

KLAIPĖDOS UNIVERSITETAS
JŪROS TECHNOLOGIJŲ IR GAMTOS MOKSLŲ FAKULTETAS
INFORMATIKOS IR STATISTIKOS KATEDRA

ROBERTAS JURKUS

**LSTM GILIŲJŲ NEURONINIŲ TINKLŲ TYRIMAS LAIVO
EIGOS PROGNOZAVIMUI NAUDOJANT DIDŽIUOSIUS
EISMO DUOMENIS**

**LSTM DEEP NEURAL NETWORK RESEARCH FOR PREDICTION OF VESSEL
MOVEMENT USING BIG TRAFFIC DATA**

Magistro baigiamasis darbas
Techninių informacinių sistemų inžinerijos studijų programa, valstybinis kodas 6211BX020
Informatikos inžinerijos studijų kryptis

Darbo vadovas: Lekt. dokt. Julius Venskus

Klaipėda, 2020

MAGISTRO BAIGIAMOJO DARBO LYDRAŠTIS

Pildo magistro baigiamojo darbo autorius

Robertas Jurkus

(magistro baigiamojo darbo autoriaus vardas, pavardė)

LSTM giliųjų neuroninių tinklų tyrimas laivo eigos prognozavimui naudojant didžiuosius eismo duomenis

(magistro baigiamojo darbo pavadinimas lietuvių kalba)

LSTM Deep Neural Network Research for Prediction of Vessel Movement Using Big Traffic Data

(magistro baigiamojo darbo pavadinimas anglų kalba)

Patvirtinu, kad magistro baigiamasis darbas parašytas savarankiškai, nepažeidžiant kitiems asmenims priklausančių autorių teisių, visas baigiamasis magistro darbas ar jo dalis nebuvo panaudotas Klaipėdos universitete ir kitose aukštosiose mokyklose.

Robertas Jurkus

(magistro baigiamojo darbo autoriaus vardas, pavardė ir parašas)

Sutinku, kad magistro baigiamasis darbas būtų naudojamas neatlygintinai 5 m. Klaipėdos universiteto studijų procese.

Robertas Jurkus

(magistro baigiamojo darbo autoriaus vardas, pavardė ir parašas)

Pildo magistro baigiamojo darbo vadovas

Magistro baigiamąjį darbą ginti

(įrašyti – leidžiu arba neleidžiu)

lekt. dokt. Julius Venskus

(data)

(magistro baigiamojo darbo vadovo vardas, pavardė ir parašas)

Pildo katedros, kuruojančios studijų programą, administratorius (sekretorius)

Baigiamasis darbas įregistruotas katedroje

(data)

Laima Brazdeikienė

(katedros sekretorės vardas, pavardė ir parašas)

Pildo katedros, kuruojančios studijų programą, vedėjas

Magistro baigiamąjį darbą ginti

(įrašyti – leidžiu arba neleidžiu)

doc. dr. Mindaugas Kurmis

(data)

(katedros vedėjo vardas, pavardė ir parašas)

Recenzentu(-ais) skiriu

(įrašyti recenzento(ų) vardą, pavardę)

doc. dr. Mindaugas Kurmis

(data)

(katedros vedėjo vardas, pavardė ir parašas)

ANOTACIJA

Šiuolaikiniai giliojo mokymosi pagrįsti algoritmai geba susidoroti su didelės apimties duomenimis ir todėl yra ypač svarbūs automatizuojant laivo eigos prognozavimą intensyvioje laivyboje. Tai būtų viena iš pagalbinių priemonių leidžiančių stebėti, valdyti intensyvėjantį eismą jūroje ir jos dalyvius. Darbe gilusis rekurentinis tinklas realizuojamas taikant Java technologiją ir į ją integruojant vieną populiariausių neuroninių tinklų Deep Learning 4 Java biblioteką.

Pritaikius giliojo mokymo algoritmą yra sukuriama pasikartojančio tinklo architektūra (LSTM) gebanti automatizuotai nuspėti plaukiančio laivo tolimesnę judėjimo eigą. Sukurta architektūra grindžiama sekomis, kai duomenys kinta laike, todėl darbe tyrinėjami optimaliausi duomenų struktūros sudarymai ir LSTM tinklo hiperparametrai, kuriais siekiama išgauti maksimaliai tiksliausius spėjimo rezultatus. Rezultatų palyginimui buvo naudojamos skirtingos rekurentinių tinklų architektūros: standartinis LSTM, LSTM stekas, autoenkoderis (AE) ir variacinis autoenkoderis (VAE). Tyrimo metu nustatyta, kad tiksliausia prognozė atliekama su autoenkoderio architektūra. Tyrimas atliktas konkrečiai nagrinėjamo Olandijos krantinės regiono duomenų imtyje, o siūlomas algoritmas gali būti pritaikomas kaip vienas iš būdų gerinančių laivybos saugumą.

PAGRINDINIAI ŽODŽIAI: gilusis mokymas, rekurentinis neuroninis tinklas, didieji eismo duomenys, laivo eigos prognozavimas, LSTM.

ANNOTATION

Modern deep learning algorithms are able to handle large amounts of data and therefore are particularly important in automating vessel movement prediction in intensive shipping. This would be one of the support tools for monitoring, managing the increasing maritime traffic and its participants. In the work, the deep recurrent network is realized by applying Java technology and integrating into it one of the most popular library of the neural networks Deep Learning 4 Java.

Applying deep learning algorithm, a recurrent network (LSTM) is created that is able to predict the further vessel movement. The developed architectural model is based on sequences when data change over time, therefore the work investigates the most optimal data structure and network hyperparameters, which aim to obtain the most accurate prediction results. Different recurrent network architectures were used to compare the results: basic LSTM, LSTM stack, autoencoder (AE) and variational autoencoder (VAE). During the study found that the most accurate prediction is performed with an autoencoder architecture. The study was performed on a specific sample of data from the Dutch coastal region and the proposed algorithm can be applied as one of the ways to improve maritime safety.

KEY WORDS: deep learning, recurrent neural network, big traffic data, prediction of vessel movement, LSTM.

SANTRAUKA

Robertas Jurkus. LSTM giliųjų neuroninių tinklų tyrimas laivo eigos prognozavimui naudojant didžiuosius eismo duomenis. Techninių informacinių sistemų inžinerijos studijų magistro baigiamasis darbas. Studijų programos valstybinis kodas 6211BX020. Darbo vadovas: Lekt. dokt. Julius Venskus. Klaipėda, 2020: Klaipėdos universitetas, Jūros technologijų ir gamtos mokslų fakultetas, Informatikos ir statistikos katedra. Aiškinamojo rašto apimtis: 52 psl., lentelių skaičius 7 vnt., paveikslėlių skaičius 32 vnt., literatūros ir šaltinių skaičius 29 vnt.

Baigiamojo darbo tikslas – ištirti LSTM giliuosius neuroninius tinklus laivo eigos prognozavimui naudojant didžiuosius eismo duomenis. Šiame darbe tyrinėjami giliojo mokymo taikytini ir mokslinėje literatūroje publikuojami naujausi prognozavimo metodai, sukuriamos LSTM rekurentinių tinklų architektūros, parengiama jų konfigūracija, apdorojami ir struktūrizuojami laivų eismo duomenys į sekas bei atliekama gautų tyrimo rezultatų analizė.

Darbe tyrinėjami specialieji rekurentinių tinklų rūšis – LSTM, kuri sugeba išmokti ilgalaikės sekos priklausomybes. Pasirinktos skirtingos jų architektūros: standartinis LSTM, LSTM stekas, autoenkoderis ir variacinis autoenkoderis. Tyrimai parodė, kad tiksliausiai laivo eigą prognozuoti pavyksta su autoenkoderio architektūra, taikant 150 LSTM celių sluoksnį, kai sekos ilgis yra 50 laiko žingsnių (~ 4 val.).

PAGRINDINIAI ŽODŽIAI: gilusis mokymas, rekurentinis neuroninis tinklas, didieji eismo duomenys, laivo eigos prognozavimas, LSTM.

SUMMARY

Robertas Jurkus. LSTM Deep Neural Network Research for Prediction of Vessel Movement Using Big Traffic Data. Master thesis of technical information systems engineering studies. State code of the study programme 6211BX020. Supervisor: Lect. doct. Julius Venskus. Klaipėda, 2020: Klaipėda University, Faculty of Marine Technology and Natural Sciences, Department of Informatics and Statistics. Volume of explanatory letter: 52 pages, number of tables 7 units, number of pictures 32 units, number of literature and sources 29 units.

The purpose of this work is to research LSTM deep neural networks for prediction of vessel movement using big traffic data. This work reviews the latest forecasting methods applicable in deep learning and published in the scientific literature. Then, LSTM recurrent network architectures are created, prepared their configuration. After that, vessel traffic data are processed and structured into sequences. Finally, an analysis of the research results is performed.

Researched a special type of recurrent networks – LSTM, which is capable of learning long term sequence dependencies. Different architectures were chosen: basic LSTM, LSTM stack, autoencoder and variational autoencoder. Studies have shown that the most accurate prediction of vessel movement is achieved with an autoencoder architecture, while using 150 LSTM cells layer and when sequence length is 50 timesteps (about 4 hours).

KEY WORDS: deep learning, recurrent neural network, big traffic data, prediction of vessel movement, LSTM.

TERMINŲ IR SANTRUMPŲ ŽODYNĖLIS

AIS (angl. *automatic identification system*) – automatinė identifikavimo sistema, kurioje vykdoma laivų eismo stebėseną.

ArcGIS (angl. *geographic information system*) – geografinės informacijos sistema.

Autoenkoderis (angl. *autoencoder, AE*) – dirbtinio neuroninio tinklo tipas, kuris atlieka duomenų dimensijos mažinimo (angl. *dimensionality reduction*) procesą. Autoenkoderio tikslas yra išmokti duomenų rinkinio kodavimą: sumažinant duomenų matmenis ir kartu juos rekonstruoti, stengiantis išgauti kuo tikslesnius pradinės įvesties informaciją.

Laivo eiga (angl. *vessel movement*) – laivo plaukimas į priekį arba atgal, naudojant laivo varytuvus LSA [1].

ETL (angl. *extract, transform, load*) – procesai, kurie sudaryti iš duomenų bazių valdymo funkcijų ir kurių metu duomenys yra ištraukiami iš duomenų šaltinio, transformuojami bei išsaugomi.

Gilusis mokymasis (angl. *deep learning*) – tai pakankamai nauja automatinio mokymosi (*machine learning*) sritis. Ji gali būti taikoma sudėtingoms neuroninių tinklų architektūroms mokyti. Čia įprastai duomenys pateikiami keliais lygiais, todėl ypač naudingi hierarchiniams duomenims analizuoti. Giliuoju mokymusi pagrįsti algoritmai geba susidoroti su didelės apimties duomenimis. Šie algoritmai gali būti sėkmingai taikomi įvairiems duomenų tyrybos uždaviniams spręsti, pradedant nuo vaizdų klasifikavimo, baigiant natūralios kalbos ar šnekos atpažinimo ir kt.

Ilgos trumpalaikės atminties tinklas (angl. *long short-term memory, LSTM*) – dirbtinio rekurentinio neuroninio tinklo (RNN) architektūros rūšis, naudojama giliojo mokymosi srityje, kuri valdydama atminties srautus gali išvengti ilgalaikės priklausomybės problemos. Skirtingai nuo standartinių neuroninių tinklų, LSTM turi grįžtamąjį ryšį bei gali ne tik apdoroti atskirus duomenų taškus (pvz., vaizdus), bet ir visą duomenų seką (pvz., garso ar vaizdo įrašą).

Išvesties vartai (angl. *output gate*) – funkcija, vykdoma LSTM celėje, kuri kontroliuoja duomenų nuskaitymą ir išvedimą konkrečiame laiko žingsnyje [19].

IUMI (angl. *International Union of Marine Insurance*) – tarptautinė jūrų draudimo sąjunga.

Įvesties vartai (angl. *input gate*) – funkcija, vykdoma LSTM celėje, kuri kontroliuoja duomenų įtraukimą į ilgalaikės būsenos būvį [19].

Laiko eilutė (žingsnis) (angl. *timestamp*) – reiškinio stebėjimo duomenų laikinė seka [2]. Sekos duomenys dažniausiai vienas nuo kito tarpusavyje priklausomi.

Latentinė erdvė (angl. *latent space*) – abstrakti daugiamatė erdvė (vektorius), kurioje išrenkami ir suspaudžiami svarbiausi modelio požymiai. Šiuos duomenis autoenkoderis naudoja, norėdamas kiek įmanoma tiksliau rekonstruoti įvestį.

Modelio požymis (angl. *model feature*) – individuali ir išmatuojama laivo duomenų savybė, apibūdinanti stebimo reiškinių ar objekto būseną.

Nykstantis gradientas (angl. *vanishing gradient*) – klaida įvykstanti tinklo mokymosi proceso metu, kai klaidos funkcijos gradientas konkrečiame svoryje yra labai mažas, todėl tinklas gali mokytis klaidingai arba visai būti sustabdomas.

Prognozė (angl. *forecast*) – tiriamų procesų, reiškinių ir kitų objektų laukiamos ar pageidaujamos būklės ateityje mokslinis numatymas, prognozių sudarymas [2].

Regresija (angl. *regression*) – priklausomo dydžio Y sąlyginio skirstinio kurio nors parametro (dispersijos, kvantilių, skalės ir panašiai) priklausomybė nuo paaiškinančio kintamojo X (ar kelių nepriklausomų kintamųjų) reikšmių [2].

Rekurentinis neuroninis tinklas (angl. *recurrent neural network, RNN*) – dirbtinio neuroninio tinklo rūšis, kuri mokymosi procesą atlieka iš sekų. RNN geba susieti objekto vektorius pagal visą anksčiau įvestų duomenų istoriją ir pagal tai atpažinti sekančius duomenis [20].

Seka (angl. *sequence*) – funkcija f , apibrėžta natūraliųjų skaičių aibėje $N = \{1, 2, \dots\}$ [2]. Šiame darbe, tai tam tikra tvarka išdėstyta laivo eigos atkarpų aibė.

Užmaršties vartai (angl. *forget gate*) – funkcija, kuri kontroliuoja ir nusprendžia, kurie LSTM celės įvesties duomenys turėtų būti ištrinami [19].

Variacinis autoenkoderis (angl. *variational autoencoder*) – giliojo mokymosi būdas (specialus autoenkoderio tinklo tipas), kurio kodavimas yra generatyvinis. Tai reiškia, kad jo latentinė erdvė pasižymi geromis savybėmis, leidžiančiomis sugeneruoti keletą naujų duomenų iš apmokomos tinklo imties.

TURINYS

| | |
|--|----|
| PAVEIKSLĖLIŲ SĄRAŠAS..... | 10 |
| LENTELIŲ SĄRAŠAS..... | 11 |
| ĮVADAS..... | 12 |
| 1. DIDŽIAISIAIS DUOMENIMIS PAREMTŲ PROGNOZAVIMO METODŲ BEI ALGORITMŲ ANALIZĖ..... | 14 |
| 1.1 Tikimybiniai ir lingvistiniai prognozavimo metodai..... | 14 |
| 1.2 Giliojo skatinamojo mokymo algoritmas..... | 17 |
| 1.3 Giliaisiais rekurentiniais tinklais paremtos architektūros..... | 19 |
| 1.4 Skyriaus apibendrinimas..... | 23 |
| 2. DUOMENŲ IR LSTM REKURENTINIŲ TINKLŲ STRUKTŪROS KŪRIMAS..... | 24 |
| 2.1 Didžiųjų eismo duomenų tyrimas..... | 24 |
| 2.1.1 Didieji duomenys laivų sekimo informacinėje sistemoje..... | 24 |
| 2.1.2 Pramoninės laivybos metaduomenys..... | 25 |
| 2.1.3 Duomenų šaltinio apdorojimo procesai..... | 26 |
| 2.2 LSTM tinklo architektūros..... | 27 |
| 2.2.1 Standartinis LSTM..... | 29 |
| 2.2.2 LSTM stekas..... | 30 |
| 2.2.3 Autoenkoderis..... | 31 |
| 2.2.4 Variacinis autoenkoderis..... | 32 |
| 2.3 Duomenų struktūros kūrimas..... | 32 |
| 2.4 Tinklų konfigūracija..... | 35 |
| 2.5 Programinė ir techninė įranga..... | 37 |
| 2.6 Skyriaus apibendrinimas..... | 38 |
| 3. SUKURTŲ LSTM REKURENTINIŲ ARCHITEKTŪRŲ TAIKYMO EKSPERIMENTAS LAIVO EIGOS PROGNOZAVIMUI..... | 39 |
| 3.1 Eksperimentinės dalies aprašas..... | 39 |
| 3.2 Tinklų mokymosi proceso eiga..... | 42 |
| 3.3 Gautų tyrimo rezultatų palyginimas..... | 43 |
| 3.4 Skyriaus apibendrinimas..... | 47 |
| IŠVADOS..... | 49 |
| LITERATŪRA..... | 50 |
| PRIEDAI..... | 53 |
| 1. LSTM architektūrų struktūra ir sluoksnių parametrai | |
| 2. Didžiųjų AIS duomenų imties pasiskirstymo santykis | |
| 3. Giliųjų tinklų mokymosi ir validacijos progresas (laivai eigoje) | |
| 4. Laivo eigos prognozės kitimas mokymosi procese | |

PAVEIKSLĖLIŲ SĄRAŠAS

| | |
|---|----|
| 1 pav. Laivo išilginės ir šoninės krypties tikimybinis pasiskirstymas [4]..... | 15 |
| 2 pav. Gauso proceso pritaikymas eigos modeliavime [4]..... | 15 |
| 3 pav. Jūros transportą apibūdinanti schema [5] | 16 |
| 4 pav. Lingvistinės trikampių narystės funkcijos [6] | 17 |
| 5 pav. Bandymo rezultatai naudojant pagrindinę atlygio funkciją [7] | 18 |
| 6 pav. Skirtingi LSTM projekciniai tinklai [8]..... | 19 |
| 7 pav. LSTM su dėmesio mechanizmo architektūra [9]..... | 20 |
| 8 pav. Bendroji CNN, LSTM ir FFNN tinklo architektūra [10] | 21 |
| 9 pav. „Convmax-Sigmoid“ tinklo efektyvumo balai [10] | 22 |
| 10 pav. Bendroji VAE tinklo veikimo schema [15] | 22 |
| 11 pav. Didžiausi pasaulio terminalai [16]..... | 25 |
| 12 pav. Tiriamo regiono AIS eismo duomenų vizualizacija | 26 |
| 13 pav. ETL procesais paremta didžiųjų duomenų apdorojimo schema..... | 27 |
| 14 pav. LSTM celės struktūra [19]..... | 28 |
| 15 pav. Standartinė LSTM tinklo architektūra | 29 |
| 16 pav. LSTM steko architektūra | 30 |
| 17 pav. Autoenkoderio architektūra | 31 |
| 18 pav. Variacinė autoenkoderio architektūra..... | 32 |
| 19 pav. Trimatis laivo duomenų struktūros vaizdas | 33 |
| 20 pav. Stovinčių ir eigoje esančių laivo duomenų pasiskirstymo grafikas..... | 33 |
| 21 pav. Laivo sekų generavimo iliustracija..... | 34 |
| 22 pav. Laivų laiko žingsnių pasiskirstymas duomenyse..... | 34 |
| 23 pav. Duomenų rinkinio geografinė vizualizacija..... | 39 |
| 24 pav. Eksperimento bandymų schema | 40 |
| 25 pav. Laivo duomenų sekos fragmentas | 41 |
| 26 pav. Architektūrų mokymosi trukmės palyginamoji diagrama | 43 |
| 27 pav. LSTM architektūrų tikslumas įvertinant visus požymius..... | 44 |
| 28 pav. LSTM architektūrų tikslumas įvertinant koordinacių požymius | 44 |
| 29 pav. LSTM architektūrų sekos vizualizacija iš stovinčių ir eigoje esančių laivų..... | 45 |
| 30 pav. Standartinės LSTM architektūros eigos prognozė be papildomų požymių..... | 46 |
| 31 pav. LSTM architektūrų tikslumas laivų eigos duomenyse | 46 |
| 32 pav. LSTM architektūrų sekos vizualizacija be stovinčių laivų..... | 47 |

LENTELIŲ SĄRAŠAS

| | |
|--|----|
| 1 lentelė. Laivų eismo metaduomenys | 25 |
| 2 lentelė. Bendrieji LSTM tinklų mokymo hiperparametrai | 36 |
| 3 lentelė. Techninės įrangos specifikacija | 37 |
| 4 lentelė. Normalizacijos statistiniai įverčiai..... | 41 |
| 5 lentelė. Prognozavimo tikslumo vertinimas su stovinčiais laivais | 43 |
| 6 lentelė. Prognozavimo tikslumo vertinimas be stovinčių laivų..... | 46 |
| 7 lentelė. Apibendrinti architektūrų tikslumo įvertinimai | 48 |

IVADAS

Jūroje vyksta intensyvus eismas, uostuose pakraunama ir iškraunama daugybė laivų, tad informacija apie plaukiojančių laivų buvimo vietą, jų trajektorijas ir galutinius tikslus yra labai svarbi saugumui jūroje, jūrų aplinkosaugai ir ekonomikai. Tarptautinė jūrų draudimo sąjunga (IUMI) skelbia, kad 2018 m. maždaug 10 % nuostolių patyrė vien dėl įvykstančių susidūrimų. Taip pat, nuostoliai patiriami ir dėl kitų faktorių (prastai apmokyta įgula, įrangos gedimai, oro sąlygos) [3]. Štai vienas incidentas 2018 m. pab. įvyko Norvegijoje, kai Norvegijos karališkojo jūrų laivyno fregata „Helge Ingstad“ susidūrė su tanklaiviu, nes jį palaikė ant kranto pastatytomis konstrukcijomis. Prognozuojant ir numatant konkretaus laivo plaukimo eigą būtų galima įvertinti susidūrimo riziką ar pastebėti neįprastus veiksmus vykstančiame eisme.

Atliekant laivo eigos prognozavimą pagrindinis aspektas tampa nuoseklus laikas, kuris išgaunamas iš didžiųjų AIS eismo istorinių duomenų. Būtent todėl klasikiniai mašininio mokymo algoritmai, veikiantys tiesioginio sklidimo būdu, nėra tinkami, nes jie nesugeba įsiminti sekos priklausomybę turinčius duomenis. Ši problema sprendžiama, taip vadinamoje giliojo mokymosi srityje, taikant sudėtingesnes rekurentinių tinklų architektūras – LSTM tinklus. Giliojo mokymo priemonėmis galima išgauti didesnes abstrakcijas, išryškinant tam tikras duomenų rinkinio savybes, kai naudojami didieji duomenys, nes jūroje eismas fiksuojamas labai didelis, įvairus ir dažnai nestruktūrizuotas. Tinkamai struktūrizavus duomenis, sudarant tęstines, persidengiančias sekas sukuriama grandinė, kuri apmokoma LSTM architektūroje. LSTM konstrukcija geba įsiminti reikšmingiausias požymius ir regresine priklausomybe prognozuoti tolimesnes laivo požymių aibes.

Nors įvairūs prognozavimo metodai jau yra taikomi, tačiau giliojo mokymo algoritmai ir jų efektyvumas keičiant LSTM konstrukcijas šioje srityje nėra visiškai ištirtas. Ypač, kai didieji laivų duomenys yra kaupiami informacinėse sistemose (AIS), tai dirbtinių neuroninių tinklų taikymas yra novatoriškas sprendimas. Šiame darbe buvo sukurta didžiaisiais eismo duomenimis paremta duomenų konstrukcija ir realizuotos LSTM tinklų architektūros. Galiausiai, atliktas palyginimas ir rezultatais įrodytas autoenkoderio efektyvumas, sprendžiant laivo eigos regresinę problemą.

Atlikti baigiamojo darbo rezultatai bus pristatomi mokslinėje konferencijoje „Jūros ir Krantų tyrimai 2020“, stendiniame pranešime, kuris dėl Lietuvos Respublikos Vyriausybės 2020 m. kovo 12 d. priimto sprendimo (ir KU įsakymo Nr. 1-094) laikinai nukeltas į rudens laikotarpį. Dalyvavimas yra patvirtintas.

Tyrimo objektas – didžiaisiais laivų eismo duomenimis apmokyti gilieji LSTM rekurentiniai tinklai.

Baigiamojo darbo tikslas – ištirti LSTM giliuosius neuroninius tinklus laivo eigos prognozavimui naudojant didžiuosius eismo duomenis.

Siekiant įgyvendinti užsibrėžtą tikslą, keliami šie **baigiamojo darbo uždaviniai**:

1. Išnagrinėti naujausias mokslines publikacijas, analizuojančias transporto eismo prognozavimo problemos sprendimo būdus, taikant gilųjį mokymą ir parinkti metodologiją laivų eismo prognozavimo uždaviniui spręsti.

2. Sukurti giliojo rekurentinio tinklo architektūras ir metodologiją laivo plaukimo eigos prognozavimui didžiuosiuose eismo duomenyse.

3. Ištirti sukurtų LSTM architektūrų prognozavimo parametrus ir įvertinti gautus rezultatus, siekiant prognozuoti kelių laiko žingsnių laivo plaukimo kryptį.

1. DIDŽIAISIAIS DUOMENIMIS PAREMTŲ PROGNOZAVIMO METODŲ BEI ALGORITMŲ ANALIZĖ

Šiame skyriuje atliekama prognozavimo metodų bei algoritmų analizė mokslinėje literatūroje. Išskiriamos keturios nagrinėjamos metodų grupės: matematiniai modeliai, skatinamasis mokymas, prižiūrimas (iš dalies prižiūrimas) mokymas. Remiantis pateikta literatūra, pasirenkama tyrimui aktuali metodologija, galimos architektūrų konstrukcijos.

1.1 Tikimybiniai ir lingvistiniai prognozavimo metodai

Prognozavimo metodai remiasi žiniomis apie objekto būklę praeityje ir dabartyje, veikiančių veiksnių pokyčiais. Nors prognozavimas visada susijęs su neapibrėžtumu ir neduoda visiškai tikslių rezultatų (turi paklaidą) [2], tačiau tai svarbi funkcija, padedanti planavimo ar monitoringo srityje, ypač stebint jūroje vykstantį eismą, jų cikliškumus, sezoninius svyravimus. Sudarant trumpalaikę ar ilgalaikę prognozę galima stiprinti saugumą jūroje, nes žinant faktinę ir planuojamą tolimesnę kurso eigą, lengviau identifikuoti nenumatytus incidentus ar net išvengti susidūrimus. Šiems metodams spręsti yra naudojami regresinės analizės modeliai, pritaikyti giliajame mokyme, kurie tiesiogiai priklausomi nuo didžiųjų duomenų ir jų ryšių priklausomybės.

H. Rongo, A.P. Teixeiraso ir C. Guedeso straipsnyje siūlomas tikimybinis eigos prognozavimo modelis. Laivo eigos spėjimo modelis, gali užfiksuoti laivo judėjimo charakteristikas ir įvertinti jų būsimas padėtis, taip praturtinant jūros stebėjimo paslaugas, remiančiais sprendimų priėmimu, įskaitant tikslo stebėjimą, susidūrimo išvengimą ir neįprasto eismo elgesio nustatymą [4]. Straipsnyje laivo judėjimo prognozė išskaidoma į šoninę ir išilginę kryptis, o tolimesnė buvimo vietos tikimybė apskaičiuojama išilgai šių dviejų krypčių. Naudojamas Bajeso modelis, kuris pagrįstas Gauso procesu, kad būtų galima rasti šoninės krypties neapibrėžtumą, tuo tarpu išilginė kryptis išgaunama iš laivo pagreičio. Teigiama, kad Gauso proceso regresijos modelis yra tinkamas ilgalaikiam eigos numatymui dėl sugebėjimo mokyti duomenis ir modeliuoti ateityje pastebimus kontrolės įvesties pokyčius arba atsižvelgti į maršruto savybes, pavyzdžiui, statines kliūtis arba posūkio ruožus, kurie turi įtakos laivo elgsenai.

Gauso proceso naudojimo pranašumas, nes patogiu naudoti vidurkių $m(x)$ ir kovariacines $k(x, x')$ funkcijas. Galima užrašyti funkcija (1).

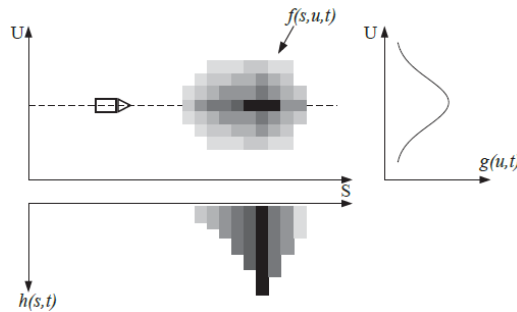
$$f(x) \sim GP(m(x), k(x, x')) \quad (1)$$

$$k(x_p, x_q) = \sigma_f \exp\left(-\frac{\|x_p - x_q\|}{2l}\right) \quad (2)$$

čia x – išilginė laivo padėtis, x' – šoninė laivo padėtis, l – ilgio skalės parametras, $./.$ Euklido atstumas. Hiperparametrais vadinami σ_f ir l kintamieji.

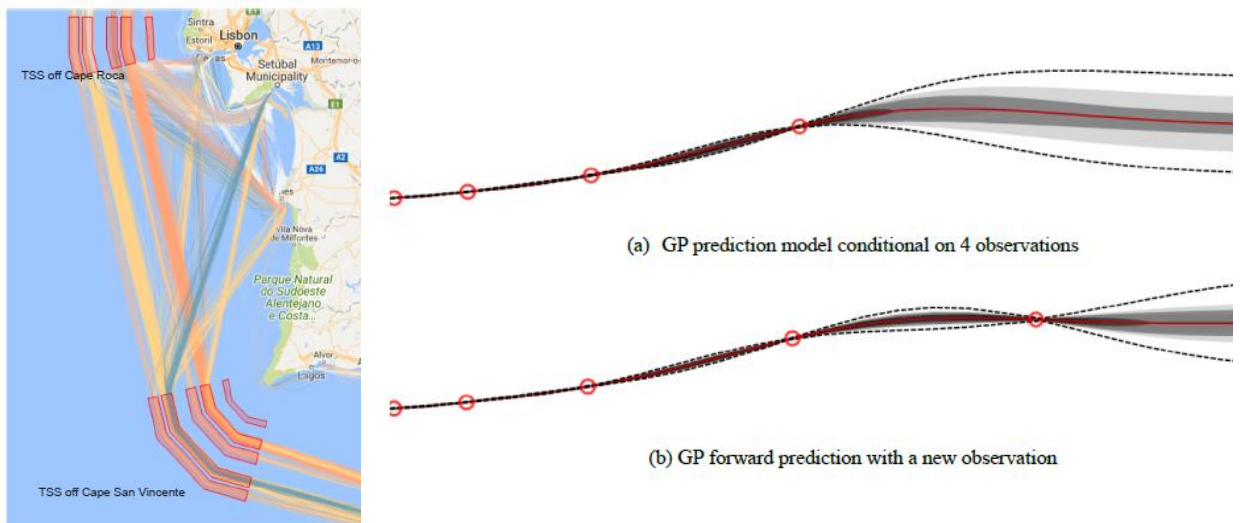
Tokiu principu (kvadrato eksponentinė kovariacinė funkcija (2)) apskaičiuojama galutinė Gauso procesu paremta kovariacinė matrica $K(X, X)$ ir išreikšta per gramo matricą (angl. *Gram matrix*).

Eigos prognozavimas remiasi AIS istoriniais trijų mėn. duomenimis. Iš geografinių koordinatų ir pagreičio apskaičiuojamos vienmatės tikimybinės tankio funkcijos. Funkcija $h(s, t)$ ir $g(u, t)$ vaizduoja šoninę (u) ir išilginę (s) tikimybę, kur t – diskretus laiko momentas. Laivo padėties jungtinė tankio funkcija $f(s, u, t)$ gaunama sudauginus h ir g vektorius. Pateikiamas ankstesnių AIS duomenų histogramos vaizdas (žr. 1 pav.).



1 pav. Laivo išilginės ir šoninės krypties tikimybinis pasiskirstymas [4]

Straipsnio bandymai atlikti žemyninės Portugalijos pakrantėje (žr. 2 pav.). Paveikslėlyje raudoni taškai nurodo laivo stebėjimo vietą, raudona linija yra nuspėjama eiga, brūkšninė linija vaizduoja standartinę nuokrypį nuo vidurkio, o pilka spalva rodo laivo buvimo vietos tikimybę. Pastebima, kad jei laivas nebuvo stebimas ilgesnį laiką, neapibrėžtumas didėja, todėl plečiasi nuspėjimo ribos, kol į Gauso proceso prognozavimo modelį yra įtraukti nauji stebėjimai.

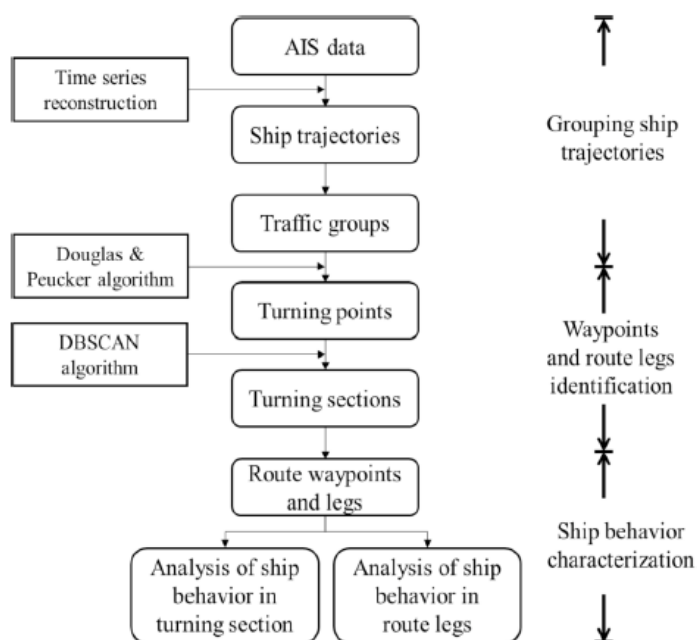


2 pav. Gauso proceso pritaikymas eigos modeliavime [4]

Autoriai atkreipia dėmesį, kad nors Gauso procesas pritaikomas neapibrėžtiniam prognozavimui ir teoriniam gerųjų analitinių savybių tyrimui, tačiau pagrindinis šio modelio trūkumas – dideli skaičiavimo kaštai. Ypač jo pritaikymas didžiųjų duomenų ir realaus laiko programose, nes skaičiavimo sudėtingumas didėja $O(N^3)$, didėjant mėginių skaičiui.

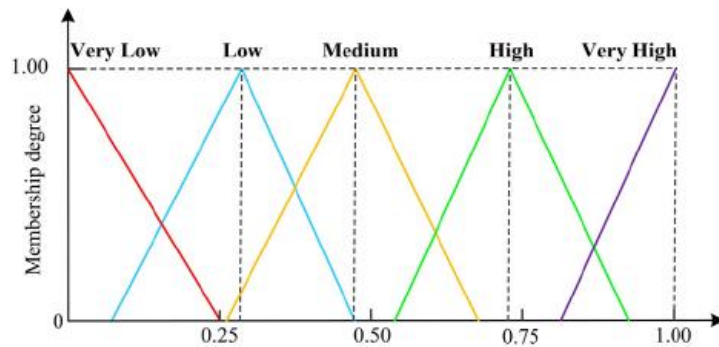
Naujausiame leidinyje, tie patys autoriai skelbia, kad nustatant laivybos maršrutus, svarbu automatiškai sugrupuoti istorinius duomenis (laivų tipą, dydžius, galutinių kelionių tikslus ir kt. charakteristikas) bei identifikuoti reikšmingus laivų navigacinius elgsenos pokyčius, anomalijas [5]. Siūloma jūros transportą apibūdinanti schema (žr. 3 pav.):

- 1) Apibrėžti transporto srautus, sugrupuojant laivų eigas pagal galutinės paskirties vietas;
- 2) Identifikuoti pagrindinius maršruto taškus ir ruožus;
- 3) Atlikti laivo elgsenos charakteristikos analizę maršruto taškuose ir maršruto ruožuose.



3 pav. Jūros transportą apibūdinanti schema [5]

Autonominių laivų saugiam manevravimui, nuspėjant kurso kryptį, naudojami ir lingvistiniai metodai. Štai pagrindinio autoriaus Jie Xue publikacijoje pateikiamas, keletą požymių apjungiantis ir manevravimo sprendimą priimantis eksperimentas, taikant pilkosios ir neraiškosios logikos teorijas (angl. *grey and fuzzy theories*) [6]. Prognozuojant saugų laivo eigos parinkimą yra atsižvelgiama į keletą veiksnių: natūralią eismo aplinką, laivo judėjimą, jėgos parametrus ir grimzlės padėtį. Įvertinant ekspertų patirtį ir galimus eismo scenarijus sudaromi neraiškosios logikos kiekybiniai trikampių narystės funkcijos (žr. 4 pav.).



4 pav. Lingvistinės trikampių narystės funkcijos [6]

Atliktas eksperimentas yra simuliuojamoje aplinkoje, todėl labiau skirtas tyrinėti autonominio laivo ir žmogaus priimtų sprendimų palyginimui, kai apjungiami daugiakriteriniai aplinkos veiksniai. Vis dėlto, metodas leidžia nustatyti labiausiai įtakojančio kriterijaus svorį ir pagal tai prognozuoti judėjimo manevrą. Prognozuojant transporto eismą svarbu išsiaiškinti kokie veiksniai turi didžiausią įtaką.

1.2 Giliojo skatinamojo mokymo algoritmas

Dauguma matematinių algoritmų yra pernelyg sudėtingos, kad jas būtų galima pritaikyti praktikoje. Kompaktiškesni sprendimai priimami taikant mašininio mokymo detalesnę komponentę – gilųjį mokymą. Yin Chengas ir Weidong Zhang sukūrė ir pritaikė giliojo skatinamojo mokymosi algoritimą (CDRLOA) nepilotuojamų laivų kliūčių vengime, gebantį prognozuoti kliūtis eigą ir parinkti tinkamą veiksmą, kad būtų išvengta susidūrimo bei pasiekiamas tikslas [7].

Algoritme apjungiamas Q-mokymosi (angl. *Q-learning*) algoritmas su konvoliuciniu tinklu. Kliūčių vengimo funkcionalumas realizuojamas ribota kontrolės strategijų seka. Skatinamojo mokymo pagalba yra suprojektuojama laivo diskretinio valdymo dėsnis T (3), remiantis jo dabartiniais stebėjimais $obs_t \in \mathbb{R}^{N_{obs}}$ ir istoriniais stebėjimais obs_{t-1} , kai N_{obs} yra parametru skaičius kiekviename tyrime, o T_p yra istoriniai stebėjimo laiko žingsniai:

$$T_t = f_{CDRLOA} (obs_t, obs_{t-1}, \dots, obs_{t-T_p}) \in \mathbb{R}^2 \quad (3)$$

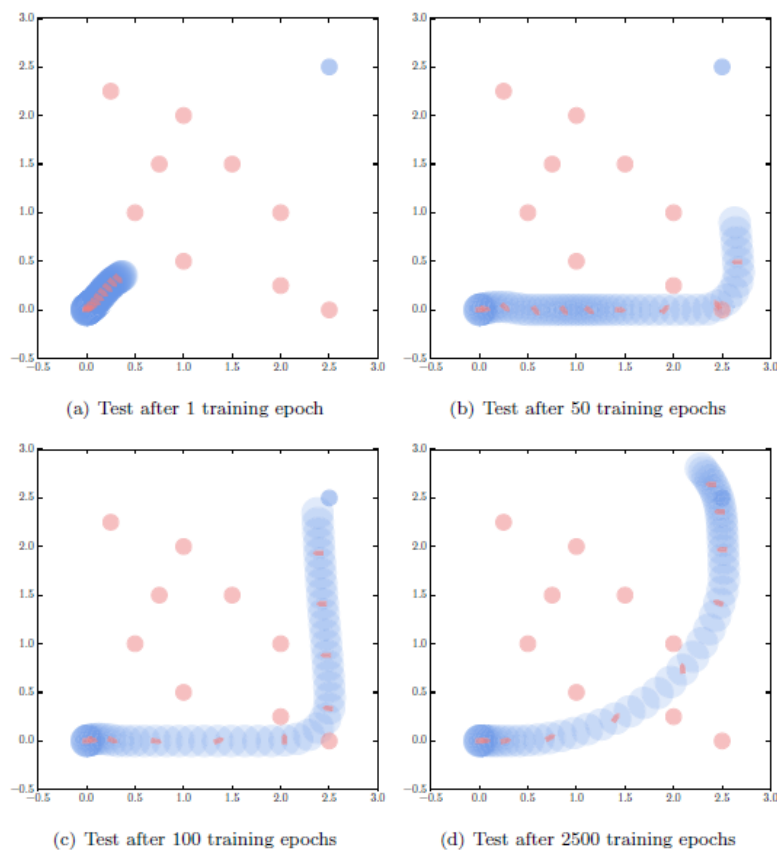
$$obs_t = [\sigma_t, v_t, \eta_t, T_{t-1}] \in \mathbb{R}^9 \quad (4)$$

Stebėjimo duomenis sudaro (4): istorinis stebėsenos vektorius (σ_t), greičio vektorius (v_t), pozicijos vektorius (η_t), paskutinio momento kontrolės elgesys (T_{t-1}). Loginis kintamasis σ_t parodo, ar tam tikrame spindulyje yra kliūčių. Naudojamas sprendimas įdomus tuo, kad į tinklą paduodamos stebėjimų sekos, o tinklo išvestis naudojama atlygio funkcijai paskaičiuoti, kurią

sudaro: atstumas iki tikslo, paskaičiavimai iki artimiausios kliūtis, saugaus laivo greičio vektorius ir bangavimo vektorius (5):

$$R(obs_t) = \lambda^T R = \begin{bmatrix} \lambda_{distance} \\ \lambda_{collision} \\ \lambda_{end} \\ \lambda_{drift} \end{bmatrix}^T \begin{bmatrix} R_{distance}(x_t, y_t) \\ R_{collision}(x_t, y_t) \\ R_{end}(x_t, y_t, u_t, v_t) \\ R_{drift}(u_t, v_t) \end{bmatrix} \quad (5)$$

Mokymo procesas atliekamas tokia eiga: stebima aplinka, įvertinama būseną, parenkamas geriausias veiksmas, duodama teigiama arba neigiama paskata. Pastebima, kad po tam tikro laiko, artėdamas prie kliūtis laivas mažina greitį ir didina posūkio kampą.



5 pav. Bandymo rezultatai naudojant pagrindinę atlygio funkciją [7]

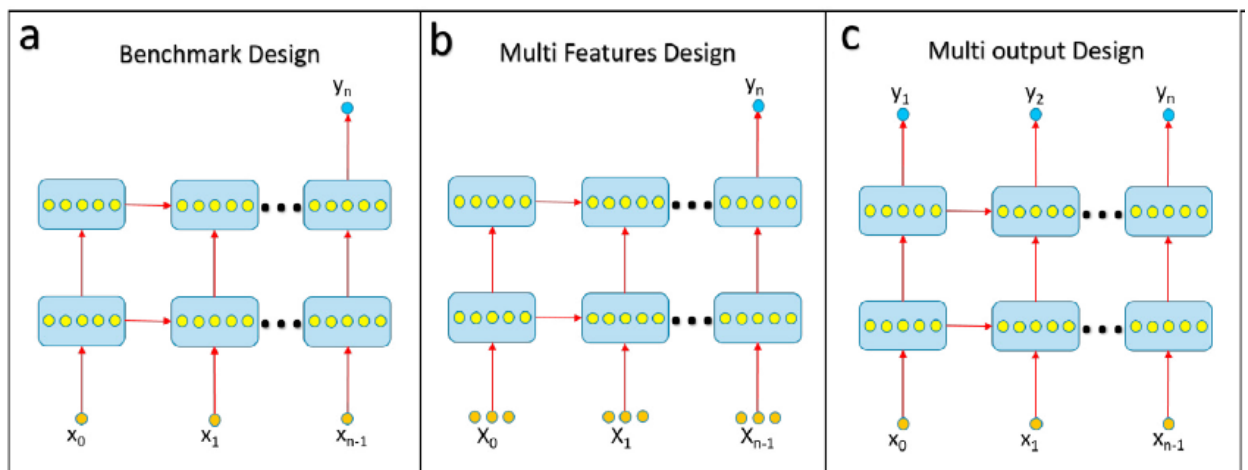
Pavaizduotame 5 paveikslėlyje pateikiamas algoritmo mokymosi procesas. Raudoni taškai žymi kliūtis, šviesiai mėlsvas taškas yra tikslas, o sujungti mėlyni ovalai rodo laivo judėjimą ir eigos parinkimą skirtingais bandymų momentais (epochomis).

Taikant siūlomą CDRLOA architektūrą, laivas imasi veiksmų remdamasis savo stebėseną, įskaitant kliūčių aptikimo rezultatus ir pagrindinius laivo eksploataavimo būsenų matavimus. Šio būdo pritaikymas realioje laivyboje būtų sudėtingesnis, nes pirmiausiai algoritmas turėtų būti apmokomas praktiškai, tačiau tai puikus būdas rinkti informaciją apie aplinką ir sąveika su ja.

1.3 Giliaisiais rekurentiniais tinklais paremtos architektūros

Prognozavimo problemą sprendžia Xu Liu ir jo kolegos, savo darbe norėdami sukurti dviračių dalijimosi galimybę ir nuspėti jų pasiekiamumą. Dėl didėjančių transporto srautų mieste didėja užterštumas, todėl lengvas dviračių radimas, žmones skatintų anglies dvideginio išmetimo mažinimą ir stiprintų jų sveikatą. Iki šiol dviračių bendro naudojimo paslaugas teikiančioms įmonėms, tai padaryti buvo sudėtinga, todėl nuspręstą panaudoti LSTM neuroninius tinklus. Darbo tikslas – numatyti turimus dviračius skirtingais laiko tarpais. Publikacijoje pateikiamos dvi LSTM konfigūracijos, įrodančios prognozavimo tikslumo pagerinimą, atlikus tinklo modifikacijas (žr. 6 pav.) nei standartinė LSTM architektūra (a) [8]:

- (b) į modelį įtraukiami nauji/papildomi įvesties požymiai (angl. *features*) (savaitės diena ir paros valanda);
- (c) naudoja kelių laiko žingsnių (angl. *multiple time steps*) spėjimą kaip išvestį.



6 pav. Skirtingi LSTM projekciniai tinklai [8]

Giliųjų neuroninių tinklų koncepcija (daugiasluoksnis perceptronas) sukurtas tinklas su dvejais paslėptais sluoksniais, kurių neuronai sujungti vieni su kitais. Teigiama, kad naudoti vien tokią koncepciją būtų klaidinga, nes struktūra yra fiksuota, nauji požymiai nėra įsimenami iš vėlesnių žingsnių. Laiko sekų problemą sprendžia RNN tinklai, kurie gali atlikti prognozę pagal ankstesnio laiko žingsnio rezultatus.

Tyrimo metu tarpusavyje palygintos trys architektūros ir kurių tikslumas paskaičiuotas trejomis metrikomis [10]: vidutinę kvadrato paklaidą (6) (MSE), vidutinę absoliučią paklaidą (7) (MAE) ir vidutinę absoliučią procentinę paklaidą (8) (MAPE). Gauti rezultatai kelis kartus geresni nei su standartiniu LSTM tinklu, tiek detalizavus modelį naujais požymiais, tiek tinklo apmokymo metu naudojant kelių laiko žingsnių išvesties paskutinįjį vektorį kaip sekancio mokymo įvestį. Rekurentinio tinklo efektyvumas dažniausiai vertinamas aukščiau aprašytomis matematinėmis

formulėmis bei kvadratinės šaknies vidutine paklaida (9) (RMSE). Jų formuluotės pateiktos žemiau, kur R_i – tikroji vertė, P_i – prognozuota vertė, W_i – prognozuotos vertės svoris, s – imties mėginių skaičius.

$$MSE = \frac{1}{s} \sum_{i=1}^s w_i (r_i - p_i)^2 \quad (6)$$

$$MAE = \frac{1}{s} \sum_{i=1}^s w_i |r_i - p_i| \quad (7)$$

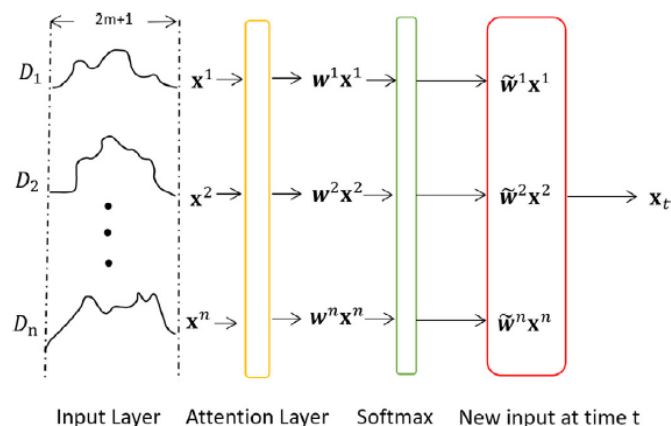
$$MAPE = \frac{100\%}{s} \sum_{i=1}^s \frac{w_i |r_i - p_i|}{p_i} \quad (8)$$

$$RMSE = \sqrt{\frac{1}{s} \sum_{i=1}^s w_i (r_i - p_i)^2} \quad (9)$$

Nors autoriai pateikė įrodančius prognozavimo pranašumus naudojant daugiamačius duomenų modelius su kelių laiko žingsnių išvestimi, tačiau patys teigė, kad būtų galima plėsti šį tyrimą įtraukiant papildomus faktorius: orus, sezoniškumą, laisvas/šventines dienas.

B. Yang ir S. Sun teigia, kad didėjant LSTM sekos ilgiui, didėja ir jos paklaida. Šios ilgalaikės priklausomybės problemai spręsti siūlomi sprendimai: duomenų apdorojimo metu mažinamas triukšmas remiantis tendencijomis; panaudoti tik didelę įtaką turinčias ypatybes sekose, kad kompensuotų mokymosi tikslumą ypač ilgose sekose [9].

Pasirinktas tyrimo objektas – transporto srutai. Darbe naudojamas taip vadinamas dėmesio mechanizmas (angl. *attention mechanism*). Tai vienas iš tinklo architektūros komponentų, kuris atsakingas už tarpusavio priklausomybės valdymą ir kiekybinį įvertinimą: tarp įvesties ir išvesties elementų.



7 pav. LSTM su dėmesio mechanizmo architektūra [9]

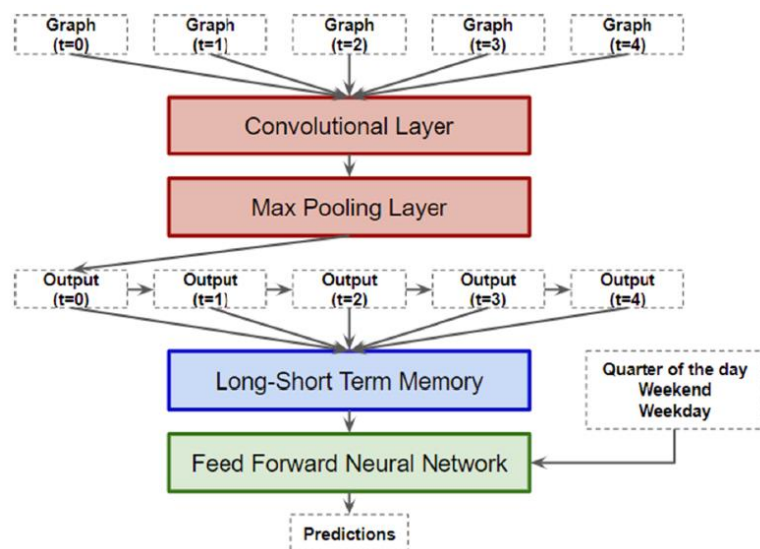
LSTM tinklas padalijamas į keturis sluoksnius (žr. 7 pav.): dėmesio mechanizmo sluoksnį, mišrų įvesties sluoksnį, paslėptą sluoksnį ir išvesties sluoksnį. Dėmesio mechanizmo sluoksnyje

naudojama daugialypė tiesinė regresija, kuri perskaičiuoja kiekvieno požymio svorio vertę, pagal nuspėjamą prognozę. Tokiu būdu svoriai įvertinami (kiekybiškai) kiekvieno laiko intervalo metu ir daroma prielaida, kad didesnės jų vertės yra svarbesnės ir turi didesnę įtaką prognozuojamai vertei. Nustatyta, kad išrinkus svarbiausius požymius ir sumažinus triukšmo lygį duomenyse, galima padidinti tikslumą ilgesnėse transporto sekų prognozėse.

Istoriniai transporto sistemų duomenys nagrinėjami T. Bogaerts'o ir bendraautorių straipsnyje, siekiant sumažinti eismo spūstis, prognozuojant eismo sąlygas kelyje. Pagrindiniai darbe atliekami bandymai [10]:

- veiksminga duomenų mažinimo technika, leidžianti pasirinkti tinkamiausias kelių jungtis tiek trumpalaikiam, tiek ilgalaikiam eismo prognozavimui;
- apjungta grafinė konvoliucinio ir rekurentinio (CNN-RNN) tinklo architektūra, pagrįsta eigos duomenimis su dviem skirtingais CNN variantų sluoksniais, siekiant dar labiau sumažinti įvestų duomenų matmenis neprarandant svarbios informacijos.

Tinklas apmokytas ir išbandytas naudojant nedidelę duomenų imtį iš GPS palydovų. Tinklo architektūra sudaryta iš 3 pagrindinių komponentų: konvoliucinis sluoksnis, LSTM sluoksnis, tiesioginio sklidimo (angl. *feed forward*, FFNN) neuroninis tinklas. Pirmame sluoksnyje sudaromos m žingsnių sekos iš istorinių duomenų, pritaikomi surinkimo (angl. *pooling*) filtrai, kurie sumažina duomenų matmenis. LSTM sluoksnis gautas matricas naudoja tam, kad išvengtų ilgalaikės priklausomybės tarp srauto duomenų ir erdvinės padėties. Galiausiai rezultatai praeina pro FFNN, kuriame apjungiami pasirinkti kontekstiniai duomenys (pvz., ketvirčiu, savaitgaliu, savaitės diena), kad galėtų atlikti spėjimą laike. Pateikiama sukurto giliojo tinklo architektūra (žr. 8 pav.).



8 pav. Bendroji CNN, LSTM ir FFNN tinklo architektūra [10]

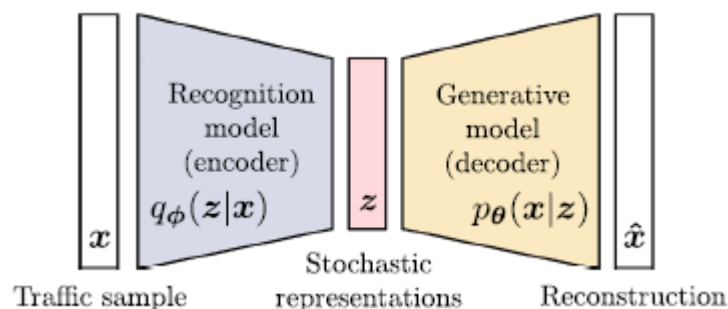
Norėdami išgauti tikslesnius prognozavimo rezultatus, autoriai apjungia keletą architektūrų. Derinant siūlomus algoritmus gaunami geresni rezultatai nei naudojami algoritmai, kurie buvo pateikti TRANSFOR19 prognozavimo varžybose. Sukurta architektūra geba atlikti kelių žingsnių prognozę (trumpalaikę, ilgalaikę) nuo 5 min. iki 4 val. Duomenų mažinimo technika pagrįsta laiko koreliacija leido pagreitinti algoritmo vykdymą. Tinklo efektyvumo įvertinimas pateikiamas 9 paveikslėlyje, kuriame matyti, kad prognozė atliekama tiksliau, nei naudojant standartinę LSTM architektūrą.

| 5 min | Rank | MAE |
|-----------------|-------|-------------------|
| Convmax-sigmoid | 1.417 | 3.430/2.315/1.201 |
| k-NN | 4.833 | 3.797/2.592/1.388 |
| LSTM | 2.833 | 3.495/2.363/1.231 |
| LSTM same info | 2.25 | 3.479/2.352/1.226 |
| SVM | 3.666 | 3.653/2.499/1.345 |
| 4 h multistep | Rank | MAE |
| Convmax-Sigmoid | 1.583 | 3.430/2.315/1.201 |
| k-NN* | 4.917 | 3.801/2.601/1.400 |
| LSTM* | 2.583 | 3.496/2.363/1.230 |
| LSTM same info* | 2.167 | 3.486/2.356/1.227 |
| SVM* | 3.75 | 3.672/2.514/1.356 |

9 pav. „Convmax-Sigmoid“ tinklo efektyvumo balai [10]

Literatūroje dažnai naudojami konvoliuciniai tinklai apjungti su LSTM architektūra, sprendžiant transporto eismo problemas ir prognozes [11], [12].

Neįprastą sprendimą pateikia Guillem‘as Boquet‘as su komanda, kurie panaudojo kodavimo ir dekodavimo VAE architektūros taikymą (žr. 10 pav.), siekiant prognozuoti kelių eismą. VAE tinklas leidžia duomenis užkoduoti į daugiamatį skirstinį ir sugeneruoti naujas realizacijas, vėliau jas dekoduoti į naujus sudėtingos struktūros duomenis, pagrįstus duomenų rinkiniais. Duomenų suspaudimui naudojamas daugiamatis Gauso modelis su įstrižine kovariacijos struktūra $q_{\phi}(z|x)$, o rekonstrukcija remiasi pakartotinio dauginimo principu $p_{\theta}(x|z)$ [13]. Šios technikos naudojimas gali sumažinti duomenų matmenis neprarandant svarbios informacijos, nes tinklas įsimena rekonstrukcijas iš tikimybinio pasiskirstymo.



10 pav. Bendroji VAE tinklo veikimo schema [13]

1.4 Skyriaus apibendrinimas

Atlikus literatūros analizę pastebima, kad mokslininkai intensyviai dirba šioje srityje, tačiau skirtingas LSTM architektūrų ir jos struktūros palyginimas jūros eisme dar mažai tyrinėtas. Yra įvairių giliojo mokymo metodų (matematinų, skatinamojo, prižiūrimo) taikymo būdų ir jų variacijų, sprendžiančių prognozavimo problemas. Viena dažniausiai regresijoje taikomų tinklų architektūrų yra LSTM, kuri plačiai naudojama apdoroti ir numatyti įvykius pagal laiko eilutes [9] bei galinti išspręsti ilgalaikės priklausomybės dilemą rekurentiniuose tinkluose. Mokslininkai mano, kad LSTM tinklų rūšis turi puikias mokymosi galimybes apibūdinti įprastas laivų eismo ypatybes [14]. Dažnai moksliniuose darbuose apjungiamos skirtingos architektūros arba taikomi duomenų rekonstrukciniai metodai tikslumui gerinti, tačiau rečiau atliekamas rekurentinių tinklų struktūros palyginimas (pavyzdžiui, įvertinant LSTM celių kiekį) jūros eismo gerinimui. Buvo nuspręsta iširti LSTM pritaikymą skirtinguose tinklų architektūrose bei atlikti laivo eigos prognozavimą didžiuosiuose eismo duomenyse.

Šiame baigiamajame darbe atliekamas tyrimas, kurio metu siekiama išsiaiškinti su kuria neuroninio tinklo architektūra LSTM tinklas veikia efektyviausiai ir kokie pagrindiniai hiperparametrai tam daugiausiai įtakos turi. Tyrimui pasirinktos architektūros, kurios nagrinėjamos skirtingose literatūros šaltiniuose: standartinis LSTM, LSTM stekas, autoenkoderis ir variacinis autoenkoderis. Palyginimas atliekamas keičiant architektūras, pritaikant skirtingus LSTM celių dydžius, keičiant duomenų rinkinius ir prognozuojamus išvesties požymių detalumus. Tinkamų ir svarbiausių požymių parinkimas (kaip teigė autoriai [5], [8], [9]) gali turėti didelės įtakos prognozavimo tikslumui.

2. DUOMENŲ IR LSTM REKURENTINIŲ TINKLŲ STRUKTŪROS KŪRIMAS

Skyriuje pateikiami didžiųjų eismo duomenys ir jų apdorojimo procesai, kurie naudojami tolimesniam struktūros generavimui bei rekurentinio neuroninio tinklo mokymui. Atliekama LSTM architektūros tipų analizė. Parengiami duomenų rinkiniai ir rekurentinių tinklų kūrimo metodologija.

2.1 Didžiųjų eismo duomenų tyrimas

2.1.1 Didieji duomenys laivų sekimo informacinėje sistemoje

Didžiųjų duomenų terminas reiškia greitą duomenų gausėjimą šių dienų organizacijose ir visuomenėse. Būtent giliojo mokymo įrankiais yra išgaunama informacija prognozei iš didelių duomenų. Jei duomenys yra sudėtingi ir standartinės regresijos funkcijos negali atlikti tikslių modeliavimų, tai gilieji neuroniniai tinklai gali sukurti sudėtingesnę matematinę struktūrą, ganėtinai tiksliai atspindinčią duomenis. Laikoma, kad didžiųjų duomenų gilieji neuroniniai tinklai visada yra sudaryti daugiau nei iš vieno paslėptojo sluoksnio.

Didieji duomenys išsiskiria nuo įprastų duomenų, dėl didžiulės apimties duomenų rinkinio, kurie laikui bėgant vis auga, tai laike kintantys istoriniai duomenys. Dažnai tradiciniai duomenų valdymo įrankiai negali jų efektyviai apdoroti. Svarbi didžiuosius duomenis identifikuojanti savybė yra įvairovė (angl. *variety*), tai reiškia, kad gali būti heterogeniniai šaltiniai, kuriuose duomenys nėra struktūrizuoti, todėl norint panaudoti kaupiamą informaciją, prieš tai būtina juos apdoroti duomenų saugyklos ETL procesais.

Eismo duomenys rinkti ir išgauti iš „Shipfinder“ informacinės sistemos. „Shipfinder“ yra laivų sekimo ir jūrų žvalgybos paslaugos tiekėja. Laivų judėjimo stebėjimas atliekamas remiantis pakrantėse esančiomis AIS (automatinė identifikavimo sistema) tinklo stotimis, kurie duomenis priima palydoviniais imtuvais. Sistemoje pateikiami realiu laiku fiksuoti eismo objektai, galima stebėti jų: būsenas, judėjimo kryptį, greitį, geografines koordinates ir kt. Didžiais duomenimis, leidžiama prognozuoti laivų prekybos veiklą, srautus ar intensyvumą. Naudojantis šia paslauga, atliktas eismo stebėjimas ir duomenų gavyba.

Jungtinių tautų vykusioje konferencijoje pateikti 2018 m. didžiausi pasaulio terminalų operatoriai, pralaidumas ir pajėgumas. Remiantis publikacijos statistika (žr. 11 pav.) parinktas tiriamasis regionas, didžiausią eismo srautą Europoje turinti šalis – Olandija ir jos krantinė.

| Ranking 2018 (throughput) | Company | Headquarters | Million TEUs | Percentage share | Growth/decline (million TEUs) | Growth/decline 2017-2018 | Million TEUs | Growth/decline 2017-2018 (percentage) |
|---------------------------|-----------------------------|----------------------|--------------|------------------|-------------------------------|--------------------------|--------------|---------------------------------------|
| 1 | COSCO | China | 105.8 | 13.5 | 14.5 | 15.9 | 130.0 | 17.8 |
| 2 | Hutchison Ports | Hong Kong, China | 82.6 | 10.5 | 0.2 | 0.3 | 112.0 | 1.6 |
| 3 | PSA International | Singapore | 80.1 | 10.2 | 6.2 | 8.4 | 112.6 | 7.9 |
| 4 | APM Terminals | Netherlands | 78.6 | 10.0 | 2.3 | 3.1 | 99.7 | -2.0 |
| 5 | DP World | United Arab Emirates | 70.0 | 8.9 | 1.3 | 1.9 | 89.7 | 3.2 |
| 6 | Terminal Investment Limited | Switzerland | 47.7 | 6.1 | 3.7 | 8.4 | 62.4 | 8.7 |
| 7 | China Merchants Ports | China | 34.5 | 4.4 | 3.5 | 11.4 | 42.9 | 5.2 |
| 8 | CMA CGM | France | 25.6 | 3.3 | 0.9 | 3.5 | 38.4 | 1.6 |
| 9 | Eurogate | Germany | 13.7 | 1.7 | -0.1 | -1.1 | 22.6 | -7.0 |
| 10 | SSA Marine | United States | 12.6 | 1.6 | 1.3 | 11.4 | 20.2 | 2.5 |

11 pav. Didžiausi pasaulio terminalai [15]

2.1.2 Pramoninės laivybos metaduomenys

Krovininiai laivai yra viena populiariausių ir dažniausiai eisme aptinkamų pramoninių laivo tipų, gabenanti krovinius, prekes ir medžiagas konteneriuose iš vieno uosto į kitą. Dėl duomenų gausos specializuotasi į šio laivo tipo eigos prognozavimą. Surinkti 5 mėn. duomenys, pasiskirstę Olandijos uostų teritorijoje (šiaurės jūroje). Detalesnė surinktų duomenų specifikacija pateikta 1 lentelėje. Neapdoroti laivų įrašai laikomi CSV failų bylose, todėl turi būti struktūrizuojami ir centralizuojami duomenų saugyklose. Remiantis literatūros išvalgomis pasirinkti lentelėje pateikti laivo požymiai.

1 lentelė. Laivų eismo metaduomenys

| Regionas | Laivo tipas | Koordinačių (įrašų) skaičius | Požymių skaičius | Periodo pradžia | Periodo pabaiga |
|-------------------------|--|---|------------------|-----------------|-----------------|
| Olandija (šiaurės jūra) | Krovininiai laivai (angl. <i>cargo</i>) | 20 826 380 | 8 | 2018-09-28 | 2019-02-25 |
| Laivo požymių struktūra | | | | | |
| Nr. | Atributas | Aprašymas | | | |
| 1 | VesselId (MMSI) | Laivą identifikuojantis unikalus numeris (angl. <i>Maritime Mobile Service Identity</i>) | | | |
| 2 | Latitude | Geografinė platumos koordinatė | | | |
| 3 | Longitude | Geografinė ilgumos koordinatė | | | |
| 4 | SpeedKnt | Laivo plaukimo greitis, kuris matuojamas mazgais | | | |
| 5 | HeadingDeg | Laivo plaukimo kryptis | | | |
| 6 | DateDiff | Dviejų laiko žingsnių (nuplaukimo) skirtumas (minutėmis) | | | |
| 7 | Δ Latitude | Dviejų laiko žingsnių platumos skirtumas (plaukimo pokytis) | | | |
| 8 | Δ Longitude | Dviejų laiko žingsnių ilgumos skirtumas (plaukimo pokytis) | | | |

Koordinačių pokyčius laike, tarp dviejų taškų, galima paskaičiuoti funkcijų išvestinėmis [16]:

$$X_{\Delta t} = t_s - t_{s-1} \quad (10)$$

$$X_{\delta Lat} = \frac{X_{Lat,s} - X_{Lat,s-1}}{X_{\Delta t}} \quad (11)$$

$$X_{\delta Lon} = \frac{X_{Lon,s} - X_{Lon,s-1}}{X_{\Delta t}} \quad (12)$$

Kai kintamasis X_{Lat} yra platumos koordinatė, X_{Lon} – ilgumos koordinatė, s – laikas, skirtingose matavimo intervaluose, $X_{\Delta t}$ – laiko skirtumas. Delta koordinatžių pokyčiai (δLat ir δLon), parodo kaip greitai įvyko pasikeitimas tarp esamo ir praėjusio laiko žingsnio, tai greičio ypatybė, apjungianti skirtingus laivo požymius į vieną visumą (konjunkcija laiko ir koordinatžių požymių).

Siekiant vizualizuoti turimus duomenis, geografinėmis informacinių sistemų (ArcGIS) priemonėmis, sugeneruotas grafinis duomenų imties vaizdas (žr. 12 pav.).



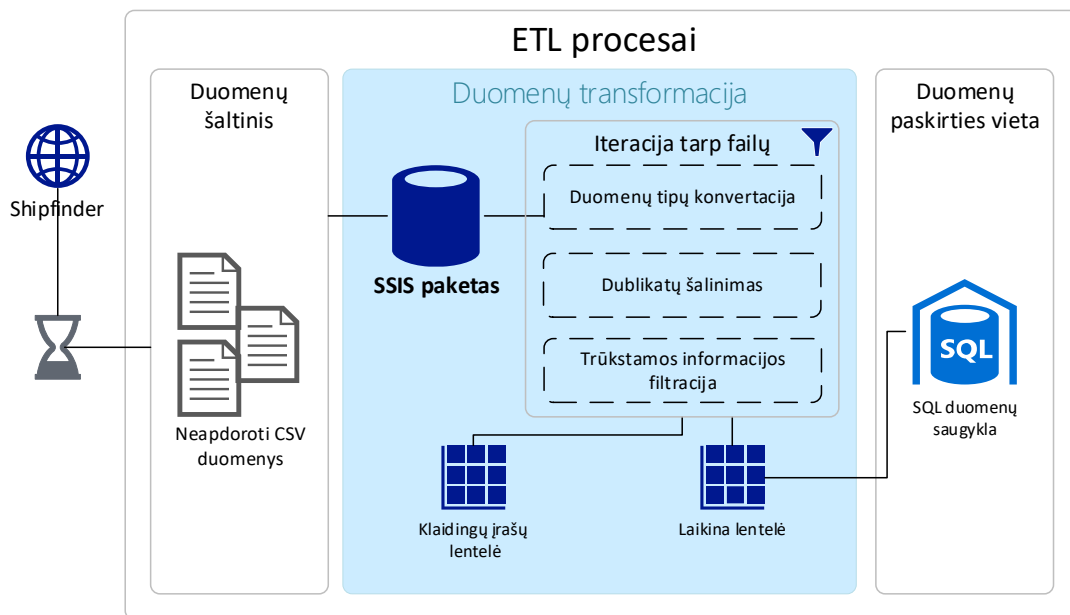
12 pav. Tiriamo regiono AIS eismo duomenų vizualizacija

Taškiniame grafike, tankiai išsidėstę duomenys žymi, kad eismo srautas vyksta ženkliai intensyviau. Pradinė duomenų imtis sudaryta iš pramoninės laivybos tipo, o regiono plotas apima visą turimų duomenų šaltinio ilgumos ir platumos plotą.

2.1.3 Duomenų šaltinio apdorojimo procesai

ETL procesai sudaryti iš trijų duomenų bazių funkcijų, sujungtos į vieną įrankį, kurio metu duomenys yra išgaunami iš duomenų šaltinių, transformuojami ir perkeltami į centrinį/pagrindinį kompiuterį (duomenų saugyklą). Laivų sekimo informacinės sistemos duomenyse gausu netvarkingų įrašų: pasikartojantys įrašų dublikatai, ne pilnos informacijos pateikimas (laivo

požymių), neapibrėžti duomenų tipai. Dėl šių priežasčių reikalinga taikyti ETL procesus (žr. 13 pav.).



13 pav. ETL procesais paremta didžiųjų duomenų apdorojimo schema

Duomenų ištraukimas (angl. *extract*) – šiame etape skirtingi duomenų šaltinių duomenys konsoliduojami.

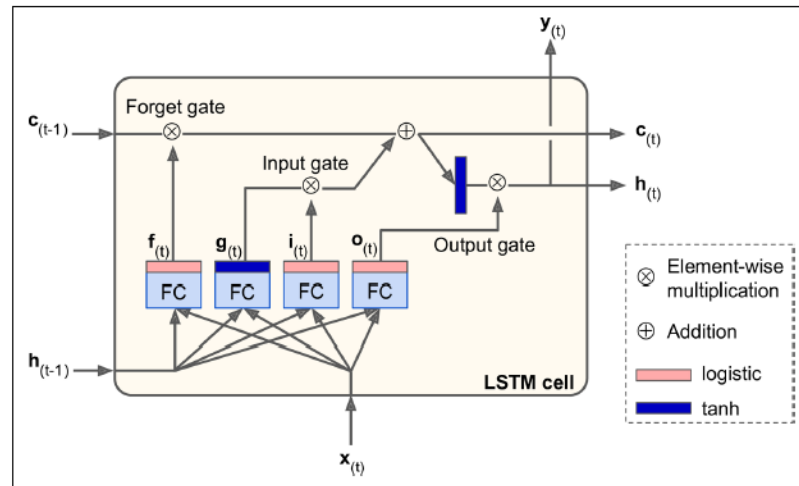
Duomenų transformacija (angl. *transform*) – konvertavimo ir valymo procesas. Šiame žingsnyje įrašų srautas, žingsnis po žingsnio, vykdomi iš anksto aprašytose SSIS paketų taisyklėse (duomenų tipų konvertavimas, dublikatų šalinimas, dėl trūkstamos informacijos duomenų filtracija).

Duomenų užkrovimas (angl. *load*) – procesas, perkeliantis duomenis į centralizuotą duomenų paskirties vietą.

2.2 LSTM tinklo architektūros

Rekurentinių tinklų charakteristikos turi bent vieną ciklinį kelią nuo sinapsinių jungčių. Visa veikla neuroniniuose tinkluose atliekama pakartotinai [17]. Tipinis RNN tinklas informaciją perduoda pasikartojančiai iš vieno bloko į kitą. Mokant RNN, informacija dažnai kartojasi, ypač ilgose sekose, todėl užtrunka tinklo svorių atnaujinimas, o senesnė informacija prarandama. Pasitaiko atveju, kai atgalinio sklaidimo metu skaičiuojamas klaidos gradientas yra labai mažas, susikaupęs jų kiekis gali tinklą paversti nestabiliu. Sprogimas įvyksta dėl eksponentinio augimo, pakartotinai padauginant gradientus per tinklo sluoksnius, kurių reikšmės yra didesnės nei 1, arba išnyksta (angl. *vanishing gradient*), jei reikšmės yra mažesnės nei 1. Dėl šių priežasčių įprastų rekurentinių tinklų tikslumas prastesnis [18].

LSTM yra patobulinta dirbtinio rekurentinio tinklo architektūra, naudojama giliojo mokymo srityje. Skirtingai nuo standartinių tiesinio sklidimo neuroninių tinklų skiriasi, nes LSTM turi grįžtamojo ryšio jungtis ir atminties valdymo struktūrą. Šios architektūros naudojimas geba spręsti nykstančio gradiento ir ilgalaikės priklausomybės problemas. Pagrindinė LSTM celės struktūra parodyta 14 paveikslėlyje. Įprastą LSTM celę sudaro: įėjimo vartai, išėjimo vartai ir užmiršimo vartai. Ląstelė įsimeina vertes per tam tikrus laiko tarpus, o trys vartai reguliuoja informacijos srautą į ląstelę ir iš jos.



14 pav. LSTM celės struktūra [19]

LSTM celės skaičiavimo algoritmas vykdomas šiomis funkcijomis:

$$\mathbf{i}_{(t)} = \sigma(\mathbf{W}_{xi}^T * \mathbf{x}_{(t)} + \mathbf{W}_{hi}^T * \mathbf{h}_{(t-1)} + \mathbf{b}_i) \quad (13)$$

$$\mathbf{f}_{(t)} = \sigma(\mathbf{W}_{xf}^T * \mathbf{x}_{(t)} + \mathbf{W}_{hf}^T * \mathbf{h}_{(t-1)} + \mathbf{b}_f) \quad (14)$$

$$\mathbf{o}_{(t)} = \sigma(\mathbf{W}_{xo}^T * \mathbf{x}_{(t)} + \mathbf{W}_{ho}^T * \mathbf{h}_{(t-1)} + \mathbf{b}_o) \quad (15)$$

$$\mathbf{g}_{(t)} = \tanh(\mathbf{W}_{xg}^T * \mathbf{x}_{(t)} + \mathbf{W}_{hg}^T * \mathbf{h}_{(t-1)} + \mathbf{b}_g) \quad (16)$$

$$\mathbf{c}_{(t)} = \mathbf{f}_{(t)} \otimes \mathbf{c}_{(t-1)} + \mathbf{i}_{(t)} \otimes \mathbf{g}_{(t)} \quad (17)$$

$$\mathbf{y}_{(t)} = \mathbf{h}_{(t)} = \mathbf{o}_{(t)} \otimes \tanh(\mathbf{c}_{(t)}) \quad (18)$$

(13) – funkcija apskaičiuojanti įvesties vartų signalą; (14) – signalo užmaršties vartų funkcija (filtracija); (15) – išvesties vartų signalo funkcija; (16) – hiperbolinė tangentine (aktyvavimo) funkcija apskaičiuojamas naujas signalas; (17) – LSTM celės būsenos signalas; (18) – galutinis LSTM celės vektorius.

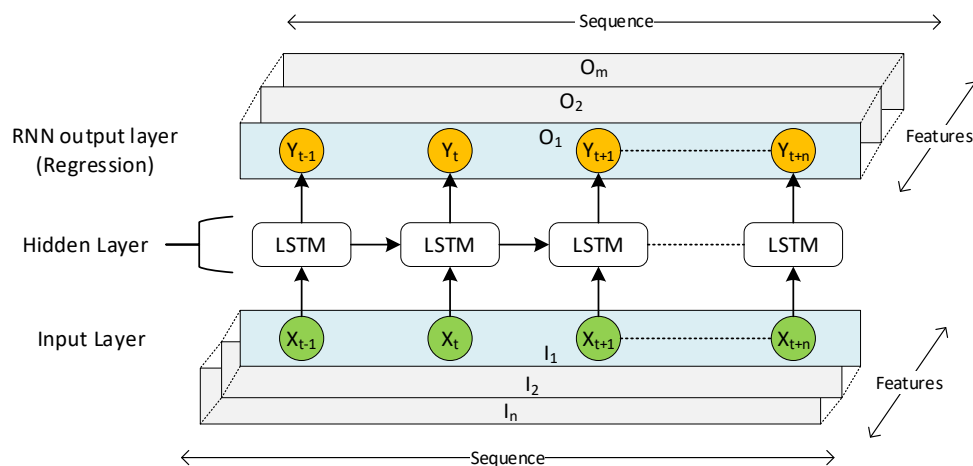
W_{xi} , W_{xf} , W_{xo} , W_{xg} yra kiekvieno iš keturių sluoksnių svorio matricos, skirtos jų prisijungimui prie įvesties vektoriaus $x(t)$.

W_{hi} , W_{hf} , W_{ho} ir W_{hg} yra kiekvieno iš keturių sluoksnių svorio matricos, susijusios su ankstesne trumpa laikina būsena $h_{(t-1)}$.

b_i , b_f , b_o ir b_g yra kiekvieno iš keturių sluoksnių papildomos poslinkio (angl. *bias*) reikšmės. Tinklo mokymo pradžioje b_f vektoriaus inicijuojamos reikšmės yra 1, o ne 0, tai neleidžia pamiršti visko treniruotės pradžioje. Visų vartų matmenys yra vienodo dydžio, kaip ir celės būsenos, kad būtų galima atlikti elementų dauginimo veiksmus.

2.2.1 Standartinis LSTM

Standartinė LSTM tinklo architektūra laikoma, tada kai tinklą sudaro vienas paslėptas LSTM celių sluoksnis [20]. LSTM yra sukonstruotas taip, kad būtų galima apdoroti sekoje esančius ir nuosekliai einančius laiko žingsnių duomenis. Standartinė konstrukcija pateikta 15 pav. Bendroje architektūroje apjungiami trys sluoksniai: įvesties sluoksnis, LSTM sluoksnis ir rekurentinis sluoksnis – sudaromas gilusis neuroninis tinklas. Paslėptąjį sluoksnį sudaro grandinėje sujungtų LSTM celių blokai. Visus juos jungia pagrindinis būsenos signalas, kuris keičiamas celės vartų pagalba. Kai įvesties vektoriai sklinda per neuroninį tinklą, dėl celėje vykstančių matematinių operacijų ir aktyvavimo funkcijų tinklo svoriai koreguojami. Projektavimo ir realizavimo dalyje tinklo sluoksniams galima priskirti skirtingas aktyvavimo funkcijas.



15 pav. Standartinė LSTM tinklo architektūra

Įvestis sukarpyta į vienodo ilgio sekas (t_n), kurios sudarytos iš laivo eigos žingsnių $X = [x_1, x_2, x_3, \dots, x_t]$, kiekviename žingsnyje turima I_n požymių. Išvesties seką sudaro $Y = [y_1, y_2, y_3, \dots, y_t]$ žingsniai ir O_m požymiai. Paskutiniame grįžtamajame sluoksnyje naudojamas papildomas RNN išvesties sluoksnis, kuriame paskaičiuojama regresijos aktyvavimo funkcija.

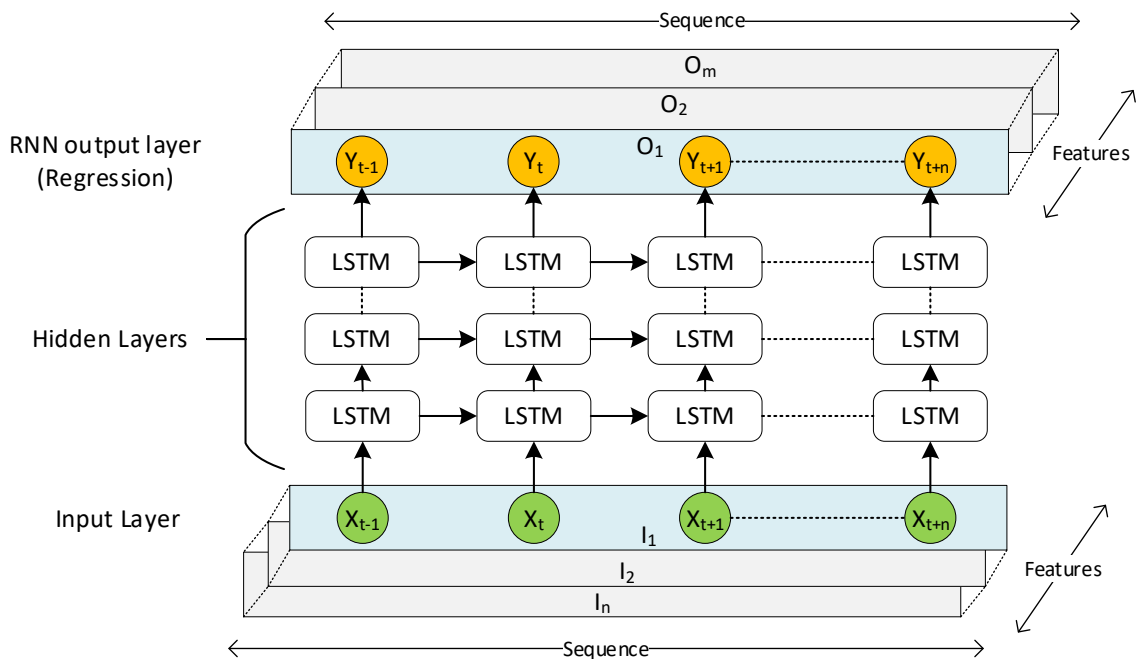
Rekurentiniai tinklai kaip ir LSTM pagal duomenų sekas yra klasifikuojami:

- ryšiu daug su daug;
- ryšiu vienas su daug;
- ryšiu daug su vienu.

Pasitaiko atveju, kai įvestyje arba išvestyje laiko žingsnių skaičius nesutampa (vienas su daug, daug su vienu), todėl norint išlaikyti vienodus ilgius yra naudojami maskavimai, kurių metu trūkstamos pozicijos užpildomos 0 reikšmėmis. Tyrime naudojamas ryšys daug su daug, nes sekos sudaromos vienodo ilgio laikantis tam tikrų taisyklių, todėl papildomas maskavimas neatliekamas. Priešingu atveju tinklas negalėtų tinkamai atlikti mokymo.

2.2.2 LSTM stekas

LSTM architektūros plėtinys yra LSTM stekas. Pagrindinis skirtumas nuo standartinės architektūros, kad konstrukcija sudaryta iš daugiau nei vieno paslėptojo sluoksnio [21] (žr. 16 pav.). Šioje architektūroje papildomai pridėti LSTM tinklo sluoksniai taip, kad būtų lygiagrečiai sujungti į steką. Kiekvieno paslėptojo LSTM sluoksnio išvestis bus sekančio paslėpto sluoksnio įvestis. Yra nustatyta, kad didinant paslėptų sluoksnių skaičiui galima išgauti geresnius rezultatus, nei naudojant tik vieną paslėptą sluoksnį [22]. Remiantis literatūroje atliktais tyrimais [23], kuriuose palygintas tinklų tikslumas keičiant sluoksnių skaičių, buvo nuspręsta LSTM steko architektūrą sukurti iš trijų paslėptų LSTM sluoksnių.



16 pav. LSTM steko architektūra

Literatūroje [24] pastebėta, kad tinklo architektūra gali būti skirstoma į grupes pagal srauto sklaidimo kryptis:

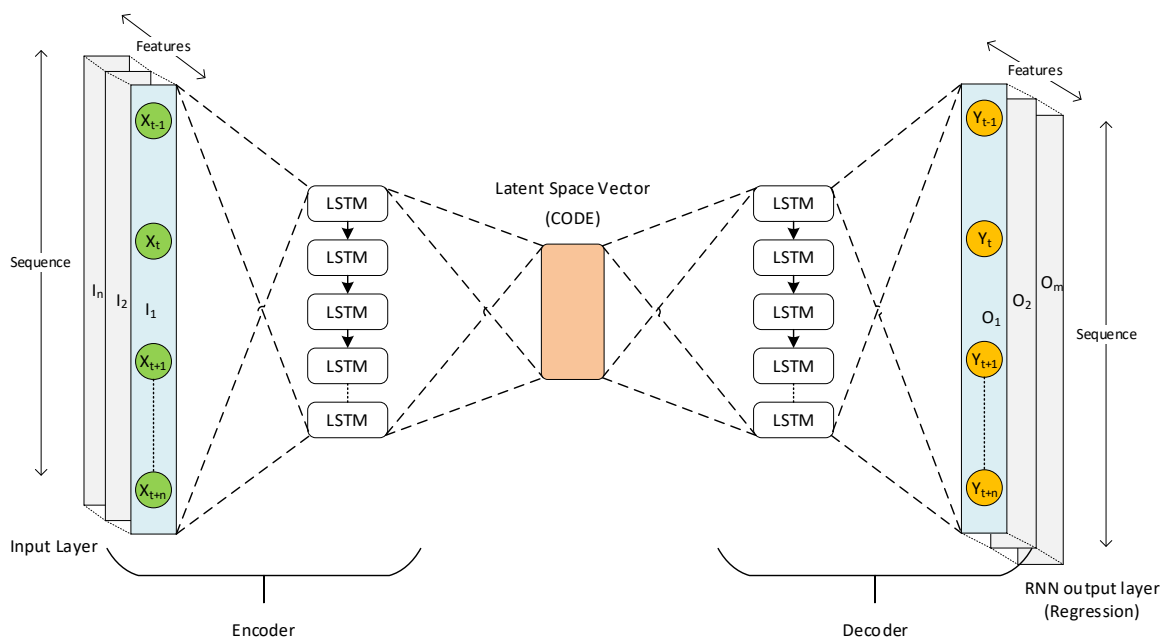
- vienkrypčius LSTM (angl. *unidirectional*);
- dvikrypčius LSTM (angl. *bidirectional*).

Vienkryptis LSTM saugo tik praeities informaciją, nes vieninteliai įėjimai, kuriuos jis mato yra iš praeities. Naudojant dvikrypčius LSTM informacija saugoma dviem būdais: viena kryptimi iš praeities duomenų į ateitį, o kita - iš ateities duomenų į praeitį. Dvikrypčiai geba saugoti informaciją abiem požiūriais, todėl yra sudėtingesni ir jų apmokymui reikalinga didžiulė skaičiavimo galia, palyginti su įprastais LSTM. Nuspręsta apsiriboti vienkrypte LSTM srauto sklaidimo kryptimis.

2.2.3 Autoenkoderis

Autoenkoderis yra specialus neuroninio tinklo tipas, kuriame įvestis yra suspaudžiama į mažesnio matmens kodą, o išvestyje suspaustas kodas rekonstruojamas. Šio tipo tinklas sudarytas iš trijų komponentų: kodavimo, latentinės erdvės vektoriaus ir dekodavimo. Vektorius yra kompaktiškas ir vadinamas latentine erdve (angl. *latent-space representation*). Tinklas priskiriamas prie neprižiūrimo ir pusiau prižiūrimo mokymo technikos, kadangi nebūtinai reikia aiškių etikečių, kad galėtų atlikti mokymą, nes iš apmokymo duomenų patys geba sugeneruoti išvestį. Kodavimas išreiškiamas funkcija $h = f(x)$, o rekonstrukcija $\hat{x} = g(h)$ [25].

Autoenkoderio struktūra pavaizduota 17 paveikslėlyje. Kodavimo dalyje įvesties duomenys yra suspaudžiami, dažniausiai mažinimas atliekamas dalijant sluoksnį per pusę, kiekviename kodavimo sluoksnyje. Duomenys dekoduojami tokia pačia tvarka kaip buvo užkoduoti t.y. dvigubinant sekos ilgį. Suspaudimo procesą pasirinkta atlikti per du sluoksnius ($(CellSize_INPUT / 2) \times 2$), tuo tarpu rekonstrukcija identiška užkodavimui ($(CellSize_CODE * 2) \times 2$). Pavyzdžiui, 100 LSTM celių sluoksnis tinkle atrodys taip: 100 – 50 – 25 – 50 – 100.

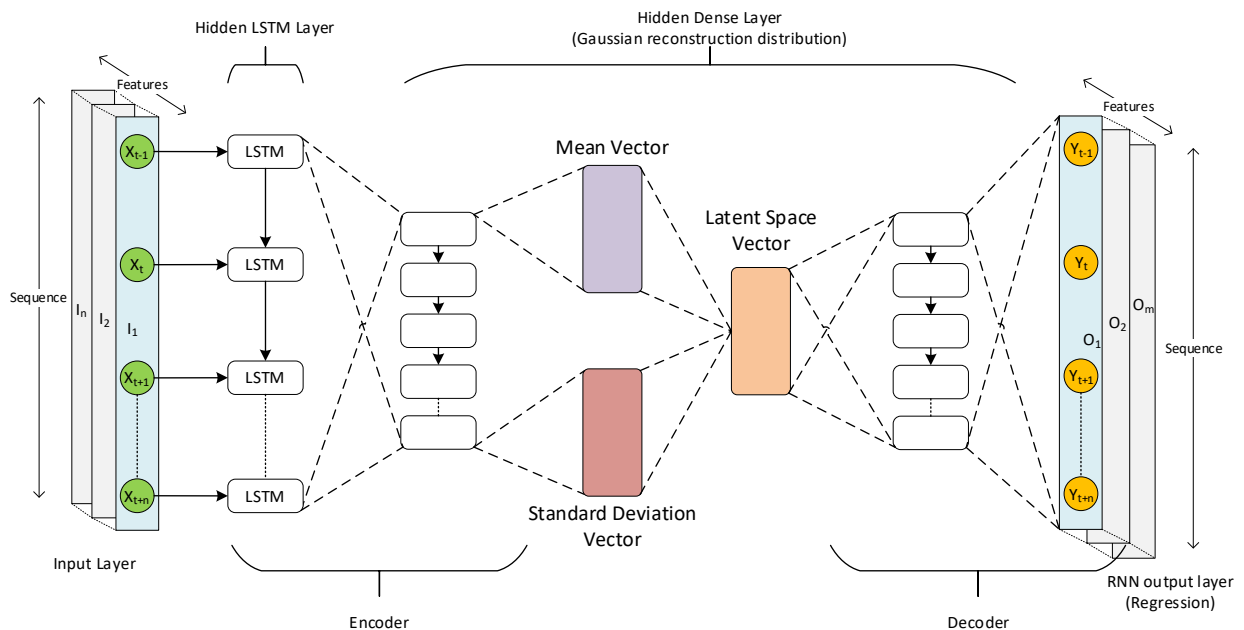


17 pav. Autoenkoderio architektūra

2.2.4 Variacinis autoenkoderis

Variacinis autoenkoderis yra taip pat neuroninis tinklas, kuris iš principo veikia panašiai kaip ir autoenkoderis, tačiau jo architektūra yra šiek tiek išplėsta (žr. 18 pav.). Kodavimo dalyje yra sugeneruojami papildomi vektoriai: vidurkio ir kovariacinė standartinio nuokrypio. Tai leidžia išplėsti latentinę erdvę atsitiktiniais dydžiais ir interpoliuoti duomenų imtį prieš dekodavimo procesą.

Architektūrą sudaryta iš: įvesties sluoksnio, LSTM sluoksnio, variacinio autoenkoderio (dense layer) ir rekurentinio išvesties sluoksnio. VAE sluoksnyje naudojamas reguliacinis Gauso pasiskirstymas (angl. *rectified Gaussian distribution*). VAE kitaip nei AE tiesiogiai neišveda suspaustų dimensijos reikšmių (iš latentinės erdvės), išvestis generuojama pagal normalųjį pasiskirstymą.



18 pav. Variacinė autoenkoderio architektūra

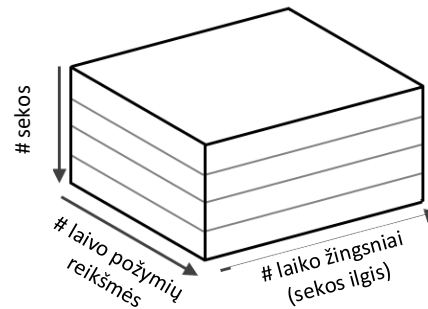
2.3 Duomenų struktūros kūrimas

Tyrimas atliekamas remiantis prižiūrimo mokymo principais, kai turima įvesties kintamuosius X ir išvesties kintamuosius Y , o taikant LSTM architektūras būtų išmokstama šių kintamųjų tarpusavio priklausomybės, kad vėliau būtų galima iš naujų duomenų rekonstruoti tolimesnę eigą. Eiga pagal laiko žingsnius gali būti skaidoma dviem būdais:

- vieno laiko žingsnio prognozė (angl. *one step*);
- kelių laiko žingsnių prognozė (angl. *multi step*).

Iškeltai problemai spręsti aktuali kelių žingsnių prognozė. Kuriama duomenų struktūra vadinama daugiamačė kelių laiko žingsnių (angl. *multivariate multi step*).

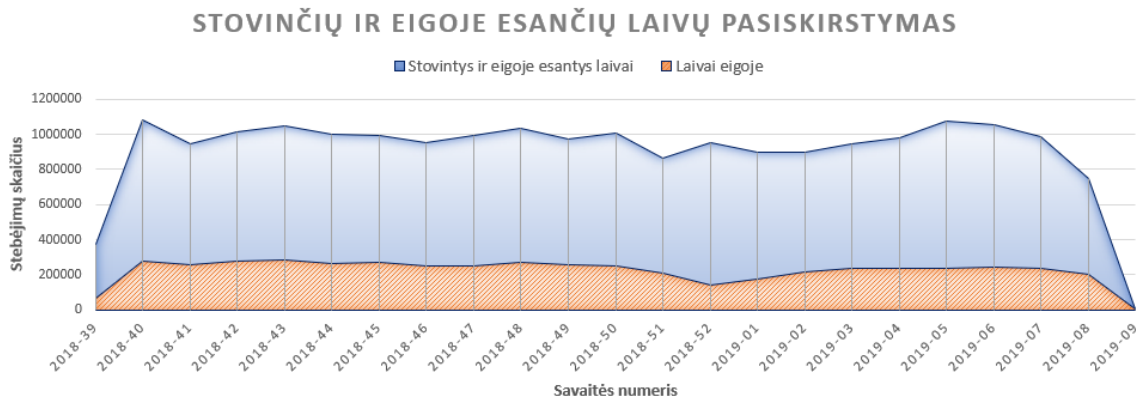
ETL procesais apdoroti duomenys turi būti išanalizuoti, struktūrizuoti bei paruošti rekurentinio tinklo naudojimui. Pagrindinis duomenų rinkinys yra trimatis (žr. 19 pav.), kurio dimensijas galima išskaidyti į: sekas, laivo požymius, laiko žingsnius.



19 pav. Trimatis laivo duomenų struktūros vaizdas

Nagrinėjant duomenis pastebėta, kad tirtame periode (žr. 20 pav.) tik apie ¼ yra laivai eigoje. Stovintys laikomi, kai greičio požymis lygus 0, todėl iš karto buvo paruošti du duomenų rinkiniai:

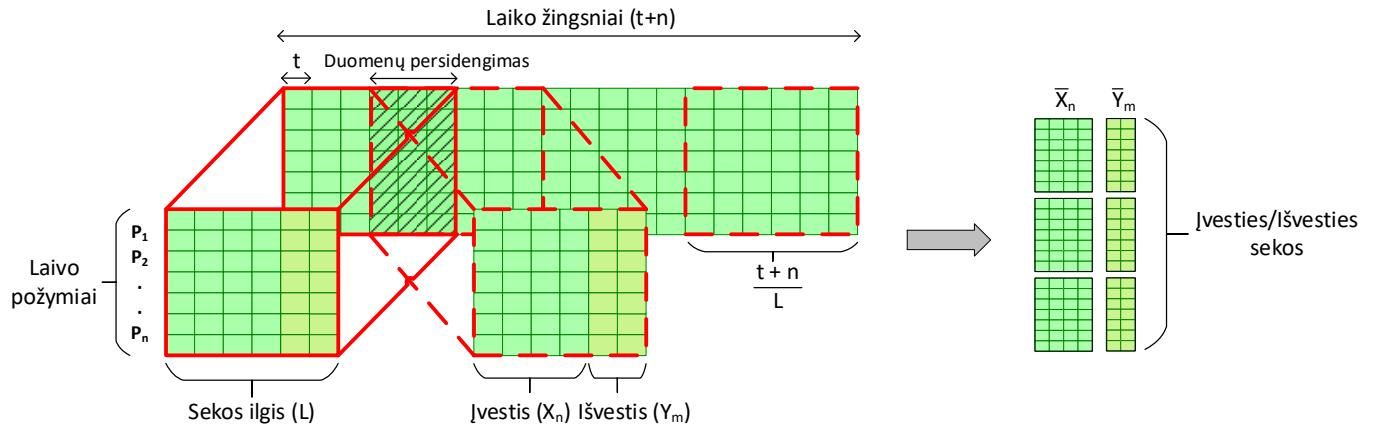
- ~ 20 mln. įrašų su stovinčiais ir eigoje esančiais laivais;
- ~ 5 mln. įrašų, kuriuose eliminuoti vietoje stovintys laivai, kadangi eigos prognozavimui tokie duomenys laikomi kaip triukšmas galintis išbalansuoti tinklo svorius.



20 pav. Stovinčių ir eigoje esančių laivo duomenų pasiskirstymo grafikas

Laivo sekų generavimas atliktas SQL lygmenyje, skaidymo procesas pateiktas toliau iliustracijoje (žr. 21 pav.). Laivų duomenys (P) yra išskaidomi į vienodo ilgio (i) sekas, kurios tarpusavyje persidengia taip vadinamu atsilikimo metodu (angl. *lag* arba *sliding window method*) bei yra surūšiuotos sinchronine eiga (pagal laiką t_n). Persidengimo metodas yra fiksuotas praleidžiamo ir įsimenamo laiko žingsnių kiekis, kai vieno stebėjimo rinkinio laiko eilutės persidengia su sekančios sekos eilutėmis, kadangi laiko eilučių prognozavimo kontekste svarbu pateikti praeities reikšmes, kad LSTM galėtų išmokti numatyti ateitį. Sekų persidengimas

atliekamas padalijant seką pusiau. Kiekviena seka išskaidoma į įvesties (\bar{X}_n) ir išvesties (\bar{Y}_m) vektorius (arba kitaip požymius ir etiketes), nes LSTM priskiriamas prie prižiūrimo mokymo. Regresijos modeliavimo metu į tinklą paduodami įvesties požymiai, o tinklas pateikia prognozuojamus išvesties požymius.

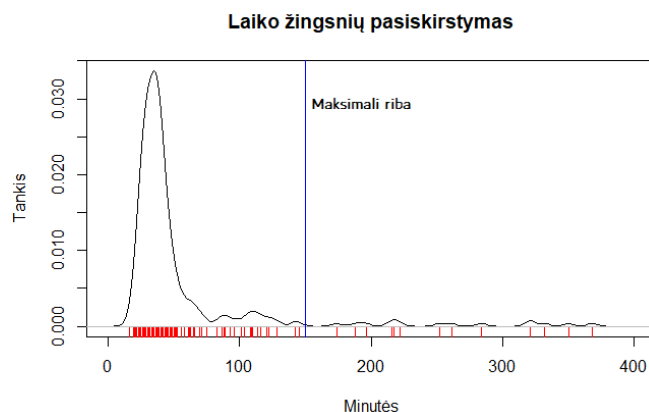


21 pav. Laivo sekų generavimo iliustracija

Sekos kūrimui taikyti specialūs kriterijai požymiams:

- išrenkami laivų įrašai, kurie priskiriami prie pramoninio tipo;
- seka negali būti sudaryta iš dviejų skirtingų MMSI;
- laiko žingsnių tolygumui ir vientisumui taikytas 150 minučių filtras. Filtras užtikrina, kad sekoje nebus didesnio laiko žingsnio tarpo nei 150 min.

Didelis laiko žingsnių tarpas aptinkamas duomenyse, nes AIS stebėjimai nebuvo atliekami nuosekliai, todėl tarpai tarp laiko žingsnių t_n ir t_{n-1} nėra vienodi. Norint išlaikyti tvirtą ir tikslią priklausomybę tarp laiko žingsnių, reikia eliminuoti dideles spragas sekoje. Filtro parametras pasirinktas remiantis apskaičiuota laiko žingsnių skirtumo dažniu (minutėmis), kuris pateiktas tankio grafike (žr. 22 pav.), nes didžiausia duomenų koncentracija pasiskirsčiusi iki 150 min. Numatoma, kad viena laivo seka vidutiniškai gali prognozuoti apie 1:30 val. tolimesnę trajektorijos eigą, kai vidutinis sekos ilgis sudarytas iš 4 val. trukmės.



22 pav. Laivų laiko žingsnių pasiskirstymas duomenyse

Laivo duomenų stebėjimų požymiai yra netolygiai svyruojantys, todėl visai struktūrai pritaikoma normalizacija [26]. Normalizavimo metu yra pakeičiami duomenų rinkinio skaitinių stulpelių reikšmės į bendrąją skalę, neiškraipant reikšmių diapazonus. Tokiu būdu pakeičiamas nepriklausomų vektorių mastelis ir visos skaitinės vertės įgauna bendrą skalę nuo 0 iki 1. Dažnu atveju ši transformacija leidžia paspartinti tinklo mokymosi procesą, nes supaprastinami skaičiavimai, kadangi sumažinama didelė įvestis ir padidinamos mažiausios reikšmės.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (19)$$

čia X yra laivo požymis, X_{max} – maksimali požymio reikšmė, X_{min} – minimali požymio reikšmė, X_{norm} – normalizuota reikšmė.

Prieš pateikiant duomenis tinklo architektūroms, sugeneruotos ir normalizuotos sekų matricos yra sumaišomos (angl. *shuffle*). Matricų maišymas užtikrina, kad kiekviena nauja seka modelyje bus nepriklausoma nuo prieš tai buvusios sekos, mažinama dispersija, todėl gradientas tampa labiau kintantis.

Duomenų imtis išskaidyta santykiu 50:25:25 (50 % mokymo duomenų, 25 % testavimo duomenų ir 25 % validacijai skirtų duomenų). Santykis pasirinktas remiantis Trevoro Hastie įžvalgomis [27].

2.4 Tinklų konfigūracija

Giliųjų neuroninių tinklų mokymosi procesas turi parametrų rinkinius, kuriuos reikia pasirinkti prieš pradėdant bet koki mokymąsi. Šie parametrai vadinami hiperparametrais. Neuroniniai tinklai turi platų hiperparametrų pasirinkimą. Kai kurie iš jų apibūdina neuroninio tinklo architektūrą, pavyzdžiui, sluoksnių skaičių ir jų dydį. Kiti apibrėžia mokymosi procesą, pavyzdžiui, mokymosi greitį ir jo sureguliuojimą. Jei tinklo konfigūracija yra netinkamai parinkta, tinklas gali mokytis lėtai arba gali visai nesimokinti.

Išskiriamos šios parametrų grupės:

- pradinių svorių nustatymas (angl. *weight initialization*);
- epochų ir iteracijų dydžiai (angl. *epochs and iterations*);
- mokymosi greičio reikšmė (angl. *learning rate*);
- aktyvinimo funkcija (angl. *activation*);
- praradimo funkcija (angl. *loss*);
- regularizacija;
- paketo dydžio sudarymas (angl. *minibatch size*);
- atnaujinimo ir optimizavimo algoritmas (angl. *updater and optimization algorithm*).

Remiantis literatūros šaltiniais [28], [29] tinklo paslėptojo sluoksnio sužadavimo funkcija pasirinkta RELU (20) (angl. *rectified linear unit*), o hiperbolinė tangento funkcija TANH (21) (angl. *hyperbolic tangent*) pasirenkama kaip rekurentinio išvesties sluoksnio aktyvavimo funkcija.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (20)$$

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \quad (21)$$

Paprastame lygmenyje aktyvavimo funkcijos padeda nuspręsti, ar reikia suaktyvinti neuroną. Tai padeda nustatyti, ar informacija, kurią gauna neuronas, yra svarbi įvesties metu. Aktyvinimo funkcija yra netiesinė transformacija, vykdoma per įvesties signalą, o transformuotas išėjimas siunčiamas į kitą neuroną.

Kadangi tiriami LSTM rekurentiniai tinklai, tai tyrimo metu išbandomos su jais labiausiai susiję parametrai. Kiekvienai naudojamai architektūrai stengtasi išlaikyti vienodas parametru kombinacijas, todėl taikyti bendrieji hiperparametru nustatymai (žr. 2 lentelėje). Detali kiekvienos architektūros struktūra ir iš jos sudaryti parametrai pateikti 1 priedas.

2 lentelė. Bendrieji LSTM tinklų mokymo hiperparametrai

| Parametro pavadinimas | Reikšmė |
|------------------------------------|--|
| Sekos įvesties ilgis | 30 |
| Sekos išvesties ilgis | 20 |
| Praradimo/klaidos funkcija | MSE |
| LSTM aktyvavimo funkcija | RELU (0.001) |
| Rekurentinė išvesties aktyvacija | TANH |
| Optimizavimo algoritmas | STOCHASTIC_GRADIENT_DESCENT |
| Gradiento atnaujinimo koeficientas | Adam (0.01) |
| Pradinių svorių vertė | RELU (normalus pasiskirstymas su dispersija) |
| Regularizacijos koeficientas | L2 (0.0001) |
| Epochų skaičius | 100 ir 500 |
| Paketo dydis | 256 |

Testuojant ir palyginant suprojektuotas tinklo konfigūracijas kaitaliojamos šios objektų grupės:

- išbandomos keturios skirtingos LSTM architektūros;
- keičiami hiperparametrai – LSTM celių kiekis;
- duomenų požymių transformacijos išvesties sluoksnyje.

2.5 Programinė ir techninė įranga

Tiriamajame darbe naudojama ši programinė įranga:

- SQL Server – duomenų laikymo ir valdymo sistema, kuri naudojama kaip saugykla. Programavimas atliekamas struktūrizuota užklausų kalba (T-SQL);
- SQL Server Integration Services (sutr. SSIS) – duomenų integravimo ir transformavimo platforma. Įrankis naudojamas pirminiam, neapdorotų duomenų surinkimui ir paruošimui į duomenų saugyklą;
- Eclipse IDE – integruota kūrimo aplinka. Neuroninių tinklų architektūros realizuotos Java programavimo kalba.

Pagrindinės bibliotekos:

- DeepLearning4J (sutr. DL4J) – atvirojo kodo, giliojo mokymo biblioteka, kuria realizuojamos aprašytos LSTM architektūros;
- N-Dimensional Arrays for Java (sutr. Nd4J) – mokslinė skaičiavimo ir tiesinės algebros biblioteka, skirta JVM. Šios bibliotekos pagalba manipuliuojami n-mačiai duomenų struktūros vektoriai;
- ArcGIS – erdvinis duomenų tyrinėjimo įrankis, kurio pagalba laivo eigos požymiai susiejami su vieta;
- JavaFX – grafinės naudotojo sąsajos biblioteka, kuria atliekami vizualiniai tinklo rezultatų išvedimai.

Giliojo mokymo metodams taikyti reikalinga didelio našumo techninė įranga, kadangi metodai iš principo yra veikiami didžiais duomenimis. DL4J biblioteka skaičiavimus leidžia atlikti tiek su CPU, tiek su GPU resursais, priklausomai nuo apsirašyto projekto nustatymų. Pagal nutylėjimą CPU naudojimas yra paprastesnis (nors ir lėtesnis), nes nereikalaujama papildomos konfigūracijos. Dėl šios priežasties ir dėl turimo pakankamai greito centrinio procesoriaus, buvo nuspręsta naudoti tik jos pajėgumus. Pateikiama detalesnė naudojamos techninės įrangos specifikacija (žr. 3 lentelėje).

3 lentelė. Techninės įrangos specifikacija

| Specifikacija | Aprašymas |
|-----------------|------------------------------------|
| CPU | Intel Core i7-6800K CPU (3.40 GHz) |
| GPU | AMD Radeon R7 360 Series 2GB |
| Kietasis diskas | SSD 500GB |
| Vidinė atmintis | 16 GB RAM (2133 MHz) |

2.6 Skyriaus apibendrinimas

ETL procesais buvo apdoroti ir centralizuoti Olandijos uoste esantys pramoninių laivų duomenys. Būtent ETL procesais atlikta pirminė AIS duomenų šaltinio filtracija. Pagrindinis didžiųjų duomenų rinkinys sudarytas iš laivo požymių: platumos, ilgumos, greičio, plaukimo krypties, laiko žingsnių skirtumo ir delta platumos, ilgumos reikšmių. Toliau šie duomenys struktūrizuoti į sekas, kurios padalijamos į vienodo dydžio įvesties ir išvesties matricas. Sekų generavimui taikyti laiko žingsnių kriterijai, kad būtų išvengta informacijos anomalijų. Sudaryti du duomenų rinkiniai: iš stovinčių ir eigoje esančių laivų (783 tūkst. sekos) ir tik kai laivai eigoje (123 tūkst. sekos).

Šiame skyriuje, suprojektuotos tirtos LSTM giliųjų neuroninių tinklų architektūros: standartinis LSTM, LSTM stekas, autoenkoderis ir variacinis autoenkoderis. Laivo eigos prognozavimui buvo pasirinkta LSTM rekurentinių tinklų rūšis, kuri išsiskiria iš įprastų rekurentinių tinklų, nes geba spręsti nykstančio gradiento ir ilgalaikės priklausomybės problemas. Pagrindinė struktūros dalis yra LSTM celė, kuri srautą valdo trimis vartais: įvesties, išvesties, užmaršties. Architektūros suprojektuotos taip, kad būtų galima palyginti standartinio sklidimo tinklus su specialiais duomenų dimensijas mažinančiais metodais. Bandoma iširti, kuri LSTM tinklų architektūra ir su koku LSTM celių kiekiu tiksliausiai atkuria kelių laiko žingsnių tolimesnę sekos eigą.

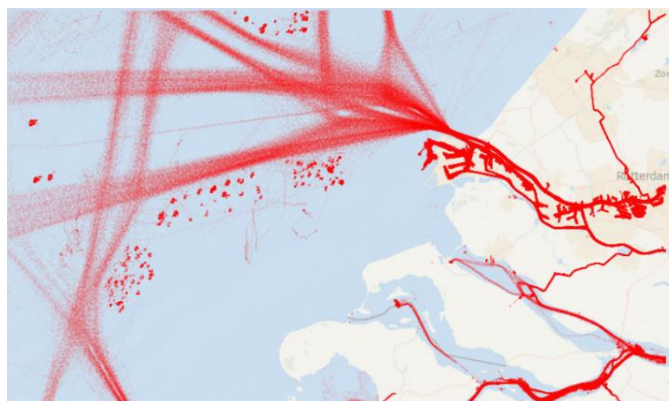
Testavimai atlikti keičiant pagrindinius tinklų objektų nustatymus: naudojant skirtingas LSTM architektūras, keičiant LSTM celių kiekį ir duomenų rinkinius (keičiant išvesties požymių informaciją). LSTM celių kiekis būna skirtingas, priklausomai nuo nagrinėjamos problemos, tačiau tirtuose pavyzdžiuose jų kiekis gali prasidėti nuo 50 celių, nuspręsta kelis kartus padidinti šį kiekį ir išbandyti skirtingas jų kombinacijas su skirtingomis LSTM architektūromis: 50, 100, 150, 200.

3. SUKURTŲ LSTM REKURENTINIŲ ARCHITEKTŪRŲ TAIKYMO EKSPERIMENTAS LAIVO EIGOS PROGNOZAVIMUI

Šiame skyriuje atliekamas sukurtų LSTM giliųjų neuroninių tinklų realizavimas (DL4J), apmokant juos eksperimentiniais duomenų rinkiniais bei atliekamas efektyvumo įvertinimas ir rezultatų palyginimas, siekiant prognozuoti laivo eigos kryptį. Tinklų tikslumas apskaičiuojamas vidutine kvadratine paklaida (MSE), kuri įvertina prognozuojamas ir faktines reikšmes. Palyginama architektūrų mokymosi proceso eiga ir trukmė. Tarpiniai rezultatai pateikiami ArcGIS ir JavaFX vizualizacijomis.

3.1 Eksperimentinės dalies aprašas

Siekiant paspartinti skirtingų architektūrų eksperimentinę dalį ir įvertinti rezultatų tikslumą buvo sumažintas tiriamo regiono plotas (koordinacių pozicijos: riba į vakarus = 3,1000; riba į pietus = 51,6350; riba į rytus = 4,5000; riba į šiaurę = 52,1500). Pasirinktas intensyviausius srautus turintis plotas (žr. 23 pav.). Vizualizuoti duomenys išskaidomi į mokymo, validavimo ir testavimo imtis pagal Trevoro siūlomą santykį. Apmokymo imtis naudojama modelių kūrimui. Validacijos imtis naudojama stebėti ir patikrinti modelių kūrimo procesą. Testavimo imtis naudojama galutinio modelio tikslumo įvertinimui. Detalesnis duomenų imties išskaidymas pateiktas 2 priedas. Šioje vizualizacijoje pateikti stovintys ir eigoje esantys laivai. Duomenys struktūrizuoti pagal antrame skyriuje aprašytą metodologiją.



23 pav. Duomenų rinkinio geografinė vizualizacija

Suprojektuotos ir realizuotos architektūros tiriamos eksperimentiniu būdu, tokia eiga (žr. 24 pav.):

1. Atliekamas visų keturių architektūrų apmokymas duomenų rinkiniu (A) su visais laivo požymiais įvestyje ir išvestyje.

2. Įvertinamas sukurtų modelių tikslumas (MSE) su visais testavimo duomenų imties požymiais. Atsitiktinai išsirinkus seką, atliekamas laivo eigos prognozavimas su tiksliausiomis LSTM celių kombinacijomis, pateikiamos iliustracijos.

3. Įvertinamas sukurtų modelių tikslumas (MSE) tik su koordinacių išvesties požymiais. Tyrimo tikslas prognozuoti laivo eigą, todėl nėra būtina į vertinimą įtraukti su geografine pozicija nesusijusių duomenų.

4. Atliekamas standartinės LSTM architektūros apmokymas duomenų rinkiniu, kai išvestyje paliekami greičio ir delta platumos bei delta ilgumos pokyčių kitimai.

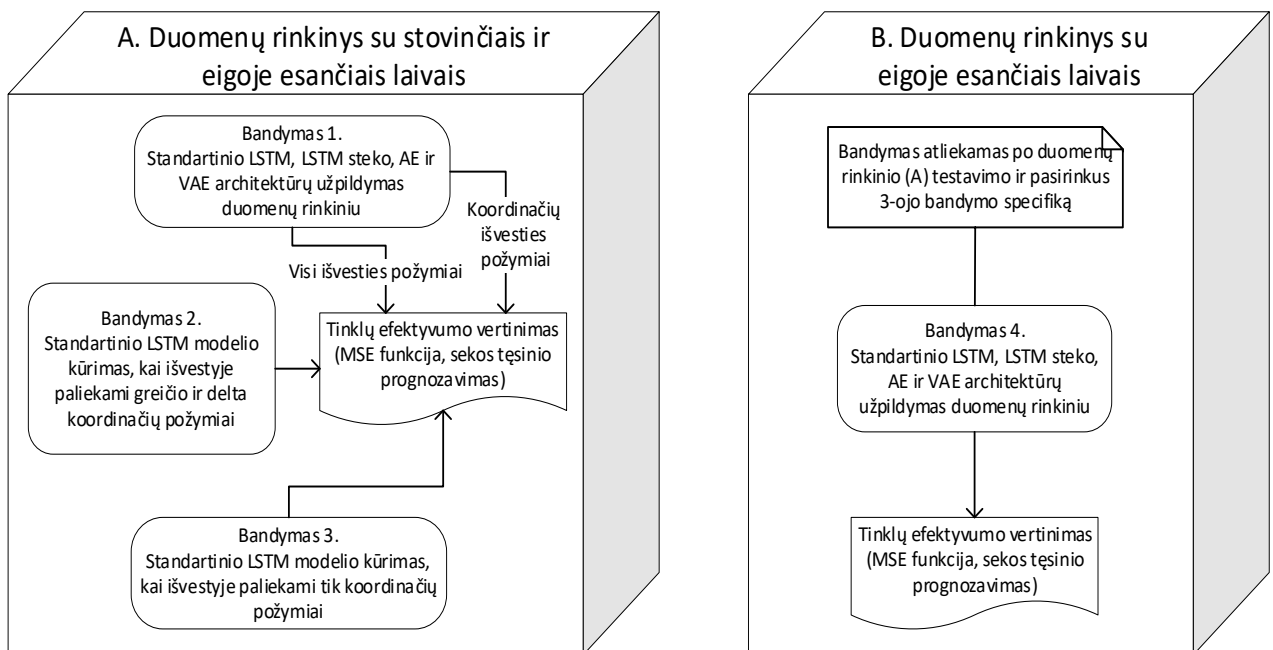
5. Atliekamas standartinės LSTM architektūros apmokymas duomenų rinkiniu, kai išvestyje paliekami tik koordinacių požymiai, kiti požymiai (greitis, kryptis ir kt.) pašalinami priskiriant 0 reikšmes.

6. Pateikiami vizualiniai 4-ame ir 5-ame žingsnyje sukurtų modelių palyginimas prognozuojant tą pačią atsitiktinę laivo eigą.

7. LSTM architektūros užpildomos nauju duomenų rinkiniu (B) tik su eigoje esančiais laivais. Po gautų rezultatų iš duomenų rinkinio (A) nuspręsta pasirinkti 3-ojo bandymo specifiką: išvestyje paliekama tik laivo geografinė padėtis (realios koordinatės).

8. Įvertinamas tinklų tikslumas (MSE) bei pateikiama sekos prognozės vizualizacija su tiksliausiomis LSTM celių kombinacijomis.

9. Ištyrus gautus rezultatus, pasirenkama tiksliausiai laivo eigos prognozę atliekanti architektūra su tinkamiausiu LSTM celių skaičiumi sluoksnyje. Paskaičiuojamas tiriamų architektūrų absoliutus procentinis nuokrypis.



24 pav. Eksperimento bandymų schema

Žemiau pateikiamas duomenų imties (sekos) fragmentas (žr. 25 pav.): platumas, ilgumas, greitis, plaukimo kryptis, laiko žingsnių skirtumas ir delta platumos, delta ilgumos reikšmės. Pavyzdyje pateikiami ne normalizuoti laivo stebėjimo duomenys, kurie išskaidyti į įvesties [30 X 7] ir išvesties [20 X 7] matricas. Visais bandymų atvejais naudojami visi įvestyje esantys požymiai. Išvesties sekos transformuojamos priklausomai nuo reikiamo apmokymo detalumo, nenaudojant požymius pakeičiant į nulines reikšmes, išlaikant tą pačią duomenų struktūrą. Kaip ir buvo minėta anksčiau duomenys yra surūšiuoti pagal laiką. Sekos laiko žingsnių skirtumo suma parodo per kiek laiko (minutėmis) laivo eigos atkarpa buvo įveikta.

| | Latitude | Longitude | SpeedKnt | HeadingDeg | DateDiff | dLat/dt | dLong/dt |
|----------------------------|----------|-----------|----------|------------|----------|--------------------|--------------------|
| Sekos įvestis [30 X 7] | ... | ... | ... | ... | ... | ... | ... |
| | 51.8929 | 4.4360 | 0.10 | 41.00 | 37 | 0.000002702702702 | 0.000008108108108 |
| | 51.8930 | 4.4363 | 1.00 | 77.10 | 2 | 0.000150000000000 | 0.000700000000000 |
| | 51.8933 | 4.4377 | 2.10 | 63.40 | 2 | 0.000300000000000 | 0.001050000000000 |
| | 51.8939 | 4.4398 | 2.60 | 68.70 | 2 | -0.000100000000000 | 0.000700000000000 |
| | 51.8937 | 4.4412 | 2.40 | 210.70 | 2 | 0.001100000000000 | -0.001800000000000 |
| | 51.8959 | 4.4376 | 9.90 | 341.80 | 3 | 0.001700000000000 | -0.001833333333333 |
| | 51.9010 | 4.4321 | 11.00 | 303.20 | 2 | 0.000650000000000 | -0.005250000000000 |
| | 51.9023 | 4.4216 | 12.50 | 262.20 | 2 | -0.001400000000000 | -0.005850000000000 |
| | 51.8995 | 4.4099 | 13.40 | 250.60 | 2 | -0.001050000000000 | -0.005850000000000 |
| Sekos išvestis [20 X 7] | 51.8974 | 4.3982 | 13.50 | 257.30 | 2 | -0.000300000000000 | -0.005950000000000 |
| | 51.8968 | 4.3863 | 13.30 | 277.40 | 2 | 0.000900000000000 | -0.005900000000000 |
| | 51.8986 | 4.3745 | 13.80 | 283.90 | 2 | 0.000500000000000 | -0.006000000000000 |
| | 51.8996 | 4.3625 | 13.30 | 267.90 | 2 | -0.000600000000000 | -0.006050000000000 |
| | 51.8984 | 4.3504 | 13.70 | 255.80 | 2 | -0.001250000000000 | -0.005800000000000 |
| | 51.8959 | 4.3388 | 13.80 | 251.30 | 2 | -0.000600000000000 | -0.006100000000000 |
| | ... | ... | ... | ... | ... | ... | ... |

25 pav. Laivo duomenų sekos fragmentas

SQL lygmenyje sudarytos duomenų sekos perkeliama į sukurtą Java projektą, kuriame Nd4j bibliotekos pagalba duomenys normalizuojami imtinai nuo 0 iki 1 su matematine funkcija (19). Normalizavimas atliekamas kiekvienam požymio stulpeliui, jų statistiniai įverčiai pateikiami 4 lentelėje. Duomenų normalizacija pagreitina tinklo mokymosi procesą, tačiau svarbu, kad tos pačios minimumo ir maksimumo reikšmės būtų naudojamos apmokymo ir testavimo metu, kai normalizuotos reikšmės atkuriamos pagal gautą statistiką. Įverčiai sudaryti atsižvelgiant į įvesties ir išvesties požymių reikšmes.

4 lentelė. Normalizacijos statistiniai įverčiai

| Laivo požymis | Minimali reikšmė | Maksimali reikšmė | Skalės riba |
|-----------------------------------|------------------|-------------------|-------------|
| platumos koordinatės | 51,6350 | 52,1500 | 0,5150 |
| ilgumos koordinatės | 3,10 | 4,50 | 1,40 |
| plaukimo greitis | 0,00 | 1023,00 | 1023,00 |
| plaukimo kryptis | 0 | 360,00 | 360,00 |
| laiko žingsnių (eigoje) skirtumas | 0 | 150,00 | 150,00 |
| platumos delta pokytis | -0,3333 | 0,3335 | 0,6668 |
| ilgumos delta pokytis | -0,5637 | 0,6311 | 1,1948 |

3.2 Tinklų mokymosi proceso eiga

Siekiant išlaikyti vienodas sąlygas visoms architektūroms, mokymo proceso metu naudoti to paties regiono duomenys ir bendri hiperparametrai. Tinklų mokymosi eigoje naudojami apmokymui ir validacijai skirtos duomenų imtys. Validacijos duomenų imtimi atliekamas nešališkas modelio vertinimas proceso metu, kadangi šie duomenys įprastai nėra įtakoję tinklo svorių. Tačiau šia verifikacija galima stebėti hiperparametrų tinkamumą bei matyti kaip tiksliai modelis nustato ryšius tarp žinomų tikslinės grupės rezultatų ir nežinomų validacijos duomenų imties savybių.

Įvertinimas atliekamas periodiškai po kiekvienos epochos. Praradimo funkcijos (22) verčių sumos vidurkis paskaičiuojamas tarp visų rinkinio sekų. Šie skaičiavimai atlikti duomenų rinkinyje be stovinčių laivų. Detalesnė mokymosi ir validacijos proceso informacija pateikta 3 priedas. Didžiausi svyravimai pastebimi per pirmąsias 500 epochas. Kadangi didesnio atotrūkio tarp apmokymo ir validacijos grafikų nėra, daroma prielaida, kad pirminiai tinklo parametrai parinkti tinkamai. Atotrūkis mažas dėl mažos dispersijos. Tinklai mokosi sėkmingai, nes kreivės krenta ir šiek tiek stabilizuojasi, bet verta paminėti, kad validacijos praradimo vertė mažesnė nei apmokymo.

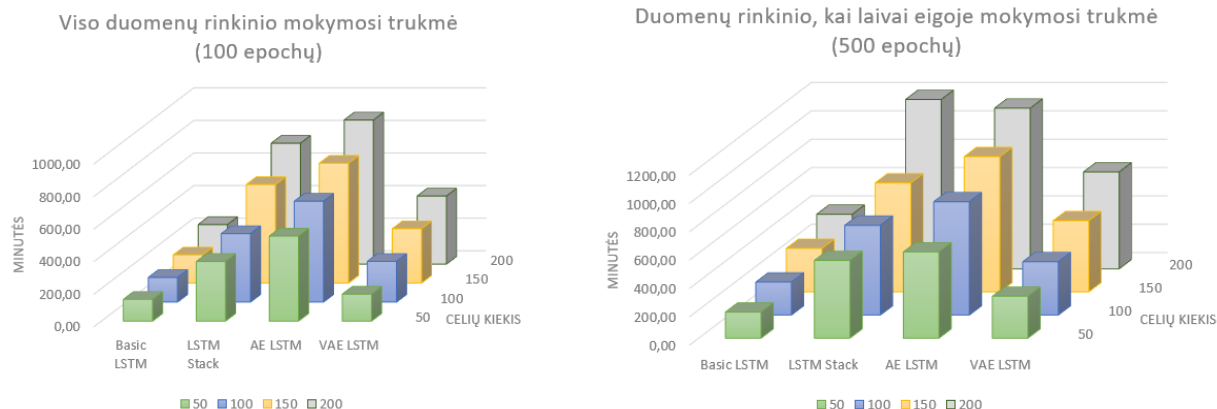
$$X_{score} = \frac{\sum_{i=1}^n (MSE_{loss})}{n} \quad (22)$$

Kai MSE_{loss} yra regresinės praradimo funkcijos suma, kuri apskaičiuota vidutine kvadratu paklaida (6), n – sekų skaičius duomenų rinkinyje, X_{score} – tinklo įvertinimo rezultatas.

Taip pat, architektūrų mokymosi eigoje buvo pasirinkta laivo seka, kuri stebėta praėjus tam tikram epochų skaičiui. Šia informacija norėta patikrinti kaip kinta sekos prognozė, kai keletą kartų praeinama pro visą duomenų rinkinį. Kiekvienos architektūros tiksliausių sekos pokyčius su atitinkamu LSTM celių kiekiu sluoksnyje galima rasti 4 priedas. Mėlyna spalva pažymėta įvesties eiga, raudona spalva pažymėta tikroji išvestis, o punktyrinės linijos žymi spėjimą prognozę. Pastebima, kad didėjant epochų skaičiui prognozės srautas artėja prie tikrosios pozicijos, tačiau tikslumas priklauso nuo naudojamos architektūros.

Atlikus architektūrų apmokymą buvo įvertinta šio proceso trukmė atskirai kiekviename duomenų rinkinyje. Pateiktame 26 paveikslėlyje įvertinta trukmė, pasirinktas minučių tikslumo matas. Architektūrose koreguoti LSTM celių dydžiai. Dėl ypač didelio duomenų kiekio pirmasis duomenų rinkinys (su stovinčiais ir eigoje esančiais laivais) įvertintas po 100 epochų, antrasis (be stovinčių laivų) po 500 epochų. Iš grafikų matoma, kad ilgiausiai trunka apmokyti autoenkoderio ir LSTM steko architektūras. Trukmė didėja eksponentiškai didinant LSTM celių skaičiui, kadangi

sluoksnis plečiamas, tai didėja ir parametų dydis (svoriai, *bias* poslinkiai), kuriuos skaičiuoja tinklas mokymosi proceso eigoje.



26 pav. Architektūrų mokymosi trukmės palyginamoji diagrama

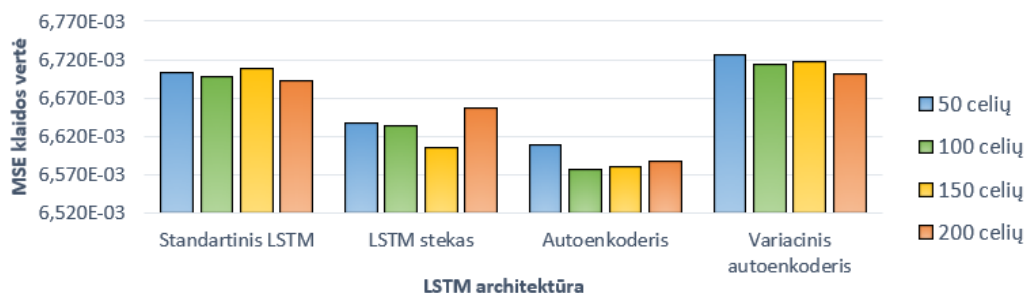
3.3 Gautų tyrimo rezultatų palyginimas

Duomenų rinkiniu su stovinčiais ir eigoje esančiais laivais (A) apmokytų architektūrų efektyvumas palyginamas dviem būdais: su visais laivo požymiais ir tik su koordinacių požymiais. Skaičiavimai atlikti pritaikius vidutinę kvadratinę paklaidą (MSE), su skirtingais LSTM celių dydžiais sluoksniuose (žr. 5 lentelėje). Kadangi architektūros buvo apmokytos su visais laivo požymiais, tai reiškia, kad modelis prognozavimo metu bando nuspėti ne tik laivo plaukimo eigą (koordinates), bet ir likusius požymius (greitį, kryptį ir kt.), todėl skaičiuojant paklaidą jų reikšmės taip pat įvertinamos. Dėl šios priežasties, tinklo vertinimo metu, išvesties matricos požymiai: greitis, plaukimo kryptis, laiko žingsnių kryptis, delta platumo ir delta ilgumos požymiai sulyginami su prognozuotomis reikšmėmis, kad įvertinta būtų tik ilguma ir platumas. Abiem atvejais tiksliausiai prognozuoja AE. Skirtumas tas, kad pasikeičia skalės vertinimo proporcijos.

5 lentelė. Prognozavimo tikslumo vertinimas su stovinčiais laivais

| Taikyta architektūra | LSTM celių skaičius sluoksnyje | | | | Mažiausia MSE klaidos vertė |
|--------------------------|---|--|--|--|--|
| | 50 | 100 | 150 | 200 | |
| Visi duomenų požymiai | | | | | |
| Standartinis LSTM | $6,703 \times 10^{-3}$ | $6,698 \times 10^{-3}$ | $6,709 \times 10^{-3}$ | $6,692 \times 10^{-3}$ | $6,692 \times 10^{-3}$ |
| LSTM stekas | $6,637 \times 10^{-3}$ | $6,633 \times 10^{-3}$ | $6,605 \times 10^{-3}$ | $6,657 \times 10^{-3}$ | $6,605 \times 10^{-3}$ |
| Autoenkoderis | $6,609 \times 10^{-3}$ | $6,577 \times 10^{-3}$ | $6,581 \times 10^{-3}$ | $6,588 \times 10^{-3}$ | $6,577 \times 10^{-3}$ |
| Variacinis autoenkoderis | $6,727 \times 10^{-3}$ | $6,714 \times 10^{-3}$ | $6,718 \times 10^{-3}$ | $6,701 \times 10^{-3}$ | $6,701 \times 10^{-3}$ |
| Koordinacių požymiai | | | | | |
| Standartinis LSTM | $1,24 \times 10^{-4}$ | $1,26 \times 10^{-4}$ | $1,27 \times 10^{-4}$ | $1,22 \times 10^{-4}$ | $1,22 \times 10^{-4}$ |
| LSTM stekas | $9,60 \times 10^{-5}$ | $9,90 \times 10^{-5}$ | $9,38 \times 10^{-5}$ | $9,37 \times 10^{-5}$ | $9,37 \times 10^{-5}$ |
| Autoenkoderis | $8,94 \times 10^{-5}$ | $9,20 \times 10^{-5}$ | $8,96 \times 10^{-5}$ | $9,04 \times 10^{-5}$ | $8,94 \times 10^{-5}$ |
| Variacinis autoenkoderis | $1,05 \times 10^{-4}$ | $1,05 \times 10^{-4}$ | $1,06 \times 10^{-4}$ | $1,04 \times 10^{-4}$ | $1,04 \times 10^{-4}$ |

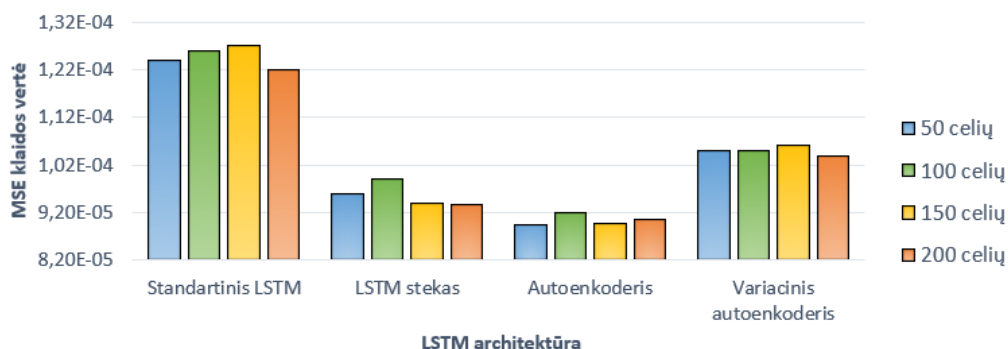
Susisteminti duomenys parodo kokia architektūra tiksliau atlieka laivo eigos prognozavimą. Regresijoje prognozė laikoma tiksli tada, kai klaidos vertė yra arčiausiai nulio. Įvertinus tinklus su visais požymiais gauta vizualizacija (27 paveikslas).



27 pav. LSTM architektūrų tikslumas įvertinant visus požymius

Iš pateikto grafiko matyti, jog pagal tikslumą architektūros rikiuojasi tokia tvarka: AE, LSTM stekas, standartinis LSTM ir VAE. Tiksliausia prognozė atlikta su AE, naudojant ir mažinant 100 celių sluoksnį. Sekančiuose sluoksniuose šis dydis suspaudžiamas į 50 ir 25 celes. Latentinės erdvės sluoksnis sudarytas iš 25 celių dekodavimo dalyje yra rekonstruojamas į pradinę struktūrą.

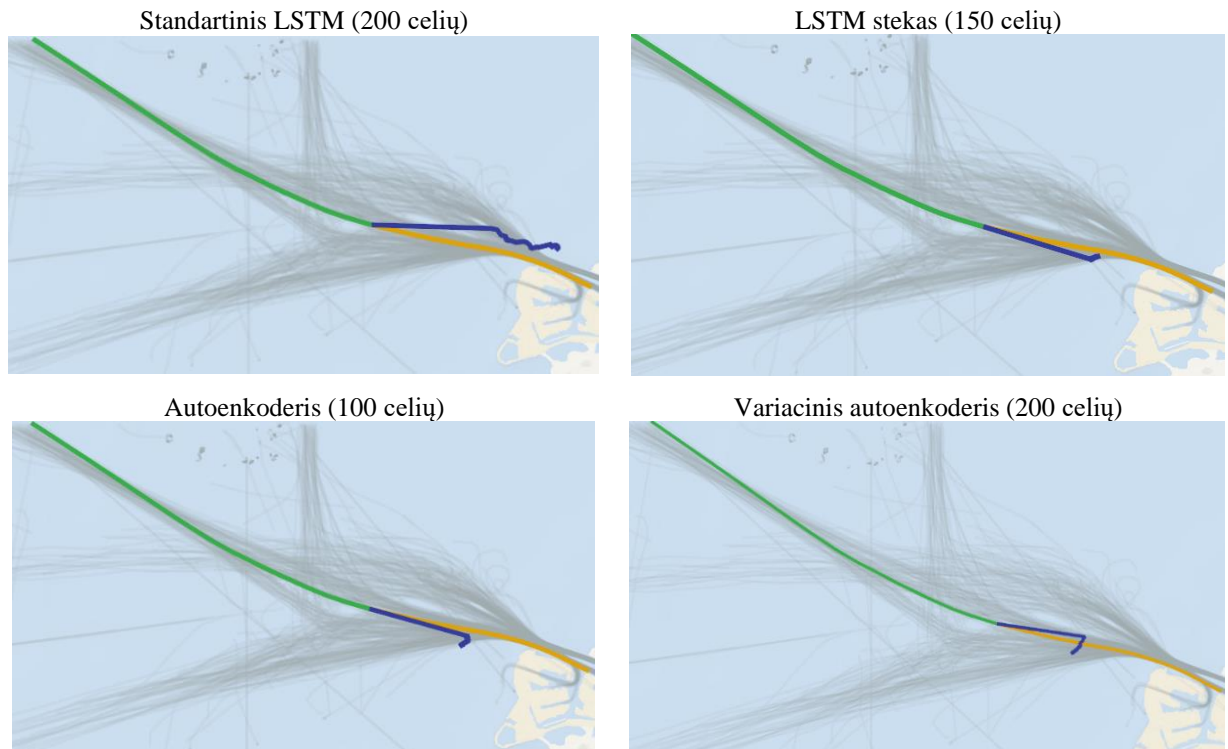
Įvertinus tinklus tik su koordinacių požymiais gauta vizualizacija (28 paveikslas).



28 pav. LSTM architektūrų tikslumas įvertinant koordinacių požymius

Palyginus su pirmuoju grafiku, pastebima, kad standartinio LSTM efektyvumas suprastėja, o VAE pagerėja. Pagal tikslumą architektūros rikiuojasi tokia tvarka: AE, LSTM stekas, VAE ir standartinis LSTM. Taip pat, neženkliai pasikeičia tiksliausios architektūros naudojamas celių skaičius (50 celių) sluoksnyje.

Atlikus atsitiktinės sekos prognozę gauti rezultatai (žr. 29 pav.). Rezultatai vizualizuoti su geriausiai LSTM celės dydžiais. Žalia linija žymi įvesties eigą (30 laiko žingsnių), geltona spalva pažymėta originali laivo išvesties eiga (20 laiko žingsnių), mėlyna spalva vizualizuoja apmokyto tinklo laivo eigos prognozavimą (20 laiko žingsnių). Vis dėlto, duomenų rinkinyje, kurioje stebimi stovintys ir eigoje esantys laivai nėra tikslūs. Eigos koordinatės išsidėsto vienas šalia kito, sudarydamos nelogiškus mazgus.

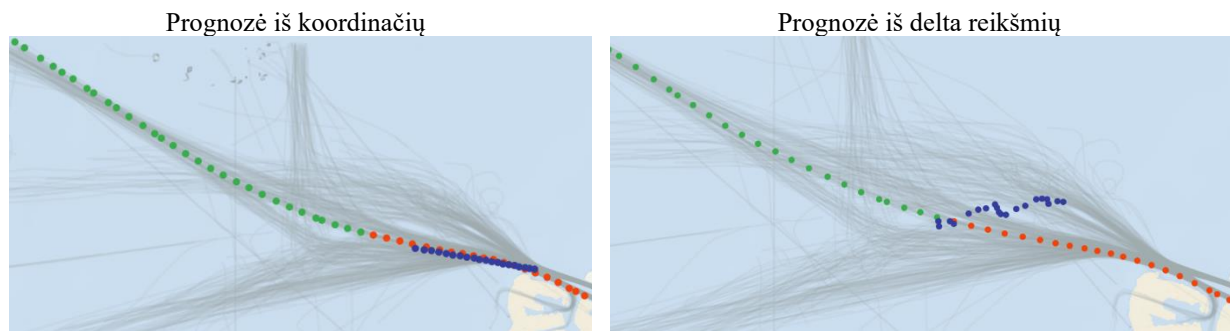


29 pav. LSTM architektūrų sekos vizualizacija iš stovinčių ir eigoje esančių laivų

Buvo atlikti papildomi bandymai su standartiniu LSTM architektūros tipu. Šio tinklo architektūra apmokyta pakeitus išvesties požymius: išvestyje palikti tik geografinių koordinatčių požymiai (prognozė kairėje) ir palikus greitį bei delta koordinates (prognozė dešinėje) (žr. 30 pav.). Kai tinklui nereikia prognozuoti kitų požymio duomenų išvestyje (pavyzdžiui, greičio), tai koordinatčių tikslumas šiek tiek pagerėja duomenų rinkinyje su stovinčiais laivais, palyginus su 29 paveikslėlyje gautais rezultatais. Laivo eigos prognozė atlikta dviem būdais: pirmu atveju prognozuojamos realios laivo koordinatės, antruoju atveju prognozuojamos delta koordinatčių reikšmės su greičio požymiais. Delta reikšmės (gautos iš 11, 12 išvestinių funkcijų) kartu su greičio požymiu gražinamos į pirminę būseną t.y. iš apskaičiuotų reikšmių atkuriamos koordinatės. Kadangi delta reikšmės sudarytos iš greičio (X_{sp}) ir laiko žingsnių skirtumo, tai žinant delta koordinatę ($X_{\delta Lat}$, $X_{\delta Long}$) ir paskutinio įvesties laiko žingsnio ($X_{Lat,s}$, $X_{Long,s}$) koordinatę galima išskaičiuoti sekančios eigos poziciją (platumą (23 formulė), ilgumą (24 formulė)). Nors eigoje nepastebima nelogiškų krypties judėjimų, taikant šiuos skaičiavimus, tačiau rezultatai vis tiek nėra tikslūs.

$$X_{Lat} = (X_{\delta Lat} * X_{sp}) + X_{Lat,s} \quad (23)$$

$$X_{Long} = (X_{\delta Long} * X_{sp}) + X_{Long,s} \quad (24)$$



30 pav. Standartinės LSTM architektūros eigos prognozė be papildomų požymių

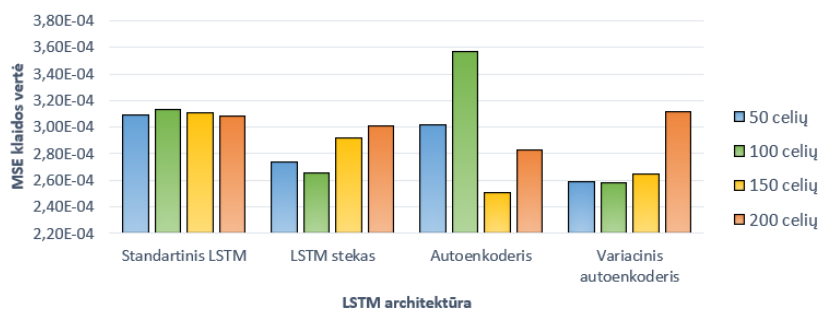
Kitas eksperimentas atliktas su nauju duomenų rinkiniu (B), kuriame nėra stovinčių laivų. Visos architektūros su skirtingais hiperparametrais (LSTM celėmis) apmokytos iš naujo. Sukurtų tinklų modeliai įvertinti testavimo duomenų imtyje, paskaičiavus vidutinę paklaidą (MSE), gauta informacija pateikta 6 lentelėje.

6 lentelė. Prognozavimo tikslumo vertinimas be stovinčių laivų

| Taikyta architektūra | LSTM celių skaičius sluoksnyje | | | | Mažiausia MSE klaidos vertė |
|--------------------------|--------------------------------|---|---|---|---|
| | 50 | 100 | 150 | 200 | |
| Koordinacių požymiai | | | | | |
| Standartinis LSTM | $3,09 \times 10^{-4}$ | $3,13 \times 10^{-4}$ | $3,11 \times 10^{-4}$ | $3,08 \times 10^{-4}$ | $3,08 \times 10^{-4}$ |
| LSTM stekas | $2,74 \times 10^{-4}$ | $2,66 \times 10^{-4}$ | $2,92 \times 10^{-4}$ | $3,01 \times 10^{-4}$ | $2,66 \times 10^{-4}$ |
| Autoenkoderis | $3,02 \times 10^{-4}$ | $3,57 \times 10^{-4}$ | $2,51 \times 10^{-4}$ | $2,83 \times 10^{-4}$ | <u>$2,51 \times 10^{-4}$</u> |
| Variacinis autoenkoderis | $2,59 \times 10^{-4}$ | $2,58 \times 10^{-4}$ | $2,65 \times 10^{-4}$ | $3,12 \times 10^{-4}$ | $2,58 \times 10^{-4}$ |

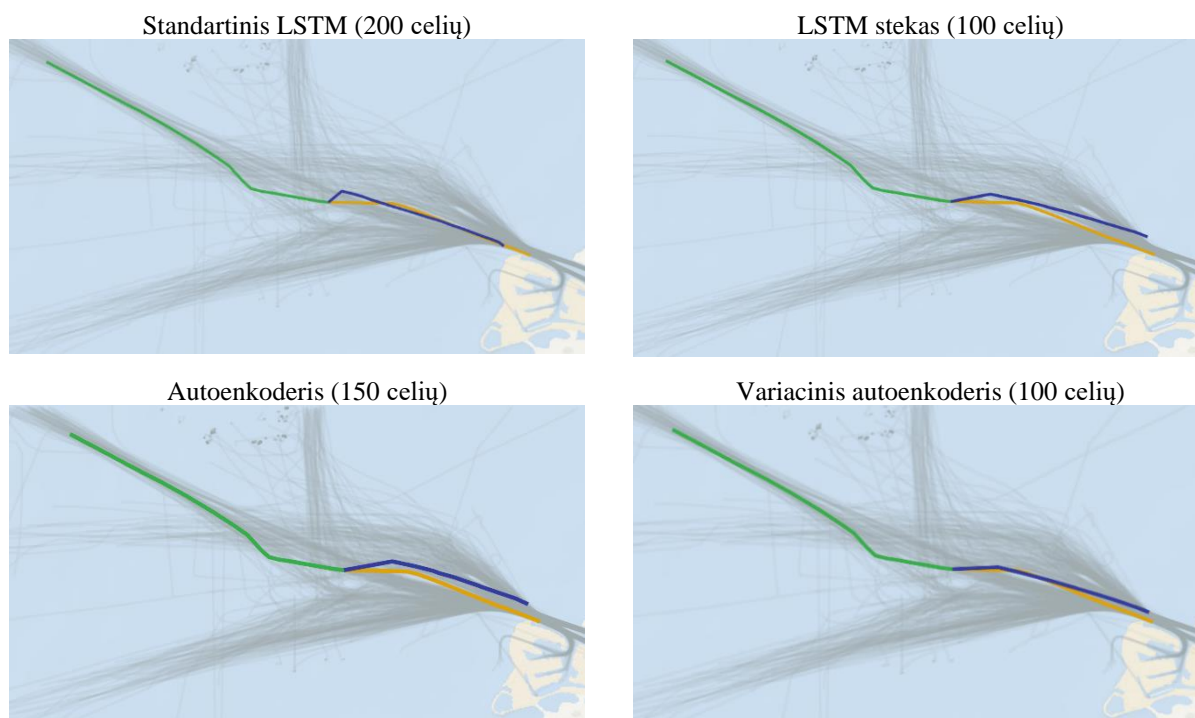
Iš pateiktos lentelės matyti, kad duomenys daugiau ar mažiau tolygiai pasiskirsto. Standartinis LSTM tiksliausiai veikia su 200 celėmis, LSTM stekas tiksliausias su 100 celėmis, autoenkoderis su 150 celėmis, o variacinis autoenkoderis su 100 celių LSTM sluoksniuose.

Vizualizuoti ir apibendrinti rezultatai pateikti 31 paveiksle. Grafiškai matyti, kad duomenų dimensijas mažinantys algoritmas veikia geriau. Įdomus reiškinys įvyksta autoenkoderio architektūroje. Naudojant 100 celių, jos suspaudžiamos iki 25 celių – dimensijos per mažos. Naudojant 200 celes, jos suspaudžiamos iki 50 celių – dimensijos per didelės. Rekonstrukcija tiksliausiai atliekama naudojant 150 celes, kurios paslėptame sluoksnyje užkoduojamos iki 37 celių.



31 pav. LSTM architektūrų tikslumas laivų eigos duomenyse

Sukurtų modelių testavimui atsitiktinai iš testavimo duomenų imties pasirinkta nauja seka. Laivo sekos prognozė atlikta su kiekviena LSTM architektūra ir tiksliausiai pritaikytu LSTM celės dydžiu. Prognozės vizualizacija pateikta 32 paveikslėlyje. Sekos prognozė ženkliai tikslesnė naudojant tik eigoje esančių laivų stebėjimus. Sekos ilgį sudaro 4 val., iš kurių 1:30 val. trukmė yra prognozuojama. Žalia spalva – įvestis, geltona spalva – išvestis, mėlyna spalva – prognozė.



32 pav. LSTM architektūrų sekos vizualizacija be stovinčių laivų

3.4 Skyriaus apibendrinimas

LSTM giliųjų neuroninių tinklų architektūros apmokytos keičiant celių kiekį ir skirtingomis duomenų rinkinių sekomis: stovinčių ir eigoje esančių laivo duomenų; laivo eigos duomenų be stovinčių laivų. Stovintis laivas laikomas tada, kai greičio požymis AIS duomenyse yra lygus 0. Tinklai apmokyti normalizuotais duomenimis, apskaičiuoti statistiniai požymių įverčiai. Mokymo procesas verifikuojamas validacijos duomenų imtimi.

Tinklų mokymosi trukmė tiesiogiai priklauso nuo LSTM celių apimties, kadangi didėjant jų kiekiui, didėja skaičiuojamų parametrų kiekis. Ilgiausiai apmokyti tinklus užtrunka autoenkoderio ir LSTM steko architektūrose. Eksperimentų tiksliausi rezultatai pateikiami apibendrintoje 7 lentelėje.

7 lentelė. Apibendrinti architektūrų tikslumo įvertinimai

| Duomenų rinkinys Architektūra | Stovintys ir eigoje esantys laivai | | Laivai eigoje |
|----------------------------------|--|---|---|
| | <i>Visi požymiai</i> | <i>Koordinacijų požymiai</i> | <i>Koordinacijų požymiai</i> |
| Standartinis LSTM | $6,692 \times 10^{-3}$ | $1,22 \times 10^{-4}$ | $3,08 \times 10^{-4}$ |
| LSTM stekas | $6,605 \times 10^{-3}$ | $9,37 \times 10^{-5}$ | $2,66 \times 10^{-4}$ |
| AE | <u>$6,577 \times 10^{-3}$</u> | <u>$8,94 \times 10^{-5}$</u> | <u>$2,51 \times 10^{-4}$</u> |
| VAE | $6,701 \times 10^{-3}$ | $1,04 \times 10^{-4}$ | $2,58 \times 10^{-4}$ |

Visais atvejais, įvertinus testavimo duomenų imties tikslumą, mažiausia klaidos vertė gauta AE architektūroje, tačiau patikrinus atsitiktines laivo eigos prognozuojamas sekas, efektyviausia kelių laiko žingsnių plaukimo kryptis išgauta, kai duomenyse išvalomas triukšmas – stovintys laivai. Pastebėta, kad AE tikslumas priklauso nuo optimalaus LSTM celės dydžio parinkimo, kad tinkamai būtų sumažintos dimensijos, tuo tarpu VAE yra lankstesnis, dėl savo struktūros geba lanksčiau interpoliuoti duomenis. AE architektūros tikslumas geresnis už VAE 2,71 % – palyginimas atliktas remiantis MAPE funkcija (8).

Šio darbo tyrimas ateityje galėtų būti plėtojamas: į duomenų rinkinį įtraukiant oro sąlygas ir keičiant hiperparametrų rinkinius ar funkcijas, pavyzdžiui, išbandyti RELU funkcijų modifikacijas. Taip pat, tęsti dimensijas mažinančių architektūrų analizę.

IŠVADOS

1. Išnagrinėjus naujausias mokslines publikacijas išsiaiškinta, kad transporto eismo prognozavimui naudojamos skirtingos giliojo mokymosi rūšys: skatinamasis, prižiūrimas arba iš dalies prižiūrimas mokymas, o matematiniai modeliai sunkiau pritaikomi dėl skaičiavimo sudėtingumo, ypač didžiuosiuose duomenyse. Taip pat, nustatyta, kad rekurentinių tinklų tikslumas yra mažesnis ilgalaikėje prognozėje, tačiau problema sprendžiama patobulintuose LSTM tinkluose, kurie geba valdyti atminties srautus. Nuspręsta ištirti skirtingas LSTM architektūras ir jų tinkamą hiperparametrų parinkimą, tiksliausios laivo eigos prognozės atlikimui: standartinis LSTM, LSTM stekas, AE ir VAE.

2. Sukurtos LSTM architektūros grindžiamos sekomis, o jų srauto informacija valdoma LSTM celėmis. Remiantis didžiųjų pasaulio terminalų statistine informacija buvo pasirinktas tiriamas Olandijos uostų regionas. Didžiųjų AIS duomenų centralizavimui ir pirminiam jų valymui pritaikyti ETL procesai. Atlikus jų analizę duomenys buvo struktūrizuoti į vienodų dimensijų matricas, sudarant du duomenų rinkinius, nes pastabėta, kad tik $\frac{1}{4}$ įrašų yra laivai eigoje, todėl dėl triukšmo padidėja netikslios prognozės rizika. Vidutiniškai viena seka gali prognozuoti apie 1:30 val. trukmės tolimesnę laivo eigą. Tinklų analizė atlikta keičiant duomenis ir architektūrų hiperparametrus: LSTM celių dydžius.

3. Sukūrus LSTM architektūrų modelius, buvo įvertinti abu (eigoje esančių bei stovinčių ir tik stovinčių laivų) duomenų rinkiniai. Gautas tinklų tikslumas išgautas konkrečiai šiame tyrime nagrinėtoje duomenų imtyje.

- Laivų sekos vizualizacijose pastebėta, kad stovinčių laivų duomenys ganėtinai išbalansavo tinklo svorius, todėl visais atvejais gauta prognozė ir efektyvumas buvo klaidingi dėl triukšmo duomenyse. Šiame duomenų rinkinyje atlikti praradimo funkcijos reikšmių įvertinimai parodė, kad autoenkoderio architektūra su visais laivo požymiais ($6,577 \times 10^{-3}$, naudojant 100 LSTM celių) ir su geografinės pozicijos požymiais ($8,94 \times 10^{-5}$, naudojant 50 LSTM celių) yra arčiausiai tikrosios trajektorijos reikšmių.

- Įvertinus duomenų rinkinį, kai jame nėra stovinčių laivų, tinklų tikslumas akivaizdžiai pagerėja. Tyrimas parodė, kad autoenkoderio tikslumas priklauso nuo optimalaus LSTM celės dydžio parinkimo sluoksnyje, kad tinkamai būtų sumažintos dimensijos, tuo tarpu variacinis autoenkoderis yra mažiau priklausomas nuo LSTM celių, nes dėl savo struktūros geba lanksčiau interpoluoti duomenis. Tikslumo skirtumas tarp AE ir VAE architektūrų tik 2,71 %.

LITERATŪRA

1. „Jūreivystės terminų vartosenos duomenynas“. Internetinis jūreivystės terminų žodynas, prieiga per internetą: <mano.ku.lt/jtvd>. Klaipėdos universiteto Lietuvių kalbos katedros – Klaipėda, 2015.
2. „Visuotinė lietuvių enciklopedija“. Internetinė enciklopedija, prieiga per internetą: <www.vle.lt>. Mokslo ir enciklopedijų leidybos centras – Vilnius, 2017.
3. A. Seltmann. „Marine Insurance – Casualty trends“. *CEFOR publications 2019 Nordic marine insurance statistics*. The Nordic Association of Marine Insurers, 2019.
4. H. Rong, A.P. Teixeira, C. Guedes Soares. „Ship trajectory uncertainty prediction based on a Gaussian Process model“. *Ocean Engineering*. Lisboa, Portugal, 2019, volume 182, p. 499-511.
5. H. Rong, A.P. Teixeira, C. Guedes Soares. „Data mining approach to shipping route characterization and anomaly detection based on AIS data“. *Ocean Engineering*. Lisboa, Portugal, 2020, volume 198.
6. J. Xue, P.H.A.J.M. V. Gelder, G. Reniers, E. Papadimitriou ir C. Wu. „Multi attribute decision-making method for prioritizing maritime traffic safety influencing factors of autonomous ships’ maneuvering decisions using grey and fuzzy theories“. *Safety Science*. Hubei, China, 2019, volume 120, p. 323-340.
7. Y. Cheng, W. Zhang. „Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels“. *Neurocomputing*. Shanghai Jiaotong University, China, 2017, p. 1-11.
8. X. Liu, A. Gherbi, W. Li ir M. Cheriet. „Multi Features and Multi-time steps LSTM Based Methodology for Bike Sharing Availability Prediction“. *Procedia Computer Science, The 14th International Conference on Future Networks and Communications (FNC)*. Halifax, Canada, 2019, volume 155, p. 394-401.
9. B. Yang, S. Sun, J. Li, X. Lin, ir Y. Tian. „Traffic flow prediction using LSTM with feature enhancement“. *Neurocomputing*. China, 2019, volume 332, p. 320-327.
10. T. Bogaerts, A. D. Masegosa, J. S. Angarita-Zapata, E. Onieva ir P. Hellinckx. „A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data“. *Transportation Research Part C*. Bilbao, Spain, 2020, volume 112, p. 62-77.
11. N. C. Petersen, F. Rodrigues, F. C. Pereira. „Multi-output bus travel time prediction with convolutional LSTM neural network“. *Expert Systems with Applications*. Kongens Lyngby, Denmark, 2018, volume 120, p. 426-435.
12. P. Li, M. Abdel-Aty, J. Yuan. „Real-time crash risk prediction on arterials based on LSTM-CNN“. *Accident Analysis and Prevention*. Orlando, FL32816, UnitedStates, 2020, volume 135, p. 1-9.
13. G. Boquet, A. Morell, J. Serrano, J. L. Vicario. „A variational autoencoder solution for road traffic forecasting systems: Missing data imputation, dimension reduction, model selection and anomaly detection“. *Transportation Research Part C*. Barcelona, Spain, 2020, volume 115, p. 1-18.
14. Z. Xie, X. Liu. „LSTM networks for vessel traffic flow prediction inland waterway“. *2018 IEEE International Conference on Big Data and Smart Computing*. Shanghai, China, 2018, p. 418-425, DOI: 10.1109/BigComp.2018.00068.

15. United Nations. „Review of Maritime Transport 2019“. *United Nations Conference on Trade and Development*. Geneva, 2019, ISSN: 0566-7682.
16. J. Venskys, P. Treigys, J. Markevičiūtė. „Detecting Maritime traffic anomalies with long-short term memory recurrent neural network“. *Nonlinear Analysis: Modelling and Control*. Vilnius, 2020, ISSN: 1392-5113.
17. A. A. Rizal, S. Soraya ir M. Tajuddin. „Sequence to sequence analysis with long short term memory for tourist arrivals prediction“. *Journal of Physics: Conference Series*. Indonesia, 2018, p. 1-8, DOI:10.1088/1742-6596/1211/1/012024.
18. R. Vinayakumar, KP. Soman, P. Prabakaran. „Applying Deep Learning Approaches for NetworkTraffic Prediction“. *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. Amrita Vishwa Vidyapeetham, India, 2017, p. 2353-2358, ISBN: 978-1-5090-6367-3
19. Aurélien Géron. „Hands-On Machine Learning with Scikit-Learn & TensorFlow“. *Concepts, Tools, and Techniques to Build Intelligent Systems: First Edition*. JAV, Sebastopol, CA, 2017, ISBN-13: 978-1491962299.
20. R. Zhao, J. Wang, R. Yan ir K. Mao. „Learning to Monitor Machine Health with Convolutional Bi-Directional LSTM Networks“. *Article in Sensors*. MDPI, Basel, Switzerland, 2017, p. 1-18, DOI: 10.3390/s17020273.
21. Y. Li, ir H. Cao. „Prediction for Tourism Flow based on LSTM Neural Network“. *Procedia Computer Science, 2017 International Conference on Identification, Information and Knowledge in the Internet of Things*. China, 2018, volume 129, p. 277-283.
22. A. G. Salman, Y. Heryadi, E. Abdurahman, W. Suparta. „Single Layer & Multi-layer Long Short-Term Memory (LSTM) Model with Intermediate Variables for Weather Forecasting“. *3rd International Conference on Computer Science and Computational Intelligence 2018*. Jakarta, Indonesia, 2018, volume 135, p. 89-98.
23. J. Venskys, P. Treigys. „Preparation of training data by filling in missing vessel type data using deep multi-stacked LSTM neural network for abnormal marine traffic evaluation“. *ITISE 2019 : International Conference on Time Series and Forecasting : proceedings of abstracts*. Granada, Spain, 2019, p. 38, ISBN 9788417970796.
24. Z. Cui, R. Ke, Z. Pu, Y. Wang. „Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction“. *International Workshop on Urban Computing*. University of Washington, Seattle, USA, 2017, p. 1-11.
25. B. Murray, L. P. Perera. „A dual linear autoencoder approach for vessel trajectory prediction using historical AIS data“. *Ocean Engineering*. Tromsø, Norwa, 2020, volume 209, p. 1-13.
26. D. Moitra ir R. Kr. Mandal. „Automated AJCC (7th edition) staging of non-small cell lung cancer (NSCLC) using deep convolutional neural network (CNN) and recurrent neural network (RNN)“. *Health*

Information Science and Systems. Siliguri, West Bengal, India, 2019, p. 1-12, DOI: 10.1007/s13755-019-0077-1.

27. Shan Suthaharan. „Machine Learning Models and Algorithms form Big Data Classification“. *Integrated Series in Information Systems 36*. Department of Computer Science, Greensboro, NC, USA, 2016, ISSN 2197-7968 (electronic).

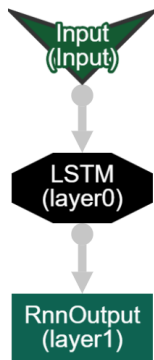
28. S. Poornima, M. Pushpalatha. „Prediction of Rainfall Using Intensified LSTM Based Recurrent Neural Network with Weighted Linear Units“. *Atmosphere*. Kattangulathur, India, 2019, volume 10, p. 1-18.

29. W. Wei, H. Wu ir H. Ma. „An AutoEncoder and LSTM-Based Traffic Flow Prediction Method“. *Article in Sensors*. Beijing 100876, China, 2019, volume 19, p. 1-16.

PRIEDAI

1 priedas. LSTM architektūrų struktūra ir sluoksnių parametrai

Standartinis LSTM



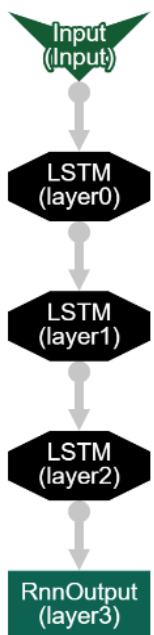
Layer Information

| Layer Name | layer0 | layer1 |
|---------------------|-----------------|-----------------|
| Layer Type | LSTM | RnnOutput |
| Input Size | 30 | 50 |
| Layer Size | 50 | 20 |
| # Parameters | 16200 | 1020 |
| Weight Init. | WeightInitRelu' | WeightInitRelu' |
| Updater | Adam | Adam |
| Activation Function | relu | tanh |

| LayerName (LayerType) | nIn,nOut | TotalParams | ParamsShape |
|-------------------------|----------|-------------|------------------------------------|
| layer0 (LSTM) | 30,50 | 16200 | W:{30,200}, RW:{50,200}, b:{1,200} |
| layer1 (RnnOutputLayer) | 50,20 | 1020 | W:{50,20}, b:{1,20} |

Total Parameters: 17220
 Trainable Parameters: 17220
 Frozen Parameters: 0

LSTM stekas



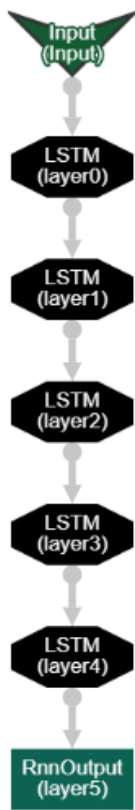
Layer Information

| Layer Name | layer0 | layer1 | layer2 | layer3 |
|---------------------|----------------|----------------|----------------|----------------|
| Layer Type | LSTM | LSTM | LSTM | RnnOutput |
| Input Size | 30 | 150 | 150 | 150 |
| Layer Size | 150 | 150 | 150 | 20 |
| # Parameters | 108600 | 180600 | 180600 | 3020 |
| Weight Init. | WeightInitRelu | WeightInitRelu | WeightInitRelu | WeightInitRelu |
| Updater | Adam | Adam | Adam | Adam |
| Activation Function | relu | relu | relu | tanh |

| LayerName (LayerType) | nIn,nOut | TotalParams | ParamsShape |
|-------------------------|----------|-------------|--------------------------------------|
| layer0 (LSTM) | 30,150 | 108600 | W:{30,600}, RW:{150,600}, b:{1,600} |
| layer1 (LSTM) | 150,150 | 180600 | W:{150,600}, RW:{150,600}, b:{1,600} |
| layer2 (LSTM) | 150,150 | 180600 | W:{150,600}, RW:{150,600}, b:{1,600} |
| layer3 (RnnOutputLayer) | 150,20 | 3020 | W:{150,20}, b:{1,20} |

Total Parameters: 472820
 Trainable Parameters: 472820
 Frozen Parameters: 0

Autoenkoderis



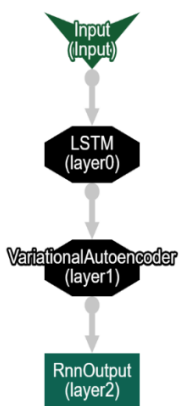
Layer Information

| Layer Name | layer0 | layer1 | layer2 | layer3 | layer4 | layer5 |
|---------------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Layer Type | LSTM | LSTM | LSTM | LSTM | LSTM | RnnOutput |
| Input Size | 30 | 150 | 75 | 37 | 75 | 150 |
| Layer Size | 150 | 75 | 37 | 75 | 150 | 20 |
| # Parameters | 108600 | 67800 | 16724 | 33900 | 135600 | 3020 |
| Weight Init. | WeightInitRelu | WeightInitRelu | WeightInitRelu | WeightInitRelu | WeightInitRelu | WeightInitRelu |
| Updater | Adam | Adam | Adam | Adam | Adam | Adam |
| Activation Function | relu | relu | relu | relu | relu | tanh |

```

=====
LayerName (LayerType)      nIn,nOut  TotalParams  ParamsShape
=====
layer0 (LSTM)              30,150    108600      W:{30,600}, RW:{150,600}, b:{1,600}
layer1 (LSTM)             150,75    67800       W:{150,300}, RW:{75,300}, b:{1,300}
layer2 (LSTM)              75,37     16724       W:{75,148}, RW:{37,148}, b:{1,148}
layer3 (LSTM)              37,75     33900       W:{37,300}, RW:{75,300}, b:{1,300}
layer4 (LSTM)              75,150    135600      W:{75,600}, RW:{150,600}, b:{1,600}
layer5 (RnnOutputLayer)   150,20    3020        W:{150,20}, b:{1,20}
=====
Total Parameters: 365644
Trainable Parameters: 365644
Frozen Parameters: 0
=====
    
```

Variacinis autoenkoderis



Layer Information

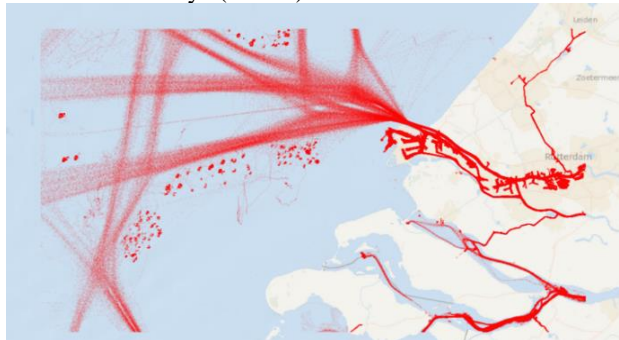
| Layer Name | layer0 | layer1 | layer2 |
|---------------------|----------------|------------------------|----------------|
| Layer Type | LSTM | VariationalAutoencoder | RnnOutput |
| Input Size | 30 | 150 | 150 |
| Layer Size | 150 | 150 | 20 |
| # Parameters | 108600 | 113324 | 3020 |
| Weight Init. | WeightInitRelu | WeightInitRelu | WeightInitRelu |
| Updater | Adam | Adam | Adam |
| Activation Function | relu | tanh | tanh |

```

=====
LayerName (LayerType)      nIn,nOut  TotalParams  ParamsShape
=====
layer0 (LSTM)              30,150    108600      W:{30,600}, RW:{150,600}, b:{1,600}
layer1 (VariationalAutoencoder)  150,150  113324      e0W:{150,150}, e0b:{1,150}, e1N:{150,75}, e1b:{1,75}, e2N:{75,37}, e2b:{1,37}, pZMeanb:{37,150}, pZMeanb:{1,150}, pZLogStd2h:{37,150}, pZLogStd2b:{1,150}, d0W:{150,37}, d0b:{1,37}, d1N:{37,75}, d1b:{1,75}, d2N:{75,150}, d2b:{1,150}, pXZW:{150,300}, pXZb:{1,300}
layer2 (RnnOutputLayer)   150,20    3020        W:{150,20}, b:{1,20}
=====
Total Parameters: 224944
Trainable Parameters: 224944
Frozen Parameters: 0
=====
    
```

2 priedas. Didžiųjų AIS duomenų imties pasiskirstymo santykis

Visi duomenys (100 %) – 20 826 380 koordinatės



Mokymo duomenys (50 %) – 10 413 190 koordinatės



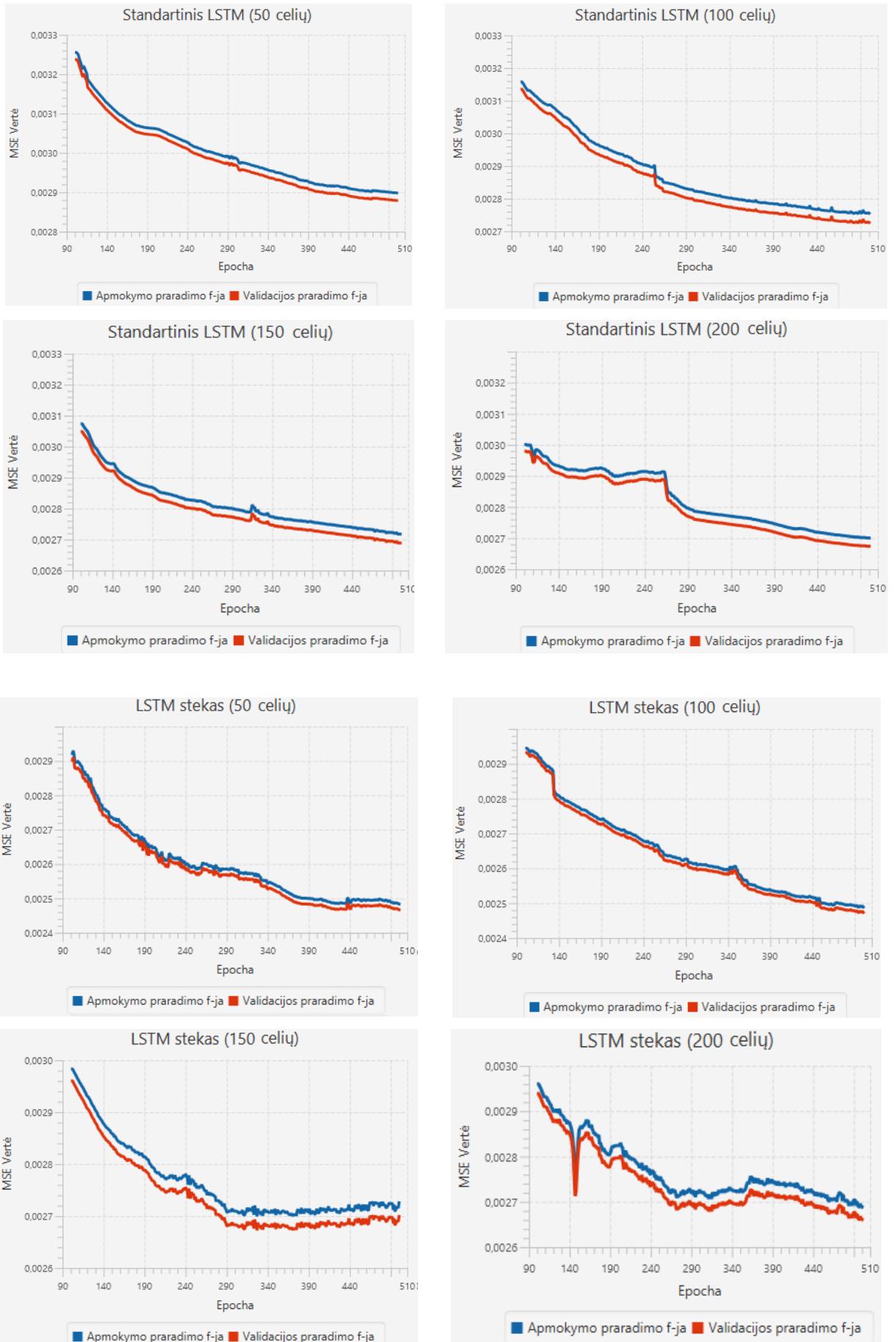
Validacijos duomenys (25 %) – 5 206 595 koordinatės

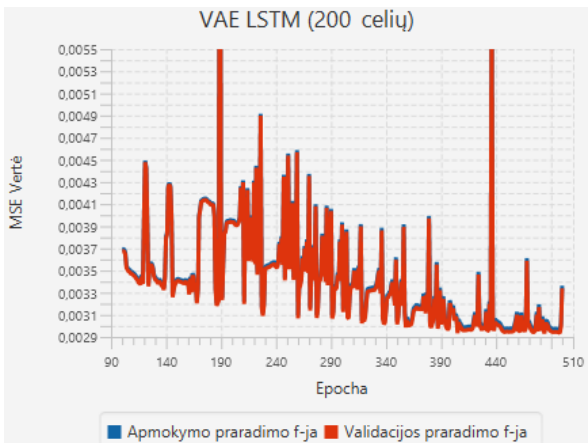
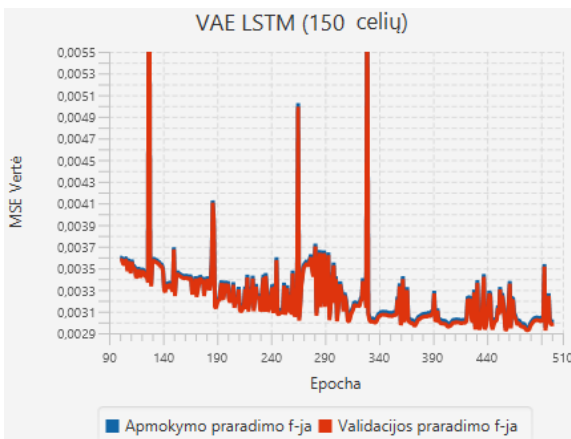
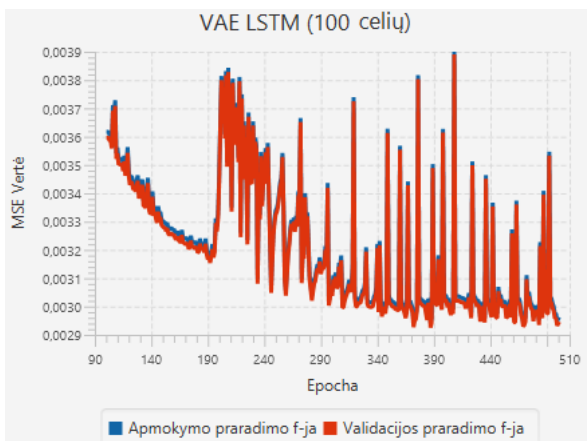
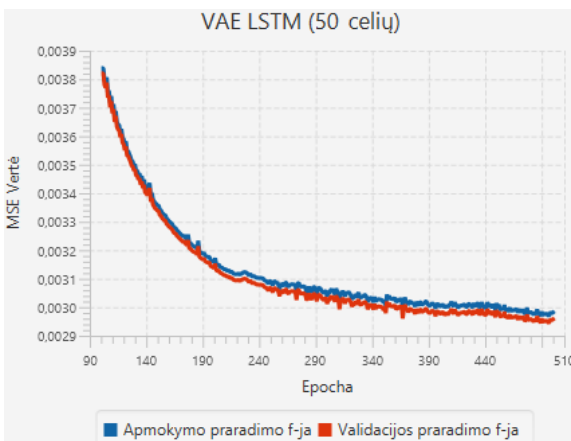
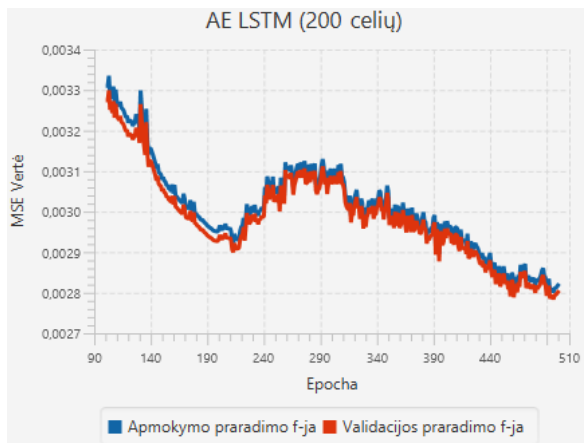
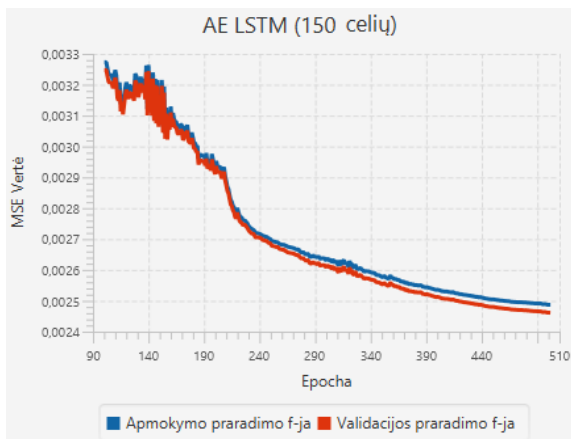
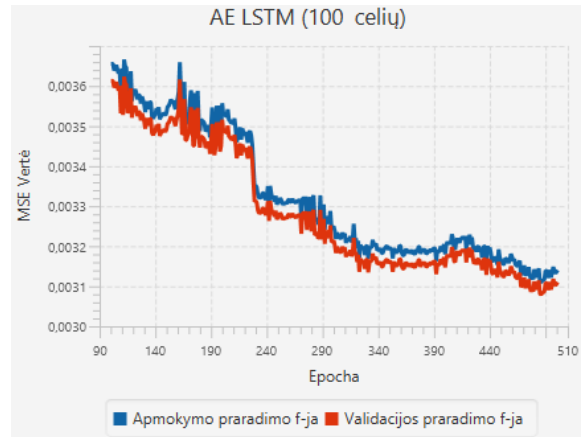
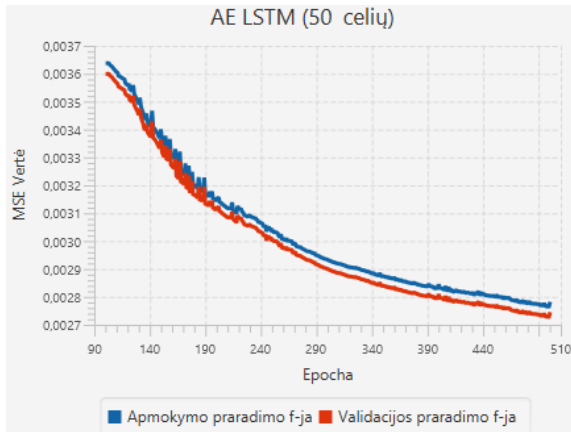


Testavimo duomenys (25 %) – 5 206 595 koordinatės



3 priedas. Giliųjų tinklų mokymosi ir validacijos progresas (laivai eigoje)





4 priedas. Laivo eigos prognozės kitimas mokymosi procese

