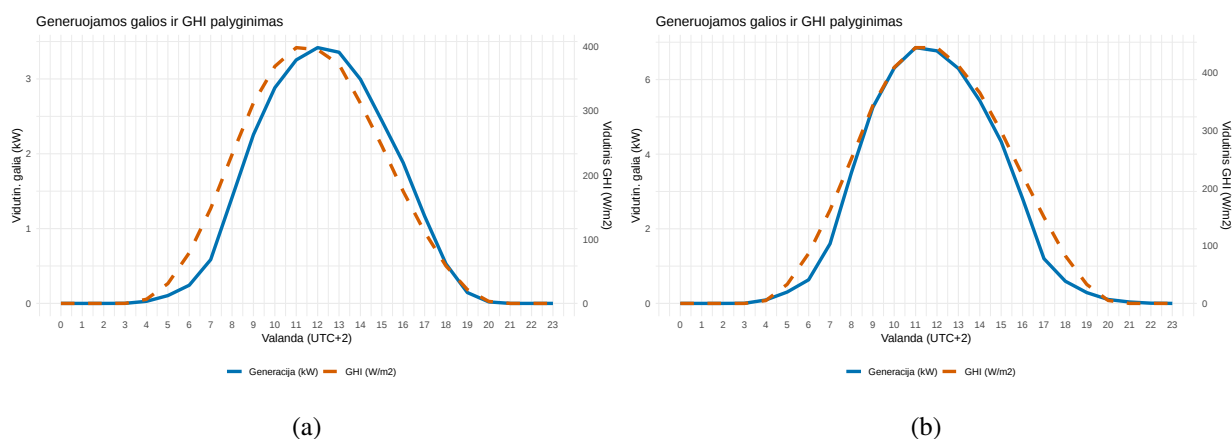


5 pav. Sezoniškumo perkodavimas į kosinuso funkciją Kauno elektrinėje

3.4.4. Elektrinių polinkio ir azimuto nustatymas

Saulės elektrinės generuojamos energijos kiekis yra tiesiogiai priklausomas nuo kolektorių orientacijos erdvėje – azimuto ir pasvirimo kampo. Literatūroje pabrėžiama, kad tikslūs geometriniai parametrai daro stiprią įtaką elektrinių generacijai, jų pikų laikams bei pačiai pikinei galiai (kWp) [19]. Dėl to šie parametrai yra būtini kuriant globalų Lietuvos modelį.

Analizuojant duomenis pastebėta, kad elektrinių valdymo sistemose šie parametrai dažnai nėra užpildomi faktinėmis reikšmėmis, o paliekami pagal nutylėjimą (pvz., azimutas 180° , pasvirimas 0°). Taip pat pastebėta, kad lyginant elektrinių generaciją su palydoviniais saulės spinduliuotės (GHI) duomenimis, atsiranda neatitikimai: kai kurių elektrinių generacijos pikai laike prasilenkia su saulės vidurdieniu, vakarėjant staigiau krinta generuojama elektra (6 pav.). Tai indikuoja, kad elektrinės fiziškai yra nukreiptos ne į pietus (paveikslėlyje a – Anykščių elektrinė).



6 pav. Vėluojantis pikas (a) Anykščiuose ir sutampantis pikas (b) Vilkaviškyje

Siekiant įvertinti šiuos neatitikimus bei padarant modelį pritaikomą daugiau skirtingų elektrinių, šiame darbe azimuto ir pasvirimo kampo nustatymui pritaikytas parametų optimizavimo metodas naudojant gardelės paiešką (angl. *Grid Search*). Šio metodo esmė – simuliuoti teorinę elektrinės generaciją esant skirtingoms azimuto ir pasvirimo kombinacijoms ir lyginti ją su realiais istoriniais duomenimis. Procesas atliktas keliais etapais: pirmiausia sudaryta galimų parametų gardelė, ku-

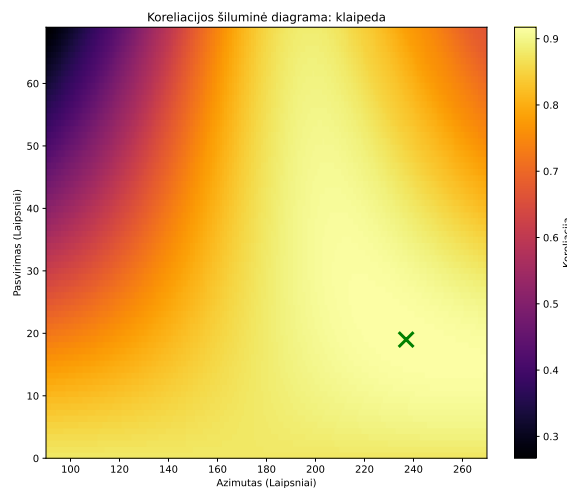
rioje azimutas kinta intervale $[90^\circ, 270^\circ]$, o pasvirimo kampas – $[0^\circ, 70^\circ]$. Tuomet kiekvienam gar-
delės taškui, naudojant *Python pvlb* biblioteką [12] ir istorinius meteorologinius duomenis (DNI,
DHI), buvo apskaičiuojama teorinė krentanti spinduliuotė į plokštumą. Buvo naudojamos tik tos
valandos, kai debesuotumas buvo < 30 . Kiekviena simuliuota elektrinė giedromis dienomis bu-
vo lyginama su faktine elektrinės generacija [15] ir kombinacija su didžiausia Piersono koreliacija
[7] buvo priskiriama atitinkamai elektrinei. Piersono koreliaciją r matematiškai galima užrašyti
formule:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (3.5)$$

Kur:

- n – imties dydis (atrinktų valandų skaičius);
- x_i – faktinė (reali) elektrinės generacija laiko momentu i ;
- y_i – teorinė (simuliuota su *pvlb*) generacija laiko momentu i ;
- $\bar{x}$ – faktinės generacijos vidurkis;
- $\bar{y}$ – teorinės generacijos vidurkis.

Koeficientas r įgyja reikšmes intervale $[-1, 1]$. Šiame uždavinyje buvo ieškoma reikšmės, arti-
miausios 1, kas rodytų stipriausią tiesinį ryšį tarp saulės pozicijos danguje ir realiai pagamintos
elektros energijos, ignoruojant absoliutinius galingumo skirtumus. Visų kombinacijų šiluminė dia-
grama pavaizduota 7 paveiksle (Klaipėdos pavyzdys), kur X žymi didžiausią koreliaciją.



7 pav. Klaipėdos optimalaus azimuto ir polinkio šiluminė diagrama

Gauti duomenys atvaizduoti 2 lentelėje. Lyginant nustatytą azimutą ir kampą Anykščiuose su 6 pa-
veikslu, galima matyti, kad generacijos kreivės viršūnė pasislinkusi į popietines valandas, o vaka-
rinė kreivės dalis išlieka aukšta ilgiau. Tai fiziškai pagrindžia algoritmo nustatytą 20° nuokrypį į va-
karus nuo pietų krypties. Priešingas efektas stebimas Vilkaviškio elektrinėje (Azimutas 167° , Piet-
ryčiai). Čia fiksuojamas staigesnis generacijos kritimas popietinėmis valandomis, kadangi saulė

2 lentelė. Tiriamų saulės elektrinių geometrinių parametrų

Lokacija	Azimutas (°)	Pasvirimo kampas (°)
Anykščiai	200	36
Biržai	174	59
Kaunas	134	16
Klaipėda	237	19
Merkinė	145	35
Šiauliai	235	47
Vilnius	141	34
Jurbarkas	144	44
Ignalina	189	25
Mažeikiai	159	59
Vilkaviškis	167	52

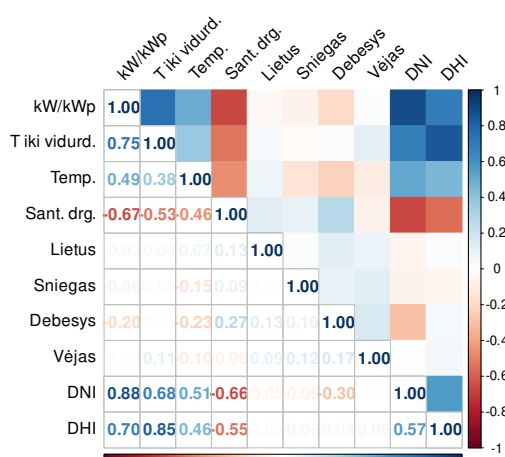
anksčiau pereina į vakarų pusę ir panelės atsiduria nepalankiame kampe. Toks profilis patvirtina pietrytinę orientaciją.

Siekiant didesnio tikslumo, elektrinėms su prieinama vizualine informacija (palydovinėmis nuotraukomis ar objekto fotografijomis) buvo taikomi paieškos režimų apribojimai. Simuliuojamų azimutų ir pasvirimo kampų režiai buvo susiaurinti atsižvelgiant į faktiškai matomą panelių orientaciją, taip eliminuojant galimas algoritmo paklaidas.

Identifikuoti parametrai buvo panaudoti generuojant naują kintamąjį – teorinę plokštumos apšvitą (angl. *Plane of Array irradiance*, toliau – POA_{opt}). Šis išvestinis dydis, sugeneruotas naudojant programavimo kalbos *Python pvlib* biblioteką visam istoriniam laikotarpiui (ne tik giedroms dienoms, kaip buvo daroma bandant nustatyti kampą ir azimutą). Vėliau šis dydis darbe naudojamas kaip papildoma LSTM modelio įvestis, leidžianti modeliams efektyviau atskirti saulės įtaką skirtingų struktūrų elektrinėms.

3.5. Galutiniai duomenys

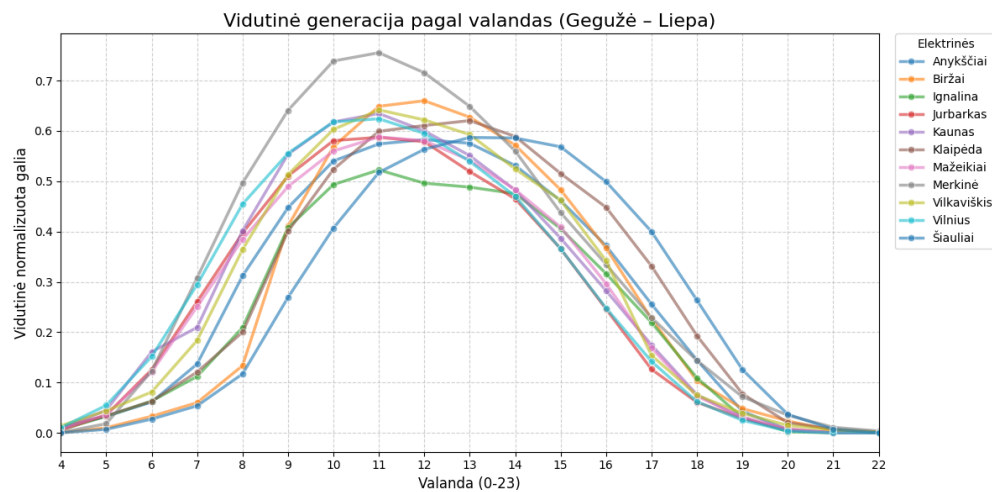
Sutvarkius duomenis buvo sudaryta koreliacinė diagrama (8 pav.).



8 pav. Duomenų koreliacinė diagrama

Joje galime matyti, kad egzistuoja stipri tiesinė koreliacija tarp normuotos galios (kW/kWp) bei laiko iki vidurdienio, DNI ir DHI. Vidutiniškai neigiamai stipri koreliacija pasireiškia su santykine oro drėgme, teigiamai – su temperatūra. Įdomu, kad koreliacija su debesuotumu yra pakankamai maža, tačiau ši diagrama atspindi tik tiesinį sąryšį. Lietuvoje naktys dažnai yra giedros, tačiau tuo metu elektra negeneruojama. Dėl šio fenomeno gaunama tokia maža reikšmė, tačiau debesuotumas modelyje toliau lieka svarbus faktorius.

Iš 9 pav. matomi generavimo pasiskirstymo skirtumai tarp skirtingų elektrinių šiltuoju laikotarpiu. Nors generacija ir yra normuota pagal maksimalią generacijos galią (kWp), pikai ir formos tarp elektrinių skiriasi. Tai pagrindžia būtinybę į modelio apmokymą įtraukti ne tik meteorologinius duomenis, bet ir elektrinės konfigūracijos parametrus (azimutą ir pasvirimo kampą). Tai patvirtina 3.4.4 skyriuje aprašytą inžinerinio požymio POA_{opt} nustatymo svarbą.



9 pav. Elektrinių generacijos normuotas vidutinis pasiskirstymas

4. Modeliavimas

Šiame darbe prognozavimui pasitelkiami du skirtingos architektūros modeliai: ilgalaikės trumpalaikės atminties neuroniniai tinklai (LSTM) bei sprendimų medžių ansambliai (XGBoost). Šis skyrius skirtas pagrįsti šiuos pasirinkimus bei detaliai aprašyti kiekvieno modelio veikimo principus. Taip pat jame bus aprašomas duomenų paruošimo modeliams etapas, efektyvus modelių hiperparametrų parinkimas ir toliau naudojamos vertinimo metrikos

4.1. LSTM modelis

Norint suprasti LSTM pasirinkimą, reikia įvertinti tradicinių rekurentinių neuroninių tinklų (angl. *recurrent neural network (RNN)*) ribotumą. Standartiniai RNN buvo sukurti apdoroti sekas, kur dabartinė išvestis priklauso nuo einamosios įvesties ir ankstesnės paslėptosios būsenos [5]. Teoriškai tai turėtų leisti tinklui „prisiminti“ praeities įvykius. Tačiau mokymo metu, naudojant atgalinio klaidų skleidimo laiką (angl. *Backpropagation Through Time - BPTT*) algoritimą [20], RNN susiduria su nykstančio gradiento problema [4].

Gradientai – tai dalinės išvestinės, nurodančios, kaip reikia koreguoti tinklo svorius siekiant sumažinti klaidą. BPTT procese gradientas yra dauginamas iš svorių matricos kiekviename laiko žingsnyje atgal į praeitį. Jei svoriai yra maži (pvz. < 1), daugkartinis dauginimas lemia gradientų konvergavimą į 0. Tai reiškia, kad tinklas nustoja mokytis iš tolimes praeities duomenų – jis „pamiršta“ informaciją, esančią vis toliau nuo einamojo laiko žingsnio. Saulės energetikoje tai kritinė problema, nes sezoniniai dėsningumai ar ilgalaikės orų tendencijos reikalauja išlaikyti informaciją ilgus periodus.

LSTM modelis, kurį 1997 m. pristatė Hochreiter ir Schmidhuber kaip RNN atšaką [11], sprendžia šią problemą įvedant ląstelės būseną – atskirą informacijos šaltinį, leidžiantį gradientams nepakisti per ilgus laiko intervalus. LSTM geba išlaikyti informaciją tūkstančius laiko žingsnių, todėl jis tapo standartu laiko eilučių uždaviniuose [21].

4.1.1. Vidinė ląstelės architektūra

LSTM tinklo pagrindas yra atminties ląstelė, kurios būseną reguliuoja trys pagrindiniai vartai: užmaršties (forget), įvesties (input) ir išvesties (output) (10 pav.). Kiekviename laiko žingsnyje t , ląstelė priima įvestį x_t ir ankstesnę paslėptąją būseną h_{t-1} . Visi vartų tipai atlieka specifinę funkciją valdant informacijos srautą, naudojant sigmoidinę (σ) ir hiperbolinio tangento ($\tanh$) aktyvacijos funkcijas.

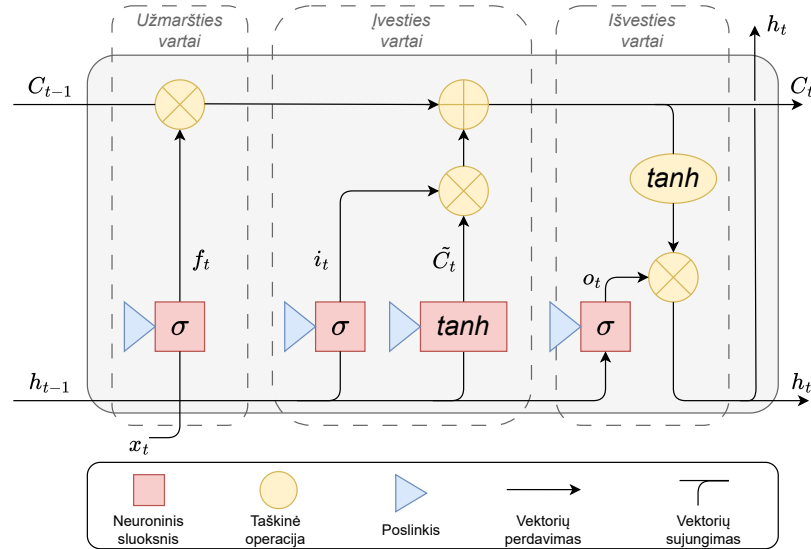
4.1.2. Užmaršties vartai

Pirmasis sprendimas, kurį priima LSTM, yra nustatyti, kuri informacija iš ankstesnės ląstelės būsenos (C_{t-1}) nebėra aktuali ir turi būti pašalinta. Saulės generacijos kontekste tai gali būti informacija apie vakarykštį debesuotumą, kuris nebėra aktualus šiandienos prognozei. Užmaršties vartus (f_t) galima užrašyti lygtimi:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \quad (4.1)$$

kur, x_t yra nauja įvestis (pvz., temperatūra, debesuotumas), h_{t-1} yra ankstesnio žingsnio paslėptoji būsena, W_f ir b_f yra mokomi svoriai ir poslinkiai. Poslinkis b leidžia neuroniniam tinklui pastumti

aktyvacijos funkcijos kreivę, suteikiant lankstumo modeliuoti duomenis, kurie nėra centruoti ties nulių, bei valdyti neuronų aktyvacijos slenkstį. Aktyvacijos σ funkcija grąžina reikšmes intervale [0-1], kur 0 reiškia visiškai pamiršti, 1 – visiškai išlaikyti. Šis mechanizmas leidžia tinklui adaptyviai išvalyti atmintį, kai pasikeičia kontekstas (pvz., pasikeitus sezonui).



10 pav. LSTM ląstelės su užmaršties vartais architektūra

4.1.3. Įvesties vartai ir kandidatų atnaujinimas

Antrasis etapas skirtas naujos informacijos įrašymui į ląstelės būseną. Šis procesas vyksta dviem lygiagrečiais žingsniais:

1. Įvesties vartai (i_t) nusprendžia, kurias reikšmes atnaujinti (arti 1 - naujinti, arti 0 - ignoruoti):

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (4.2)$$

2. Sukuriamas naujų galimų reikšmių vektorius ($\tilde{C}_t$) naudojant tanh funkciją, kuri normalizuoja duomenis intervale [-1, 1]. Jame saugoma potenciali nauja informacija, paruošta pridėjimui prie ląstelės būsenos.

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (4.3)$$

Tada atnaujinama pati ląstelės būsena (C_t):

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (4.4)$$

Senoji būsena C_{t-1} atnaujinama į naują C_t . Senoji informacija padauginama iš užmaršties vartų f_t (pamirštama nereikalinga dalis) ir pridedama nauja informacija $i_t \cdot \tilde{C}_t$.

4.1.4. Išvesties vartai

Galiausiai, modelis generuoja išvestį (h_t), kuri bus naudojama kaip prognozė ar perduodama kitam laiko žingsniui. Išvesties vartai (o_t) nustato, kurią atminties dalį atskleisti.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (4.5)$$

Lastelės būsenai pritaikoma hiperbolinio tangento aktyvacijos funkcija ir tada padauginama iš vartų reikšmės.

$$h_t = o_t \cdot \tanh(C_t) \quad (4.6)$$

Ši architektūra suteikia LSTM modeliui išskirtinį gebėjimą modeliuoti sudėtingas, netiesines laiko priklausomybes, kurios yra būdingos meteorologiniams duomenims ir saulės generacijai.

4.1.5. LSTM taikymas saulės energetikoje

Saulės energetikos prognozavimui LSTM yra ypač tinkamas, nes geba susieti momentines sąlygas (temperatūrą, debesuotumą) su platesniu kontekstu. Tai leidžia modeliui įvertinti situacijas, kai momentiniai orų duomenys gali būti klaidingi bei atsižvelgti į bendrus sezoninius pokyčius. Šiame darbe duomenys skaidomi į 24 valandų (paros) sekas. Kelių paslėptųjų sluoksnių naudojimas leidžia modeliui vertinti ne tik tiesioginę įvestį, bet ir ilgesnio laikotarpio dėsninumus.

Visgi, šio metodo taikymas turi ir ribojimų: LSTM reikalauja didesnių skaičiavimo resursų (GPU), ilgo apmokymo laiko bei kruopštaus hiperparametrų (sluoksnių, neuronų skaičiaus, mokymosi greičio) derinimo.

4.2. Sprendimų medžių ansambliai

Lyginamajai analizei ir rezultatų validavimui pasirinktas „XGBoost“ (angl. *eXtreme Gradient Boosting*) algoritmas. Tai yra vienas efektyviausių ansamblinio mokymosi metodų, pagrįstas sprendimų medžių konstravimu naudojant gradiento stiprinimą [9]. Šis algoritmas literatūroje vertinamas dėl didelio skaičiavimo greičio, efektyvaus atminties panaudojimo ir gebėjimo modeliuoti sudėtingas netiesines priklausomybes struktūrizuotuose duomenyse. Skirtingai nei giliojo mokymosi modeliai, XGBoost suteikia galimybę tiesiogiai įvertinti kintamųjų svarbą. Šiame darbe tai leidžia atlikti komponentų įtakos analizę ir pagrįsti sprendimų (tokių kaip spinduliuotės kampo skaičiavimas) naudą galutinei prognozei.

Pagrindinis šio modelio struktūros principas remiasi nuosekliu sprendimų medžių kūrimu, siekiant optimizuoti specifinę tikslo funkciją $\mathcal{L}(\phi)$. XGBoost pritaiko antros eilės Teiloro eilutės aproksimaciją ir įtraukia reguliarizacijos narius, kas leidžia efektyviau valdyti persimokymą.

4.2.1. Tikslo funkcija

Modelio mokymo metu siekiama minimizuoti tikslo funkciją $\mathcal{L}(\phi)$, kuri susideda iš dviejų dalių: nuostolių funkcijos l (kuri matuoja skirtumą tarp prognozės $\hat{y}_i$ ir realios reikšmės y_i) ir reguliarizacijos nario Ω :

$$\mathcal{L}(\phi) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (4.7)$$

Čia n yra stebėjimų skaičius, o K – medžių skaičius.

4.2.2. Regularizacija ir modelio sudėtingumas

Esminis XGBoost pranašumas prieš standartinius sprendimų medžių ansamblius yra reguliari-
zacijos narys $\Omega(f)$, kuris baudžia modelį už pernelyg didelį sudėtingumą.

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda ||w||^2 \quad (4.8)$$

Kur:

- T – lapų skaičius medyje (kontroliuoja medžio gylį ir struktūrą).
- $||w||^2$ – lapų svorių L2 norma.
- γ (gamma) ir λ (lambda) – hiperparametrai, reguliuojantys baudos dydį už papildomą medžio šakojimąsi.

Regularizacija yra svarbi dirbant su triukšmingais duomenimis, siekiant išvengti persimokymo. Šiame darbe to svarba išauga, nes modelis bus taikomas naujoms, mokyme nenaudotoms elektrinėms. Todėl tokie faktoriai kaip momentiška saulę užstojančias kamino neturi daryti įtakos prognozei.

4.2.3. Optimizavimas naudojant antros eilės aproksimaciją

Tradicinis gradientinis spartinimas naudoja tik pirmosios eilės išvestinę (gradientą) optimizuojant nuostolių funkciją. XGBoost taiko antrosios eilės Teiloro eilutės skleidinį:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^n [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_i(x_i) + \frac{1}{2} h_i f_i^2(x_i)] + \Omega(f_i) \quad (4.9)$$

Čia g_i yra pirmosios eilės gradientas, o h_i – antrosios eilės gradientas (Hesianas). Naudojant h_i , XGBoost mato ne tik klaidos kryptį, bet ir paviršiaus išlinkimą, todėl gali greičiau konverguoti į optimalų sprendinį.

Dėl šių savybių XGBoost yra tinkamas ne tik kaip palyginamasis bazinis modelis, bet ir kaip lygiavertis įrankis prognozuoti saulės elektrinių generaciją, ypač kai duomenys pateikti struktūrizuota forma, apimant visos paros kontekstą.

4.3. Duomenų paruošimas skirtingiems modeliams

Skirtingai nei LSTM modeliai, kurie duomenis apdoroja laiko žingsniais (trimačiai duomenys), XGBoost algoritmas reikalauja dvimačio duomenų pateikimo. Siekiant išsaugoti paros laiko kontekstą ir leisti medžių ansamblui įvertinti ryšius tarp skirtingų valandų, buvo atlikta duomenų transformacija. Kiekvienos paros 24 valandų įrašai, turintys po 15 kintamųjų (14, jei ne paskaičiuotos teorinės apšvietos), buvo sujungti į vieną lentelę su 360 požymių (24×15) (arba 336). Tai leido XGBoost modeliui vienu metu matyti visos paros kontekstą taip turint panašias mokymosi sąlygas kaip LSTM.

4.4. Modelio hiperparametrų optimizavimas

Hiperparametrai yra mašininių modelių dalis, kurią galima keisti, taip labiau pritaikant modelius individualiems duomenims. Šie nulemia modelio struktūrą, jo mokymo procesą, taip turėdami daug įtakos jo tikslumui ir taikymui. Tačiau hiperparametrus reikia parinkti prieš mokymą, priešingu būdu nei modelio parametrus, nustatomus bemokant modelius.

Jų parinkimas gali nulemti modelio persimokymą, sudėtingumą, mokymo laiką, dydį. Tinkamai nustatyti hiperparametrai gali leisti modeliui konverguoti į globalų minimumą, kas ir yra mokymo tikslas.

4.4.1. Hiperjuostinis parametrų parinkimas

Egzistuoja daugybė hiperparametrų parinkimo metodų. Patys elementariausi - aklaiai arba pagal literatūrą parinkti keli hiperparametrus, pilnai apmokyti modelius, juos palyginti pagal testavimo duomenis ir parinkti geriausią. Tačiau šis procesas reikalauja daug mechaninio darbo, kartojimosi, užima daug laiko.

Kitas populiarus metodas yra paieškos gardelės naudojimas (angl. *search grid*), kurios dėka yra išbandomos visos įmanomos hiperparametrų kombinacijos. Šis metodas tikrai parinks geriausią modelį, tačiau visos įmanomos kombinacijos sparčiai auga didinant nustatymų skaičių. Dėl to šio varianto naudojimas reikalauja labai daug resursų bei laiko.

Vienas iš pažangesnių metodų yra hiperjuostinis² parametrų derinimas [13] (angl. *hyperband parameter tuning*³). Pastarasis yra efektyvus algoritmas, skirtas geriausių hiperparametrų nustatymui, siekiantis ištestuoti kuo daugiau, tačiau mažinant skaičiavimo resursus parenkant geriausius pretendentes besiremiant „Skaldymo pusiau paeiliui“ (angl. *Successive Halving (SHA)*) strategija.

Pagrindinė algoritmo mintis yra pradėti mokymo procesą vertinant daug skirtingų hiperparametrų konfigūracijų, tačiau apribojant jų mokymosi resursus, kaip epochas. Apmokius skirtingas versijas seka pagal validacijos duomenis prastai pasirodžiusių kombinacijų eliminavimas ir geresnių mokymas skiriant daugiau resursų. Mokant yra sukuriami pogrupiai, kuriuose bandomi skirtingi mokymo resursų ir parametrų kombinacijų santykiai. Procesas kartojamas iki kol iš daugiau resursų mokymui turėjusių variantų randamas geriausiai pasirodęs.

Iš visų išvardintų metodų šiame darbe ir buvo naudojamas „Hyperband“ metodas LSTM modelių hiperparametrų paieškai.

4.4.2. Vertinimui paruošti hiperparametrai

Remiantis ankstesniame šio darbo etape (MTDP) atliktais eksperimentais su mažesne imtimi (16 priedas), buvo nustatytos preliminarios parametrų ribos, kurios leido geriau apibrėžti galutinę paieškos erdvę. Siekiant sukurti universalų Lietuvos modelį ir efektyviai išnaudoti skaičiavimo resursus, šiame etape LSTM mokymui buvo vertinamos šių hiperparametrų kombinacijos:

- **Perceptronų skaičius** kiekviename paslėptame sluoksnyje (viso buvo 2 sluoksniai). Šio parametro keitimas nulemia modelio dimensiškumą ir galimybę pastebėti duomenų šablonus. Pradinės paieškos reikšmės buvo 50, 100, 150, 200. Tyrimą praplėtus, pasirinktos reikšmės 64, 128, 192, 256, nes ankstesni rezultatai parodė, kad didesnis neuronų skaičius (200) koreliavo su mažesne paklaida, todėl galutiniam modeliui viršutinė riba buvo praplėsta iki 256.

²Toliau bus naudojamas originalus pavadinimas dėl potencialaus vertimo netikslumo

³https://keras.io/api/keras_tuner/tuners/hyperband/

Taip pat pasirinktas žingsnis, suderintas su dvejetainine sistema, siekiant geresnės GPU atminties optimizacijos naudojant *CUDA* architektūrą [16].

- **Išmetimo dažnis** (angl. *dropout rate*) tai modelio reguliarizacijos metodas, kuris padeda išvengti permokymo, dalį įvesčių priskiriant nuliui. Taip gaunamas bendresnis modelis. Pradinės galimos reikšmės: 0,1 - 0,5 (kas 0,1). Išplėtus tyrimą susiaurinta iki 0 iki 0,3 (kas 0,1), nes pirminiai tyrimai parodė, kad agresyvi reguliarizacija ($>0,3$) reikšmingai blogino modelio tikslumą. Be to, šiame etape atsisakyta rekurentinio išmetimo (angl. *recurrent dropout*) naudojimo, pirmenybę teikiant CuDNN bibliotekos optimizacijai. Tai leido žymiai pagreitinti skaičiavimus ir atlikti išsamesnę „Hyperband“ paiešką per ribotą laiką.
- **Optimizacijos funkcijos.** Jos parinkimas nulemia, kaip efektyviai ir greitai modelis atnaujinama savo svorius, siekdamas minimizuoti prognozavimo klaidą. Nuo jos tiesiogiai priklauso mokymosi proceso stabilumas bei gebėjimas rasti geriausią įmanomą (globalų) sprendinį, išvengiant užstrigimo lokaliuose minimumuose. Galimos reikšmės:
 1. Adam - dinamiškai reguliuoja mokymosi greitį kiekvienam parametrai. Jam buvo koreguojamas mokymosi greičio parametras (0,0001 - 0,01);
 2. RMSprop - dinamiškai reguliuoja mokymosi greitį kiekvienam parametrai, gerai tvarkosi su skirtingu modelio dalių mokymosi greičiu. Jam buvo koreguojamas mokymosi greičio parametras (0,0001 - 0,01);

Stochastinio gradiento nusileidimo (SGD) metodas pirminiuose bandymuose demonstravo prasčiausius rezultatus lyginant su adaptyviais (Adam ir RMSprop) metodais, todėl galutiniame modeliavime jo buvo atsisakyta.

4.4.3. Atsitiktinis gardelės parametru parinkimas

Nors XGBoost algoritmas pasižymi greitu veikimu, jo tikslumas stipriai priklauso nuo teisingai parinktų hiperparametru. Kadangi pilna gardelės paieška, tikrinanti visas įmanomas kombinacijas, reikalauja didelių skaičiavimo resursų, šiame darbe buvo pritaikyta atsitiktinė paieška. Šis metodas iš anksto apibrėžtoje parametru erdvėje atsitiktinai atrinka fiksuotą skaičių kombinacijų (šiuo darbe 50). Optimizavimo procesas buvo vykdomas naudojant GPU akceleraciją, kas leido atlikti visus 50 eksperimentų per maždaug 15 minučių.

Validacijai, analogiškai kaip ir LSTM atveju, buvo naudojama fiksuota validacinė aibė – Jurbarko saulės elektrinės duomenys, siekiant išvengti duomenų nutekėjimo ir užtikrinti sąžiningą palyginimą tarp skirtingų architektūrų.

Paieškos metu buvo optimizuojami šie esminiai modelio hiperparametrai:

- Medžių skaičius: nurodo, kiek sprendimų medžių sudarys galutinį ansamblį. Tirta aibė: [300, 500, 800]. Didesnis skaičius leidžia modeliui išmokyti sudėtingesnius ryšius, tačiau didina persimokymo riziką ir skaičiavimo laiką.
- maksimalus medžio gylis: Kontroliuoja modelio sudėtingumą ir gebėjimą fiksuoti specifines sąveikas tarp kintamųjų (pvz., debesuotumo ir paros laiko). Tirta aibė: [4, 6, 8]. Saulės generacijos duomenims per gilus medis gali lemti triukšmo įsiminimą.
- mokymosi greitis: Nustato, koku mastu kiekvienas naujas medis koreguoja bendrą modelio klaidą. Tirta aibė: [0.03, 0.05, 0.1]. Mažesnės reikšmės reikalauja daugiau medžių, tačiau dažniausiai užtikrina geresnį modelio apibendrinimą.

- imties dalis: Nurodo, kokia dalis mokymo duomenų atsitiktinai atrenkama kiekvieno medžio kūrimui. Tirta aibė: [0.7, 0.9]. Reikšmė < 1.0 įveda stochastiškumą, kuris veikia kaip reguliarizacija ir mažina persimokymą.
- stulpelių dalis: Nurodo, kokia dalis požymių (stulpelių) naudojama kiekvienam medžiui. Kadangi modelyje naudojamas išskleistas 24 valandų kontekstas (360 kintamųjų), šis parametras (reikšmės [0.7, 0.9]) neleidžia modeliui pernelyg priklausyti nuo kelių dominuojančių kintamųjų.
- minimalus „vaiko“ svoris (angl. *child weight*): Nurodo minimalią svorių sumą, reikalingą naujai šakai sukurti. Tirta aibė: [1, 5, 10]. Didesnės reikšmės veikia kaip filtras, neleidžiantis modeliui kurti taisyklių, pagrįstų tik keletu išskirčių (pvz., trumpalaikiu šešėliu), taip didinant modelio konservatyvumą ir stabilumą.

4.5. Keras

Šiame darbe bus naudojama giliajam mokymui skirta atviro kodo programų programavimo sąsaja (API) „Keras“. Ši *Python* biblioteka leidžia gana greitai ir paprastai mokytis, testuoti neuroninius tinklus, su jais eksperimentuoti. „Keras“ yra vienas populiariausių būdų mokytis giliuosius neuroninius tinklus (tuo pačiu ir LSTM) tiek praktikoje, tiek akademinėje veikloje, todėl ši sąsaja pasirinkta ir šiame darbe.

4.6. Dviejų etapų optimizavimas

Siekiant išvengti ankstesnio tyrimo (MTDP) ribotumų, kai validacijai buvo naudojami tik pasutiniai 25 % duomenų (dominuojami vieno pavasario sezono), šiame darbe pasirinkta plačiau pritaikoma testavimo strategija. Siekiant efektyviai išnaudoti skaičiavimo resursus ir sukurti universaliai pritaikomą modelį, procesas buvo išskaidytas į du etapus:

1. Architektūros ir hiperparametrų paieška. Pirmajame etape, naudojant Hyperband algoritmą (LSTM modeliui) bei atsitiktinę paiešką (XGBoost), buvo ieškoma optimalios tinklo struktūros. Siekiant išvengti duomenų nutekėjimo laike, validacijai buvo pasirinktas ne laiko tarpsnis, o atskira vietovė – Jurbarko saulės elektrinė. Ši elektrinė pasirinkta dėl savo geografinės padėties (sąlyginai netoli Lietuvos centro), darant prielaidą, kad ji geriausiai atspindi vidutinės šalies meteorologines sąlygas. Modelis buvo mokomas su likusių 10-ies elektrinių duomenimis. Testavimas šioje fazėje nebuvo atliekamas.
2. Erdvinis validavimas: fiksuoti optimalūs hiperparametrai, gauti pirmajame etape, buvo naudojami modelių apmokymui ir testavimui taikant strategiją, kai viena elektrinė paliekama testavimui, 10 - mokymui ir validacijai. Šiame etape buvo perskaičiuojami tik modelio svoriai, bet nekeičiami hiperparametrai. Tam buvo atliekama 11 atskirų ciklų, kad kiekviena elektrinė būtų 1 kartą pakliuvusi į testavimo aibę, 10 kartų į mokymo.

4.7. Vertinimo metrikos

Modelio mokyme kaip nuostolių funkcija buvo naudojama vidutinė kvadratinė paklaida (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2.$$

Modelio interpretacijai naudojama vidutinė absoliuti paklaida, kuri leidžia lengviau interpretuoti rezultatus:

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i|.$$

Čia n - imties dydis, Y - reali reikšmė, $\hat{Y}$ - prognozuota reikšmė.

5. Rezultatai

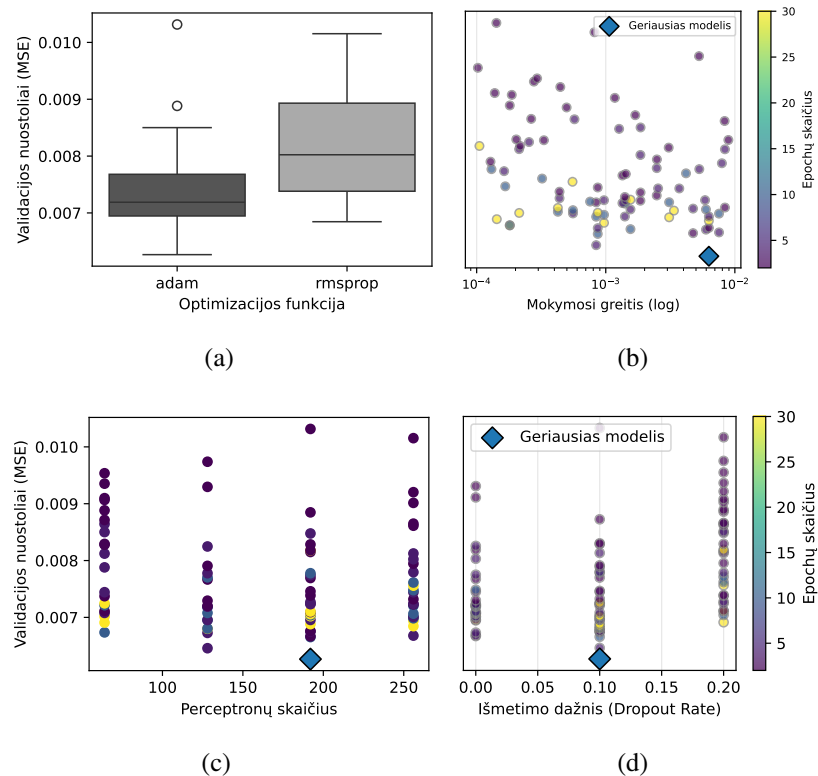
Modelių mokymas ir eksperimentai buvo atliekami naudojantis „Google Colab Pro” debesų kompiuterijos resursais. Skaičiavimų spartinimui aktyvuota GPU akceleracija su „Tesla T4” vaizdo plokštėmis bei pasirinktas praplėstos operatyviosios atminties (angl. *High-RAM*) režimas.

Lyginant su pradiniu tiriamuoju darbu, duomenų imtis buvo reikšmingai išplėsta: eilučių skaičius padidėjo nuo 80436 iki 154813. Stulpelių skaičius sumažėjo iki 15, nes buvo atsisakyta perteklinių kategorinių kintamųjų, o 24 valandų laiko kontekstas modeliams buvo pateikiamas skirtingais formatais (kaip 3D sekos LSTM tinklui ir kaip išskleisti vektoriai XGBoost modeliui).

5.1. Geriausias modelis

5.1.1. LSTM

LSTM modelio hiperparametrų paieškai naudotas „Hyperband” algoritmas. Optimizavimo procesas, kurio metu buvo ištirta 90 skirtingų modelio konfigūracijų, truko apie 2 valandas. Geriausias modelis, naudojantis visus kintamuosius, pasiekė validacijos MSE reikšmę lygią $\approx 0,0063$, o MAE $\approx 0,03872$. Šie rezultatai gauti naudojant 192 neuronų paslėptuosiuose sluoksniuose, 0,1 išmetimo dažnį, $\approx 0,00629$ mokymosi greitį ir Adam optimizacijos funkciją.



11 pav. Hyperband algoritmo nuostolių funkcijų diagramos pagal optimizacijos funkciją (a), mokymosi greitį (b), perceptronų skaičių (c) ir išmetimo dažnį (d)

Kokius validacijos duomenų tikslumus skirtingos konfigūracijos pasiekė galima matyti 11 paveiksle. Iš stačiakampės diagramos (a) galima matyti, kad vidutiniškai geriausiai pasirodė modeliai naudojantys Adam optimizavimo funkciją. Iš (b) sklaidos diagramos galima pastebėti, kad net ir su nedaug mokymosi epochų kai kurie modeliai pasirodė labai gerai. Dauguma

jų naudojo mažą mokymosi greitį. Iš (c) sklaidos diagramos matosi, kad buvo pakankamai gerų modelių net ir su 64 perceptronų paslėptuose sluoksniuose. Iš (d) sklaidos diagramos - kad optimaliausias išmetimo dažnis buvo 0, 1.

Siekiant įvertinti sprendimo vertinti saulės elektrinių azimutą ir pasvirimo kampą ir iš jų suskaičiuoti teorinę plokštumos apšvitą (3.4.4 skyrius), buvo apmokyti modeliai be POA_{opt} kintamojo. Šis procesas truko apie 1,5 valandos. Geriausias modelis be POA_{opt} pasiekė prastesnius rezultatus: validacijos MSE – 0,0072, o MAE – 0,0426 su 256 neuronų paslėptuosiuose sluoksniuose, 0,1 išmetimo dažniu, mokymosi greičiu $\approx 0,00299$ ir Adam optimizacijos funkcija (17 pav.). Tai preliminariai parodo, kad geometrinių parametrų įtraukimas pagerino modelio tikslumą apie 10 %.

5.1.2. XGBoost

XGBoost modelio optimizacijai pritaikyta atsitiktinė paieška naudojant 50 iteracijų. Dėl efektyvaus histogramų metodo ir GPU panaudojimo visas optimizavimo procesas užtruko tik apie 15 minučių – tai yra maždaug 8 kartus greičiau nei LSTM paieška.

Geriausias XGBoost modelis pasiekė validacijos MSE reikšmę lygią 0,0060. Tai yra konkurencingas rezultatas, minimaliai lenkiantis geriausią LSTM modelį (0,0063), tačiau pasiektas per trumpesnę mokymo laiką. 3 lentelėje apibendrinama tirta parametrų erdvė ir geriausios rastos hiperparametrų reikšmės.

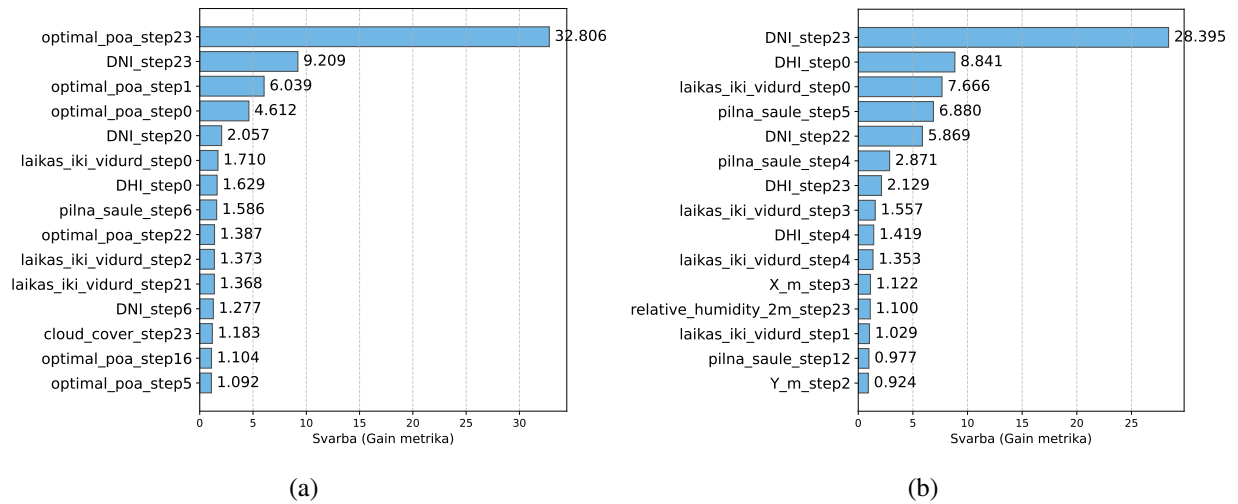
3 lentelė. XGBoost hiperparametrų paieškos erdvė ir geriausi rezultatai

Hiperparametras	Tirta erdvė	Geriausia reikšmė
Medžių skaičius ($n_estimators$)	[300, 500, 800]	800
Maksimalus gylis (max_depth)	[4, 6, 8]	6
Mokymosi greitis ($learning_rate$)	[0.03, 0.05, 0.1]	0.05
Imties dalis ($subsample$)	[0.7, 0.9]	0.9
Stulpelių dalis ($colsample_bytree$)	[0.7, 0.9]	0.7
Min. „vaiko“ svoris (min_child_weight)	[1, 5, 10]	1

Geriausia kombinacija rodo, kad modelis geriausiai veikia mokydamasis lėčiau, bet su didesniu medžių skaičiumi. Taip pat parametras $colsample_bytree=0.7$ indikuoja, kad modelis efektyviai mažina persimokymą, kiekvienam medžiui naudodamas tik 70 % turimų požymių. Reikšmė $min_child_weight = 1$ indikuoja, kad saulės elektrinių generacijos duomenys pasižymi dėsningumais, kuriems aprašyti reikalingas didelis modelio lankstumas.

Kaip ir su LSTM modeliu, buvo atliktas analogiškas XGBoost modelio optimizavimas eliminuojant POA_{opt} kintamąjį. Procesas truko panašiai kaip ir pilno modelio atveju – apie 15 minučių. Geriausias modelis be POA_{opt} pasiekė validacijos MSE = 0,008. Lyginant šį rezultatą su pilnu modeliu, kurio MSE siekė 0,006, matomas tikslumo sumažėjimas. Tai preliminariai patvirtina, kad geometrinių parametrų (azimuto, polinkio ir iš jų išvestos teorinės apšvitos) įtraukimas yra svarbus ne tik LSTM, bet ir sprendimų medžių ansambliams, norint pasiekti geresnę prognozavimo tikslumą. Taip pat verta paminėti, kad geriausių hiperparametrų kombinacija modeliui be POA_{opt} kintamojo išliko identiška pilnam modeliui (3 lentelė).

XGBoost modelis taip pat leidžia nustatyti kintamųjų svarbą, ko negalima padaryti su LSTM modeliu. Tam buvo nubraižytas 12 paveikslas, kuriame galima matyti modelio su POA_{opt} parametrų svarbą (a) ir be (b).



12 pav. Svarbiausi XGBoost parametrai įtraukus (a) ir neįtraukus (b) POA_{opt}

Svarba vertinama pagal „Gain” metriką, kuri parodo, kiek vidutiniškai kiekvienas kintamasis sumažino prognozės paklaidą sprendimų medžiuose. Modelyje su teorine plokštumos apšvita svarbiausias kintamasis yra $optimal_poa_step23$ (teorinė apšvita į plokštumą einamuoju laiku), kurio svarbos rodiklis ($\approx 32,8$) yra daugiau nei 3,5 karto didesnis už artimiausią meteorologinį kintamąjį DNI_step23 ($\approx 9,2$). Tai patvirtina prielaidą, kad saulės elektrinės generacija tiesiogiai priklauso ne nuo bendrosios atmosferos spinduliuotės, o nuo konkretaus energijos kiekio, krentančio statmenai į kolektoriaus paviršių. Modelis šį ryšį sėkmingai identifikavo kaip patikimiausią. Taip pat grafike matyti, kad tarp svarbiausių kintamųjų patenka $optimal_poa_step0$ bei $optimal_poa_step1$ (duomenys prieš 23–24 val.). Tai rodo, kad modelis išmoko paros cikliškumą ir naudoja vakarykštę to paties laiko situaciją kaip atskaitos tašką, kad įvertintų šiandienos potencialą.

Pašalinus POA_{opt} kintamąjį, dominuojančią poziciją užima DNI_step23 (tiesioginė normalioji spinduliuotė) su svarbos rodikliu $\approx 28,4$. Nors tai yra stiprus rodiklis, jo „Gain” reikšmė yra mažesnė nei $optimal_poa$ pirmame modelyje. Svarbu pastebėti, kad be POA_{opt} modelis yra priverstas labiau pasikliauti netiesioginiais indikatoriais. Pavyzdžiui, kintamojo DHI_step0 (difuzinė spinduliuotė prieš 24 val.) svarba išauga iki $\approx 8,8$, o $laikas_iki_vidurd_step0$ iki $\approx 7,7$. Tai rodo, kad neturėdamas tikslios geometrinės informacijos, algoritmas bando ją „atkurti” derindamas saulės padėties laiką su bendraisiais spinduliuotės duomenimis, tačiau, kaip parodė validacijos rezultatai (didesnė MSE paklaida), ši strategija yra mažiau efektyvi.

Ši analizė byloja, kad išvestinis $optimal_poa$ kintamasis nėra perteklinė informacija, o kritinis modelio komponentas. Jis suteikia algoritmui tiesioginį fizikinį ryšį, kurį naudojant pasiekiamas didesnis tikslumas ir stabilumas, lyginant su modeliu, kuris remiasi tik meteorologiniais duomenimis.

5.2. Erdvinio prognozavimo tikslumo analizė

Šiame skyriuje atliekamas išsamus keturių skirtingų modelių palyginimas, siekiant įvertinti jų tikslumą prognozuojant saulės elektrinių generaciją naujose, mokymo metu nematytose vietovėse, taip įrodant modelio konceptualumą bei išrenkant geriausią modelio tipą, duomenų aibę. Tyrimui buvo apmokyti 11 modelių, iš kurių kiekvienam 10 vietovių buvo naudojamos mokymui ir validavimui, o likusi 1 - testavimui. Modeliams buvo naudoti geriausi nustatyti hiperparametrai iš 5.1 skyriaus.

5.2.1. Tyrimo eiga ir metodika

Siekiant užtikrinti pačios idėjos rezultatų patikimumą tarp skirtingų, buvo atliktas testavimas visose 11 tyrimo vietovių. Kiekvienos iteracijos metu:

1. Viena vietovė išskiriama kaip testinė aibė. Likusios 10 vietovių sudaro mokymo aibę. Duomenys skaidomi 75% mokymui, 25% validacijai, pastarajai nukerpant paskutines eilutes.
2. Duomenų normalizavimas atliekamas iš naujo kiekviename cikle, parametrus apskaičiuojant tik iš naujos mokymo aibės. Tai leidžia modeliams būti nepriklausomiems.
3. Visi 4 modeliai sukurti tame cikle testuojami su likusia vietoje, įvertinamos MAE ir RMSE metrikos.

Šiame etape buvo lyginami keturi modeliai:

- LSTM (su POA_{opt}): Giliojo mokymosi modelis su inžineriniu geometrinės apšvitos kintamuoju.
- LSTM (be POA_{opt}): Kontrolinis LSTM modelis tik su meteorologiniais duomenimis.
- XGBoost (su POA_{opt}): Sprendimų medžių ansamblis su geometrinės apšvitos kintamuoju.
- XGBoost (be POA_{opt}): Kontrolinis XGBoost modelis.

Praėjusiame skyriuje buvo galima pastebėti POA_{opt} svarbą pagal validacijos duomenis. Modelių kūrimas su ir be šio parametro su testinėmis aibėmis parodys, ar tikrai šis kintamasis prisideda prie tikslumo.

5.2.2. Rezultatų analizė

Gauti prognozavimo rezultatai (MAE ir RMSE metrikos) kiekvienai lokacijai pateikiami 4 lentelėje.

Tyrimas parodė, kad šio tipo uždaviniui XGBoost algoritmas veikia stabiliau ir tiksliau nei LSTM. Vidutinė $XGBoost_{POA}$ absoliutinė paklaida per visas vietas siekia 0,0356, tuo tarpu $LSTM_{POA}$ – 0,0417. Ypač ryškus skirtumas atsiranda Klaipėdoje. LSTM modelio paklaida čia išaugo iki 0,0643, tuo tarpu XGBoost išlaikė aukštą tikslumą (0,0298). Taip gali būti todėl, kad sprendimų medžių ansambliai yra atsparesni lokalioms meteorologinėms anomalijoms (pvz., pajūrio vėjams ar greitai kintančiam debesuotumui), kurios gali sukelti triukšmą smulkesniuose neuroniniuose tinkluose su mažiau paslėptų sluoksnių (visi LSTM modeliai čia buvo su 2 paslėptais sluoksniais).

Išvestino kintamojo POA_{opt} svarba buvo patvirtinta. Modeliuose su įtrauktu šiuo kintamuoju vidutinė paklaida yra mažesnė. Tai rodo, kad fizikinės saulės geometrijos įtraukimas leidžia modeliui geriau generalizuoti ir tiksliau prognozuoti generaciją nepriklausomai nuo geografinės vietos.

Geriausias modelis $XGBoost_{POA}$ demonstravo mažiausią rezultatų sklaidą tarp skirtingų miestų (nuo 0,0280 Kaune iki 0,0440 Merkinėje). Tai indikuoja, kad sukurtas modelis yra sėkmingai generalizuotas ir tinkamas naudoti visoje Lietuvos teritorijoje, nes jo tikslumas drastiškai nekrenta keičiant vietas.

Verta pastebėti ir Anykščių testinėje aibėje laimėjusį modelį - XGBoost be teorinės plokštumos apšvitos. Tai parodo, kad galbūt Anykščiuose buvo pakankamai netiksliai įvertinti azimuto ir pasvirimo kampo parametrai. Tai savaime nulemia ir netikslų POA_{opt} išvestinį dydį.

4 lentelė. Galutiniai modelių tikslumo rezultatai (MAE ir RMSE) skirtingose lokacijose

Lokacija	LSTM _{POA}		LSTM		XGB _{POA}		XGB	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Anykščiai	0,0300	0,0645	0,0379	0,0793	0,0292	0,0632	0,0288	0,0617
Biržai	0,0380	0,0739	0,0593	0,1166	0,0381	0,0777	0,0386	0,0790
Ignalina	0,0470	0,0818	0,0427	0,0845	0,0370	0,0761	0,0450	0,0905
Jurbarkas	0,0430	0,0886	0,0484	0,0981	0,0394	0,0827	0,0448	0,0913
Kaunas	0,0271	0,0594	0,0318	0,0644	0,0280	0,0613	0,0297	0,0629
Klaipėda	0,0643	0,1179	0,0567	0,1134	0,0298	0,0637	0,0374	0,0784
Mažeikiai	0,0534	0,1071	0,0795	0,1556	0,0420	0,0871	0,0568	0,1126
Merkinė	0,0482	0,0892	0,0496	0,0903	0,0440	0,0826	0,0444	0,0856
Šiauliai	0,0424	0,0805	0,0618	0,1208	0,0386	0,0737	0,0496	0,0949
Vilkaviškis	0,0345	0,0699	0,0372	0,0750	0,0358	0,0695	0,0348	0,0702
Vilnius	0,0304	0,0623	0,0322	0,0647	0,0297	0,0633	0,0332	0,0667
Vidurkis	0,0417	0,0814	0,0488	0,0966	0,0356	0,0728	0,0403	0,0813

5.3. Lokalaus ir bendro modelių palyginimas

Šiame skyriuje atlikta detali lyginamoji analizė, siekiant nustatyti, kiek tikslumo prarandama naudojant universalų (bendrą) modelį vietoje lokalaus, apmokyto tik su konkrečios elektrinės istoriniais duomenimis. Tyrimui naudota Vilkaviškio saulės elektrinė, turinti vieną ilgiausių stebėjimo istorijų (virš 2 metų). Bendras modelis buvo pasirinktas *XGBoost_{POA}* dėl savo stabilumo per visas vietas. Lokaliai modeliui buvo pasirinktas XGBoost modelis su *POA_{opt}* duomenimis bei jau anksčiau nustatytais optimaliais hiperparametrais.

5.3.1. Rezultatų apžvalga

Siekiant palyginti lokalų ir bendrą modelius, standartinis duomenų padalijimas į mokymo ir testavimo aibes (pvz. 80/20) šiuo atveju būtų buvęs nepatikimas dėl riboto duomenų kiekio (2 metai), dėl ko testavimui būtų atitekę tik vieno sezono mėnesiai. Dėl to paklaidos taip pat gali būti nekorektiškos: jei testavimui būtų priskirtas tik šaltas periodas, jos būtų mažesnės, jei šiltas - didesnės. Todėl šiame tyrime pritaikytas „Augančio lango“ (angl. *Expanding Window*) principas, atliekant 3 iteracijas. Vienas šio metodo privalumas yra realaus scenarijaus simuliacija: metodas atkartoja kaip lokalus modelis būtų mokomas iš riboto, bet vis augančio duomenų kiekio. Pradžioje (1 iteracija pavyzdyje) modelis turi sukaupęs mažiau istorinių duomenų, todėl jo tikslumas gali būti mažesnis. Laikui bėgant (2 ir 3 iteracijos), į mokymo aibę įtraukiami naujai gauti duomenys, todėl modelis geriau prisitaiko prie elektrinės specifikos. Kita nauda yra sezoninis padengimas: atlikus tris testus skirtingais laikotarpiais, užtikrinama, kad modeliai būtų patikrinti per skirtingas meteorologines sąlygas. Testavimas padengs šiuos sezonus:

1. Iteracija (Ruduo): Testuojama, kaip modeliai tvarkosi su mažėjančia saulės spinduliuote ir dažniais debesuotumais.

2. Iteracija (Žiema): Vertinamas tikslumas esant minimaliai generacijai ir galimam sniego poveikiui.
3. Iteracija (Pavasaris): Svarbiausias etapas, kuomet generacija yra viena didžiausių, o orai permainingi, kas paprastai nulemia didesnes paklaidas.

Galiausiai augančio lango metodas bus geras bendro modelio stabilumo testas: pastarasis modelis visose iteracijose išlieka tas pats (apmokytas 5.2 skyriuje ant kitų 10-ies elektrinių). Tokiu būdu tikrinama, ar jis geba išlaikyti stabilų rezultatą skirtingais metų laikais, lyginant jį su besimokančiu lokaliu modeliu.

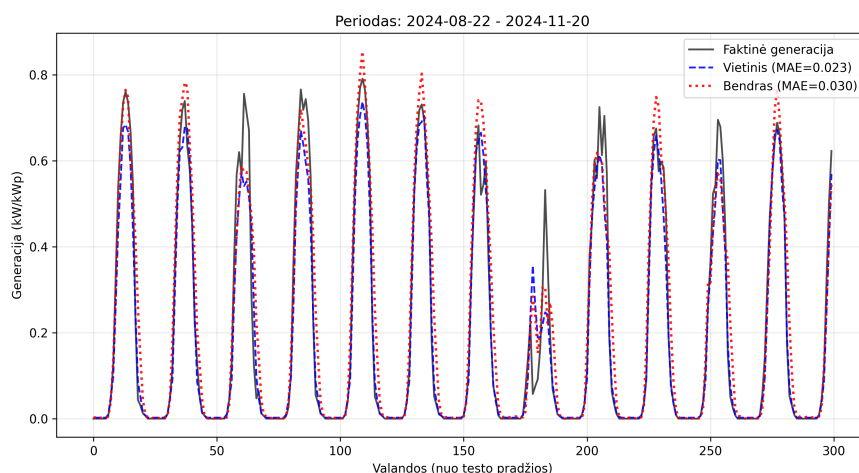
Atlikus testavimą, gauti rezultatai (žr. 5 lentelė) rodo lokalaus modelio pranašumą visuose testavimo perioduose.

5 lentelė. Lokalaus ir Generalizuoto modelių tikslumo palyginimas (Vilkaviškio atvejo analizė)

Iteracija	Testavimo periodas	Sezonas	Lokalus MAE	Bendras MAE
1	2024-08-22 – 2024-11-20	Ruduo	0.0234	0.0298
2	2024-11-20 – 2025-02-18	Žiema	0.0072	0.0109
3	2025-02-18 – 2025-05-19	Pavasaris	0.0345	0.0484
Vidurkis:			0.0217	0.0297

Lokalus modelis pasiekė 0.0217 MAE, tuo tarpu bendras modelis – 0.0297. Žiemos periodu (itaracija 2) abu modeliai demonstravo itin mažas paklaidas (0.007 prieš 0.011) dėl žemos bendros generacijos. Pavasario periodu (itaracija 3) skirtumas išryškėjo labiausiai (0.034 prieš 0.048). Rudenį (itaracija 1) jis buvo mažiau ryškus, nes lokalus modelis turėjo tik šiek tiek virš metų duomenų, kas gali nulemti prastesnį pritaikymą.

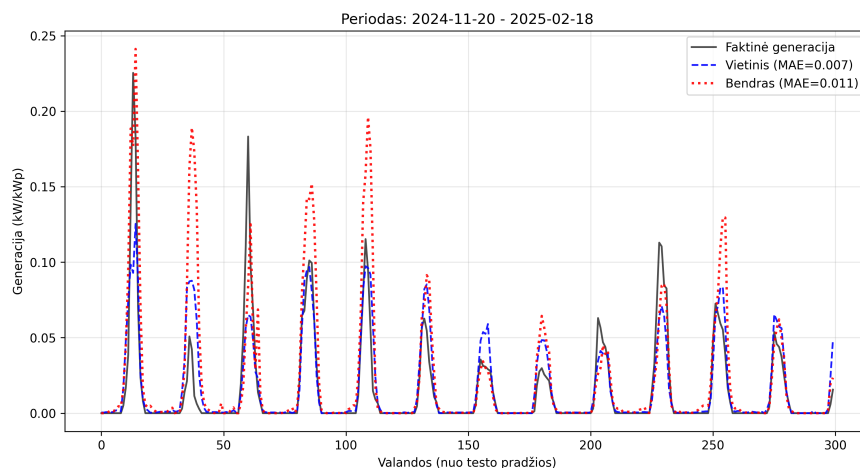
Analizuojant prognozių grafikus (pav. 13, 14, 15), galima pastebėti tendenciją, kad bendras modelis yra linkęs prognozuoti didesnę generaciją piko metu nei faktiškai užfiksuojama Vilkaviškyje.



13 pav. Vilkaviškio realūs ir prognozuoti duomenys (vietinio ir globalaus modelių), 1 iter.

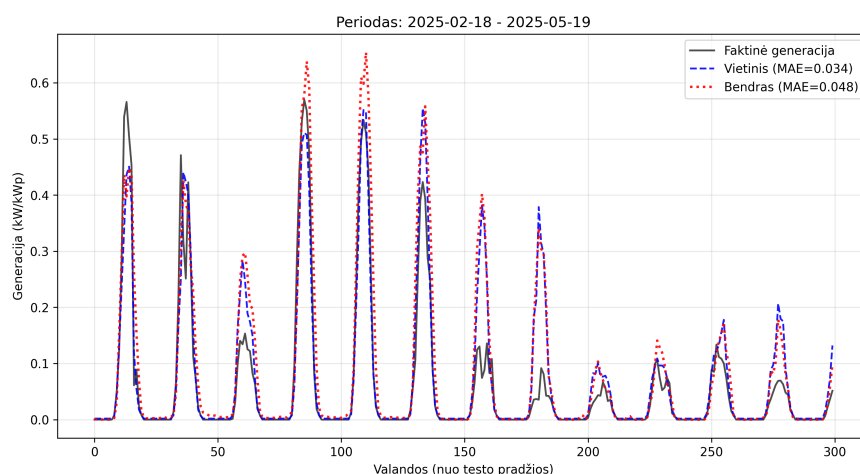
Tai puikiai matoma 14,15 grafikuose, kur raudona kreivė viršija faktinę (juodą) ir netgi lokalią (mėlyną) kreives. Tą buvo galima pastebėti ir pradiniam darbe (18 ir 19 priedai), kur globalus

modelis prognozavo daugiau, nei iš tiesų buvo generuojama.



14 pav. Vilkaviškio realūs ir prognozuoti duomenys (vietinio ir globalaus modelių), 2 iter.

Šį neatitikimą galima bandyti pagrįsti tuo, kad lokalus modelis išmoko specifinius šios elektrinės apribojimus (pvz., galimas krentantis šešėlis, inverterio efektyvumas), kurių nėra bendruose duomenyse. Bendras modelis buvo apmokytas su 10 kitų elektrinių duomenimis. Gali būti, kad kitos elektrinės yra techniškai efektyvesnės arba geriau orientuotos nei Vilkaviškio jėgainė, todėl modelis tikisi didesnio našumo esant toms pačioms oro sąlygoms.



15 pav. Vilkaviškio realūs ir prognozuoti duomenys (vietinio ir globalaus modelių), 3 iter.

Iš grafikų galima matyti, kad ir lokalus modelis nėra visiškai tikslus ir turi panašių problemų, kaip ir globalus. Šiuos netikslumus lemia ne tik modelių architektūros ribotumai, bet ir paties duomenų šaltinio prigimtis. Galima išskirti dvi pagrindines netikslumų priežastis:

1. Meteorologinių duomenų aproksimacija: Tyrime naudojami meteorologiniai duomenys yra gauti iš palydovinių stebėjimų ir reanalizės modelių, o ne iš vietinės meteorologinės stotelės, esančios šalia elektrinės. Palydoviniai duomenys dažnai pateikiami tinkleliu (pvz., 11×11 km ar 30×30 km). Tai reiškia, kad modelis gauna informaciją apie vidutinį debesuotumą tame plote. Jei virš konkrečios elektrinės tuo metu yra lokalus debesis, o aplinkui giedra – palydovas to gali nefiksuoti, todėl modelis prognozuos aukštą generaciją, nors realybėje ji bus sumažėjusi.

2. Generacijos duomenų patikimumas: kintamasis (y) – faktinė elektros gamyba – yra gaunamas iš „Solarman” sistemos. Svarbu pažymėti, kad tai nėra tiesioginiai, neapdoroti inverterio rodmenys realiu laiku. Taip pat reikia pabrėžti, kad „Solarman” platforma galimai atlieka pirminį duomenų apdorojimą – duomenys yra siunčiami paketais (pvz., kas 5 minutes) ir vėliau agreguojami, tai gali nulemti laiko sinchronizacijos paklaidas, duomenų glodinimą įtraukiant išskirtis ir panašiai. Galiausiai nėra tiksliai žinoma, kokius filtravimo algoritmus taiko gamintojas prieš pateikdamas duomenis vartotojui sąsajoje, iš kurios ir buvo gauti duomenys. Todėl modelis bando išmokyti dėsningumus iš duomenų, kurie jau gali būti iš dalies modifikuoti, o tai gali įnešti papildomo triukšmo mokymo procese.

5.3.2. Bendrojo modelio pranašumai

Nors lokalus modelis yra tikslesnis, svarbu įvertinti laiko ir duomenų kaštus. Lokalus modelis šiame tyrime reikalavo beveik 2 metų istorinių duomenų kaupimo, kad pasiektų tokį tikslumą. Tuo tarpu bendras modelis, nors ir turėdamas šiek tiek didesnę paklaidą, gali būti diegiamas tą pačią dieną, kai elektrinė prijungiama prie tinklo.

Bendras modelis taip pat atveria galimybes ir investiciniam planavimui. Įvedus planuojamus techninius parametrus (azimutą, pasvirimo kampą, instaliuotą galią kWp, koordinates) ir pasitelkus istorinius meteorologinius duomenis, modelis gali veikti kaip generacijos simulatorius dar neegzistuojančioms elektrinėms. Tai leidžia įvertinti konkrečios vietovės efektyvumą, apskaičiuoti tikėtiną metinę gamybą ir projekto atsiperkamumą dar prieš pradedant montavimo darbus. Toks įrankis suteikia vertingą informaciją vartotojams sprendžiant dėl investicijų į atsinaujinančią energetiką tikslingumo.

Tad nors lokalus modelis yra tikslesnis turint pakankamai duomenų apie elektrinę, tačiau bendras modelis demonstruoja pakankamą tikslumą ($MAE < 0.05$) universaliai pritaikymui.

6. Išvados bei rekomendacijos

Šiame darbe buvo ištirtos saulės elektrinių generuojamos galios prognozavimo galimybės taikant giliojo mokymosi (LSTM) ir mašininio mokymosi (XGBoost) algoritmus Lietuvos klimato sąlygomis. Atlikus išsamią 11-os skirtingų elektrinių duomenų analizę ir modeliavimo eksperimentus, gautos šios esminės nustatytos, kad sprendimų medžių ansamblių metodas XGBoost šiam uždaviniui yra pranašesnis už rekurentinius neuroninius tinklus (LSTM). XGBoost modelis tarp visų 11 eksperimentų pasiekė vidutinę MAE paklaidą 0,0356 kW/kWp, tuo tarpu LSTM – 0,0417 kW/kWp. Be to, XGBoost pasižymėjo 8 kartus greitesniu hiperparametrų optimizavimo procesu, greitesniu mokymu ir didesniu stabilumu prognozuojant rezultatus geografiškai nutolusiose vietovėse (pvz., Klaipėdoje), kur LSTM modelis fiksavo didesnius nuokrypius.

Darbe taip pat įrodyta, kad teorinės plokštumos apšvitos (POA_{opt}) įtraukimas yra kritiškai svarbus modelio tikslumui. Kintamųjų svarbos analizė XGBoost modelyje parodė, kad POA_{opt} yra informatyviausias rodiklis (svarba viršija artimiausią meteorologinį kintamąjį 3,5 karto). Modeliai su šiuo kintamuoju vidutiniškai demonstravo 10–12 % mažesnę paklaidą nei modeliai, naudojantys tik standartinius orų prognozių duomenis (GHI, DNI, orų sąlygygos). Kai šio kintamojo nebuvo, modelis labiau rėmėsi bendrais parametrais kaip vietovės koordinatės, valandos saulės apšvitos dalis, laikas iki vidurdienio.

Tyrimas taip pat atskleidė galimybes taikyti modelį neturint tos elektrinės istorinių duomenų. Nors lokalus modelis, apmokytas su konkrečios elektrinės 2 metų istorija, yra tikslesnis (Vilka viškio atvejo analizėje vidutinė MAE 0,0217 (lokalus) prieš 0,0297 (bendras)), sukurtas bendras modelis demonstruoja pakankamą tikslumą ($MAE < 0,05$) praktiniam taikymui. Tai leidžia prognozuoti generaciją naujose elektrinėse iškart po jų įrengimo, nekaupiant istorinių duomenų. Taip pat tai atveria galimybę įvertinti kiek elektros būtų sugeneruota dar nepastatytose elektrinėse, kur žinomi tik istoriniai orų duomenys ir elektrinių meta duomenys (simuliacijos galimybės).

Tačiau darbe nustatyti duomenų kokybės ribotumai kurie suponuoja, kad didžiausios prognozavimo paklaidos kyla ne dėl modelių architektūros, o dėl įvesties duomenų neapibrėžtumo. Naudojant palydovinius reanalizės duomenis vietoje vietinių piranometrų ar šalia esančių meteorologinių stočių, modeliai negali tiksliai įvertinti lokalaus debesuotumo, kas lemia momentinius netikslumus.

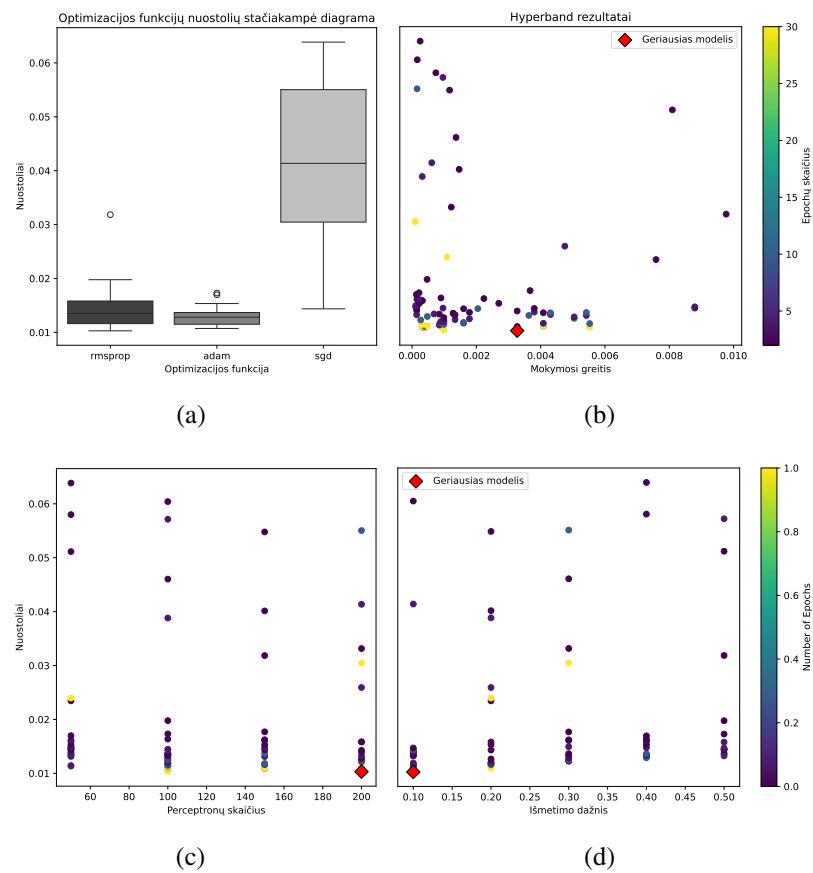
Ateities tyrimuose rekomenduojama integruoti palydovines debesų nuotraukas realiu laiku, siekiant sumažinti momentines paklaidas debesuotomis dienomis. Taip pat, siekiant dar didesnio tikslumo, siūloma tirti hibridines architektūras (pvz., CNN-LSTM), kurios galėtų apdoroti erdvinę informaciją iš kelių artimiausių orų stotelių. Tačiau neverta atmesti ir mašininio mokymo algoritmų, kurie vis dar gali būti pranašesni, ypač su mažesniais duomenų kiekiais ar skaičiavimo resursais. Ateityje, įtraukiant daugiau elektrinių, verta pagalvoti ir apie klasterizavimo uždavinius, kurie padėtų atskirti elektrines į vienašlaites, dvišlaites ir kitas, taip modeliams padedant dar geriau suprasti skirtingas sąlygas.

Literatūros šaltiniai

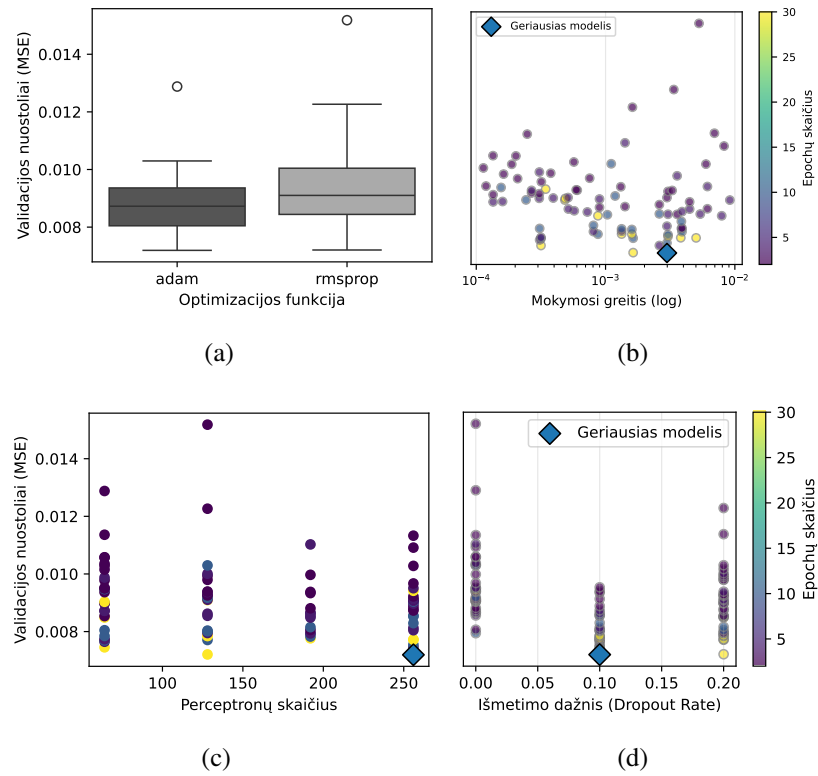
- [1] Razin Ahmed, Victor Sreeram, Y Mishra, and MD Arif. A review and evaluation of the state-of-the-art in pv solar power forecasting: Techniques and optimization. *Renewable and Sustainable Energy Reviews*, 124:109792, 2020.
- [2] Sameer Al-Dahidi, Osama Ayadi, Mohammed Alrbai, and Jihad Adeeb. Ensemble approach of optimized artificial neural networks for solar photovoltaic power prediction. *IEEE Access*, 7:81741–81758, 2019.
- [3] Jose Manuel Barrera, Alejandro Reina, Alejandro Maté, and Juan Carlos Trujillo. Solar energy prediction model based on artificial neural networks and open data. *Sustainability*, 12(17):6915, 2020.
- [4] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.
- [5] Filippo Maria Bianchi, Enrico Maiorino, Michael C Kampffmeyer, Antonello Rizzi, and Robert Jenssen. Recurrent neural network architectures. In *Recurrent Neural Networks for Short-Term Load Forecasting: An Overview and Comparative Analysis*, pages 23–29. Springer, 2017.
- [6] Alexandre Boilley and Lucien Wald. Comparison between meteorological re-analyses from era-interim and merra and measurements of daily solar irradiation at surface. *Renewable Energy*, 75:135–143, 2015.
- [7] Luis Ramirez Camargo and Johannes Schmidt. Simulation of multi-annual time series of solar photovoltaic power: Is the era5-land reanalysis the next big step? *Sustainable Energy Technologies and Assessments*, 42:100829, 2020.
- [8] Sergio Cantillo-Luna, Ricardo Moreno-Chuquen, David Celeita, and George Anders. Deep and machine learning models to forecast photovoltaic power generation. *Energies*, 16(10):4097, 2023.
- [9] Tianqi Chen. Xgboost: A scalable tree boosting system. *Cornell University*, 2016.
- [10] Rahul Gupta, Anil Kumar Yadav, SK Jha, and Pawan Kumar Pathak. Time series forecasting of solar power generation using facebook prophet and xg boost. In *2022 IEEE Delhi section conference (DELCON)*, pages 1–5. IEEE, 2022.
- [11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [12] William F Holmgren, Clifford W Hansen, and Mark A Mikofski. pvlib python: A python package for modeling solar energy systems. *Journal of Open Source Software*, 3(29):884, 2018.
- [13] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018.

- [14] Lietuvos Respublikos energetikos ministerija. Atsinaujinantys energijos ištekliai: Sektoriaus strategija, 2024-06-19. Paskutinė prieiga: 2024-12-01.
- [15] Bin Meng, Roel CGM Loonen, and Jan LM Hensen. Data-driven inference of unknown tilt and azimuth of distributed pv systems. *Solar Energy*, 211:418–432, 2020.
- [16] NVIDIA Corporation. *Deep Learning Performance Guide: Fully Connected Layers*, 2023. Skyrius: Quick Start Checklist. Prieiga per internetą: 2025 12 07.
- [17] Arif Ozbek, Alper Yildirim, and Mehmet Bilgili. Deep learning approach for one-hour ahead forecasting of energy production in a solar-pv plant. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, 44(4):10465–10480, 2022.
- [18] Aistis Raudys and Julius Gaidukevičius. Forecasting solar energy generation and household energy usage for efficient utilisation. *Energies*, 17(5):1256, 2024.
- [19] Joshua D Rhodes, Charles R Upshaw, Wesley J Cole, Chris L Holcomb, and Michael E Webber. A multi-objective assessment of the effect of solar pv array orientation and tilt on energy production and system economics. *Solar Energy*, 108:28–40, 2014.
- [20] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 2002.
- [21] Ariana Yunita, MHD Iqbal Pratama, Muhammad Zaki Almuzakki, Hani Ramadhan, Emelia Akashah P Akhir, Andi Besse Firdausiah Mansur, and Ahmad Hoirul Basori. Performance analysis of neural network architectures for time series forecasting: A comparative study of rnn, lstm, gru, and hybrid models. *MethodsX*, 15:103462, 2025.
- [22] Haoran Zhang, Meng Yuan, Yongtu Liang, Bohong Wang, Wan Zhang, and Jianqin Zheng. A risk assessment based optimization method for route selection of hazardous liquid railway network. *Safety science*, 110:217–229, 2018.
- [23] Yixuan Zhang, Ningbo Cui, Yu Feng, Daozhi Gong, and Xiaotao Hu. Comparison of bp, pso-bp and statistical models for predicting daily global solar radiation in arid northwest china. *Computers and Electronics in Agriculture*, 164:104905, 2019.
- [24] Jianqin Zheng, Haoran Zhang, Yuanhao Dai, Bohong Wang, Taicheng Zheng, Qi Liao, Yongtu Liang, Fengwei Zhang, and Xuan Song. Time series prediction for output of multi-region solar power plants. *Applied Energy*, 257:114001, 2020.
- [25] Jianqin Zheng, Haoran Zhang, Long Yin, Yongtu Liang, Bohong Wang, Zhengbing Li, Xuan Song, and Yu Zhang. A voyage with minimal fuel consumption for cruise ships. *Journal of Cleaner Production*, 215:144–153, 2019.

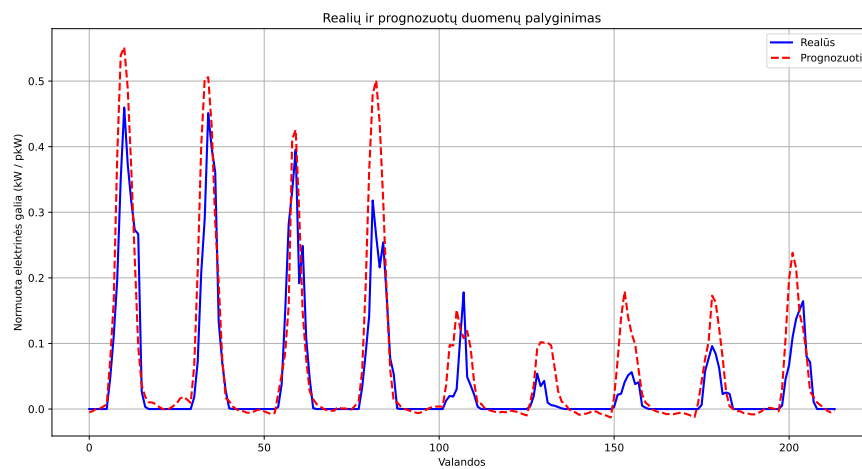
7. Priedai



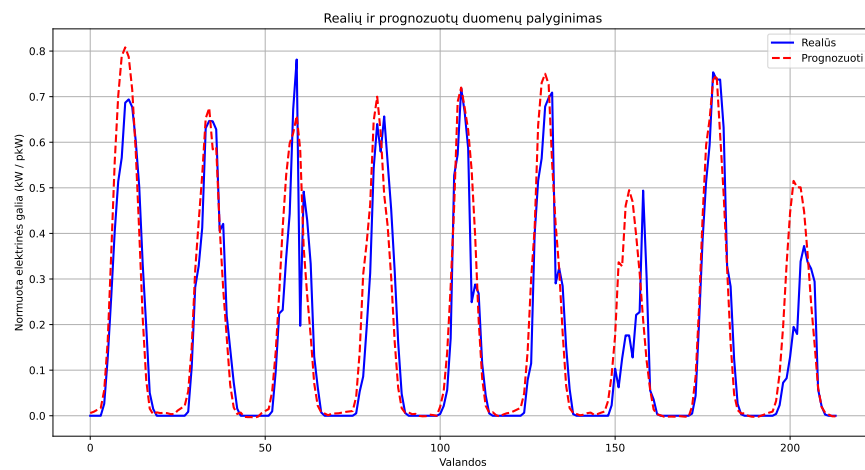
16 pav. Hyperband algoritmo nuostolių funkcijų diagramos mažesnės apimties darbe pagal optimizacijos funkciją (a), mokymosi greitį (b), perceptronų skaičių (c) ir išmetimo dažnį (d)



17 pav. Hyperband algoritmo nuostolių funkcijų diagramos be teorinės plokštumos apšvitos pagal optimizacijos funkciją (a), mokymosi greitį (b), perceptronų skaičių (c) ir išmetimo dažnį (d)



18 pav. Realūs ir prognozuoti rudens duomenys pirminiame modelyje



19 pav. Realūs ir prognozuoti vasaros duomenys pirminiame modelyje