Vilniaus universiteto
Fizikos fakulteto
Taikomosios elektrodinamikos ir telekomunikacijų institutas

Deividas Garuolis

LTE RYŠIO TINKLO PRALAIDUMO MODELIAVIMAS NAUDOJANT MAŠININIO MOKYMOSI METODUS

Magistrantūros studijų baigiamasis darbas

Elektronikos ir telekomunikacijų technologijų
studijų programa

Studentas	Deividas Garuolis
Leista ginti	2021-05-24
Darbo vadovas	doc. Rimvydas Aleksiejūnas
Recenzentas	doc. Vytautas Jonkus
Instituto direktorius	prof. Jonas Matukas

Vilnius 2021

Turinys

Sutrumpinimai

ARIMA – autoregresinis integruotas slenkamojo vidurkio modelis (*angl. AutoRegressive Integrated Moving Average*).

CQI - kanalo kokybės indikatorius (*angl. Channel quality indicator*).

EPC – paketinių duomenų sistema (*angl. Evolved Packet Core*).

E-UTRAN – išvystytas universalus antžeminės prieigos tinklas (*angl. Evolved Universal Terrestrial Radio Access Network*).

HSS – laikomi autentifikacijos duomenys (*angl. Home Subscriber Server*).

IP - interneto protokolas (*angl. Internet protocol*).

LSTM – ilgalaikės bei trumpalaikės atminties neuroninis tinklas (*angl. Longshort-term-memory*).

LTE – ketvirtos kartos ryšio technologija (*angl. Long Term Evolution*).

MAE – vidutinė absoliutinė paklaida (*angl. Mean Absolute Error*).

MME – mobilumo valdymo vienetas (*angl. Mobility Management Entity*).

MSE – vidutinė kvadratinė paklaida (*angl. Mean Squared Error*).

OFDMA – ortogonalus dažnio dalinimas daugialypei prieigai (*angl. Orthogonal Frequency Division Multiple Access*) .

PCRF – užtikrina ir nustato politikos taisykles tinkle (*angl. Policy and Charging Rules Function*).

PDN – paketinių duomenų tinklas (*angl. Packet Data Network*).

PDN-GW – paketinių duomenų tinklo sąsaja (*angl. Packet Data Network Gateway*).

P-GW – LTE paketinių duomenų tinklo sąsaja (*angl. Packet Data Gateway*).

PS - paketų komutacija (*angl. Patcket Switching*).

RMSE – vidutinė kvadratinė šaknies paklaida (*angl. Root Mean Squared Error*).

RRC - sujungimų skaičius (*angl. Radio Resource Control*).

RSRP – atskaitos signalo galia (*angl. Reference Signal Received Power*).

RSRQ – atskaitos signalo kokybė (*angl. Reference Signal Received Quality*).

SC-FDMA – vieno nešlio dažnio dalinimas daugialypei prieigai (*angl.Single Carrier Frequency Division Multiple Access*).

S-GW – tarpinė tinklų sąsaja (*angl. Serving Gateway*) .

SON – išmanusis tinklas, paremtas mašininiu mokymusi (*angl. Self organizing network*).

THP - pralaidumas arba sėkmingų paketų perdavimo greitis (*angl. Throughput*)

UE – galinis vartotojas (*angl. User Equipment*).

UMTS – trečios kartos ryšio technologija (*angl. Universal Mobile Telecommunications System*).

VoIp – balsas per interneto protokolą (*angl. Voice over Internet Protocol*).

QoS – paslaugų kokybė (*angl. Quality of Service*).

Įvadas

Mobiliųjų tinklų generuojamų duomenų kiekis bei vartotojų skaičius nepailstamai didėja. Naujos kartos technologijos bei įrenginiai jau negali apsieiti be bevielio ryšio. LTE ryšio technologijos karta buvo sukurta, kad palaikytų didelius kiekius duomenų ir išlaikytų didelės spartos komunikaciją. Išaugęs bazinių stočių skaičius, padidėjęs dėmesys tinklo kokybės rodiklių optimizacijai, galinio vartotojo patirties įvertinimas – visa tai apsunkino diegimą bei eksploataciją bevielio ryšio technologijų. Pastaruoju metu buvo pradėtas plačiai naudoti mašininis mokymasis kartu su didelės apimties duomenimis (*angl. big data*), kad supaprastintume tinklo kokybės rodiklių analizę. Tinklo analizė ir prognozavimas tapo viena iš perspektyvių sričių siekiant išvengti tinklo perkrovos paskirstant išteklius, atsižvelgiant į prognozuojamą srautą [1]. Įvairūs metodai, kaip rekurentiniai neuroniai tinklai, tiesiniai, netiesiniai mašininio mokymosi algoritmai yra naudojami, norint gauti tvarius rezultatus. Tačiau tradiciniai modeliai pasižymi lėtu prognozės apskaičiavimu bei per dideliu apmokymu, kas lemia gerą prognozės tikslumą tik tai pačiai laikinei eilutei [2]. Baigimasis darbas yra tęstinis ir praeituose darbuose buvo fokusuojamasi į tradicinių tiesinių modelių taikymą aptinkant anomalijas bei ieškant tendencijos pokyčių laikinėse duomenų srauto priklausomybėse.

Šio darbo tikslas - naudojantis sumodeliuoto LTE tinklo statistiniais duomenimis ir papildomais duomenų šaltiniais sukurti mašininio mokymosi modelius, tinkamus prognozuoti laikines duomenų srauto priklausomybes.

Baigiamojo darbo dėstymo skyriai: pirmoje dalyje supažindinama su ketvirtos kartos judriojo ryšio technologija bei veikimo principais, 2 skyriuje trumpai aptariamos didžiųjų duomenų kategorijos, pateikiami šiuo metu naudojami analitikos tipai bei jų svarba, po to mašininio mokymosi metodų apžvalga, supažindinama kaip buvo gautos laikinės eilutės modelių apmokymui, taikymai, modelio veikimo principas ir 3 skyriuje gauti rezultatai.

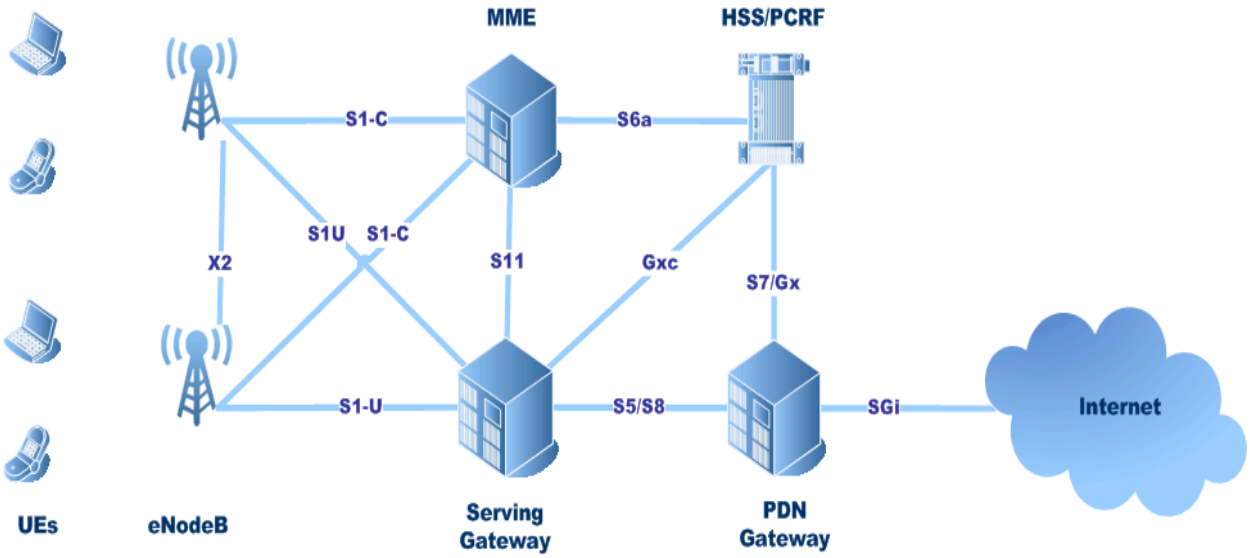
1. LTE ryšio tinklo technologija

1.1 LTE atsiradimą lėmusios priežastys

Judriojo ryšio technologijos buvo pradėtos vystyti 1946 m., kai buvo sukurta pirma telefoninė bevielio ryšio sistema. Operatorius turėjo asistuoti tiek priimant, tiek perduodant skambutį į norimą telefoną. Laikui bėgant didėjo susidomėjimas ir „Bell Labs“ išvystė judriojo ryšio technologiją, kuri gavo standartizavimą bei buvo priskirta pirmajai judriojo ryšio kartai 1G. Didėjantis vartotojų kiekis ir išaugę poreikiai operatorius pastūmėjo ieškoti efektyvesnės bei geresne kokybe pasižyminčios sistemos. Tai lėmė 2G atsiradimą, kur jau naudojamas tiek laikinis tankinimas, tiek kodinio tankinimo technologijos. Naujoji technologijos karta palaikė ir paketų perdavimą. Išaugęs interneto prieigos poreikis lėmė ir trečiosios judriojo ryšio kartos atsiradimą dar žinomai kaip UMTS. Tačiau eksponentiškai didėjantis vartotojų skaičius, patikimesnio duomenų perdavimo būdo paieškos, bent vidutinės ryšio kokybės poreikis lėmė ir ketvirtosios judriojo ryšio kartos 4G atsiradimą.

1.2 Architektūra

LTE – ketvirtos kartos judriojo ryšio tinklo architektūra paremta IP (*angl. Internet Protocol*) ir paketų komutacija PS (*angl. Packet Switching*) bei sukurta tam, kad palaikytų didelės spartos komunikaciją. Visų pirmą tam, kad būtų sumažintas delsimo laikas, kol paketai keliauja nuo IP sluoksnio iki galinio vartotojo, yra sumažinamas sistemos taškų skaičius, per kuriuos keliauja informacija. LTE tinklo architektūrą sudaro dvi pagrindinės dalys E-UTRAN(*angl. Evolved Universal Terrestrial Radio Access Network*) ir EPC (*angl. Evolved Packet Core*).



1 pav. LTE tinklo supaprastinta schema [3].

Kol vartotojas UE (*angl. User Equipment*) pasiekia ryšio paslaugų tiekėjo arba interneto lygmenį, dar iki tol yra E-UTRAN ir EPC lygmenys.

E-UTRAN lygmuo yra sudarytas iš bazinių stočių arba LTE terminologijoje dar žinomų kaip eNBs arba eNodeBs. Taip pat šį lygmenį sudaro vartotojas ir perdavimo terpė. Užduotys, kurias atlieka šis lygmuo yra panašios kaip ir nodeBs ir RNC naudojamos UMTS. Pagrindinis tikslas šio supaprastinimo, tai vėlinimo trukmės sumažinimas. Bazinės stotys vienos su kitomis komunikuoja per X2 jungtį ir perduoda informaciją apie perjungimą (*angl. Handover*), trukdžius, tinklo apkrovą. Taip pat palaikomi protokolai : PDCP , RLC, MAC, PHY.

Per S1 jungtį yra sujungiami E-UTRAN ir EPC lygmenys. EPC lygmenyje MME(*angl. Mobility Management Entity*) yra atsakingas už saugumo, autentifikavimo, duomenų ir jungčių valdymą. P-GW (*angl. Packet Data Gateway*) atsakingas už galinio vartotojo sujungimą su LTE tinklais. Iš paketinių duomenų tinklo PDN(*angl. Packet Data Network*) perduodami duomenys į PDN-GW (*angl. Packet Data Network Gateway*) vienu metu per kelis P-GW. PDN-GW pasinaudojęs duomenų baze PCRF(*angl. Policy and Charging Rules Function*) nustato galinį vartotoją bei jam siunčia paslaugų politikos atnaujinimą. Funkcijos, kurias palaiko P-GW yra VoIp (*angl. Voice over Internet Protocol*), vaizdo konferencijos arba vaizdo skambučiai. Taip pat esantis S-GW (*angl. Serving Gateway*) palaiko duomenų perdavimą su senesnėmis technologijomis bei perduoda duomenis tarp P-GW ir bazinės stoties. Duomenų bazė, kuri skirta atnaujinti informacijai apie vartotojo identifikaciją, saugumo raktus yra HSS (*angl. Home Subscriber Server*). Apibendrintai, galime teigti, jog E-UTRAN ir EPC yra atsakingi už paslaugų kokybę QoS(*angl. Quality of Service*) [3].

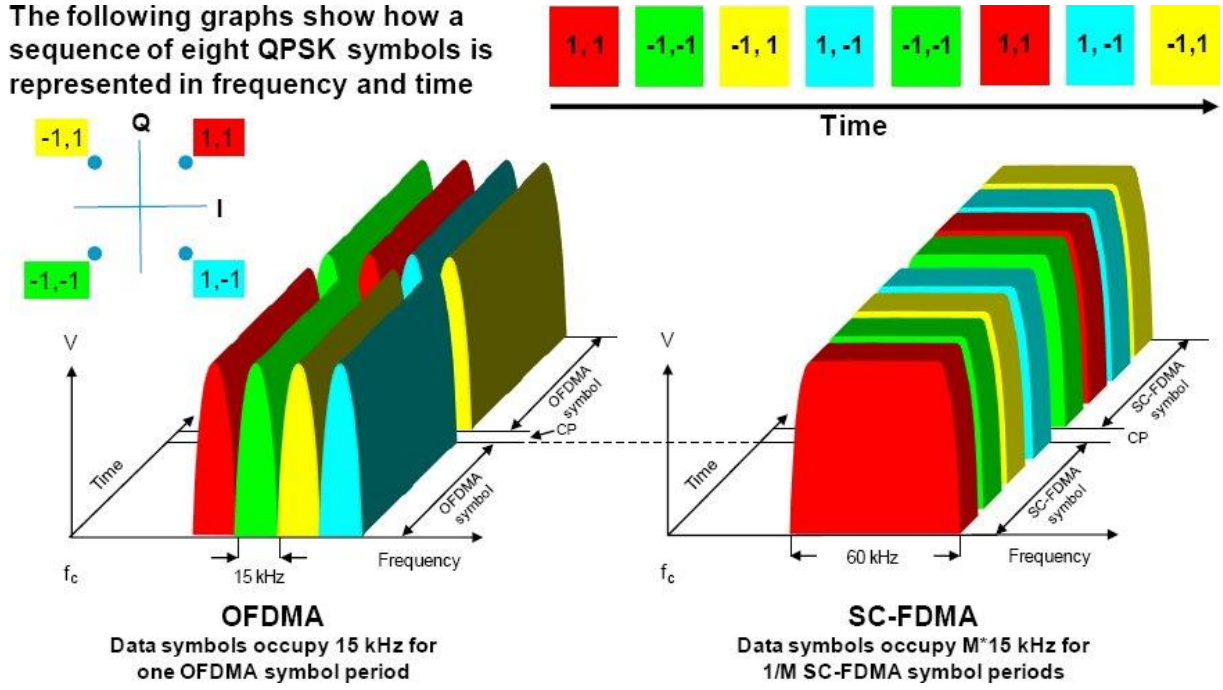
1.3 Moduliacijos

Vienas iš pagrindinių LTE privalumų, tai galimybė pasirinkti dažnių juostą, o kanalai gali būti 1,4-20 MHz pločio ir sub-nešliui yra skirtas 15kHz. Moduliacijos : OFDMA(*angl. Orthogonal Frequency-Division Multiple Access*) ir SC-FDMA (*angl. Single Carrier Frequency Division Multiple Access*) yra naudojamos skirtingiems atvejams.

OFDMA moduliacija yra naudojama žemalinkių signalų tankinimui. Dažnių resursai yra padalinami į lygiagrečius sub-nešlius. Vienas sub-nešlys gali turėti tik vieną moduliacijos simbolį. Tada skirtingi sub-nešliai yra grupuojami, kad suformuotų kanalą, kuris naudojamas kaip pagrindinis, duomenų perdavimui. Ankstesnėje kartoje buvo naudojamas WCDMA, tačiau OFDMA veikimas pasižymi aukštesniu spektriniu naudingumu t.y. kiek duomenų gali būti perduodama duotam juostos pločiui bei papildomomis galimybėmis : dažnių srities planavimas, MIMO (*angl. Multiple input*

multiple output) ir trukdžių koordinacija (*angl. Interference coordination*). 2 pav. matome, jog duomenų simboliai iš skirtingų vartotojų yra priskirti skirtingiems sub-nešliams, priklausomai nuo dažnių juostos skirtos konkrečiam vartotojui, ir visa tai yra daroma dažnių srityje. Turimai informacijai yra atliekama atvirkštinė FFT (*angl. Fast Fourier Transformation*) ir iš dažnių srities paverčiama į laiko sritį. Po to pridedamas ciklinis priešdėlis CP (*angl. Cyclic prefix*) ir signalas jau yra paruoštas perdavimui.

SC-FDMA yra naudojama aukštalinkių signalų tankinimui. Šio tankinimo savybės yra labai panašios kaip ir OFDMA(pvz. ortogonalumas, dažnių srities planavimas). Tačiau SC-FDMA yra naudojamas aukštalinkiams signalams, nes keliami mažesni reikalavimai signalo galios stiprintuvui ir efektyviau išnaudojama dažnių juosta .Taigi, rezultate vidutinė perdavimo galia gali būti didesnė naudojant SC-FMA nei OFDMA. Taip yra todėl, nes vartotojai yra ortogonalūs ir jie nėra trukdžiai vienas kitam dažnių srityje. Galinio vartotoju atveju, signalo siuntimas vyksta taip: CP yra pašalinamas ir tada laiko srityje yra atliekama FFT, kad simboliai prie kiekvieno sub-nešlio galėtų būti išrenkami.



2 pav. OFDMA ir SC-FDMA moduliacijų palyginimas [3].

Kadangi bazinėje stotyje yra lengviau pastatyti geresnį stiprintuvą nei kiekvienam vartotojui įdiegti, todėl ir naudojamos skirtingos moduliacijos. Taip pat galinio vartotojo įrenginiui naudojant SC-FDMA sutaupoma energija [3].

1.4 LTE tinklo kokybės rodikliai

THP (*angl. Throughput)* – pralaidumas yra sėkmingų pranešimų perdavimo greitis. Pralaidumas paprastai matuojamas bitais per sekundę, o po to atitinkamai konvertuojamas į megabitus.

CQI (*angl. Channel quality indicator*) - kanalo kokybės indikatorius yra skirtas matuoti galino vartotojo naudojamo kanalo kokybę. Tam, kad apskaičiuotume šį parametrą remiamasi SNR (*angl. Signal Noise Ratio*) ir SINR (*angl. Signal to Interference plus Noise Ratio*).

RSRP (*angl. Reference Signal Received Power*) – yra apibūdintas kaip tiesinis vidurkis galios pasiskirstymo W. Jis matuojamas imtuve, t.y. galinio vartotojo įrenginyje. Pagrindiniai matavimo vienetai dBm. Iš šio dydžio galime spręsti apie signalą tam tikrame taške. Kuo RSRP didesnis, tuo signalas stipresnis.

RSRQ (*angl. Reference Signal Received Quality*) – atskaitos signalo kokybė imtuve yra apibūdinama N-tojo PRB ir RSRP santykiu su RSSI, kur N yra PRB(*angl. Physical Resource block*s). Skaičius, kuriuose RSSI (*angl. Received signal strength indication*) yra matuojamas, paprastai lygus sistemos juostos pločiui. Matavimo vienetai dB:

$$RSRQ = N_{prb} \frac{RSRP}{(E - UTRA\ Carrier\ RSSI)} \quad (1)$$

RRC (*angl. Radio Resource Control*) – LTE ir UMTS protokolas egzistuojantis oro terpėje. Šis sluoksnis yra tarp galinio vartotojo ir bazinės stoties. Pagrindinė funkcija, kurią atlieka šis protokolas yra sujungimas, kai vartotojas pareikalauja prisijungimo. RRC patvirtina šią užklausą ir priskiriamas kanalas srautui [3].

2. Tyrimo metodika

2.1 Didžiųjų duomenų kategorijos

Didžiuosius duomenis galime suskirstyti į vidinius bei išorinius. Tinklo operatorių duomenys, kurie yra prieinami tik kompanijos tinklo analitikams yra vadinami vidiniais, o išoriniai duomenys yra anonimizuojami, sugrupuojami ir dažniausiai publikuojami tik praėjus tam tikram laikotarpiui. Vidiniai duomenys yra renkami tiek iš korinio tinklo, tiek iš RAN dalies. Koriniame tinkle yra apibūdinami profiliai, kurių pagalba kaupiama informacija apie tinklo darbo charakteristikas, skambučius, duomenų naudojimą. RAN dalyje yra informacija apie bazines stotis, konfigūracijas, trukdžius ir kitas metrikas. Taip pat pastarojoje dalyje yra kaupiama informacija apie galinius vartotojus, signalo lygį, kokybę ar triukšmą. Išoriniai duomenys gaunami iš galinio vartotojo įrenginio. Šie duomenys nurodo galinio vartotojo profilį, kas apibūdina jo elgesį, duomenų naudojimą. Pastaraisiais metais yra ypač kreipiamas dėmesys į išmaniųjų įrenginių ir jų programėlių naudojimosi rodiklius, kokybę, pasiekiamą pralaidumą. Apibendrinant, tai vadinama galinio vartotojo patirtimi. Taigi taikymo lygmuo (*angl. Application level*) tapo viena iš aktualijų tinklo analizės srityje. Kalbant apie didžiuosius duomenis, vidiniai ir išoriniai duomenys yra dar skirstomi į struktūrizuotus, kurie pasižymi tam tikra tvarką bei laukų pavadinimais t.y. stulpeliais, kad būtų pastovi ir paprasta prieiga prie jų. Duomenims esant nestruktūrizuotiems be duomenų inžinieriaus bei tinklo inžinieriaus įsikišimo jie yra sunkiai suprantami. Duomenų gavyba (*angl. Data mining*) yra vienas iš analitikos būdų, kad duomenys būtų struktūrizuojami, juos sujungiant, grupuojant ir ieškant analizės metodų [4], [5], [6], [7].

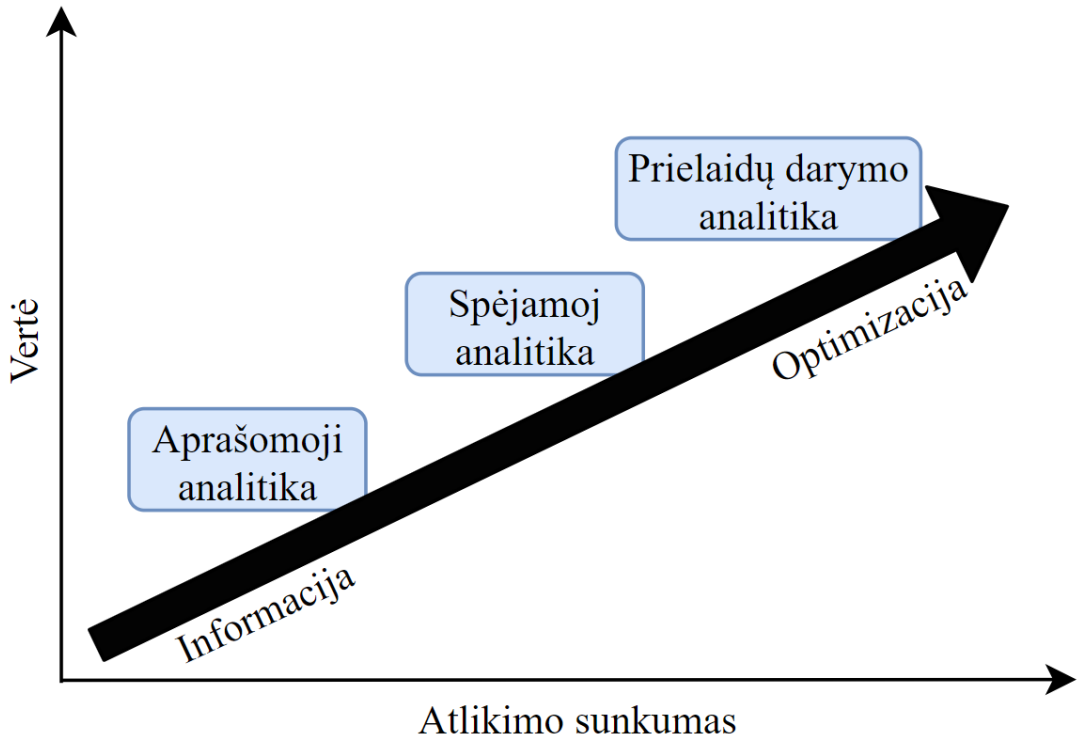
2.2 Analitikos tipai

Pastarosiomis dienomis greitai besivystančios analitikos sritys ir išsiplėtusios galimybės lėmė tinklo operatorių atkreiptą dėmesį. Tai yra įvertinama kaip perspektyvi sritis norint sumodeliuoti tinklo scenarijus, aptikti tinklo kokybės anomalijas, surasti klaidingus tinklo kokybės rodiklius bei pačius tinklo nustatymus. Didžiųjų duomenų analitika yra skirstoma į keturis tipus:

1. Aprašomoji analitika (*angl. Descriptive analytics*) – pagrindinis klausimas į kurį atsako „kas nutiko?“. Remiamasi istoriniais bei realaus laiko duomenimis ir įvertinus pokyčius galime matyti, kas vyksta šiuo metu tinkle.

2. Diagnostinė analitika (*angl. Diagnostic analytics*) – remiantis šiuo analitikos tipu lyginame turimus istorinius duomenis su papildomais duomenų šaltiniais, kurių metu galime identifikuoti anomalijas bei rasti jas sukėlusias priežastis.
3. Spėjamoji analitika (*angl. Predictive analytics*) – nusako, kas gali nutikti tinkle už tam tikro laikotarpio. Šiuo atveju yra remiamasi aprašomosios ir diagnostinės analitikos įžvalgomis, kurių metu galime sugrupuoti duomenis pagal jų pobūdį bei apdoroti ir atlikti tinklo rodiklio ateities prognozę. Būtina įvertinti tai, jog rezultatai negali būti tikslūs, tačiau esant pakankamai gerai apmokytam modeliui (*angl. Prediction model*) galime turėti rezultatus artimus realybei (pvz.: tinklo pralaidumo pasikeitimai, vartotojų skaičiaus išaugimas).
4. Prielaidų darymo analitika (*angl. Prescriptive analytics*) – šio analitikos tipo tikslas yra nurodyti žingsnius, kuriuos reikia atlikti, norint išvengti prognozuotų problemų (pvz.: renginio metu pagal žmonių skaičių ir vietovės tipą sukuriamas modelis, kuris įvertina dabartines tinklo galimybes ir nurodo koks reikalingas tinklo praplėtimas, norint išvengti tinklo kokybės rodiklių nukritimo).

Pastarosiomis dienomis yra ypač didelis dėmesys skiriamas tiek spėjimo analitikai, tiek ir prielaidų darymo analitikai. Abiem atvejais naudojamasi mašininiu mokymusi, modeliavimu, koreliacijos matricomis, duomenų gavyba, tačiau galutinis rezultatas priklauso ne tik nuo modelio, tačiau ir nuo paruoštų duomenų apmokymui. Prielaidų darymo analitika pranoksta spėjamąją analitiką, nes gali pasiūlyti tam tikrus veiksmus, kurie galėtų padėti išvengti tinklo kokybės rodiklių nukritimo. Tam, kad būtų sukurtą sistemos visuma, turi būti pakankamas kiekis duomenų, grįžtamasis ryšys, kuris įvertintų daromą įtaką. Šiuo metu yra ypač sudėtinga rasti atvirai prieinamų tinklo duomenų, dėl duomenų apsaugos įstatymų, tačiau ir jie pasižymi tuo, jog yra bent kelių metų senumo, todėl yra mažiau aktualūs. 3 pav. pateikta analitikos tipo vertė, priklausomai nuo jos įgyvendinimo sudėtingumo. Aprašomoji analitika pasižymi išsamia informacija, tačiau tik prielaidų darymo analitika gali padėti optimizuoti tinklą. Esant modelio prognozei, tinklo operatoriui yra galimybė prioretizuoti savo poreikius, pagal kuriuos parenkami sistemos apribojimai [4], [5], [6].



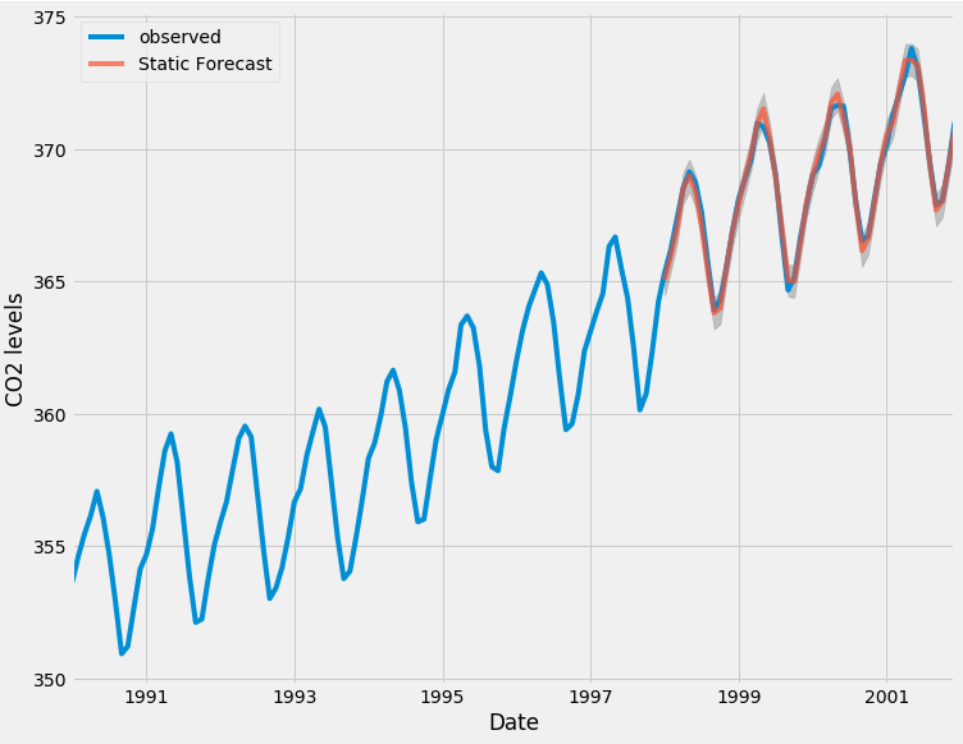
3 pav. Analitikos tipų vertės priklausomybė nuo atlikimo sunkumo.

2.3 Mašininis mokymasis

Dėl didėjančio duomenų kiekio ir sudėtingėjančių problemų naudojamas mašininis mokymasis bei neuroniniai tinklai. Kuriant universalius modelius bei prognozuojant tinklo veiklą galime ne tik optimizuoti tinklą bei numatyti būsimą apkrovą, tačiau ir aptikti anomalijas, kurios išsiskiria nuo įprastos tinklo statistikos. Taip pat artėjame prie galimybės įdiegti SON. Aptarsime kelis mašininio mokymosi standartinius metodus bei neuroninių tinklų modelį LSTM.

2.4 ARIMA modelis

ARIMA modelis, tai yra tiesinis laiko eilučių prognozavimo modelis, kuris nuspėja būsimas vertes priklausomai nuo ankstesnių verčių. ARIMA vienas iš plačiai paplitusių anomalijų aptikimo modelių [8], [9]. ARIMA modeliu galima prognozuoti tinklo srautus remiantis jau surinkta istorija. Po to yra palyginamas skirtumas tarp prognozuotos tendencijos ir realaus laiko srauto, kad būtų galima nustatyti skirtumus.



4 pav. ARIMA modelio prognozės ir realių duomenų palyginimas [10].

Naudojant ARIMA modelį tinklo veikimui apibūdinti yra daroma prielaida, kad modelio parametrai yra išmokti, lyginant dalį duomenų su prognozuotais duomenimis. Yra trys pagrindiniai parametrai, kurie gali būti parinkti: p (*angl. AutoRegressive*) parametras, žingsnių skaičius d ir slenkamasis vidurkis q. Iš esmės ARIMA modelį galime aprašyti kaip: ARIMA(p; d; q), o procesas yra apibendrinimas 2 formule.

$$z_t - \sum_{i=1}^{p} \varphi_i z_{t-i} = e_t - \sum_{j=1}^{q} \theta_j z_{t-j}\,, \tag{2}$$

kur e_t yra prognozavimo paklaida intervale t , o t, $t \in \{1, 2, \ldots, T\}$, φ ir θ yra atitinkamai p, q autoregresyvumo ir slenkančio vidurkio baigtiniai koeficientai. Pats duomenų rinkinys yra diferencijuojamas d kartų, norint gauti z_t . Tinklo segmentų sukūrimas vyksta atliekant srauto analizę iš ARIMA modelio, naudojant nustatyto laikotarpio istorinius duomenis mokymui ir kiekvienos naujos dienos pokyčius modelio kalibravimui. Pirmasis žingsnis yra įvertinti parametrą d naudojant standartinį įvertinimo metodą [8], o parametras gali būti įvertintas ir išbandytas naudojant autokoreliacijos funkciją. Parametrams p ir d įvertinti naudojama ciklinio perrinkimo (*angl. Grid Search*) technika. Informacijos kriterijus Akaike (AIC) yra naudojamas siekiant gauti pusiausvyrą tarp aukšto resursų naudojimo ir mažos nustatymo paklaidos. Ciklinio perrinkimo metodu ieškoma

visų galimų kombinacijų derinių *p* ir q , norint pasiekti minimalų AIC. Pasak [11], efektyvus metodas anomalijoms nustatyti yra užtikrintumo juosta, nurodanti intervalą, kai duomenų kitimas laikomas normaliu. Užtikrintumo juostai apskaičiuoti naudojame simetrinį metodą, kai apatinis ir viršutinis slenksčiai yra apskaičiuojami:

$$IT_t = X'_t - \left(\frac{var(X')}{mean(X')}\right), \tag{3}$$

$$ST_t = X'_t - \left(\frac{var(X')}{mean(X')}\right). \tag{4}$$

Žemesnė (IT) ir aukštesnė (ST) ribos apskaičiuojamos naudojant prognozuojamas vertes. Kintamasis *t* nurodo analizuojamą laiko intervalą, o X_t ' yra apskaičiuota prognozė tam tikram *t* . Operatoriai *var()* ir *mean()* atitinkamai grąžina prognozuojamų taškų dispersiją ir vidurkį.

2.5 fbProphet algoritmas

fbProphet – laiko eilučių prognozavimo modelis. Svarbu tai, kad šis modelis turi parametrus, kuriuos galima koreguoti. Galimas pritaikymas tiek Python, tiek R kalboje. Modelis yra skaidomas į tris pagrindines komponentes: tendencija, sezoniškumas, šventinės dienos. Lygtis aprašanti visus parametrus:

$$\mathrm{y(t)} = \mathrm{g(t)} + \mathrm{s(t)} + \mathrm{h(t)} + \varepsilon\mathrm{t}. \tag{5}$$

Šioje lygtyje *g(t)* yra tendencijos funkcija ir rodo neperiodinius pokyčius laike, *s(t)* žymi periodinius pokyčius (paros, savaitės ar metų sezoniškumas), o *h(t)* žymi šventines dienas, kurias galime pasiimti iš kalendoriaus, o εt neįprasti pasikeitimai, kurie nesusiję nei su viena iš anksčiau paminėtų reikšmių. Apibūdinimas yra panašus į regresijos modelį, kur pritaikoma netiesinė aproksimacija, tačiau šiuo atveju naudojame laiką kaip regresorių, o tiesinės ir netiesinės aproksimacijos yra kaip komponentės. Sezoniškumas yra tiesiog faktorius iš kurio yra dauginama.

Svarbiausią lygties dalį sudaro tendencijos funkcija *g(t)*, tačiau modeliui esant netiesiniam *g(t)* yra:

$$g(t) = \frac{C}{1 + \exp\left(-(k + a(t)^T\delta)(t - (m + a(t)^T\gamma)\right)}, \tag{6}$$

kur *C* yra numatoma sistemos talpa, *k* yra augimo greitis, *m* kompensacijos parametras. Augimo greitis *k* ir poslinkio parametras *m* nėra konstantos.

Į modelį įtraukti yra ir tendencijos pokyčiai, kur pokytis yra netiesinis ir gali kisti. Tarkim *S* skaičius yra pasikeitimo taškai laiko momentais s_j , kur j = 1;..; *S*. Apibūdiname kitimo greičio vektorių $\delta \in R^S$, kur δ_j yra pokyčio greitis tam tikrame taške s_j. Kitimo greitis bet kuriuo laiko momentu *t* yra sudarytas iš pagrindinio kitimo greičio *k* bei įskaičiuojami visi pokyčiai iki to taško $k + \sum_{j;t>sj} \delta_j$. Visa tai galime apibūdinti vektoriumi $a(t) \in \{0, 1\}^S$, kuris yra:

$$a_j(t) = \begin{cases} 1, if\ t \geq s_j \\ 0, \ otherwise. \end{cases}. \tag{7}$$

Pokyčio greitis laiko momentu *t* yra $k + a(t)^T\delta$. Kai pokyčio greitis yra nustatomas, parametras *m* taip pat yra keičiamas, kad sujungtų segmentų taškus. Teisingas kalibravimas yra nustatomas pokyčio taške ir apskaičiuojamas:

$$\gamma_j = \left(s_j - m - \sum_{l<j} \gamma_l\right)\left(1 - \frac{k + \sum_{l<\mathrm{j}} \delta_l}{k + \sum_{l\geq \mathrm{j}} \delta_l}\right). \tag{8}$$

Tada augimo modelis yra:

$$g(t) = \frac{C(t)}{1+\exp\left(-(k+a(t)^T\delta)(t-(m+a(t)^T\gamma))\right)}, \tag{9}$$

Svarbus parametrų derinys modelyje yra *C(t)* arba tikėtina sistemos talpa bet kuriuo laiko momentu. Taip pat gali būti naudojami papildomi duomenų šaltiniai, tokie kaip gyventojų tankis arba tikėtinas populiacijos didėjimas vietovėje.

Galimas ir automatinis pokyčio taškų parinkimas. Pokyčio taškas s_j gali būti nurodytas analitiko, naudojant žinomas datas apie tam tikrus įvykius ar šventes arba nurodant rinkinį galimų datų. Automatinis aptikimas gali būti gautas naudojant (15) bei tiesinės tendencijos formulę:

$$g(t) = (k + a(t)^T\delta)t + (m + a(t)^\gamma T). \tag{10}$$

Dažnai yra nurodomas didelis skaičius pokyčio taškų(pvz.: kartą per mėnesį, kuris kartojasi jau kelerius metus). Parametras τ valdo modelio prisitaikymą, kitaip sakant, gali keisti pokyčio greitį.

Kai modelis mokosi iš praeities sekų, tendencija turės pastovų pokyčio greitį. Neapibrėžtumas yra įvertinamas prognozuojant veiklą. Turint *T* taškų istorijoje taip pat yra ir *S* pokyčio taškų, kurių kiekviename yra ir pokyčio greitis. Būsimi pokyčiai gali būti imituojami keičiant dispersiją, o būsimi pokyčio taškai yra tokiu atsitiktiniu būdu sukuriami, kad vidutinis pokytis sutampa su buvusiais istorijoje. Po to yra matuojamas diapazono parametras $\lambda = \frac{1}{S}\sum_{j=1}^{S}\left|\delta_j\right|$. Galiausiai atlikus prognozę ir lyginant su istoriniais duomenimis galima apskaičiuoti neapibrėžtumo intervalus, darant prielaidą, kad diapazonas ir pokyčio greitis bus pastovūs.

Laiko eilutės dažnai turi kelis periodiškumus t.y. sezoniškumą. Pavyzdžiui: gali būti penkių darbo dienų periodiškumas, kuris kartojasi kiekvieną savaitę, o tarkim šventinės dienos ar mokinių

atostogos gali pasižymėti metiniu periodiškumu. Norint prognozuoti tokį poveikį, turime nurodyti sezoniškumo modelį, kuris yra periodinė funkciją *t*. Yra naudojamos Furje eilutės, prognozuojant periodiškumo poveikį. Tarkim *P* yra įprastinis periodiškumas, kurį laiko eilutės turi. Taigi, galime apytiksliai įvertinti sezoniškumo poveikį naudojant :

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right)\right). \quad (11)$$

Sezoniškumui reikalinga sudaryti vektorių matricą, kiekvienai *t* vertei praeities ir būsimiems duomenims. Automatiniam parametrų optimizavimui gali būti naudojama atrankos procedūra AIC.

Šventinės dienos arba tarptautinės šventės dažnai sukelia nenuspėjamą pokytį laiko eilutėse ir dažnai kinta ne pagal periodišką tendenciją, todėl jų sukelti padariniai negali būti tiksliai sumodeliuoti. Dažniausiai tokio įvykio poveikis gali būti nuspėtas, tik tuo atveju, jeigu turime praeitų metų pavyzdžių. Taigi, galime nurodyti sąrašą su šventinėmis dienomis, šalimis, kad algoritmas būtų universalesnis ir tiktų ne tik vienai šaliai. Pritaikant šventinių dienų modelį, daroma prielaida, kad šventinių dienų poveikis laiko eilutėms yra nepriklausomas. Prognozės atliekamos panašiai kaip ir sezoniškumo atveju, naudojant regresorių matricą:

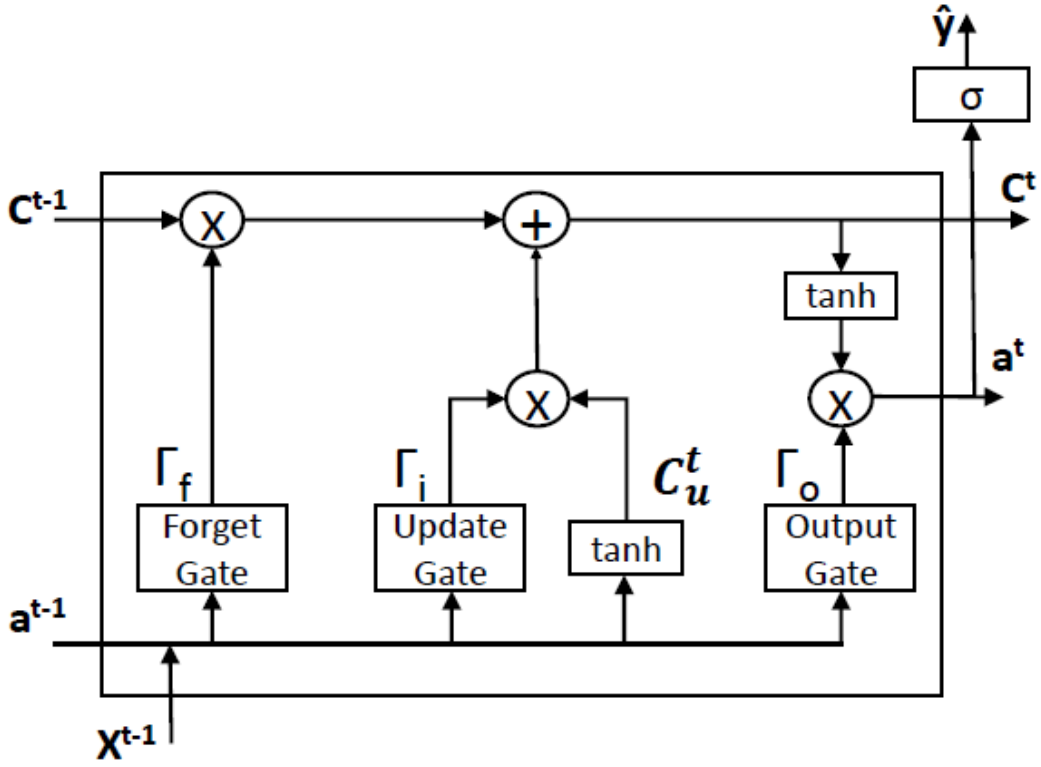
$$Z(t) = [1(t \in D_1), \dots, 1(t \in D_L)]. \quad (12)$$

Taip pat yra naudinga atkreipti dėmesį į dienas prieš ir po šventinės dienos, kadangi dėl žmonių judėjimo, laiko eilutės gali būti artimos šventinėms dienoms [12].

2.6 LSTM modelis

LSTM - ilgalaikės bei trumpalaikės atminties neuroninis tinklas, kuris naudoja grįžtamąjį ryšį skaičiuojant nuostolius. Kuriant tokius neuroninius tinklus galime juos panaudoti sudėtingoms laikinių eilučių problemoms spręsti. Taikant LSTM vietoje neuronų yra naudojami atminties blokai, kurie yra sujungiami per skirtingus sluoksnius. Blokas naudoja užtūras, kurios valdo būsenas, išvesties rezultatus, o tuo pačiu ir modelio svorio koeficientus. Priklausomai nuo įvedamos sekos į bloką, kiekviena užtūra naudoja aktyvacijos funkciją, kuri neuroniniame tinkle yra skirta transformuoti įvesties duomenis. Pagal šios funkcijos rezultatą sprendžiama ar neuronas bus aktyvuotas. Išviso yra trys tipai užtūrų, kurios veikia priklausomai nuo sąlygų:

1. FG (*angl. forget gate*) Γ_f : nusprendžia, kokią informaciją pamiršti bloke.
2. UG (*angl. Update Gate*) Γ_i : nusprendžia, kurios įvesties reikšmės atnaujins atminties bloką.
3. OG (*angl. Output Gate*) Γ_o : nusprendžia išvesties informaciją, remiantis įvestimi bei bloko atmintimi.



5 pav. Vieno atminties bloko LSTM struktūra. [13]

$$\Gamma_f^t = \sigma(W_f[a^{t-1}, X^t] + b_f\,, \tag{13}$$

$$\Gamma_i^t = \sigma(W_i[a^{t-1}, X^t] + b_i, \tag{14}$$

$$C_u^t = \varphi(W_c[a^{t-1}, X^t] + b_c\,, \tag{15}$$

$$\Gamma_o^t = \sigma(W_o[a^{t-1}, X^t] + b_o\,, \tag{16}$$

$$C^t = \Gamma_f^t * C^{t-1} + \Gamma * C_u^t\,, \tag{17}$$

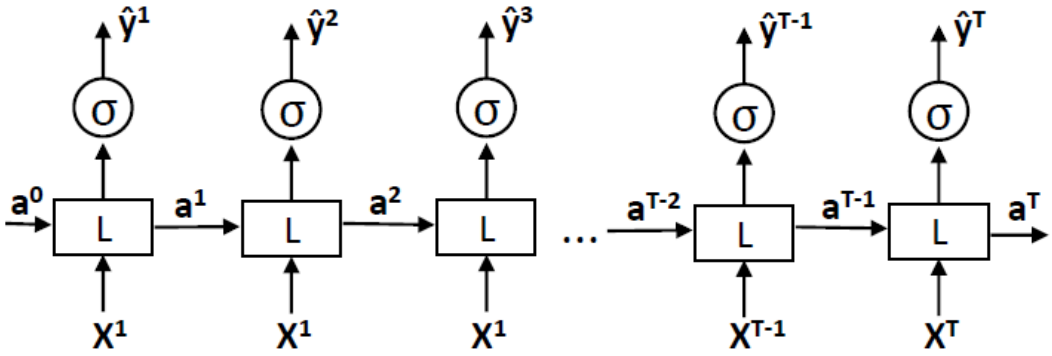
$$a^t = \Gamma_o^t * \varphi C^t\,, \tag{18}$$

$$y^t = \sigma\left(W_y a^t + b_y\right). \tag{19}$$

Būsena C^t, kaip 5 pav. pavaizduota, yra atnaujinama informacijos, kuri patenka pro UG. Esama vertė yra atnaujinama arba ne, priklausomai nuo informacijos, kuri yra patalpinta prieš tai esančioje būsenoje ir įvestyje. Tada UG nusprendžia ar informacija pateks. Galiausiai OG leidžia informacijai patekti iš dabartines būsenos. Šiuo atveju FG leidžia atminties blokui pasilikti arba pamiršti informacija, kurią gavo iš praeitos įvesties. Prognozavimas atliekamas, kai $\hat{y}$ praeina pro aktyvacijos funkciją. Dažniausiai naudojama sigmoidinė aktyvacijos funkcija, kuri lygtyse yra aprašoma kaip σ. Formali sigmoidinės funkcijos išraiška yra:

$$\sigma(x) = (1 + e^{-x})^{-1}. \tag{20}$$

Aktyvacijos funkcija taip pat gali būti ir φ, kaip nurodyta 18 formulėje. Lygtyse daugybos simbolis reiškia kiekvieno elemento daugybą paeiliui, o *W* bei *b* yra vektorių svoriai. Po to LSTM blokai yra sujungiami, kur jie sudaro, kaip 6 pav. pateikta, LSTM tinklą. Kiekvienas blokas skaičiuoja prognozę vienam laiko žingsniui ir perduoda tą informaciją kitam blokui.



6 pav. LSTM tinklas. [13]

Pagrindiniai LSTM modelio parametrai:

1. Imties dydis (*angl. batch size*).
2. Apmokymo epochų kiekis.
3. Neuronų skaičius.
4. Sluoksnių skaičius.
5. Aktyvacijos funkcija.
6. Optimizavimo algoritmas.

Imties dydis nurodo, kiek į įvestį patenka taškų iš duomenų rinkinio. Epochų kiekis lemia, kiek kartų visa laikinė eilutė bus pakartota per tuos pačius sluoksnius, kas šiuo atveju gali lemti per didelį apmokymą (*angl. overfitting*) arba per mažą apmokymą. Neuronų skaičius yra blokų skaičius, per kuriuos apmokome imtį [13], [14], [15], [16].

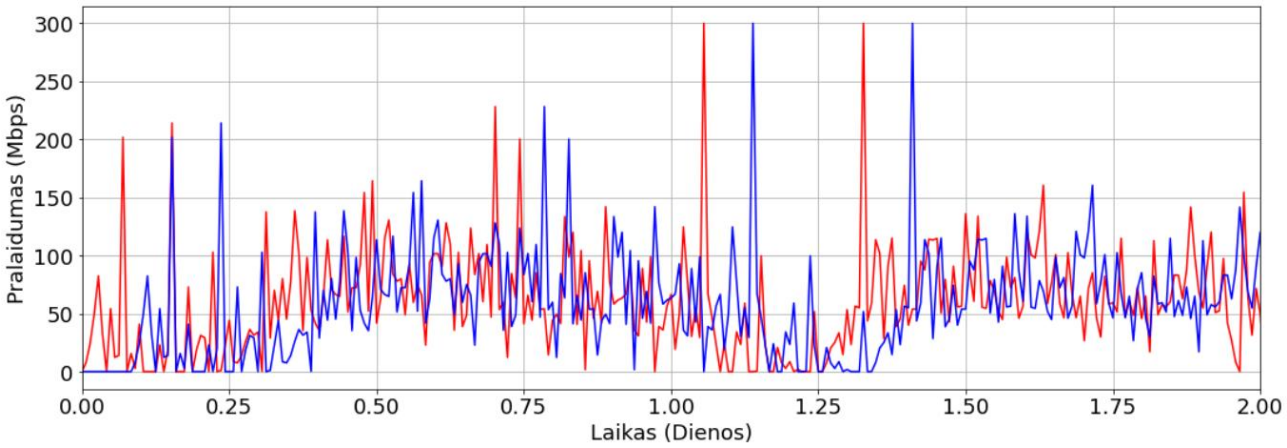
Prognozės tikslumui bei nuostoliams nustatyti naudojamos metrikos. Pagal nutylėjimą naudojama ir labiausiai paplitusi yra vidutinės kvadratinės paklaidos metrika:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(|x_i - y_i|)^2 . \quad (21)$$

Vidutinė kvadratinė paklaida yra apskaičiuojama kaip prognozuotų ir tikrųjų verčių kvadratų skirtumų vidurkis. Rezultatas yra visada teigiamas, neatsižvelgiant į verčių ženklą. Šiuo atveju kvadratas reiškia, kad esant didesnei paklaidai jos svoris yra didesnis nei esant mažai paklaidai, todėl modelio kūrimui tai yra labai svarbu. Vidutinė kvadratinė paklaida atsižvelgia tik į paklaidą esančią ordinatės ašyje, tačiau neatsižvelgia į tai ar seka yra paslinkta ar ne. Tuo atveju, jeigu seka yra

paslinkta, vidutinė kvadratinė paklaida išauga, o ypač dažnai tai pasitaiko esant periodiniams duomenims.

[17] buvo apsvarstyta įvesti kryžminės koreliacijos paklaidos metriką. Įvedus šią metriką modelis turėtų ne tik įvertinti vidutinę kvadratinę paklaidą tarp laiko eilučių, tačiau ir kokia yra daroma poslinkio įtaka. Turime dvi laikines eilutes: tikrąją x_n ir prognozuotą y_n. Sakykime prognozuota eilutė, kaip pateikta 7 pav., yra pasislinkusi per *n*.

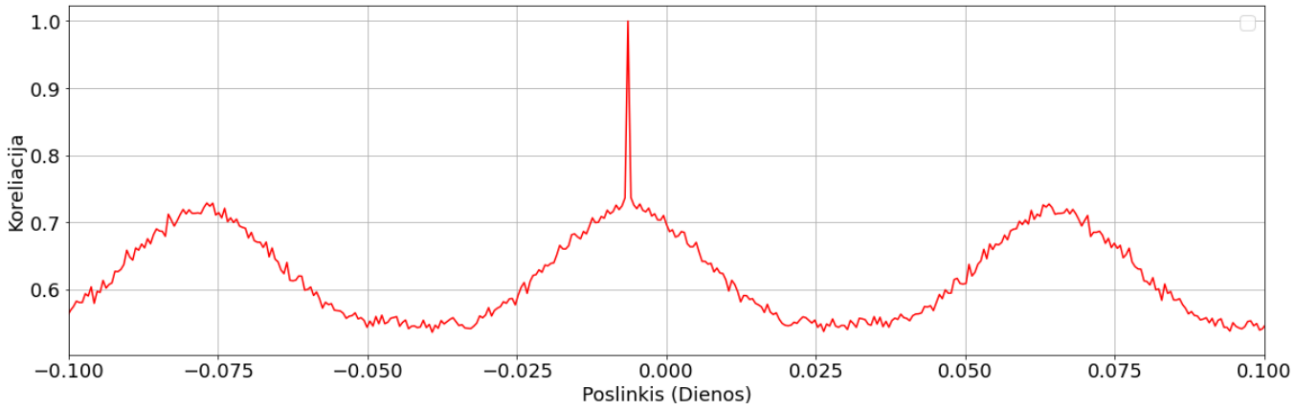


7 pav. Dvi identiškos eilutės, kurios yra paslinktos viena kitos atžvilgiu.

Slinkime prognozuotą laikinę eilutę nuo *-n* iki *+n*. Slinkdami laikinę eilutę ir skaičiuodami koreliacijos koeficientą kiekviename poslinkio taške gausime poslinkio priklausomybę nuo koreliacijos koeficiento. Radus didžiausią koreliacijos koeficiento vertę, galėsime apskaičiuoti poslinkio vertę. Po to galime įvertinti kryžminės koreliacijos paklaidą:

$$e_{corr} = 1 - corr_{max} + k * \Delta n, \tag{22}$$

kur $k = 2\ h^{-1}$ ir yra eksperimentinė vertė, o $\Delta n = \text{argmax(corr)}$



8 pav. Koreliacijos koeficiento priklausomybė nuo poslinkio.

Taip pat viena iš metrikų yra vidutinė absoliutinė paklaida:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - y_i| \quad . \tag{23}$$

Vidutinė kvadratinės šaknies paklaida:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(|x_i - y_i|)^2}\,. \tag{24}$$

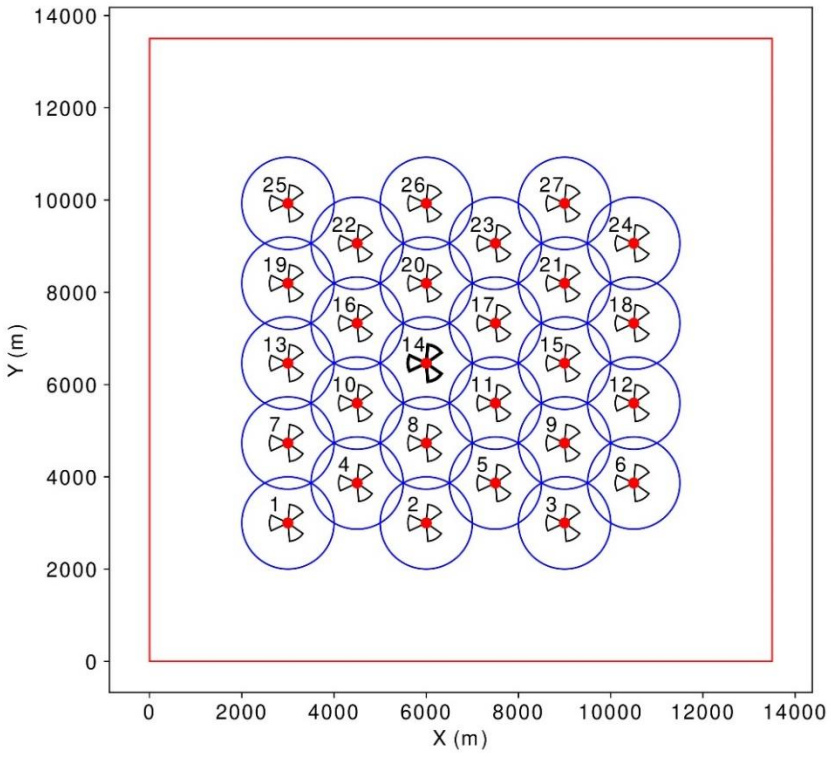
Norint objektyviai įvertinti metrikas ir lyginti jas su publikuotų straipsnių rezultatais, įvedami normuoti dydžiai. Metrika yra padalinama iš laikinės eilutės standartinio nuokrypio ir gaunama normuota metrika.

2.7 Duomenų pralaidumo laikinės eilutės

Šiomis dienomis atsiradus papildomiems duomenų apsaugos įstatymams yra sunku rasti laisvai prieinamų 4G tinklo duomenų apie pralaidumą ar talpą. Taigi, reikalinga susigeneruoti duomenis patiems, kadangi yra gerai apibrėžtų modelių apskaičiuoti talpai bei pralaidumui. Darome prielaidą, jog LTE versija yra 12 arba vėlesnė, kurioje maksimali moduliacija QAM256, o jos maksimalus spektrinis naudingumas C_{15} = 7.4063 bits/Hz. Iš to galime apskaičiuoti maksimalų pralaidumą esant MIMO 2x2 antenoms. Tačiau realiu atveju dažniausia moduliacija pasiekiama QAM64, CQI = 9, o pralaidumas yra apskaičiuojamas :

$$\mathrm{C_{max}} = \Delta \mathrm{f N_{RB} M_x C}(1-\eta), \tag{25}$$

taigi spektrinis efektyvumas C_8 = 3.3223 bits/Hz , o pralaidumas C = 89.7 Mbps.

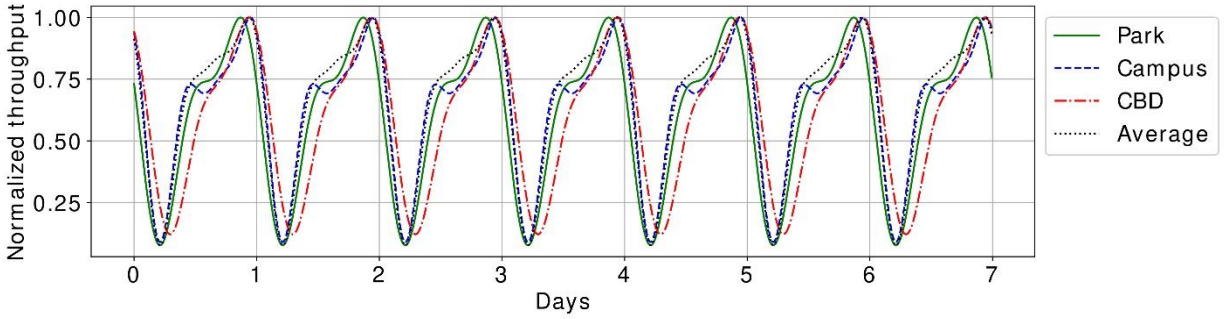


9 pav. LTE tinklo bazinių stočių išdėstymas [18].

Sektoriaus matmenys buvo įvertinti remiantis signalo galia RSRP:

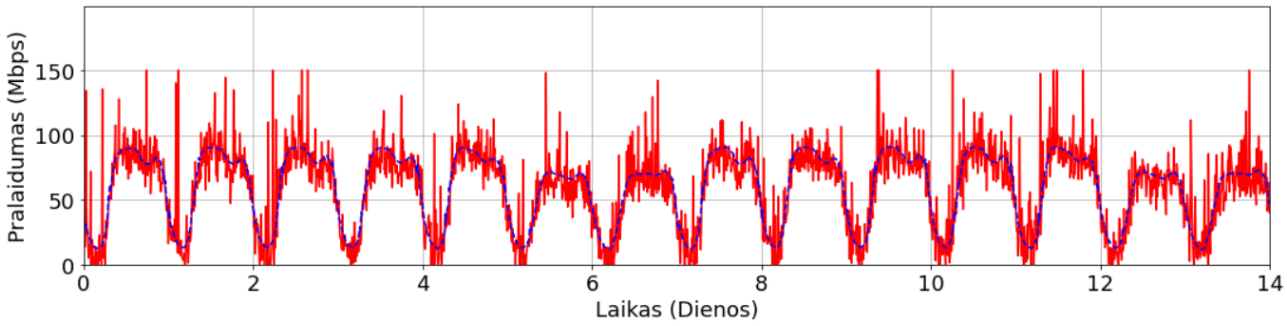
$$P_{RSRP}(d,\theta,\varphi) = P_{Tx} + G_{BS}(\theta,\varphi) - L_{Hata}(d) - 10\log_{10}(12N_{RB}), \quad (26)$$

kur P_{Tx} yra visa išėjimo galia, $G_{BS}(\theta,\varphi)$ spinduliavimo diagrama, o $L_{Hata}(d)$ yra perdavimo nuostoliai, kurie apskaičiuoti naudojantis Ericsson's Hata 9999 lygtimi ir N_{RB} yra resursų blokų skaičius. Taigi, norint susikurti pralaidumo rezultatus, reikalinga panaudoti jau publikuotas laikines charakteristikas. Daugelis paskelbtų laiko eilučių kitimų pasižymi dienos ir savaitės tendencijomis.



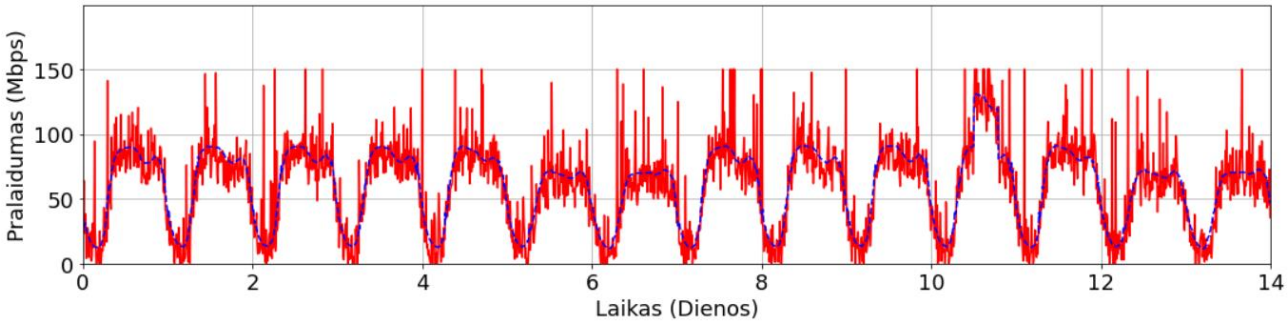
10 pav. Savaitės pralaidumo laikinis kitimas, esant skirtingoms vietovėms : parkas, miestelis, verslo rajonas, bendras vaizdas [18].

Dažniausiai laiko eilutės yra pateikiamos kaip pralaidumas (Mbps), pralaidumo tankis (Mbps/km^2) arba normalizuotos vertės. Laiko eilučių kitimo žingsnis yra 10 minučių. Visos laiko eilutės vaizduoja vidutinę variaciją, o norint duomenis padaryti artimesnius realiems, reikalinga sumodeliuoti triukšmą bei pridėti logaritminį kitimą, kuris buvo pasiūlytas [18]. Šių kitimų amplitudė apibūdinama standartiniu nuokrypiu σ_T. Laikiniai kitimai, panaudojus logaritminį triukšmo pasiskirstymą, pavaizduoti 11 pav.



11 pav. Dviejų savaičių pralaidumo laikinis kitimas esant vidutinei agreguotai vertei bei su logaritminiu triukšmu, kurio standartinis nuokrypis σ_T= 10 Mbps.

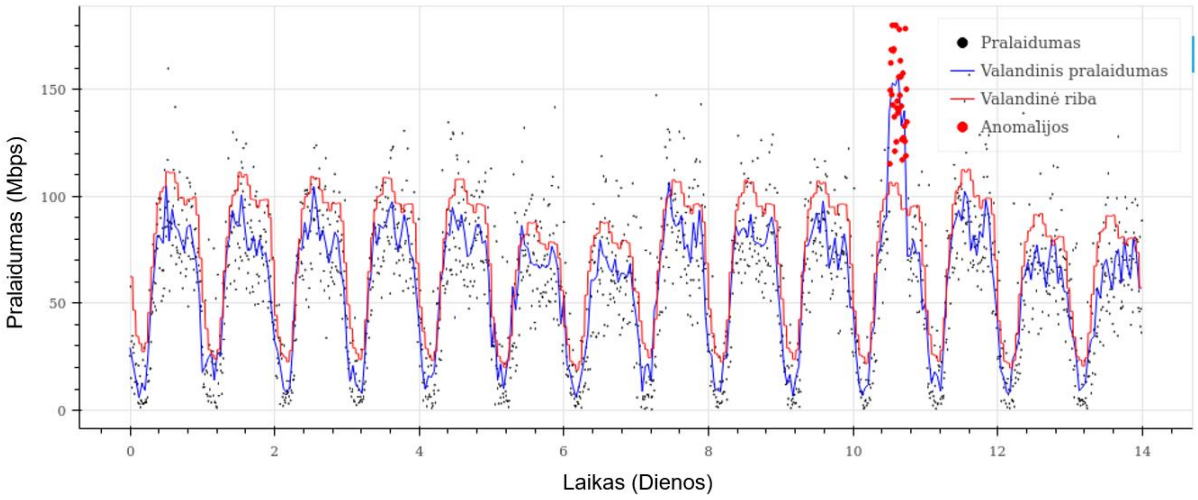
Taip pat galime sumodeliuoti tinklo laikinės anomalijas, remiantis tinklo persiskirstymu, kai viena iš bazinių stočių išsijungia. Išjungimo metu laikinėje eilutėje atsiranda anomalija spyglio formos, kuri tęsiasi tiek, kiek laiko buvo išjungta bazinė stotis. Tinklo vartotojai persiskirsto į kaimynines bazines stotis, pagal geriausią signalo lygį.



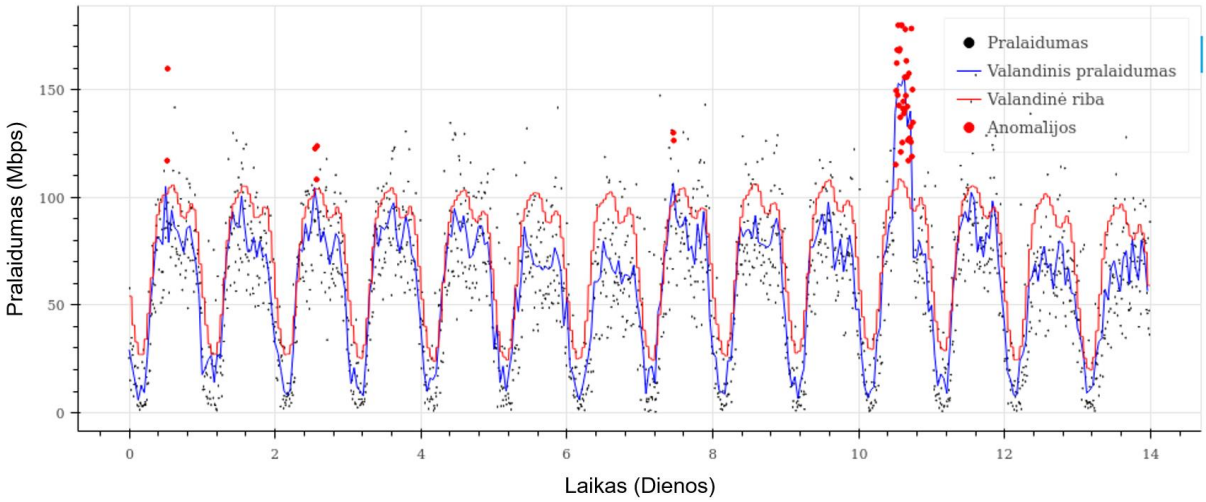
12 pav. Dviejų savaičių pralaidumo laikinis kitimas ir anomalija vienuoliktoje dienoje.

2.8 Regresinių modelių taikymai

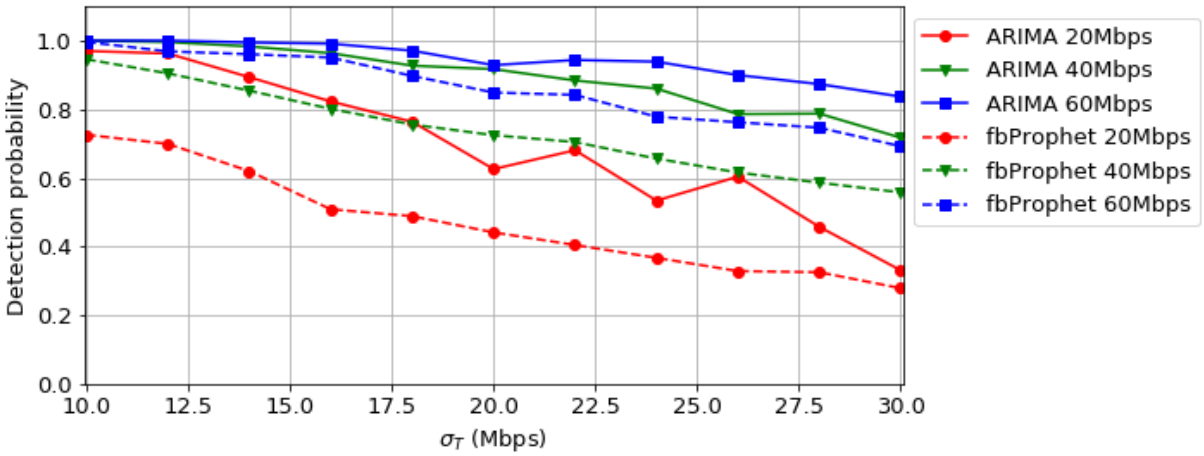
Mokslinės tiriamosios praktikos darbų metu buvo atlikti tiesiniai regresiniai modelių taikymai tokie kaip anomalaus pralaidumo aptikimas bei tendencijos pokyčio tipo tinklo kokybės anomalijų aptikimas. Taip pat buvo palygintas jų efektyvumas. ARIMA bei fbProphet modeliai yra autoregresinių bei slenkamojo vidurkio modelių derinys, kuriame prognozuojama vertė yra išreikšta tiesiškai pagal ankstesnes vertes bei ankstesnių verčių daromą įtaką. Regresinių modelių atveju laiko eilučių vidutinė vertė, o taip pat ir kovariacija laikui bėgant nekinta. Toliau pateikėme dalį svarbiausių rezultatų, kurie buvo pasiekti ir išsiųsti į „Wireless Communications“ žurnalą [18]. 13 pav. bei 14 pav. grafikuose yra pateikiamos ne tik minutinės vertės, tačiau ir valandinės vertės, viršutinės ribos kitimo valandinės vertės bei aptiktų anomalijų taškai.



13 pav. ARIMA modelis, kur standartinis nuokrypis 20 Mbps, o amplitudė 60 Mbps.



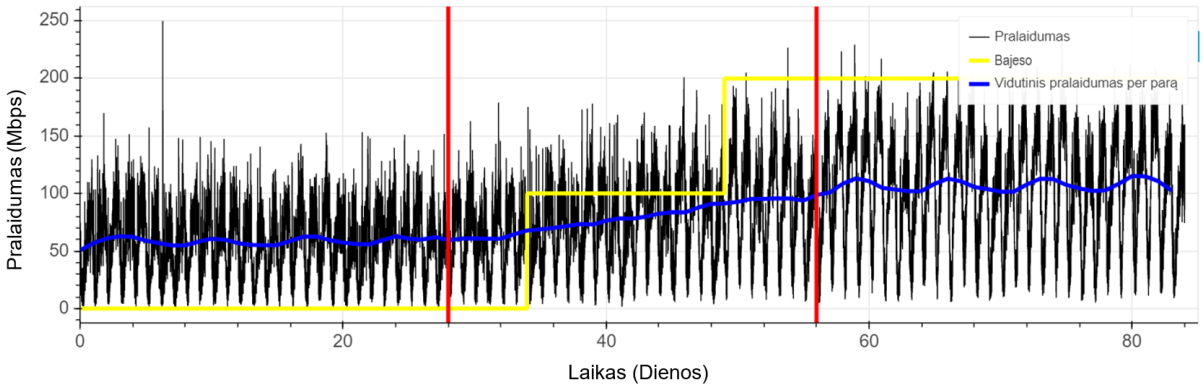
14 pav. fbProphet modelis, kur standartinis nuokrypis 20 Mbps, o amplitudė 60 Mbps.



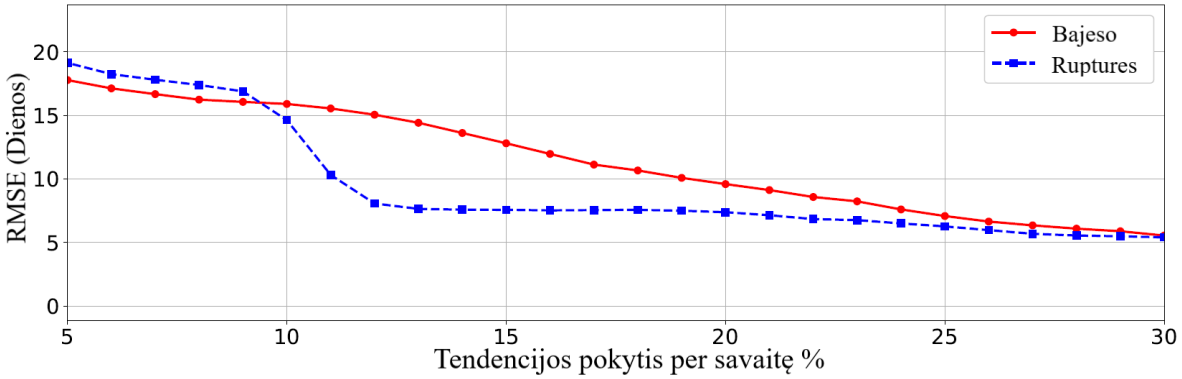
15 pav. Pralaidumo anomalijos aptikimo tikimybės priklausomybė nuo logaritminio standartinio nuokrypio ir esant skirtingoms anomalijos amplitudėms, lyginant ARIMA ir fbProphet modelius [18].

15 pav. pavaizduota pralaidumo anomalijos aptikimo tikimybė, kuri buvo gauta apskaičiavus vidurkį dešimčiai duomenų rinkinių, kiekvienai standartinio nuokrypio bei anomalijos amplitudės vertei. Matome, jog didėjant standartiniam nuokrypiui, anomalijos aptikimo tikimybė mažėja, o esant didesnei anomalijos amplitudei, tikimybė aptikti anomaliją abiejuose modeliuose didėja. Lyginant abu modelius, pagal tikimybės kreivę, galime matyti, kad ARIMA modelis, duotoms laiko eilutėms, aptinka anomalijas vidutiniškai 0,1549 didesne tikimybe. Naudojant 330 skirtingų laiko eilučių iteracijų, kai logaritminis standartinis nuokrypis kito intervale 10 – 30 Mbps ir anomalijos amplitudės variacijos buvo 20, 40, 60 Mbps, ARIMA modelis aptiko 3649 netikras anomalijas, tuo tarpu fbProphet 4876, o tai yra 33,6% daugiau nei ARIMA [18].

Tendencijos pokyčio tipo anomalijoms identifikuoti buvo naudojami Bajeso bei Ruptures algoritmai. 16 pav. grafike yra pateikiamos minutinės vertės, vidutinio per parą pralaidumo vertės, tendencijos pokyčių tikrosios vietos bei algoritmų aptiktos tendencijos.



16 pav. Bajeso modelis, kai standartinis nuokrypis 20 Mbps, o tendencijos pokytis 15% per savaitę.



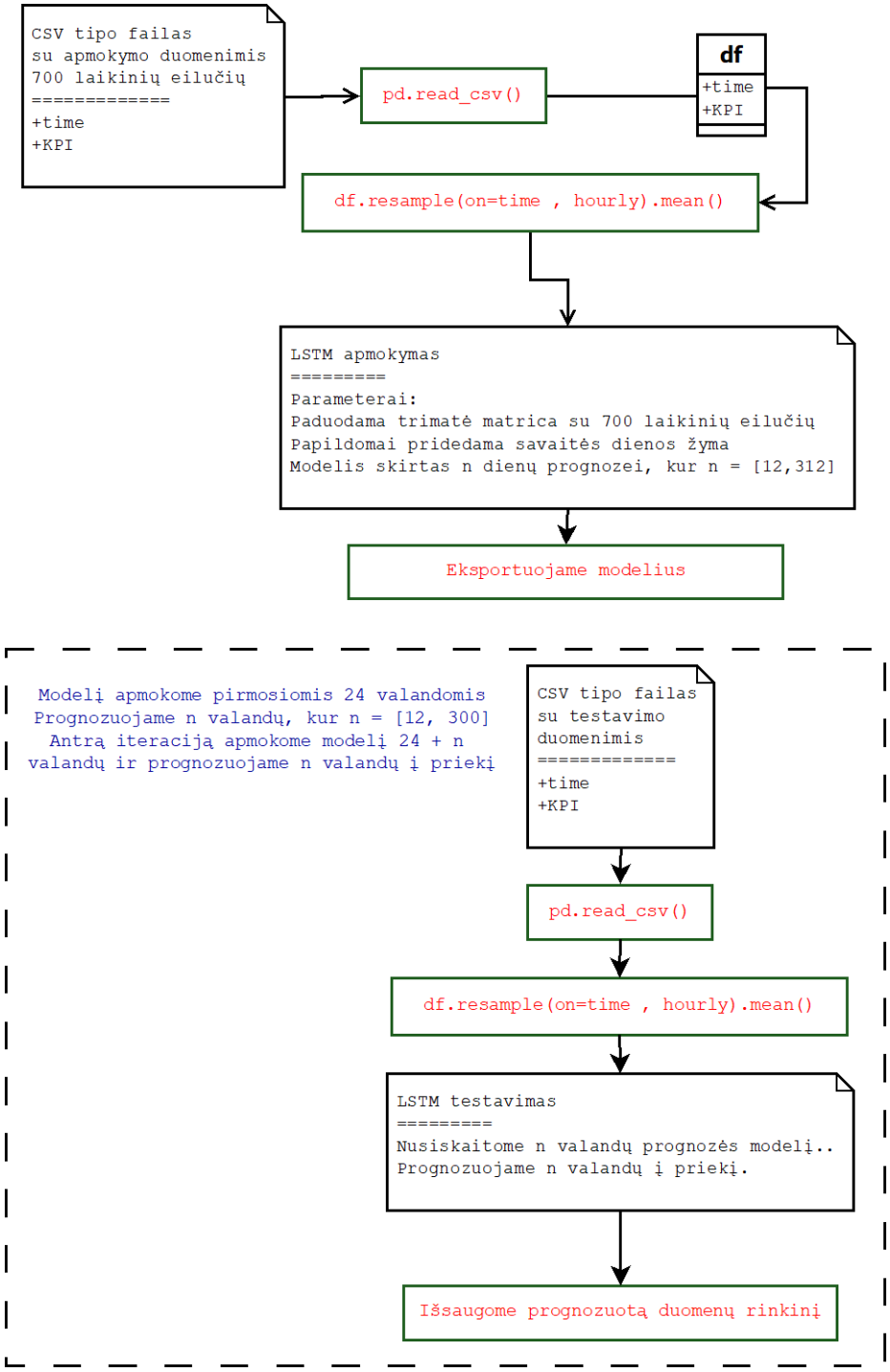
17 pav. Ruptures modelis, kai standartinis nuokrypis 20 Mbps, o tendencijos pokytis 15% per savaitę.

Naudojant Bajeso ir Ruptures modelius bei papildomas funkcijas su duomenimis buvo sukurtas algoritmas, kuris aptinka tinklo pralaidumo tendencijos pokyčius. Ruptures modelis aptiko tendencijos pokyčius, kai jie yra didesni nei 8% per savaitę, o Bajeso modelis aptiko tendencijos pokyčius, kai jie yra didesni nei 10%. Didžiausias apie 7 dienų RMSE skirtumas yra prie mažų tendencijos pokyčių(2-15%) per savaitę, o toliau didinant tendencijos pokyčius RMSE skirtumas mažėja ir ties 30% modelių paklaidos susilygina [18].

2.9 Programavimo aplinka bei duomenų bazė

Python – aukšto lygio į objektą orientuota kalba su dinamine sematika, kuri leidžia kurti tiek programinę įrangą, tiek sudėtingus modelius. Klaidų paieškos ir vykdymo analizė (*angl. debugger*) tikrina vietinius bei globalius kintamuosius [19]. Python vis plačiau yra naudojamas mokslinėje veikloje, tačiau pastaruoju metu yra ypač paplitęs inžineriniuose skaičiavimuose bei modelių kūrime. Pagrindinės naudojamos bibliotekos:

1) Pandas – atviro kodo biblioteka, skirta duomenų rinkimui ir paruošimui.
2) Matplotlib, Seaborn, Bokeh – vizualizacijos bibliotekos.
3) Scikit-learn - atviro kodo biblioteka, skirta atlikti mašininio mokymosi užduotims.
4) NumPy – Python išplėtimo modulis, skirtas darbui su vienarūšiais masyvais. Skirta manipuliuoti skaitiniais duomenimis analogiškai kaip MATLAB.
5) fbProphet – biblioteka skirta laiko eilučių prognozavimui
6) Keras, Tensorflow – rekurentinių neuroninių tinklų bibliotekos, kurias naudojant dirbama su masyvais, norint pasiekti optimaliausią skaičiavimo greitį.

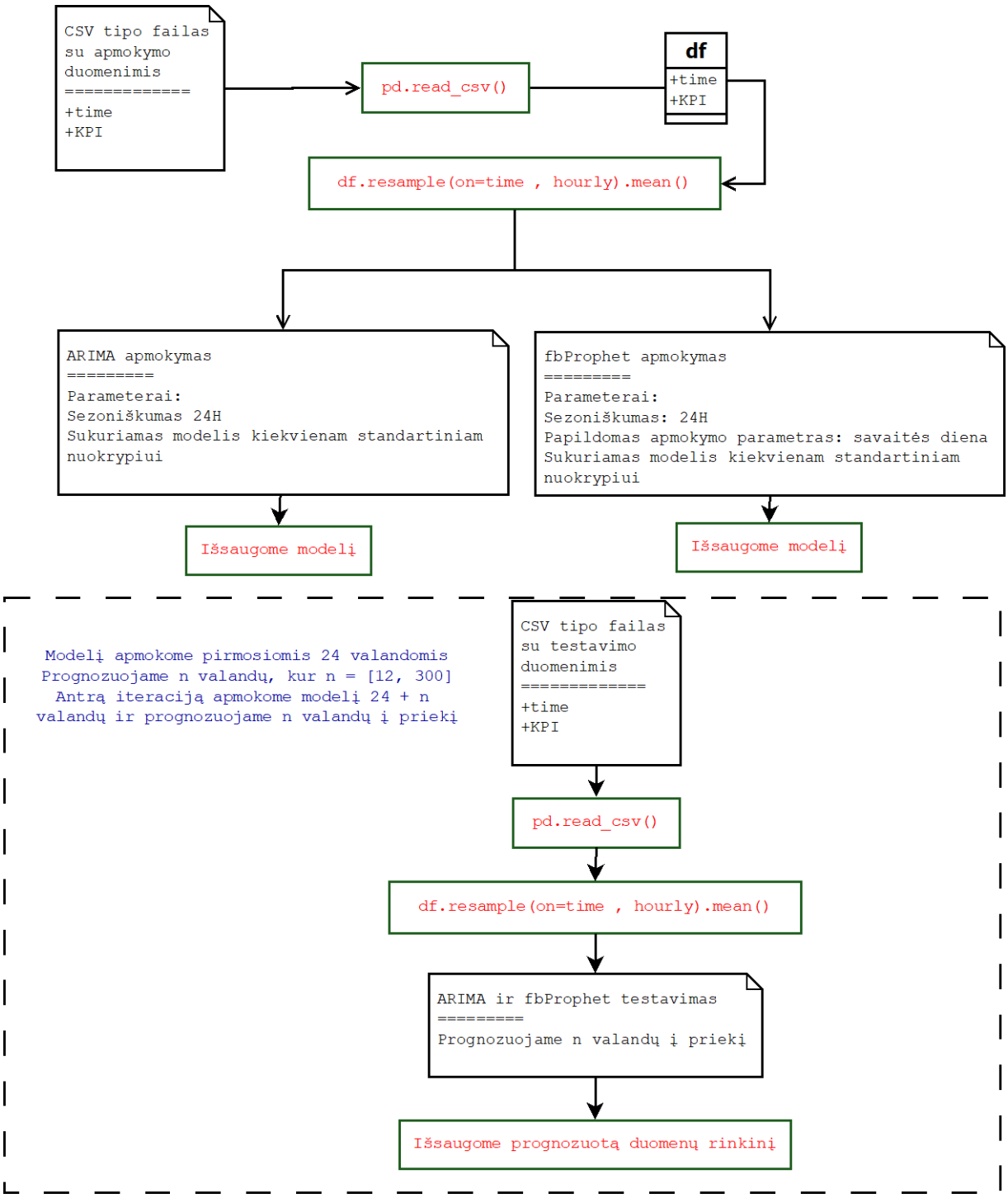


18 pav. LSTM modelio blokinė schema

Duomenų bazė yra sudaryta iš datos ir tinklo kokybės rodiklių, o šiuo atveju pralaidumo. Apmokymas vyksta su 700 sugeneruotų sintetinių laiko eilučių, kurių standartinis nuokrypis kinta nuo 10 iki 30 Mbps. Laiko žingsnis – 10 minučių. Duomenys yra agreguojami į valandinius ir yra sutvarkomi taip, kad patektų į LSTM modelį kaip trimatė matrica, o savaitės diena yra pridedama prie rinkinio kaip papildomas požymis. Apmokome modelį su 700 skirtingų rinkinių. Nusiskaitome naują duomenų rinkinį, kuris dar nebuvo matytas modelio. Apmokome modelį dar kartą pirmomis 24 valandomis naujo rinkinio ir atliekame prognozavimą. Kitą iteraciją modelis apmokomas 24 valandomis bei pridedant praeitos prognozės trukmę. Modelio sluoksnių konfigūracija:

1. LSTM sluoksnis, kur neuronų skaičius lygus prognozės ilgiui. Įėjime yra matrica, kurioje nurodo savaitės diena kaip ypatybė.
2. LSTM sluoksnis, kur neuronų skaičius lygus dvigubai prognozės trukmei.
3. Tankusis sluoksnis (*angl. Dense*), kur neuronų skaičius lygus dvigubai prognozės trukmei, o aktyvacijos funkcija „relu“.
4. Tankusis sluoksnis, kur neuronų skaičius lygus prognozės trukmei, o aktyvacijos funkcija „tanh“.
5. Tankusis sluoksnis, kur neuronų skaičius lygus prognozės trukmei.

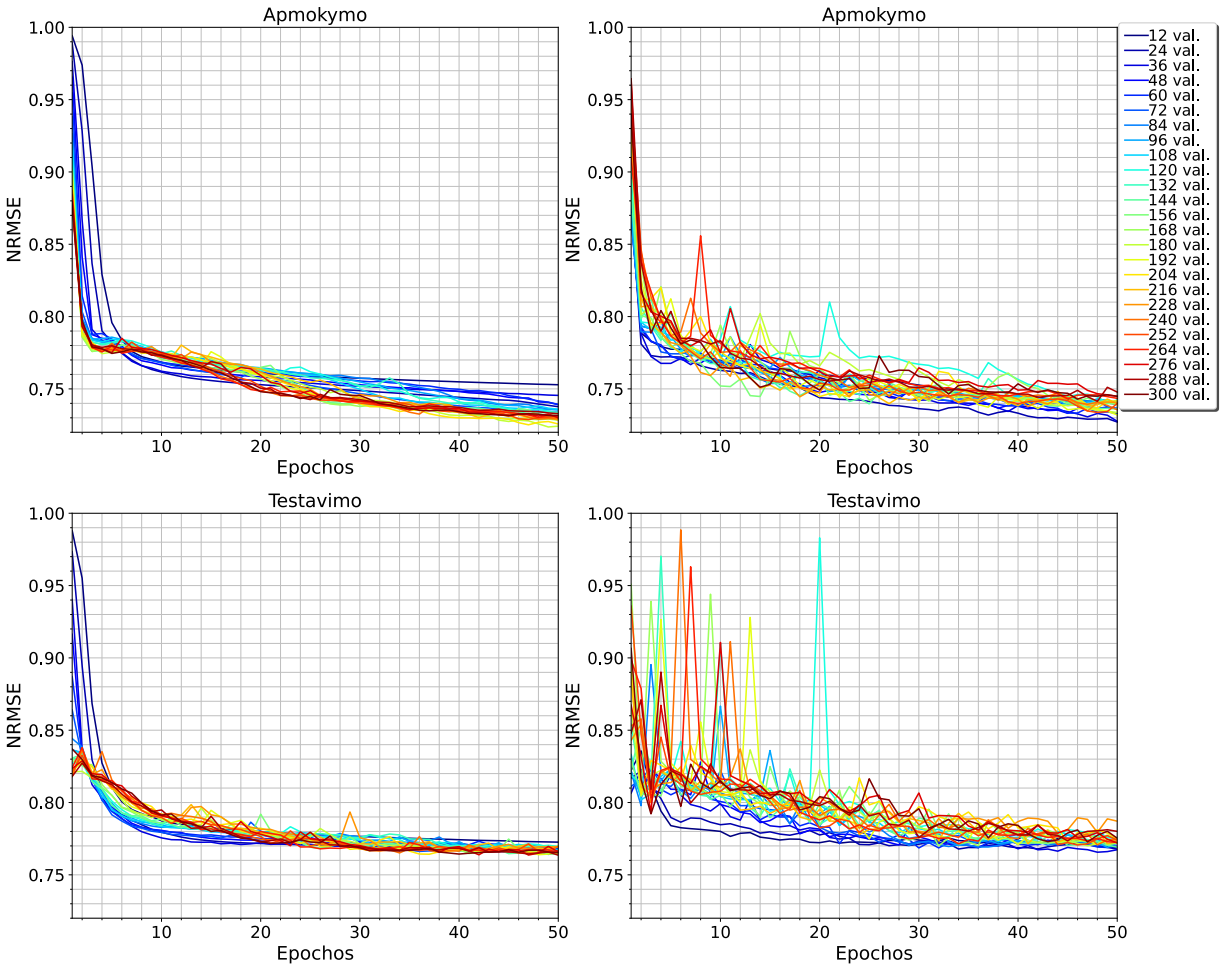
ARIMA bei fbProphet buvo apmokytas parenkant standartinį nuokrypį. Modelio blokinė schema pateikta 14 pav., o kiekvienai laikinei eilutei turime po modelį. Po to apmokome naujo, dar nematyto rinkinio pirmas 24 valandas ir prognozuojame tinklo statistiką.



19 pav. ARIMA ir fbProphet modelių blokinė schema

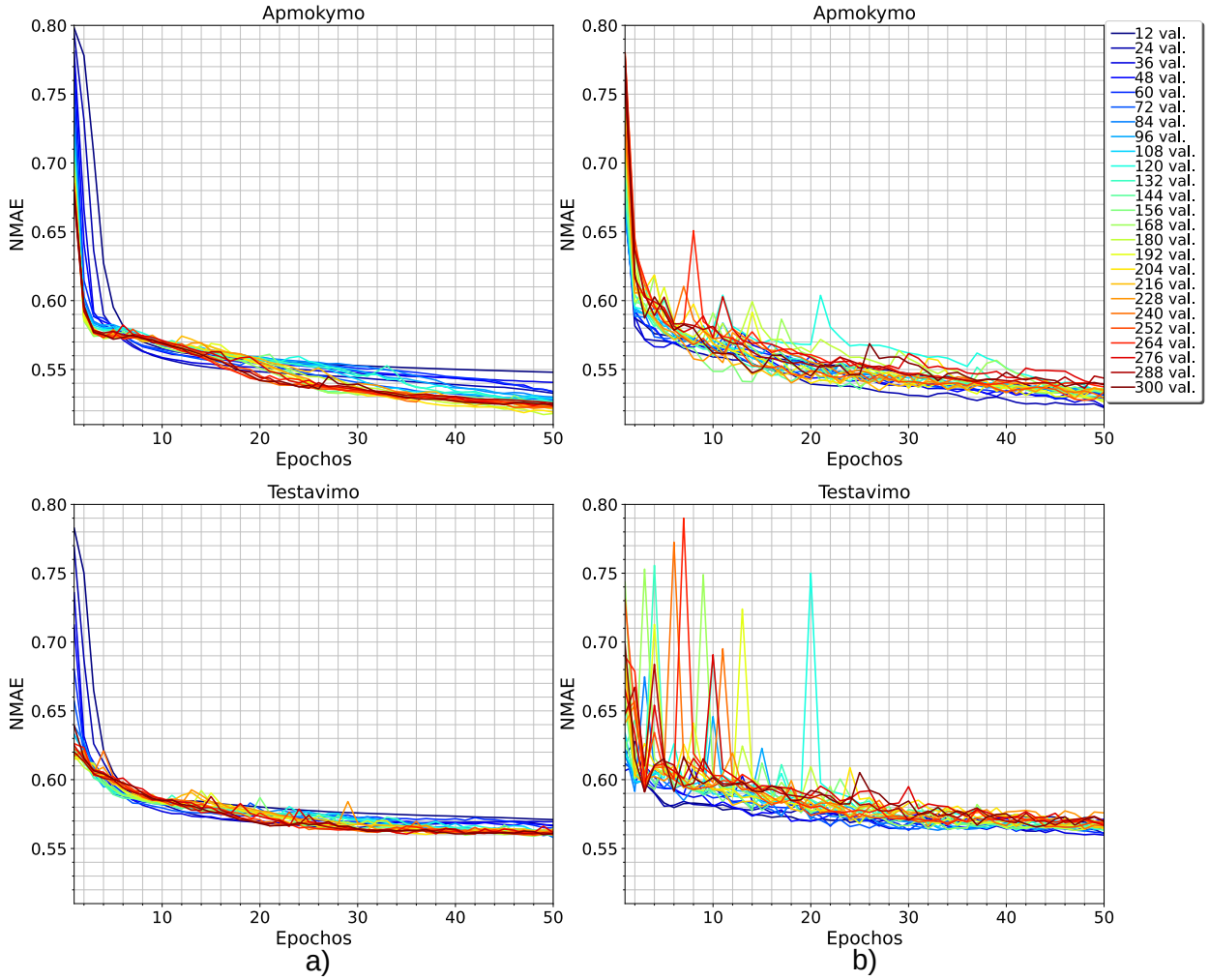
3. Rezultatai

Rezultatai buvo gauti sugeneravus 700 laikinių eilučių apmokymams, kurių standartinis nuokrypis buvo keičiamas nuo 10 Mbps iki 30 Mbps. Skaičiavimams naudotas 2 CPU Xeon E5-2670 serveris, VirtualBox virtualizacija, 30 gijų, Ubuntu operacinė sistema. Tensorflow mašininio mokymosi biblioteka palaiko lygiagrečius skaičiavimus, kai galime lygiagrečiai naudoti daugiau nei vieną giją užduotims atlikti, tačiau net ir vykdant skaičiavimus tokiu būdu vidutiniškai vieno modelio kūrimas trunka apie 3 valandas, o modeliai buvo kuriami 12-300 valandų prognozavimo intervale, kas 12 valandų. Testavimo vienos laikinės eilutės prognozavimas vidutiniškai trunka 5-8 minutes. Tuo tikslu kuriant modelį iš pradžių buvo naudojamos 2-5 laikinės eilutės, kad vyktų greitesni skaičiavimai ir galėtume padaryti įžvalgų apie parametrų kombinacijų daromą įtaką. Parametrų optimizacija vyko ieškant geriausių kombinacijų, esant skirtingiems skaičiams sluoksnių, neuronų bei skirtingoms aktyvacijos funkcijoms. Geriausiais rezultatais pasižymėjo modelis su 5 sluoksniais. Imties dydis yra 24, o tai yra susiję su duomenų periodiškumu, nes turime 24 valandų periodinę seką, o nuostoliai įvertinti su vidutine kvadratine paklaida. Po to buvo sukurta nauja mašininio mokymosi Tensorflow bibliotekoje dar neegzistuojanti metrika – kryžminės koreliacijos paklaida. Panaudojome tuos pačius modelius, kurie atsižvelgia tik į MSE ir papildėme skaičiavimo algoritmą kryžminės koreliacijos paklaidos metrika. Šiuo atveju modelis lygina ne tik pralaidumo vertes, tačiau ir atsižvelgia ir į verčių galimą poslinkį.



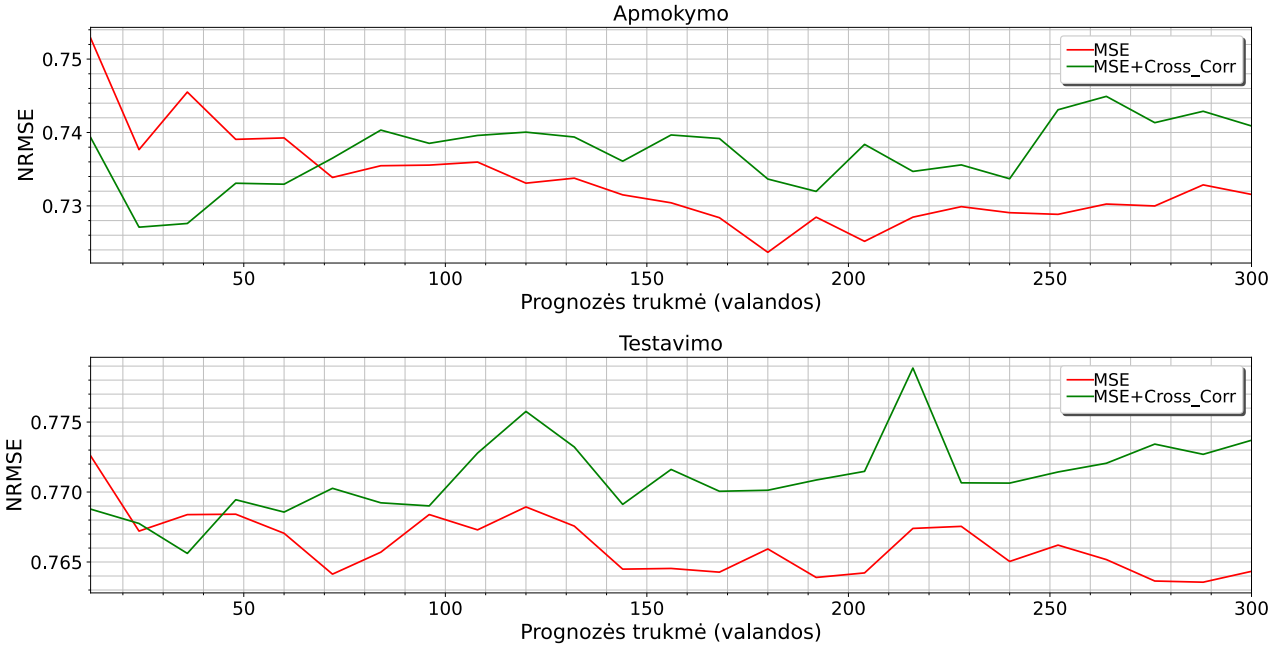
20 pav. Apmokymo bei testavimo NRMSE, kai prognozės trukmė 12-300 valandų intervale. Stulpelyje a) apmokant atsižvelgiama tik į MSE, stulpelyje b) apmokant atsižvelgiama į MSE bei į kryžminės koreliacijos paklaidą.

Lyginant standartinio modelio, kuris atsižvelgia tik į MSE rezultatus, iš 20 pav. a) galime matyti, kaip keičiasi apmokymo ir testavimo NRMSE, keičiant prognozės trukmę. Didžiausiu NRMSE tiek apmokymo, tiek testavimo etapais pasižymi modelis su 12 valandų prognoze, o mažiausiu NRMSE apmokymo fazėje pasižymi 168 valandų prognozė su 0,725 NRMSE. Testavimo fazėje mažiausia verte pasižymi 276 valandų prognozė, kurios NRMSE 0,761. Iš 20 pav. b) galime matyti, jog įvedus naują metriką tiek apmokymo, tiek testavimo fazėse iki 23 epochos matomi nestabilumai, tačiau toliau didinant epochų skaičių rezultatai stabilizuojasi. Stebėdami grafikus 20 pav. b) bei lygindami juos su 20 pav. a) matome, jog rezultatuose su nauja metrika, didinant prognozės trukmę didėja NRMSE, nors naudojant tik MSE metriką, didėjant prognozės trukmei mažėjo NRMSE. Apmokymo metu mažiausią NRMSE, kuris yra 0,727, turėjo 24 valandų prognozė, kai tuo tarpu to pačio modelio testavimo NRMSE buvo 0,767. Testavimo metu mažiausią NRMSE 0,765 turėjo 36 valandų prognozės modelis.



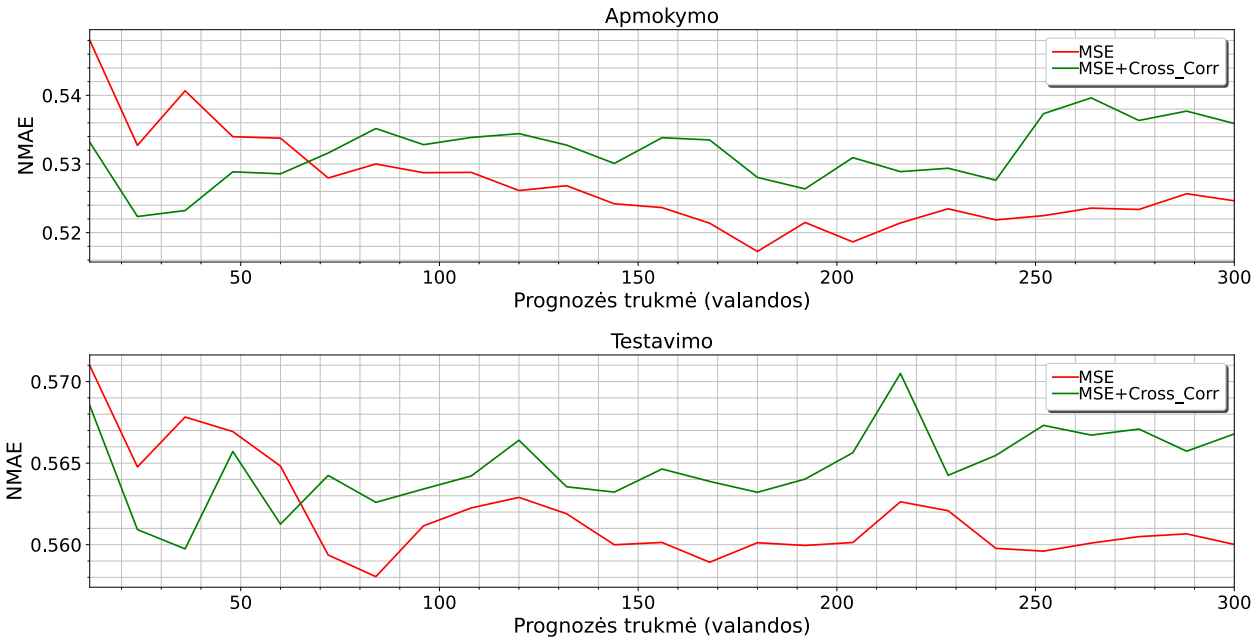
21 pav. Apmokymo bei testavimo NMAE, kai prognozės trukmė 12-300 valandų intervale. Stulpelyje a) apmokant LSTM atsižvelgiama tik į MSE, stulpelyje b) apmokant LSTM atsižvelgiama į MSE bei į kryžminės koreliacijos paklaidą.

Lyginant normuotą vidutinę absoliutinę paklaidą, kuri pateikta 21 pav. a) galime matyti, jog didžiausiu MAE pasižymėjo 12 valandų prognozė tiek apmokymo, tiek testavimo fazėse. Šiuo atveju mažiausias MAE, kuris pasiektas apmokymo fazėje yra 0,511, kai prognozės trukmė 180 valandų. Testavimo fazėje mažiausia vidutine absoliutine paklaida pasižymi 84 valandų prognozė, kurios vertė yra 0,552. Įvertinus 21 pav. b) modelio su kryžminės koreliacijos paklaidos metrika rezultatus matome, jog modelio vidutinė absoliutinė paklaida pasižymi dideliu nestabilumu iki 23 epochos ir esant prognozės trukmei 120-264 valandų intervale. Nestabilumai yra susiję su darbo dienos ir savaitgalio komponente, nes 120 valandų yra 5 darbo dienos, todėl ilgiau trunka identifikuoti pastarąsias komponentes. Tačiau 24 valandų prognozė tiek apmokymo, tiek testavimo metu pasižymėjo mažiausiu NMAE. Apmokymo fazėje mažiausia vidutinė absoliutinė paklaida buvo 0,522 , o testavimo fazėje 0,560.



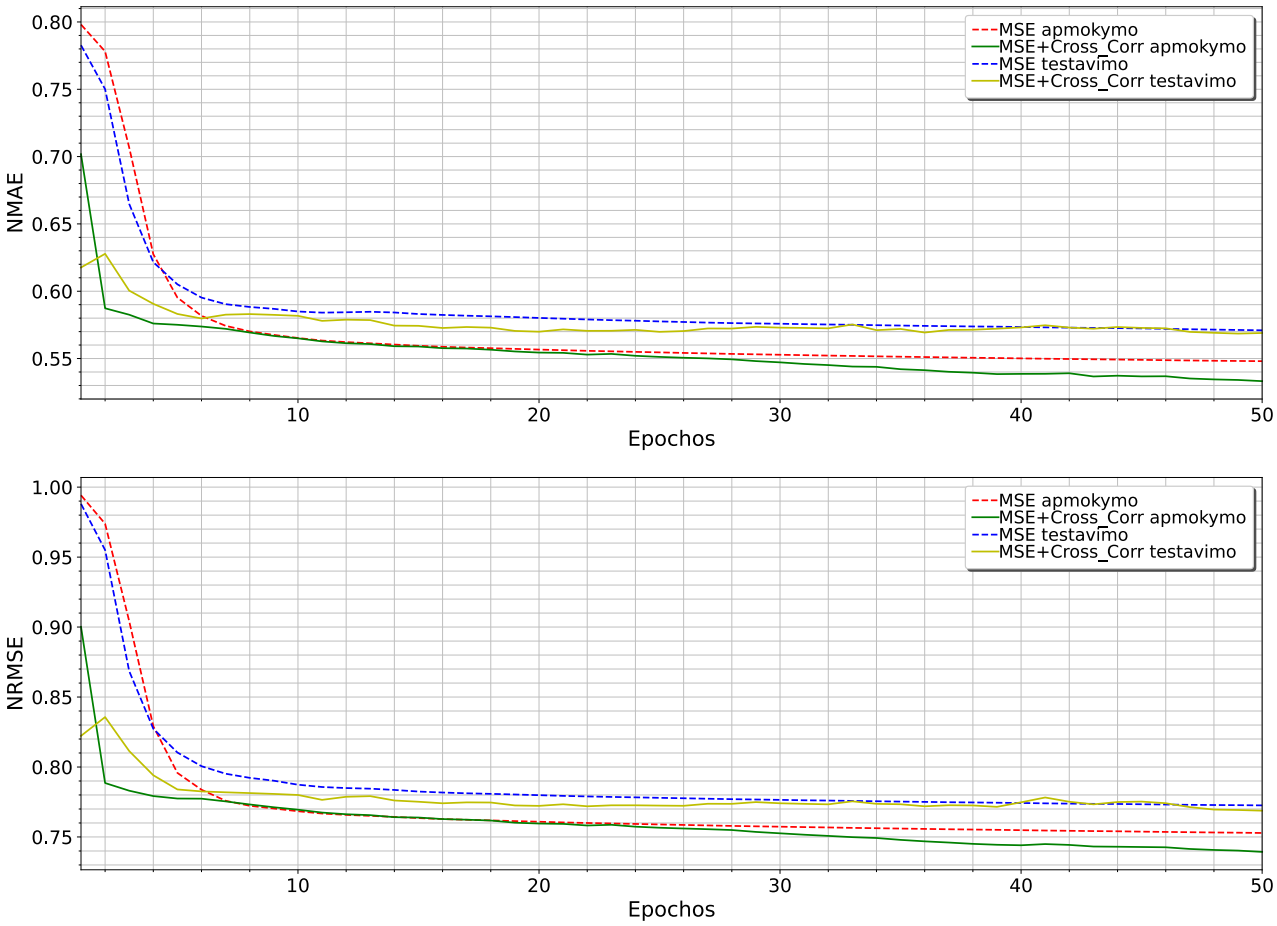
22 pav. Apmokymo bei testavimo fazėse mažiausio normuoto RMSE priklausomybė nuo prognozės trukmės valandomis, kur MSE – LSTM modelis naudojantis vidutinės kvadratinės paklaidos metriką, o MSE+Cross_Corr – LSTM modelis naudojant MSE bei kryžminės koreliacijos paklaidos metrikas.

22 pav. palyginome abiejų modelių mažiausias NRMSE vertės apmokymo ir testavimo metu. Apmokymo fazės grafike galime matyti, jog iki 72 valandos prognozės trukmės modelis su kryžminės koreliacijos paklaida pasižymi vidutiniškai 0,011 mažesniu NRMSE, tačiau nuo 72 valandos prognozės trukmės modelis tik su MSE metrika pasižymi 0,007 mažesniu NRMSE. Statistinis reikšmingumas su 95% patikimumo intervalu buvo nustatytas panaudojant Stjudento testą. Gautos *p* vertės iki 72 valandos buvo mažesnės nei $1{,}6 * 10^{-8}$, o nuo 72 valandos daugiau nei 0,161. Palyginę testavimo NRMSE rezultatus 17 pav. galime matyti, jog šioje fazėje nuo 48 valandos modelis tik su MSE pasižymėjo vidutiniškai 0,005 geresniais rezultatais, tačiau įvertinę *p*, kuri buvo visomis valandomis daugiau nei 0,05, iš Stjudento testo galime tai vertinti kaip lygius rezultatus.



23 pav. Apmokymo bei testavimo fazėse mažiausio normuoto MAE priklausomybė nuo prognozės trukmės valandomis, kur MSE – LSTM modelis naudojantis vidutinės kvadratinės paklaidos metriką, o MSE+Cross_Corr – LSTM modelis naudojant MSE bei kryžmines koreliacijos paklaidos metrikas.

Įvertinus mažiausio NMAE priklausomybę nuo prognozės trukmės, 23 pav. galime matyti, jog apmokymo tiek testavimo metu iki 72 valandos prognozės trukmės modelis su kryžminės koreliacijos paklaida pasižymi vidutiniškai 0,012 NMAE mažesniais rezultatais, o reikšmingumą patvirtina Stjudento testas, kurio *p* vertės buvo mažesnės nei $1{,}2 * 10^{-7}$. Nuo 72 valandos prognozės trukmės modelis tik su MSE metrika pasižymi 0,008 geresniais rezultatais, bet tai galima įvardyti kaip lygius rezultatus, kadangi *p* vertės buvo didesnės nei 0,05.

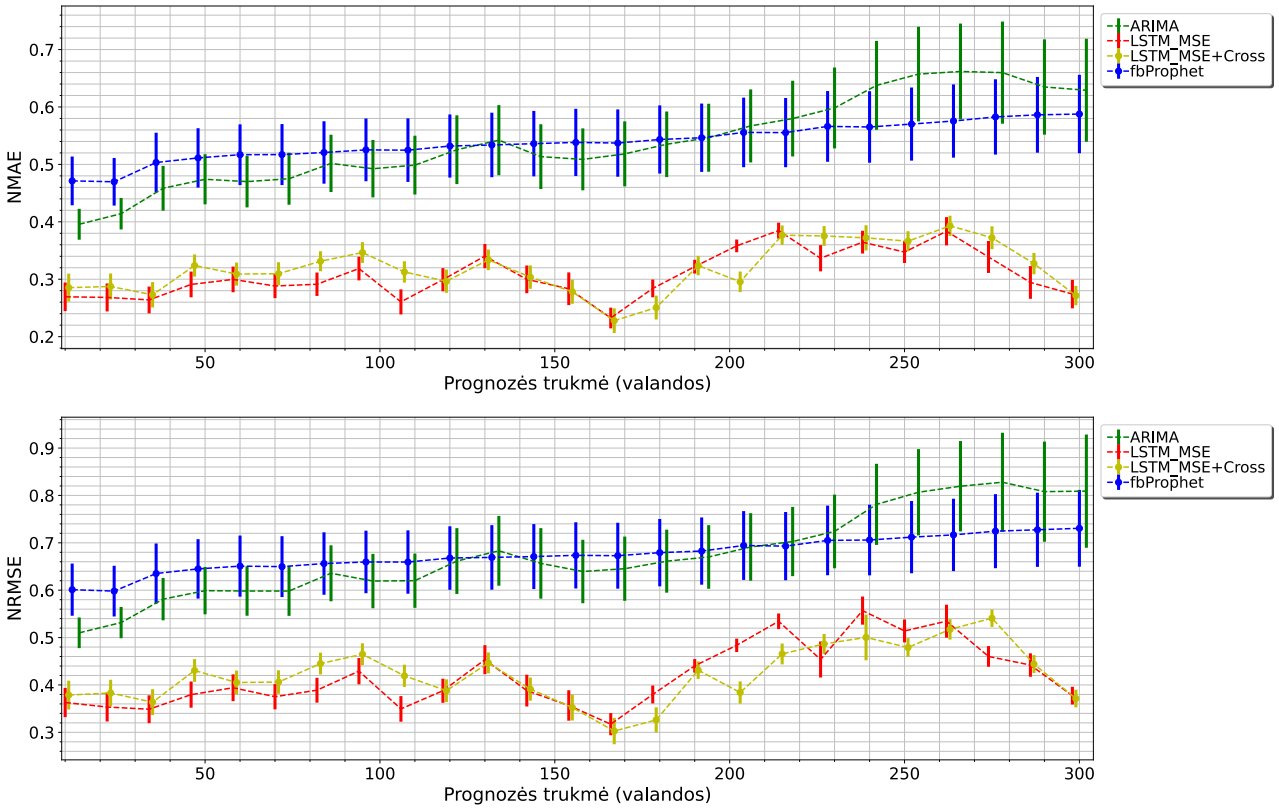


24 pav. MAE bei NRMSE priklausomybės nuo epochų skaičiaus, esant 12 valandų trukmės prognozei, kur MSE – LSTM modelis naudojantis vidutinės kvadratinės paklaidos metriką, o MSE+Cross_Corr – LSTM modelis naudojant MSE bei kryžmines koreliacijos paklaidos metrikas.

24 pav. palyginome apmokymo ir testavimo NRMSE LSTM modelio, kuris apmokomas atsižvelgia į vidutinę absoliutinę paklaidą ir LSTM modelio, kuris atsižvelgia dar ir į kryžminę koreliacijos paklaidą. Esant 12 valandų prognozei, NRMSE testavimo vertės yra artimos viena kitai, ką jau matėme iš prieš tai pateiktų grafikų, tačiau čia dar galime pastebėti, jog modelis apmokomas greičiau su kryžmine koreliacijos paklaida ir jau 3 epochą pasiekia 0,811, kai tuo tarpu modelis su MSE metrika tik 6 epochą pasiekia 0,818 NRMSE. Apmokymo greitis prognozėms iki 48 valandų padidėjo ir jau 3 epochą pasiekiamas vidutiniškai 0,112 mažesnis NRMSE . Normuotos vidutinės absoliutinės paklaidos priklausomybė, kuri taip pat pateikta 19 pav., patvirtina tendenciją, jog modelis greičiau apsimoko. LSTM, kuris naudoja tik MSE metriką, NMAE vertės susilygina su kryžminės koreliacijos modelio vertėmis tik septintą epochą.

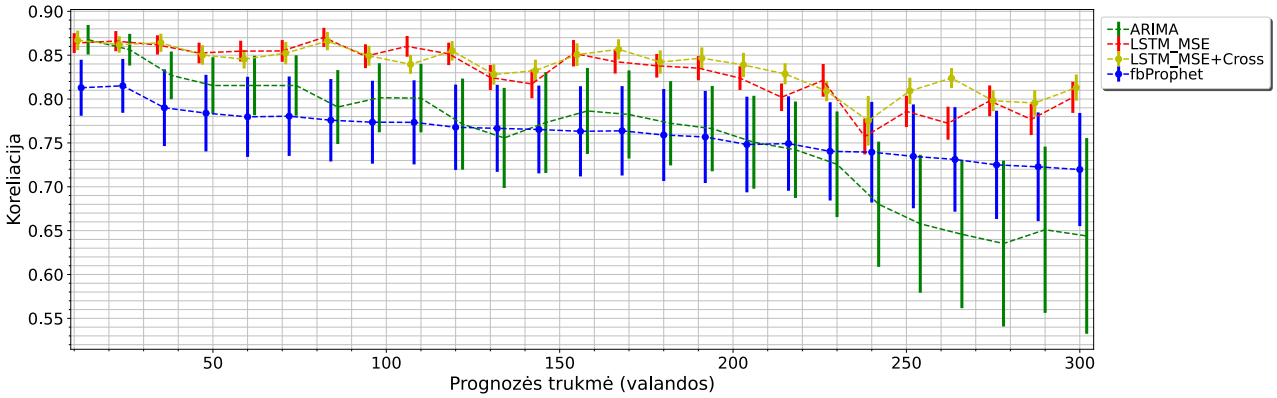
Norint objektyviai įvertinti modelio efektyvumą buvo sugeneruotos dar 300 modeliams nematytų laikinių eilučių ir apskaičiuoti NMAE bei NRMSE prognozuojant visą laikinę eilutę t.y. 13 dienų, tačiau prieš tai modelis apmokomas pirmomis 24 valandomis naujos laikinės eilutės. Metrikos buvo normuotos, tačiau RMSE kitimo skalė yra 8-30 Mbps. Taip pat buvo sukurti ARIMA bei

fbProphet modeliai, kurie [16] minimi kaip baziniai (*angl. baseline*), kad palygintume neuroninių tinklų gaunamus rezultatus su gerai žinomais tiesiniais modeliais.



25 pav. Normuoto RMSE bei MAE priklausomybė nuo prognozės trukmės valandomis, o vertikaliomis linijomis žymimas standartinis nuokrypis. Legendoje LSTM_MSE – LSTM modelis naudojantis vidutinės kvadratinės paklaidos metriką, LSTM_MSE+Cross_Corr – LSTM modelis naudojant MSE bei kryžmines koreliacijos paklaidos metrikas, ARIMA bei fbProphet – tiesiniai laiko eilučių prognozavimo modeliai.

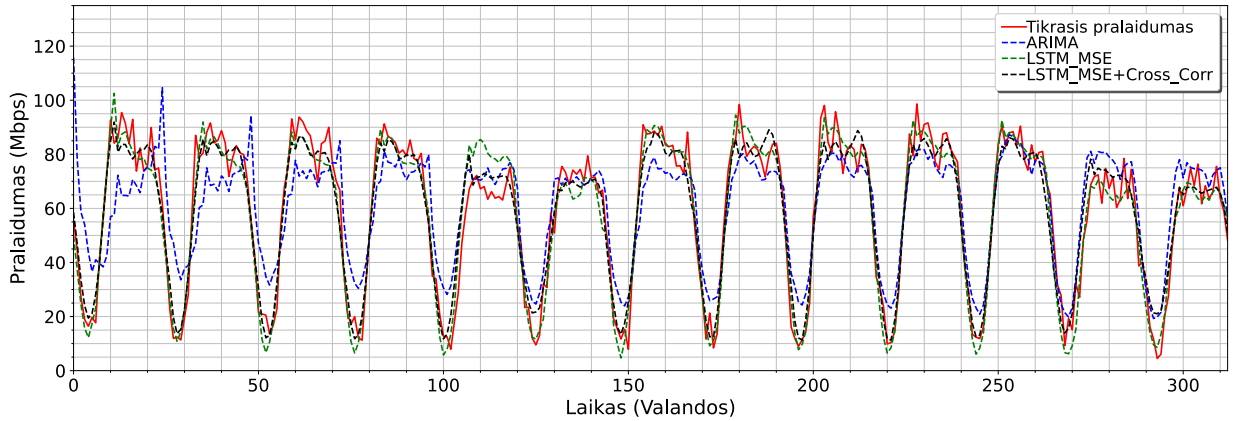
Stebėdami 25 pav. galime matyti, jog lyginant su baziniais ARIMA bei fbProphet modeliais LSTM pasižymi vidutiniškai 0,261 mažesniu NRMSE. MSE LSTM modelis iki 120 prognozės valandos pasižymėjo vidutiniškai 0,031 mažesniu NRMSE, lyginant jį su kryžminės koreliacijos modeliu. Taip pat pateikėme ir normuoto MAE priklausomybę nuo prognozės trukmės 20 pav. Turime analogišką vaizdą normuotai RMSE priklausomybei. Lyginant su baziniais modeliais normuotas MAE esant LSTM modeliams yra vidutiniškai 0,233 mažesnis. LSTM modeliui esant tik su MSE metrika jis pasižymi vidutiniškai 0,003 mažesniu MAE nei esant su kryžminės koreliacijos metrika.



26 pav. Koreliacijos priklausomybė nuo prognozės trukmės valandomis, o vertikaliomis linijomis žymimas standartinis nuokrypis. Legendoje LSTM_MSE – LSTM modelis naudojantis vidutinės kvadratinės paklaidos metriką, LSTM_MSE+Cross_Corr – LSTM modelis naudojant MSE bei kryžmines koreliacijos paklaidos metrikas, ARIMA bei fbProphet – tiesiniai laiko eilučių prognozavimo modeliai.

Stebint koreliacijos priklausomybę nuo prognozės trukmės, galime matyti, jog LSTM modeliai pranoksta bazinius modelius vidutiniškai per 0,083, o Stjudento testo p vertė yra $3{,}5 * 10^{-5}$, todėl galime laikyti statistiškai svarbiu skirtumu. Skirtumas tarp LSTM modelių yra minimalus 0,01. Didėjant prognozės trukmei koreliacija tarp tikrosios laikinės eilutės ir prognozuotos mažėja. LSTM modelio koreliacijos mažėjimo polinkis yra 0,003 per 12 valandų, kai tuo tarpu bazinių modelių vidutiniškai 0,012.

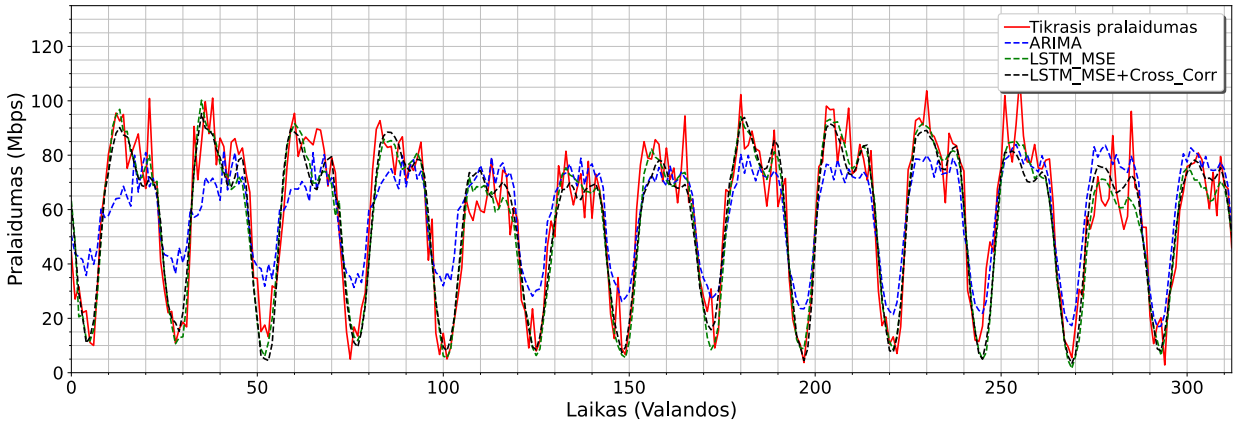
Po to buvo pasirinkti duomenų rinkiniai, kurių NRMSE yra artimiausias ARIMA modelio vidutiniam NRMSE bei palygintos laikinės eilutės prognozės su įprastiniu LSTM modeliu bei su kryžminės koreliacijos LSTM modeliu.



27 pav. Tikrojo ir prognozuoto pralaidumo laikinis vaizdas, kai prognozės trukmė 12 valandų.

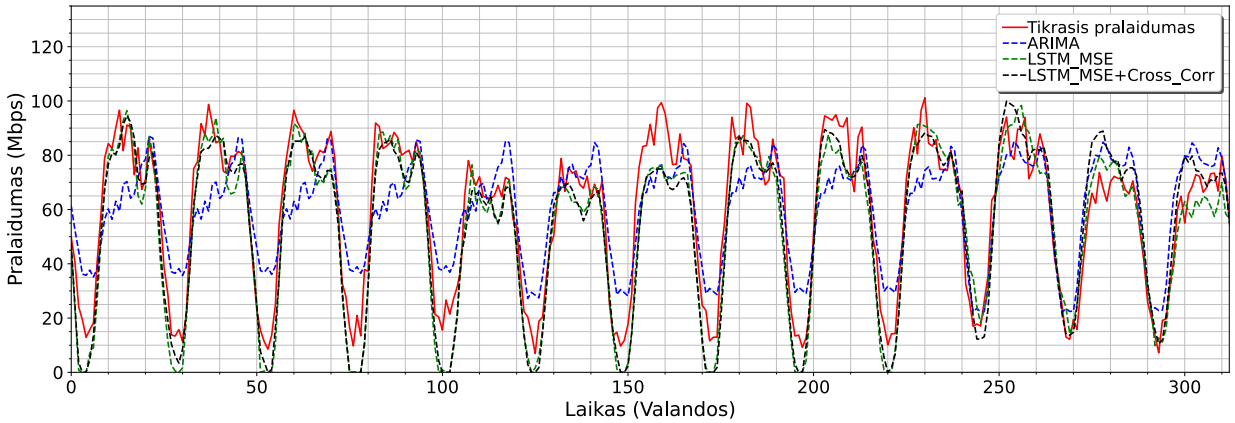
Iš 27 pav. galime matyti tinklo pralaidumo prognozuotą ir tikrąjį laikinį vaizdą. Šiuo atveju NRMSE yra ARIMA yra 0,515 , LSTM su MSE 0,232, o LSTM modelio su kryžmine koreliacija 0,241. Taip pat atitinkamai ir koreliacijos ARIMA 0,864 , LSTM su MSE 0,881 , LSTM su kryžmine

koreliacija 0,912. ARIMA modelio prognozė pirmąją savaitę pasižymėjo poslinkiu bei besiskiriančia amplitude. Antrąją savaitę poslinkio galime nebepastebėti , tačiau prognozės pralaidumo amplitudė dar vis skiriasi nuo tikrosios. LSTM abiejų modelių atveju poslinkis vizualiai yra nepastebimas 13 dienų laikotarpyje. Tiek pirmą tiek antrą savaitę abu modeliai pasižymi pralaidumo kitimo amplitude analogiška tikrajam pralaidumui.



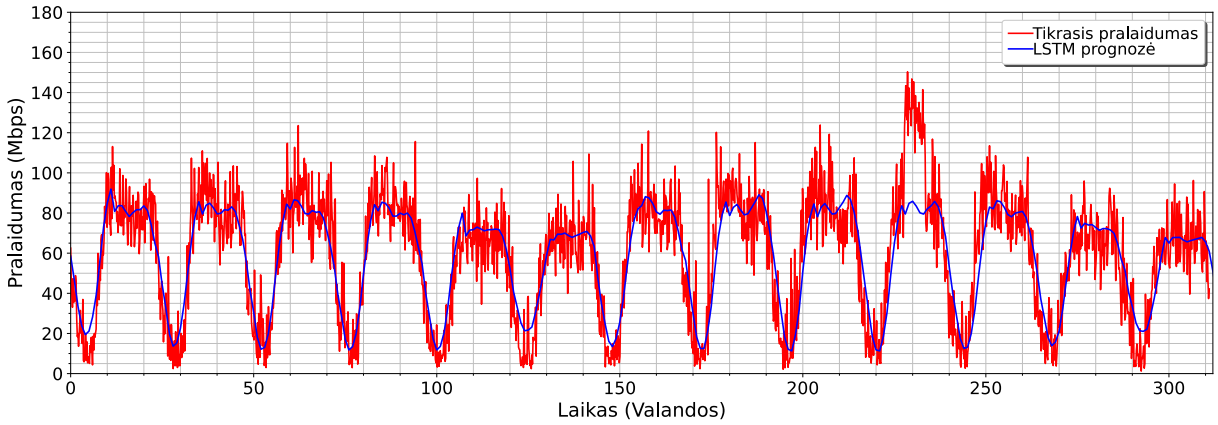
28 pav. Tikrojo ir prognozuoto pralaidumo laikinis vaizdas, kai prognozės trukmė 36 valandos.

Pasirinkus modelius, kai prognozės trukmė 36 valandos , ARIMA , LSTM su MSE ir LSTM su kryžmine koreliacija NRMSE yra 0,551, 0,332, 0,341 bei koreliacija 0,852, 0,877, 0,898 atitinkamai modeliams. Stebint 28 pav. galime matyti, jog pirmąją savaitę ARIMA modelio atveju turime analogišką vaizdą kaip ir 12 valandų prognozei t.y. pirmąją savaitę matomas amplitudes skirtumas. LSTM atveju nei poslinkis, nei amplitudės skirtumai vizualiai nepastebimi.



29 pav. Tikrojo ir prognozuoto pralaidumo laikinis vaizdas, kai prognozės trukmė 120 valandų.

Esant 5 dienų prognozei, kuri pateikta 29 pav., galime pastebėti jau ir LSTM modelių netikslumus. Pirmąsias 9 dienas pasižymėjo vidutiniškai 8 Mbps mažesne apatine riba. Šiuo atveju modelių NRMSE buvo: ARIMA 0,671, LSTM su MSE 0,332, o LSTM su kryžmine koreliacijos paklaida 0,322.



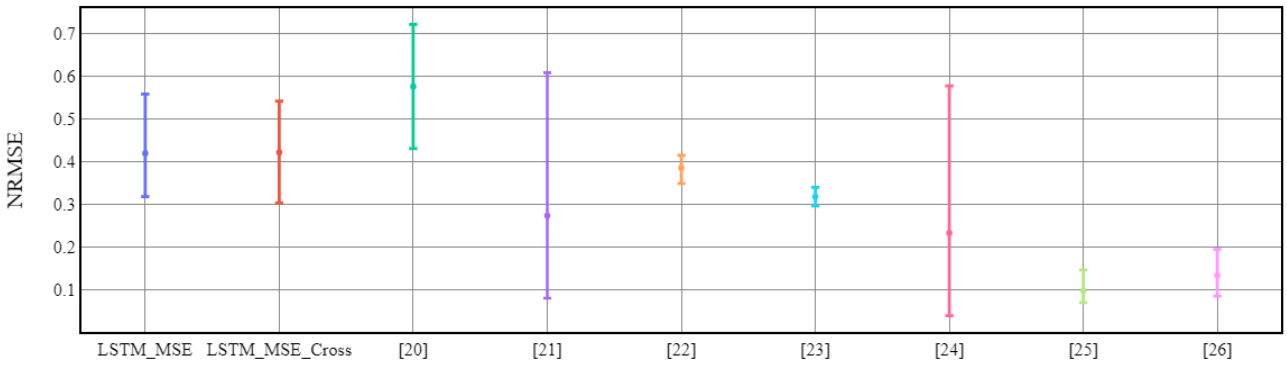
30 pav. Tikrojo pralaidumo su anomalija ir LSTM valandinio prognozuoto pralaidumo laikinis vaizdas, kai prognozės trukmė 24 valandos.

Taip pat buvo pateikta 30 pav. laikinė eilutė, kurioje turime anomaliją 227-234 valandų intervale bei LSTM modelio prognozė. Pritaikydami papildomą grupavimą pagal valandas, lyginant pasiskirstymus bei nustatant kritinę ribą, naudojant sukurtą modelį galime identifikuoti anomalijas išskirdami jas iš įprastos, tinklui būdingos veiklos. Šiuo atveju sumodeliuotas vienos iš bazinių stočių išsijungimas, kas lemia tinklo persiskirstymą pagal geriausią signalo lygį.

Galiausiai galime palyginti LSTM modelius su skirtingais publikuotų straipsnių rezultatais pasiektais naudojant LSTM modelius bei prognozuojant laikines eilutes. Kadangi duomenys buvo pateikti kaip nenormuoti, todėl skaitmenizavome straipsniuose pateiktus grafikus ir apskaičiavome normuotą RMSE. Trumpai aptarsime straipsniuose pateiktų modelių principus:

1. [20] pateikiami du eksperimentiniai LSTM modeliai. Pirmas yra apmokomas porcijomis, o antras apmokomas realiu laiku. Prognozės trukmė 5-60 minučių.
2. [21] atveju pateiktas įprastinis LSTM modelis bei palyginta su rezultatais, kai kombinuojama su Gausinių procesų regresija. Prognozės trukmė 1-30 valandų.
3. [22] modeliui buvo pateikta papildomų bruožų apie duomenų rinkinį, kas lėmė modelio prognozės pagerėjimą. Prognozės trukmė 15-30 minučių.
4. [23] buvo panaudotas LSTM konvoliucijos principu. Prognozės trukmė 2 valandos.
5. [24] pateiktas erdvinis bei laikinis prognozavimas. Naudojamas K-vidutinių verčių grupavimas, o po to, remiantis duomenų grupėmis, pagal panašumą tai vietovei prognozuotos laikinės eilutės. Grupavimas lėmė prognozės pagerėjimą. Prognozavimo trukmė yra 1 valanda.

6. [25] naudojamas deterministinis gradientas norint pagerinti standartinio LSTM modelio prognozę. Deterministrinio gradiento būdu apmokomas modelis daugiau kartų kartoja apmokymą bei parinktas labai mažas apmokymo greitis. Prognozės trukmė 6 valandos.
7. [26] buvo panaudotas konvoliucinis LSTM, duomenų grupavimas, kad suskirstytume juos į zonas. Prognozės trukmė 1 valanda.



31 pav. Sukurto LSTM modelio palyginimas su publikuotų straipsnių rezultatais. Vertikaliomis linijomis žymimi didžiausias bei mažiausias pasiektas NRMSE.

Pagal 31 pav. bei įvertinus Stjudento testo rezultatus gavome, jog rezultatai yra artimi bei palyginami su [20] , [21], [22] straipsnių rezultatais su 95% patikimumo intervalu. Tačiau reikšmingai geresniais rezultatais pasižymėjo [23], [24], [25] bei [26] straipsniai. Mažiausias 0,08 NRMSE buvo pasiektas [25] pasinaudojant deterministiniu gradientu. Tačiau būtina paminėti tai, jog mūsų darbo metu buvo ištirta modelio NRMSE priklausomybė nuo prognozės trukmės dviejų savaičių laikotarpyje, panaudota 700 laikinių eilučių pasižyminčių skirtingomis tendencijomis apmokymui, o testavimo rezultatai gauti naudojant 300 naujų laikinių eilučių taip pat su skirtingais standartiniais nuokrypiais, kas reiškia naujo ir dar nematyto tinklo imitavimą.

Dalis darbo rezultatų buvo išsiųsti į žurnalą [18] "Wireless Personal Communications“, panaudojant bazinius modelius ir papildomus statistinius skaičiavimus identifikuojant anomalijas bei tendencijos pokyčius. Modelių pritaikymo galimybės yra labai plačios: identifikavimas probleminių vietų tinkle, kaip pateikta 30 pav., bazinių stočių užmigdymas, kad būtų mažinamas elektros suvartojimas, kaip pateikta [25], tačiau siekiamybė yra SON įgyvendinimas, kad tinklas pats keistų antenų polinkio kampus vienai iš bazinių stočių išsijungus, prisidėtų perjungimams kaimynus, praneštų apie technines problemas bazinėse stotyse, trukdžius bei mažą signalo lygį [27].

4. Išvados

1. Naudojant LSTM rekurentinius neuroninius tinklus bei papildomas duomenų transformacijos funkcijas buvo sukurti modeliai, kurie prognozuoja mobiliojo ryšio tinklo pralaidumo laikinę priklausomybę 12-300 valandų intervale. Gauti rezultatai buvo palyginti su 7 straipsniais, su 3 yra lygiaverčiai, o nusileidžia 4 straipsniams su statistiškai reikšmingu 95% patikimumo lygiu.
2. Įvedus naują kryžminės koreliacijos paklaidos metriką modelio apmokymo normuota vidutinė kvadratinės šaknies paklaida NRMSE iki 72 prognozės valandos yra vidutiniškai 0,011 mažesnė nei modelio tik su vidutine absoliutine paklaida, kai tuo tarpu testavimo NRMSE išlaikomas nepakitęs, naudojant 95% reikšmingumo kriterijų.
3. Pridėjus kryžminės koreliacijos paklaidos metriką modelio apmokymo greitis prognozėms iki 48 valandų padidėjo ir jau 3 epochą pasiekiamas vidutiniškai 0,112 mažesnis NRMSE.
4. Lyginant sukurtą LSTM tinklo modelį su tradiciniais ARIMA bei fbProphet modeliais, LSTM pasižymi vidutiniškai 0,261 mažesne NRMSE paklaida, vidutiniškai 0,083 didesne koreliacija, o rezultatas yra statistiškai reikšmingas.

Literatūra

[1] Joshi, Manish, and Theyazn Hassn Hadi. “A Review of Network Traffic Analysis and Prediction Techniques.” July 2015.

[2] Chen, Mingzi, et al. “Deep-Broad Learning System for Traffic Flow Prediction toward 5G Cellular Wireless Network.” 2020 International Wireless Communications and Mobile Computing (IWCMC), 2020, pp. 940–45. IEEE Xplore, doi:10.1109/IWCMC48107.2020.9148092.

[3] D.Garuolis bakalauro studijų baigiamasis darbas "LTE ryšio tinklo kokybės rodiklių analizė naudojant didžiųjų duomenų analitikos metodus", 2019.

[4] Kibria, Mirza Golam, et al. “Big Data Analytics, Machine Learning, and Artificial Intelligence in Next-Generation Wireless Networks.” IEEE Access, vol. 6, 2018, pp. 32328–38. IEEE Xplore, doi:10.1109/ACCESS.2018.2837692.

[5] Geng, Hwaiyu. “DATA ANALYTICS AND PREDICTIVE ANALYTICS IN THE ERA OF BIG DATA.” Internet of Things and Data Analytics Handbook, Wiley, 2017, pp. 329–45. IEEE Xplore, doi:10.1002/9781119173601.ch19.

[6] Parwez, Md Salik, et al. “Big Data Analytics for User-Activity Analysis and User-Anomaly Detection in Mobile Wireless Network.” IEEE Transactions on Industrial Informatics, vol. 13, no. 4, Aug. 2017, pp. 2058–65. IEEE Xplore, doi:10.1109/TII.2017.2650206.

[7] G. E. P. Box and G. M. Jenkins, Time Series Analysis: Forecasting and Control, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1994.

[8] Zhu, B. and S. Sastry. “Revisit Dynamic ARIMA Based Anomaly Detection.” 2011 IEEE Third Int'l Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third Int'l Conference on Social Computing (2011): 1263-1268.

[9] Yaacob, Asrul H., et al. “ARIMA Based Network Anomaly Detection.” 2010 Second International Conference on Communication Software and Networks, 2010, pp. 205–09. IEEE Xplore, doi:10.1109/ICCSN.2010.55.

[10] "ARIMA" https://assets.digitalocean.com/articles/eng_python/arima/part_2_fig_3.png [Tikrinta: 2021-05-23]

[11] J. D. Brutlag, “Aberrant Behavior Detection in Time Series for Network Monitoring,” Proceedings of the 14th Systems Administration Conference (LISA 2000), Dec. 2000.

[12] Taylor, Sean & Letham, Benjamin "Forecasting at Scale", 2017.

[13] Jaffry, Shan. “Cellular Traffic Prediction with Recurrent Neural Network.” ArXiv:2003.02807 [Cs], Mar. 2020

[14] Huang, Chih-Wei & Chiang, Chiu-Ti & Li, Qiuhui "A study of deep learning networks on mobile traffic forecasting.", 2017, 1-6. 10.1109/PIMRC.2017.8292737.

[15] Abdel-Nasser et al., Abdel-Nasser, M., Mahmoud, K., Omer, O. A., Lehtonen, M., and Puig, D. "Link quality prediction in wireless community networks using deep recurrent neural networks", 2020.

[16] Iqbal, Muhammad & Zahid, Muhammad & Habib, Durdana & John, Lizy "Efficient Prediction of Network Traffic for Real-Time Applications. Journal of Computer Networks and Communications", 2019. 1-11. 10.1155/2019/4067135.

[17] Villegas, Javier, et al. “Social-Aware Forecasting for Cellular Networks Metrics.” IEEE Communications Letters, 2021, pp. 1–1. IEEE Xplore, doi:10.1109/LCOMM.2021.3065812.

[18] Rimvydas Aleksiejūnas, Deividas Garuolis, "Usage of Published Network Traffic Datasets for Anomaly and Change Point Detection". Išsiųsta į "Wireless Personal Communications" 2021-03-23.

[19] “Python.” https://www.python.org/doc/essays/blurb/. [Tikrinta: 2021-05-23].

[20] Mackenzie, Jonathan, et al. “An Evaluation of HTM and LSTM for Short-Term Arterial Traffic Flow Prediction.” IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 5, May 2019, pp. 1847–57. IEEE Xplore, doi:10.1109/TITS.2018.2843349.

[21] Wang, Wei, et al. “Cellular Traffic Load Prediction with LSTM and Gaussian Process Regression.” ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 2020, pp. 1–6. IEEE Xplore, doi:10.1109/ICC40277.2020.9148738. [22] D. Chen, C. Xiong and M. Zhong, "Improved LSTM Based on Attention Mechanism for Short-term Traffic Flow Prediction", 2020.

[23] Essien, Aniekan E., et al. “A Scalable Deep Convolutional LSTM Neural Network for Large-Scale Urban Traffic Flow Prediction Using Recurrence Plots.” 2019 IEEE AFRICON, 2019, pp. 1–7. IEEE Xplore, doi:10.1109/AFRICON46755.2019.9134031.

[24] Paul, Udita, et al. “Traffic-Profile and Machine Learning Based Regional Data Center Design and Operation for 5G Network.” Journal of Communications and Networks, vol. 21, no. 6, Dec. 2019, pp. 569–83. IEEE Xplore, doi:10.1109/JCN.2019.000055.

[25] Wu, Qiong, et al. “Deep Reinforcement Learning with Spatio-Temporal Traffic Forecasting for Data-Driven Base Station Sleep Control.” ArXiv:2101.08391 [Cs], Jan. 2021.

[26] Zhang, Chuanting, et al. “Deep Transfer Learning for Intelligent Cellular Traffic Prediction Based on Cross-Domain Big Data.” IEEE Journal on Selected Areas in Communications, vol. 37, no. 6, June 2019, pp. 1389–401. IEEE Xplore, doi:10.1109/JSAC.2019.2904363.

[27] Lin, Zhengyi “A Machine Learning Assisted Method of Coverage and Capacity Optimization (CCO) in 4G LTE Self Organizing Networks (SON).” 2019 Wireless Telecommunications Symposium (WTS), 2019, pp. 1–9. IEEE Xplore, doi:10.1109/WTS.2019.8715538.

Santrauka

LTE ryšio tinklo pralaidumo modeliavimas naudojant mašininio mokymosi metodus

Judriojo ryšio progresas nenumaldomai greitėja. Naujos bevielio ryšio technologijos bei jų generuojamų duomenų skaičius auga kasmet. Padidėję duomenų kiekiai bei išaugęs vartotojų skaičius privertė tinklo operatorius ieškoti pažangesnių būdų norint optimizuoti tinklą. Didžiųjų duomenų analitikos bei mašininio mokymosi kombinacija tapo perspektyvia sritimi, norint geriau įvertinti tinklo veiklą, galinio vartotojo patirtį bei identifikuoti problemines vietas tinkle. Šio darbo pagrindinis tikslas, naudojantis sumodeliuoto LTE tinklo statistiniais duomenimis ir papildomais duomenų šaltiniais sudaryti didžiųjų duomenų analizei skirtus modelius, kurie galėtų prognozuoti tinklo veiklą. Buvo pasirinkti mašininio mokymosi baziniai algoritmai – ARIMA, fbProphei bei rekurentinių neuroninių tinklų modelis - LSTM. Darbo metu buvo ištirtos NRMSE, NMSE, koreliacijos metrikų priklausomybės, keičiant prognozės trukmę 12-300 valandų intervale. Įvedus naują kryžminės koreliacijos paklaidos metriką modelio apmokymo normuota vidutinė kvadratinės šaknies paklaida NRMSE iki 72 prognozės valandos yra vidutiniškai 0,011 mažesnė, kai tuo tarpu testavimo NRMSE išlaikomas nepakitęs, o gautų rezultatų statistinis reikšmingumas patvirtintas su Stjudento testu. Pridėjus kryžminės koreliacijos paklaidos metriką modelio apmokymo greitis iki 48 prognozės valandos padidėjo ir jau 3 epochą pasiekiamas vidutiniškai 0,112 mažesnis NRMSE. Lyginant sukurtą LSTM tinklo modelį su tradiciniais ARIMA bei fbProphet modeliais, LSTM pasižymi vidutiniškai 0,261 mažesne NRMSE paklaida nei modelis tik su MSE metrika, o taip pat vidutiniškai 0,083 didesne koreliacija, pagrindžiant reikšmingumą Stjudento testu. Gauti rezultatai buvo palyginti su 7 straipsniai, su 3 yra lygiaverčiai, o nusileidžia 4 straipsniams su statistiškai reikšmingu 95% patikimumo lygiu.

Summary

Modeling of LTE Network Traffic Using Machine Learning Methods

The progress of mobile communication is relentlessly increasing. New mobile technologies and the amount of generated data is growing every year. Increased data volumes and number of users forced network operators to find more advanced solution to optimize their network. The combination of big data and machine learning has become a promising are to better assess network performance indicators, end user experience and identify problematic areas in network. The main goal of this thesis is by using the statistics of the modeled LTE network and additional data sources to create models for big data analysis that could predict the network performance. We used LSTM recurrent neural network model and ARIMA, fbProphet as baseline models. During the work we investigated NRMSE, NMSE, correlation metric dependencies when forecast was in range of 12-300 hours. We introduced new cross-correlation error metric that led NRMSE in training phase to decrease by 0,011 compared to LSTM model only with MSE metric, while testing phase NRMSE was around the same and the statistical significance of the obtained results is confirmed by Student's test. With the addition of cross-correlation error metric, the training rate of the model increased with forecast of up to 48 hours and in third epoch reached and average of 0,112 lower NRMSE compared with model that is using only MSE as loss metric. By comparing the developed LSTM network model with baseline ARIMA, fbProphet models, LSTM has an in average of 0,261 lower NRMSE as well as in average of 0,083 higher correlation, substantiating the significance by Student's test. The obtained results are comparable with the results of 3 published articles, but lag behind compared to other 4.