



VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
STUDIJŲ PROGRAMA: INFORMATIKA

Žmonių srauto prognozavimas viešame transporte naudojant DI metodus bei autobusų tvarkaraščio ir dydžio planavimas

People Traffic Forecasting in Public Transport Using AI Methods and Bus Schedule and Size Planning

Baigiamasis magistro darbas

Atliko: Paulius Grigaliūnas
VU el. p.: paulius.grigaliunas@mif.stud.vu.lt

Darbo vadovas: Prof. dr. Aistis Raudys

Recenzentas: Asist. dr. Gintaras Skersys

Vilnius
2022

Turinys

1. Įvadas	3
2. Darbo Tikslas	5
2.1. Darbo Uždaviniai	5
3. Tyrimo objektas	6
4. Matematiniai-statistiniai prognozuojamieji metodai	7
4.1. Auto-regresija (AR)	7
4.2. Slankusis vidurkis (MA)	7
4.3. Auto-regresyvus integruotas slenkamasis vidurkio modelis (ARIMA)	8
4.4. Palaikomasis vektoriaus regresavimas (SVR)	8
5. Prognozuojamieji neuroniniai tinklai	9
5.1. Paprastieji viena-krypčiai neuroniniai tinklai	9
5.2. Abipusio sklidimo neuroniniai tinklai	10
5.2.1. Gradiento nusileidimo metodas	10
5.2.2. Stochastinis gradiento nusileidimo metodas	12
5.3. Rekurentiniai neuroniniai tinklai (RNN)	12
5.3.1. Primityvieji rekurentiniai neuroniniai tinklai	12
5.3.2. Ilgalaikė trumpalaikės atminties tinklas (LSTM)	13
5.3.3. Sulaikomo pasikartojančio vieneto (GRU)	15
6. Specifiniai modeliai skirti žmonių srauto prognozavimui	18
6.1. Žmonių srauto prognozavimas traukinių sistemoje	18
6.1.1. Populiariausių prognozuojamųjų neuroninių tinklų tikslumo palyginimas	18
6.1.2. LSTM dviejų būsenų treniravimo modelis	18
6.2. Neuroninio tinklo apmokymas	20
6.3. Seq2seq modelis	20
7. Literatūros apžvalgos apibendrinimas	23
8. Viešojo transporto tvarkaraščio rengimas	24
8.1. Autobusų kiekio paskirstymas	24
8.2. Transporto priemonių didžio bei komforto pasirinkimas	24
8.3. Žmonių srauto prognozavimas VT sistemoje	25
8.4. Viešojo transporto srauto optimizavimas	25
8.4.1. VT tinklo maršrutų projektavimas	25
9. Testinių duomenų atrinkimas	27
9.1. VT judėjimo duomenys	27
9.2. Surinkti duomenis turi turėti papildomų etikečių	29
9.3. Autobusų charakteristikos	30
9.4. Oro sąlygos	30
9.5. Covid-19 statistika	31
9.6. Apskaičiuoti keleivių skaičių VT priemonėje	32
9.7. Duomenų grupavimas	32
9.8. Duomenų sujungimas	32
9.9. Duomenų normalizacija	33
10. Servisai ir įrankiai	34
10.1. Tensorflow	34
10.2. Keras	34
10.3. Google Colab	34
11. Tyrimo metodika	36
11.1. Maršruto keleivių prognozavimas metų dienų atžvilgiu	36
11.2. Maršruto keleivių prognozavimas valandos intervale	37

11.3. Maršruto keleivių prognozavimas mėnesių atžvilgiu	37
11.4. LSTM neuroninio tinklo taikymas būsimų keleivių srautams nustatyti	37
11.5. Statisinių-matematinių prognozavimo modeliai	38
11.5.1. ARIMA maksimalios paros keleivių skaičiaus prognozės	39
11.5.2. SVR maksimalios paros keleivių skaičiaus prognozės	39
11.6. Neuroninių tinklų prognozavimo modeliai	40
11.6.1. Abipusio LSTM neuroninio tinklo taikymas nuspėti maksimalų paros keleivių skaičių	40
11.6.2. Abipusio GRU neuroninio tinklo taikymas nuspėti maksimalų paros keleivių skaičių	42
11.6.3. Abipusio RNN tinklo taikymas nuspėti maksimalų paros keleivių skaičių	43
11.6.4. Abipusio LSTM neuroninio tinklo taikymas nuspėti maksimalų valandinį kelei- vių kiekį	44
11.6.5. Abipusio GRU neuroninio tinklo taikymas nuspėti maksimalų valandinį keleivių kiekį	46
11.6.6. Abipusio RNN neuroninio tinklo taikymas nuspėti maksimalų valandinį keleivių kiekį	47
11.7. Matematinis-statistinis neuroninių tinklų prognozuojamasis modelis	48
11.7.1. SVR-LSTM modelis	48
11.7.2. Kiti modeliai	49
12. Rezultatai	51
13. Išvados	52
Literatūra	53
Santrumpos	55

1. Įvadas

Viešasis transportas (toliau VT) yra patogus, ekonomiškas, ekologiškas ir gana jau įprastas būdas keliauti. Galime pasidžiaugti, jog Lietuvos VT tinklas yra išsibarstęs po visą šalį. VT leidžia žmonėms keliauti miestuose bei už jų ribų. Nors teoriškai visuotinė VT nauda yra ištirta įvairiais aspektais, plačiai aprašyta ir gerai žinoma, tačiau net ir išsivysčiusiose šalyse susiduriama su įvairiomis VT paklausos ir kontrolės problemomis:

- netinkamas VT srautų planavimas (perkrautos arba pustuštės VT priemonės),
- žemos iš VT paslaugų gaunamos pajamos,
- netinkamai išdėstytos autobusų stotelės,
- nukrypimai nuo eismo tvarkaraščių,
- nepakankama VT priemonių kokybė,
- spūstys, kurios didina kuro suvartojimą ir greitina VT priemonių nusidėvėjimą,
- didelės bendrosios sąnaudos, neužtikrinamas finansavimas,
- nepakankamas gyventojų pasitikėjimas daugiausia dėl nepatogumo ir saugumo [S4I14].

Kadangi gyvenamame laisvoje, judrioje ir besikeičiančioje visuomenėje, keleivių įnoriai yra dinamiški ir kartais gali nutikti taip, kad nusistovėję VT maršrutai bei jų laikai, gali tapti nepatogūs. Tai reiškia, jog yra būtina vykdyti nuolatinę VT sektoriaus stebėjimus ir analizuoti pokyčius. Analizė turi apimti šiuos veiksnius:

- esamos situacijos apibūdinimas, apimant tiek VT sektorių, tiek kitus veiksnius, kurie daro įtaką VT sektoriui,
- būtinų pokyčių apibrėžimas / pageidaujamos situacijos apibūdinimas,
- veiksmų / priemonių pageidaujamai būsenai pasiekti nustatymas,
- atliktų veiksmų, įgyvendintų priemonių efektyvumo vertinimas[S4I14].

Kad VT taptų patrauklesnis keliavimo būdas yra įkuriamos įmonės kurios užsiima VT stebėjimo veikla, nustato žmonių srautų pasitelkiant prognozavimo statistinius modelius bei VT infrastruktūros gerinimo darbus. Šiame darbe bus koncentruojamasi į žmonių srauto prognozavimo dalį. Šiai problemai spręsti bus naudojami ne tik statistiniai prognozavimo modeliai bet ir prognozavimo modeliai kurie prognozėms pasitelkia dirbtinio intelekto metodus.

Normaliomis sąlygomis (iki pandemijos) didžiausia problema buvo reti autobusai piko metu bei nepatogus VT išvykimo tvarkaraščiai. Covid-19 pandemijos metu kilo didelė grėsmė žmonių sveikatai. Buvo įvesti įvairūs apribojimai siekiant užtikrinti saugias sąlygas: tai yra užtikrinti saugius tarp asmenų 2 metrų atstumus, nustatyti ir reguliuoti saugų žmonių kiekį patalpose, taip pat

ir VT priemonėse [MIN20]. Svarbu pastebėti kokią įtaką turėjo pandemija VT keleivių skaičiui ir kaip keleivių skaičius keitėsi kintant sergančiųjų skaičiui.

Pastaruoju metu lankstus VT srauto reguliavimas įgavo didelę svarbą. Dabar yra itin sunku sudaryti optimalų ir besikeičiantį VT tvarkaraštį atsižvelgiant į itin greitai besikeičiančias aplinkos sąlygas bei keleivių norus. Yra paskaičiuota, jog pastarosios karantino bangos metu keleivių srautą sudarė tik 20-30 % srauto, kuris būna normaliomis sąlygomis. Vežimo įmonės dirbo nuostolingai [Vai20]. Nors žmonės galimai yra įpratę prie statinių tvarkaraščių, t.y. tokių kurių grafikas keičiasi retai, kartą į pusmetį ar panašiai, bet dabar yra puiki galimybė pritaikyti lanksčius tvarkaraščius. Atsižvelgiant į tai, jog vis populiarėja VT programėlės, kurių pagalba būtų galima palaikyti VT tvarkaraščio veikimą. Lankstūs VT tvarkaraščiai užtikrintų patogų ir abipusiškai ekonomišką susiekimą gyventojams.

2. Darbo Tikslas

Pritaikius prognozuojamuosius neuroninių tinklų modelius apskaičiuoti būsimus maršruto keleivių srautus ir nustatyti kuris modelis keleivių prognozavimui tinka geriausiai.

2.1. Darbo Uždaviniai

- Atlikti Vilniaus miesto VT duomenų analizę. Duomenis segmentuoti pagal dienos parametrus, oro sąlygas, covid-19 sergamumo statistika.
- Panagrinėti statistinių prognozavimo modelių veikimą kaip: ARIMA bei SVR, prognozuojant keleivių srautus.
- Pritaikyti ir adaptuoti įvairius prognozuojamųjų neuroninių tinklų modelius kaip: LSTM, GRU, SVR-LSTM, seq2seq modelius, kurie gebėtų prognozuoti keleivių srautus.
- Sukurti modelį kuris gebėtų gana tiksliai prognozuoti būsimus keleivių srautus dienų atžvilgiu. (paklaidos LOSS funkcijos reikšmė neviršytų 10%)
- Sukurti modelį kuris gebėtų gana tiksliai prognozuoti būsimus keleivių srautus valandų atžvilgiu. (paklaidos LOSS funkcijos reikšmė neviršytų 10%)
- Apmokyti ir panaudoti prognozuojamuosius modelius keliems maršrutams.
- Nustatyti optimalų maršruto autobusų kiekį.

3. Tyrimo objektas

Tyrimo objektas: Dirbtinio intelekto prognozuojamasis modelis su statistinėmis analizės operacijomis, kuris gebėtų prognozuoti rytojaus viešojo transporto keleivių srautus.

Dirbtinio intelekto prognozuojamasis modelis bus kuriamas remiantis šiais statistinio prognozavimo modeliais:

- Autoregressive integrated moving average (ARIMA).
- Support Vector Regression (SVR).

Bei šiais prognozuojamaisiais neuroninio tinklo modeliais:

- Recurrent Neural Networks (RNN).
- Long Short-Term Memory (LSTM).
- Gated Recurrent Unit (GRU).

Sukurti modeliai gali pasižymėti tokiomis savybėmis:

- Feed-Forward Neural Network (FFNN).
- Back-Forward Propagation Network (BPN).

Dirbtinio intelekto modelio apmokymas ir pritaikymas bus vykdomas pasitelkiant realius Vilniau miesto viešojo transporto keleivių srautų duomenis. Pirmiausiai duomenys bus apdorojami tai yra

- Atrenkami VT parametrai: autobuso numeris, stotelė, kryptis, valanda, savaitės diena, kiek keleivių įlipo, kiek išlipo.
- Atrenkami oro sąlygų parametrai: saulėtumas, drėgnumas, oro temperatūra.
- Atrenkami covid-19 sergamumo parametrai: susirgusiųjų skaičius, mirusiųjų skaičius karantino dieną.

4. Matematiniai-statistiniai prognozuojamieji metodai

Matematiniai-statistiniai prognozuojamieji metodai yra bene populiariausias būdas prognozuoti ateities įvykius. Kad būtų galima sudaryti prognozes reikia turėti:

- Apdirbtus statistinius duomenis iš tam tikro laiko intervalo.
- Pritaikyti atitinkamą matematinį-statistinį prognozuojamąjį modelį.

4.1. Auto-regresija (AR)

Auto-regresija - tai vienas iš laiko eilutės (angl. "time series") analizės metodų. Šis metodas prognozuoja būsimumus rezultatus atsižvelgiant į buvusias reikšmes padauginus tam tikrus koeficientus. Nesunku pastebėti, kad paprastai laiko eilutės reikšmės stipriai koreliuoja su prieš jas bei po jų einančiomis tos pačios laiko eilutės reikšmėmis. Tuo remiantis sudaromi auto-regresijos lygtys - modeliai:

Pirmos eilės auto-regresijos lygtis:

$$Z_i = \alpha + \phi_1 \cdot Z_{t-1} + \varepsilon_t; \quad (1)$$

Antros eilės auto-regresijos lygtis:

$$Z_i = \alpha + \phi_1 \cdot Z_{t-1} + \phi_2 \cdot Z_{t-2} + \varepsilon_t; \quad (2)$$

p-os eilės auto-regresijos lygtis:

$$Z_i = \alpha + \theta_1 \cdot Z_{i-1} + \theta_2 \cdot Z_{i-2} + \dots + \theta_p \cdot Z_{i-t} + \varepsilon_t; \quad (3)$$

čia: $Z_t, Z_{t-1}, \dots, Z_{t-p}$ - laiko eilutės reikšmės laiko momentais $t, t-1, \dots, t-p$. α - laisvasis narys, $\phi_1, \phi_2, \dots, \phi_p$ - koeficientai, apskaičiuojami mažiausiųjų kvadratų (regresijos) metodu, ε_t - nekoreliuojanti atsitiktinė dedamoji (paklaida) su nuliniu vidurkiu. Pirmosios eilės auto-regresijos lygtis nagrinėja ryšius tik tarp gretimų laiko eilutės narių, Antrosios eilės - per tris gretimus laiko momentus atitinkančių laiko eilutės reikšmių ir t.t.

4.2. Slankusis vidurkis (MA)

Slankaus vidurkio (angl. *Moving average*) modelis taip pat yra naudojamas laiko eilutės prognozėms atlikti. Kitaip nei auto-regresijos modelis prognozes atliekamos ne pagal buvusias reikšmes, o pagal spėjimo santykius su realiosiomis reikšmėmis. Modelio formulė atrodo taip:

$$X_i = \alpha + \varepsilon_t + \phi_1 \cdot \varepsilon_{t-1} + \dots + \phi_q \cdot \varepsilon_{t-q}; \quad (4)$$

α yra eilutės vidurkis, $\varepsilon_t, \varepsilon_{t-1}, \dots, \varepsilon_{t-q}$ yra prognozės bei faktinės reikšmės nuokrypis, ϕ_1, \dots, ϕ_q yra modelio nuokrypių svoriai.

4.3. Auto-regresyvus integruotas slenkamasis vidurkio modelis (ARIMA)

ARIMA modelis yra auto-regresijos (AR) modelio ir slenkamo vidurkio (MA) modelio kombinacija su Integruotu (I) skirtumo parametru. I („integruotas“) rodo, kad duomenų reikšmės buvo pakeistos esamų įverčių ir ankstesnių įverčių skirtumais (ir šis diferencijavimo procesas galėjo būti atliktas daugiau nei vieną kartą). Kadangi Integralumas leidžia tirti ne pavienes reikšmes, o jų vidurkį tai suteikia didesnę pranašumą atliekant analizę su staigiais duomenų reikšmių nuokrypiais. ARMA ir ARIMA modeliai atrodo taip:

$$ARMA(p,q) = Z_p + X_q \quad (5)$$

$$ARIMA(p,d,q) = \alpha + \sum_{i=1}^p \phi \cdot y_{t-i} + \sum_{i=1}^P \Phi \cdot y_{t-is} + \varepsilon_t - \sum_{i=1}^q \theta \cdot \varepsilon_{t-i} + \sum_{i=1}^Q \Theta \cdot \varepsilon_{t-is} \quad (6)$$

p , d , q parametrai yra kintamieji. Parametras p nurodo į kelius įrašus iš AR įtraukti į skaičiavimą, q kelias įrašus iš MA įtraukti į skaičiavimą ir d nurodo integralumo laipsnį (kiek kartų buvo atliekamas skirtumas tarp duomenų įverčių). Kiekvienos iš šių funkcijų tikslas – kad bendras modelis kuo geriau gebėtų nuspėti būsimus duomenis [Zha03].

4.4. Palaikomasis vektorius regresavimas (SVR)

SVR yra populiarus statistinis algoritmas kuris leidžia spręsti klasifikavimo problemą. Algoritmas klasifikuoja naujus duomenis pagal tai kokius klasifikuotus duomenis jau turime. Naujas elementas klasifikuojamas atsižvelgiant į trumpiausią vektorinį atstumą nuo atitinkamos klasifikuotos grupės išskirtų charakteristikų vidurkio. Pavyzdžiui, jei bandome atskirti obuolius nuo apelsinų, tai jei naujo daikto viena iš charakteristikų - spalva yra oranžinė, ta greičiausiai tai bus apelsinas. Paprasčiausiojo linijinio SVR klasifikavimo funkcija pateikta čia (žr. 7).

$$MIN \sum_{i=1}^n (y_i - w_i \cdot x_i)^2 \quad (7)$$

y_i yra naujojo elemento ypatybės skaitinė reikšmė, w_i yra ypatybės svarbos koeficientas, o x_i yra prognozuojamos grupės narių bendroji ypatybės reikšmė.

5. Prognozuojamieji neuroniniai tinklai

Norint sukurti neuroninį tinklą kuris gebėtų prognozuoti, reikia išsiaiškinti kokie neuroniniai tinklai egzistuoja, kokios yra jų panaudojimo galimybės, privalumai bei trūkumai. Taip pat išskirti esminius principus kaip neuroninis tinklas apdoroja duomenis, prisitaiko bei atsimena. Atmintis prognozuojamiesiems neuroniniams tinklas yra itin svarbi, nes jei individas geba atsiminti daug specifinės srities įvykių tai jis geba ir tiksliau nuspėti ateities įvykius [AWS07]. Tolesniuose poskyriuose yra pateikti ir aprašyti keli pagrindiniai neuroninių tinklų modeliai, kurie geba kaupti informaciją t. y. atmintį, paskutiniuose poskyriuose yra išanalizuoti sudėtingos neuroninių tinklų architektūros, kurios sprendžia panašias problemas kaip šio darbo.

5.1. Paprastieji viena-krypčiai neuroniniai tinklai

Dirbtinis neuronas (angl. '*artificial neuron*') atsiradimo pradžia laikoma 1943 metais kai ne-urumokslininkas S. McCulloch ir logikas W.Pitts sukūrė pirmąjį konceptualų dirbtinio neuroninio tinklo modelį kurį pateikė moksliniame straipsnyje "*A logical calculus of the ideas imminent in nervous activity*") Mokslininkai matematinėmis formulėmis aprašė biologinių neuronų veiklą. Dirbtinis neuronas tai yra funkcija kuri priima daug skaitinių įvesčių su kuriomis atlieka tam tikrus skaičiavimus ir pateikia teigimą arba neigiamą išvestį [SCR⁺90]. Sujungus tokius neuronus į didelį tinklą turėtų būti galima sukurti „Elektronines smegenis“ (angl. „*Electronic Brain*“), kurios gebėtų „mąstyti“ ir priimti sprendimus. Spręsti prognozavimo uždavinį su vienu neuronu yra neįmanoma, nes šis modelis nei atlieka mokymosi proceso tai yra neatlieka duodamų parametrų svorių (svarbumo) koregavimo proceso pagal testinius duomenis, nei neišsaugo jokių praeitų įvesčių.

20 amžiaus šeštajame dešimtmetyje buvo sukurtas mašininio mokymosi algoritmas – Perceptronas (angl. Perceptron), autorius – F. Rosenblatt, 1957. Šio neuroninio tinklo atsiradimas siejamas su antrąją dirbtinio intelekto kūrimo bangos pradžia [GBC16]. Tai tapo pirmuoju modeliu, galinčiu atlikti neuronų apmokymą, tai yra apskaičiuoti svorius, apibrėžiančius kategorijas. Perceptronas modelis yra sudarytas iš vieno neuronų sluoksnio, bet tame sluoksnyje yra išdėstyti keletas ar keliasdešimt neuronų kurie yra atsakingi už tam tikras sritis. O svarbiausia, jog modelis gebėjo atlikti neuronų svorių koregavimo veiksmus [Ros58].

Vėliau nuo 1980 metų pasirodė vis daugiau naujų daugiasluoksnių neuroninių tinklų. Matematikai pradėjo eksperimentuoti su neuronų grupėmis pradėjo juos sujungti į grandines. Buvo pastebėta, jog atliekami skaičiavimai tampa intelektualūs (nenuspėjami) kai neuronai sąveikauja vienas su kitu. Taip pradėjo rasti sudėtingesnės Vienakrypčiai neuroniniai tinklai (FFNN) (angl. *Feed-forward neural network (FFNN)*) architektūros[GBC16]. Tai apibūdina tokius neuroninių tinklų grupę kurių neuronai sujungti nuosekliai nėra jokių ciklinių ryšių. FFNN informacija juda tik viena kryptimi - nuo įvesties sluoksnio per paslėptus sluoksnius į išvesties sluoksnį. Informacija juda tiesiai per tinklą ir niekada neliečia mazgo du kartus. FFNN tinklas gali būti pritaikytas atlikti pasirinktos srities prognozavimui tik tuo atveju, jei visi neuronų svoriai yra parinkti tinkamai antraip funkcijos rezultatai bus netikslūs.

5.2. Abipusio sklaidimo neuroniniai tinklai

Įprastai FFNN apmokyti yra taikomas atgalinis judėjimo (BPN) (*angl. Backpropagation*) metodas. Tuo metu yra atsižvelgiama į esamą neuroninio tinklo rezultatą jį pakeičia tikėtinas rezultatas. Veikimo algoritmas paleidžiamas atvirkštine tvarka taip, jog tikėtini rezultatai ir pradinės įvestys išliktų stabilios, o kistų tik neuronų funkcijų svoriai. Atgalinio judėjimo algoritmas buvo sukurtas 1986 metais mokslininkų Rumelhart, Hinton ir Williams [RHW86].

Abipusis sklaidimo neuroninis tinklas FFNN kuris tinklo apmokymui naudoja atgalinio judėjimo algoritmą (BPTT) (*angl. Backpropagation through time*). Tokio tipo neuroninis tinklas geba atlikti prognozavimą. Yra sukurta architektūrų remiantis abipusio sklaidimo modeliu, kurie geba atlikti orų prognozavimą [NP12] arba upės tėkmės prognozavimą [SNO02]. Abu tyrimai buvo atlikti remiantis daugiasluoksniu FFNT su BPN. Tyrimai parodė, kad protingai sujungtos prognozės gali būti žymiai patikimesnės tos kurios orientuojasi į kuo didesnę ir įvairesnę grupę parametrų nei tie modeliai kurie orientuojasi į vieną parametą ar mažą jų grupę, pavyzdžiui upės srautui nustatyti nustatyti pasitelkiant modelį kurio įvestis tik atskirų kritulių kiekis [SNO02].

5.2.1. Gradiento nusileidimo metodas

Nors prognozuojamųjų neuroniniai tinklų architektūros nuolat tobulėja, jų mokymo metodai iš esmės liko tie patys [LZ14]. Įvairius neuroninius tinklus moko pasitelkiant Gradiento nusileidimo principą (*angl. Gradient Descent (GD)*). Šis metodas padeda neuroninio tinklo funkcijų konstantų reikšmes. Gradiento nusileidimo metodas yra taikomas tuomet kai nėra aiškiai žinomas grafikas, žinoma tik tam tikra grafiko taškų aibė, o norima surasti grafiko ekstremumo (maksimumo arba minimumo) tašką. Deja šis metodas nėra deterministinis, jis padeda atrasti lokaliuosius grafiko ekstremumo taškus. Tai yra ekstremumo taškus specifiniame grafiko intervale.

Neuroninio tinklo apmokymo metu Gradiento nusileidimo metodas yra naudojamas siekiant surasti minimalią prognozuotų reikšmių ir realiųjų reikšmių skirtumų sumą, arba pasitelkiant kurią kitą paklaidų skaičiavimo formulę. Ši funkcija žinoma kaip nuostolių (*angl. loss*) pavadinimu. Populiariausios paklaidų, kitaip vadinamos nuokrypio, skaičiavimo funkcijos:

- Absoliučioji nuokrypio funkcija (*angl. Absolute Error*) (AE). n - visos iteracijos, i - konkreiti iteracija, $y - i$ - reali reikšmė, $\hat{y} - i$ - prognozuota reikšmė.:

$$AE = \sum_{i=1}^n |y_i - \hat{y}_i| \quad (8)$$

- Vidutinė absoliutinė paklaida (*angl. Mean Absolute Error*) (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (9)$$

- Vidutinės absoliutinės procentinės nuokrypio funkcija (*angl. The Mean Absolute Percentage*

Error) (MAPE).

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_i} \times 100\% \quad (10)$$

- Kvadratinio nuokrypio funkcija (angl. *Squared Error* (SE)):

$$SE = \sum_{i=1}^n |y_i - \hat{y}_i|^2 \quad (11)$$

- Kvadratinio vidurkio nuokrypio funkcija (angl. *Mean Squared Error* (MSE)):

$$MSE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|^2 \quad (12)$$

- Nuokrypio kvadratų kvadratinės šaknies funkcija (angl. *Root Square Error*) (RSE):

$$RSE = \sqrt{\sum_{i=1}^n |y_i - \hat{y}_i|^2} \quad (13)$$

- Vidutinė nuokrypio kvadratų kvadratinės šaknies funkcija (angl. *Root Mean Square Error*) (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^n |y_i - \hat{y}_i|^2}{n}} \quad (14)$$

Neuroninio tinklo apmokymo tikslas yra nustatyti kuo tikslesnius neuronų funkcijų svorius. Paprastai neuronų funkcijos atrodo taip:

$$\hat{y} = b + k \times x \quad (15)$$

\hat{y} - naujos reikšmės prognozė, b - reikšmė kur grafikas kerta oy ašį, k - grafiko nuolydis, x - atstumas ox ašyje. Mokymosi metu būna į funkciją x reikšmė atrinkta iš realių duomenų ir įstatomos kintamos mokymo metu esančios reikšmės b (interceptorius) ir k (nuolydžio koeficientas). Gauta \hat{y} reikšmė yra palyginama su realiąja y reikšme pasitelkiant vieną iš nuostolių funkcijų. Skaičiavimai atliekami su visais testiniais duomenimis. Siekiant atrasti tikslesnius b ir k koeficientus yra apskaičiuojama pasirinktosios nuostolių funkcijos išvestinė su pagrindu b ir tos pačios nuostolių funkcijos išvestinė su pagrindu k . Gautosios reikšmės yra sudauginamos su žingsniu γ , kuris būna sąlyginai mažas dydis, pavyzdžiui 0.01. Formulė kitiems b ir k koeficientams rasti atrodo taip ($\nabla F(x)$ reiškia tą patį kaip \hat{y} išvestinė):

$$k_{n+1} = k_n - \gamma \nabla F(k_n) \quad (16)$$

$$b_{n+1} = b_n - \gamma \nabla F(b_n) \quad (17)$$

Naujosios b ir k reikšmės įstatomos į funkciją 15 ir kartojami skaičiavimai. Ciklas sukasi tol

kol baigiasi nustatytas maksimalus iteracijų skaičius paprastai tai būna apie 1000 arba iki tol kol visų kintamųjų koeficientų pokytis pasidaro labai mažas, artimas 0 ($k_n - k_{n+1} < 0.0001$).

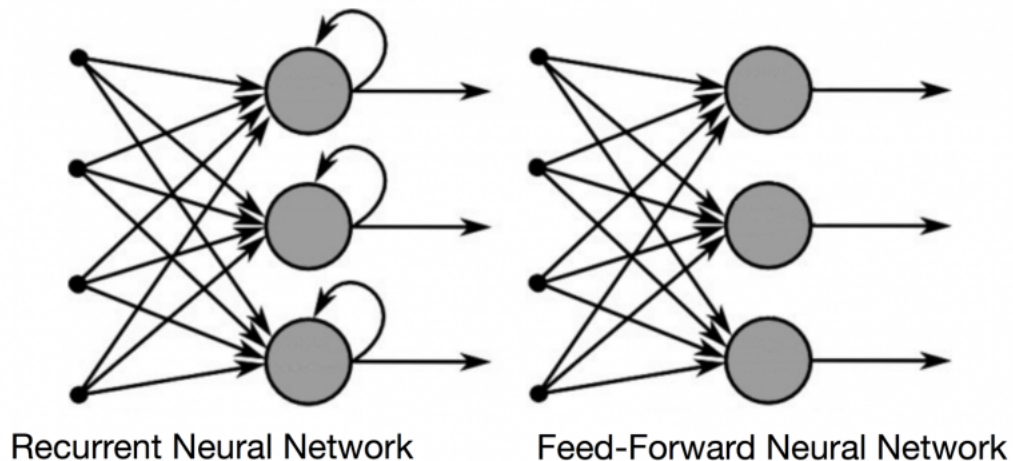
Palyginus visus optimizatorius buvo išsiaiškinta, jog kvadratinio vidurkio nuokrypio funkcija (MSE) išlaiko gerą paprastumo naudoti ir veiksmingumo santykį. MSE yra puiki metrika optimizavimo kontekste. MSE pasižymi labai maloniomis išgaubtumo, simetrijos ir diferencijavimo savybėmis, kas leidžia ieškoti minimumo arba maksimumo taškų. ??

5.2.2. Stochastinis gradiento nusileidimo metodas

Gradiento nusileidimo metodas gali būti skaičiavimo požiūriu labai brangi operacija, kadangi jai atlikti kiekvienoje iteracijoje yra naudojama visi testiniai duomenys. Optimizuotas sprendimas yra Stochastinis gradiento nusileidimo metodas (angl. *Stochastic Gradient Descent (SGD)*). Jo veikimo principas yra labai panašus į paprastojo GD, skiriasi tik tuo, jog kiekvienai iteracijai yra atrinkti keletas ne visi testiniai duomenis. Šis pokytis padaro apmokymo procesą greitesnį ir sumažina persimokymo tikimybę. Įrodyta atliekant bandymus [LZ14].

5.3. Rekurentiniai neuroniniai tinklai (RNN)

Amerikietis, filosofas David Rumelhart pristatė Rekurentinį neuroninį tinklą (angl. *Recurrent Neural Networks (RNN)*). RNN grupė išsiskiria iš FFNN tuo, jog turi ciklinę priklausomybę tarp tinklo neuronų (1 pav.). Tai leidžia tam pačiam neuronui ar neuronų grandinei gauti praeitos iteracijos rezultatą kaip įvestį, kuri vėliau įtakoja to paties neurono ar grupės būsimą rezultatą.

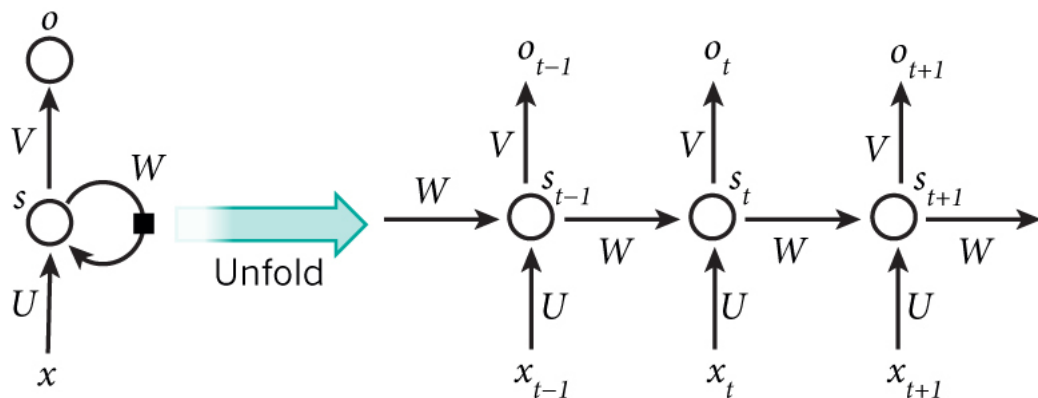


1 pav. rekurentinio neuroninio tinklo schema

5.3.1. Primityvieji rekurentiniai neuroniniai tinklai

Primityvieji RNN dažniausiai būna naudojamas atpažinti ar prognozuoti garsus ar parašytus žodžius. RNN modelio išvyniotą schemą yra pateikta žemiau (2 pav.) Paprasčiausias RNN veikimas principas:

1. Kiekvieną iteraciją pavadinkime bloką, į kurį patenka įvesties (angl. *input*), būsenos (angl. *state*) vektoriai - x , s , ir išvesties (angl. *output*) vektorius - o .
2. U, V, W reprezentuoja svorius kurie yra nekintančios konstantos algoritmo veikimo metu.
3. x_t tai žymi įvesties vektorių t laiko momentu.
4. s_t tai yra paslėptosios būsenos vektorius (atmintis) t laiko momentu. s_t yra apskaičiuojama pagal prieš tai buvusią būseną s_{t-1} ir dabartinio žingsnio įvestį x_t , formulė: $s_t = f(Ux_t + Ws_{t-1})$. f dažnai yra netiesiška funkcija dažniausiai tai būna \tanh . O pirmojoje iteracijoje kai nėra jokio prieš tai buvusios būsenos vektoriaus, s_{t-1} kai $t = 0$, tai s_t priskiriama nulinis vektorius.
5. \tanh funkcija normalizuoja įvesties vektorių reikmes, kad jos neišeitų iš intervalo $[-1; 1]$.
6. o_t yra t žingsnio išvestis keliauja į išvestį ir į sekantį bloką. Išvesties pavyzdys iš teksto apžinimo RNN, jei norėtume numatyti kitą sakinio žodį, tai vektorius turi būti mūsų žodyne su tam tikru nuokrypiu. Reikšmei normalizuoti yra pasitelkiama $o_t = \text{softmax}(Vs_t)$ [She20].



2 pav. rekurentinio neuroninio tinklo schema

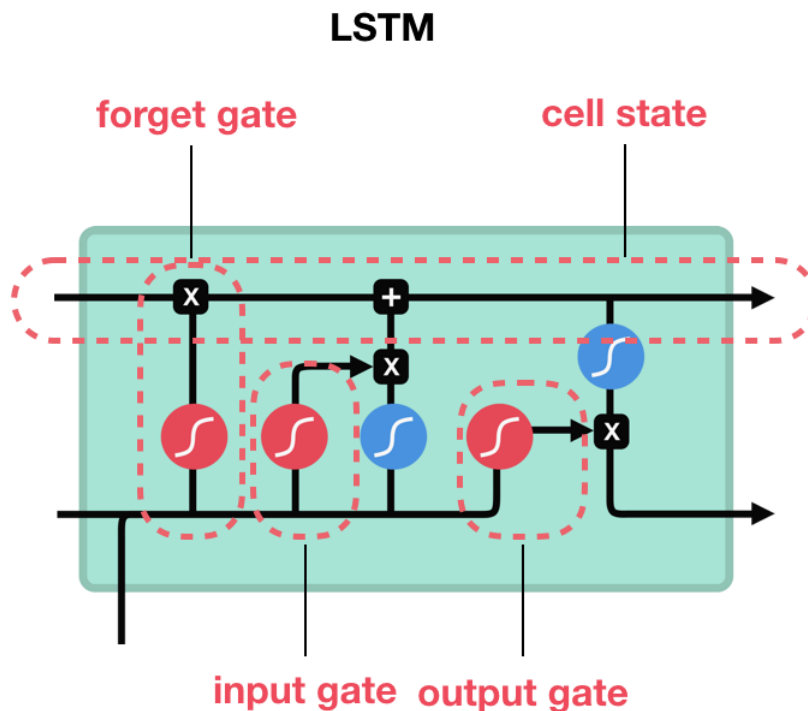
Paprastieji RNN pasižymi trumpa atmintimi. Atsimena tik vieną ar dvi praeitas įvestis. Taip pat RNN apmokymo metu yra būdinga nykstančio gradiento problema. RNN apmokymui paprastai naudojamas BPN. Nykstantys gradientai atsiranda, kai gradiento reikšmės yra pernelyg mažos ir modelis nustoja mokytis arba dėl to užtrunka per ilgai [Gro17; She20].

5.3.2. Ilgalaikė trumpalaikės atminties tinklas (LSTM)

Ilgalaikiai trumpalaikiai atminties tinklai (angl. *Long Short-Term Memory (LSTM)*) yra papildytas RNN modelis, kuris iš esmės išplečia modelio atmintį. Taip yra todėl, kad LSTM saugo informaciją atmintyje, panašiai kaip kompiuteris. LSTM gali skaityti, rašyti ir ištrinti informaciją iš savo atminties. LSTM blokas turi tris kanalus (angl. *gates*): pamiršimo, įvesties ir išvesties bei dvi blogo būsenos gijas: matomąją (angl. *cell state*) ir paslėptąją *hidden cell state*). Nauja informacija

yra padalijama į kanalus, kiekviename iš jų yra atliekami skirtingi skaičiavimai. Gautos išvesties vektoriai praleidžiami per normalizavimo σ (sigmoidinę) arba hiperbolinę funkcijas.

Iš pamiršimo ir įvesties kanalų gauti vektoriai yra dauginami ar sudedami su bloko būsenos gijos vektoriumi ir gauta reikšmė keliauja į sekantį LSTM bloką. Būsenos gyja ir reprezentuoja ilgalaikę modelio atmintį. Ilgainiui keliaujant informacijai tinklu yra išskiriamos tik svarbiausia informacija, o nesvarbi yra ištrinama. Žemiau yra LSTM iliustracija (3 pav.):



3 pav. ilgalaikiai trumpalaikiai atminties tinklai

1. Pirmiausia mes turime ištrinimo kanalą $f = \sigma(x_t U^f + s_{t-1} W^f)$. Algoritmas nusprendžia, kokia informacija turėtų būti išmesta ar saugoma. Informacija iš ankstesnės paslėptos būsenos s_{t-1} ir informacija iš dabartinės įvesties x_t yra perduodama per σ (sigmoidinę) funkciją. Vertės išeina tarp 0 ir 1. Kuo arčiau 0 reiškia pamiršti, o arčiau 1 reiškia išlaikyti.
2. Įvesties kanalas atskiria svarbią informaciją iš įvesties vektoriaus $i = \sigma(x_t U^i + s_{t-1} W^i)$. Pirmą, mes perduodame ankstesnę paslėptą būseną s_{t-1} ir dabartinę įvestį x_t į sigmoidinę σ funkciją. Tai nusprendžia, kurios reikšmės bus atnaujintos, pavertus reikšmes nuo 0 iki 1. 0 reiškia, kad nesvarbu, o 1 kai yra svarbu.
3. Įvesties kanalo pašonėje yra atskiras pokanalis piešinyje jis yra dešinėje įvesties kanalo. Jame yra atliekama operacija būsimam "kandidatui" g į busimą paslėptąją būseną gauti $g = \tanh(x_t U^g + s_{t-1} W^g)$.
4. Tuo tarpu būsenos gijoje praeito bloko būseną yra sudauginam su pamiršimo vektoriumi. Šis veiksmas dažnai sumažina bloko būsenos reikšmes. Noriu akcentuoti, jog kai atliekama vektorių daugyba vektorių reikšmės kurių vertės yra lygus nuliui paverčia iš kito dauginamojo

vektoriaus atitinkamas reikšmes nuliais, taip ir atliekamas informacijos trinimas. Tada pa-
imama išvestis iš įvesties kanalo ir atliekama sudėtis su nauja bloko būseną, kuri sustiprina
vyraujančių požymių egzistavimą. Taip sukuriama nauja bloko būseną. Šio žingsnio formulė
- $c_t = c_{t-1} \cdot f + g \cdot i$.

5. Galiausiai mes turime išvesties kanalą $o = \sigma(x_t U^o + s_{t-1} W^o)$. Išvesties kanalas nuspren-
džia, kokia kita paslėptoji būseną turėtų būti. išvesties kanalas apibrėžia, kiek vidinės būse-
nos bus perduota išoriniam tinklui (aukštesniems sluoksniams, kitai iteracijai).
6. Šią paslėptąją būseną giją vadinama trumpąją atmintimi (s), ji keliauja į kitą bloką kur ten
su kita įvestimi patenka į kito bloko kanalus. Į Išvesties kanalą yra perduodama ankstesnė
paslėptoji būseną ir dabartinė įvestis. Jos normalizuojamos per sigmoidinę funkciją. Tada
yra perduodama nauja modifikuota ląstelės atminties būseną c_t (paprasčia, nepaslėptoji) į tanh
funkciją ir sudauginama su išvesties kanalo vektoriumi o . Atliekama funkcija: $s_t = \tanh(c_t) \cdot$
 o . Galiausiai nauja bloko būseną ir paslėptoji bloko būseną keliauja į sekantį bloką.

Visos paminėtos LSTM formulės:

$$i = \sigma(x_t U^i + s_{t-1} W^i) \quad (18)$$

$$f = \sigma(x_t U^f + s_{t-1} W^f) \quad (19)$$

$$o = \sigma(x_t U^o + s_{t-1} W^o) \quad (20)$$

$$g = \tanh(x_t U^g + s_{t-1} W^g) \quad (21)$$

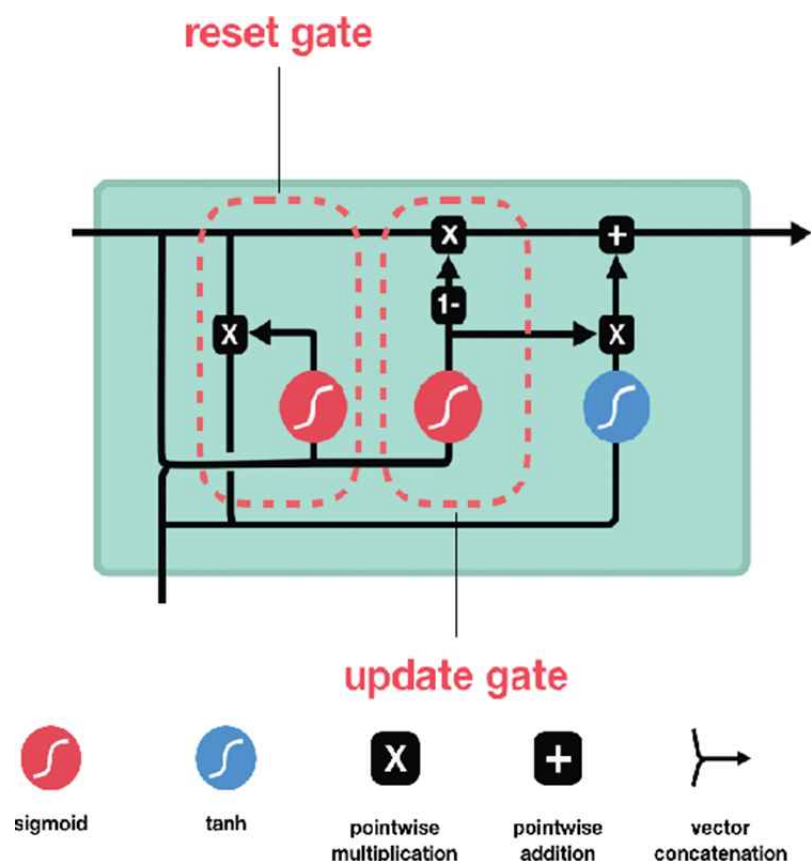
$$c_t = c_{t-1} \cdot f + g \cdot i \quad (22)$$

$$s_t = \tanh(c_t) \cdot o \quad (23)$$

Svarbios informacijos išskyrimą jei tinklo neuronų funkcijų svoriai yra tinkami. LSTM atlieka
mokymo procesą kurio metu būna suderinami kanalų funkcijų svoriai. Kadangi LSTM modelis
turi daug kanalų ir neuroninių tinklų bei šį modelį yra itin sudėtinga apmokyti [She20].

5.3.3. Sulaikomo pasikartojančio vieneto (GRU)

Sulaikomo pasikartojančio vieneto tinklas (angl. *Gated recurrent unit (GRU)*) yra gana naujas
modelis pasirodęs 2014 metais. Jis yra panašūs į LSTM, tačiau naudoja supaprastintą struktū-
rą. GRU taip pat naudoja kanalų rinkinį informacijos srautui valdyti jį sudaro atnaujinimo (angl.
Update Gate), perrinkimo (angl. *Reset Gate*) ir dabartinės atminties (angl. *Current Memory Gate*)
kanalai. Be to šis modelis neturi būsenos gijos, sukauptai atminčiai naudojama tik paslėptoji atmin-
ties gija *hidden cell state*). Detali modelio schema yra pavaizduota žemiau esančiame paveikslėlyje
(4 pav.):



4 pav. sulaikomo pasikartojančio vieneto tinklo schema

Į GRU bloką patenka: x_t - dabartinės informacijos vektorius ir s_{t-1} - praeitos iteracijos būsenos vektorius, o išeina naujas būsenos vektorius s_t . GRU veikimo žingsniai:

1. Atnaujinimo kanalas veikia panašiai kaip LSTM pamiršimo ir įvesties kanalas. Ji nusprendžia, kokią informaciją išmesti ir kokią naują informaciją pridėti. Formulė: $r = \sigma(x_t U^r + s_{t-1} W^r)$, r - atnaujinimo vektorius.
2. Perrinkimo kanalas yra skirtas nustatyti kiek senesnę informaciją reikia ištrinti. Formulė: $z = \sigma(x_t U^z + s_{t-1} W^z)$, z - perrinkimo vektorius.
3. Dabartinės atminties kanalas yra tiesiai susijęs su atnaujinimo kanalu. Kanalo atliekama funkcija $h = \tanh(x_t U^h + (s_{t-1} \cdot r) W^h)$. h - Dabartinės atminties vektorius.
4. Galu gale yra apskaičiuojama naujoji atminties būseną $s_t = (1 - z) \cdot h + c \cdot s_{t-1}$.

Visos paminėtos GRU formulės:

$$z = \sigma(x_t U^z + s_{t-1} W^z) \quad (24)$$

$$r = \sigma(x_t U^r + s_{t-1} W^r) \quad (25)$$

$$h = \tanh(x_t U^h + (s_{t-1} \cdot r) W^h) \quad (26)$$

$$s_t = (1 - z) \cdot h + c \cdot s_{t-1} \quad (27)$$

GRU turi mažiau kanalų ir neuronų operacijų nei LSTM, todėl tokį modelį yra šiek tiek greičiau apmokyti, bei skaičiavimai vykdymo metu trunka trumpiau nei LSTM. Bet atliekant realius tyrimus nėra nustatyta kuris modelis prognozavimą atlieką greičiau ir tiksliau, todėl yra taikomi abu modeliai [CGC⁺14].

6. Specifiniai modeliai skirti žmonių srauto prognozavimui

Atlikęs populiariausių prognozavimo dirbtinio intelekto tinklų (angl. *Artificial Neural Networks*) (*ANN*) analizę išsiaiškinau, jog geriausiai šį darbą atlieka tinklai kurių pagrindas yra LSTM arba GRU modelis. Atsižvelgdamas į tai, jog šio darbo tema susijusi su žmonių srauto prognozavimu viešajame transporte, tai ir ieškojau mokslinių darbų kurie būtų orientuoti į panašią problemą 2.

6.1. Žmonių srauto prognozavimas traukinių sistemoje

Kinijoje žmonių srautai yra labai dideli. Nors ir yra sukurta traukinių tinklų sistema, ją valdyti taip, kad būtų išvengta žmonių srautų spūsčių yra sudėtinga [SLZ⁺20]. Yra gausu mokslinių darbų kuriuose ši problema yra sprendžiama pasitelkiant prognozuojamuosius neuroninius tinklus.

6.1.1. Populiariausių prognozuojamųjų neuroninių tinklų tikslumo palyginimas

Moksliniame darbe - 'RNN-Based Subway Passenger Flow Rolling Prediction' [SLZ⁺20] prognozuoti būsimus keleivių srautams yra pasitelkiama BPN, LSTM ir GRU modeliai, siekiant palyginti ir išsiaiškinti kuris modelis prognozes atliks tiksliausiai. Pastebėti tyrimo bruožai:

- Pirmiausia yra surinkti duomenis apie buvusius keleivių srautus. Duomenys turi būti periodiški, pavyzdžiui, pateikti informaciją apie keleivių srautų situacija kas penkiolika minučių.
- Į medžiagą įtraukti kuo daugiau parametrų. Pavyzdžiui, parametrai gali būti šie: kiek žmonių įėjo į stotį, kiek išėjo iš stoties, kiek įlipo į traukinį, kiek iš jo išlipo, kokia savaitės diena, kokia oro temperatūra, koks drėgnumas, koks vėjo pūtimo greitis.
- Įtraukiamos tik toks valandos kuriomis vyksta traukinių judėjimas.
- Pradiniam tinklo apmokymui buvo pasirinkta viena konkreti stotis "Nanjing East Road Station in Shanghai" ir mėnesio trukmės laiko intervalas.

Tyrimas nustatė, jog tiksliausiai žmonių srauto prognozavimą atliko GRU modelis. "Palyginimas ir analizė patvirtina, kad GRU tinklas, kurio laiko tarpas yra 1,5 valandos, turi geriausią prognozavimo efektą ilgalaikiam (daugiau nei 3 valandų) prognozavimui, o GRU, kurio laiko tarpas yra 45 minutės, geriausias rezultatas yra trumpalaikis. termino prognozavimas"[SLZ⁺20]

6.1.2. LSTM dviejų būsenų treniravimo modelis

2019 metais Pekino Jiaotongo universiteto profesorė Jianyuan Guo su kitais kolegomis pristatė "Nenormalaus keleivių srauto prognozavimo modelį" [GXQ⁺19]. Šis modelis geba nuspėti staigius keleivių srautų padidėjimus, kurie jau viršytų nustatytąsias normas. Šiame tyrime pagrindinis tyrinamasis modelis yra SVR-LSTM. Palaikomasis vektorius regresavimas (SVR) (angl. *support vector regression*) yra dažniausiai naudojamas klasifikavimo uždaviniams spręsti. SVR šiuo atveju yra skirtas pagerinti neuroninio tinklo mokymosi procesą. SVR išskiria ribines erdves (angl.

hyperplane) į kurių režius patenkančios naujos įvestys yra priskiriamos tai grupei kuri vyrauja toje erdvėje [GXQ⁺19]. SVR modelio pritaikymas atfiltruoja staigius nuokrypius nuo tendencingų reikšmių.

Taip pat moksliniame straipsnyje yra aprašytas ypatingas LSTM dviejų būsenų treniravimo modelis (angl. *LSTM Two-stage training*). Prieš pat atliekant apmokymą duomenys yra perskirstomi į dvi aibes. Pirmoje aibėje yra duomenys iš senesnių laikų ir ši aibė vadiname "off-line", o kita aibėje, kuri vadinasi "on-line", yra saugomi patys naujausi duomenys. Apmokymas vyksta dvejais srautai atskirai vyksta mokymasis su "off-line" duomenimis ir "on-line" duomenimis ir galiausiai kai stotis užsidaro duomenys per-migruojami naujausi duomenis patenka į "on-line" aibę, o pasenę duomenys iš "on-line" keliauja į "off-line" aibę. Autorių teigimu, toks mokymo principas leidžia LSTM modelio veikimo metu tiksliau nuspėti trumpalaikius žmonių srauto pokyčius [GXQ⁺19].

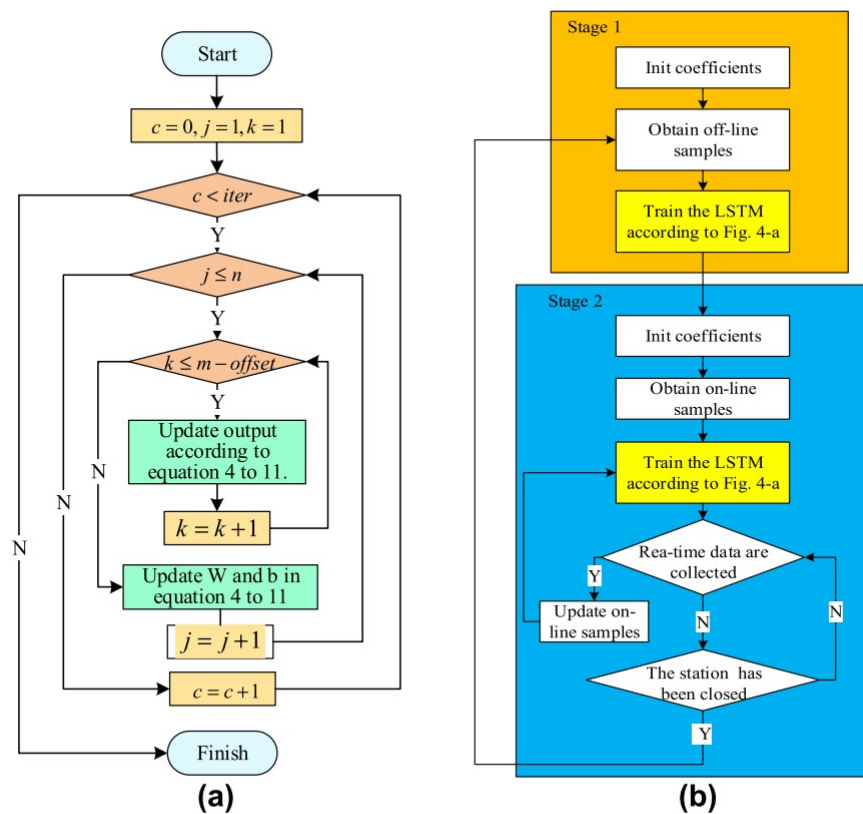


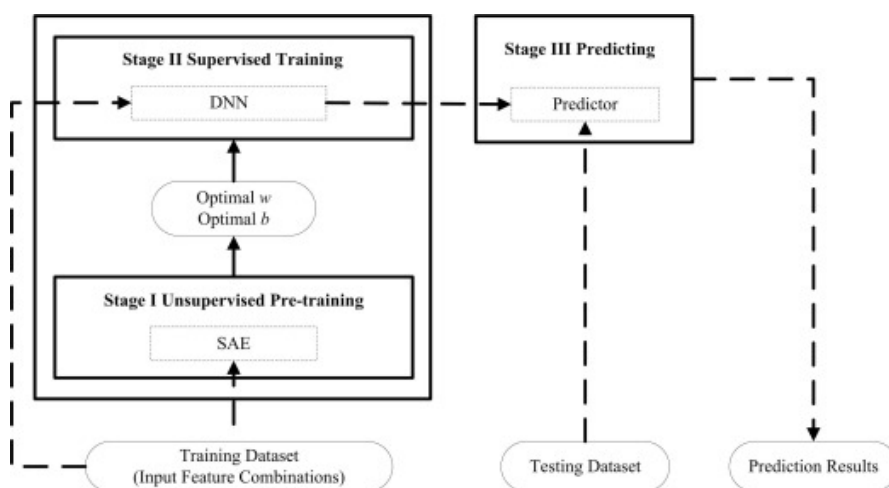
FIGURE 4. Training Flow of LSTM. (a) Basic training. (b) Two-stage training.

5 pav. LSTM dviejų būsenų treniravimo modelis [GXQ⁺19]

Tyrimo metu buvo naudojamas ne tik SVR-LSTM ir LSTM modeliai, bet ir ARIMA (angl. *Autoregressive Integrated Moving Average*, Fusion-KNN, SVR modifikuoti modeliai). To paskirtis buvo patikrinti, ar SVR-LSTM prognozuojamas tam tikru laiku žmonių kiekis yra labiau realus nei gaunama kitų modelių. Tyrėjai teigia, jog tiksliausiai žmonių srautus prognozuoja SVR-LSTM, argumentuoja tuo, jog paklaidos su realiais duomenimis (klaidų funkcijos reikšmės) buvo mažiausios [GXQ⁺19]. Paklaidos yra skaičiuojamos pasitelkiant MAE, MAPE, RMSE skaičiavimo funkcijas (9, 10 ir 14 funkcijos iš 5.2.1 poskyrio).

6.2. Neuroninio tinklo apmokymas

2017 metais profesoriai Lijuan Liu ir Rung-Ching Chen iš informacijos valdymo departamento, Chaoyang technologijų universiteto esančiame Taivane (angl. *Department of Information Management, Chaoyang University of Technology, Taiwan*) parengė "naująjį" keleivių srauto prognozavimo modelį [LC17]. Šis modelis ypatingas tuo, jog modelio apmokymą vykdomas ne tik prižiūrimuoju (angl. *supervised*) būdu bet ir savarankiškuoju (angl. *unsupervised*) būdu. Detalus modelis pateiktas žemiau (6 pav.):

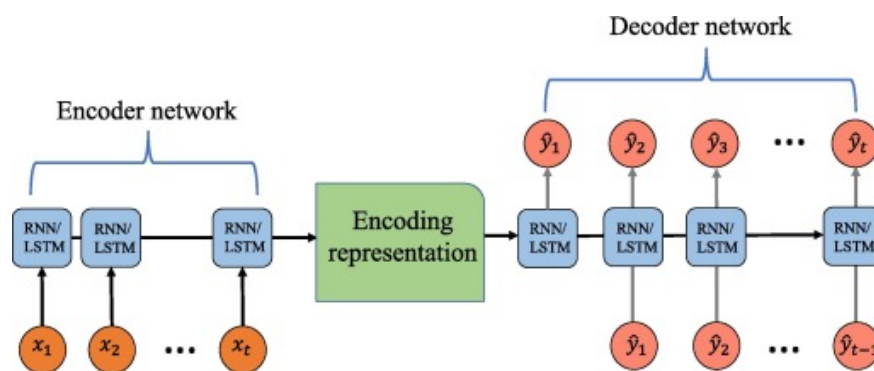


6 pav. LSTM dviejų būsenų treniravimo modelis [LC17]

Prižiūrimasis mokymasis tai yra įprastinis mokymas kai yra žinomas tikslus tikėtinas rezultatas. Savarankiškasis būdas - kai nėra aiškūs rezultatas. Savarankiškasis mokymasis nagrinėjamame modelyje yra sudarytas iš dviejų dalių: sukrautojo auto šifravimo (angl. *'stacked autoencoders'* (SAE)) ir prižiūrimojo mokymo. Sukrautasis auto šifravimas yra kelių sluoksnių neuronų tinklas kurio vidurinis sluoksnis yra sudarytas iš mažo kiekio neuronų. Taip keliant informacija tokiu neuroniniu tinklu "suspaudžiama", tai yra atrenkama tik svarbiausia informacija. Mokymo metu SAE tinklas yra treniruojamas pasitelkus realius duomenis. Autorių teigimu šis patobulinimas padeda išvengti persimokymo pavojaus, modelį padaro patikimesnį [LC17].

6.3. Seq2seq modelis

Ilgalaikėms keleivių srautų prognozavimui yra pritaikytas sekos į seką modelis [HLZ19]. Kitaip nei paprastieji laiko eilutės prognozuojamieji modeliai, kurie iš savybių sekos atlieka prognozes specifiniam vienetui, šis modelis turi apibrėžtą konkretų prognozuojamos sekos vienetų skaičių. Detali modelio struktūra pateikta žemiau (7 pav.):



7 pav. LSTM dviejų būsenų treniravimo modelis [HLZ19]

Kaip parodyta 7 pav. seq2seq modelis susideda iš dviejų RNN/LSTM tinklų: kodavimo tinklo (angl. *Encoder network*) kairėje pusėje ir šifravimo tinklo (angl. *decoder network*) dešinėje pusėje. Kodavimo tinklas suvokia ir koduoja įvesties seką į fiksuoto dydžio kontekstinį vektorinių (angl. *context vector*), paveikslėlyje tai pažymėta kaip šifravimo reprezentacija (angl. *Encoding representation*), kuri gaunama iš paskutinio kodavimo tinklo išvesties sluoksniu. Dešinėje esantis dekodavimo tinklas yra apibrėžto dydžio seka kuri priima sekos parametrus su reikšmėmis kaip įvestis į RNN ar LSTM tinklą. Grandinė šių tinklų sukuria šifravimo vektorius, kuris pažymėtas paveikslėlyje (žr. 7 pav.) kaip šifravimo reprezentacija (angl. *Encoding representation*). Šifravimo vektorius keliauja į dešifravimo grandinę. Kontekstinis vektorius yra dekodavimo tinklo būsenos įvestis, o išėjimo vertė iš kiekvieno laiko žingsnio bloko bus įtraukta į kitą RNN/LSTM tinklo bloką kaip įvestis, kad būtų galima laipsniškai prognozuoti būsimas laiko eilutės reikšmes. Dešifravimo blokų spėjimai sukuria išvesties seką. Autorių teigimu seq2seq modelis yra išmokytas maksimaliai padidinti sąlyginę tikslinės sekos tikimybę, atsižvelgiant į įvesties seką. Taip pat seq2seq modelis buvo patvirtintas kaip nepaprastai efektyvus įvairiose srityse, pradedant mašininio vertimu ir baigiant kalbos atpažinimu, ypač kai norima nustatyti ilgalaikes (ilgesnio laiko eilutės intervalo) prognozes [HLZ19].

Sąlyginės tikimybės formulė, kad neuroninio tinklo išvestys atitiks įvestis apskaičiuojama taip (c žymi kontekstinį vektorius):

$$p(y_1, \dots, y_t | x_1, \dots, x_t) = \prod_{t=1}^t p(y_t | y_1, \dots, y_{t-1}, c) \quad (28)$$

Šio modelio pagrindinis tikslas rasti apibrėžtą fiksuoto dydžio vektorius, kuris leistų tiksliai pagal turimus duomenis nuspėti būsimus keleivių srautus. Bet norint tai atlikti reikia daug realių duomenų ir apmokyti neuroninį tinklą taip, kad jis gebėtų pasirinkti svarbiausias tiriamų duomenų charakteristikas. Modeliui atskirti pagrindinius bruožus padeda atidumo mechanizmas (angl. *attention mechanism*) [BCB14]. Modelio aktyvavimo vektorius žymimas α_t . Jei skaičiavimai atliekami pirmine kryptimi nuo įvesčių į išvestis tai aktyvavimo vektorius žymimas $\vec{\alpha}_t$, jei atvirkštine - $\vec{\alpha}_t$). Kadangi LSTM tinklo apmokymas vyksta abejomis kryptimis tai žymima α_{tt} . Vektorius apskaičiuojamas pagal formulę:

$$\alpha_{tt} = \frac{\exp(e_{tt})}{\sum_{t'=1}^{T_x} \exp(e_{tt'})} \quad (29)$$

T_x reiškia įvesties sekos ilgį, o e_{tt} reiškia tarpinį energijos terminą, kuris apskaičiuojamas taip:

$$e_{tt} = FC(s_{t-1}, \alpha_t) \quad (30)$$

Kaip parodyta lygtyje (30), s_{t-1} žymi paslėptos būsenos suaktyvinimą iš ankstesnio dekode-rio LSTM laiko žingsnio, FC reiškia visiškai sujungto neuroninio tinklo veikimo funkciją, kad būtų sugeneruoti kodavimo tinklo kiekvieno laiko žingsnio būsenos balai. Formulė (29) yra „Soft-max“ funkcija, skirta normalizuoti visus atitinkamus įvesties taškus, kad būtų sukurtas tikimybių pasiskirstymas, pagrįstas tiksliniu testu. Tada kontekstinį vektorių, skirtą išvesties prognozavimui, galima apskaičiuoti kaip svertinį vidurkį, apibrėžtą taip:

$$c_t = \sum_{t'=1}^{T_x} \alpha_{tt'} \alpha_{t'} \quad (31)$$

Galiausiai, konteksto vektorių c_t ir paslėptoji būsena nuo paskutinio žingsnio yra paduodami į kitą dešifravimo neuroną kaip parametrai.

7. Literatūros apžvalgos apibendrinimas

Surinktos teorinės žinios apie neuroninių tinklų veikimą bei keleivių srautų prognozavimą pasitelkiant dirbtinį intelektą suteikė atspirtį vykdyti tyrimąjį darbo dalį. Buvo išsiaiškinta, jog darbų šia tema yra gausu ir įvairiausi neuroninių modelių yra taikomi šiai problemai spręsti. Buvo nuspręsta atlikti tyrimus su statistiniais modeliais ARIMA ir SVR, naudoti rekurentinio tipo modelius kaip LSTM ir GRU siekiant atlikti kuriamojo mokslinio praktinę darbo dalį. Atrinkta patikimiausia modelio paklaidos skaičiavimo funkcija - Kvadratinio vidurkio nuokrypio funkcija (MSE). Taip pat atrinktos pagrindinės modelių savybės, išanalizuoti modelių veikimo, apmokymo algoritmai iš sudėtingesnių dirbtinio intelekto prognozavimo modelių kaip auto-šifravimo modelis bei seq2seq modelis.

8. Viešojo transporto tvarkaraščio rengimas

Žmonės yra labai skirtingi ir autobusų važinėja skirtingais laikais. Pavyzdžiui, fabriku parduotuvių, biudžeto, paslaugų sektoriaus darbuotojai turi gana griežtą darbo grafiką ir dirba nuo 9 valandos iki 17 valandos, tai tokiems keleiviams yra svarbu nevēluoti ir tikimasi autobuso kuris ryte pastoviu, konkrečiu laiku nuvežtų į darbą, o vakare parvežtu atgal. Yra grupė žmonių kurie dirbą pagal laisvą darbo grafiką: menininkai, rašytojai, programuotojai, akademikai ir kiti. Šios grupės žmonės dirba nuolat kintančiais laikais tai ir jų išvykimo laikas labiau svyruoja. Galima išskirti daug daugiau įvairiausių grupių, bet svarbiausia, jog tvarkaraštis tenkintu visų grupių kelionių poreikius. Kadangi Vilniaus miesto transportas juda linijomis pirmyn ir atgal, tai analizavau linijinį viešojo transporto judėjimo modelį [dPL01]. Šio modelio veikimo principas pagrįstas tuo, jog yra išdėliotas preliminarus autobuso tvarkaraštis atsižvelgiant į žmonių grupių poreikius. Žinoma, jog kiekvienai žmonių grupei yra būdingos skirtingos toleravimo atvažiavimo per anksti ir vėlavimo paklaidos. Pavyzdžiui, dauguma į darbą nori nuvykti laiku, arba truputį ankščiau, vėluoti nenori. O išvykti iš darbo laiku arba vėliau, bet vengti - ankščiau palikti darbo vietą. Svarbiausia reikia išvengti, kad viešojo transporto priemonės nevažinėtų tuščios.

8.1. Autobusų kiekio paskirstymas

Pagal JUDU ataskaitos duomenis didžiosios dalies autobusų (58 proc. 306 vienetų) nominali talpa yra tarp 50 ir 99 keleivių. Autobusai talpinantys tarp 100 ir 149 keleivių yra 165, tai sudaro 31 proc. viso autobusų parko, talpinantys 150 ir daugiau – 46 autobusai, 8,7 proc. Didžiausias autobusų parke esantis autobusas nominaliai talpina 164 keleivius. Bendras autobusų keleivių vietų nominalus skaičius 55,3 tūkst

Kad būtų galima apskaičiuoti optimalų autobusų kiekį maršrutui atitinkamam laiko tarpui reikia sužinoti koks yra maksimalus keleivių skaičius maršrute per atitinkamą laiko tarpą. Sekti galimą potencialų keleivių kiekį keliaujančiu maršrutu tarp stotelių galima apskaičiuoti: jei prie esančių keleivių skaičiaus kas stotelę pridėsime įlipančių keleivių ir išlipančių keleivių skirtumą. Sužinoję maksimalų keleivių skaičių maršrute galima priskirti atitinkamus autobusus.

Remiantis JUDU pateikta informacija: Daugiausiai keleivių pervežantis maršrutas, kurio vidutinis dienos pervežtų keleivių skaičius yra 22,75 tūkst., yra 2G maršrutas (Santariškės–Stotis). Kiti populiariausi maršrutai pervežantys daugiausiai keleivių yra 4G (Pilaitė–Saulėtekis), 1G (Stotis–Santariškės), 3G (Fabijoniškės–Oro uostas) ir 2 troleibusas (Saulėtekis–Stotis).

8.2. Transporto priemonių didžio bei komforto pasirinkimas

Paprastai, žmonės viešojo transporto keliavimo būdo atsisako todėl, kad:

- transporto priemonės būna perpildytos keleivių.
- nuvilia transporto priemonių komforto lygis[Ced11].

Dėl komforto sprendimas paprastas reikia prižiūrėti ir atnaujinti transporto priemones. O dėl perpildytų transporto priemonių sprendimas būtų reaguoti į periodines autobusų perpildymo situacijas ir

per piko laikus paskirti važiuoti didesniems autobusams arba numatyti papildomus autobusus piko metu [Ced11].

8.3. Žmonių srauto prognozavimas VT sistemoje

Žmonių judėjimo pobūdis traukiniais ar autobusais beveik nesiskiria. Problema yra ta, jog traukinių ir metro stotelėse yra griežtesnė patikra. Pavyzdžiui, patekti į metro reikia pereiti vartus ir užtenka paprastos žmonių fiksavimo ir skaičiavimo sistemos, kad gautume gana tikslius žmonių srautų duomenis realiu metu. Deja Autobusų tinklas neturi tokios griežtos priežiūros sistemos (bent Vilniuje). Informaciją apie keleivių srautus dabar dažnai yra surenkama pasitelkus išmaniųjų infraraudonųjų spindulių kamerų sistemą, kuri yra integruota naujausiuose miesto autobusuose. Kaip galime pastebėti ne visi miesto autobusai yra nauji, o stebėjimo sistemos kartas genda ar padaro klaidų. Atliekant darbą duomenų failuose buvo atrasta klaidų, kai kuriuos įrašus teko panaikinti.

8.4. Viešojo transporto srauto optimizavimas

Miesto viešojo transporto priemonių ir dirbančiųjų vairuotojų skaičius yra ribotas. Negalime padaryti taip, jog per kiekvieną laiko momentą būtų galima įsėsti į norimą autobusą ir važiuoti norima linkme, nes tai būtų labai brangu. Negalime padaryti ir taip, jog važiuojančių autobusų kiekis būtų labai mažas, nes tai gali sukelti keleiviams nepatogumų, kas galiausiai lems mažėjančius keleivių kiekius. Kad būtų pasiektas balansas transporto srautus reikia optimizuoti. Optimizuoti viešojo transporto srautus galime atsižvelgę į šiuos aspektus:

- tinklo maršruto projektavimą
- tvarkaraščio rengimas
- transporto priemonių didžio bei komforto pasirinkimas

8.4.1. VT tinklo maršrutų projektavimas

Viešojo transporto maršrutus galima pavaizduoti grafu $G(V,E)$, kur stotelės žymi mazgus - V , o atitinkamo viešojo transporto judėjimo linijas jungiančias stoteles - E . Mazgai turi būti parinkti prie lankomų vietų: mokyklų, prekybos centų, gyvenamųjų rajonų centų, lankytinų vietų. O sujungti taškus reikia taip, jog tarp jų būtų kiek įmanoma artimesnis atstumas [PRR95]. Vilniaus miesto viešojo transporto maršrutų grafinis žemėlapis yra pavaizduotas žemiau (žr. 8 pav.).



8 pav. Vilniaus miesto maršrutinio transporto schema

Kad būtų galima nustatyti žmonių kiekį kiekvienoje viešojoje transporto priemonėje tam tikru metu reikia kiekvienoje stotelėje perskaičiuoti žmonių skaičių, tai yra pridėti įlipusiųjų kiekį ir atimti išlipusiųjų. Turint keleivių skaičius kiekvienoje stotelėje, galima išsivesti kiekių keleivių kiekių vidurkius atitinkamu metu. Šie skaičiai padės nustatyti kuriame ruože viešasis transportas yra apkrautas, o kuriuose yra tuštoka [PRR95]. Jeigu būtų žinomi maksimalūs keleivių skaičiai autobusuose tam tikro laikotarpiu, analitikai galėtų įvertinti esamą situaciją ir pagal tai reaguoti. Pavyzdžiui, jeigu tam tikrą valandą yra numatomi dideli keleivių kiekiai būtų įvertinama dabartinė situacija ir pakeisti autobusai į didesnius, arba padidintas jų kiekis. Jei numatomas galimas keleivių skaičiaus sumažėjimas tuomet autobusai galėtų būti pakeisti į mažesnės talpos autobusus.

9. Testinių duomenų atrinkimas

Kad būtų galima atlikti tyrimą reikia gauti realius ir tikslius duomenis apie esamus keleivių srautus. Pasirinkti tyrimo duomenys - Vilniaus miesto VT duomenys gauti iš Vilniaus savivaldybės įmonės "Susisiekimo paslaugos" (toliau JUDU). Nors vis dar populiariausias būdas surinkti VT keleivių srautų duomenis tebėra rankinis skaičiavimas. Nors teigiama, kad tokiu būdu patiriamos gana mažos investicijos bei užfiksuoti duomenys yra detalai aprašyti (pabrėžiamos situacijos kai keleiviai dėl grūsties negalėjo įlipti į VT priemonę ir panašiai), tai nėra itin tikslus būdas (ypač kai esama didelių žmonių srautų) nustatyti situacija dėl žmogiškojo faktoriaus. Šiuo metu Vilniaus 37 % viešojo transporto sistemos sudaro išmanūs autobusai kurie turi sumontuota integruotą keleivių skaičiavimo techniką su infraraudonųjų spindulių fiksavimo davikliais [S4I14]. JUDU atstovė teigia, jog ši sistema yra itin patikima - davikliai labai jautrūs, o sistema atlieka itin tikslų (>95 %) įlipančių ir išlipančių keleivių skaičiavimą. Tyrime naudosisi keleivių srauto duomenis VT priemonėse kurie yra surinkti automatinėse sistemose žmonių fiksavimo pagalba.

9.1. VT judėjimo duomenys

Iš JUDU yra gauti išrašai apie naujausių autobusų veiklą - **VT judėjimo duomenys** nuo 2018-12-31 iki 2020-04-13. Visi šie duomenys užima apie 25.6 GB vietos. Pateikti viešojo transporto parametrai (paryškinti parametrai yra naudoti tyrime):

- Maršruto Versija (org. "Version")
- Stotelės tipas (org. "Type")
- Operatorius (org. "Operator")
- Aikštelės numeris (org. "Site")
- **Transporto priemonės tipas (org. "Vehicle Type")**
- **Atvykimo į stotelę laikas (pagal vietos laiką) (org. "Arrival Local")**
- Atvykimo į stotelę laikas (pagal universalų laiką) (org. "Arrival UTC")
- Išvykimo iš stotelės laikas (pagal vietos laiką)(org. "Departure Local")
- Išvykimo iš stotelės laikas (pagal universalų laiką) (org. "Departure UTC")
- Paskutinė maršruto stotelė (org. "Way Last Station")
- Transporto priemonės Visas nuvažiuotas kelias (org. "Total Way")
- **Maršruto numeris (org. "Line")**
- **Stotelė (org. "Station")**
- Kita stotelė (org. "Next Station") (deja šis parametras yra neužpildytas)

- Buvusi stotelė (org. "Left Station") (deja šis parametras yra neužpildytas)
- Stebinčių palydovų skaičius (org. "Satellites")
- Geografinė Platuma (org. "Latitude")
- Visas keleivių pokytis stotelėje (org. "Total Exchange")
- **Visas įlipusių keleivių stotelėje (org. "Total In")**
- **Visas išlipusių keleivių stotelėje (org. "Total Out")**
- Sistemos būseną (org. "System State")
- Pirmos durys (org. "Door001")
- Įlipusių keleivių skaičius per pirmas duris (org. "In001")
- Išlipusių keleivių skaičius per pirmas duris (org. "Out001")
- Keleivių skaičiaus pokytis per pirmas duris (org. "Exchange001")
- Keleivių skaičiaus papildymas per pirmas duris (org. "Supplement001")
- Antros durys (org. "Door002")
- Įlipusių keleivių skaičius per antras duris (org. "In002")
- Išlipusių keleivių skaičius per antras duris (org. "Out002")
- Keleivių skaičiaus pokytis per antras duris (org. "Exchange002")
- Keleivių skaičiaus papildymas per antras duris (org. "Supplement002")
- Trečios durys (org. "Door003")
- Įlipusių keleivių skaičius per trečias duris (org. "In003")
- Išlipusių keleivių skaičius per trečias duris (org. "Out003")
- Keleivių skaičiaus pokytis per trečias duris (org. "Exchange003")
- Keleivių skaičiaus papildymas per trečias duris (org. "Supplement003")

Gauti maršrutų numeriai : 1, 1G, 2, 2G, 3, 3G, 4, 4G, 5, 5G, 6, 6G, 7, 8, 9, 10, 11, 12, 13, 13T, 14, 15, 16, 17, 18, 18T, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 46, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 61, 62, 63, 65, 66, 67, 68, 69, 73, 74, 75, 76, 78, 82, 86, 87, 88, 88N, 89, 90, 91, 92, 93, 94, 100, 101N, 102N, 103N, 104N, 105N,

Prieš taikant šiuos duomenis gilesiems prognozuojamiesiems neuroniniams tinklams apmokyti, kurie gebėtų prognozuoti būsimus keleivių srautus, reikia atrinkti prognozėms įtaigius rodiklius. Tam buvo atsižvelgta į keletą publikacijų, kurios tinkamam duomenų surinkimui išskiria ypatingą dėmesį (Žr. į 2).

1 lentelė. 10 populiariausių maršrutų pagal apdirbtus duomenis kurie gauti iš JUDU

Dešimt populiariausių Vilniaus miesto autobusų maršrutų		
Maršrutas	įrašų kiekis dienos atžvilgiu	įrašų kiekis valandos atžvilgiu
1G	1200	24105
10	1200	24389
2	1200	23855
16	1199	24296
18	1198	24536
2G	1196	24109
7	1196	24179
30	1196	23602
3G	1196	17277
49	1196	23955

2 lentelė. Duomenų sintezės pavyzdžių palyginimas iš skirtingų mokslinių darbų, siekiant prognozuoti būsimus keleivių srautus

Duomenų sintezės pavyzdžių palyginimas iš skirtingų mokslinių darbų			
Straipsnis	Tiriamas objektas	Laiko intervalas	Ypatumai
[KZY ⁺ 17]	Taksi	Valandos	Piko valandos, ne piko valandos, miego valandos
[LC17]	Autobusai	Valandos	Savaitės, šventės, paprastos darbo dienos, kryptis, praeitos vidutinės vertės
[MXM ⁺ 14]	Autobusai	Kas 30 minučių	Realaus laiko stebėjimas
[SLZ ⁺ 20]	Metro tinklas	Kas 15, 30, 45 minučių	Atsižvelgiama į savaitės dienas, šventes, esamus orus.
[WC12]	Metro tinklas	Kas 5, 15, 60 minučių	Įvairūs laiko režimai
[GXQ ⁺ 19]	Traukiniai	Kas 15 minučių	Kombinacija ankstesnių keleivių srautų vidurkiai ir realaus laiko aptikimo skaičiai
[HLZ19]	Viešasis transportas	Periodais (1-6 valandos)	Dienos metas, savaitės diena, orai.

9.2. Surinkti duomenis turi turėti papildomų etikečių

Tinkamai neapdirbta surinkta statistika apie buvusius keleivių srautus gali būti "neiškalinga". Pavyzdžiui, jei turėsime kiekvienos maršruto stotelės įlipusių ir išlipusių keleivių skaičius be jokių

laiko žymos etikečių tai ir žmogui apytiksliai nuspėti kokio žmonių srauto tikėtis. Juk ir kiekvienam viešojo transporto naudotojui yra aišku, jog darbo dienomis piko metu yra tikimasi didelio keleivių srauto, o per įprastines darbo valandas (9h-12h ir 13-16h), vakare, bei savaitgaliais galimai bus keleivių srautai maži. Šiuos rodiklius reikia pateikti ir į testinius duomenis, kuriais bus apmokomas neuroninis tinklas. Pagrindiniai minimi vidiniai rodikliai:

- valandos
- paros laikas (ar piko metas?)
- savaitės diena
- mėnuo

9.3. Autobusų charakteristikos

Akivaizdu, jog Vilniaus miesto transporto priemonių didžiai yra skirtingi reikia ir į šiuos parametrus atsižvelgti. Pavyzdžiui, vienas autobusas gali talpinti apie 50 keleivių, o kitas virš šimto, tai palyginti autobuso apkrovą žinant tik momentinį keleivių skaičių yra neįmanoma atrinkti parametrai.

- Sėdimų vietų kiekis
- Stovimų vietų kiekis
- Bendras vietų kiekis

9.4. Oro sąlygos

Oro sąlygos yra labai svarbus veiksnys, kuris įtakoja mūsų žmonių rutiną. Išoriniai oro parametrai yra svarbus aspektas keleivių srauto prognozavimui. Kelios mokslinės publikacijos [HLZ19; SLZ⁺20] pamini, jog svarbu prie surinktų keleivių srauto duomenų pažymėti to meto orų sąlygas. Renkami rodikliai:

- saulėtumas
 - debesuotumas
 - trumpųjų bangų apšvita (ALLSKY SFC SW DWN)
 - ilgųjų bangų apšvita (CLRSKY SFC SW DWN)
 - Viso dangaus insoliacijos skaidrumo indeksas (ALLSKY KT)
 - Viso dangaus normalizuoto insoliacijos skaidrumo indeksas (ALLSKY NKT)
- oro temperatūra
 - žemės paviršiaus temperatūra

- amplitudė oro temperatūros 2 metrų spindulyje (T2M)
- minimali oro temperatūra 2 metrų spindulyje (T2M MIN)
- maksimali oro temperatūra 2 metrų spindulyje (T2M MAX)
- vėjo greitis bei slėgis
 - žemės paviršiaus slėgis
 - amplitudė vėjo greičio 50 metrų spindulyje (WS50M)
 - minimalus vėjo greitis 50 metrų spindulyje (WS50M MIN)
 - maksimalus vėjo greitis 50 metrų spindulyje (WS50M MAX)
- drėgnumas
 - Santykinė oro drėgnumas 2 metrų spindulyje (RH2M)
 - specifinis oro drėgnumas 2 metrų spindulyje (QV2M)
 - kritulių kiekis (mm/dieną) (PRECTOTCORR)

Oro sąlygos buvo atrinktos iš NASA stebėjimo duomenų bazė: <https://power.larc.nasa.gov/data-access-viewer/>, Šis serveris leidžia atsisiųsti duomenis iš pasirinktos vietovės, pasirinktu laiku bei įvairiais intervalais (paros, valandos, mėnesinio ...). Buvo išrinkti Vilniaus miesto centro duomenys iš 2018-12-31 - 2022-04-13 laikotarpio su paros ir valandos intervalais. Bendras parametrų skaičius - 16.

9.5. Covid-19 statistika

Covid-19 viruso sukelta pandemija turėjo akivaizdų poveikį viešajam gyvenimui, kaip ir VT paklausai. Buvo atrinkti iš The Humanitarian Data Exchange puslapio duomenys: <https://data.humdata.org/dataset/novel-coronavirus-2019-ncov-cases>.

Konkretūs parametrai:

- dienos susirgimo atvejų skaičius
- bendras susirgimo atvejų skaičius
- dienos mirčių skaičius
- bendras mirčių skaičius

Iš duomenų išskirti Lietuvos ir pasaulio covid-19 statistika. Bendras parametrų skaičius - (2 * 4 =) 8.

9.6. Apskaičiuoti keleivių skaičių VT priemonėje

Nors ir išmaniosios VT fiksuoja įlipančius ir išlipančius keleivių srautus, bet nėra fiksuojami esamas keleivių srautas VT priemonėje. Tam buvo pasitelktas paprastas algoritmas kiekvienai VT sesijai, kiekvienoje stotelėje yra apskaičiuotas keleivių skaičius pagal formulę (žr.32):

$$N_i = N_{i-1} + IN_i - OUT_i \quad (32)$$

- (naujas keleivių skaičius) = (keleivių skaičius praeitoje stotelėje) + (įlipusių keleivių skaičius) - (išlipusių keleivių skaičius)

N_i keleivių skaičius esamu etapu, N_{i-1} keleivių skaičius praeitame etape. etapas tai laiko tarpas kaip VT nuo vienos stotelės keliauja iki kitos. IN_i įlipusių keleivių skaičius praeitoje stotelėje, OUT_i išlipusių keleivių skaičius praeitoje stotelėje. Gauto keleivių srautai įtraukti į VT judėjimo duomenis.

9.7. Duomenų grupavimas

Buvo atlikti du skirtingi VT judėjimo duomenų grupavimai. Grupavimo parametrai:

- Data
- Maršruto numeris
- Transporto priemonės tipas
- Stotelė (keliems specifiniams tyrimams)
- Valanda (tik antram tyrimui)

Atrenkama:

- Grupės bendras įlipusiųjų žmonių kiekis
- Grupės bendras išlipusiųjų žmonių kiekis
- Grupės maksimalus keleivių kiekis VT
- Grupės minimalus laisvų vietų skaičius VT

Ir tai gautos dvi lentelės: viena iš jų orientuota į paros rezultatus, kita į valandos.

9.8. Duomenų sujungimas

Tyrimui atlikti buvo surinktos keturios duomenų lentelės:

- Sugrupuoti VT judėjimo duomenys
- Autobusų charakteristikos

- Orų sąlygos
- Covid-19 statistika

Jos buvo sujungtos į vieną pagal šiuos stulpelius.

- VT judėjimo duomenys ir Autobusų charakteristikos pagal autobuso tipą
- VT judėjimo duomenys ir Orų sąlygos pagal datą (ir valandą)
- VT judėjimo duomenys ir Covid-19 statistika pagal datą

9.9. Duomenų normalizacija

Norint pilnai parengti testavimo duomenis reikia juos normalizuoti. Normalizavimas tai duomenų priskyrimas tam konkretiems režiams, kad nebūtų jokių didelių nukirpimų, kurie galėtų lemti neuroninio tinklo apmokymo klaidas. Vienas iš populiariausių normalizavimo algoritmų minimumo-maksimumo normalizacija (angl. *Min-Max normalization*). Normalizacijos funkcija atrodo taip 33:

$$z = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (33)$$

x - pradinė reikšmė, z - normalizuota reikšmė.

10. Servisai ir įrankiai

Duomenų apdirbimui bei tyrimui atlikti buvo pasitelkta:

1. Python programavimo kalba (3.8.10 versija)
2. Paketų valdytojas - pip (21.3.1)
3. Visual studio code kodo redaktorius
4. Debesijos servisai: Google colab (PaaS), Google drive (SaaS)
5. Modeliui konstruoti panaudotos Tensorflow ir Keras bibliotekos
6. Pandas duomenų analizavimo biblioteka
7. Matplotlib grafikų braižymo biblioteka

10.1. Tensorflow

Tensorflow programinė įranga yra skirta apmokyti neuroninius tinklus. Ji buvo parašyta C++ kalba „Google“ kompanijos 2015 metais. Dabar ši programinė įranga yra labai populiari, nes ji yra nemokama, greita ir universali. Universali tuo atžvilgiu, kad ją galima taikyti įvairiose platformose: kompiuteriuose, debesinėse sistemose, mobiliuose įrenginiuose, bei aplikacijos sąsaja (API) yra palaikoma įvairių programavimo kalbų: Python, Java, C++, Go. Taip pat TensorFlow mokymo procesas gali būti vykdomas skirtingose terpėse:

1. pasitelkiant centrinius apdorojimo vienetais (CPU),
2. grafinio apdorojimo vienetais (GPU),
3. visai neseniai Google sukurtais tensoriniais apdorojimo vienetais (TPU).

10.2. Keras

Keras yra giliojo mokymosi API (Aplikacijų programavimo sąsaja), kuri palaikoma Tensorflow platformos. Keras leidžia lengvai apsibrėžti neuroninio tinklo modelį. Nereikia detaliai aprašyti kiekvieno modelio sluoksnio, užtenka nurodyti sluoksnių parametrus ir pradinio sluoksnio įvestis, o Keras savarankiškai sudeda sluoksnius į neuroninį tinklą. Taip pat Keras biblioteka veikia greitai ir geba dirbti su dideliu duomenų kiekiu.

10.3. Google Colab

Tensorflow bendruomenė yra suinteresuota atrasti kuo pažangesnį neuroninių tinklų modelį, todėl sukūrė debesinę servisą - Google Colab platformą kuri leidžia nemokamai apmokyti neuroninius tinklus, naudojant Google CPU, GPU ar TPU serverius.

Tyrimo metu buvo išbandytos visos Google Colab pateikiamos skaičiavimo terpės. Greičiausiai skaičiavimai buvo atliekami TPU terpėje, todėl ji buvo pasirinkta tyrimams atlikti.

3 lentelė. Keras sluoksnių pavadinimų ir neuroninio tinklo sluoksnių sugretinimas

Keras sluoksnių paaiškinimas	
Keras sluoksnis	Paaiškinimas
LSTM	LSTM neuroninis tinklas
GRU	GRU neuroninis tinklas
SimpleRNN	RNN neuroninis tinklas
Bidirectional	Dvi kryptis neuroninis tinklas (BPN)
Dropout	Išmetimo sluoksnis
Dense	Išlyginimasis sluoksnis

11. Tyrimo metodika

Pradinis tikslas yra sukurti matematinį-statistinį bei dirbtinio intelekto prognozuojamąjį modelį, kuris gebėtų gana tiksliai prognozuoti būsimų viešojo transporto maršruto keleivių srautus. Tyrimui atlikti pasitelkiau paruoštus duomenis:

1. Konkretaus maršruto keleivių prognozavimas paros atžvilgiu.
2. Konkretaus maršruto keleivių prognozavimas valandos atžvilgiu.

Kadangi duomenų apie 1G maršruto autobusą buvo bene daugiausiai tai jis buvo pasirinktas kaip pagrindinis tyrimo objektas, t. y. atsisiltruoti visi jo duomenys.

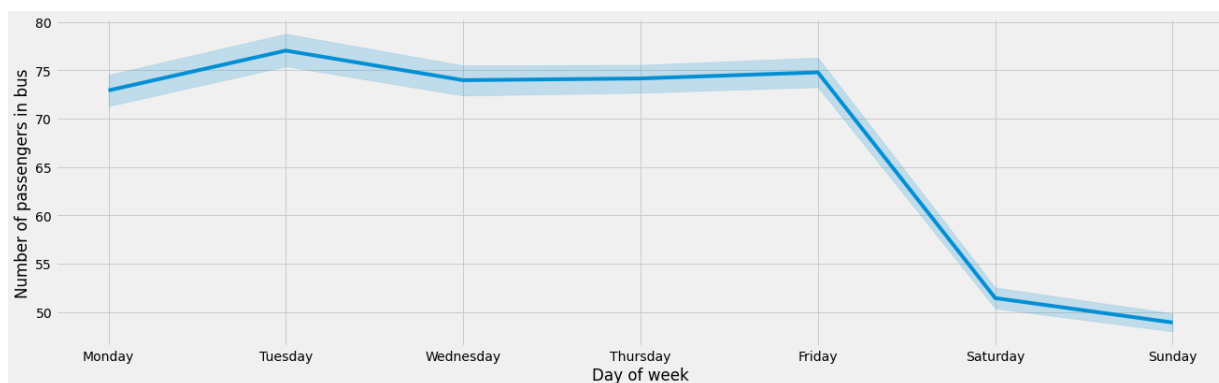
11.1. Maršruto keleivių prognozavimas metų dienų atžvilgiu

Kad būtų galima įžvelgti tam tikras duomenų kitimo tendencijas siekiau išsiaiškinti kokią priklausomybę nuo metų dienos ir keleivių srauto kiekio gali aptikti AI modelis. Buvo sugrupuoti duomenis pagal:

- Atvykimo į stotelę laiko datą
- Savaitės dieną
- Maršruto numerį

Surinktos visos kiekvienos grupės įlipusių bei išlipusių keleivių kiekis stotelėse. Atrinkti keli autobusų maršrutus: 1G, 13 ir 19. Šie maršrutai pasirinkti, nes buvo sutikta skirtingi kiekiai įrašų apie juos ir buvo svarbu išsiaiškinti ar jie pasižymės tomis pačiomis savybėmis.

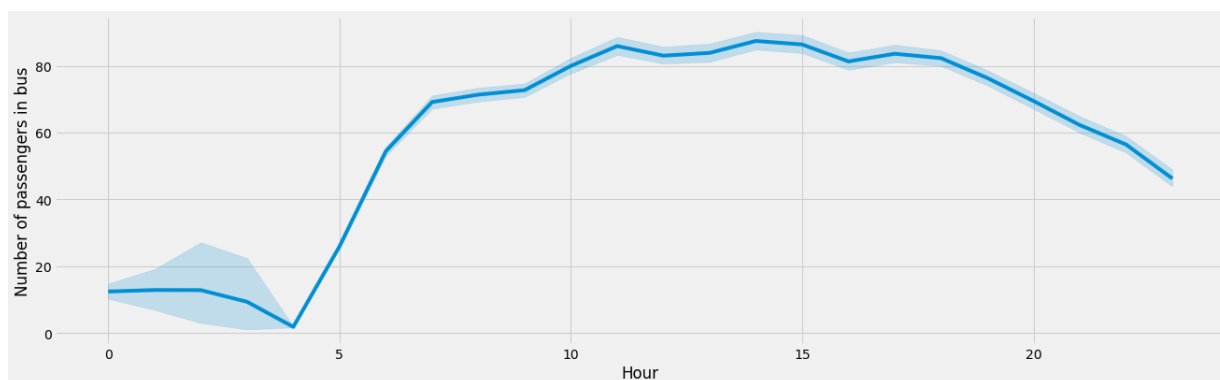
Pagal atrinktus 1G autobuso maršruto duomenis sudariau linijinį grafiką, kuris parodo kokia yra keleivių priklausomybė nuo konkrečios savaitės dienos (Žr. į 9). Iš grafike aiškiai matosi, jog keleivių intensyvumas yra didelis darbo dienomis, o sąlyginai mažas savaitgaliais. Su 13 ir 19 autobusais gavau panašias tendencijas.



9 pav. 1G keleivių skaičiaus priklausomybė pagal savaitės dieną

11.2. Maršruto keleivių prognozavimas valandos intervale

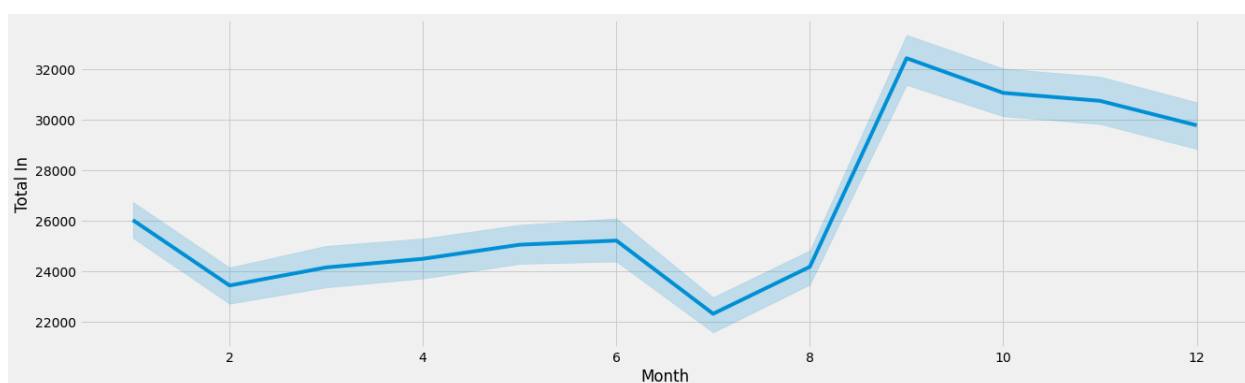
Atliktas pagrupavimas iš JUDU pateiktiems duomenims apie VT veiklą pagal atvykimo datą, valandą ir maršruto numerį. Atlikau duomenų keleivių srautų didžio analizę paros valandų atžvilgiu (žr. 10 pav.). Grafikas sudarytas pasitelkiant tik 1G autobuso maršruto duomenimis ir jis sudarytas iš dviejų ašių: valandos ir maksimaliu keleivių kiekiu 1G transporto priemonėje. Galima pastebėti, jog viešasis transportas labiausiai apkrautas 10-18 valandomis virš 80 keleivių, o 0-5 valandomis labai mažas.



10 pav. 1G keleivių skaičiaus priklausomybė pagal paros valandas

11.3. Maršruto keleivių prognozavimas mėnesių atžvilgiu

Atliktas duomenų keleivių srautų didžio analizę mėnesio atžvilgiu (žr. 11 pav.). Grafikas sudarytas pasitelkiant su 1G autobuso maršruto duomenimis ir jis sudarytas iš dviejų ašių: valandos ir per jas įlipančių keleivių bendrasis skaičius (išlipusiųjų skaičius sutampa su įlipusiųjų). Galima pastebėti, jog 1G autobusas labiausiai apkrautas rugsėjį virš 32 tūkstančių keleivių, o mažiausiai keleivių būna liepą apie 23 tūkstančius keleivių.



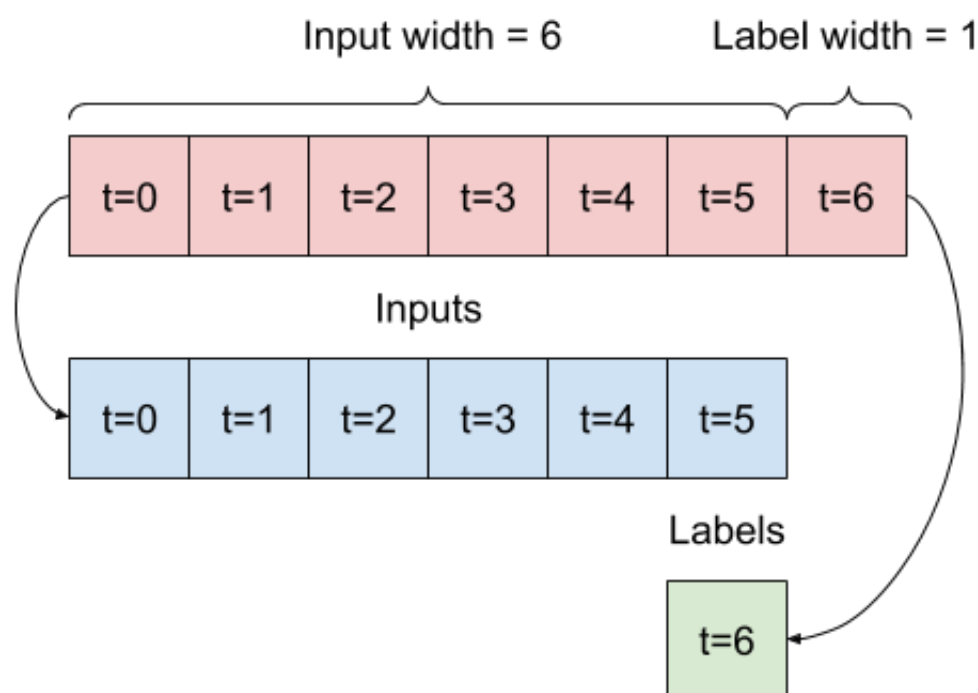
11 pav. 1G keleivių skaičiaus priklausomybė pagal mėnesį

11.4. LSTM neuroninio tinklo taikymas būsimų keleivių srautams nustatyti

Prieš pateikiant duomenis modeliui buvo atlikti perteklinių reikšmių pašalinimo darbai. Buvo ištrinti apibrėžiamieji parametrai kaip: data, savaitės dienos pavadinimas, išlipusių keleivių kiekio

suma (nes šis skaičius sutapo su įlipusiųjų skaičiumi) bei autobuso numerį. Likusius parametrus normalizavau pasitelkiant "RobustScaler" biblioteka. Duomenų normalizavimas yra svarbus veiksnys, nes parametrų reikšmės gali būti išsimėčiusios po platų reikšmių intervalą, kas neuroninio tinklo apmokymo metu gali sukelti neapibrėžtumo problemų.

Kadangi prognozuojamieji neuroniniai tinklai tokie kaip LSTM yra orientuoti į ne pavienius duomenis, o į jų sekas. buvo sudarytos 30 elementų (24 elementų sekos valandiniam keleivių srautui prognozuoti) ilgio sekos. Kuriose saugoma iki tol buvusių dienų parametrai (oro sąlygos, covid-19 statistika, savaitės diena, mėnuo) ir saugomas paskutinio elemento rezultatas, šiuo atveju paskutinio laiko vieneto tam tikro maršruto maksimalus VT priemonėje važiuojančių keleivių bendras skaičius. Sekų sudarymo vizualas pateiktas žemiau (Žr. į 12).



12 pav. Sekų sudarymas

Duomenys buvo suskirstyti į dvi grupes mokymo ir testavimo. Testavimui buvo pasirinkti 100 paskutinių įrašų, o likę apie duomenys (kurie apima 3 metus) - mokymui. Sukūriau mokymo modelį.

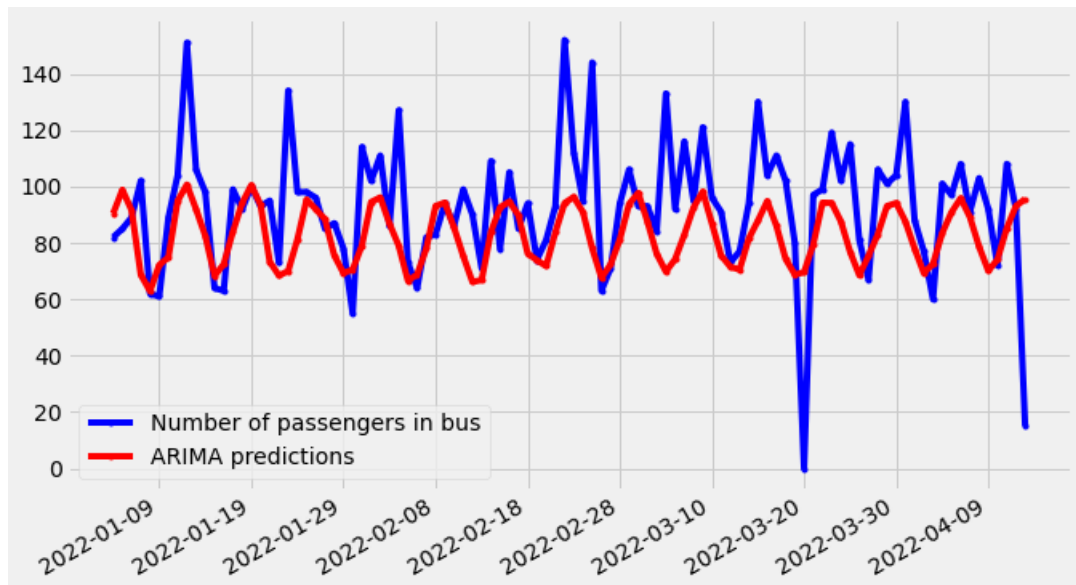
11.5. Statisinių-matematinių prognozavimo modeliai

Kad būtų galima išvelgti naudą dirbtinio intelekto pritaikymo keleivių maksimalaus kiekio prognozavime reikia įsitikinti, jog jis geba tiksliau atlikti darbą nei paprasti statistiniai-matematiniai prognozavimo modeliai. Pasirinkti modeliai:

- ARIMA
- SVR su funkcija rbf (angl. "angl. Radial basis function")

11.5.1. ARIMA maksimalios paros keleivių skaičiaus prognozės

ARIMA modelis buvo sukurtas pasinaudojus *statsmodels.tsa.statespace.sarimax* biblioteka. Kadangi ARIMOS modeliui reikia nurodyti optimalius parametus: p , d , q tam buvo pasitelktas *pmdarima.auto_arima* metodas kuris praleidžia daugumą modelių ir pasiūlo geriausius modelio parametrus. Tyrimo metu buvo gautas (4,1,3) parametų vektorius. ARIMA modeliui apmokyti buvo skirti visi išskyrus paskutinius 100 įrašų. Pastarieji buvo skirti modelio veikimo validavimui (žr. į 13 pav.).



13 pav. Maksimalaus keleivių skaičiaus ARIMA modelio prognozavimo rezultatų bei realių reikšmių grafikai dienos atžvilgiu

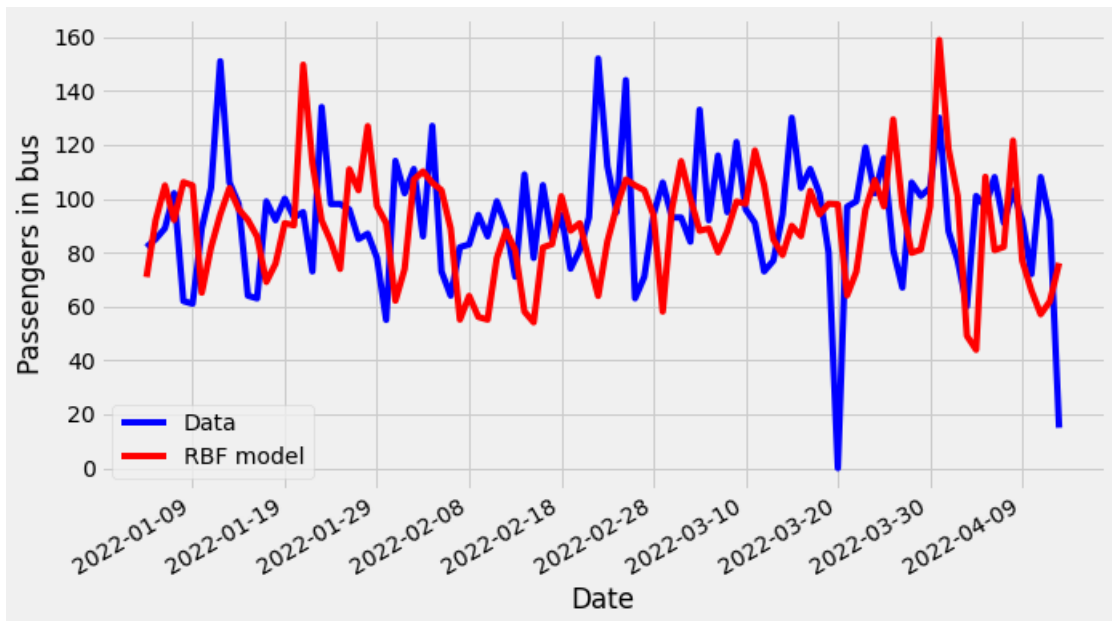
ARIMA MSE paklaida siekia apie 543.6 %. Iš diagramos matosi, jog spėjimai yra sezoniniai, cikliška pasikartojantys. Su ARIMA modeliu yra sunku atlikti tikslius maksimalaus keleivių spėjimus, nes šis modelis orientuojasi tik į vieną parametą - "maksimalų keleivių skaičių", kiti parametrai kurie gali būti naudingi yra nenaudojami.

11.5.2. SVR maksimalios paros keleivių skaičiaus prognozės

SVR modelis su rbf funkcija prognozavo maksimalų keleivių skaičių. SVR Modelis panaudotas iš *sklearn.svm* bibliotekos. SVR parametrai :

- data (skaičiaus išraiška).
- savaitės diena (skaičiaus išraiška).

SVR modeliui apmokyti buvo skirti visi išskyrus paskutinius 100 įrašų. Pastarieji buvo skirti - validavimui (žr. į 14 pav.).



14 pav. Maksimalaus keleivių skaičiaus SVR modelio prognozavimo rezultatų bei realių reikšmių grafikai dienos atžvilgiu

SVR modelio prognozavimo MSE paklaida 916%. Tai reiškia, jog modelis validavimą atliko prastai. Validavimo pagerinimui reikia pateikti daugiau duomenis apibūdinančių parametru.

11.6. Neuroninių tinklų prognozavimo modeliai

Neuroninių tinklų prognozavimo modeliai geba savarankiškai spręsti įvairias problemas, jiems nereikia priskirinti įvairių parametru kaip kaip statistiniams-matematiniams prognozavimo modeliams.

11.6.1. Abipusio LSTM neuroninio tinklo taikymas nuspėti maksimalų paros keleivių skaičių

Tyrimo buvo pritaikytas dvikryptis LSTM modelis (žr. į 15 pav.). LSTM modelio parametrai:

- Įvesties vektoriaus formos parametrai: 30 (sekos ilgis) * 27 (bruožų skaičius)
- Abipusis skaičiavimų judėjimo tinklas.
- 128 skaičiavimo vienetai (angl. "units").
- Įvesties matricos dimensijas.
- Skaičiavimo mazgų išmetimo dalis 20%.
- Pritaikyta išlygiavimo funkcija (angl. "dense").
- Skaičiavimų tikslumui nustatyti pritaikyta - "Vidutinė absoliutinė paklaida" ir "adam" optimizatorius.

- Modeliui apmokyti skirtos 50 epochų,
- Paketo iš 32 elementų
- Validavimui skirta 10% treniravimo duomenų.
- Uždraustas duomenų maišymas, nes nuoseklumas yra svarbus šiam modeliui

```

from tensorflow import keras

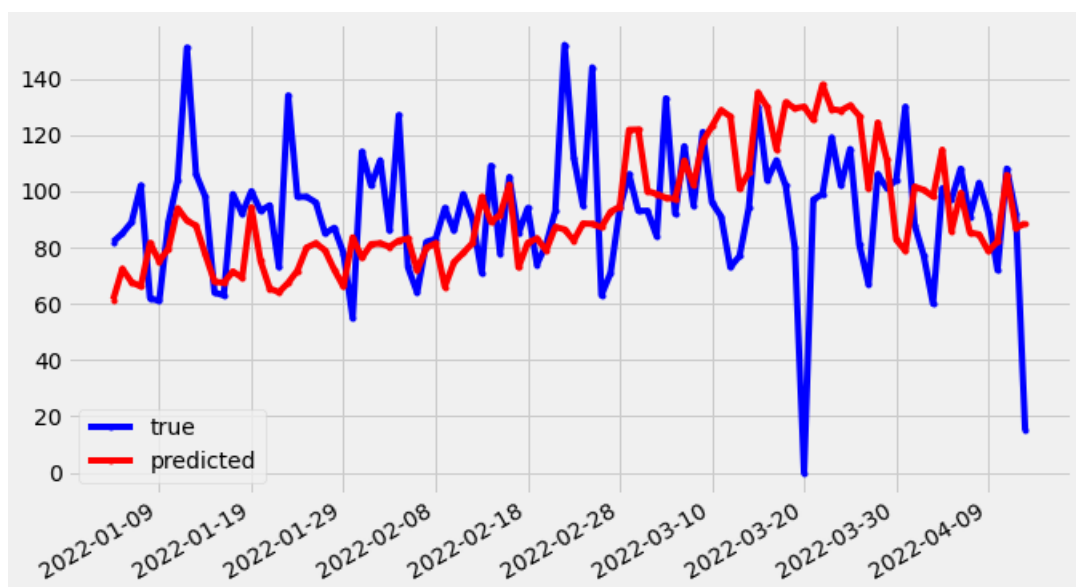
model = keras.Sequential()
model.add(
    keras.layers.Bidirectional(
        keras.layers.LSTM(
            units=128,
            input_shape=(X_train.shape[1], X_train.shape[2])
        )
    )
)
model.add(keras.layers.Dropout(rate=0.2))
model.add(keras.layers.Dense(units=1))
model.compile(loss='mean_squared_error', optimizer='adam')

history = model.fit(
    X_train, Y_train,
    epochs=30,
    batch_size=32,
    validation_split=0.1,
    shuffle=False
)

```

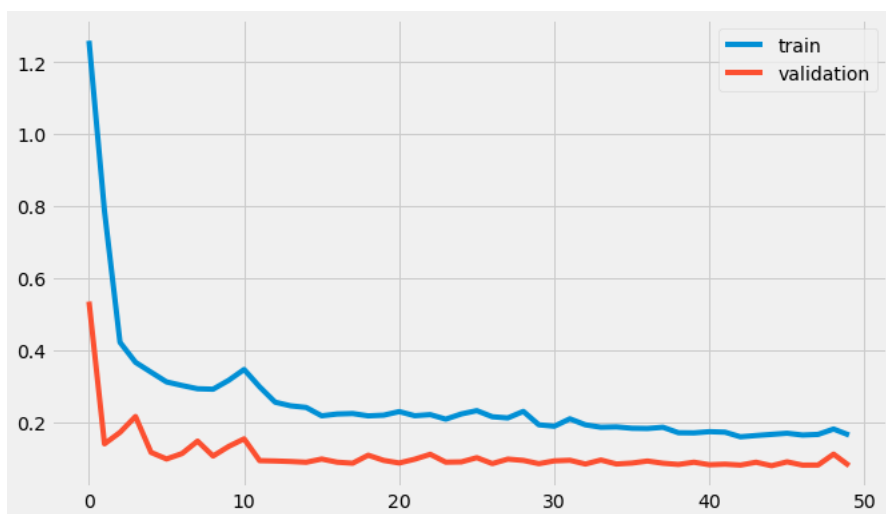
15 pav. LSTM modelis

LSTM modelio rezultatai pavaizduoti linijiniame grafike (žr. 16 pav.), raudona linija žymi modelio prognozuotas reikšmes, mėlyna - realias.



16 pav. Maksimalaus keleivių skaičiaus LSTM modelio prognozavimo rezultatų bei realių reikšmių grafikai dienos atžvilgiu

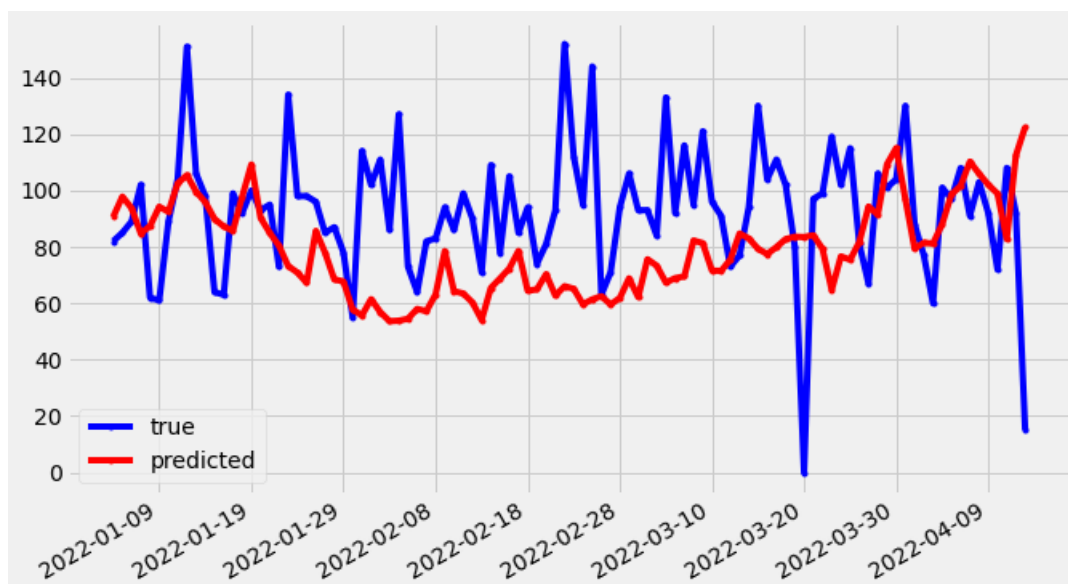
Kad būtų aiškiau sužinoti LSTM modelio veikimo tikslumą buvo pritaikyta LOSS funkcijos diagrama. Kaip galima pastebėti modelis su validuojamais duomenimis po 50 ciklo prognozavimą vykdo su ganėtinai maža paklaida 7.5 %. Tai reiškia, jog modelis geba išmokyti atpažinti tam tikras situacijas ir kaip jose elgtis.(žr. 17 pav.).



17 pav. LOSS funkcija LSTM modelio mokymo stadijose

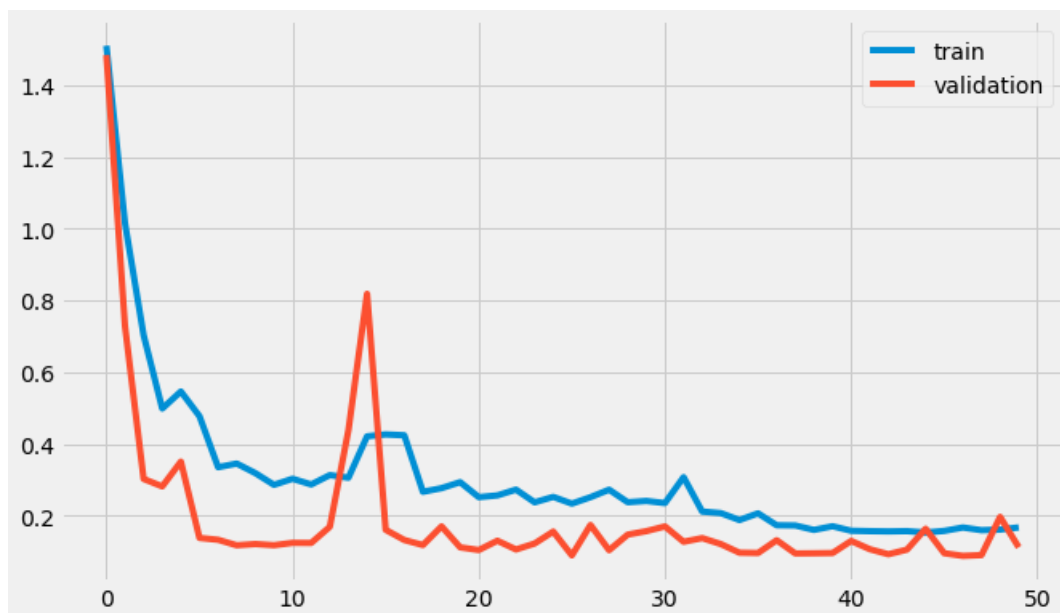
11.6.2. Abipusio GRU neuroninio tinklo taikymas nuspėti maksimalų paros keleivių skaičių

Lygiai taip pat kaip su LSTM tinklu buvo atliktas tyrimas su GRU tinklu. GRU modelio parametrai palikti tie patys išskyrus LSTM sluosnį pakeitė GRU. Gauti modelio prognozavimo rezultatai (žr. 18 pav.).



18 pav. Maksimalaus keleivių skaičiaus GRU modelio prognozavimo rezultatų bei realių reikšmių grafikai dienos atžvilgiu

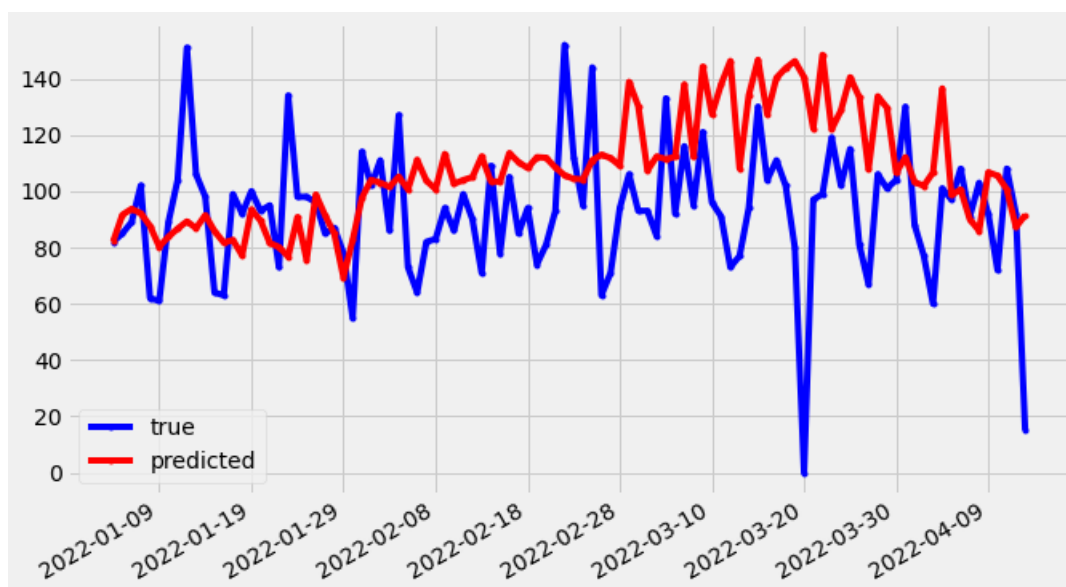
GRU modelio paklaida 11.5%. Visos modelio apmokymo paklaidos (žr. 19 pav.).



19 pav. LOSS funkcija GRU modelio mokymo stadijose

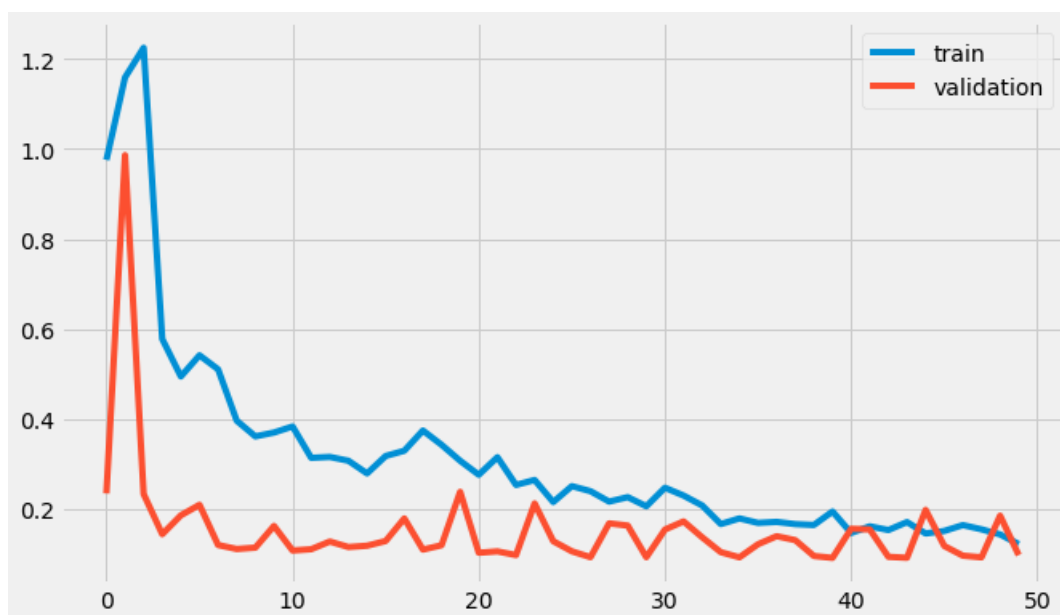
11.6.3. Abipusio RNN tinklo taikymas nuspėti maksimalų paros keleivių skaičių

Lygiai taip pat kaip su LSTM tinklu buvo atliktas tyrimas su RNN tinklu. RNN modelio parametrai palikti tie patys išskyrus LSTM sluosnį pakeitė RNN. Gauti modelio prognozavimo rezultatai (žr. 20 pav.).



20 pav. Maksimalaus keleivių skaičiaus RNN modelio prognozavimo rezultatų bei realių reikšmių grafikai dienos atžvilgiu

RNN modelio paklaida 9.7%. Visos modelio apmokymo paklaidos (žr. 21 pav.).



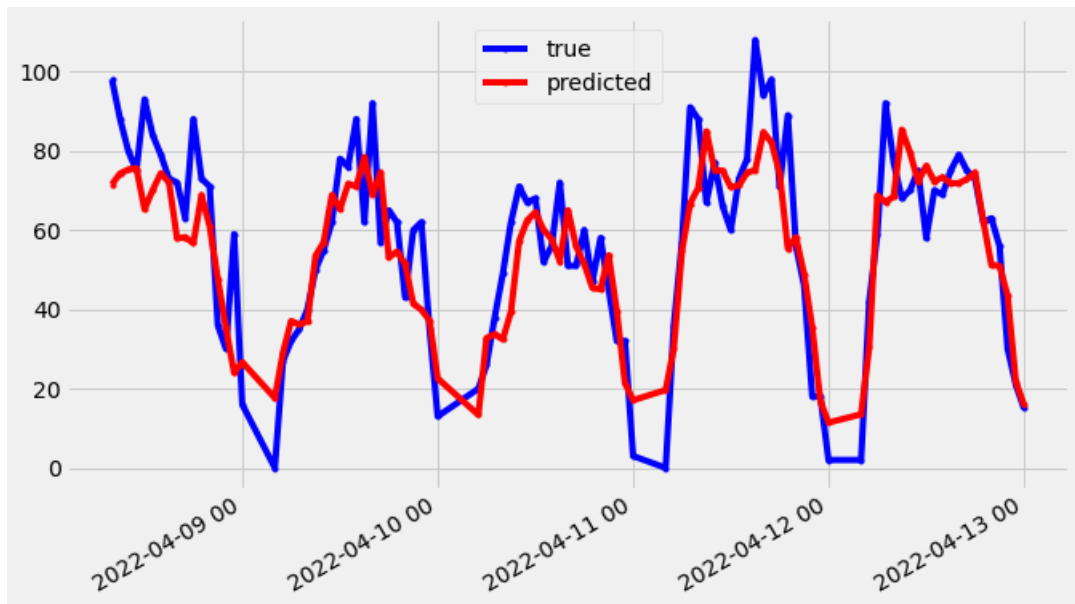
21 pav. LOSS funkcija RNN modelio mokymo stadijose

11.6.4. Abipusio LSTM neuroninio tinklo taikymas nuspėti maksimalų valandinį keleivių kiekį

LSTM modelis kuris gebėtų prognozuoti valandinius keleivių srautus buvo pritaikyti tie patys metodai kaip taip pat kaip su abipusiu LSTM modeliu kuris skirtas atlikti prognozavimą su dienos keleivių srautais. Tyrimui buvo pasitelkiama valandiniais VT duomenimis. Atliktas duomenų sugrupavimas į sekas, tik šį kartą sekų ilgis buvo pasirinktas 24, nes parą sudaro 24 valandos. Apmokymui naudojau paprastą dvikryptį LSTM neuroninio tinklo modelį. LSTM modelio parametrai:

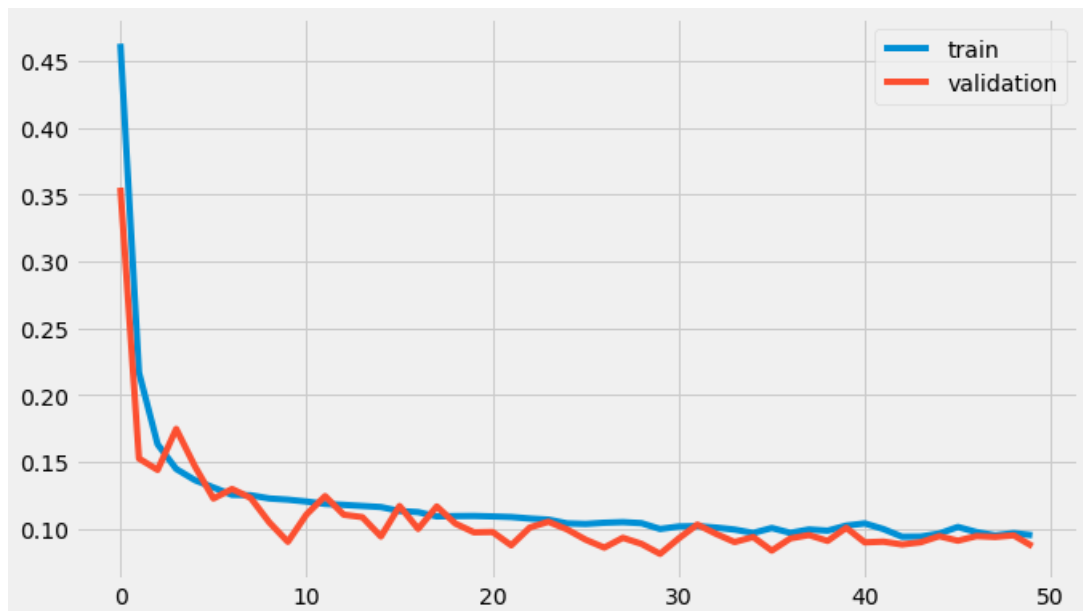
- Abipusis skaičiavimų judėjimo tinklas.
- 128 skaičiavimo vienetai (angl. "units").
- Įvesties matricos dimensijas.
- Skaičiavimo mazgų išmetimo dalis 20%.
- Pritaikyta išlygiavimo funkcija (angl. "dense").
- Skaičiavimų tikslumui nustatyti pritaikyta - "Vidutinė absoliutinė paklaida" ir "adam" optimizatorius.
- Modeliui apmokyti skirtos 50 epochų,
- Paketo iš 32 elementų
- Validavimui skirta 10% treniravimo duomenų.
- Uždraustas duomenų maišymas, nes nuoseklumas yra svarbus šiam modeliui

Apmokytas modelis gana tiksliai atliko būsimų keleivių srautų prognozę (žr. 22 pav.).



22 pav. maksimalus keleivių skaičiaus VT prognozavimo su LSTM modeliui rezultatų bei realių reikšmių grafikai paros valandų atžvilgiu

Apmokytą modelio tikslumą įvertintame apskaičiuojant LOSS funkciją (žr. 23 pav.). Paskutinė iteracijos metu modelio paklaida siekė tik 8.7 %, kas parodo, jog modelis gana tikslus. Taip pat galima išvelgti, jog grafikų su mokymo ir testavimo duomenų paklaidomis kryptingai mažėja. Tai galima pritaikyti daugiau epochų galime išmokyti modelį, kad jis gebėtų geriau prognozuoti būsimus keleivių maksimumus.



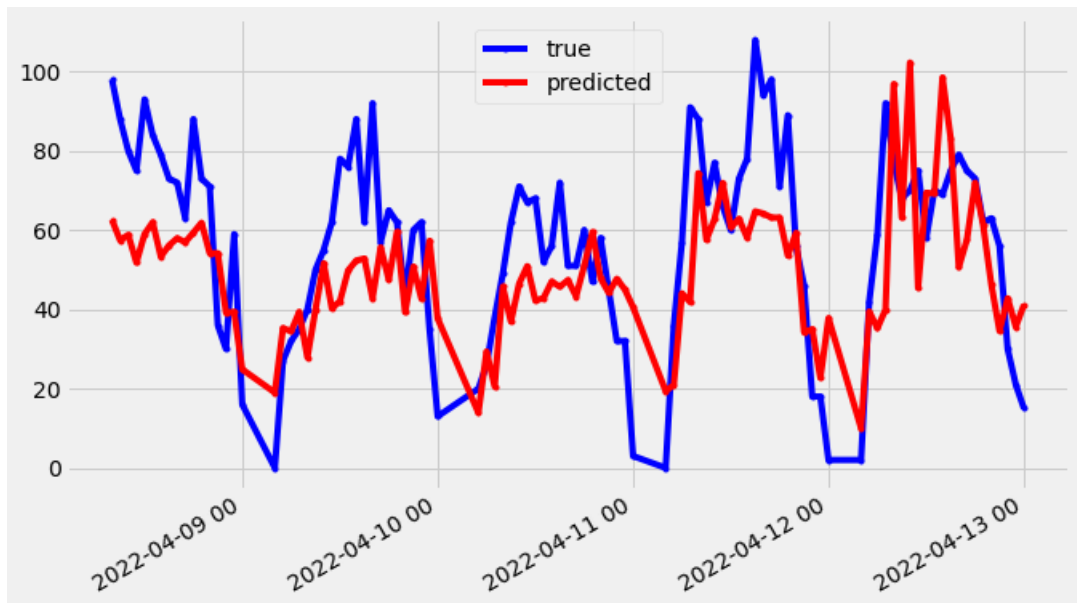
23 pav. Valandinių LSTM modelio prognozių LOSS funkcija

Abipusis LSTM apmokytas modelis turi potencialo gana tiksliai nuspėti būsimus pasirinkto maršruto valandinius maksimalius keleivių kiekius. Tai šis modelis gali būti nuolatos tobulinamas

ir papildomas naujais duomenimis ir taip būtų galima optimizuoti viešojo transporto srautus paros valandų atžvilgiu.

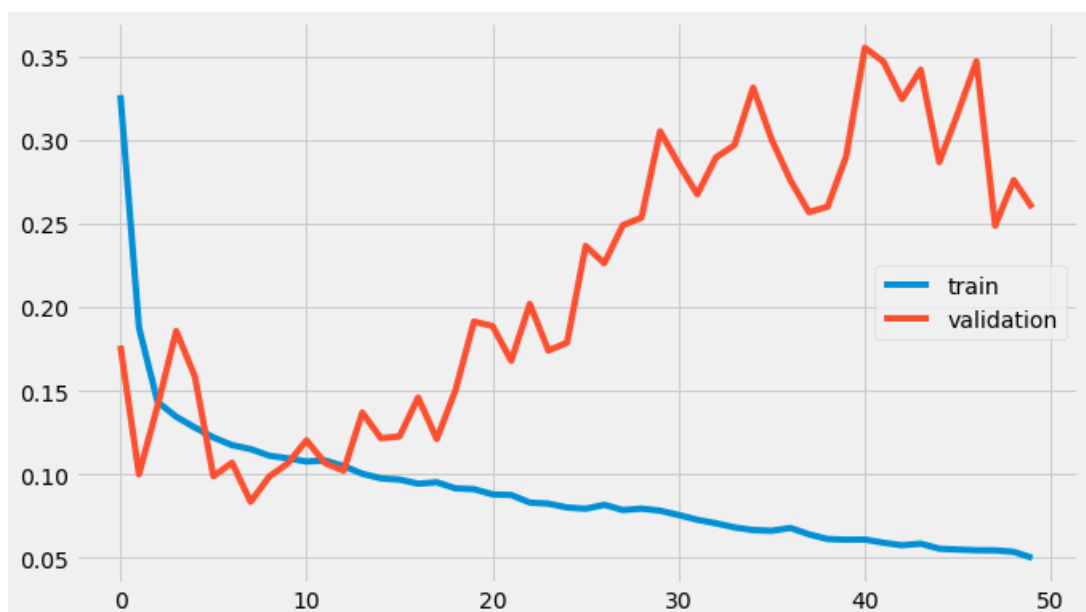
11.6.5. Abipusio GRU neuroninio tinklo taikymas nuspėti maksimalų valandinį keleivių kiekį

Bandymas buvo atliktas taip pat kaip ir abipusis LSTM modelis tik pakeistas LSTM sluoksnis į GRU (žr. 24 pav.).



24 pav. Maksimalus keleivių skaičiaus VT prognozavimo su GRU modeliu rezultatų bei realių reikšmių grafikai paros valandų atžvilgiu

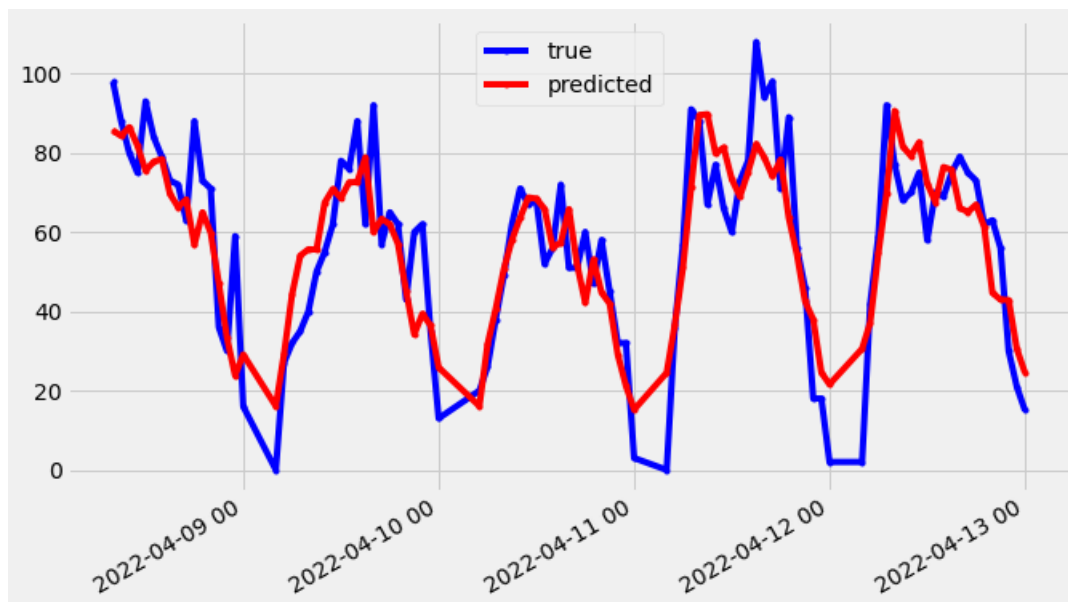
Tiriamą abipusio GRU modelio paklaida siekė 26 %. Tai reiškia, jog šis modelis prognozavimą atlieka prastai (žr. 25).



25 pav. Valandinių GRU modelio prognozių LOSS funkcija

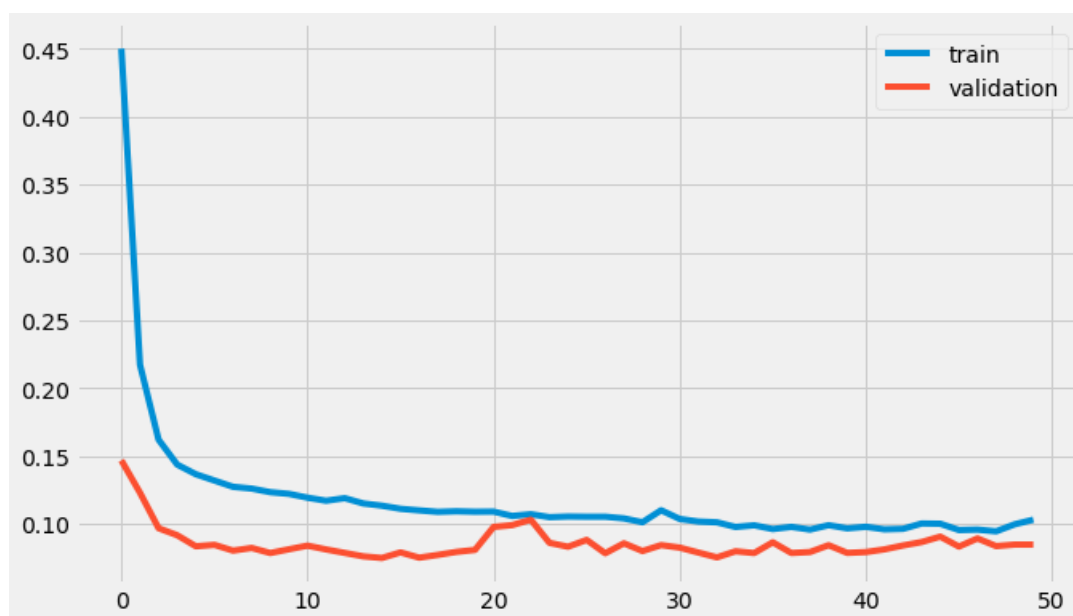
11.6.6. Abipusio RNN neuroninio tinklo taikymas nuspėti maksimalų valandinių keleivių kieki

Bandymas buvo atliktas taip pat kaip ir abipusis LSTM modelis tik pakeistas LSTM sluoksnius į RNN (žr. 26 pav.).



26 pav. Maksimalus keleivių skaičiaus VT prognozavimo Su RNN modeliu rezultatų bei realių reikšmių grafikai paros valandų atžvilgiu

Tiriamą abipusio RNN modelio paklaida siekė 8.5 %. Tai reiškia, jog šis modelis geba gana tiksliai atlikti prognozes (žr. 27).



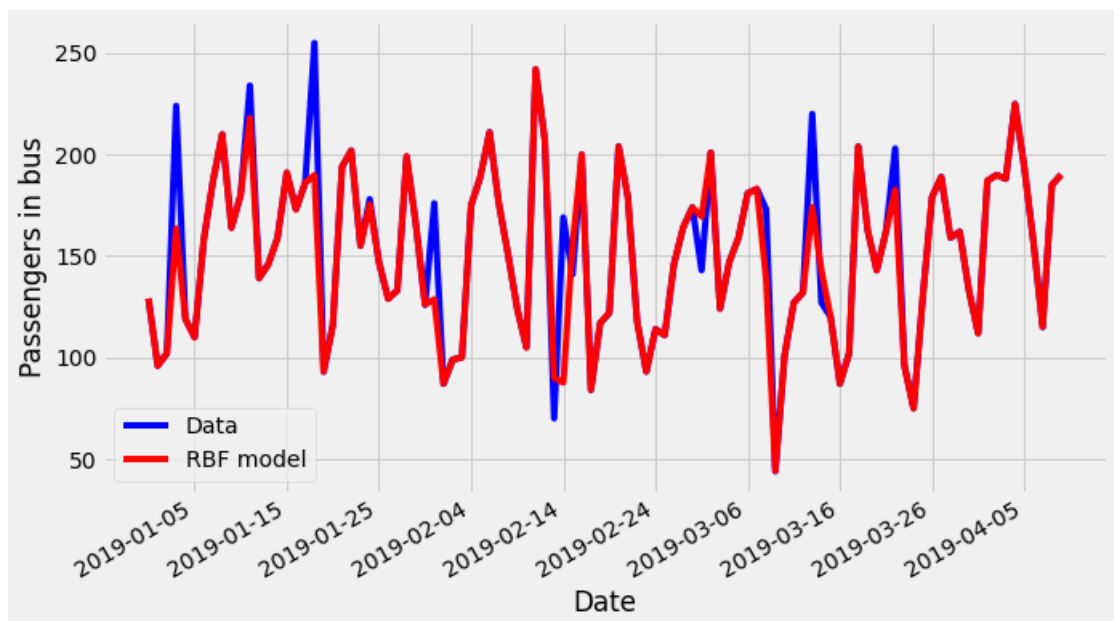
27 pav. Valandinių RNN modelio prognozių LOSS funkcija

11.7. Matematinis-statistinis neuroninių tinklų prognozuojamasis modelis

Galima pastebėti, jog modeliai neidealiai atlieka prognozes. Atlikti keli bandymai su sudėtingesniais mišriais matematiniais-statistiniais ir neuroninio tinklo modeliais.

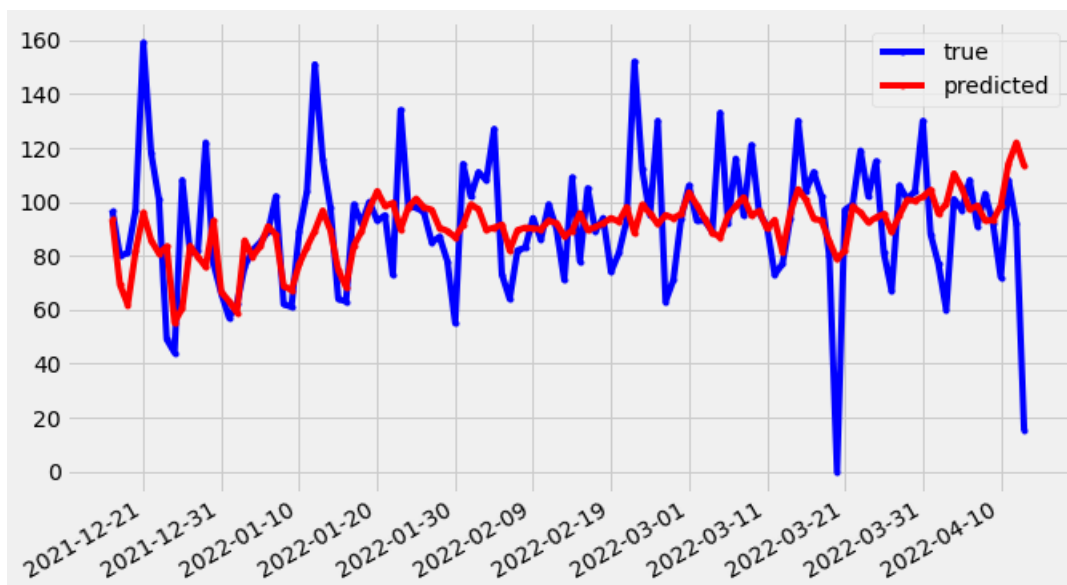
11.7.1. SVR-LSTM modelis

Atliekant modelių mokymus buvo pastebėta, jog atsiranda tokių atvejų, kada laiko grandinėje yra tokių atsitiktinių nukrypimo taškų kurie atsiranda dėl skaičiavimo klaidos ar įvykusio "neeilinio" įvykio. Nukirpimai gali pabloginti apmokymo procesą, nes neuroninis tinklas ieško dėsningumų, o staigūs nukrypimai gali išderinti dėsningumus. Šiuos nuokrypius geba pašalinti SVR modelis. SVR modeliui buvo paduota visi įrašai apie maksimalius paros keleivių srautus su papildomais parametrais: data bei savaitės diena. Dalis SVR filtro rezultatų pavaizduoti diagramoje (žr. 27 pav.).



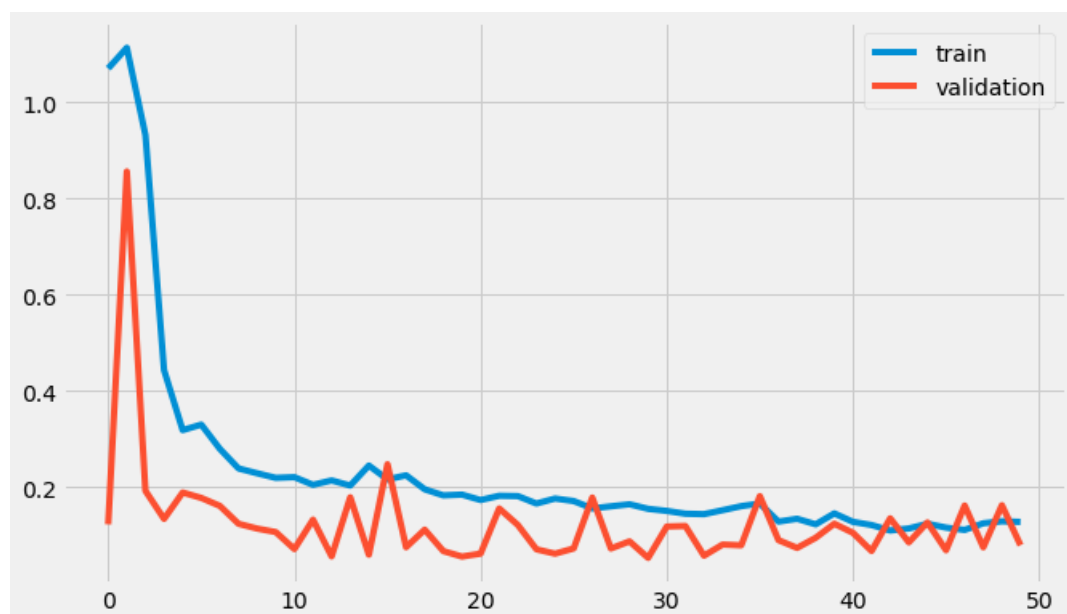
28 pav. SVR filtras

SVR modelio pakoreguoti duomenys yra užrašomi į duomenų lentelę ir paduodami į abipusį LSTM apmokytą modelį. Modelio prognozės pavaizduotos diagramoje (žr. 29 pav.).



29 pav. Maksimalus keleivių skaičiaus VT prognozavimo Su SVR-LSTM rezultatų bei realių reikšmių grafikai dienos atžvilgiu

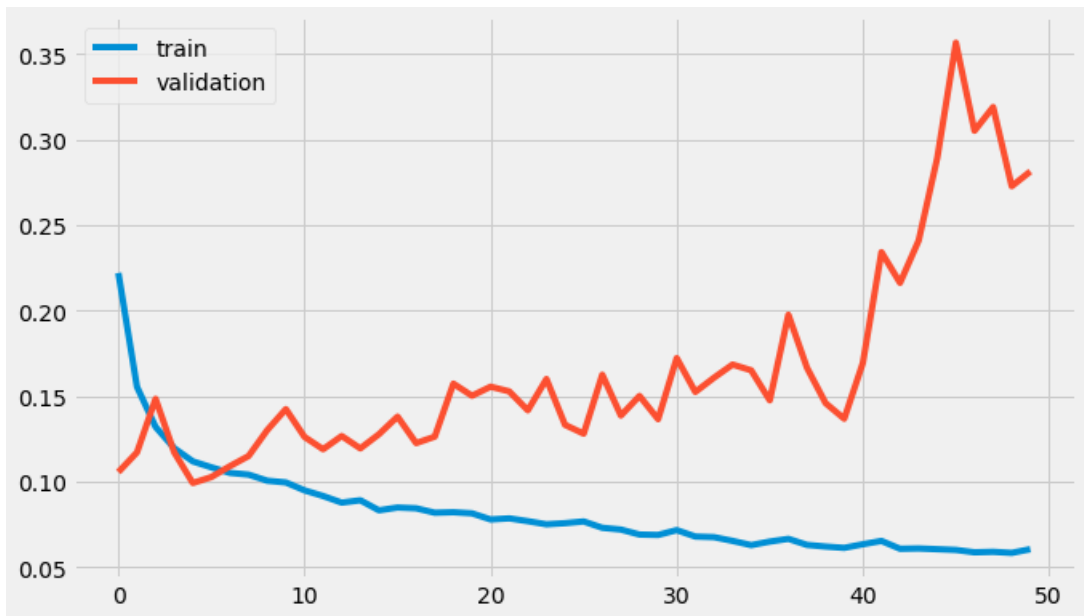
Iš diagramos galima pastebėti, SVR filtras pagerino LSTM modelio prognozavimą. Taip pat matosi, jog spėjimai yra centruoti bei išlaiko tendencijas tokias pačias kaip realūs duomenys. Modelio MSE siekė tik 7.3 % (žr. 29 pav.).



30 pav. SVR-LSTM modelio dieninių prognozių LOSS funkcija

11.7.2. Kiti modeliai

Siekiant atrasti tobulesnį apmokymo modelį buvo pritaikyti ir daugiasluoksniai DI modeliai. Bet deja nepavyko pasiekti didesnio tikslumo. Sudėtingesniems modeliams tapo modeliams tapo būdinga persimokymo problema, t. y. modelis kiekvienos metu išmoksta vis geriau testinius duomenis, bet jis "nebepastebi" esminių tendencijų ir priklosomybių, todėl ateities spėjimai - prastėja. Persimokymo problemą galima įžvelgti iš LOSS funkcijos (žr. 31 pav.).



31 pav. Sudėtingesnių modelių dieninių prognozių LOSS funkcijos pavyzdys

12. Rezultatai

Iš atliktų bandymų matosi, jog LSTM ir paprastas RNN modeliai geba tiksliausiai prognozuoti būsimus maksimalius keleivių srautus 1G maršruto autobusuose (žr. 4 lentelę). Tikėtina, jog atlikus duomenų filtravimą su SVR modeliu bei pritaikius LSTM ir RNN modelių kombinaciją, galima gauti modelį kuris prognozuotų keleivius tiksliau.

4 lentelė. Prognozuojamųjų modelių tikslumų palyginimas

Modelis	Dieninis	Valandinis
Abipusis LSTM	7.5 %	8.7 %
Abipusis GRU	11.5 %	26 %
Abipusis RNN	9.7 %	8.5 %
Abipusis SVR-LSTM	7.3 %	–

13. Išvados

Prognozuojami neuroniniai tinklai gana tiksliai prognozuoja keleivių maksimalų kiekį VT dienos ir valandos atžvilgiu. Loss MSE funkcijos paklaida siekė iki 7.3%. LSTM ir paprastas RNN modeliai keleivių srautą nustatyti dienos laiko intervale. Taip pat pastebėta, jog filtruojant reikšmes su SVR modeliu galima pasiekti didesnę LSTM modelio prognozavimo tikslumą. SVR-LSTM modelis leidžia atlikti ateities keleivių prognozes. SVR-LSTM, LSTM modelių prognozės gali būti naudojamos VT Tinklų analitikų siekiant optimizuoti keleivių srautus.

Literatūra

- [AWS07] Donna Rose Addis, Alana T Wong ir Daniel L Schacter. Remembering the past and imagining the future: common and distinct neural substrates during event construction and elaboration. *Neuropsychologia*, 45(7):1363–1377, 2007.
- [BCB14] Dzmitry Bahdanau, Kyunghyun Cho ir Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [Ced11] Avishai (Avi) Ceder. Public-transport vehicle scheduling with multi vehicle type. *Transportation Research Part C: Emerging Technologies*, 19(3):485–497, 2011. ISSN: 0968-090X. DOI: <https://doi.org/10.1016/j.trc.2010.07.007>. URL: <https://www.sciencedirect.com/science/article/pii/S0968090X10001166>.
- [CGC⁺14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho ir Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [dPL01] André de Palma ir Robin Lindsey. Optimal timetables for public transportation. *Transportation Research Part B: Methodological*, 35(8):789–813, 2001.
- [GBC16] Ian Goodfellow, Yoshua Bengio ir Aaron Courville. *Deep learning*. MIT press, 2016.
- [Gro17] Roger Grosse. Lecture 15: exploding and vanishing gradients. *University of Toronto Computer Science*, 2017.
- [GXQ⁺19] Jianyuan Guo, Zhen Xie, Yong Qin, Limin Jia ir Yaguan Wang. Short-term abnormal passenger flow prediction based on the fusion of svr and lstm. *IEEE Access*, 7:42946–42955, 2019.
- [HLZ19] Siyu Hao, Der-Horng Lee ir De Zhao. Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system. *Transportation Research Part C: Emerging Technologies*, 107:287–300, 2019.
- [KZY⁺17] Jintao Ke, Hongyu Zheng, Hai Yang ir Xiqun (Michael) Chen. Short-term forecasting of passenger demand under on-demand ride services: a spatio-temporal deep learning approach. *Transportation Research Part C: Emerging Technologies*, 85:591–608, 2017. ISSN: 0968-090X. DOI: <https://doi.org/10.1016/j.trc.2017.10.016>. URL: <https://www.sciencedirect.com/science/article/pii/S0968090X17302899>.
- [LC17] Lijuan Liu ir Rung-Ching Chen. A novel passenger flow prediction model using deep learning methods. *Transportation Research Part C: Emerging Technologies*, 84:74–91, 2017.
- [LZ14] Qi Lyu ir Jun Zhu. Revisit long short-term memory: an optimization perspective. *Advances in neural information processing systems workshop on deep Learning and representation Learning*, p.p. 1–9. Citeseer, 2014.

- [MIN20] LIETUVOS RESPUBLIKOS SVEIKATOS APSAUGOS MINISTERIJA. Dėl paslaugų teikimo privačioje erdvėje karantino metu, 2020.
- [MXM⁺14] Zhenliang Ma, Jianping Xing, Mahmoud Mesbah ir Luis Ferreira. Predicting short-term bus passenger demand using a pattern hybrid approach. *Transportation Research Part C: Emerging Technologies*, 39:148–163, 2014.
- [NP12] Arti R Naik ir SK Pathan. Weather classification and forecasting using back propagation feed-forward neural network. *International journal of scientific and research publications*, 2(12):1–3, 2012.
- [PRR95] Uwe Pape, Yean-Suk Reinecke ir Erwin Reinecke. Line network planning. *Computer-Aided Transit Scheduling*, p.p. 1–7. Springer, 1995.
- [RHW86] David E Rumelhart, Geoffrey E Hinton ir Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.
- [Ros58] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- [S4I14] Susisiekimo paslaugos S4ID. Vilniaus miesto viešojo transporto keleivių srautų analizės sistemos sukūrimas, Vilnius, Lietuva, 2014.
- [SCR⁺90] Yehuda Salu, LEONARDO G Cohen, Douglas Rose, S Sxato, CONRAD Kufta ir MARK Hallett. An improved method for localizing electric brain dipoles. *IEEE Transactions on Biomedical Engineering*, 37(7):699–705, 1990.
- [She20] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.
- [SLZ⁺20] Shouwei Sha, Jing Li, Ke Zhang, Zifan Yang, Zijian Wei, Xueyan Li ir Xin Zhu. Rnn-based subway passenger flow rolling prediction. *IEEE Access*, 8:15232–15240, 2020.
- [SNO02] Asaad Y Shamseldin, Ahmed E Nasr ir Kieran M O’connor. Comparison of different forms of the multi-layer feed-forward neural network method used for river flow forecasting. *hydrology and earth system sciences*, 6(4):671–684, 2002.
- [Vai20] Paulius Vaitekėnas. Tarp miestinių kelionių atstovai palikti nežinioje: kaip viskas keisis nuo trečiadienio ir ar gali būti stabdomi maršrutai. *LRT, Lietuvos nacionalinis radijas ir televizija*, 2020.
- [WC12] Yu Wei ir Mu-Chen Chen. Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks. *Transportation Research Part C: Emerging Technologies*, 21(1):148–162, 2012.
- [Zha03] G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.

Santrumpos

BPN - Abipusis judėjimo tinklas
BPTT - Atgalinio judėjimo algoritmas
DI - dirbtinisi intelektas
FFNN - Vienakrypčiai neuroniniai tinklas
GRU - Sulaikomo pasikartojančio vieneto
GD - Gradiento nusileidimas
LSTM - ilgalaikė trumpalaikės atminties tinklas
MSE - Kvadratinio vidurkio nuokrypio funkcija
SAE - Sukrautasis auto šifravimas
Seq2seq - Iš sekos į seką algoritmas
SGD - Stochastinis Gradiento nusileidimas
SVR - Palaikomasis vektoriaus regresavimas
RNN - Rekurentinis neuroninis tinklas