

KLAIPĖDOS UNIVERSITETAS

Jūros technologijų ir gamtos mokslų fakultetas

Informatikos ir statistikos katedra

**DIDŽIAISIAIS DUOMENIMIS
APSIMOKANČIO ALGORITMO IR
POSISTEMIO NEĮPRASTAM LAIVŲ
EISMUI ATPAŽINTI JŪRŲ UOSTE
KŪRIMAS**

**CREATION OF ALGORITHM FOR
MACHINE LEARNING ON BIG DATA AND
DEVELOPMENT OF SUBSYSTEM FOR
RECOGNITION OF ABNORMAL MARITIME
TRAFFIC IN SEAPORT**

Magistro baigiamasis darbas

Autorius

TMNTS14 gr. stud. Julius Venskus

Vadovas

Prof. dr. Arūnas Andziulis

Konsultantas

Lekt. dr. Mindaugas Kurmis

Klaipėda, 2016

PADĖKA

Dėkoju KU informatikos ir statistikos katedros darbuotojams, darbo vadovui profesoriui dr. Arūnui Andziuliui, konsultantui dr. Mindaugui Kurmiui už kantrybę, pagalbą ir suteiktas žinias. Ypač dėkoju reikliems ir daug žinių suteikusiems dėstytojams dr. Jurij Tekutov, doc. dr. Violetai Bulbenkienei, prof. dr. Vitalij Denisov. Nuoširdžiai dėkoju Viešpačiui Dievui, savo šeimai ir artimiesiems už paramą, supratingumą ir sąlygų tęsti mokslus sudarymą.

ANOTACIJA

Šiuo metu sukurti metodai ir algoritmai neįprastam jūrų laivų eismui aptikti nėra pritaikyti dirbti su didžiais duomenimis. Dėl šios priežasties įprasto eismo modelio apmokymas vyksta neleistinai ilgai arba naudojamas mažesnis duomenų kiekis, ko rezultate krenta klasifikavimo preciziškumas, o tai lemia kad neįprasto eismo aptikimo algoritmai negali sumažinti laivų eismo kontrolės operatorių darbo krūvio.

Darbe sukurtas didžiais duomenimis apsimokantis algoritmas ir posistemis, kurio pagrindą sudaro modifikuotas Kohoneno SOM tinklas. SOM tinklas buvo modifikuotas pridendant prie neuronų virtualaus feromono žymę, sudarant atstumo tarp neuronų skaičiavimo funkciją ne Euklido erdvėje. Neuronų kaimynystės funkcijai panaudota „Meksikietiškos kepurės“ funkcija. SOM pritaikymui apsimokyti didžiais duomenimis buvo panaudotas tiesioginis mašininis mokymas su dviejų SOM tinklų sinteze. Sukurto algoritmo pagrindu sudarytas neįprasto laivų eismo aptikimo posistemis, kuris integruojasi į laivų eismo ir kontrolės sistemą. Posistemio ir algoritmo verifikacijai buvo sukurtas prototipas, kurio verifikacija ir algoritmo tyrimas atliktas naudojant Klaipėdos jūrų uosto laivų eismo didieji duomenis ir pasiektas 90% klasifikavimo preciziškumas esant 80% klasifikavimo jautrumui.

Raktažodžiai: didieji duomenys, jūrų laivų eismas, neįprastas eismas, neprižiūrimas mokymas, SOM, jūrų uostas, algoritmas, virtualus feromonas, Klaipėdos jūrų uostas, Google maps API.

ABSTRACT

At present, methods and algorithms that have been developed to recognise abnormal maritime traffic are not suitable for work with big data. For this reason, the learning of normal traffic pattern takes too much time. If smaller data is used, the precision of classification suffers, therefore algorithms for recognition of abnormal maritime traffic cannot reduce the workload for traffic control operators.

The paper presents both an algorithm for machine learning on big data subsystem and subsystem which is based on modified Kohonen SOM network. SOM network was modified by adding a virtual pheromone tag to neurons thus creating a distance calculation function between neurons in non-Euclidean space. The Mexican hat wavelet was used for neural neighbourhood function. Direct machine learning with a fusion of two SOM networks was used for adaptation of SOM for learning from big data. A subsystem for recognition of abnormal maritime traffic was created on basis of the created algorithm. The subsystem integrates into maritime traffic and monitoring system. A prototype was created for verification of the subsystem and algorithm. The prototype verification and field test for algorithm was carried out using the big data of Klaipėda seaport maritime traffic. The classification precision of 90% at 80% sensitivity of classification was achieved.

Keywords: big data, maritime traffic, abnormal traffic, unsupervised learning, SOM, seaport, algorithm, virtual pheromone, Klaipėda Seaport, Google Maps API

SANTRAUKA

Julius Venskus. Didžiaisiais duomenimis apsimokančio algoritmo ir posistemio neįprastam laivų eismui atpažinti jūrų uoste kūrimas. Techninių informacinių sistemų inžinerijos magistro studijų baigiamasis darbas / darbo vadovas: Prof. Dr. Andziulis. Konsultantas: lekt. dr. M. Kurmis; Klaipėdos universitetas, Jūros technologijų ir gamtos mokslų fakultetas, Informatikos ir statistikos katedra. – Klaipėda, 2016. 54p. Paveikslėlių 35, lentelių: 5

Šiuo metu sukurti metodai ir algoritmai neįprastam jūrų laivų eismui aptikti nėra pritaikyti dirbti su didžiaisiais duomenimis. Dėl šios priežasties įprasto eismo modelio apmokymas vyksta neleistinai ilgai arba naudojamas mažesnis duomenų kiekis, ko rezultate krenta klasifikavimo preciziškumas, o tai lemia kad neįprasto eismo aptikimo algoritmai negali sumažinti laivų eismo kontrolės operatorių darbo krūvio.

Darbo tikslas: sukurti didžiaisiais duomenimis apsimokantį algoritmą ir posistemį neįprastam laivų eismui aptikti jūrų uoste egzistuojančioms laivų stebėjimo ir kontrolės sistemoms, kurių klasifikavimo preciziškumas aukštesnis negu 88% ir algoritmo laiko sudėtingumas mažesnis negu $\Theta(n^6)$.

Uždaviniai: išanalizuoti mokslinę literatūrą esamų neįprasto laivų eismo aptikimo tema ir pasirinkti tobulintinus metodus bei algoritmus, kuriems reikalingi pakeitimai, pritaikant dirbti su laivų eismo didžiaisiais duomenimis; sukurti didžiaisiais duomenimis apsimokantį algoritmą ir posistemį neįprastam laivų eismui aptikti egzistuojančiose laivų stebėjimo ir kontrolės sistemose; verifikuoti sukurtą prototipą ir ištirti algoritmo klasifikavimo preciziškumą ir jautrumą, naudojant didžiųjų duomenų imtį iš Klaipėdos Valstybinio jūrų uosto stebėjimo ir kontrolės sistemos.

Darbe sukurtas didžiaisiais duomenimis apsimokantis algoritmas ir posistemis, kurių pagrindą sudaro modifikuotas Kohoneno SOM tinklas. SOM tinklas buvo modifikuotas, pridėdant prie neuronų virtualaus feromono žymę, sudarant atstumo tarp neuronų skaičiavimo funkciją ne Euklido erdvėje. Neuronų kaimynystės funkcijai panaudota „Meksikietiškos kepurės“ funkcija. SOM pritaikymui apsimokyti didžiaisiais duomenimis buvo panaudotas tiesioginis mašininis mokymas su dviejų SOM tinklų sinteze. Siūlomo algoritmo pagrindu sukurtas neįprasto laivų eismo aptikimo posistemis, kuris integruojasi į laivų eismo ir kontrolės sistemą. Posistemio ir algoritmo verifikacijai buvo sukurtas prototipas, kurio verifikacija ir algoritmo tyrimas atliktas, naudojant Klaipėdos jūrų uosto laivų eismo didieji duomenys ir pasiektas 90% klasifikavimo preciziškumas, esant 80% klasifikavimo jautrumui.

Raktažodžiai: didieji duomenys, jūrų laivų eismas, neįprastas eismas, neprižiūrimas mokymas, SOM, jūrų uostas, algoritmas, virtualus feromonas, Klaipėdos jūrų uostas, Google maps API.

SUMMARY

Julius Venskus. Creation of algorithm for machine learning on big data and development of subsystem for recognition of abnormal maritime traffic in seaport. Technical Information Systems Engineering Master's thesis / supervisor: Prof. Dr. Andziulis. Advisor: lecturer Dr. M. Kurmis; Klaipeda University, Faculty of Marine Technology and Natural Sciences, Department of Informatics and Statistics. - Klaipeda, 2016. 54p. 35 pictures, 5 tables

At present, methods and algorithms that have been developed to recognise abnormal maritime traffic are not suitable for work with big data. For this reason, the learning of normal traffic pattern takes too much time. If smaller data is used, the precision of classification suffers, therefore algorithms for recognition of abnormal maritime traffic cannot reduce the workload for traffic control operators.

Research objectives: Creation of an algorithm for machine learning on big data and development of subsystem for recognition of abnormal maritime traffic in seaport for existing maritime traffic and monitoring systems with classification precision higher than 88% and the time complexity of algorithm less than $\Theta(n^6)$.

Tasks: To analyse current scientific literature on existing algorithms for recognition of abnormal maritime traffic and choose methods and algorithms for improvement by changing and adapting them for work with big data of maritime traffic To create algorithm for machine learning on big data and to develop subsystem for recognition of abnormal maritime traffic for maritime traffic and monitoring systems To verify the created prototype and investigate precision and sensitivity of classification by algorithm using a big data sample from traffic and monitoring system at Klaipeda State Seaport

The paper presents both an algorithm for machine learning on big data subsystem and subsystem which is based on modified Kohonen SOM network. SOM network was modified by adding a virtual pheromone tag to neurons thus creating a distance calculation function between neurons in non-Euclidean space. The Mexican hat wavelet was used for neural neighbourhood function. Direct machine learning with a fusion of two SOM networks was used for adaptation of SOM for learning from big data. A subsystem for recognition of abnormal maritime traffic was created on basis of the created algorithm. The subsystem integrates into maritime traffic and monitoring system. A prototype was created for verification of the subsystem and algorithm. The prototype verification and field test for algorithm was carried out using the big data of Klaipeda seaport maritime traffic. The classification precision of 90% at 80% sensitivity of classification was achieved.

Keywords: big data, maritime traffic, abnormal traffic, unsupervised learning, SOM, seaport, algorithm, virtual pheromone, Klaipeda Seaport, Google Maps API

PAVEIKSLĖLIŲ SĄRAŠAS

1 pav.	AIS įrenginio priekinis vaizdas ir atvaizdavimas elektroniniame jūrlapyje.....	12
2 pav.	AIS sistemos ryšio perdavimo kanalai	13
3 pav.	Scenarijaus analizės procesas [9].....	15
4 pav.	Anomalijų aptikimas laivų eismo trajektorijose [11].....	16
5 pav.	Geografinis laivų eismo sudėtingumo pasiskirstymas Šenčeno jūros kanale. [4]	17
6 pav.	Gdansko įlankos laivų eismo lauko potencialų vizualizavimas [12].	18
7 pav.	Anomalijų aptikimas, priklausomai nuo sektoriaus dydžio [12]	18
8 pav.	Dvimačio SOM tinklo loginė schema	21
9 pav.	SOM tinklo struktūra: a) stačiakampė, b) šešiakampė	22
10 pav.	Neuronų vektorių perskaičiavimas pagal atstumą iki neurono laimėtojo.	24
11 pav.	Binarinio klasifikatoriaus rodikliai	26
12 pav.	SOM tinklo struktūra: a) stačiakampė, b) šešiakampė	28
13 pav.	Modifikuotas SOM tinklas su virtualaus feromono žyme.....	29
14 pav.	Didesnio tikėtimumo transformacija į virtualaus feromono intensyvumą.	31
15 pav.	Tiesioginio mokymosi duomenų imtys	32
16 pav.	Dviejų SOM tinklų sintezė	33
17 pav.	WGS84 koordinacių sistema	34
18 pav.	Laivo judėjimo krypties poliarinių koordinacių sistema	35
19 pav.	Atstumo matavimas tarp dviejų poliarinių vektorių.....	36
20 pav.	„Meksikietiškos kepurės“ kaimynystės funkcijos grafikas	37
21 pav.	„Meksikietiškos kepurės“ kaimynystės funkcijos kitimas pagal epochos numerį	38
22 pav.	Neįprasto laivų eismo aptikimo algoritmo diagrama	39
23 pav.	Neįprasto eismo aptikimo algoritmo dalis, vykdoma nuotoliniame serveryje	40
24 pav.	Jūrų uosto laivų stebėjimo ir kontrolės sistemos architektūros diagrama.	43
25 pav.	Kuriamo posistemio architektūra loginiu požiūriu	44
26 pav.	Kuriamo posistemio architektūra diegimo požiūriu.....	46
27 pav.	Neįprasto laivų eismo aptikimo posistemio prototipo sandara.....	47
28 pav.	Tiriamasis regionas laivų eismo intensyvumo žemėlapyje	50
29 pav.	Modifikuoto SOM tinklo apmokymo pradžios vizualizacija	52
30 pav.	Apmokyto SOM tinklo dviejų laivų trajektorijos duomenimis vizualizacija	53
31 pav.	Modelio ir naujai sukurto SOM tinklų suliejimo vizualizacija.....	54
32 pav.	Krovinio laivo įprastinį judėjimą klasifikuojančių neuronų vizualizacija	55
33 pav.	Klasifikavimo preciziškumo kitimas, priklausomai nuo mokymo duomenų imties.....	56
34 pav.	Virtualaus feromono intensyvumo reikšmių dažnis neuronuose	57
35 pav.	Neįprasto eismo klasifikavimo ROC kreivė	57

LENTELIŲ SĄRAŠAS

1 lentelė.	Laivų eismo duomenų struktūra.....	13
2 lentelė.	Algoritmų ir metodų pasirinkimo kriterijai	19
3 lentelė.	Algoritmų ir metodų vertinimo lentelė.....	20
4 lentelė.	Įeinančio į SOM tinklą vektorišaus matmenų duomenų struktūra	51
5 lentelė.	Įeinančių duomenų imties paskirstymas.....	55

TERMINŲ IR SANTRUMPŲ ŽODYNĖLIS

AIS - automatinė identifikavimo sistema.

Algoritmo laiko sudėtingumas – (angl.: *algorithm time complexity*) laiko sudėtingumo skaičiavimas vertina kiek laiko reiktų tam tikrai problemai su tam tikru duomenų dydžiu spręsti efektyviausiu algoritmu. Tarkime, kad turint n bitų duomenų kiekį, problema išsprendžiama per n^2 žingsnių; tokia problema yra n^2 sudėtingumo. Iš tiesų, kiekvienas algoritmo įgyvendinimas spręstą problemą skirtingu žingsnių skaičiumi, todėl sąlyginis žingsnių skaičius (eilė) žymima $O(n^2)$ [1].

Anomalijos aptikimas - (angl.: *anomaly detection*) tai objektų identifikavimas, įvykiai ar reiškiniai, kurie neatitinka tikėtiniams modeliams, šablonams ar kitiems objektams duomenų imtyje [2].

Anomalijos slenkstis – tai reiškinio, įvykio ar objekto savybių parametrų rinkinys, nusakantis ribą tarp tikėtinų reiškinų, įvykių, objektų ir anomalijų (netikėtinų) [2].

Binarinis (dvimatis) klasifikavimas – Tai tokia klasifikavimo užduotis, kuri pagal nustatytą taisyklę suskirsto aibę į dvi grupes.

Didieji duomenys – duomenys, kurių apimtis tokia didelė, kad jai netinka tradiciniai saugojimo ir apdorojimo metodai bei programos. Pagrindiniai didžiųjų duomenų bruožai yra didelė apimtis (angl.: *high Volume*), didelė sparta (angl.: *high Velocity*), didelė įvairovė (angl.: *high Variety*) [3].

Epocha – SOM tinklo mokymo proceso dalis, kurios metu visi mokymo aibės vektoriai visi po vieną kartą pateikiami į tinklą nuosekliai arba atsitiktine tvarka [4].

Euklido erdvė – Realioji vektorinė erdvė, kurioje apibrėžta skaliarinė daugyba. Erdvė pasižymi komutatyvumu, distributyvumu, homogeniškumu ir reguliarumu savybėmis [5].

Farvateris – (ol.: *vaarwater*) ntk., laivakelis -paženklintas saugus vandens kelias laivams.

FP – (angl.: *false positive*) algoritmo klasifikavimo atvejis, kai laivo eismas klasifikuojamas teigiamai (kaip „neįprastas“), o ekspertas klasifikavo kaip neigiamą („neįprastą“), t.y. kaip klaidingai teigiamai nustatytą rezultatą (algoritmo ir eksperto klasifikavimas nesutampa).

FN – (angl.: *false negative*) algoritmo klasifikavimo atvejis, kai laivo eismas klasifikuojamas neigiamai (kaip „įprastas“), o ekspertas klasifikavo kaip teigiamą („neįprastą“), t.y. kaip klaidingai neigiamai nustatytą rezultatą (algoritmo ir eksperto klasifikavimas nesutampa).

IMO – Tarptautinė jūrų organizacija (International Maritime Organization)

Jūrinis incidentas – įvykis, susijęs su vidaus vandenų transporto priemonės panaudojimu, kai yra bent viena iš šių aplinkybių: vidaus vandenų transporto priemonė sugadinama taip, kad jos savybėms atkurti reikia ne daugiau kaip 72 valandų; vidaus vandenų transporto priemonė sugadina kranto statinį, įrenginį krante ar vandenyje, kuriems atkurti reikia ne daugiau kaip 48 valandų; vidaus vandenų transporto priemonė užplaukia ant seklumos ir nuo jos nuvelkama greičiau nei per 48 valandas; sugadinami navigaciniai ženklai (įrenginiai), povandeniniai laidai (kabeliai), vamzdynai; nesunkiai sužalojamas vienas ar daugiau asmenų [6].

Kaimynystės funkcija – SOM tinklo neuronų tarpusavio sąryšio priklausomybė.

Klasifikavimo preciziškumas – klasifikatoriaus kokybinis rodiklis nusakantis santykį tarp teisingai nustatytų teigiamų elementų ir viso teigiamai klasifikuotų grupėje [2].

Klasifikavimo jautrumas – klasifikatoriaus kokybinis rodiklis nusakantis santykį tarp teisingai nustatytų teigiamų elementų ir visų esančių teigiamų elementų [2].

Mašininis mokymas – kai kompiuterio programa mokosi iš patyrimo E atsižvelgdama į tam tikrą užduočių klasę T ir veiklos įvertį P , jei jos veikla užduotyje T , matuojant pagal įvertį P , gerėja patyrimo E atžvilgiu [2].

Mokymo aibė – tokia domenų aibė kuria apmokomas mašininio mokymo algoritmas.

Neįprastinis laivų eismas – (angl.: *abnormal marine traffic*) laivų ar laivo eismas, neatitinkantis tikėtiniams eismo modeliams ar šablonams.

Neprižiūrimasis mokymas – Neprižiūrimasis (angl. *unsupervised learning*), „mokymasis be pavyzdžių“, mokymosi algoritmas negauna mokymosi duomenų rezultato rinkinio ir turi pats nustatyti kaip klasifikuoti įvedimo duomenis.

Neuronas laimėtojas – neuronas, kuris vienos SOM iteracijos metų buvo nustatytas kaip artimiausias tos iteracijos įėjimo vektoriui.

OLE DB – tai aibė duomenų pasiekimo programinių sąsajų, kurios leidžia pasiekti daug skirtingų duomenų šaltinių. OLE DB leidžia programinei įranga pasiekti vienodu būdu skirtingų tipų ir šaltinių duomenų šaltinius.

Radaras –angliško termino RAdio Detection And Ranging sutrumpinimas. Tai objektų aptikimo sistema, kuri naudoja elektromagnetines radijo bangas, kad nustatytų atstumą iki objekto, aukštį, kryptį ir greitį.

ROC kreivė – (angl.: *receiver operating characteristic*) kreivė kuri iliustruoja binarinio klasifikatoriaus kokybinį našumą kai keičiamas skirstymo slenkstis. Kreivė sukuriama brėžiant klasifikavimo preciziškumo ir jautrumo reikšmes esant skirtingiems skirstymo slenkščio reikšmėms [2].

SOM – (angl.: *self-organizing map*) Kohoneno save-organizuojantis neuroninis tinklas.

SQL – (angl. *structured Query Language*) populiariausia iš šiuo metu naudojamų kalbų, skirtų aprašyti duomenis ir manipuluoti jais reliacinių duomenų bazių valdymo sistemose

Tiesioginis mašininis mokymas – (angl.: *online machine learning*) mašininio mokymo metodas, kuriame mokymo duomenys imami dalimis paėiliui ir yra naudojami atnaujinti prognozavimo modelį kiekviename žingsnyje.

Testavimo aibė – duomenų aibė, kuri nebuvo naudota mašininio algoritmo modelio apmokymui ar validavimui ir skirtą išmatuoti pagrindinius klasifikavimo rodiklius.

TP – (angl.: *true positive*) algoritmo klasifikavimo atvejis, kai laivo eismas klasifikuojamas teigiamai (kaip „neįprastas“) ir ekspertas klasifikavo kaip teigiamą („neįprastą“), t.y. kaip teisingai teigiamai nustatytą rezultatą (algoritmo ir eksperto klasifikavimas sutampa).

TN – (angl.: *true negative*) algoritmo klasifikavimo atvejis, kai laivo eismas klasifikuojamas neigiamai (kaip „įprastas“) ir ekspertas klasifikavo kaip neigiamą („įprastą“), t.y. kaip teisingai neigiamai nustatytą rezultatą (algoritmo ir eksperto klasifikavimas sutampa).

TSP – Keliaujančio pardavėjo problema. (Angliškai – Traveling salesman problem). Klasikinė pavyzdinė NP – hard sudėtingumo užduotis, ieškanti optimaliausio kelio tarp n kiekio miestų pardavėjui, kad apkeliautų visus, sugaišdamas mažiausiai laiko arba išleisdamas kelionei mažiausiai pinigų.

VHF – Very high frequency (VHF). Ryšių reguliavimo tarnybos nustatytas elektromagnetinio bangų diapazonas nuo 30Mhz iki 300Mhz.

VFIN – (virtualaus feromono intensyvumo santykinė reikšmė neurone) – santykinė reikšmė nurodanti koks santykis neuronų SOM tinkle turi konkrečią virtualaus feromono intensyvumo reikšmę.

TURINYS

PAVEIKSLĖLIŲ SĄRAŠAS.....	5
LENTELIŲ SĄRAŠAS	6
TERMINŲ IR SANTRUMPŲ ŽODYNĖLIS.....	7
ĮVADAS	10
1 MOKSLINĖS LITERATŪROS ANALIZĖ IR TOBULINTINŲ METODŲ PARINKIMAS	12
1.1 Automatinė identifikavimo sistema (AIS)	12
1.2 Metodų ir algoritmų apžvalga ir parinkimas	14
1.2.1 Galimų incidentų aptikimo sistemos ir algoritmai.....	14
1.2.2 Tobulintinų metodų ir algoritmų pasirinkimas	19
1.3 Parinktų metodų naudojamų sukurtame algoritme ir posistemyje apžvalga.....	20
1.3.1 SOM Kohoneno tinklo sandara	21
1.3.2 SOM tinklo apmokymas.....	22
1.3.3 Algoritmo binarinio klasifikavimo rodiklių nustatymas.	25
1.4 Apibendrinimas	27
2 DIDŽIAISIAIS DUOMENIMIS APSIMOKANČIO ALGORITMO IR POSISTEMIO KŪRIMAS	28
2.1 Algoritmo kūrimas.....	28
2.1.1 SOM ir virtualaus feromonų metodų apjungimas.....	28
2.1.2 Dviejų modifikuotų SOM modelių sintezė, naudojant tiesioginio mokymosi metodą.....	31
2.1.3 Atstumo tarp neuronų ir jėgimo vektorių skaičiavimas ne Euklido erdvėje.	34
2.1.4 SOM kaimynystės funkcijos modifikavimas.....	37
2.1.5 Neįprasto laivų eismo aptikimo modelio apmokymo algoritmas	38
2.1.6 Algoritmo laiko sudėtingumo skaičiavimas	41
2.2 Posistemio kūrimas	42
2.2.1 Posistemio integravimas į egzistuojančias laivų stebėjimo ir kontrolės sistemas	42
2.2.2 Neįprasto eismo aptikimo jūrų uoste posistemio architektūra	43
2.3 Sukurto prototipo architektūra ir sandara.....	47
3 PROTOTIPO VERIFIKAVIMAS IR ALGORITMO KLASIFIKAVIMO SĄVYBIŲ TYRIMAS ...	49
3.1 Verifikavimo ir tyrimo eiga.....	49
3.2 Tiriamų duomenų imtis.....	50
3.3 Įeinančių į algoritmą duomenų struktūra	51
3.4 Įprasto eismo modelio apmokymas dviejų laivų trajektorija.	51
3.5 Visa testavimo duomenų aibe apmokyto SOM tinklo vizualizavimas	54
3.6 Algoritmo klasifikavimo savybių tyrimas.....	55
IŠVADOS IR PASIŪLYMAI	58
PRIEDAI	Error! Bookmark not defined.

IVADAS

Jūrinis sektorius yra vienas iš labiausiai kuriančių veiklos galimybes tarptautinėje erdvėje. Vien jūrų transportu pasaulinėje prekyboje yra pervežama daugiau negu 90% visų krovinių [7]. Klaipėdos jūrų uostas pasižymi vis didėjančių krovinių ir laivų srautu. 2015 m. sausio–gruodžio mėn. Klaipėdos uosto krovos apyvarta augo 5,8% [8]. Šis sektorius išlieka vienas iš svarbiausių žmonių veiklos sričių. Dėl intensyvios veiklos ši sritis taip pat yra viena iš pavojingiausių pasaulyje. Nors šiai dienai navigacinė technologija yra labai išvystyta, laivai yra smarkiai didesni, greitesni, saugesni, o personalas profesionaliau paruoštas, nepavyksta išvengti pakankamai dažnų jūrinių incidentų, tokių kaip laivų sudužimas, užgrobimas [9, 7]. Stebėjimo ir kontrolės sistemos suteikia tik duomenų rinkimo, jų apjungimo ir paprastas automatinio informavimo funkcijas. Tačiau realus situacijos supratimas aktyvioje jūrų transporto eismo būsenoje vis dar yra procesas, reikalaujantis labai intensyvių žmogaus kognityvinių pastangų. Turint ribotus žmogiškuosius resursus didelių jūrų uostų generuojami jūrų transporto duomenys praktiškai neleidžia kognityviai suvokti realios situacijos [10].

Pastaraisiais metais mokslininkų bendruomenė aktyviai dirbo, kuriant algoritmus modeliuojančius įprastą laivų judėjimą jūrų uoste ar jo prieigose, kad optimizuotų žmogiškuosius resursus perkeliant dalį atliekamų užduočių į skaičiavimo mašinas. Deja, šiai dienai sukurti algoritmai gerai dirba tik su nedidelės apimties duomenimis. Aktyvėjant ir sudėtingėjant laivų eismui smarkiai didėja duomenų srautas, o esant prieinamai duomenų saugyklų kainai, visi duomenys yra kaupiami. Vienas iš pagrindinių AIS duomenų teikėjų „MarineTraffic.com“ AIS laivų eismo duomenis kaupia jau nuo 2009m. Tai apima 500 tūkst. įvairių laivų transportą pasauliniu mastu. Vienas judantis laivas siunčia AIS duomenis apie savo būseną ne rečiau, negu kas dvi minutes, tai „MarineTraffic.com“ yra sukaukęs apie $7,9 \cdot 10^{11}$ AIS įrašų, kas sudaro virš 1 PB duomenų. Kiekvieną valandą ši AIS sistema gauna 20GB naujų duomenų. Šiuo metu sukurti laivų neįprasto eismo aptikimo algoritmai paremti Bajeso, klasikinio SOM, potencinių laukų ir kitais metodais negali efektingai realiu ar artimai realiu laiku apdoroti tokio didelio duomenų kiekio. Kad būtų apmokintas neįprasto judėjimo modelis, būtinas visų duomenų apdorojimas (*angl. batch processing*), dažniausiai keliais ciklais. Tokių metodų parametru (svorių) konvergavimui į lokalų optimumą dažniausiai naudojamas gradientinio pakilimo (*angl.: gradient ascent*) algoritmas, kur vienam žingsniui atlikti skaičiuojami visi duomenys, o reikiamų atlikti žingsnių yra šimtai ar net tūkstančiai.

Kitas šiuo metu sukurtų algoritmų trūkumas yra tas, kad įprastam laivų eismo modelio apmokymui naudojama nepakankamai įeinančių duomenų, tokių kaip laivo tipas, ilgis, meteorologinės sąlygos (vėjas, bangavimas). Tokių duomenų nenaudojimas padaro modelį per paprastą ir atitinkamai netikslų, esant prastesnėms oro sąlygoms.

Mano bakalauro baigiamasis darbas buvo pirmasis bandymas sukurti laivų įprasto eismo modelio apmokymo algoritmą, kuris pasižymi mažu algoritmo laiko sudėtingumu (angl.: running complexity) ir tiesioginių realaus laiko mokymu (angl.: online learning), bet mažas įeinantys laivų eismo vektorių matmenų skaičius neleido sukurti pakankamai tikslaus modelio.

Nepaisant gausybės tyrimų, sritis nėra iki galo iširta ir reikalauja gilesnės analizės, ypač kuriant algoritmus didžiųjų duomenų masyvams (angl.: big data).

Darbo problema: Šiuo metu sukurti metodai ir algoritmai neišprastam jūrų laivų eismui aptikti nėra pritaikyti dirbti su didžiais duomenimis. Dėl šios priežasties modelio apmokymas vyksta neleistinai ilgai arba naudojamas mažesnis duomenų kiekis, kadangi krenta klasifikavimo preciziškumas. Šie trūkumai lemia, kad neišprasto aptikimo posistemis negali sumažinti laivų eismo kontrolės operatorių darbo krūvio.

Tikslas: Sukurti didžiais duomenimis apsimokantį algoritmą ir posistemį neišprastam laivų eismui aptikti jūrų uoste egzistuojančioms laivų stebėjimo ir kontrolės sistemoms, kurių klasifikavimo preciziškumas aukštesnis negu 88% ir algoritmo laiko sudėtingumas mažesnis negu $\Theta(n^6)$.

Uždaviniai:

1. Išanalizuoti mokslinę literatūrą esamų neišprasto laivų eismo aptikimo tema ir pasirinkti tobulintinus metodus bei algoritmus, kuriems reikalingi pakeitimai, pritaikant dirbti su laivų eismo didžiais duomenimis.
2. Sukurti didžiais duomenimis apsimokantį algoritmą ir posistemį neišprastam laivų eismui atpažinti jūrų uoste egzistuojančioms laivų stebėjimo ir kontrolės sistemoms.
3. Verifikuoti sukurtą prototipą ir iširti algoritmo klasifikavimo preciziškumą ir jautrumą, naudojant didžiųjų duomenų imtį iš Klaipėdos Valstybinio jūrų uosto stebėjimo ir kontrolės sistemos

Darbo rezultatų aprobavimas. Dalis darbo rezultatų pateikta recenzuojamuose mokslo leidiniuose:

J.Venskųs, M. Kurmis, A.Andziulis, Ž.Lukošius. Self-Learning Adaptive Algorithm For Maritime Traffic Abnormal Movement Detection Based On Virtual Pheromone Method. 2015 International Symposium on Performance Evaluation of Computer and Telecommunication Systems. IEEE Xplore. 26-29, 2015, Chicago, IL, USA.

J.Venskųs, M.Kurmis, D.Dzemydienė, A.Andziulis, T.Limba. Adaptyvaus galimų incidentų prevencijos apsimokančio algoritmo kūrimas jūrų transporto eismo uoste stebėjimo sistemai. 7th national marine science ant technologies conference "Jūros ir krantų tyrimai 2013". Klaipėda, 2013

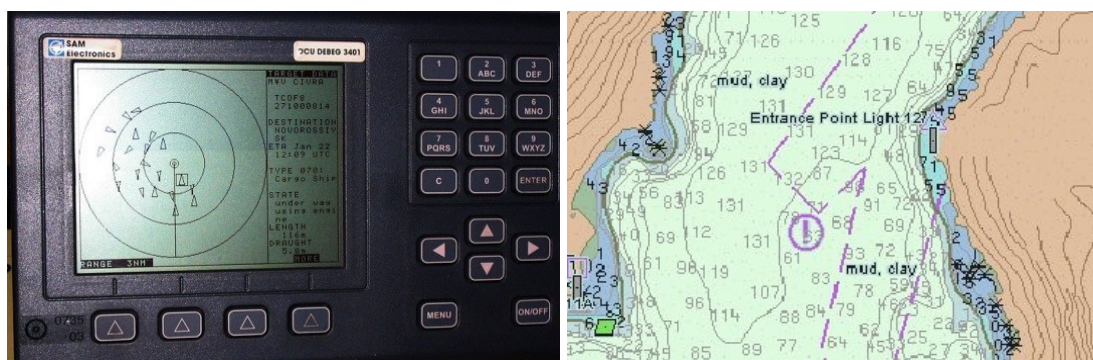
1 MOKSLINĖS LITERATŪROS ANALIZĖ IR TOBULINTINŲ METODŲ PARINKIMAS

Skyriuje analizuojama mokslinė literatūra esamų neįprasto laivų eismo aptikimo algoritmų tema ir pasirenkami tobulintini metodai, kuriems reikalingi pakeitimai, pritaikant dirbti su didžiaisiais laivų eismo duomenimis. Apžvelgiama esama technologija, galimų incidentų aptikimo sistemos ir algoritmai. Pagal nusistatytus kriterijus pasirenkami metodai ir algoritmai. Apžvelgiami pasirinkti metodai algoritmai, tinkami pritaikymui dirbti su didžiaisiais duomenimis.

1.1 Automatinė identifikavimo sistema (AIS)

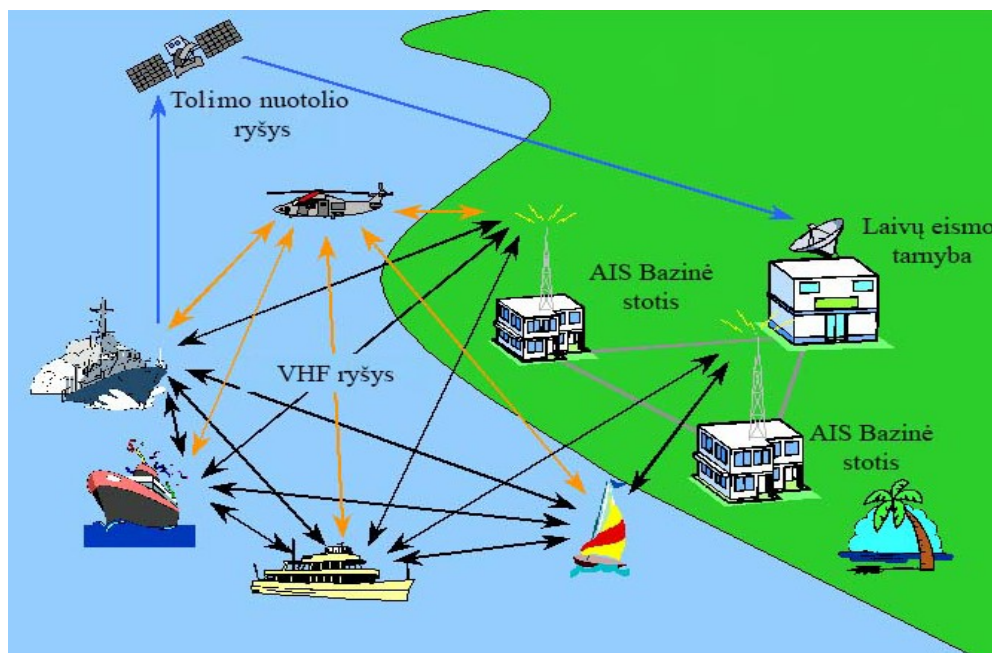
Automatinė identifikavimo sistema (toliau AIS) yra automatinė stebėjimo sistema naudojama jūrų laivuose ir Laivų eismo tarnybos, kad identifikuotų ir nustatytų laivų geografinę padėtį naudojant elektroninę duomenų perdavimo sistemą per kitus laivus, AIS bazines stotis ar žemės palydovus. AIS informacija papildo jūros radarus, kurie yra pagrindinė stebėjimo ir kontrolės informacinės sistemos dalis.

AIS įranga pateikia informaciją tokia kaip: unikali identifikacija, geografinė pozicija, kursas ir greitis. Ši informacija gali būti pavaizduota įrenginio ekrane (1 pav.) arba elektroniniame jūrlapyje (1 pav.). Pagrindinė AIS paskirtis - padėti laivo pamainos viršininkui ir suteikti galimybes jūrų tarnyboms stebėti ir sekti laivo eismą. Į AIS integruota standartizuotas VHF ultra-trumpųjų dažnių radijo ryšio sistema kartu su pozicionavimo sistema GPS arba LORAN-C. Taip pat gali būti integruoti kiti sensoriai, tokie, kaip giroskopas ar apskukų indikatorius. Laivų, kuriuose įrengta AIS siųstuvai, poziciją gali stebėti AIS bazinės stotys, išdėliotos pagal kranto liniją (2 pav.). Kai laivas yra bazinių stočių nepasiekiamumo zonoje, signalas perduodamas per dirbtinius žemės palydovus, kuriuose įmontuoti AIS imtuvai.



1 pav. AIS įrenginio priekinis vaizdas ir atvaizdavimas elektroniniame jūrlapyje.

Tarptautinė jūrų organizacijos (IMO) Tarptautinė gyvybės saugumo jūroje konvencija reikalauja, kad AIS sistema būtų įdiegta kiekviename, didesnio, negu 300 t bendrojo tonažo, jūrų laive ir kiekviename keleiviniame laive, nepriklausomai nuo dydžio [11].



2 pav. AIS sistemos ryšio perdavimo kanalai

AIS sistema siunčia į stebėjimo ir kontrolės sistemą laivų eismo duomenis, kurių sandara pateikta 1-oje lentelėje.

1 lentelė. Laivų eismo duomenų struktūra

<i>Eilė numeris</i>	<i>Duomenų lauko pavadinimas</i>	<i>Lauko paskirtis</i>	<i>Tipas</i>
1	Id	Įrašo identifikacinis numeris	Long, unique, primary key
2	Date	Įrašo data	date time
3	MMSI	„Maritime Mobile Service Identity“ unikalus laivo identifikacinis numeris	varchar
4	Status	Navigacinė būseną. (pvz. Prisišvartavęs, kelyje, ..)	varchar
5	rate Turn	Vairo posūkio kampas	Integer
6	speed	Laivo greitis mazgais	Integer
7	Course	Laivo kursas pagal šiaurės polių	Integer
8	Longitude	Geografinė Ilguma	long
9	Latitude	Geografinė Platuma	long
10	Accuracy	Pozicijos tikslumas	float
11	source	Duomens šaltinis(pvz. AIS, Radarų sistema)	varchar

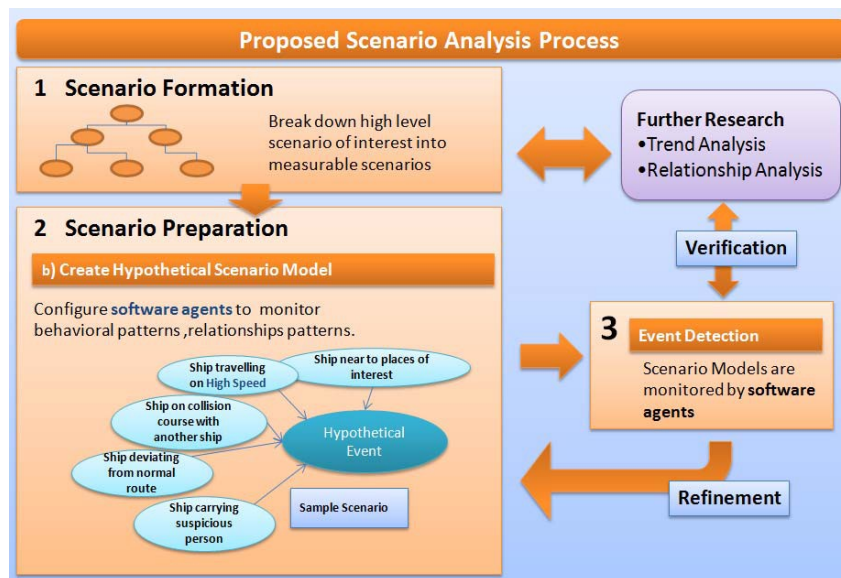
1.2 Metodų ir algoritmų apžvalga ir parinkimas

1.2.1 Galimų incidentų aptikimo sistemos ir algoritmai

Jūrų uosto informacinės sistemos paprastai apima duomenų surinkimo, informacijos sintezės ir bazinio duomenų pateikimo operatoriui funkcijas. Atpažinti galimus incidentus bei nustatyti nukrypimą nuo įprasto režimo išlieka labai intensyvaus žmogaus darbo reikalaujančia užduotimi. Esant ribotiems žmogiškiesiems resursams, sudėtinga ir neefektyvu rankiniu būdu apdoroti didžiulius duomenų kiekius visą parą veikiančiame jūrų uoste. Efektyvi žinių gavybos sistema uosto stebėjimui turi apjungti žmogaus ir dirbtinio intelekto galimybes, transformuojant surinktus duomenis į žiniomis grindžiamą sprendimų paramą operatoriams, kad pasiektas maksimalus efektyvumas ir tikslumas, nustatant atitinkamas situacijas jūrų uoste [12] [13]. Šią uosto saugumo ir efektyvumo prasme aktualią sritį aktyviai tyrinėja pasaulinės mokslininkų bendruomenės.

O. Tan ir S. Tng siūlo integruotą jūrinę samprotavimo ir stebėjimo sistemą, paremtą Bajeso samprotavimu, esybių tinklo analize ir judėjimo modelių šablonų analize [14]. Šiame darbe autoriai aktyviai rašo apie galimų incidentų analizę, stebint laivų plaukimo trajektoriją, ir, esant nukrypimams nuo pagal iš anksto numatytas taisykles, signalizuoti apie tai operatoriui. O. Tan ir S. Tng siūlo naudoti laivų eismo scenarijaus analizės procesą (3 pav.), kurio pirmajame žingsnyje aukštesnio lygmens scenarijus išskirstomas į koncentruoto dėmesio žemesnius scenarijus, ten eismo būsenos modelius analizuoja programiniai agentai, kurie nagrinėja pagal tris būsenų indikatorius:

- Elgsenos indikatoriai (pvz. nukrypimai nuo žinomų laivų kelių ar farvaterio, patekimas į draudžiamas teritorijas, neleidžiamas laivo greitis)
- Sąryšio indikatoriai (pvz. sąryšis tarp dviejų būsenų padidina pavojaus lygmenį)
- Aplinkos charakteristikų ir būsenų savybių indikatoriai (pvz. paros laikas, laivo klasė)



3 pav. Scenarijaus analizės procesas [14]

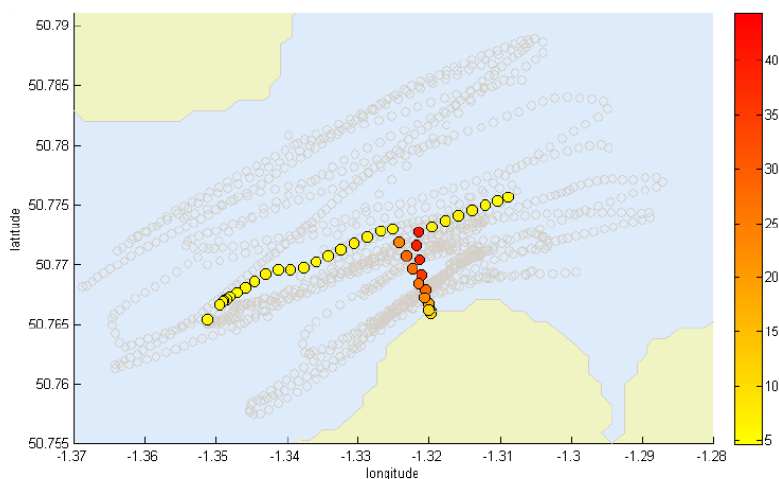
M. Riveiro ir G. Falkman analizuoja interaktyvų normalaus elgesio modelių ir ekspertinių taisyklių vizualizavimą laivybos anomalijų aptikimui. Autoriai teigia, kad tai būtinas elementas tokių modelių supratimui ir ekspertinių duomenų įterpimui. Taip pat tikima, kad duomenų tyrybos vizualizavimas gali padidinti vartotojų pasitikėjimą sistema dėl didesnio sistemos skaidrumo [15]. Autoriai savo straipsnyje naudoja „anomalijos“ koncepcijos sąvoką. Ji naudojama kompiuterinio tinklo apsaugoje, vaizdo apsaugoje, žmonių veiksmų stebėjime ir gedimų diagnostikoje ir t.t. Autoriai, kad būtų aptikta anomali laivo elgsena, naudoja mišrų būdą, kombinuodami duomenimis pagrįstą metodą su žiniomis grįstu aptikimo modulių [15]. Duomenimis pagrįstas metodas naudoja „save organizuojančius Kohoneno žemėlapius“ (SOM) ir „Gauso suminius modelius“ (GMM), kad būtų sukurtas laivo normalios elgsenos modelis. Normalios elgsenos modelis - tai tikimybių tankio funkcija, sukurta iš sumuotų svorinių „Gauso daugiamacių skirstinių“. Ši funkcija išreikšta formule:

$$P(x_1, \dots, x_n) = \sum_{i=1}^D \sum_{j=1}^D p_{ij} P_{ij}(x_1, \dots, x_n) \quad (1)$$

Individualūs skirstiniai P_{ij} šiuo atveju atitinka modelių vektorius, kurie gaunami iš Kohoneno žemėlapių. Kiekvienas modelio vektorius charakterizuojamas daugiamacių Gauso tikimybių skirstinio funkcija (praktikoje $n=5$ analizuojant kinetinius elgsenos modelius).

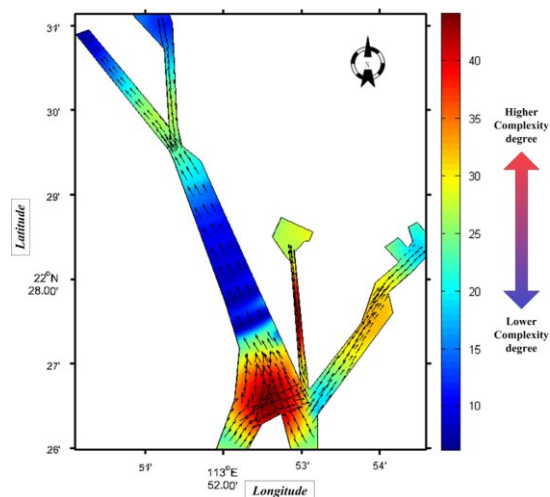
K. Kowalska ir L. Peel tyrinėja jūrų laivyno anomalijų aptikimą, naudojant Gauso proceso aktyvų mokymą, apjungtą su aktyviu mokymusi. Šis metodas naudojamas automatinio identifikavimo sistemos duomenų rinkinių sumažinimui [16]. Savo straipsnyje autoriai analizuoja laivų eismo duomenis, gautus iš trijų sistemų: Radarų, AIS ir GPS sekimo įrenginių. Autoriai iš stebėjimo sistemų gautus duomenis skirsto pagal skirtingus laivų tipus, kadangi skirtingi laivų tipai turi skirtingą

„normalų“ eismo modelį. „Normalus“ modelis mokomas pagal gautas laivo koordinatas, kryptį ir greitį. Iliustracijoje (4 pav.) pilka spalva pažymėti įprastos („normalios“) trajektorijos koordinatinių taškų. Autoriai imitavo laivo trajektoriją, kokia susidaro žmonių kontrabandos atveju (nelegalus žmonių pervežimas per valstybinę sieną), kai nuo kranto nelegaliai paimami žmonės. Ši „anomalė“, neatitinkanti įprasto modelio, trajektorija iliustracijoje (4 pav.) pažymėta geltona ir raudona spalvomis. Pagrindinis siūlomo metodo trūkumas – negalėjimas dirbti realiu laiku [16].



4 pav. Anomalijų aptikimas laivų eismo trajektorijose [16]

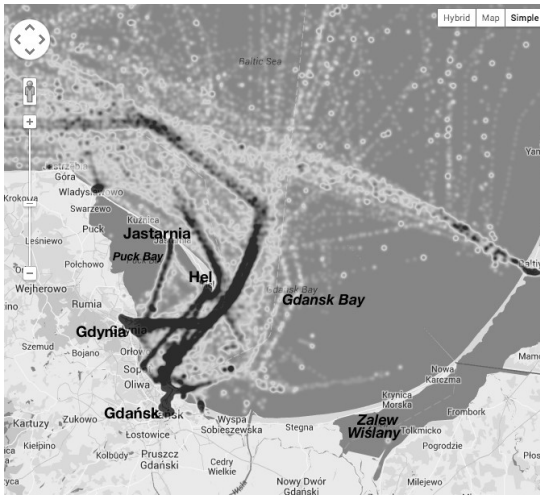
Yuanqiao Wen, Yamin Huang ir kiti savo straipsnyje [9] akcentuoja, kad išaugęs greitaeigių, labai didelio tonazų laivų skaičius padarė laivybos eismą žymiai sudėtingesniu. Eismo būsenas ir situacijas tapo žymiai sudėtingiau valdyti. Autoriai laivų eismo elementą modeliuoja, gretindami laivo eigos charakteristikas, tokias, kaip reliatyvus atstumas, reliatyvus greitis ir susikertančios trajektorijos. Autoriai didelį dėmesį kreipia į laivų eismo greičio vektoriaus tankį, kaip eismo sudėtingumą nusakantį pagrindinį rodiklį. Paveikslėlyje (5 pav.) pateiktas geografinis laivų eismo sudėtingumo pasiskirstymas Šengeno vakarų jūros kanale. Raudona spalva pažymėtas aukšto sudėtingumo eismas. Autoriai įrodo, kad didesnė laivų incidentų rizikos tikimybė susidaro būtent šio tipo eismo zonoje.



5 pav. Geografinis laivų eismo sudėtingumo pasiskirstymas Šenčeno jūros kanale. [9]

Ewa Osekowska ir Stefan Axelssonas straipsnyje [17] modeliuoja laivo judėjimą vandeni, naudojant lauko potencialų metodą (angl.: potential fields). Jų pagalba autoriai modeliuoja laivų navigavimo ir valdymo procesą stebint laivų elgseną virš vandens. Sukurti modeliai yra naudojami apibūdinti eismo savybių pagrindinius parametrus skirtus jūrinės situacijos supratimui (angl. maritime situational awareness) ir įspėti apie galimas pavojingas eismo elgsenas (anomalijos aptikimas). Autoriai sukūrė šiuo metodo pagrindu neįprasto, anomalinio eismo aptikimo sistemą „Strand“, kurios branduolį sudaro lauko potencialų metodas, pritaikytas AIS duomenų masyvui.

Paveikslėlyje (6 pav. a dalis) pavaizduota pagrindinė įprastinio laivo eismo elgsenos modelio idėja. Tai - Lenkijos šiaurinės pakrantės Gdansko įlankos žemėlapis. Žemėlapyje pilkos spalvos paletė (nuo baltos iki juodos spalvos) pavaizduoti lauko potencialai. Šiuo atveju „laukas“ reiškia visų laivų eismą be specifinio parametro (kurso, greičio ar paros laiko). Iš karto aiškiai matosi išskirtos didesnio intensyvumo vietos, tokios, kaip juostai ar įprastai naudojami farvateriai ir, priešingai, didžioji dalis žemėlapis yra be žymėjimo, kas reiškia eismo nebuvimą. Paveikslo (6 pav.) b dalyje pavaizduotas potencialų laukas tik laivams judantiems šiaurės kryptimi.



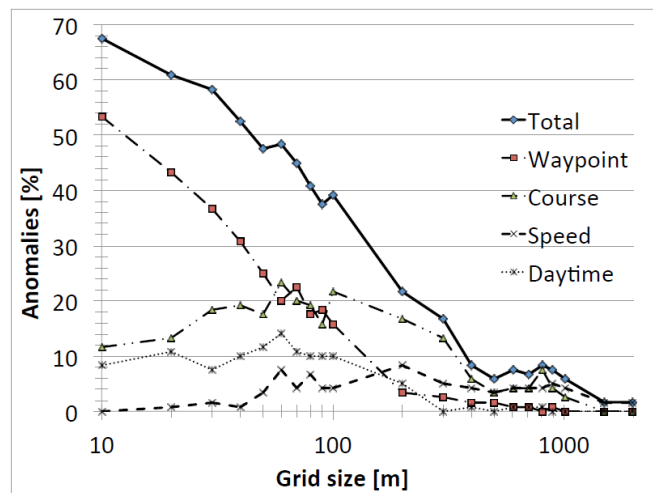
a)



b)

6 pav. Gdanskos įlankos laivų eismo lauko potencialų vizualizavimas [17].

Naudojant šiuos laukus autoriai sugebėjo į modelį sudėti greičio vektorius pagal žemėlapių sektorių (angl. grid) ir analizavo anomalijų (nukrypimai nuo įprasto modelio) priklausomybę nuo analizuojamo sektoriaus dydžio. Jų pateiktame grafike (7 pav.) matosi, kad didėjant langeliui modelis tampa sudėtingesnis ir anomalijų smarkiai sumažėja.



7 pav. Anomalijų aptikimas, priklausomai nuo sektoriaus dydžio [17]

Šis darbas įdomus tuo kad autoriai sujungė geografinę laivo padėtį, jo greičio vektorių ir paros laiką, panaudodami tai neįprastam, anomaliam eismui aptikti. Autoriai nepateikia jų sukurto algoritmo klasifikavimo pagrindinių rodiklių, kaip tikslumas ir priklausomybė nuo apmokymo duomenų.

1.2.2 Tobulintinių metodų ir algoritmų pasirinkimas

Prieš pasirenkant algoritmus ir metodus buvo nustatyti atrankos kriterijai ir jų svoriai. Kriterijų sąrašas ir jų aprašymas pateiktas lentelėje (2 lentelė). Kiekvienam kriterijui priskiriamas jo svarbumo ir įtakos į pasirinkimą svoris nuo 1 iki 3, kur 1 yra mažiau svarbus, 2 – vidutiniškai ir 3 – labai svarbus. Kiekvienas vertinamas metodas ar algoritmas tikrinamas šių kriterijų atitikimui 5 balų skalėje, kur 1- visiškai neatitinka, 2-mažai atitinka, 3- vidutiniškai atitinka, 4-labai atitinka, 5-labiausiai atitinka. Šie balai surašomi į lentelę prie kiekvieno vertinamo metodo ar algoritmo ir paskaičiuojamas suminis svorinis balas pagal formulę:

$$SSB = \sum_{i=1}^k S_i B_i, \quad (2)$$

čia i – kriterijaus numeris, k – viso kriterijų, S_i – Kriterijaus svoris, B_i – algoritmo ar metodo atitikimo vertinimo balas.

2 lentelė. Algoritmų ir metodų pasirinkimo kriterijai

Eil. Nr.	Kriterijaus pavadinimas	Vertinimo aprašymas	Kriterijaus svoris
1	Neprižiūrimasis mokymas	Vertinamas metodo ar algoritmo mokymasis iš anksto, neturint informacijos apie laivų eismo priskyrimo neįprasto eismo klasei, t.y. neprižiūrint mokymosi (<i>angl. unsupervised learning</i>)	2
2	Klasifikavimo preciziškumas ir specifiškumas	Vertinamas metodo ar algoritmo klasifikavimo preciziškumas ir specifiškumas, priklausomai, kokį pavyko pasiekti straipsnių autoriams.	1
3	Algoritmo laiko sudėtingumas	Vertinamas algoritmo ar metodo laiko sudėtingumas, kiek laiko jam reikėtų išspręsti mokymosi ar klasifikavimo uždavinį, priklausomai nuo įeinančių duomenų kiekio. Pateikiamas asimptotinis žymėjimas.	1
4	Mokymasis daugiamačiais vektoriais	Vertinama algoritmo ar metodo galimybės apmokyti modelį daugiamačiais įeinančių duomenų vektoriais.	2
5	Veikimas su ne Euklidinės erdvės vektoriais	Vertinama algoritmo ar metodo galimybė apsimokyti ne Euklidinės erdvės vektoriais arba galima modifikuoti algoritmą darbui šioje erdvėje.	1
6	Tiesioginis mokymasis didžiais duomenimis (<i>angl.: online learning</i>) arba galimybė adaptuoti jam	Vertinama galimybė realizuoti algoritme ar metode tiesioginio mokymosi didžiais duomenimis metodus, tokius kaip stochastinis gradientinis pakilimas, inkrementinis SGD ir kiti.	3

Algoritmų ir metodų vertinimo lentelėje (3 lentelė) pateikti metodai ir jų vertinimas atlikus pirminę atranką. Pirminė atranka buvo atliekama, analizuojant mokslinę literatūrą neįprasto laivų eismo aptikimo tema ir vertinant algoritmo ar metodo mokymąsi neprižiūrint (*angl. unsupervised learning*) ir panaudojimą neįprastam laivų eismui aptikti. Kiekvienas metodas vertinamas konkrečia kriterijaus atitikimui balu, kuris surašomas į lentelę. Ir paskaičiuojamas jo SSB (suminis svorinis balas). Pasirenkami metodai, turintys didžiausią balo vertę.

3 lentelė. Algoritmų ir metodų vertinimo lentelė

Eil. Nr.	Algoritmas ar metodas, panaudotas straipsnyje neįprastam laivų eismui aptikti	Algoritmo ar metodo atitikimas kriterijui balas (nuo 1 iki 5), kriterijus žymimas numeriu pagal lentelę nr. 2						Suminis svorinis balas
		1	2	3	4	5	6	
1	Bajeso metodas (angl.: Bayes method) [14]	5	2	2	5	3	1	30
2	„Gauso suminis modelis“ (GMM) [15]	5	2	2	5	3	1	30
3	Kohoneno SOM žemėlapis [18] [15]	5	5	2	5	4	4	43
4	Gauso procesas (angl.: Gaussian process) [16]	5	4	3	4	3	2	34
5	Eismo sudėtingumo metodas (angl.: Traffic complexity method) [9]	5	2	4	2	2	2	28
6	Lauko potencialų metodas (angl.: potential fields) [17]	5	4	4	2	2	4	36
7	Virtualaus feromono metodas [19]	5	4	5	3	3	5	43

Atlikus vertinimą nustatyta, kad didžiausius balus gavo Kohoneno SOM žemėlapio metodas ir Virtualaus feromono metodas. Jie pasirinkti tolimesniam tobulumui, modifikavimui ir algoritmo, apsimokančio didžiausiais duomenimis neįprastam laivų eismui aptikti jūrų uoste.

1.3 Parinktų metodų naudojamų sukurtame algoritme ir sistemyje apžvalga

Šiame skyriuje apžvelgiami algoritmai ir metodai naudojami kuriamame algoritme ir sistemyje. Analizuojama jų matematinės ir loginės savybės, išskiriant pagrindinius naudotinus komponentus.

1.3.1 SOM Kohoneno tinklo sandara

Sukurtaime algoritme naudojamas SOM savi-organizuojantis neuroninis tinklas (*angl.: Self-organizing map*). Šių neuroninių tinklų pavadinimas atsirado, kad SOM, naudodamas mokymo aibę, pats save organizuoja. Vienas iš pagrindinių tinklo tikslų yra išlaikyti duomenų topologiją. SOM tinklai gali būti naudojami, siekiant vizualiai pateikti duomenų klasterius ir ieškant daugiamačių duomenų projekcijų į mažesnio skaičiaus matmenų erdvę, paprastai į plokštumą [18, 4]. SOM neuroninis tinklas yra neuronų, įprastai išdėstytų dvimačio tinklelio, kitaip vadinamu žemėlapiu, mazguose, masyvas. Kuriamame algoritme SOM tinklas aprašomas:

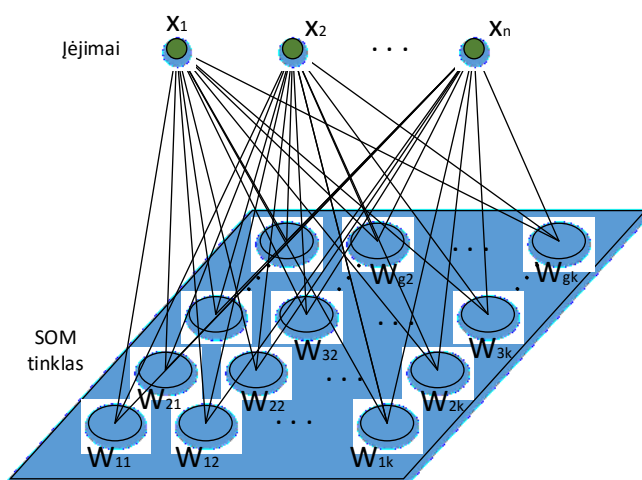
$$M = \{W_{ij} \mid i \in (1, g), \quad j \in (1, k), \quad i, j, k, g \in \mathbb{N}\} \quad (3)$$

$$W_{ij} = \{(W_{ij1}, W_{ij2}, \dots, W_{ijn}) \mid n \in \mathbb{N}\} \quad (4)$$

čia W_{ij} SOM tinklo neuronų matricos elementas, žymintis tam tikrų neuronų įėjimo svorius. i ir j , atitinkamai reiškia neuronų poziciją matricoje. Kiekvienas matricos elementas sudarytas iš vektoriaus, kurio elementai yra $W_{ij1}, W_{ij2}, \dots, W_{ijn}$, n – analizuojamos aibės matmenų skaičius (savybių skaičius). k, g – SOM tinklo matmenys.

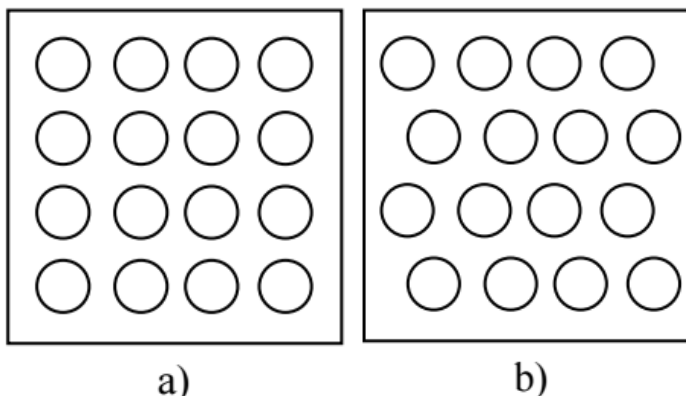
Neuroninio tinklo įėjimą sudaro vektorius $\bar{X} = (x_1, x_2, \dots, x_n)$, kur kiekvienas elementas sujungtas per atitinkamo matmens (savybės) jungtį su kiekvienu tinklo neuronu.

Neuroninio tinklo loginė struktūra pavaizduota 8 pav. Kiekvienas tinklo neuronas sujungtas su kiekviena įėjimo komponente. Priešingai, negu tiesioginio sklidimo neuronai, SOM neuronai neturi nulinio įėjimo.



8 pav. Dvimačio SOM tinklo loginė schema

Kuriamame algoritme SOM tinklas atliks duomenų savybių žemėlapio (*angl.: Feature map*) sukūrimui. Projektuojamame algoritme bus naudojamas stačiakampės struktūros SOM tinklas (9 pav. a)



9 pav. SOM tinklo struktūra: a) stačiakampė, b) šešiakampė

1.3.2 SOM tinklo apmokymas

SOM tinklo apmokymas priklauso mašininio mokymo algoritmų kategorijai, kai iš anksto nežinomos išėjimo reikšmės, taip vadinamas apmokymas be mokytojo (*angl.: Unsupervised learning*). SOM Tinklas bus apmokytas n -mačiais vektoriais $\bar{X}_1, \bar{X}_2, \dots, \bar{X}_m$, kuriuos atitiks turimos laivų eismo AIS duomenys, prieš tai praėjus duomenų filtravimo procesą. Po apmokymo kiekvienas šios imties vektorius bus susietas su vienu tinklo neuronu, kuris taip pat yra n -matis vektorius, t.y. kiekvienas laivų eismo duomuo po apmokymo bus priskirtas ir klasifikuotas pagal konkretų viena SOM tinklo neuroną. Vektorių nusakantis neuronas bus žymimas $W_{ij} = \{(W_{ij1}, W_{ij2}, \dots, W_{ijn}) \mid n \in \mathbb{N}\} \in \mathbb{R}^n$.

Įprastai mokymo pradžioje neuronų (vektorių) W_{ij} elementų pradinės reikšmės nustatomos atsitiktinai. Šiame algoritme bus naudojamas euristinis metodas pradinių reikšmių nustatymui pagal turimus duomenis.

Neuroniniam tinklui daug kartų pateikiami apmokymo duomenys. Šio algoritmo atveju apmokymo duomenis sudaro AIS duomenys apie laivų eismą ir meteorologinės sąlygos pasirinktame geografiniame regione. Pateikiami duomenys nusakomi n -mačiais vektoriais $\bar{X}_1, \bar{X}_2, \dots, \bar{X}_m$

- Kiekviename mokymo žingsnyje (iteracijoje) į tinklą pateikiamas vienas mokymo aibės vektorius $\bar{X}_k \in \{\bar{X}_1, \bar{X}_2, \dots, \bar{X}_m\}$.

- Vektorius \bar{X}_k palyginamas su visais neuronais W_{ij} skaičiuojant atstumą. Dažniausiai naudojamas Euklido atstumas ($\|\bar{X}_k - W_{ij}\|$) tarp šio vektoriaus ir \bar{X}_k ir kiekvieno SOM tinklo neurono vektoriumi W_{ij} .
- Randama, iki kurio neurono $W_c \in \{W_{ij}, i = 1, \dots, g, j = 1, \dots, k\}$ atstumas yra mažiausias. Rastas neuronas W_c vadinamas neuronu (vektoriumi) nugalėtoju (angl.: Winner neuron). Pažymime i_c eilutę, o j_c stulpelį nusakanti neurono vietą W_c vietą tinklo matricoje, čia c yra skaičių i_c ir j_c pora ir randama pagal formulę [4, 18]:

$$c = \underset{i,j}{\operatorname{argmin}}\{\|\bar{X}_k - W_{ij}\|\}, \quad \|\bar{X}_k - W_c\| = \min_{i,j}\{\|\bar{X}_k - W_{ij}\|\} \quad (5)$$

- Po to keičiamos visų tinklo neuronų komponentės naudojantis iteracine formule [4, 18]:

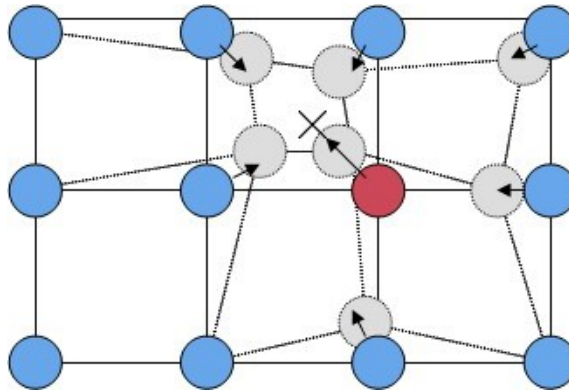
$$W_{ij}(t+1) = W_{ij}(t) + h_{ij}^c(t)(\bar{X}_k - \bar{W}_{ij}) \quad (6)$$

Čia t yra iteracijos numeris, $h_{ij}^c(t)$ – kaimynystės funkcija [4, 18]:

$$h_{ij}^c(t) = h_{ij}^c(t) \left(\|\hat{R}_c - \hat{R}_{ij}\|, t \right), \quad (7)$$

čia \hat{R}_c ir \hat{R}_{ij} – dviejų matų vektoriai, kurie yra W_c ir W_{ij} indeksai (eilutės ir stulpeliai numerių), nurodančių vektoriaus \bar{X}_k neurono laimėtojo W_c ir perskaičiuojamojo neurono W_{ij} padėtį SOM tinkle [4, 18]. Procesui konverguoti būtina, kad $h_{ij}^c(t) \rightarrow 0$, kai $t \rightarrow \infty$. Dydis $\|\hat{R}_c - \hat{R}_{ij}\|$ yra atstumo funkcija tarp vektorių \hat{R}_c ir \hat{R}_{ij} . Jam didėjant, funkcijos $h_{ij}^c(t)$ reikšmė artėja prie nulio ($h_{ij}^c(t) \rightarrow 0$). Atstumui taip \hat{R}_c ir \hat{R}_{ij} galima naudoti Euklido, Gauso, Meksikietiškos kepurės (angl.: Mexican hat), Prancūziškos kepurės (angl.: French hat), ar kitas atstumo funkcijas. Kuriamame algoritme panaudota Meksikietiškos kepurės atstumo funkcija skaičiuoti neuronų atstumui taip neuronų tinkle.

Vykdamas tinklo apmokymą (10 pav.) vienos iteracijos metu yra perskaičiuojami visi neuronai pagal duomens vektorių \bar{X}_k kiekybiškai priklausomai pagal funkcija $h_{ij}^c(t)$ nuo atstumo iki neurono laimėtojo. Paveikslėlyje (10 pav.) neuronas nugalėtojas W_c pažymėtas raudona spalva, kuris perstumiamas link duomens vektoriaus labiausiai. Kiti neuronų vektoriai (10 pav., pažymėti mėlyna spalva) perstumiami, priklausomai nuo atstumo iki laimėtojo.



10 pav. Neuronų vektorių perskaičiavimas pagal atstumą iki neurono laimėtojo.

Apmokant SOM tinklą vienos iteracijos metu pateikiamas vienas vektorius. Geresniam tinklo apmokymui į tinklą dažnai tas pats vektorius pateikiamas keletą kartų. Dažniausiai naudojami trys būdai [4, 18]

- įėjimo imties vektoriai pateikiami tinklui iš eilės po vieną cikle;
- į SOM paduodamas atsitiktinai įėjimo imties vektorius, po to vėl atsitiktinai paduodamas kitas ir t. t.
- vektoriai pateikiami atsitiktine tvarka, t. y. vektoriai sumaišomi ir tada vienas po kito pateikiami į tinklą; kai visi jau pateikti, permaišomi ir vėl pateikiami į tinklą ir t. t.;

Kuriamame algoritme pasirenkamas trečias būdas. Mokymo proceso dalis, kurios metu visi mokymo aibės vektoriai visi $\bar{X}_1, \bar{X}_2, \dots, \bar{X}_m$ (m – vektorių skaičius) po vieną kartą pateikiami į tinklą nuosekliai arba atsitiktine tvarka vadinasi EPOCHA [4, 18].

Viena iš galimų kaimynystės funkcijos h_{ij}^c išraiškų yra tokia [4, 18]:

$$h_{ij}^c(t) = \frac{\sigma}{\sigma \eta_{ij}^c + 1}, \quad (8)$$

Čia σ – tai funkcija, nusakanti neuroninio tinklo kaimynystės erdvę (angl.: space of the neighbourhood). Pradžioje ji įtraukia beveik visą SOM neuroninį tinklo žemėlapi. Procesui konverguoti būtina, kad $h_{ij}^c(t) \rightarrow 0$, kai $t \rightarrow \infty$. Šioje kaimynystės funkcijoje už šį konvergavimą atsakinga σ funkcija. Ją išreiškta formule:

$$\sigma = \max(k, g) \cdot \frac{E + 1 - \hat{E}}{E}, \quad (9)$$

Čia E - prieš SOM apmokymą pasirinktas viso mokymo epochų skaičius, \hat{E} – vykdomos epochos eilės numeris, k , g – neuroninio tinklo plotis, išreikštas neuronų skaičiumi eilutėje arba stulpelyje.

1.3.3 Algoritmo binarinio klasifikavimo rodiklių nustatymas.

Pagrindinė algoritmo užduotis klasifikuoti laivų judėjimą kaip „įprastą“ arba kaip galimą incidentą t.y. „neįprastą“. Kadangi klasifikuojama į dvi grupes, klasifikavimo efektyvumui vertinti galima naudoti binarinio klasifikatoriaus rodiklius.

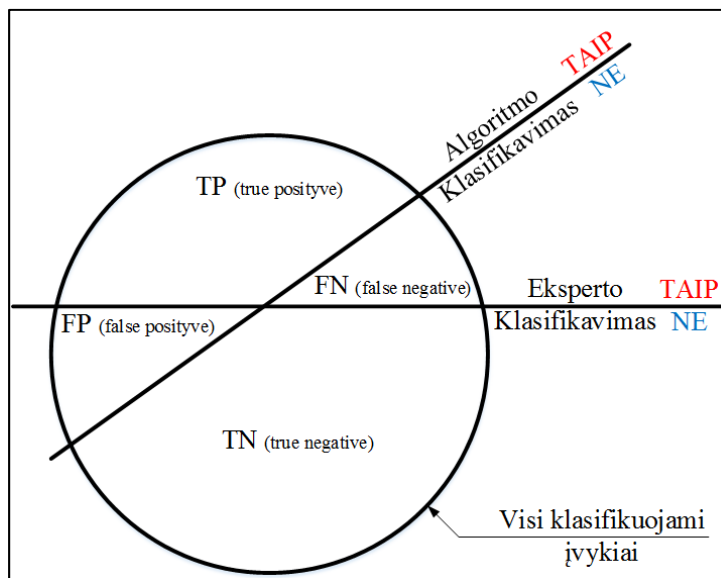
Algoritmo klasifikavimo našumui išmatuoti naudojami klasifikavimo rodikliai, atsižvelgiant į eksperto klasifikavimo duomenis.

- Jei algoritmas klasifikavo jūrų laivo eismą kaip „neįprastą“ ir ekspertas klasifikavo kaip „neįprastą“, tada žymime kelią kaip klasifikuotą TP (*angl.: True positive*), t.y. kaip teisingai teigiamai nustatytą rezultatą (algoritmo ir eksperto klasifikavimas sutampa).
- Jei algoritmas klasifikavo laivo kelią kaip „neįprastą“, o ekspertas klasifikavo kaip „įprastą“, tada žymime kelią kaip klasifikuotą FP (*angl.: False positive*), t.y. kaip klaidingai teigiamai nustatytą rezultatą (algoritmo ir eksperto klasifikavimas nesutampa).
- Jei algoritmas klasifikavo laivo kelią kaip „įprastą“ ir ekspertas klasifikavo kaip „įprastą“, tada žymime kelią kaip klasifikuotą TN (*angl.: True negative*), t.y. kaip teisingai neigiamai nustatytą rezultatą (algoritmo ir eksperto klasifikavimas sutampa).
- Jei algoritmas klasifikavo laivo kelią kaip „įprastą“, ekspertas klasifikavo kaip „neįprastą“, tada žymime kelią kaip klasifikuotą FN (*angl.: False negative*), t.y. kaip klaidingai neigiamai nustatytą rezultatą (algoritmo ir eksperto klasifikavimas nesutampa).

Paveikslėlyje (11 pav.) pavaizduoti binarinio klasifikavimo rodikliai. Apskritimas simbolizuoja visus klasifikuojamus laivo kelius. Eksperto klasifikavimas šiuos kelius suskirsto į dvi grupes: klasifikuoti kaip teigiami („neįprastas“ laivo kelias) ir klasifikuoti kaip neigiami („įprastas“ laivo kelias). Tiriamų algoritmų klasifikavimas taip pat skaido kelius į dvi kitas grupes: klasifikuoti kaip teigiami („neįprastas“ laivo kelias) ir klasifikuoti kaip neigiami („įprastas“ laivo kelias).

Kadangi įprastai algoritmo ir ekspertų klasifikavimai nesutampa, susidaro keturios klasifikavimo rodiklių grupės(11 pav.):

- **TP (angl.: true positive)** – kai sutampa algoritmo ir eksperto klasifikavimas kaip „teigiamas“,
- **FP (angl.: false positive)** – kai nesutampa algoritmo klasifikavimas „teigiamai“ su eksperto klasifikavimu,
- **TN (angl.: true negative)** – kai sutampa algoritmo ir eksperto klasifikavimas kaip „neigiamas“,
- **FN (angl.: false negative)** – kai nesutampa algoritmo klasifikavimas „neigiamai“ su eksperto klasifikavimu.



11 pav. Binarinio klasifikatoriaus rodikliai

Klasifikavimo rodikliai TP, FP, TN, FN priklauso kiekybinių rodiklių grupei. Tiksliau ir aiškiau algoritmo klasifikavimo savybes nusako santykiniai klasifikavimo rodikliai:

- **preciziškumas** – (angl.: *precision or true positive rate PPV*) kuris išreikštas formule:

$$PPV = \frac{TP}{TP + FP} \cdot 100\%, \quad (10)$$

- **jautrumas** – (angl.: *sensitivity or true positive rate (TPR)*), kuris išreikštas formule:

$$TPR = \frac{TP}{TP + FN} \cdot 100\%, \quad (11)$$

- **tikslumas** – (angl.: *accuracy (ACC)*), kuris išreikštas formule:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \cdot 100\%, \quad (12)$$

čia ACC – klasifikavimo tikslumas, TPR – klasifikavimo jautrumas, SPC – klasifikavimo specifiškumas, TP – teisingai klasifikuotas teigiamas rezultatas, FP – klaidingai klasifikuotas teigiamas rezultatas, TN – teisingai klasifikuotas neigiamas rezultatas, FN – klaidingai klasifikuotas neigiamas rezultatas.

1.4 Apibendrinimas

Dauguma autorių įprastam eismui aptikti naudoja „anomalijos“ sąvoką. Ji apibrėžiama kaip statiškai reikšmingas nuokrypis nuo įprastinio modelio, kurį sudaro įprastinė statistinė dauguma. Dažniausiai autoriai naudoja įprastiniam modeliui apmokyti neprižiūrimąjį mokymo metodą, kuris koncentruojasi į įprastinį laivų judėjimo modelį [16] [15] [14]. Kiekvienas nukrypimas nuo įprastinio judėjimo modelio vertinamas kaip potencialaus galimo incidento atvejis [15]. Šiuo metu mokslininkai tyrinėdami „įprastinio“ laivų eismo modelius dažnai koncentruojasi į metodus tokius kaip: Bajeso teorema, Gauso procesas, save organizuojantys Kohoneno žemėlapiai (SOM) ir keliaujančio pardavėjo algoritmas (TSP), kurie naudoja paketinį mokymą (*angl.: batch learning*). Paketinio mokymo atveju modelio apmokymui kiekviename cikle ar epochoje naudojama visa mokymosi duomenų imtis. Todėl šie metodai pasižymi tuo, kad jais paremtų algoritmų laiko sudėtingumas auga eksponentiškai, kai laivų eismo duomenys auga linijiniu būdu. Ši priežastis neleidžia šių algoritmų ir metodų be papildomo modifikavimo panaudoti didžiais duomenis apsimokančio algoritmo kūrimui.

Išanalizavus mokslinę literatūrą esamų neįprasto laivų eismo algoritmų tema buvo pasirinkti kriterijai, kuriems buvo priskirti svoriai. Pagal svorinius suminius balus buvo išrinkti SOM Kohoneno ir virtualaus feromono metodai tobulinimui ir pritaikymui dirbti su didžiais laivų eismo duomenimis. Šie metodai buvo detaliam išnagrinėti ir surasti pritaikymo metodai didiesiems duomenims, tokiems kaip SOM ir virtualaus metodų apjungimas, stochastinis gradientinis pakilimas, modelių sintezė bei paskirstytos skaičiuojamosios SQL užklauskos didiesiems duomenims.

2 DIDŽIAISIAIS DUOMENIMIS APSIMOKANČIO ALGORITMO IR POSISTEMIO KŪRIMAS

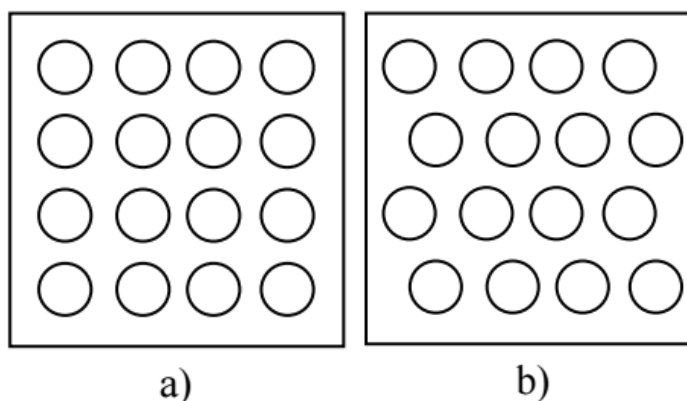
Šiame skyriuje aprašomas sukurtas apsimokančio algoritmo ir posistemio kūrimas neįprastam laivų eismui aptikti jūrų uoste egzistuojančioms laivų stebėjimo ir kontrolės sistemoms. Pradžioje aprašomas naudojamų metodų modifikavimas, pritaikant neprižiūrimam apsimokymui didžiais AIS duomenimis. Po to aprašomas algoritmo kūrimas šių jau modifikuotų metodų pagrindu. Sekančioje dalyje aprašomas sukurtas posistemis ir jo integravimas į egzistuojančią sistemą. Algoritmo ir posistemio pagrindu sukuriama prototipas ir jis aprašomas.

2.1 Algoritmo kūrimas

Šiame poskyryje aprašomas metodų modifikavimas, pritaikant mokymuisi AIS didžiaisiais duomenimis ir panaudojant šiuos modifikuotus metodus algoritmo sukūrimo neįprastam jūrų laivų eismui aptikti. Pagrindinė algoritmo užduotis aptikti eismą neatitinkanti modelį, kurį sudaro statistinė įeinančių eismo vektorių dauguma.

2.1.1 SOM ir virtualaus feromonų metodų apjungimas

Kuriame algoritme naudojamas Kohoneno SOM savi-organizuojantis neuroninis tinklas (angl.: self-organizing map). Kuriamame algoritme SOM tinklas atliks didžiųjų duomenų klasterizavimą. Projektuojamame algoritme bus naudojamas stačiakampės struktūros SOM tinklas (12 pav. a), kuriame kiekvienas neuronas bus sujungtas su keturiais kaimynais.

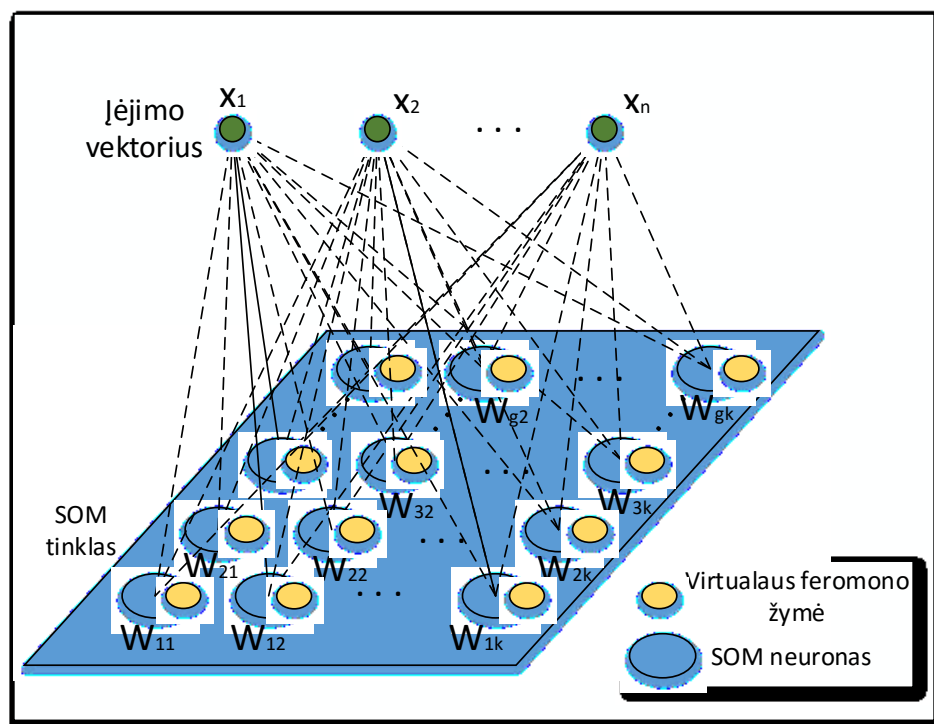


12 pav. SOM tinklo struktūra: a) stačiakampė, b) šešiakampė

Neuroninio tinklo įėjimo vektorių sudarys tokios uosto laivų savybės kaip: Judančio laivo geografinė ilguma, platumą, laivo tipas pagal AIS, laivo ilgis, judėjimo greitis, judėjimo kryptis, meteorologinės sąlygos, laivo kategorija.

Sukurtame algoritme SOM atlieka klasterizavimo funkciją. Pagal klasteryje esančių vektorių kiekį galima spręsti, kiek šis klasteris reprezentuoja daugumą, t.y. jei klasteryje yra daugiau už nustatytą slenkstį elementų, tada galima traktuoti, kad laivų eismas yra mažas, tada tai neįprastas eismas (anomalija). Įprastinėje SOM tinklo versijoje, norint surasti, kiek klasteryje yra elementų būtina visus įeinančių duomenų vektorių palyginti su SOM tinklo neuronais. Artimiausias neuronas bus konkretaus duomens klasterio žymė. Įprastinėje SOM versijoje, norint suskaičiuoti, kiek klasteryje yra vektorių, būtina visiems įeinantiems vektorių surasti SOM neuronus laimėtojus. Tam atlikti būtina panaudoti visus įeinančius duomenis, kad didžiųjų duomenų atveju reikalingi dideli skaičiavimo resursai, ypač, kai duomenų imtyje yra virš kelių pentabaitų.

Pritaikant SOM mokymuisi didžiais duomenimis būtina mažinti skaičiavimo operacijų skaičių ir ypač, tokių kurios eina per visą didžiųjų duomenų imtį. Tam, kad kiekvieno atnaujinimo metu nereiktų perskaičiuoti, kiek klasteryje yra elementų, prie kiekvieno SOM neurono pridama virtualus feromono žymė. Modifikuoto SOM struktūra parodyta paveikslėlyje (13 pav.). Kaip ir klasikinio SOM atveju tinklą sudaro neuronų grupė, tarpusavyje sujungta kaimynystės funkcija ir turinti sąryšį su kiekvienu įeinančiu neuronu.



13 pav. Modifikuotas SOM tinklas su virtualaus feromono žyme.

Virtualaus feromono žymė veikia tokiu principu, kad, kai išrenkamas neuronas laimėtojas, feromono intensyvumas padidinamas vienu žingsniu. Tokiu būdu kuo dažniau neuronas yra laimėtojas (turi daugiau elementų klasteryje), tuo jo žymės feromono intensyvumas yra didesnis.

Neuronas laimėtojas paskaičiuojamas pagal formulę:

$$c = \underset{i,j}{\operatorname{argmin}}\{\operatorname{dist}(\bar{X}_k, W_{ij})\}, \quad (13)$$

čia c – neuronas laimėtojas, k – įeinančio duomens numeris, \bar{X}_k – k duomens vektorius, W_{ij} – ij SOM neuronas, dist – ne Euklido erdvės atstumo funkcija.

Modifikuoto SOM tinklo neprižiūrimas mokymas vyksta panašiai kaip klasikinio tik keičiama virtualaus feromono žymės intensyvumo reikmę pagal formules [19]:

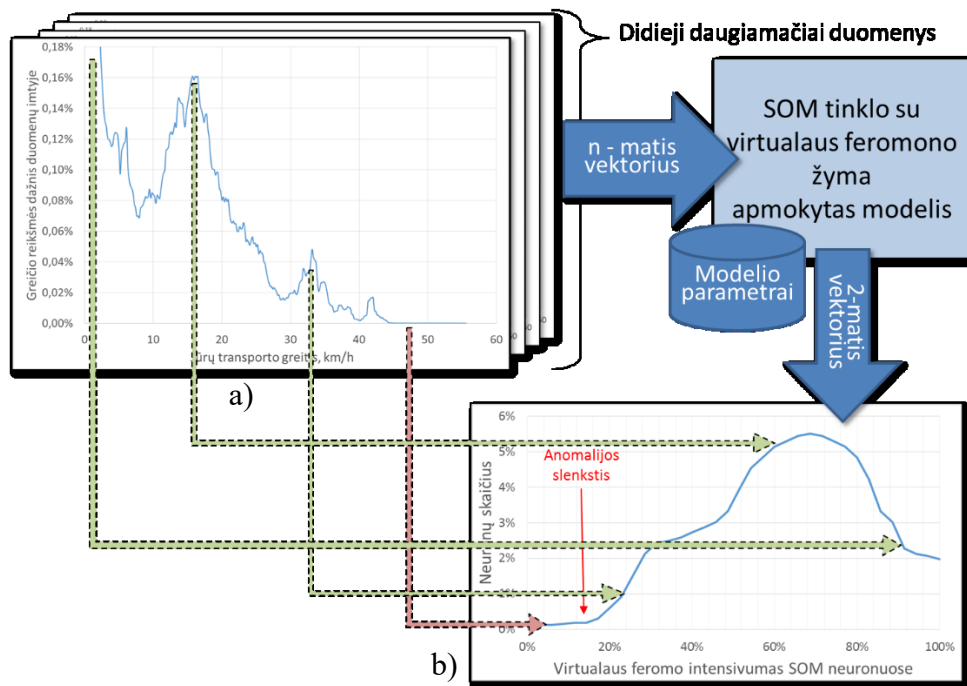
$$\tau_{ij}(t + n) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij} \quad (14)$$

$$\Delta\tau_{ij} = \sum_{k=1}^m 1[W_{ij} = c] \quad (15)$$

Čia τ_{ij} – virtualaus feromono intensyvumas SOM žymėje, t – laikas, n – laiko žingsnis, ρ – feromono garavimo koeficientas, c – neuronas nugalėtojas atitinkamam įėjimo vektoriui, m – įeinančių vektorių skaičius, W_{ij} - ij neurono vektorius.

SOM neuronai grupuoja į klasterius didesnes ir mažesnes įeinančių vektorių grupes. Didesnio klasterio SOM neuronas turės didesnio intensyvumo feromono žymę, o mažesnio klasterio - mažesnę. Didesnis elementų kiekis klasteryje taip pat reiškia, kad šis klasteris reprezentuoja didesnio tikėtimumo duomenis, kad jie yra įprasti, t.y. kuo daugiau elementų klasteryje, tuo labiau tikėtina, kad tai įprasto eismo klasteris.

Paveikslėlyje (14 pav.) pavaizduota, kaip daugiamačiai didžiųjų duomenų įprasto eismo tikėtimumas transformuojamas į virtualaus feromono intensyvumo pasiskirstymą SOM tinkle.



14 pav. Didesnio tikėtinumo transformacija į virtualaus feromono intensyvumą.

Paveikslėlio a dalyje pavaizduota greičio reikšmių dažnio histograma, kuri yra viena iš daugiamatį vektorių histogramų. Modifikuoto SOM tinklo pagalba daugiamatis vektorius transformuojamas į dvimatį virtualaus feromono reikšmių dažnio neuronuose vektorių (14 pav. b). Raudona linija parodoma mažesnio tikėtinumo perkėlimą į mažesnio feromono intensyvumo reikšmę, o žalia - didesnio tikėtinumo į mažesnę. Virtualaus feromono intensyvumo SOM neuronuose histogramos pagalba matomas įprasto eismo tikėtinumo pasiskirstymas. Nustačius feromono intensyvumo slenkstį (anomalijos slenkstis), eismas klasifikuojamas kaip įprastas, jei yra didesnis, negu slenkščio reikšmė, arba neįprastas, nei reikšmė mažesnė. Derinant slenkščio reikšmę galima reguliuoti algoritmo preciziškumą ir specifiškumą.

2.1.2 Dviejų modifikuotų SOM modelių sintezė, naudojant tiesioginio mokymosi metodą

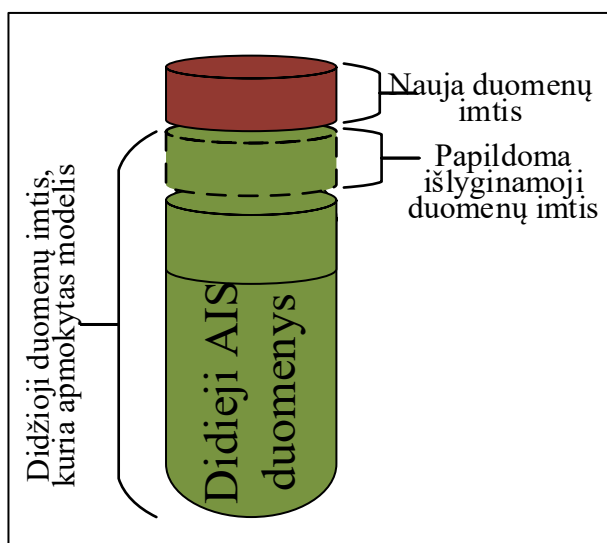
Įprastiniam SOM tinklui naudojamas paketinis (*angl.: batch learning*) mokymo metodas, kai modelio parametrų atnaujinimui naudojama visa duomenų imtis. SOM algoritmo laiko sudėtingumas yra [18]:

$$\theta(n^2 \cdot m \cdot E), \quad (16)$$

čia n – duomenų imties elementų skaičius, m – neuronų skaičius tinkle, E – mokymo epochų skaičius.

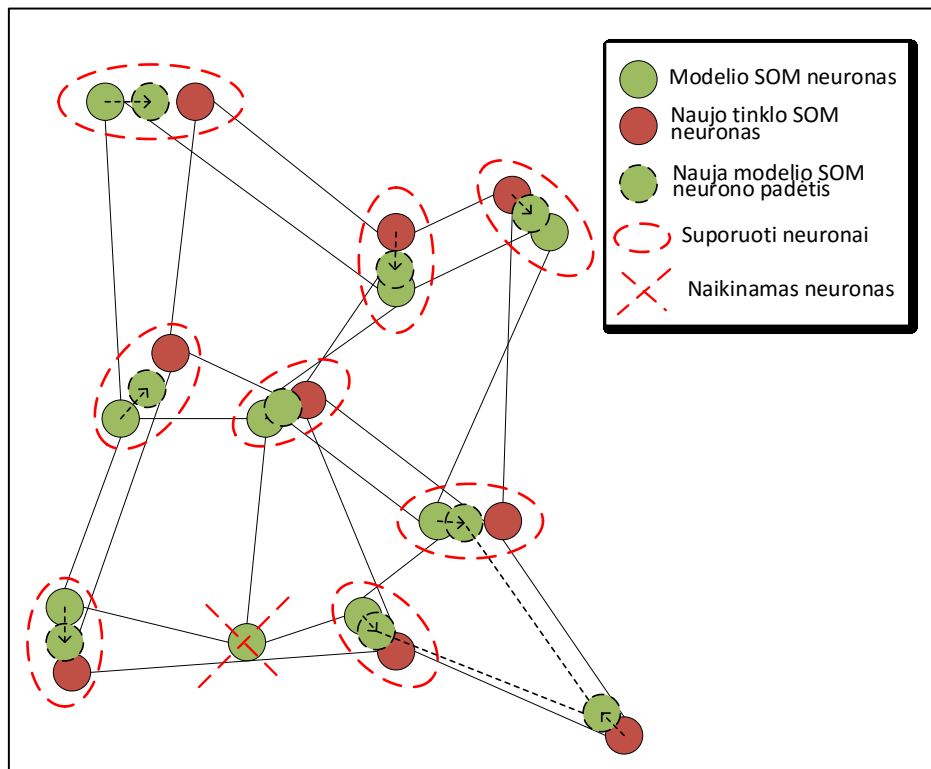
Esant nedidelės apimties duomenims, paketinis SOM apmokymas gali būti naudojamas modelio apmokymui realiu laiku. Atsiradus mokymo duomenų imtyje papildomų duomenų, paketinio mokymosi atveju būtų naudojama visa duomenų imtis, kas kiekvieną kartą sudarytų polinominio laiko sudėtingumo skaičiavimo operacijų skaičių (pagal formulę 16). Didžiųjų duomenų atveju skaičiavimo prasme tai - labai imlu.

Algoritme panaudotas tiesioginio mašininio mokymo metodas (*angl.: online machine learning*). Šio metodo pagalba SOM tinklas apmokomas ne visa didžiųjų duomenų mokymosi imtimi, o tik ta dalimi, kuri pasikeičia arba atnaujinama. Paveikslėlyje (15 pav.) pavaizduota visa didžiųjų duomenų imtis, kuri padalinta į tris dalis. Įprasto eismo modelis apmokomas paėiliui, pateikiant duomenis. Suformuojami du įprasto eismo modeliai: vienas apmokytas iš anksčiau didžiais duomenimis, kitas apmokomas nauja duomenų imtimi. Nauja imtis gali būti naujai į bendrą imtį patekę duomenys arba sekancio žingsnio imtis, esanti žingsniniam apmokymui. Po to abu modeliai sujungiami SOM sintezės būdu į vieną modelį. Tiesioginiam mokymui keliami reikalavimai, kad nauja duomenų imtis būtų tokio pat pasiskirstymo. Norint tai užtikrinti, naujo modelio apmokymui naudojama iš esančių duomenų papildoma išlyginamoji duomenų imtis, kurios dydis yra toks pat kaip ir naujų duomenų. Tokiu būdu sumažinamas duomenų triukšmas ir užtikrinamas modelio SOM tinklo neuronų konvergavimas į lokalius optimumus.



15 pav. Tiesioginio mokymosi duomenų imtis

Apmokius modifikuotą SOM nauja duomenų imtimi, su juo ir su prieš tai apmokytu SOM atliekama modelių sintezė. Modelių sintezė pavaizduota paveikslėlyje (16 pav.). Žalia spalva pavaizduotas prieš tai apmokyto modeliu SOM tinklas, raudona - naujos duomenų imties apmokytas SOM tinklas. Šviesiai žalia nauja neurono padėtis po sintezės. Abiejų SOM tinklų matmenys yra vienodi. Naudojami modifikuotas SOM su pridėta virtualaus feromono reikšme.



16 pav. Dviejų SOM tinklų sintezė

Atliekant SOM tinklų sintezę vykdomi šie veiksmai.

- Iš pradžių surandamos abiejų tinklų neuronų poros, naudojant artimiausio kaimyno metodą. (16 pav. raudona punktyrinė elipsė).
- Jei atstumas iki artimiausio kaimyno yra didesnis negu $(\delta + 1)\bar{d}$, tada naujo tinklo neuronas paliekamas be poros ir sukuriama naujas neuronas modelio tinkle bei sujungiamas su kitais keturiais artimiausiais neuronais. Čia \bar{d} – atstumo tarp surastos poros neuronų vidurkis, δ – neurono atmetimo koeficientas. Praktinėje realizacijoje naudota 1,5 reikšmė.
- Jei modelio virtualaus feromono žymė yra lygi nuliui, tada neuronas naikinamas (16 pav. raudonos punktyrinės linijos).
- Modelio neuronas vektorius perstumiamas naujojo SOM neurono porininko kryptimi. Perstūmimui naudojama formulė:

$$\hat{W}_{ij} = W_{ij} + \alpha \left(\|W_{ij} - C_{ij}\| \right), \quad (17)$$

Čia ij – neurono identifikatorius tinkle, \hat{W}_{ij} – nauja modelio neurono vektoriaus reikšmė, W_{ij} – modelio SOM neurono reikmė, C_{ij} – nauja duomenų imtimi apmokyto SOM neurono reikšmė, α – mokymosi greičio koeficientas (praktinėje realizacijoje naudota 0,1 reikšmė).

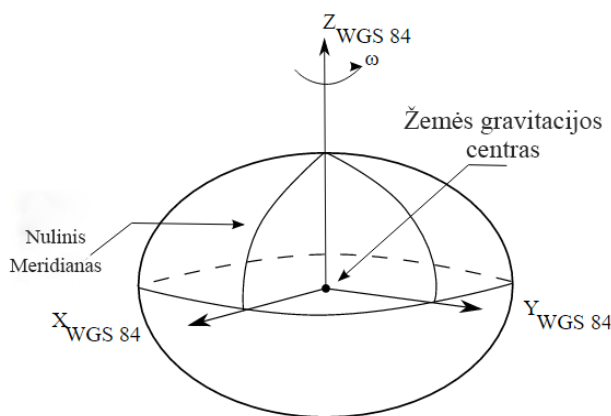
Toks SOM tinklų suliejimas leidžia ne tik realizuoti tiesioginį mašininį mokymą didžiais duomenimis, bet ir užtikrinti algoritmo adaptyvumą, pasikeitus įprastam eismui.

2.1.3 Atstumo tarp neuronų ir įėjimo vektorių skaičiavimas ne Euklido erdvėje.

Įprastinėje SOM tinklo algoritmo modifikacijoje atstumui tarp vektorių lyginti naudojamas Euklido atstumas. Šis būdas gali būti taikomas tik Euklidinės erdvės vektoriams. Kadangi dalis į SOM neuroninį tinklo įėjimą patenkančių vektorių elementų duomenys ne priklauso Euklido erdvei. Norint atstumo matavimo funkciją adaptuoti vektoriams, kuriuos sudaro Euklido ir ne Euklido erdvės elementai, reikia sukurti adaptuotą atstumo skaičiavimo funkciją. Teisingam skirtumų vertinimui tarp dviejų įėjties duomenų parametrų būtina teisingai vertinti visus įėjties vektoriaus elementų dydžius ir jų atstūmus vienas nuo kito, o dar svarbiau, atstumą tarp skirtingų vektoriaus elementų reikšmių. Todėl būtina sudaryti vientisą atstumo vertinimo funkciją, kuri leistų objektyviai įvertinti skirtumus tarp įėjties vektorių. Išanalizuojame į SOM tinklo patenkančių elementų prigimtį ir parenkame tinkamus atstumo skaičiavimo metodus juos apjungdami į vientisą funkciją.

Atstumo skaičiavimas WGS84 koordinatėse erdvėje.

Laivo geografinė pozicija AIS sistemoje yra nusakoma geodezinėmis WGS84 standarto koordinatėmis. Tai įprastinė Žemės planetai naudojama koordinatėse sistema, sudaryta iš įprastinio sferoidinio atskaitos paviršiaus (17 pav.).



17 pav. WGS84 koordinatėse sistema

Objekto pozicijos nusakymui naudojami kampiniai dydžiai, tokie, kaip geografinė platumas ir geografinė ilguma. Geografinė platumas vadinamas kampas, kurį sudaro Žemės rutulio spindulys, nuleistas iš ieškomo taško į Žemės centrą su ekvatoriaus plokštuma ($\angle \varphi$). Geografinė ilguma

vadinamas kampas, kurį sudaro dvi besikertančios meridianų plokštumos, viena iš kurių yra pradinė ($\angle\lambda$). Geografinė ilguma matuojama nuo pradinio meridiano į rytus ir į vakarus. Ji kinta nuo 0 iki 180°.

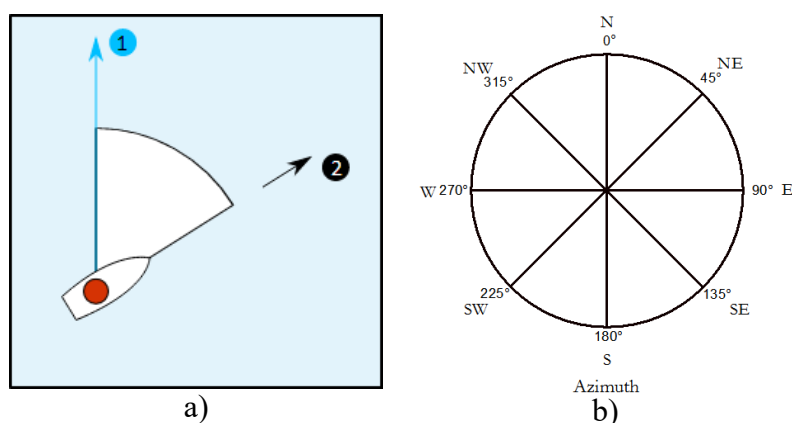
Šios erdvės matavimo būdas nepriklauso Euklido erdvei, todėl negalima naudoti ir tarp dviejų koordinatinių atstumui nustatyti Euklido atstumo metodu. Norint tiksliai įvertinti atstumą tarp laivų, būtina matuoti atstumą, kurį būtina nuplaukti laivui tarp reikiamų taškų. Tai būtų atstumo matavimas pagal žemės rutulio sferos paviršių. Šiam atstumui naudojama Haversine formulė [20]:

$$hav() = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right), \quad (18)$$

čia $hav()$ – atstumas kilometrais tarp dviejų koordinatinių pagal sferos plokštumą, r – sferos radiusas kilometrais (šiuo atveju Žemės), φ_2, φ_1 – platumos 1-ame ir 2-ame taškuose radianais, λ_2, λ_1 – ilguma 1-ame ir 2-ame taškuose radianais.

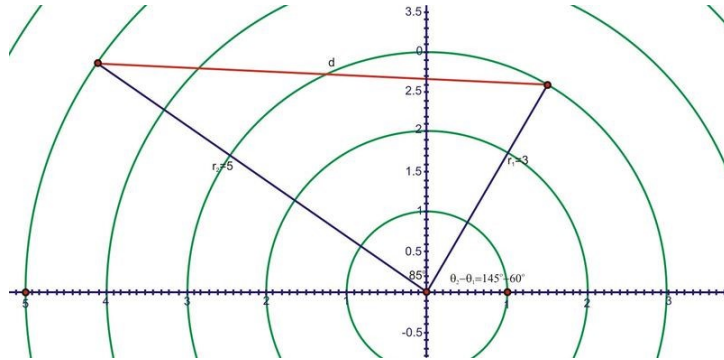
Atstumo skaičiavimas poliarinių koordinatinių erdvėje

AIS sistemos yra gaunami duomenys apie laivo judėjimo kryptį ir greitį. Jūrinėje navigacijoje judėjimo kryptis (*angl.: bearing*), tai kampas tarp magnetinio Šiaurės poliaus arba tikrojo Šiaurės poliaus ir laivo judėjimo krypties. Numeriu 1 (18 pav. a dalyje) yra pažymėtas kryptis kurioje randasi šiaurinis polius, o skaičiumi 2 pažymėta laivo plaukimo kryptis. Kampas tarp jų sudaro laivo judėjimo kryptį (azimutą). Kampas skaičiuojamas nuo šiaurės krypties (18 pav. b dalis), kur Šiaurė N yra 0°, Rytai E yra 90° ir t.t.



18 pav. Laivo judėjimo krypties poliarinių koordinatinių sistema

Laivo greitis AIS sistemoje yra matuojamas jūriniais mazgais. Kadangi judėjimo kryptis ir greitis yra tarpusavyje glaudžiai susiję, buvo nutarta tarpusavyje juos sujungti į vieną poliarinės koordinatinių sistemų vektorių ir atstumas bus matuojamas tarp dviejų poliarinių vektorių (19 pav.).



19 pav. Atstumo matavimas tarp dviejų poliarinių vektorių

Atstumui matuoti tarp greičio vektorių naudosime atstumo tarp dviejų poliarinių vektorių skaičiavimo formulę:

$$d = \sqrt{v_1^2 + v_2^2 - 2v_1v_2\cos(\theta_1 - \theta_2)}, \quad (19)$$

Čia d – atstumas tarp vektorių, v_1, v_2 – 1-o ir 2-o laivo greitis, θ_1, θ_2 – Laivo judėjimo kryptis (azimutas).

Šis atstumo tarp dviejų laivų judėjimo vektorių matavimo būdas leis nustatyti objektyvų skirtumą tarp dviejų judančių laivų krypčių ir greičių.

Bendro atstumo tarp vektorių formulės sudarymas

Atstumui skaičiuoti tarp vektorių sudaroma atstumo skaičiavimo funkcija, kurioje suderinamos skirtingų erdvių vektoriai. Šiai formulei sudaryti panaudotos atstumo skaičiavimo formulės, tokios kaip haversine (formulė 18), poliarinių koordinatinių ir Euklido erdvės. Pagal jas sudaryta formulė:

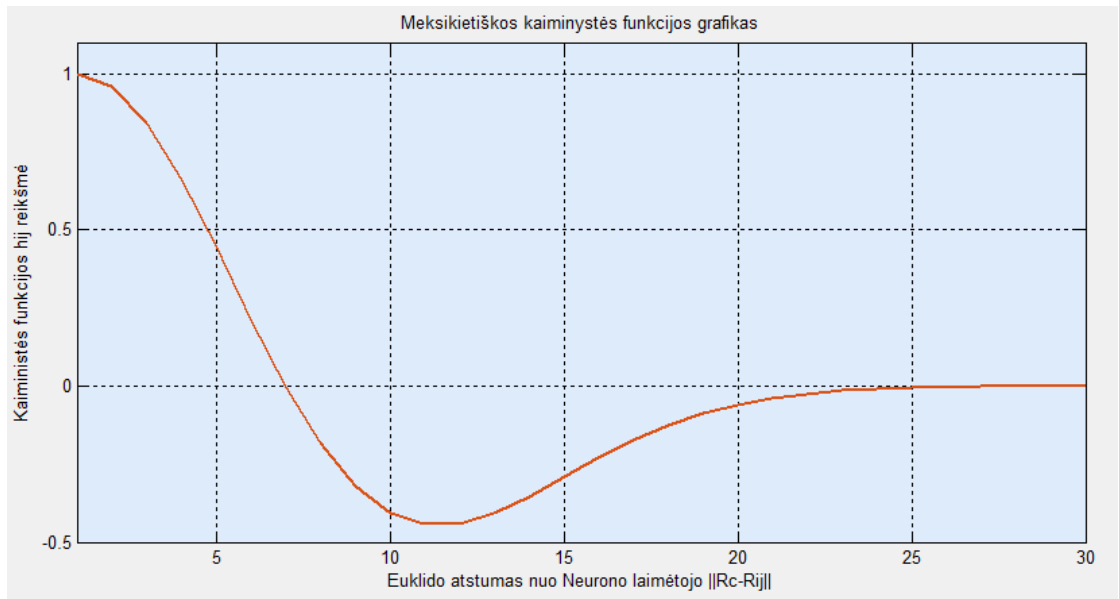
$$d_{AIS} = \sqrt{\begin{aligned} &hav(\bar{X}_{kLat}, W_{ijLat}, \bar{X}_{kLon}, W_{ijLon})^2 + (|\bar{X}_k - W_{ij}|)^2 + \\ &+ \bar{X}_{kSs}^2 + W_{ijSs}^2 - 2\bar{X}_{kSs}W_{ijSs} \cos(\bar{X}_{kSd} - W_{ijSd}) + \\ &+ \bar{X}_{kWs}^2 + W_{ijWs}^2 - 2\bar{X}_{kWs}W_{ijWs} \cos(\bar{X}_{kwd} - W_{ijwd}) \\ &+ \bar{X}_{kBs}^2 + W_{ijBs}^2 - 2\bar{X}_{kBs}W_{ijBs} \cos(\bar{X}_{kBd} - W_{ijBd}) \end{aligned}} \quad (20)$$

čia d_{AIS} – atstumas tarp AIS duomenų vektorių. $hav()$ – haversine atstumo funkcija (formulė 18), \bar{X}_{kLat} – įeinančio duomens geografinė platuma, \bar{X}_{kLon} – įeinančio duomens geografinė ilguma, W_{ijLat} – neurono vektoriaus geografinė platuma, W_{ijLon} – Neurono vektoriaus geografinė ilguma, $\bar{X}_{kSs}, \bar{X}_{kSd}$ – įeinančio duomens laivo greitis ir kryptis, W_{ijSs}, W_{ijSd} – neurono vektoriaus laivo

greitis ir kryptis, \bar{X}_{kWS} , \bar{X}_{kWd} - įeinančio duomens vėjo stiprumas ir kryptis, \bar{X}_{kBS} , \bar{X}_{kBd} - bangavimo stiprumas ir kryptis, $|\bar{X}_k - W_{ij}|$ – likusių vektorių esančių Euklido erdvėje koordinatų skirtumas.

2.1.4 SOM kaimynystės funkcijos modifikavimas

Kuriamame algoritme naudojama „Meksikietiškos kepurės“ atstumo funkcija (20 pav. **Error! eference source not found.**).



20 pav. „Meksikietiškos kepurės“ kaimynystės funkcijos grafikas

Ji išreikšta formule:

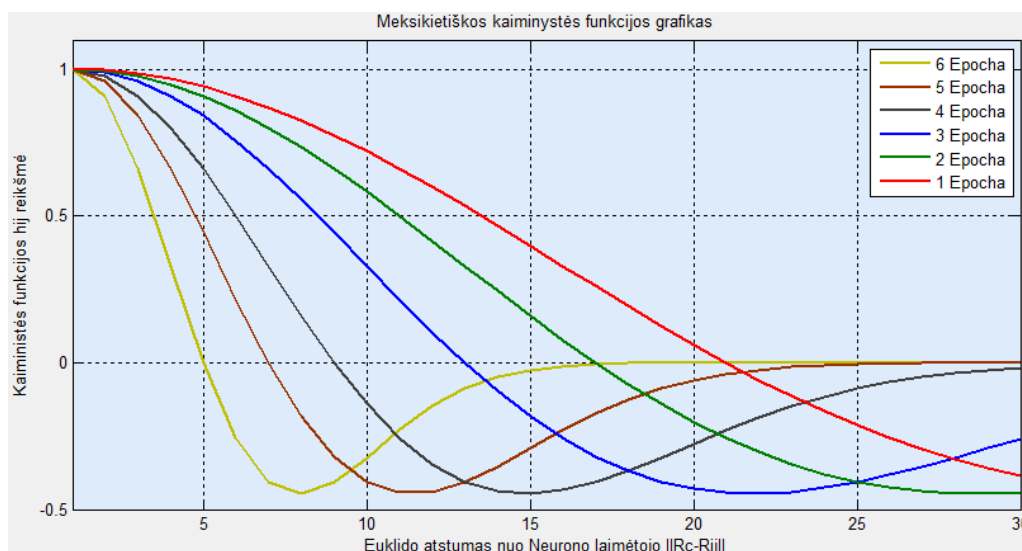
$$h_{ij}^c(t) = \left(1 - \frac{1}{\sigma^2(t)} \cdot \|\hat{R}_c - \hat{R}_{ij}\|^2\right) \cdot e^{-\frac{1}{2} \left(\frac{\|\hat{R}_c - \hat{R}_{ij}\|^2}{\sigma^2(t)}\right)}, \quad (21)$$

Čia σ – tai funkcija, nusakanti neuroninio tinklo kaimynystės erdvę (*angl.: space of the neighborhood*). Pradžioje ji įtraukia beveik visą SOM neuroninį tinklo žemėlapi. Procesui konverguoti būtina, kad $h_{ij}^c(t) \rightarrow 0$, kai $t \rightarrow \infty$. Šioje kaimynystės funkcijoje už šį konvergavimą atsakinga σ funkcija. Ją išreikšta formule [18] [4]:

$$\sigma = \max(k, g) \cdot \frac{E+1-\hat{E}}{E}, \quad (22)$$

Čia E - prieš tinklo mokymą nustatytas viso mokymo epochų skaičius, \hat{E} – vykdomos epochos numeris, k, g – neuroninio tinklo plotis, išreikštas neuronų skaičiumi eilutėje arba stulpelyje. Paveiksle (21 pav.) pavaizduotas „Meksikietiškos kepurės“ kaimynystės funkcijos kitimas, priklausomai nuo epochos. Vykdomai epochai artėjant link iš viso nustatytų epochų skaičiui, kaimynystės funkcijos pagalba keičiami tik vis artimesni nuo laimėtojo neuronai. Būtent

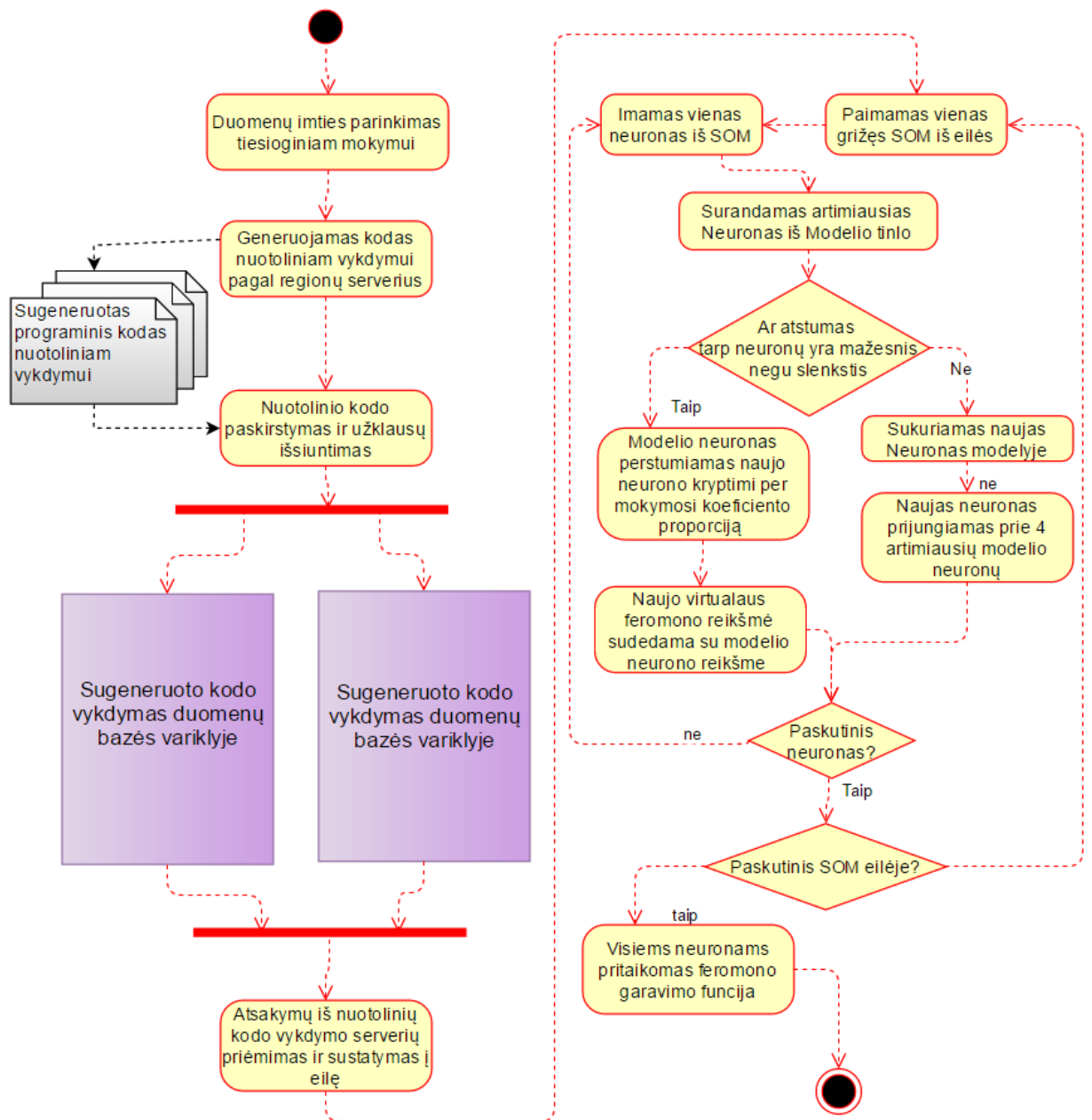
„Meksikietiškos kepurės“ funkcija neuroninio tinklo apmokymui turi vieną išskirtinę savybę, kad artimesnius kaimynus pritraukia prie laimėtojo, tolimesnius atstumia, tokiu būdu sukeldama didesnę kontrastą tarp apmokamų klasterių. Stebint šios funkcijos kitimą (21 pav. 6 epocha) matome, kad kaimynystės funkcija iki $h_{ij}^c(t)$ yra teigiama iki Eulido atstumo vertės 5, o po šios vertės neigiama. Teigiama reikšmė nusako, kad neuronai kurio Euklido atstumas yra iki 5 priartinami prie įėjimo vektoriaus reikšmės mokymo metu, neigiama, kad atitolinami. Ši savybė padeda išlaikyti didesnius atstūmus tarp neuronų, sudarant aiškesnes ribas tarp klasterių.



21 pav. „Meksikietiškos kepurės“ kaimynystės funkcijos kitimas pagal epochos numerį

2.1.5 Neįprasto laivų eismo aptikimo modelio apmokymo algoritmas

Sukurto apsimokančio didžiaisiais duomenimis algoritmo neįprastam laivų eismui aptikti diagrama pavaizduota paveikslėlyje (22 pav.). Šiame algoritme realizuoti aukščiau aprašyti metodai ir modifikacijos. Algoritmas sudarytas iš šių pagrindinių dalių: duomenų apimties pasirinkimas ir nuotoliai vykdomo kodo generavimas, išskirstyto kodo vykdymas, naujų SOM tinklų ir modelio SOM tinklo sintezė.



22 pav. Neįprasto laivų eismo aptikimo algoritmo diagrama

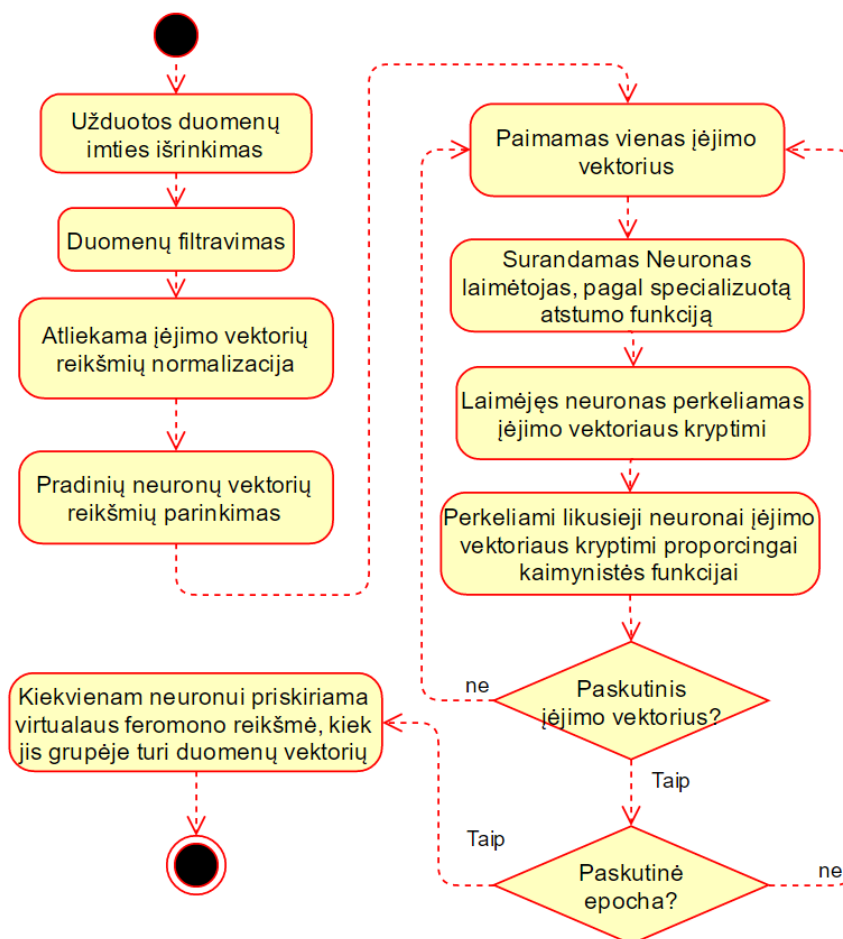
Algoritmo vykdymo eiga.

Duomenų apimties pasirinkimas ir nuotoliai vykdomo kodo generavimas (22 pav.).

1. Pasirenkama duomenų imtis didžiųjų duomenų masyve, kurią sudaro nauji duomenys ir lyginamoji duomenų imtis.
2. Pagal pasirinktas duomenų imtis generuojamas programinis kodas nuotoliniam vykdymui skirtingose duomenų saugyklose.
3. Sugeneruoto kodo išskirstymas ir išsiuntimas vykdymui skirtingose duomenų bazėse.

Išskirstyto kodo vykdymas nuotolinėje duomenų bazėje (23 pav.).

4. Išrenkami duomenys pagal imties parinktį.
5. Atliekamas duomenų filtravimas, atmetant pasikartojančius ir klaidingus duomenis.
6. Atliekama jėgimo vektorių reikšmių normalizacija.
7. Euristiniu būdu parenkamos pradinės SOM tinklo neuronų vektorių reikšmės.
8. Paimamas vienas jėgimo vektorius iš parinktos duomenų imties.
9. Surandamas neuronas laimėtojas, naudojant sudarytą ne Euklido erdvės funkciją.
10. Laimėjęs neuronas perkeliamas jėgimo vektoriaus kryptimi mokymosi konstantos santykiu.
11. Perkeliami kiti tinklo neuronai jėgimo kryptimi proporcingai kaimynystės funkcijos reikšmei pagal konkretų neuroną.
12. Kartojama nuo 8 punkto, kol nepraeinami visi jėgimo vektoriai.
13. Kartojama nuo 8 punkto, kol neužbaigiamos visos epochos.
14. Kiekvienam SOM neuronui priskiriama virtualaus feromono žymė, kurios intensyvumo reikšmė lygi konkretaus neurono klasterio dydžiui.



23 pav. Neįprasto eismo aptikimo algoritmo dalis, vykdoma nuotoliniame serveryje

Naujų SOM tinklų ir modelio SOM tinklo sintezė (22 pav.).

15. Priimami atsakymai iš nuotoliai vykdyto išskirstyto programinio kodo.
16. Paimamas vienas iš grįžusių SOM tinklų.
17. Iš pasirinkto SOM imamas neuronas.
18. Surandama artimiausias neuronas modelio SOM tinkle.
19. Jei atstumas tarp neuronų mažesnis už slenkstį, jis perstumiamas naujo neurono linkme ir jų virtualaus feromono žymių reikšmės sudedamos.
20. Jei atstumas tarp neuronų didesnis už slenkstį, sukuriamas naujas neuronas modelyje ir prijungiamas prie artimiausių keturių neuronų.
21. Kartojamas nuo 17 punkto, kol neapskaičiuojami visi neuronai.
22. Kartojamas nuo 16 punkto, kol neapskaičiuojami visi neuronai.
23. Panaikinami neuronai, kurių virtualaus feromono intensyvumo reikšmė lygi nuliui.
24. Visoms virtualaus feromono žymėms pritaikoma feromono garavimo funkcija.

2.1.6 **Algoritmo laiko sudėtingumo skaičiavimas**

Skaičiuojant algoritmo laiko sudėtingumą [1], atsižvelgiama į du apmokymo atvejus:

- kai apmokomas modelis nuo pradinės būsenos visa didžiųjų duomenų imtimi – viršutinė riba
- kai vyksta modelio apmokymas naujais duomenimis – apatinė riba

Čia n – didžiųjų duomenų kiekis, m – naujų duomenų kiekis

Viršutinė laiko sudėtingumo riba, kai algoritmas apmokomas nuo pradinės modelio būsenos:

Pradžioje apskaičiuojame algoritmo vykdymo laiko priklausomybę nuo įeinančių duomenų:

$$T(n, m) = \left\lceil \frac{n}{m} \right\rceil (T_1 + T_2 + T_3 + T_4 + T_5 + T_6 2m + T_7 m + T_{8:14} (m^2 NE) + T_{15:24} N), \quad (23)$$

čia $T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_{8:14}, T_{15:24}$ – atitinkamų algoritmo žingsnių vykdymo laiko konstantos, priklausančios nuo techninės įrangos skaičiuojamųjų resursų, N – neuronų skaičius SOM tinkle. E – epochų skaičius.

Pagal sudarytą vykdymo laiko priklausomybės funkciją sudarome algoritmo laiko sudėtingumo funkciją asimptotine notacija:

$$\theta(n, m) = \theta(n \cdot m), \quad (24)$$

Apatinė laiko sudėtingumo riba kai algoritmas apmokomas atsinaujinančiais duomenimis:

Pradžioje apskaičiuojame algoritmo vykdymo laiko priklausomybę nuo įeinančių duomenų:

$$T(m) = (T_1 + T_2 + T_3 + T_4 + T_5 + T_6 2m + T_7 m + T_{8:14}(m^2 NE) + T_{15:24} N), \quad (25)$$

čia $T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_{8:14}, T_{15:24}$ – atitinkamų algoritmo žingsnių vykdymo laiko konstantos, priklausančios nuo techninės įrangos skaičiuojamųjų resursų, N – neuronų skaičius SOM tinkle. E – epochų skaičius.

Pagal sudarytą vykdymo laiko priklausomybės funkciją sudarome algoritmo laiko sudėtingumo funkciją asimptotinė notacija:

$$O(m) = O(m^2), \quad (26)$$

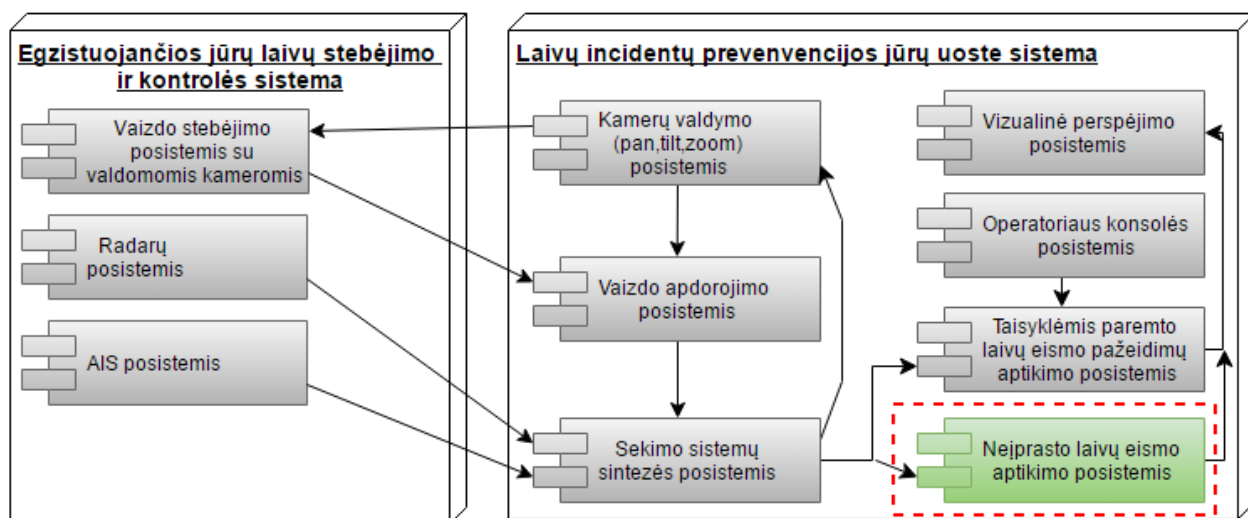
čia $m \ll n$ – atsinaujinantys duomenys ženkliai mažesni už visą didžiųjų duomenų imtį.

2.2 Posistemio kūrimas

2.2.1 Posistemio integravimas į egzistuojančias laivų stebėjimo ir kontrolės sistemas

Sukurtas algoritmas ir posistemis yra jūrų uosto transporto eismo kontrolės sistemos dalis. Pakrantės jūrų transporto kontrolės sistemos architektūra pavaizduota (24 pav.). Ši sistema skirta aptikti jūrų transporto judėjimo pažeidimus pagal nustatytas taisykles. Į šią sistemą bus integruotas kuriamas posistemis su neįprasto laivų aptikimo algoritmu, kuris nustato įprastą transporto judėjimą ir galimą nukrypimą nuo jų. Jūrų transporto kontrolės sistema integruojama su egzistuojančiomis jūrų transporto sistemomis, tokiomis, kaip Radarų sistema, AIS (automatinio identifikavimo sistema) bei pakrantės ir jūrų uosto vaizdo stebėjimo kameros. Kadangi stebėjimo vaizdo kameromis sistema negali nustatyti realios laivo geografinės pozicijos, todėl yra būtina vaizdinę informaciją apdoroti skaitmeniniu būdu, atlikti pirminę analizę ir šiuos duomenis perduoti tolimesnei analizei. Informacija apie jūrų transporto geografinę poziciją iš egzistuojančių stebėjimo sistemų perduodama į sekimo sistemų sintezės posistemę, kurioje apjungta informacija toliau perduodama į galimų pažeidimų aptikimų posistemės. Sistemoje yra dviejų tipų posistemės: taisyklėmis grįsta laivų eismo galimų

incidentų aptikimo posistemė ir adaptyvaus algoritmo galimų incidentų prevencijai jūrų uoste posistemė.



24 pav. Jūrų uosto laivų stebėjimo ir kontrolės sistemos architektūros diagrama.

Pagrindinė taisyklėmis grįstos posistemės užduotis - aptikti pažeidimą pagal iš anksto operatoriaus ar administratoriaus nustatytas taisykles. Kaip pavyzdys tai galėtų būti: pavojingos sekumos, rifai ir kt. Pagrindinė sąlyga taisyklių nustatymui – jų kiekis turi būti ribotas ir su laiku nekisti.

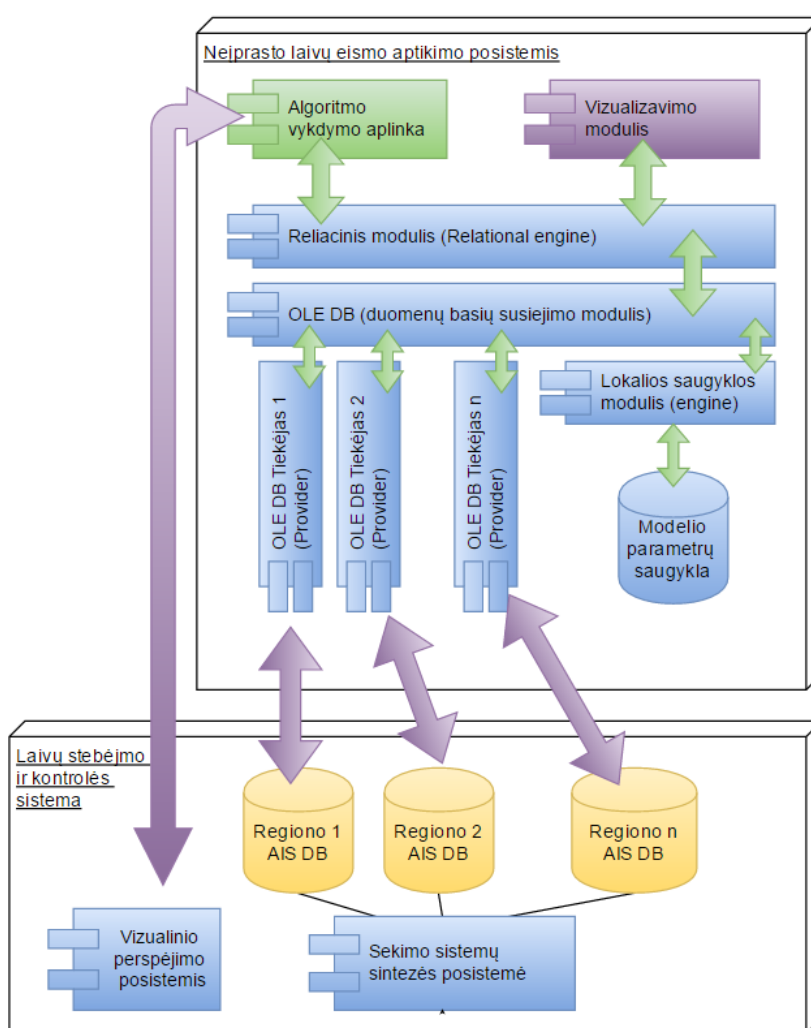
Didžiaisiais duomenimis apsimokančio algoritmo neįprastam laivų eismui aptikti jūrų uoste posistemio pagrindinis tikslas – pagal gaunamus duomenis iš sekimo sistemų sintezės posistemės išmokti įprastus laivų eismo modelių tinklelius ir signalizuoti apie galimą pažeidimą, esant nukrypimui nuo įprastinio eismo modelio. Tiek apsimokymas, tiek signalizavimas atliekamas, vertinant tokius faktorius, kaip laivo klasė, savaitės diena ir metų laikas.

2.2.2 Neįprasto eismo aptikimo jūrų uoste posistemio architektūra

Paveikslėlyje (25 pav.) pavaizduota neįprasto laivų eismo aptikimo posistemio architektūra loginiu požiūriu. Joje pavaizduoti du posistemiai: neįprasto eismo aptikimo ir laivų stebėjimo ir kontrolės. Šie du posistemiai glaudžiai susiję. Neįprasto eismo aptikimo posistemis kreipiasi į kontrolės sistemos didžiuosius duomenis, jų pagrindų apmoko įprasto eismo modelį bei atlieka eismo klasifikavimą į įprastą ir neįprastą. Klasifikavimo rezultatai perduodami į laivų stebėjimo sistemos vizualizavimo posistemį.

Neįprasto laivų eismo aptikimo posistemį sudaro šios dalys (moduliai)(25 pav.):

- algoritmo vykdymo aplinka – čia vykdomas algoritmo programinis kodas, kuris generuoja nuotolinio vykdymo programinį kodą, SOM tinklų sintezę ir eismo klasifikaciją;
- reliacinis modulis – priima užklausas iš algoritmo vykdymo aplinkos, išskaido SQL užklausą į loginius vienetus ir perduoda į OLE DB duomenų bazių susiejimo modulį;
- OLE DB (duomenų bazių susiejimo modulis) – susieja įvairių domenų šaltinius. Tai būtina, kadangi algoritmas turi dirbti su įvairiomis duomenų bazėmis, o OLE užtikrina permatomą skirtingų gamintojų duomenų bazių pasiekiamumą;
- OLE DB Tiekėjas (*angl.: provider*) – skirtingų duomenų bazių variklių tvarkyklės, užtikrinančios konkretaus gamintojo formato užklausų konversiją;



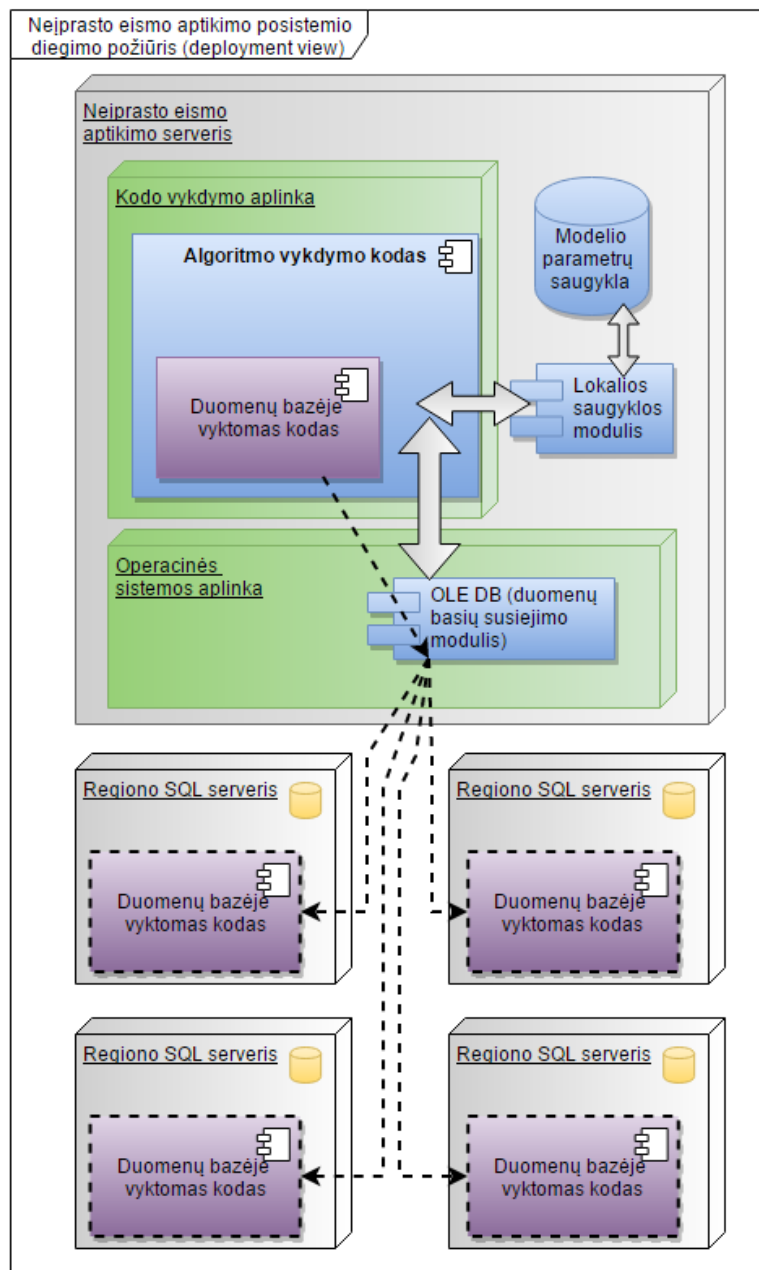
25 pav. Kuriamo posistemio architektūra loginiu požiūriu

- lokali saugyklos modulis – lokali duomenų bazės variklis, kurioje saugomas modelio parametrų informacija;

- vizualizavimo modulis – atsakingas už duomenų vizualizavimą algoritmo derinimo tikslui;
- regiono AIS DB – regiono skirtingų gamintojų duomenų bazės, kuriose laikomi AIS didieji duomenys;

Paveikslėlyje pavaizduota (26 pav.) neįprasto aptikimo posistemis diegimo požiūriu. Išskiriamos dvi serverių grupės: neįprasto aptikimo serveris ir Regiono SQL serveriai, kur laikomi AIS didieji duomenys. Ši architektūra leidžia, kad SQL užklausos galėtų dinamiškai ir lygiagrečiai būti išskirstytos į skirtingų gamintojų ir tiekėjų duomenų bazes.

Kodo vykdymo aplinkoje algoritmo vykdymo programinis kodas generuoja SQL skaičiuojamąsias užklausas, kurios OLE DB pagalba lygiagrečiai išskirstomos į skirtingų regionų SQL serverius, kur vykdomas naujo SOM tinklo apmokymas.

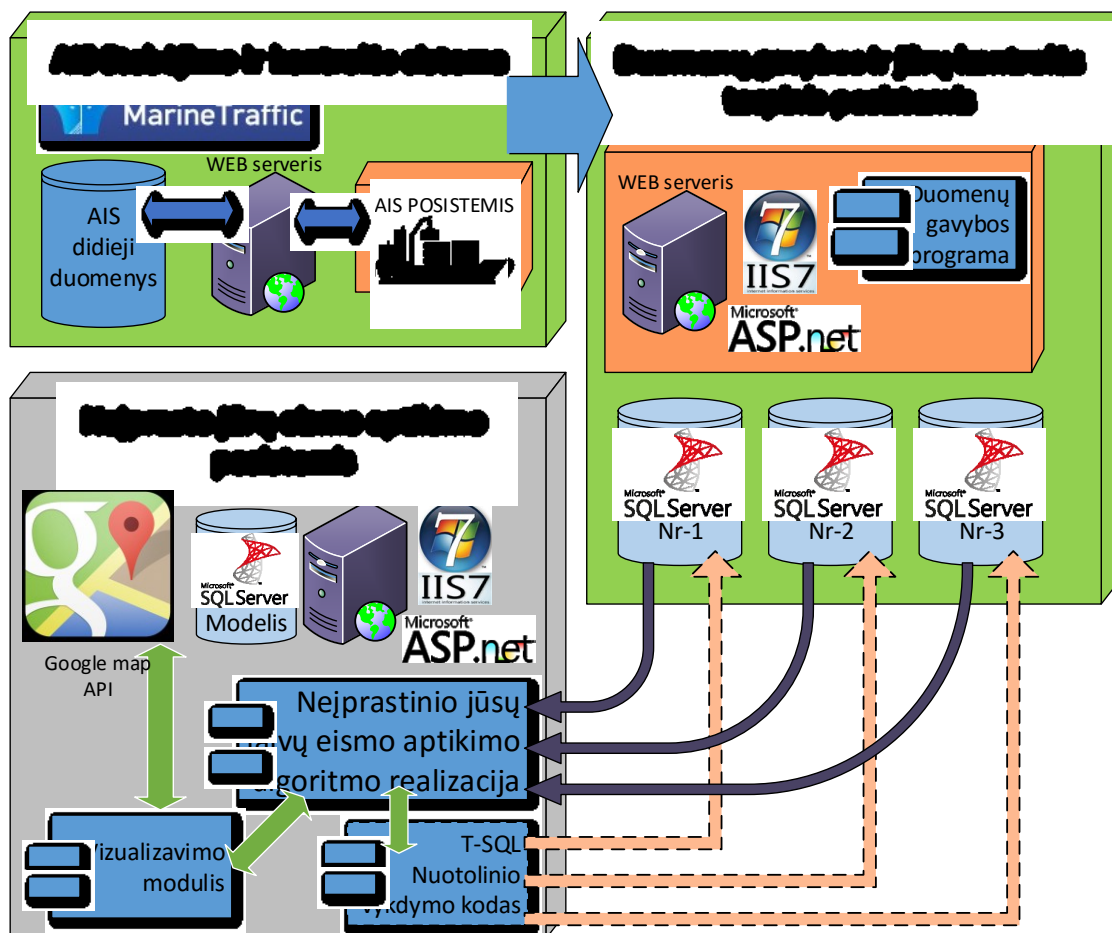


26 pav. Kuriamo posistemo architektūra diegimo požiūriu

Pagrindinė šios architektūros ypatybė yra ta, kad jos pagalba galima pasiekti skirtingų gamintojų ir tiekėjų duomenų bazes, užklauskos vykdomos išskirstytai ir lygiagrečiai, ir didžiojo skaičiuojamoji galia išskirstoma daugeliui serverių. Tokiu būdu labai pagreitinamas apsimokymas didžiaisiais duomenimis.

2.3 Sukurto prototipo architektūra ir sandara

Sukurto algoritmo ir posistemio pagrindu buvo sukurtas didžiaisiais duomenimis apsimokančio algoritmo neįprastam laivų eismui aptikti. Šio prototipo sandara pavaizduota paveikslėlyje (27 pav.). Šiame paveikslėlyje pavaizduoti trys posistemiai: AIS Stebėjimo ir kontrolės sistema, duomenų gavybos ir jūrų kontrolės tarpinis posistemis ir neįprasto jūrų eismo aptikimo posistemis.



27 pav. Neįprasto laivų eismo aptikimo posistemio prototipo sandara

Šis prototipas buvo sukurtas atsižvelgiant į tai, kad nemokama prieiga prie didžiųjų duomenų yra ribota, o pilnam algoritmo ir posistemio verifikavimui būtina pilna. Pilnos prieigos imitavimui buvo sukurta tarpinis duomenų gavybos ir jūrų eismo kontrolės posistemis, kuris periodiškai kreipiasi į „MarineTraffic.com“ AIS duomenų tiekėjo WEB serverį. Gauti duomenis kaupiami tarpinės sistemos SQL serverių klasteryje. Tokiu būdu per kelis mėnesius buvo sukaupta 50 mln. AIS geografinės pozicijos įrašų.

Prototipo neįprasto jūrų eismo aptikimo posistemio realizacijos sandara:

- techninė įranga: PowerEdge R220 Rack Server, PowerEdge R220 Rack Server, PowerEdge R220 Rack Server, PowerEdge R220 Rack Server, Intel® Celeron® G1820 processor, 4GB memory and 500GB hard drive,
- operacinė sistema: Microsoft Windows Server 2014 R2, SQL Server 2014, IIS7, ASP.NET, OLE DB drivers,
- sukurto algoritmo realizacija atlikta C# kalboje naudojant bibliotekas: ASP.NET, MVC5, EF6,
- nuotolinio vykdymo programinis kodas realizuotas: T-SQL programavimo kalboje.
- vizualizavimo modulis realizuotas: JavaScript programinėje kalboje, naudojant Google MAPS API.

3 PROTOTIPO VERIFIKAVIMAS IR ALGORITMO KLASIFIKAVIMO SĄVYBIŲ TYRIMAS

3.1 Verifikavimo ir tyrimo eiga

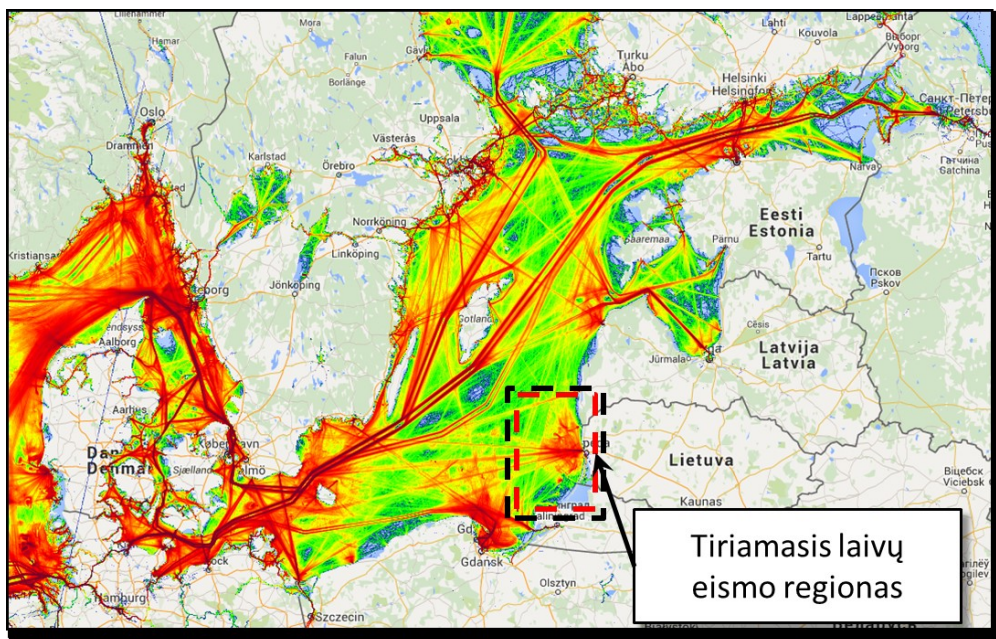
Sukurto prototipo (27 pav.) ir algoritmo klasifikavimo savybių tyrimui buvo sudaryta ši darbo eiga.

1. Pasirenkama AIS didžiųjų duomenų imtis, nustatant aktyvios laivybos geografinį regioną ir tiriamąjį laikotarpį.
2. Prototipo (27 pav.) duomenų gavybos ir laivų eismo kontrolės sistemos tarpinis posistemis nustatomas kaupiti duomenis iš „MarineTraffic.com“ serverio tiesioginius laivų eismo AIS duomenis. Kaupimas pradamas prieš nustatytą tiriamąjį laikotarpį, o baigiamas po. Taip pat pradamos kaupiti meteorologiniai nustatyto regiono duomenys, tokie kaip: Oro temperatūra, vėjo kryptis ir stiprumas, bangavimo kryptis ir stiprumas.
3. Įeinančių duomenų vektorių sudarymas, apjungiant AIS geografinius duomenis, laivo techninius parametrus ir meteorologinius duomenis. Atliekamas duomenų filtravimas pašalinant pasikartojančius duomenis ir griežtai apsiribojant tiriamajame regione.
4. Atrenkami dviejų laivų eismo trajektorijos ir tikrinama ar SOM neuronai išsidėsto pagal įeinančių duomenų vektorius. Rezultatas vizualizuojamas.
5. Patikrinama, ar vyksta SOM tinklų sintezė. Rezultatas vizualizuojamas.
6. Tiriamoji AIS didžiųjų duomenų imtis atsitiktinai suskirstoma į dvi grupes: 90% - mokymo aibė, 10% - testavimo aibė.
7. Testavimo aibėje sužymimi duomenys pagal klases, kurie priklauso įprastam eismui ir kurie neįprastam. Žymėjimas atliekamas pasitelkus Klaipėdos valstybinio jūrų uosto Saugios laivybos administratoriaus vyresniojo specialisto Ramūno Šurkaus pagalba.
8. Prototipui nustatoma, kad modelį apmokyti, naudodamas mokymo duomenų aibę
9. Prototipui apmokant modelį po kiekvieno 5000 duomens paskaičiuojamas klasifikavimo preciziškumas. Klasifikavimo preciziškumas skaičiuojamas, naudojant testavimo aibę.
10. Klasifikavimo preciziškumo kitimas vaizduojamas grafike, kuriame nustatomas mokymosi ir adaptavimosi intervalai.
11. Naudojant apmokyto modelio SOM tinkle esančių virtualaus feromono žymių reikšmes, sudaromas šių reikšmių pasiskirstymo neuronuose histograma.

12. Keičiant VFIN (virtualaus feromono intensyvumas neurone) anomalijos slenksčio reikšmę, sudaroma ROC (angl.: receiver operating characteristic) kreivė.

3.2 Tiriamų duomenų imtis.

Prototipo verifikavimui ir algoritmo tyrimui buvo pasirinktas vidutinio laivų eismo intensyvumo regionas (28 pav.).



28 pav. Tiriamasis regionas laivų eismo intensyvumo žemėlapyje

Duomenų šaltiniai:

AIS laivų eismo duomenys – Marinetrafic.com

Meteorologiniai duomenys – Lietuvos meteorologijos tarnyba

Geografinis regionas:

Klaipėdos valstybinis jūrų uostas ir jo prieigos

Geografinio regiono koordinatės:

55°04'N - 56°45'N

19°48'E - 22°29'N

Laikotarpis:

2016-01-01 – 2016-02-29

Įėjimo vektorių skaičius:

50 mln. įrašų

Įėjimo vektoriaus elementų skaičius:

78 vnt.

Duomenų talpa diske:

1,5 TB

3.3 Įeinančių į algoritmą duomenų struktūra

Sujungus AIS geografinės padėties duomenis su laivų techninėmis charakteristikos duomenimis ir meteorologiniais duomenimis buvo sudarytas SOM tinklo įėjimo vektorius. Šio vektoriaus sudėtis pateikta lentelėje (4 lentelė). Kadangi SOM tinklas negali apmokymui priimti kategorinių duomenų tipų, tokie laukai kaip laivo tipas ir būseną išskirstyti į atskirus laukus kiekvienai kategorijai. Atitikimas kategorijai išreiškiamas loginiu vienetu.

4 lentelė. Įeinančio į SOM tinklą vektoriaus matmenų duomenų struktūra

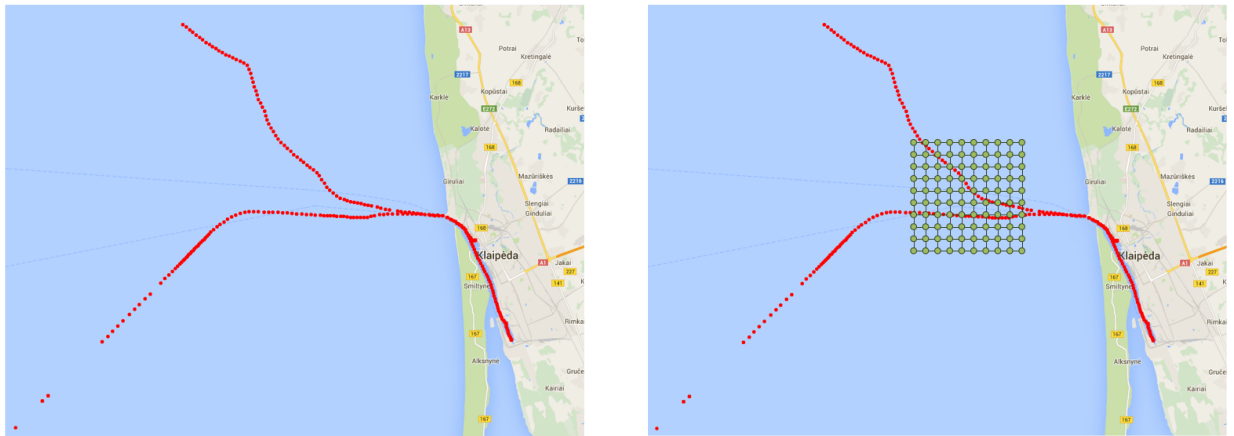
Eilė numeris	Duomenų lauko pavadinimas	Lauko paskirtis	Tipas
1	Longitude	Geografinė Ilguma	float
2	Latitude	Geografinė Platuma	float
3	Heading	Plaukimo kryptis	integer
4	Length	Laivo ilgis	integer
5	Width	Laivo ilgis	integer
6	Tonnage	Laivo tonažas	integer
7	Speed	Laivo plaukimo greitis	integer
8	Wind direction	Vėjo kryptis	Integer
9	Wind speed	Vėjo stiprumas	Integer
10	Waves direction	Bangavimo kryptis	Integer
11	Waves strength	Bangavimo stiprumas	Integer
12	Status: moored	Būsena: prisišvartavęs	bit
13	Status: on the way	Būsena: Kelyje	bit
14	Status: waiting	Būsena: Laukimas	bit
	-----*	-----*	
34	Ship type: Cargo	Laivo tipas: Krovininis	bit
35	Ship type: Passenger	Laivo tipas: Keleivinis	bit
38	Ship type: Tanker	Laivo tipas: Tankeris	bit
78	-----*	-----*	bit

*Visa lentelė pateikta priede Nr. 3

3.4 Įprasto eismo modelio apmokymas dviejų laivų trajektorija.

Pradžioje prototipo verifikacija atlikta supaprastintu modeliu. Iš visos apmokymo duomenų aibės buvo paimtos dviejų laivų trajektorijos (29 pav. a). Pagrindinis šios verifikacijos tikslas -

patikrinti kaip SOM neuronai savi-organizuoja neuronų padėtį laivo trajektorijos atžvilgiu. Šiai verifikacijai naudotas SOM tinklas sudarytas iš 100 neuronų (10x10). Pradinė tinklo būseną prieš apmokymą pavaizduota paveikslėlyje (29 pav. b).

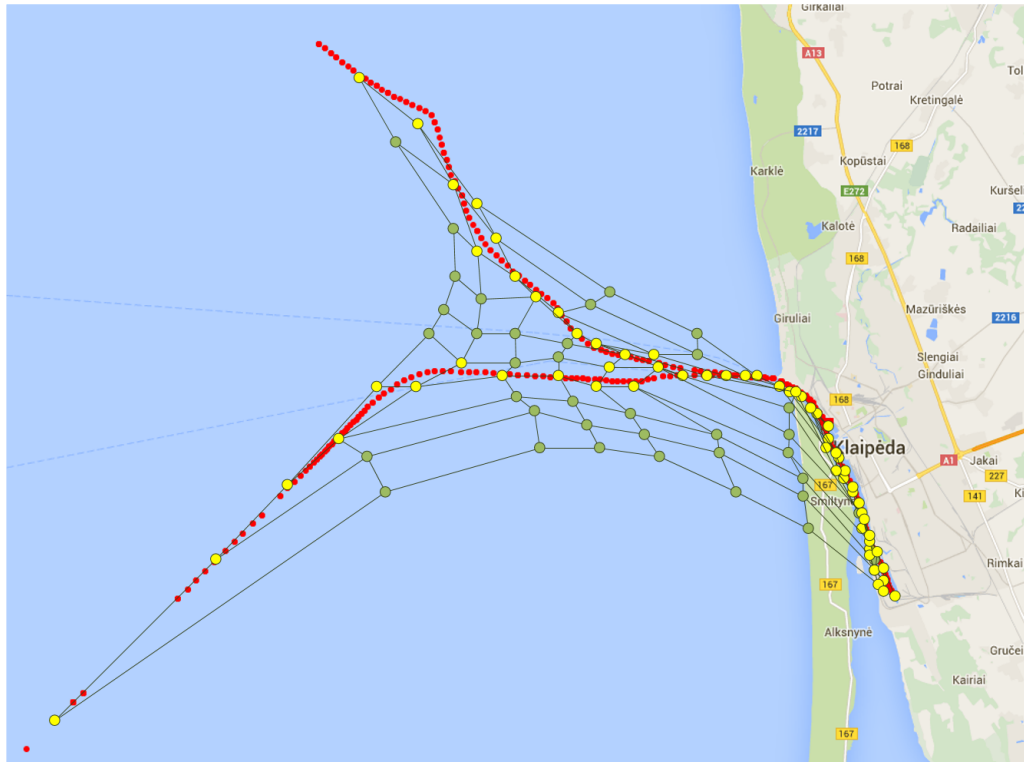


a)

b)

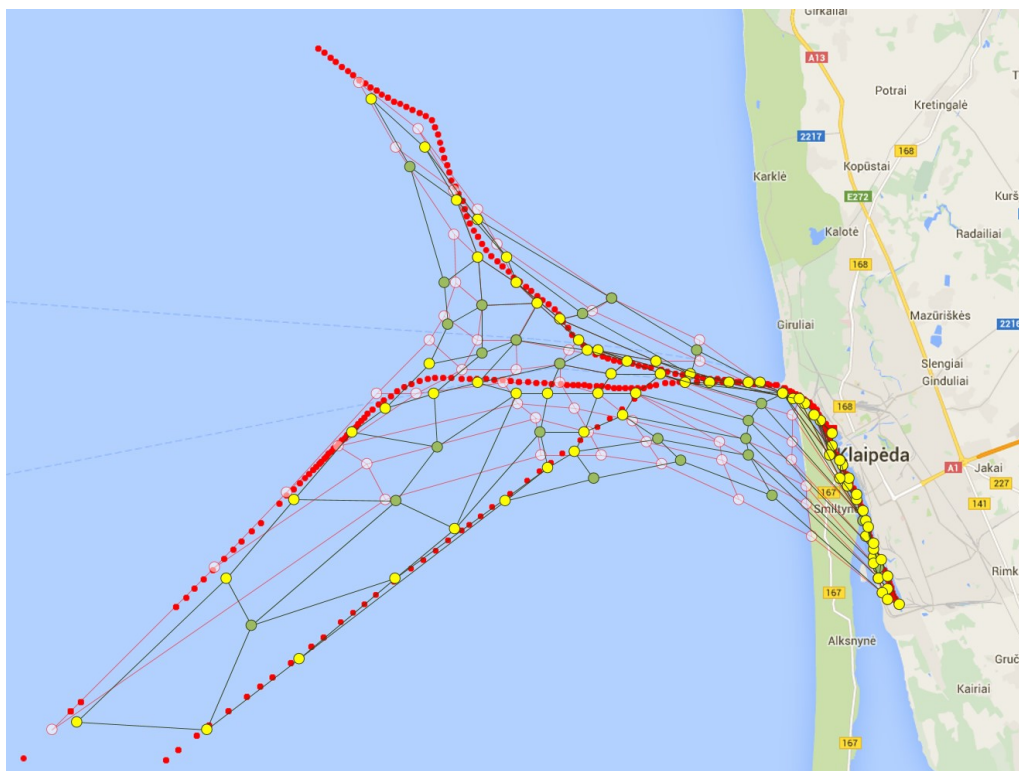
29 pav. Modifikuoto SOM tinklo apmokymo pradžios vizualizacija

Naudojant prototipą šiais duomenimis buvo apmokytas modifikuotas SOM tinklas. Šio apmokyto tinklo vizualizacijoje (30 pav.) matyti, kad tinklo neuronai išsidėstė pagal laivų trajektoriją. Vizualizacijoje geltona spalva pažymėti neuronai, kurių virtualaus feromono intensyvumo žymė didesnė už nulį, o žalios reikšmė lygi nuliui. Feromono reikšmė nelygi nuliui nurodo, kad neuronas turi savo klasterį laivo eismo duomenų vektorių. Vizualizacijoje matome, kad geltoni neuronai išsidėstę visiškai šalia trajektorijos taškų, o žali išsidėstę atokiau. Klasifikuojant sekantį plaukiantį laivą, atliekame priskyrimą prie neurono. Sekantis laivas plaukia tokia pačia trajektorija, tokiu pačiu greičiu toje atkarpoje ir pagal kitus vektorių matavimus, jis patenka į tuos pačius „geltonus“ feromono intensyvumą turinčius neuronus. Pagal tokį neuronų išsidėstymą galime spręsti, kad šio supaprastinto modelio apmokymas vyksta sėkmingai.



30 pav. Apmokyto SOM tinklo dviejų laivų trajektorijos duomenimis vizualizacija

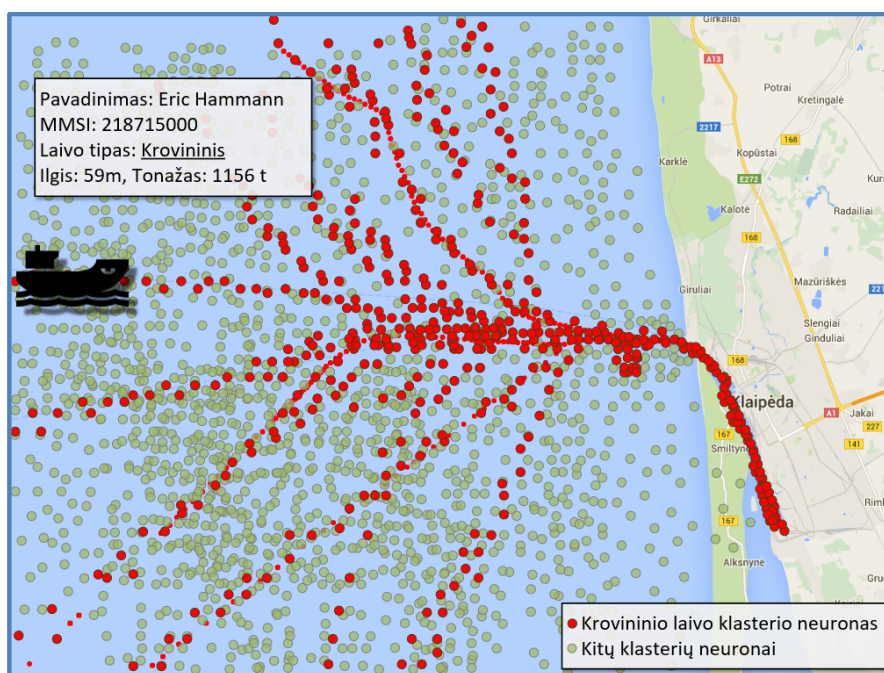
Sekantis svarbus žingsnis tai patikrinti kaip vyksta dviejų SOM tinklų sintezė. Pridėjus dar vieno laivo trajektoriją apmokomas naujas SOM neuronas. Atlikus neuronų sintezę vizualizuojamas šio suliejimo rezultatas (31 pav.). Ryškiomis spalvomis pažymėtas naujai po sintezės suformuotas tinklas, blankiomis - prieš tai buvęs SOM tinklas. Matome, kad „geltoni“ neuronai persistūmė link naujai atsiradusios laivo trajektorijos. Pagal tai matome, kad tinklų sintezė vyksta sėkmingai.



31 pav. Modelio ir naujai sukurto SOM tinklų suliejimo vizualizacija.

3.5 Visa testavimo duomenų aibe apmokyto SOM tinklo vizualizavimas

Apmokyto visa didžiųjų duomenų testavimo aibe SOM tinklo struktūrai buvo panaudota 100x100 (viso 10000) neuronų stačiakampė konfigūracija. Paveikslėlyje (32 pav.) pavaizduota dalis apmokyto SOM tinklo neuronų. Raudona spalva išskirti neuronai, kurių vektoriaus matmens „Laivo tipas: Krovininis“ reikšmė yra daugiau 50% maksimalios reikšmės ir VFIN (virtualaus feromono intensyvumas neurone) reikšmė daugiau 0,5%. Į šią išskirtą grupę patenka neuronai, kurie klasifikuoja įprastą krovininio laivo palaukimo trajektoriją. Krovininiai laivai plaukdamie pagal šių neuronų išsidėstymą klasifikuojami kaip įprastiniai. Paveikslėlyje matomi laivų keliai link Karlshamno, Stokholmo ir kitų didžiųjų uostų. Ties „uosto vartais“ susiformavęs kelio išsikišimas, kuriame laivai įprastai laukia leidimo įplaukti į uostą. Panaudojant laivų įprasto eismo modelį galime prognozuoti laivo plaukymo kryptį, greitį pagal laivo tipą ar oro sąlygas.



32 pav. Krovinio laivo įprastinį judėjimą klasifikuojančių neuronų vizualizacija

3.6 Algoritmo klasifikavimo savybių tyrimas

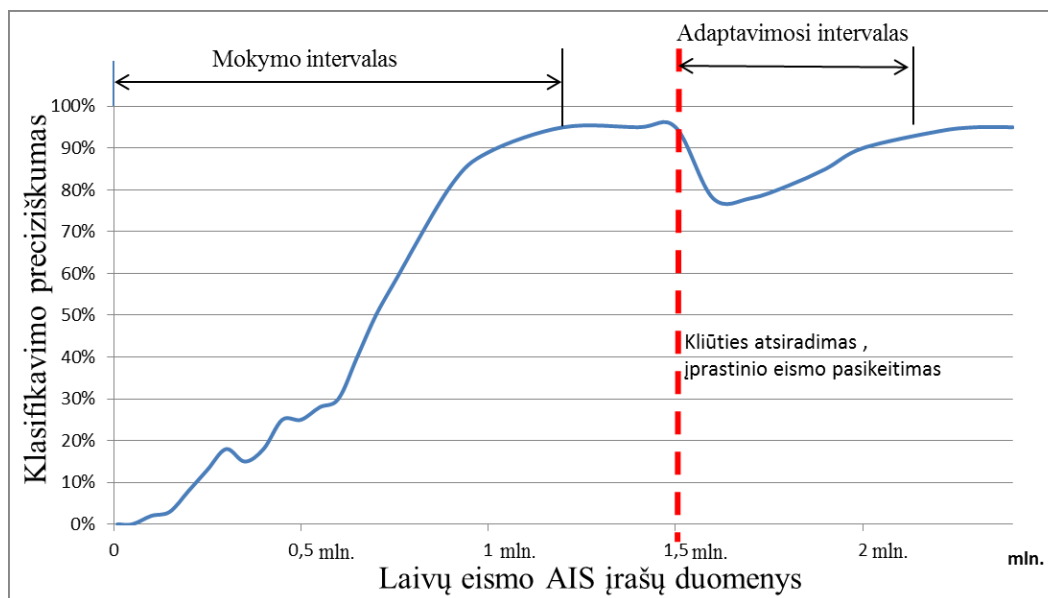
Visa tiriamoji didžiųjų duomenų imtis suskirstyta į dvi imtis: mokymo aibė, testavimo aibė (). Mokymo aibė skirta modelio apmokymui ir anomalijos slenksčio parinkimui. Testavimo aibė skirta klasifikavimo rodiklių skaičiavimui ir tikrinimui. Toks mokymo ir testavimo duomenų atskyrimas atliekamas, kad būtų tikrinami algoritmo rodikliai universalios paklaidos atžvilgiu. (angl.: generalization error) [2].

5 lentelė. Įeinančių duomenų imties paskirstymas

<i>Duomenų imties aibės pavadinimas</i>	<i>Santykis su visa imtimi</i>	<i>Vektorių skaičius aibėje</i>	<i>Neįprasto eismo vektorių santykis aibėje, kuriuos nustatė ekspertas</i>
Mokymo aibė	90%	$45 \cdot 10^6$	--
Testavimo aibė	10%	$5 \cdot 10^6$	0,5%

Norint patikrinti, ar algoritmas apsimoko, buvo vadovautasi Tomo Mitchelo mašininio mokymo sąvoka, kad algoritmas mokosi tada, kai jo užduoties rodikliai gerėja priklausomai nuo sukauptos patirties [2]. Šiuo atveju patirtį atitinka AIS didieji duomenys, o užduoties rodikliu pasirinktas klasifikavimo preciziškumas, nes klasifikavimo tikslumas dėl išbalansuotos duomenų imties klaidingai charakterizuotų klasifikavimą.

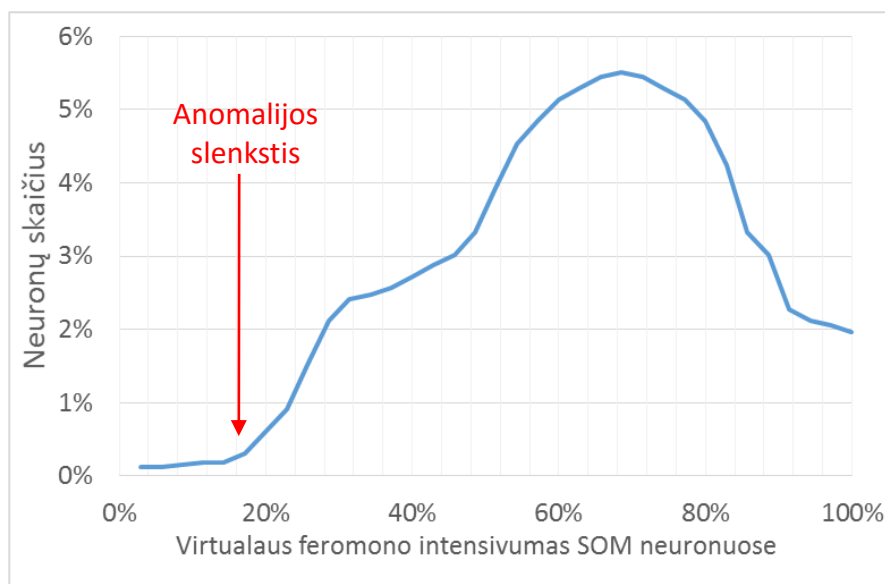
Sudarant klasifikavimo preciziškumo kitimo nuo mokymo duomenų imties kreivę, prototipas buvo apmokamas skirtingu įeinančių vektorių skaičiumi, didinant 5000 vektorių vienetais. Po to apmokyto modelio klasifikatoriumi tikriname testavimo duomenų aibę. Sulyginus rezultatus su eksperto pateiktais duomenimis, apskaičiuojamas klasifikavimo preciziškumas (formulė 10). Rezultatas pateiktas paveikslėlyje (33 pav.).



33 pav. Klasifikavimo preciziškumo kitimas, priklausomai nuo mokymo duomenų imties

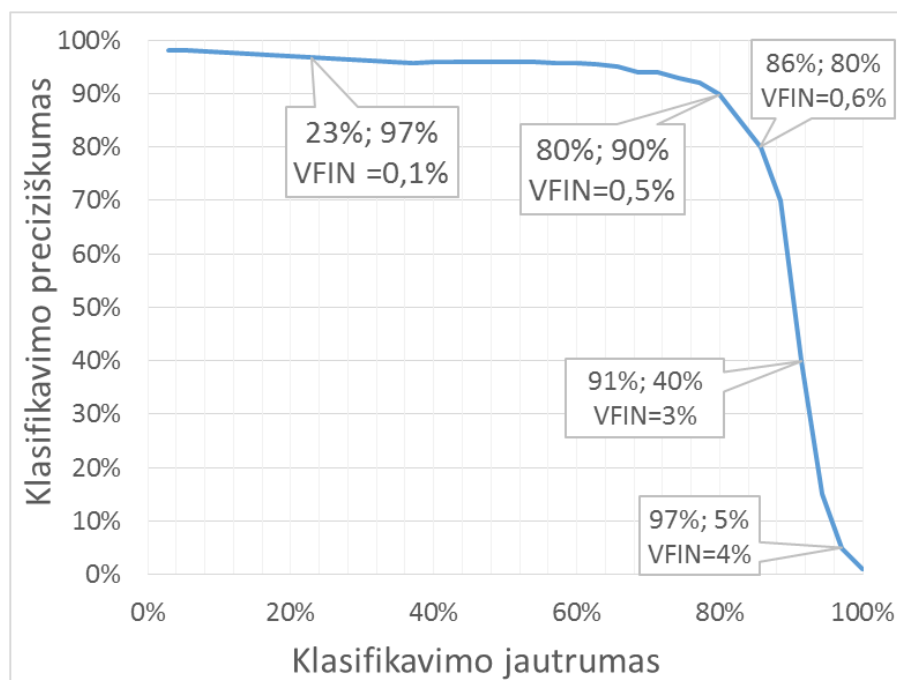
Pagal kitimo kreivę matome, kad pradžioje preciziškumas yra ties 0% reikšme. Daugėjant apsimokymo duomenims kyla ir preciziškumo reikšmė, kuri stabilizuojasi po 1,2 mln. laivų eismo įrašų skaičiaus ties 95%. Toks preciziškumo augimas įrodo, kad algoritmas apsimoko ir mokymosi intervalas šiai duomenų imčiai yra apie 1,2 mln. AIS įrašų (įeinančių duomenų vektorių).

Kreivėje (34 pav.) pavaizduotas įprasto laivų eismo modelio, apmokyto Klaipėdos uosto eismo didžiaisiais duomenimis, virtualaus feromono reikšmių dažnio pasiskirstymas neuronuose (VFIN). Mažesnė feromono reikšmė reiškia retesnę eismą neurono klasteryje. Neįprastą eismą nusako, kiek retas yra konkretaus tipo eismas yra klasteryje. Kadangi VFIN (virtualaus feromono intensyvumas neurone) reikšmė yra tiesiogiai susijusi su eismo dažniu, tai, nustatius slenkstį (anomalijos slenkstis) galime pagal šią reikšmę klasifikuoti įprastą ar neįprastą eismą. Keičiant šią reikšmę, galime sudaryti klasifikavimo preciziškumo ir jautrumo kitimo kreivę (ROC – angl.: receiver operating characteristic).



34 pav. Virtualaus feromono intensyvumo reikšmių dažnis neuronuose

Keičiant anomalijos slenkstį ir matuojant apmokyto modelio klasifikavimo preciziškumą ir jautrumą testavimo aibėje buvo sudaryta ROC kreivė (35 pav.). Klasifikavimo preciziškumas ir jautrumas Kreivė parodo, kad mažinant VFIN reikšmę auga, klasifikavimo preciziškumas, o jautrumo reikšmė krenta. Šios kreivės pagalba galime parinkti klasifikavimo slenkščio reikšmę pagal reikiamą jautrumo ir preciziškumo reikšmių santykį.



35 pav. Neįprasto eismo klasifikavimo ROC kreivė

Tolimesniam tyrimui buvo pasirinkta VFIN=0,5% reikšmė, kuriai esant klasifikavimo jautrumas yra 80%, o preciziškumas 90%.

IŠVADOS

1. Išanalizavus mokslinę literatūrą esamų neįprasto laivų eismo aptikimo algoritmų tema, buvo pasirinkti vertinimo kriterijai tobulintino metodo atrankai. Pasirinkti kriterijai: neprižiūrimasis mokymasis, klasifikavimo preciziškumas ir jautrumas, algoritmo laiko sudėtingumas, mokymasis daugiamačiais vektoriais, veikimas ne Euklido erdvėje bei galimybė adaptuoti tiesioginiam mokymuisi didžiaisiais duomenimis. Pagal vertinimo kriterijų suminį svorinį balą buvo pasirinktas tobulinimui SOM Kohoneno tinklas. SOM pritaikymui dirbti su didžiaisiais AIS duomenimis pasirinkti virtualaus feromono ir tiesioginio mašininio mokymo metodai.
2. Pasirinktas SOM Kohoneno tinklas pritaikymui dirbti su didžiaisiais AIS duomenimis. Atlikti šie SOM pakeitimai: Į SOM tinklą integruotos virtualaus feromono žymės, sudaryta funkcija atstumui skaičiuoti tarp AIS ne Euklido erdvės vektoriams, SOM tinklo mokymasis pritaikytas tiesioginiam mašininiam mokymui naudojant SOM tinklų sintezę. Modifikuotu SOM pagrindu sukurtas didžiaisiais duomenimis apsimokantis algoritmas. Algoritmui sudaryti panaudoti išskirstytų skaičiuojamųjų SQL užklausų, modifikuoto SOM, virtualaus feromono, SOM tinklų sintezės metodai. Šio algoritmo pagrindu sukurtas neįprasto laivų eismo aptikimo posistemis apsimokantis didžiaisiais AIS duomenimis ir integruotas į laivų eismo stebėjimo ir kontrolės sistemą, kurioje naudoja eismo kontrolės AIS didžiuosius duomenis. Sukurtas algoritmas ir posistemis, kurie realizuoti sukuriant prototipą. Prototipui sukurti panaudotas MS IIS7 WEB serveris, MS-SQL-2014 serveris. Prototipas suprogramuotas C#, JS, T-SQL, HTML, CSS programavimo kalbomis. Rezultatams vizualizuoti panaudota Google Maps API biblioteka. Prototipo prijungimui prie didžiųjų duomenų sukurtas tarpinis posistemis sudarytas iš 3 vnt. MS-SQL serverių ir 2vnt. MS IIS7 WEB serverių.
3. Verifikuojant sukurtą prototipą buvo naudojami Klaipėdos valstybinio jūrų uosto laivų eismo AIS ir meteorologiniai duomenys. Verifikuojant nustatyta, kad algoritmas apsimoko AIS didžiaisiais duomenimis ir pasiekia 95% klasifikavimo preciziškumą. Verifikavimo metu nustatyta, kad prototipas apsimoko didžiaisiais AIS duomenimis ir naudojant 0,5% VFIN reikšmę pasiekia 90% preciziškumą esant 80% jautrumui. Ištirtas algoritmo jautrumo ir preciziškumo kitimas (ROC kreivė), keičiant anomalijos slenksčio reikšmę. Apskaičiuotas algoritmo laiko sudėtingumas, kuris yra $\theta(n \cdot m)$, pradiniam apmokymui ir $\theta(m^2)$, duomenų atnaujinimui ir $\theta(n \cdot m) \gg \theta(m^2)$. Čia n – didžiųjų duomenų kiekis, m – naujų duomenų kiekis ir $m \ll n$ – atsinaujinantys duomenys ženkliai mažesni už visą didžiųjų duomenų imtį.

Tolimesni darbai:

Ateityje planuojama tęsti tyrimą, kurio metų būtų tirama:

- duomenų imties statistinių rodiklių įtaka anomalijos slenksčiui,
- SOM tinklo dydžio įtaka preciziškumui ir jautrumui,
- laivo trajektorijos prognozavimas,
- tikėtino laivų susidūrimo aptikimas,
- meteorologinių sąlygų įtaka laivų eismo modeliui,
- pridėti į įeinančius duomenis atstumą tarp laivų ir tirti laivų tarpusavyje susijusį eismą,
- panaudoti gilumini mokymą laivų šablonų grupių sudarymui,
- kito transporto rūšių eismo modelių sudarymas.

LITERATŪRA

- [1] S. Dasgupta, C. Papadimitriou ir U. Vazirani, Algorithms, New York: McGraw-Hill, 2008.
- [2] T. M. Mitchell, Machine Learning, International Edition 1997 mont., McGraw-Hill, 1997.
- [3] D. Laney, „3D Data Management: Controlling Data Volume, Velocity and Variety,“ *Application Delivery Strategies*, Gartner, 2001.
- [4] G. Dzemyda, O. Kurasova ir J. Žilinskas, Daugiamačių duomenų vizualizavimo metodai, Vilnius: Mokslo laida, 2008.
- [5] E. Gaigalas, Algebros užduotys ir rekomendacijos, Vilnius: Vilniaus Univrsitetas, 1992.
- [6] Lietuvos respublikos susisiekimo ministro įsakymas, *Dėl Lietuvos respublikos vidaus vandenų transporto priemonių avarijų ir incidentų tyrimo taisyklių patvirtinimo*, Vilnius, 2013.
- [7] C. Liang, L. Zhengjiang ir Oyang, „Research on the Evolution of Maritime Safety based on the Theory of Selforganization,“ 2013.
- [8] Klaipėdos valstybinio jūrų uosto direkcija, „Klaipėdos uosto, Būtingės terminalo ir kitų Baltijos Jūros rytinės pakrantės uostų krovos apžvalga,“ Klaipėdos valstybinio jūrų uosto direkcija, Klaipėda, 2016.
- [9] Y. Wen, Y. Huang, C. Zhou, J. Yang, C. Xiao ir X. Wu, „Modelling of marinetraffic flow complexity,“ *Ocean Engineering*, nr. 104, pp. 500-510, 2015.
- [10] C. D. Wickens ir J. G. Hollands, Engineering Psychology and Human Performance 3rd ed., Prentice Hall, 1999.
- [11] International Maritime Organisation (IMO), *Guidelines for the Onboard Operational Use of Shipborne Automatic Identification Systems (AIS), as Amended by ITU-1371. Resolution A.917(22).*, London: International Maritime Organisation (IMO), 2001.
- [12] H. Shenping ir Z. J. Jinpeng, „Risk assessment of marine traffic safety at coastal water area,“ *Procedia Engineering*, t. 45, pp. 31-37, 2012.
- [13] L. Perera, P. Oliveira ir C. S. Guedes, „Perera, L.P., Oliveira, P.; Guedes Soares, C. Maritime Traffic Monitoring Based on Vessel Detection, Tracking, State Estimation, and Trajectory Prediction,“ *Intelligent Transportation Systems, IEEE Transactions*, t. 13, nr. 3, p. 1188 – 1190, 2012.
- [14] K. S. O. Tan ir S. S. Tng, „An Integrated Maritime Reasoning and Monitoring System,“ įtraukta *Information Fusion (FUSION), 2012 15th International Conference*, 2012.
- [15] M. Riveiro ir G. Falkman, „Interactive Visualization of Normal Behavioral Models and Expert Rules for Maritime Anomaly Detection,“ įtraukta *2009 Sixth International Conference on Computer Graphics, Imaging and Visualization*, IEEE, 2009.
- [16] K. Kowalska ir L. Peel, „Maritime Anomaly Detection using Gaussian Process Active Learning,“ įtraukta *Information Fusion (FUSION), 2012 15th International Conference*, IEEE, 2012.
- [17] E. Osekowska ir S. Axelsson, „Potential Fields in Modeling Transport over Water,“ *Operations Research*, t. 58, pp. 259-280, 2015.

- [18] T. Kohonen, „Self-Organizing Maps,“ *Springer Series in Information Science*, t. 30, 2001.
- [19] J. Venskus, M. Kurmis, A. Andziulis ir Ž. Lukošius, „Self-Learning Adaptive Algorithm For Maritime Traffic Abnormal Movement Detection Based On Virtual Pheromone Method,“ įtraukta *2015 International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, Chicago, IL, USA, 2015.
- [20] R. W. Sinnott, „Virtues of the Haversine,“ *Sky and Telescope*, t. 68, nr. 2, p. 159, 1984.
- [21] G. Praetorius, E. Hollnagel ir J. Dahlman, „Modelling VesselTraffic Servicetounderstandresilience,“ *ReliabilityEngineeringandSystemSafety*, t. 141, pp. 10-21, 2015.
- [22] M. Vespe, H. Greidanus ir M. A. Alvarez, „The declining impact of piracy on maritime transport in the Indian Ocean: Statistical analysis of 5-year vessel tracking data,“ *Marine Policy*, pp. 50-60, 2015.
- [23] Marine Accident Investigation Branch, „Annual Report,“ Marine Accident Investigation Branch, Southampton, UK, 2012.
- [24] C. Baker ir A. Seah, „Maritime Accidents and Human Performance: The Statistical Trial,“ įtraukta *MARTECH 2004*, Singapore, 2004.
- [25] B. Idiri ir A. Napoli, „The automatic indentification system of marine accident risk using rule-based reasoning,“ įtraukta *System of Systems Engineering (SoSE), 2012 7th International Conference*, Genoa , 2012.
- [26] J. Zhang, T. A. Santos, C. G. Soares ir X. Yan, „Sequential ship traffic scheduling model for restricted two-way waterway transportation,“ *Engineering for the Maritime Environment*, %1 t. iš %2I-II, 2015.
- [27] J. Verstichel, P. De Causmaecker ir F. Spieksma, „Exact and heuristic methods for placing ships in locks,“ *Eur J Oper Res*, t. 235, nr. 2, pp. 387-398, 2014.
- [28] A. Patcha ir J.-M. Park, „An overview of anomaly detection techniques: Existing solutions and latest technological trends,“ *Computer Networks*, t. 51, p. 3448–3470, 2007.
- [29] F. MAIORANA, „Performance improvements of a Kohonen self organizing classification algorithm on sparse data sets,“ *MATHEMATICAL METHODS, COMPUTATIONAL TECHNIQUES, NON-LINEAR SYSTEMS, INTELLIGENT SYSTEMS*, pp. 347-352, 2014.
- [30] P. Kujala, M. Hanninen, T. Arola ir J. Ylitalo, „Analysis of the marine traffic safety in the Gulf of Finland,“ *Reliability Engineering and System Safety*, t. 94, pp. 1349-1357, 2009.
- [31] E. Osekowska, H. Johnson ir B. Carlsson, „Grid size optimization for potential field based maritime anomaly detection,“ *Transportation Research Procedia*, t. 3, pp. 720-729, 2014.
- [32] C.-H. Chen, L. P. Khoo, Y. T. Chong ir X. F. Yin, „Knowledge discovery using genetic algorithm for maritime situational awareness,“ *Expert Systems with Applications*, t. 41, pp. 2742-2753, 2014.
- [33] S. Kazemi, S. Abghari, N. Lavesson, H. Johnson ir P. Ryman, „Open data for anomaly detection in maritime surveillance,“ *Expert Systems with Applications*, t. 40, pp. 5719-5729, 2013.
- [34] Y. Zhao, W. Li ir P. Shi, „A real-time collision avoidance learning system for Unmanned Surface Vessels,“ *Neurocomputing*, t. 182, pp. 255-266, 2016.

- [35] S. Mascaroa, A. Nicholson ir K. Korb, „Anomaly detection in vessel tracks using Bayesian networks,“ *International Journal of Approximate Reasoning*, t. 55, pp. 84-98, 2014.
- [36] J. Larson, M. Bruch, R. Halterman, J. Rogers ir R. Webster, „Advances in Autonomous Obstacle Avoidance for Unmanned Surface Vehicles,“ *Space and Naval Warfare Systems Center*, 2007.
- [37] R. Laxhammar, G. Falkman ir E. Sviestins, „Anomaly detection in sea traffic—a comparison of the Gaussian Mixture Model and the Kernel Density Estimator,“ *Proceedings of the 12th IEEE International Conference on Information Fusion*, pp. 759-763, 2009.
- [38] T. Helldin ir M. Riveiro, „Explanation methods for Bayesian networks: Review and application to a maritime scenario,“ *Proceedings of the 3rd Annual Skövde Workshop on Information Fusion Topics*, pp. 11-19, 2009.
- [39] A. Cansado ir A. Soto, „Unsupervised anomaly detection in large databases using Bayesian networks,“ *Applied Artificial Intelligence*, t. 22, 2008.
- [40] C. Loy, T. Xiang ir S. Gong, „Detecting and discriminating behavioural anomalies,“ *Pattern Recognition*, t. 44, pp. 117-132, 2010.
- [41] „Evaluating the usability of visualizations of normal behavioral models for analytical reasoning,“ *Computer Graphics, Imaging and Visualization (CGIV)*, pp. 179-185, 2010.
- [42] R. Laxhammar, „Anomaly detection for sea surveillance,“ *The 11th International Conference on Information Fusion*, p. 55–62.
- [43] UAB „SAVVIN“, „Lietuvos jūrinio sektoriaus vystymo galimybių studija,“ Asociacija „Baltijos slėnis“, Klaipėda, 2011.
- [44] C. Brax, A. Karlsson, S. F. Andler, R. Johansson ir L. Niklasson, „Evaluating precise and imprecise state-based anomaly detectors for maritime surveillance,“ įtraukta *Proceedings of the 13th Conference on Information Fusion (FUSION)*, 2010.
- [45] R. Laxhammar ir G. Falkman, „Sequential conformal anomaly detection in trajectories based on hausdorff distance,“ įtraukta *Proceedings of the 14th International Conference on Information*, 2011.
- [46] B. Ristic, B. L. Scala, M. Morelande ir N. Gordon, „Statistical analysis of motion patterns in AIS data: Anomaly detection and motion prediction,“ įtraukta *Proceedings of the 11th International Conference on Information Fusion*, 2008.