VILNIAUS UNIVERSITETAS
INFORMATIKOS INSTITUTAS
PROGRAMŲ SISTEMŲ KATEDRA

Vilniaus miesto nusikaltimų prognozė

Vilnius City Crime Prediction

Bakalauro baigiamasis darbas

Atliko: Tomaš Monkevič (parašas)

Darbo vadovas: asist. dr. Vytautas Valaitis (parašas)

Darbo recenzentas: j. asist. Linas Petkevičius (parašas)

Vilnius, 2019

Santrauka

Šiame darbe buvo prognozuojamos Vilniaus miesto nusikaltimo vietos kitai dienai, pasinaudojant ilgos trumpalaikės atminties modeliu ir praėjusių 4 metų (2015-2018) nusikaltimų žmogaus sveikatai istorija. Taip pat, buvo tiriama ar papildomi duomenys apie aplinką padeda pagerinti modelio prognozę. Po modelio apmokymo su skirtingais parametrais, jo prognozės nepasiekė tokio tikslumo kaip kituose šaltiniuose, nepaisant to, ar buvo naudojami papildomi aplinkos duomenys, ar ne. Vis dėlto, dažniausiai modelio prognozės yra labai arti celės, kurioje išties buvo įvykdytas nusikaltimas. Todėl atsižvelgiant į tai, galima daryti prielaidą, kad tokios prognozės patyrusiems policininkams galėtų suteikti naujų įžvalgų ir galbūt padėtų policijos pareigūnams sudarinėti patruliavimo maršrutus.

Raktiniai žodžiai: nusikaltimų prognozė, ilgos trumpalaikės atminties modelis, rekurentiniai neuroniniai tinklai.

Summary

In this work, Vilnius city crime sites were predicted for the next day, using the long short-term memory model and the history of crimes against human health of the past 4 years (2015-2018). Also, additional research was done to see if using additional environmental data for model training would improve the model prediction results. When the model was trained with different parameters, its predictions did not reach the accuracy as in other sources, regardless of whether additional environmental data was used or not. However, most of the model predictions are very close to the cell where the crime was actually committed. Therefore, experienced police officers could use such predictions to gain new insights and could possibly help them make patrolling routes.

Keywords: crime prediction, long short-term memory (LSTM), recurrent neural networks

TURINYS

ĮVADAS

Šiais laikais dauguma žmonių turi lengvą prieigą prie interneto, o tai reiškia lengvą prieigą prie daugybės informacijos. Tai įgalina žmogų savarankiškai mokytis, dalintis savo nuomone ir bendradarbiauti su kitais. Būtent todėl šiuo metu yra labai daug atviro kodo (*angl. open source*) projektų ir atvirų duomenų (*angl. open data*). Atviri duomenys yra ypač svarbūs kalbant apie valdžią, nes tai atveria galimybę pagerinti šalies ekonomiką ir padėti piliečiams [KC14, JCZ12]. Lietuvos Respublika neatsilieka nuo pasaulio ir taip pat viešina įvairius duomenis savo piliečiams. Pavyzdžiui, Vilniaus miesto savivaldybė viešina tokius duomenis kaip: dviračių takų, viešųjų tualetų, atvirų sporto salių žemėlapius ir t.t. Tačiau šiam darbui svarbiausi duomenys, kuriuos paviešino Lietuvos Respublikos valdžia yra nusikaltimų žemėlapis, kuris parodo nusikaltimus nuo 2010 metų ir kiekvieną dieną yra atnaujinamas. Turint tiek daug duomenų apie Lietuvos nusikaltimus, galima sukurti dirbtinio neuroninio tinklo modelį, kuris, manytina, gebėtų prognozuoti nusikaltimų vietas.

Panašus tyrimas buvo atliktas naudojant Amsterdamo miesto nusikaltimų duomenis [RHP17]. Naudojant daugiasluoksnį perceptroną (*angl. multilayer perceptron*) su vienu paslėptu būsenų vektorių sluoksniu, kai duomenys buvo suskaidyti į dienos ir nakties nusikaltimus mėnesiniu periodu, gebėjo prognozuoti nusikaltimų vietas su tiesioginiu pataikymu virš 50% (t. y. teisingai prognozuotų nusikaltimų vietų procentinė dalis) ir tikslumu virš 50% (t. y. teisingų prognozių procentinė dalis, palyginus su bendru spėjimų skaičiumi). Daugiasluoksnio perceptrono modelyje kiekvieno perceptrono išeitis yra perduodama tik į aukštesnio lygio būsenų vektorius, toks dirbtinio neuroninio tinklo modelis vadinamas tiesioginio perdavimo (*angl. feedforward*). Kitais žodžiais, tokio modelio sprendimo neįtakoja prieš tai padaryti sprendimai, todėl toks modelis nėra tinkamiausias mėginant prognozuoti nusikaltimus, kadangi nusikaltimai priklauso nuo prieš tai buvusių nusikaltimų [BJ05, 75]. Būtent šią problemą sprendžia rekurentiniai neuroniniai tinklai (*angl. recurrent neural networks*).

Rekurentiniai neuroniniai tinklai yra panašūs savo struktūra į daugiasluoksnio perceptrono modelį, tačiau rekurentiniuosie tinkluose paslėptų būsenų vektorių išeitis yra perduodama ne tik į aukštesnio lygio būsenų vektorius, bet dar yra išsaugoma ir bus naudojama kitoje iteracijoje. Tokiu būdu, rekurentinio neuroninio tinklo sprendimą įtakoja prieš tai daromi sprendimai, tačiau toks rekurentinis modelis, kuo daugiau yra apmokomas, tuo daugiau jis pamiršta senesnius sąryšius [GB10, 253]. Šią problemą sprendžia ilgos trumpalaikės atminties (*angl. long short-term memory*) modelis (toliau – LSTM). LSTM modelis struktūriškai yra toks pats kaip rekurentinis neuroninis tinklas tik perceptronai yra pakeisti į LSTM elementus. LSTM modelis naudojamas spręsti tokius

uždavinius kaip: kalbos atpažinimas, muzikos modeliavimas, rašto atpažinimas ir daugelyje kitų uždavinių, kur yra svarbi duomenų seka.

Nusikaltimuose taip pat svarbi duomenų seka, todėl yra manoma, kad LSTM modelis parodys geresnius rezultatus, nei daugiasluoksnio perceptrono modelis. Toks eksperimentas, kur LSTM modelis buvo naudojamas prognozuoti kitos dienos nusikaltimų vietas buvo atliktas [Cor18] šaltinyje. [Cor18] tyrimo autorius pasinaudojo nusikaltimų duomenimis iš Gvatemalos policijos, kuriuos sugrupavo pagal erdvinę priklausomybę panaudojant „K-Means“ algoritmą. Sugrupuoti nusikaltimų duomenys buvo panaudoti apmokyti skirtingus LSTM modelius su skirtingais parametrais. LSTM modelis išvesdavo kitos dienos nusikaltimų grupės prognozę. [Cor18] tyrimo eigoje buvo nustatyti geriausi parametrai modelio apmokymui. Taip pat, pagal [Cor18] tyrimą buvo nustatyta, kad visi tiriami LSTM modeliai parodė geresnius rezultatus už tradicinius stebėjimo signalo (*angl. tracking signal*) algoritmus ir už mašininio mokymosi algoritmus, tačiau [Cor18] šaltinyje nebuvo naudojami jokie papildomi duomenys. Remiantis [RHP17] šaltiniu, kur buvo naudojami papildomi duomenys, manoma, kad pridėjus papildomus duomenis prie LSTM modelio, bus pagerintas prognozavimo tikslumas.

Atsižvelgiant į aukščiau išdėstytą informaciją, šiame darbe buvo nuspręsta daryti Vilniaus miesto nusikaltimų prognozę, naudojant ilgos trumpalaikės atminties (LSTM) modelį, kuris bus apmokytas remiantis ne tik praeitų nusikaltimų duomenimis, bet taip pat ir papildomais duomenimis. **Šio darbo tikslas yra sukurti ir apmokyti ilgos trumpalaikės atminties (LSTM) modelį, kuris gebės prognozuoti nusikaltimų vietas kitai dienai Vilniuje**.

Šiam tikslui pasiekti buvo iškelti trys uždaviniai:

1. Surinkti reikalingus duomenis ir paruošti juos LSTM modeliui.
2. Pasirinkti LSTM modelio tikslumo vertinimo strategiją.
3. Sukurti ir apmokyti LSTM modelį.

1. DIRBTINIŲ NEURONINIŲ TINKLŲ APŽVALGA

Šiame skyriuje yra išdėstyta teorija, reikalinga skaitant šį darbą. Skyrius prasidės nuo įvado į dirbtinius neuroninius tinklus, jo svarbiausius komponentus. Toliau bus aprašyti rekurentiniai neuroniniai tinklai, jų privalumai ir trūkumai, taip pat nurodoma, kuo jie skiriasi nuo paprastų dirbtinių neuroninių tinklų. Galiausiai bus aprašytas LSTM modelis ir jo privalumai, lyginant su rekurentiniais neuroniniais tinklais.

1.1. Dirbtiniai neuroniniai tinklai

Dirbtiniai neuroniniai tinklai – tai sujungti tarpusavyje įvesties/išvesties vienetai, kur kiekvienas sujungimas turi savo svorį [Ste10]. Šie įvesties/išvesties vienetai yra vadinami neuronais arba perceptronais. Toks matematinis neuronas daugina visas įvestis iš atitinkamo svorio ir skaičiuoja jų sumą, vėliau tą sumą perduoda į aktyvacijos funkciją (*angl. activation function*), o aktyvacijos funkcijos išvestis ir yra neurono išvestis, žr. 1 formulę [JMM96, 34].

$$y = \theta\left(\sum_{j=1}^{n} w_j x_j - u\right) \quad (1)$$

Čia y – neurono išvestis, θ –aktyvacijos funkcija, w_j – j-tasis svoris, x_j – j-toji įvestis, u – polinkis (*angl. bias*) ir n – įvesties skaičius.

Šis matematinis modelis buvo įkvėptas stebint gyvūnų smegenų veikimą. Gyvūnai, prieš kažką darant, iškart turi tai išmokti. Su dirbtiniais neuroniniais tinklais yra panašiai, todėl prieš naudojant neuroninį tinklą, jį iš karto reikia apmokyti. Apmokymo etape pagal duotą įvestį ir norimą išvesties klasę tinklas koreguoja savo sujungimo svorius ir tokiu būdu mokosi išvesti teisingą klasę. Sujungimo svorių koregavimas vyksta panaudojant optimizavimo funkciją (*angl. optimization function*), kartu su nuostolių funkcija (*angl. cost or loss function*). Nuostolių funkcija yra reikalinga tam, kad optimizavimo algoritmas žinotų kokią vertę reikia minimizuoti. Dažnai naudojamos nuostolių funkcijos yra: sigmoidė (*angl. sigmoid*), hiperbolinio tangento funkcija (tanh) ir minkštojo maksimumo (*angl. softmax*) funkcija. Bendras nuostolių funkcijų veikimo principas yra tas, kad ši funkcija išveda skaičių, kuris nusako, kiek stipriai norimas rezultatas skiriasi nuo gautojo [JMM96, 34].

Neuroniniai tinklai gali būti traktuojami kaip orientuotas svorinis grafas, kur neuronai yra viršūnės, o lankai su svoriais būtų neuronų sujungimai. Neuroniniai tinklai gali būti sugrupuoti į dvi skirtingas kategorijas pagal tinklo neuronų sujungimus [JMM96, 34]:

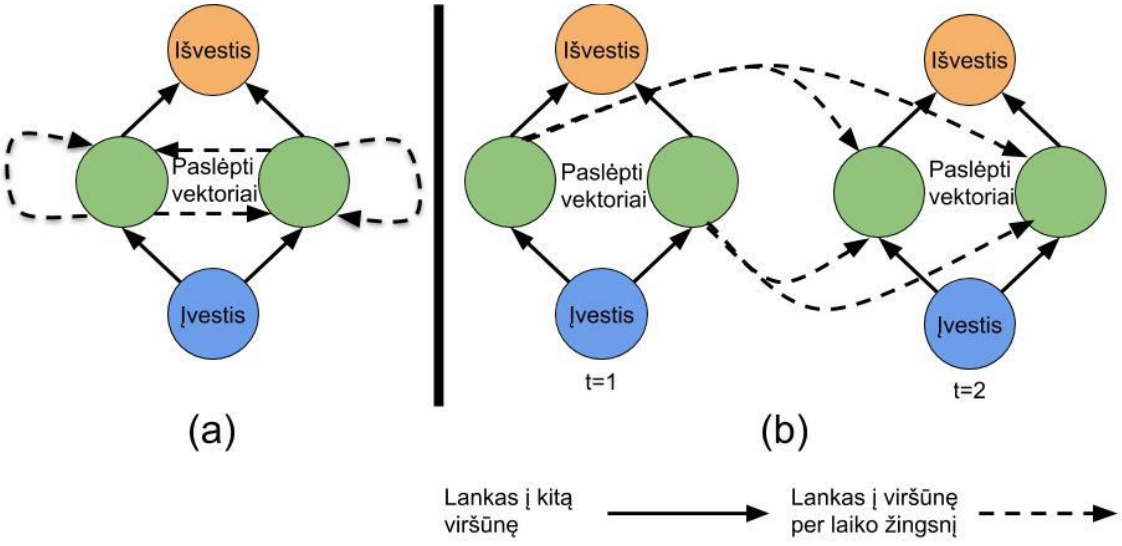
- Tiesioginio perdavimo tinklas, kur grafas neturi jokių ciklų;
- Rekurentinis arba atgalinio perdavimo (*angl. feedback*) tinklas, kur yra ciklai dėl atgalinio perdavimo sujungimų.

Tiesioginio perdavimo tinklas yra statinis, kitaip tariant, jo išvedimas yra viena aibė, o ne vienetų seka duotai įvesčiai. Tiesioginio perdavimo tinklas, neturi atminties, t. y., prieš tai buvusi tinklo būsena neįtakoja kito išvedamo rezultato. Kita vertus, rekurentiniai arba atgalinio perdavimo tinklai yra dinaminiai. Kai yra nauja įvestis pristatoma į neuroną, yra skaičiuojama jo išvestis ir dėl atgalinio perdavimo, kiekvieno neurono įvestis yra pakeičiama, kas lemia tinklo perėjimą į kitą būseną [JMM96, 34].

1.2. **Rekurentiniai neuroniniai tinklai**

Rekurentinio neuroninio tinklo įvestis ir/arba jo išvedimas yra seka. Įvesties seka gali būti žymima $(x^{(1)},x^{(2)},...,x^{(t)})$, kur kiekvienas $x^{(t)}$ yra vektorius ypatybių (*angl. features*) laiko momentų *t*. Panašiai gali būti žymima tinklo išvestis $(y^{(1)},y^{(2)},...,y^{(t)})$ [LBE15, 5].

Rekurentiniai neuroniniai tinklai yra tokie pat, kaip ir tiesioginio perdavimo neuroniniai tinklai, tik su papildomai pridėtais lankais, kurie sujungia gretimus laiko žingsnius (*angl. time steps*). Tokiu būdu yra įvedama laiko sąvoka modeliui. Kaip tiesioginio perdavimo neuroniniai tinklai gali neturėti ciklų tarp paprastų lankų, taip ir rekurentiniai neuroniniai tinklai jų irgi gali neturėti, tačiau lankai kurie sujungia gretimus laiko žingsnius yra vadinami rekurentiniais lankais, kurie gali formuoti ciklus, įskaitant ciklus, kurie yra vienetinio ilgo ir yra sujungti patys su savimi per laiko žingsnius. Laiko žingsnyje *t*, viršūnė su rekurentiniais lankais gauna dabartinę įvestį $x^{(t)}$ ir įvestį iš paslėptos viršūnės $h^{(t-1)}$, kuri atėjo iš buvusios dirbtinio neuroninio tinklo būsenos. Išvestis kiekvienu laiko žingsniu *t* yra skaičiuojama pasinaudojant paslėptos viršūnes išvestimi $h^{(t)}$ laiko momentu *t*. Įvestis $x^{(t-1)}$ laiko žingsnių *t -1* gali įtakoti išvedimo $y^{(t)}$ laiko žingsnių *t* ir vėlesnius rekurentinius lankus [LBE15, 10].



1 pav. (a) – paprastas rekurentinis tinklas, (b) – rekurentinis tinklas išvyniotas laike (parengta pagal [LBE15])

Dirbtinio neuroninio tinklo dinamiškumas parodytas 1.(a) pav. gali būti išvyniotas per laiko žingsnius, kaip yra nurodyta 1.(b) pav. Atsižvelgiant į 1.(b) pav., tinklas gali būti aiškinamas ne kaip ciklinis, bet kaip gilusis tinklas, su vienu sluoksniu per laiko žingsnį, kur svoriai yra dalinami tarp laiko žingsnių. Iš to matyti, kad išvyniotas tinklas gali būti apmokomas per kelis laiko žingsnius, panaudojant sklidimo atgal (*angl. backpropagation*) algoritmą. Toks algoritmas vadinamas sklidimo atgal per laiką (*angl. backpropagation throught time*) algoritmu [LBE15].

Jau seniai yra žinoma, kad neuroninių tinklų apmokymas yra netriviali užduotis. Net standartinių tiesioginio perdavimo neuroninių tinklų optimizavimas yra NP–pilnumo uždavinys [BR93]. Rekurentinių neuroninių tinklų apmokymas yra ypač sudėtingas dėl sunkumo išmokti ilgalaikius sąryšius, kaip yra aprašoma [Hoc01] šaltinyje. Nykstančio ir sprogstančio nuolydžio (*angl. vanishing and expoiding gradient*) problemos atsiranda tada, kuomet atgalinio sklidimo klaidos yra perduodamos per kelis laiko žingsnius [LBE15, 13].

1.3. Ilgos trumpalaikės atminties modelis

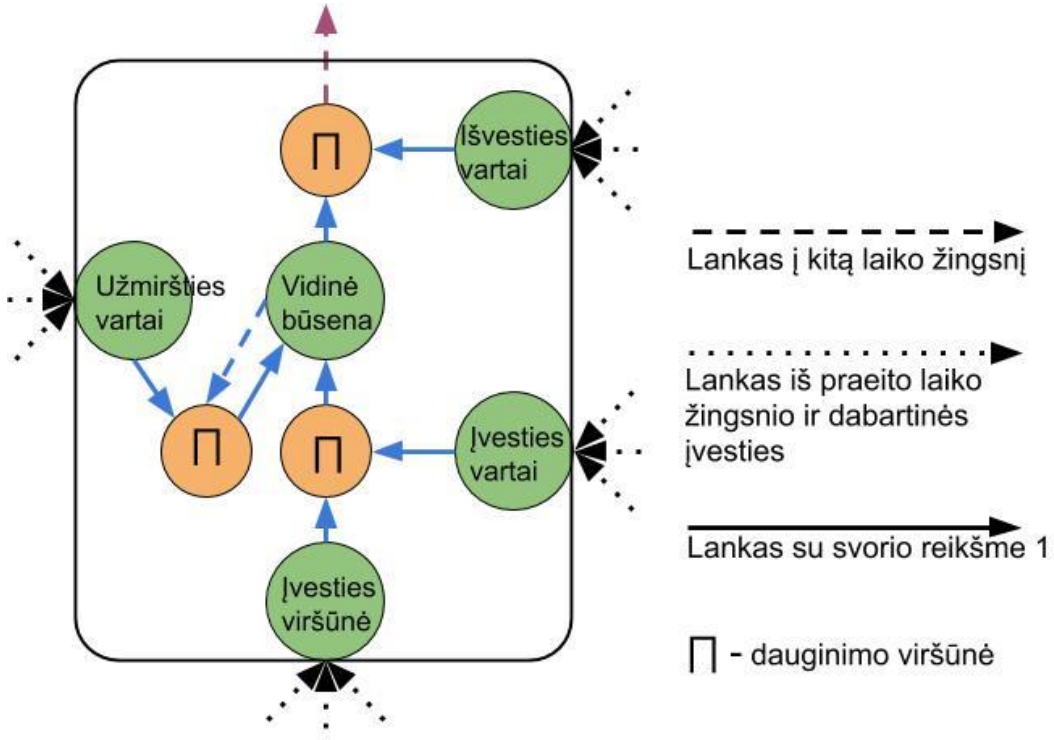
1997 metais buvo pristatytas LSTM modelis iš esmės siekiant išspręsti nykstančio ir sprogstančio nuolydžio problemas. Šis modelis panašus į standartinį rekurentinį neuroninį tinklą su paslėptais vektoriais, tačiau kiekvienas paprastas paslėptas neuronas yra pakeistas į atminties elementą (*angl. memory cell*). Kiekvienas atminties elementas turi neuroną, kuris turi į save patį jungiantį rekurentinį lanką su fiksuotu vienetiniu svoriu, kad užtikrintų, jog nuolydis galėtų pereiti per daugelį laiko žingsnių ir jis nebūtų pradangintas arba išpūstas [LBE15].

Terminas „ilga trumpalaikė atmintis“ kyla iš tolimesnių intuicijų. Paprasti rekurentiniai neuroniniai tinklai turi ilgalaikę atminį svorių pavidalu. Apmokymo metu svoriai yra lėtai keičiami, tokiu būdu užkoduojant bendras žinias apie duomenis. Taip pat, jie turi trumpalaikę atminį labai trumpos aktyvacijos formą, kuri yra perduodama iš vieno neurono į kitą. LSTM atminties elementas yra sudėtinis vienetas, sudarytas ir paprastų neuronų, sujungtų specifiniu būdu. Prie to dar yra įtraukti nauji dauginimo mazgai (*angl. multiplicative nodes*), diagramoje (žr. 2 pav.) žymimi raide Π. Visi LSTM atminties elemento komponentai yra aprašyti apačioje [LBE15].

- Įvesties viršūnė (*angl. input node*) – šis elementas yra žymimas g_c. Tai viršūnė, kuri esamu laiko žingsniu priima aktyvaciją standartiniu būdu iš įvesties sluoksnio $x^{(t)}$ ir iš paslėpto praeito laiko žingsnio $h^{(t-1)}$ sluoksnio [LBE15, 18].
- Įvesties vartai (*angl. input gate*). Vartai yra išskirtinis LSTM modelio bruožas. Vartai yra sigmoidinis elementas, kuris kaip ir įvesties viršūnė, priima aktyvaciją iš dabartinio duomenų taško $x^{(t)}$ ir paslėpto sluoksnio iš praeito laiko žingsnio. Elementas yra vadinamas vartais, nes jeigu įeinanti reikšmė yra nulis, tada srautas nuo kito neurono yra

nepraleistas. Jeigu reikšmė yra vienetas, tai visas srautas yra praleidžiamas toliau. Įvesties vartų reikšmė i_c daugina įvesties viršūnės reikšmę [LBE15, 18].

- Vidinė būsena (*angl. internal state*). Šis neuronas naudoja tiesinę aktyvaciją ir yra žymima s_c. Vidinės būsenos viršūnė turi į pačią save prijungtą rekurentinį lanką su vienetiniu svoriu. Kadangi šis lankas apima gretimus laiko žingsnius su fiksuotais svoriais, klaida gali eiti per laiko žingsnius, be išnykimo ar išsipūtimo. Šis lankas dažnai yra vadinamas fiksuotos klaidos karusele (*angl. constant error carousel*) [LBE15, 18].
- Užmiršties vartai (*angl. forget gate*). Šie vartai yra žymimi f_c. Jie suteikia metodą, kuriuo tinklas gali išmokti išmesti vidinės būsenos reikšmę [LBE15, 18].
- Išvesties vartai (*angl. output gate*). Atminties elemento išeitis v_c yra gaunama dauginant vidinės būsenos reikšmę s_c su išvesties vartų reikšme o_c. Dažniausiai vidinė būsenos reikšmė iš pradžių yra paleista per tangento aktyvacijos funkciją, nes tai suteikia kiekvienai atminties elemento išeičiai tokį patį dinaminį diapazoną, kaip ir paprasto paslėpto neurono, tačiau kiti neuroninio tinklo tyrimai parodė, kad naudojama ReLu aktyvacijos funkcija turi didesnį dinaminį diapazoną ir tokiu būdu LSTM modelis yra lengviau apmokomas, todėl kartais netiesinė funkcija vidinės būsenos viršūnėje yra praleista [LBE15, 18].



2 pav. LSTM atminties elementas su užmiršties vartais (parengta pagal [LBE15])

2. METODOLOGIJA

Šiame skyriuje yra analizuojami panašūs tyrimai ir juose panaudoti metodai. Taip pat, šiame skyriuje yra aprašomi darbo įrankiai, kurie buvo naudojami atliekant eksperimentą.

2.1. Panašūs tyrimai

Šioje srityje (nusikaltimų prognozėje) yra atlikta gan nemažai tyrimų, kur buvo taikomi skirtingi prognozavimo metodai ir skirtingi duomenys.

[WGB12] tyrime yra panaudoti „Twitter“ naudotojų pranešimai, prognozuojant autoįvykių nusikaltimus, kur nusikaltėlis pasišalino iš eismo įvykio vietos (*angl. hit-and-run crimes*). [WGB12] šaltinio autoriai išgavo lokalius „Twitter“ pranešimus iš naujienų šaltinių, kurie padengia Šarlotsvilio miestą (Virdžinija, JAV). Pranešimai toliau buvo apdoroti naudojant naujausias natūralios kalbos apdorojimo (*angl. natural language processing*) technikas, kad išgautų visą reikalingą informaciją iš pranešimo. Toliau buvo nustatomos įvykių pagrįstos temos, naudojant neprižiūrimų temų modelį (*angl. unsupervised topic model*). Tuomet, pasinaudojant nustatytomis temomis, buvo prognozuojami būsimi nusikaltimai. Naudojant šį modelį rezultatai buvo geresni, negu kitų analizuojamų modelių rezultatai su testavimo duomenimis. Pažymėtina, kad socialiniuose tinkluose yra labai mažai kalbama apie nusikaltimus Vilniaus mieste, todėl toks metodas šiam tyrimui netiktų. Vis dėlto, nors tokio tyrimo metodai netinka šiam darbui, yra daug kitų darbų, kur yra naudojami praeitų nusikaltimų istorija apmokymui.

Šaltinyje [Pra18] yra išsamiai analizuojami pagrindiniai nusikaltimų tipai, kurie įvyko San Franciske. [Pra18] tyrimo metu buvo stebimas nusikaltimų tendencijų kitimas per metus ir nustatyta, kaip skirtingi atributai, pavyzdžiui, metų laikai, gali įtakoti specifinius nusikaltimų tipus. Remiantis analizės rezultatais [Pra18] šaltinio autoriai sukūrė modelį, prognozuojantį, koks nusikaltimų tipas įvyks specifinėje miesto dalyje. Siekiant sukurti tokį modelį, [Pra18] šaltinio autoriai naudojo populiarius mašininio mokymosi algoritmus, tokius kaip:

- K-arčiausių kaimynų (*angl. k-nearest neighbors*);
- Daugiaklasės regresijos (*angl. multi-class regression*);
- Sprendimų medžio (*angl. decision tree*);
- Atsitiktinio miško (*angl. random forest*);
- Naivaus polinkio (*angl. naive bayse*).

[Pra18] rezultatai buvo palyginti su kitais šaltiniais ir galutiniai rezultatai buvo panašūs. Daugiau paskaityti apie [Pra18] šaltinio rezultatus galima [Pra18, 28]. [Pra18] šaltinio autoriai panaudojo tik prieš tai buvusių nusikaltimų istoriją, todėl tolimesniuose tyrimuose autoriai nurodė, kad norėtų pamėginti įtraukti daugiau duomenų, tokių kaip populiacijos, apgyvendinimo,

transporto duomenis. Būtent papildomi duomenys, tokie kaip geografiniai ir demografiniai, buvo naudojami [WB11] ir [RHP17] šaltiniuose. Taip pat, [Pra18] šaltinyje tolimesniuose tyrimuose buvo paminėta, kad autoriai norėtų panaudoti dirbtinio neuroninio tinklo modelį, ką ir padarė [RHP17] šaltinio autoriai.

[RHP17] šaltinyje buvo prognozuojami Amsterdamo nusikaltimai, naudojant daugiasluoksnį perceptroną, loginės regresijos (*angl. logical regression*) algoritmą ir sprendimų medžio metodą. [RHP17] šaltinio autorių naudojami duomenys buvo net tik praeitų nusikaltimų duomenys, bet ir papildomi duomenys, tokie kaip: demografiniai, socialiniai ir ekonominiai bei aplinkos duomenys [RHP17, 257]. [RHP17] šaltinyje buvo prognozuojamos nusikaltimų vietos iš karto dviejų savaičių periodu. Tokiu būdu [RHP17] šaltinio autoriai gavo tiesioginį prognozės pataikymą virš 20% (teisingai prognozuotų nusikaltimų vietų procentinė dalis) ir tikslumu virš 25% (teisingų prognozių procentinė dalis, palyginus su bendru spėjimų skaičiumi). Toliau duomenys buvo suskaidyti į dienos ir nakties nusikaltimus, mėnesiniu periodu. Tokiu būdu pagerėjo prognozės rezultatai – tiesioginis pataikymas pakilo virš 50%, o tikslumas taip pat pakilo virš 50%. Vis dėlto, [RHP17] tyrime neuroninis tinklas stipriai neišsiskyrė nuo kitų mašininių mokymosi modelių. Daugiasluoksnio perceptrono modelio architektūra nėra tinkamiausia bandant prognozuoti nusikaltimus, kadangi nusikaltimai priklauso nuo prieš tai buvusių nusikaltimų [BJ05, 75], o kaip buvo minėta anksčiau, daugiasluoksnio perceptrono architektūra yra labiau tinkama statistiniams duomenims.

Toks eksperimentas, kuriame naudojamas rekurentinis neuroninis tinklas, o tiksliau LSTM modelis naudojamas siekiant prognozuoti kitos dienos nusikaltimų vietas, buvo atliktas [Cor18] šaltinyje. [Cor18] tyrimo autorius panaudojo nusikaltimų duomenis, gautus iš Gvatemalos policijos, šiuos duomenis sugrupavo pagal erdvinę priklausomybę, panaudojant „K-Means“ algoritmą. Sugrupuoti nusikaltimų duomenys buvo panaudoti apmokyti skirtingus LSTM modelius:

- Binarinis LSTM klasifikatorius, kuris turi priklausomybę tarp nusikaltimo grupių;
- Binarinis LSTM klasifikatorius, kuris neturi priklausomybės tarp nusikaltimo grupių;
- Regresinis LSTM klasifikatorius, kuris neturi priklausomybės tarp nusikaltimo grupių;
- Regresinis LSTM klasifikatorius, kuris turi priklausomybę tarp nusikaltimo grupių.

Regresinis LSTM klasifikatorius skiriasi nuo binarinio tuo, kad regresinio modelio įvestis yra nusikaltimų erdvinės grupės nusikaltimų skaičius, o binariniame yra tik 1 arba 0, kuris nusako, ar toje erdvinėje grupėje įvyko nusikaltimas, ar ne. Analogiškai yra ir su išvestimi. Binarinis LSTM modelis išveda reikšmę nuo 0 iki 1, kurie nusako, kiek modelis yra įsitikinęs, jog toje grupėje bus (ne)įvykdytas nusikaltimas. Tuo tarpu, regresinis LSTM modelis išveda tos erdvinės grupės prognozuojamą nusikaltimų skaičių. Pagal modelio architektūrą, šie du modeliai beveik

niekuo nesiskiria, išskyrus aktyvacijos funkciją. Taip yra dėl to, kad būtent aktyvacijos funkcija nusako, koks bus išvedimo formatas. Kadangi binarinio ir regresinio modelio išvestis turi būti skirtinga, jų aktyvacijos funkcijos taip pat skiriasi. Binarinis LSTM modelis naudoja minkštojo maksimumo aktyvacijos funkciją, o regresinis LSTM modelis – tiesinės aktyvacijos funkciją [Cor18, 320].

Skirtumas tarp modelio, kuris turi priklausomybę tarp nusikaltimo grupių ir modelio, kuris priklausomybės tarp nusikaltimo grupių neturi yra tik jo apmokymo procese. Modelis, kuris turi priklausomybę tarp nusikaltimų grupių, yra apmokomas su visomis erdvinėmis grupėmis iškart. Tokiu būdu modelis turėtų įsisavinti priklausomybes tarp skirtingų grupių. Kita vertus, modelis, kuris neturi priklausomybės tarp nusikaltimų grupių, yra apmokomas tik vienai specifiniai erdvinei nusikaltimų grupei. Tai lemia, kad kiekvienai specifinei erdvės grupei reikės atskiro LSTM modelio, kur kiekvienas turės būti atskirai apmokytas [Cor18, 320].

Pažymėtina, jog [Cor18] šaltinyje kiekvienas modelis buvo apmokytas su skirtingais parametrais, siekiant surasti geriausius parametrus. Tokiu būdu, LSTM modelis išvesdavo kitos dienos nusikaltimų grupės prognozę. Taip pat, pagal [Cor18] tyrimą buvo nustatyta, kad visi tiriami LSTM modeliai parodė geresnius rezultatus už tradicinius stebėjimo signalo (*angl. tracking signal*) algoritmus ir už mašininio mokymosi algoritmus, tačiau [Cor18] šaltinyje nebuvo naudojami jokie papildomi duomenys. Remiantis [RHP17] ir [WB11] šaltiniais, kuriuose naudoti papildomi duomenys, galima pagrįstai teigti, kad pridėjus papildomus duomenis prie LSTM modelio, bus pagerintas prognozavimo tikslumas.

Atsižvelgiant į [Cor18] tyrimą, šiam darbui buvo pasirinktas binarinis LSTM modelis, kuris turės priklausomybę tarp skirtingų nusikaltimų erdvinių grupių.

2.2. **Pasirinkti įrankiai**

Šiame darbe pasirinkti giliojo mokymosi įrankiai yra „Keras“ su „TensorFlow“, pasirinkta programavimo kalba yra Python, o kodo saugojimui bei redagavimui naudotas „Google Colaboratory“ įrankis.

„Keras“ yra aukšto lygio neuroninių tinklų programavimo sąsaja (*angl. application programming interface, API*) parašyta Python programavimo kalba, išplečianti „TensorFlow“, „CNTK“ arba „Theano“ įrankius. „Keras“ buvo sukurtas siekiant greitai eksperimentuoti su neuroniniais tinklais. Turėti galimybę greitai paversti idėją rezultatu su kaip įmanoma mažesniu uždelsimu yra svarbiausia gero tyrimo dalis. Būtent dėl šios priežasties šiam tyrimui įgyvendinti ir buvo pasirinkta „Keras“ programavimo sąsaja. Taip pat, šiame tyrime „Keras“ vidiniam programavimui (*angl. backend*) yra naudojamas „TensorFlow“.

Kadangi „Keras“ aplikacijų programavimo sąsaja yra skirta tik Python programavimo kalbai, visas šio tyrimo kodas buvo parašytas Python programavimo kalba.

Gilusis mokymasis reikalauja didelės apdorojimo galios. Tam, kad būtų paspartintas LSTM modelio apmokymo procesas, nuspręsta naudoti „Google Colaboratory“ įrankį. „Google Colaboratory“ yra nemokama „gyvo kodo“ (kaip „Jupiter Notebook“) aplinka, kuriai nereikia jokių išankstinių nustatymų ir kodas yra vykdomas debesyje. Su „Google Colaboratory“ per savo naršyklę galima nemokamai rašyti ir vykdyti kodą, išsaugoti ir bendrinti savo analizę ir pasiekti galingų kompiuterių išteklius.

3. EKSPERIMENTAS

Šiame skyriuje yra aprašoma tyrimo eiga. Pirmame poskyryje yra aprašoma, kokie duomenys buvo surinkti ir kaip jie buvo apdorojami. Antrame poskyryje yra aprašoma pasirinkta LSTM prognozės tikslumo vertinimo strategija. Paskutiniame, trečiame poskyryje, yra aprašoma, kokia LSTM modelio architektūra buvo naudojama, bei kaip modelis buvo apmokomas.

3.1. Reikalingų duomenų surinkimas ir paruošimas LSTM modeliui

Tyrimas prasidėjo nuo nusikaltimų duomenų rinkimo. Nusikaltimų duomenys buvo gauti iš nusikalstamų veikų žinybinio registro duomenų žemėlapio (NVŽR)[1] per jų vidinio programavimo sąsają. Tyrimui buvo pasirinkti nusikaltimai, kurie sukėlė žmogaus sveikatai žalą pagal ES nusikalstamų veikų klasifikavimo sistemą. Buvo pasirinkta būtent ši nusikalstamų veikų grupė, kadangi šio tipo nusikaltimų buvo daugiausia. Pasirinktas nusikaltimų laikotarpis buvo 4 paskutinieji metai nuo tyrimo pradžios, tai yra nuo 2015 metų pradžios iki 2018 metų pabaigos. Toks intervalas buvo pasirinktas, kadangi ir kituose šaltiniuose buvo imamas panašus nusikaltimų istorijos laiko intervalas. Taip pat, buvo imami nusikaltimai, kurie įėjo į 4.2 km ant 4.2 km kvadrato plotą, kurio centras buvo Vinco Kudirkos aikštė, esanti Vilniaus mieste. Tyrimui pasirinktas šis plotas , kadangi nusikaltimų žmogaus sveikatai Vilniaus centre buvo daugiausiai, lyginant su kitais Vilniaus miesto rajonais.

Surinkti nusikaltimų duomenys buvo sugrupuoti pagal erdvinę priklausomybę, siekiant apmokyti LSTM modelį. Duomenys buvo suskirstyti į tinklelį (panašiai kaip [RHP17] šaltinyje), kur vienos celės plotis yra 700×700 m^2. Pasirinktas šis grupavimo metodas, kadangi viena tinklelio celė savyje gali laikyti kitus duomenis. Tokiu būdu galima apmokyti modelį tiek be papildomų duomenų, tiek ir su papildomais duomenimis, bei palyginti jų tikslumą. Paruošus nusikaltimų duomenis, buvo pradėti rinkti papildomi duomenys.

Vykdant papildomų duomenų paiešką nebuvo rasta jokių tinkamų demografinių arba socialinių ir ekonominių duomenų, nes dauguma šių duomenų buvo renkami Lietuvos mastu ir laiko intervalas dažniausia buvo metinis, o šiam darbui buvo reikalingi duomenys, kurie yra dieninio arba mažesnio intervalo ir erdvinis išskaidymas būtų bent miesto mikrorajonų lygio. Kita vertus, buvo surinkti aplinkos duomenys, naudojant „Google Places“ paslaugą, kurios pagalba galima nustatyti, kokios vietovės yra arti norimos vietos. Tokiu būdu buvo surinktos vietovės, kurios padės geriau nusakyti aplinką celėje, kur buvo įvykdytas nusikaltimas. Buvo atrinktos 7 vietovių kategorijos:

- barai,

[1] https://www.ird.lt/nvzrgis/map/ [žiūrėta 2019-05-16]

- restoranai,
- parduotuvės,
- naktiniai klubai,
- parkai,
- bankai,
- viešojo transporto stotelės.

Pažymėtina, kad buvo pasirinktos šios vietos, kadangi jos dažniausiai pasitaikančios mieste ir turi skirtingas paskirtis, kas reiškia, kad šios vietos traukia skirtingą žmonių srautą ir žmonių elgesys šiose vietose skiriasi, todėl gali įtakoti miesto nusikaltimų tendencijas. Verta paminėti ir tai, kad aplinkos duomenys nesikeičia pagal laiką. Taip yra todėl, kad „Google Places“ paslauga grąžina tik vietoves, atitinkančias užklausos metų. Naudojant „Google Places“ paslaugą, negalima užklausti, kokios vietovės buvo specifiniu laiku, todėl senesni nusikaltimų duomenys yra sujungiami su aplinkos duomenimis, kurie yra naujesni negu nusikaltimų duomenys. Taip pat, visi aplinkos duomenys, kaip ir nusikaltimų duomenys, buvo sugrupuoti į tinklelį, kad šiuos duomenis būtų galima panaudoti LSTM modelio apmokymui.

3.2. LSTM modelio vertinimo strategijos pasirinkimas

Turint visus reikalingus duomenis, reikėjo nuspręsti, kaip bus vertinamas LSTM modelio tikslumas. Kadangi [Cor18] šaltinyje buvo naudojami LSTM modeliai, o šiame darbe irgi buvo naudojamas LSTM modelis, buvo nuspręsta naudoti [Cor18] šaltinio vertinimo strategiją. [Cor18] šaltinio autorius naudojo keturis matus, kurie parodė modelio prognozės tikslumą: bendras tikslumas (*angl. accuracy*), prisiminimas (*angl. recall*), pataikymo tikslumas (*angl. precision*), F_1 matas (*angl. F-measure*). Visose kitose formulėse *tp* reiškia tikrai teigiamų prognozių skaičius (*angl. true positive*), *tn* – tikrai neigiamų prognozių (*angl. true negative*) skaičius, *fp* – klaidingai teigiamų (*angl. false positive*) prognozių skaičius, *fn* – klaidingai neigiamų (*angl. false negative*) prognozių skaičius. Atitinkamai, daugiaklasei klasifikacijai (*angl. multiclass classification*) tp_i tn_i *fpi* ir fn_i yra klasės C_i vertinimai, o l – klasių kiekis. Bendros klasifikacijos kokybė paprastai vertinama dviem būdais. Pirmas, yra vertinimo vieneto kiekvienos klasės $C_1, \dots, C_l$ vidurkis. Antras, skaičiuojant sumą visų vienetų *tp, fn, tn, fp* kiekvienai klasei $C_1, \dots, C_l$ ir po to skaičiuojant norimą vertinimo vienetą.[SL09, 430]. Šiame darbe buvo pasirinktas būtent antras variantas.

- Bendras tikslumas parodo klasifikatoriaus bendrą veiksmingumą ir yra skaičiuojamas pagal 2 formulę [SL09, 430]. Paprastai tariant, tai yra procentinė dalis, kiek klasių teisingai buvo prognozuota, įskaitant ir teigiamas klases, ir neigiamas.

$$\frac{tp+tn}{tp+fn+fp+fn} \tag{2}$$

Tačiau šiame darbe, kai modelis prognozuoja, kurioje celėje tinklelyje bus įvykdytas nusikaltimas, tai jau yra daugiaklasinis prognozavimo uždavinys, todėl 2 formulė netinka ir yra imamas bendro tikslumo vidurkis, kuris yra skaičiuojamas pagal 3 formulę.

$$\frac{\sum_{i=1}^{l}\frac{tp_i+tn_i}{tp_i+fn_i+fp_i+tn_i}}{l} \quad (3)$$

- Pataikymo tikslumas (dar žinomas kaip specifiškumas) – tai teisingai prognozuotos teigiamos klasės (nusikaltimų celės) santykis su suma neteisingai ir teisingai teigiamų klasių prognozės suma (žr. 4 formulę).

$$\frac{tp}{tp+fp} \quad (4)$$

4 formulę galima pritaikyti tik uždaviniuose, kuriuose yra tik dvi galimos klasės. Daugiaklasiems uždaviniams yra naudojama 5 formulė, kur yra sumuojamas kiekvienos klasės teisingai teigiamų prognozių skaičius ir po to šis skaičius dalinamas iš visų tos klasės tikrai teigiamų ir klaidingai teigiamų prognozių skaičiaus. Kitaip tariant, tai yra teigiamų teisingų prognozių procentinė dalis, palyginus su bendru spėjimų skaičiumi. Kuo šis vienetas labiau artėja 100%, tuo klaidingai teigiamų prognozuojamų klasių skaičius mažėja.

$$\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp_i+fp_i)} \quad (5)$$

- Prisiminimas parodo klasifikatoriaus veiksmingumą, nustatant teigiamas etiketes ir yra skaičiuojamas panaudojus 6 formulę, kai yra tik dvi klasės, o 7 formulė yra naudojama, kai yra daugiau nei dvi klases.

$$\frac{tp}{tp+fn} \quad (6)$$

Kitaip tariant, prisiminimas yra teisingai prognozuotų nusikaltimų vietų procentinė dalis iš visų ten įvykusių nusikaltimų. Prisiminimas dar kitaip vadinamas tiesioginiu pataikymo procentu (*angl. direct hit rate*).

$$\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp_i+fn_i)} \quad (7)$$

- F1 matas sujungia pataikymo tikslumą ir prisiminimą. Binariniam klasifikatoriui naudojama 8 formulė, o daugiaklasiui klasifikatoriui 9 formulė. F1 matas iš esmės yra reikalingas tam, kad lengviau būtų galima spręsti, koks modelis geriau prognozuoja. Kadangi F1 matas sujungia du skirtingus matus (pataikymo tikslumą ir prisiminimą), užteks palyginti tik vieną skaičių, o ne du. O tai, kas yra svarbiau, t. y., ar pataikymo tikslumas, ar prisiminimas, sprendžia β parametras. Kuo didesnis yra β parametras, tuo pataikymo tikslumas yra svarbesnis.

$$\frac{(\beta^2+1)tp}{(\beta^2+1)tp+\beta^2 fn+fp} \quad (8)$$

Šiame darbe β reikšmė yra 0.95. Tokia reikšmė buvo pasirinkta tam, kad pataikymo tikslumas F1 mate turėtų daugiau svorio.

$$\frac{(\beta^2+1)*PataikymoTiklumas*Prisiminimas}{\beta^2 PataikymoTiklumas+Prisiminimas} \quad (9)$$

3.3. **LSTM modelio sukūrimas ir apmokymas**

Žinant, kaip bus vertinamas modelis, buvo galima pradėti kurti LSTM modelį ir jį apmokyti. Remiantis [Cor18] šaltinio rezultatais, buvo pasirinkta tokia pati LSTM modelio architektūra, kaip ir [Cor18] šaltinyje. LSTM modelio architektūrą sudarė „Adam“ optimizatorius, 100 LSTM elementų su 0,4 išmetimo (*angl. dropout*) dydžiu ir minkštojo maksimumo aktyvacijos funkcija. Apmokymo parametrai, kurie parodė geriausius rezultatus iš šaltinio [Cor18] buvo paimti kaip atspirties taškas, todėl pradinis epochų skaičius buvo 100 ir slenkančio lango dydis buvo 8. Duomenys buvo suskaidyti į apmokymo duomenis ir testavimo duomenis. Iš viso buvo 1461 dienų (4 metų) duomenys, tad apmokymui buvo išskirta 90% duomenų (1314 dienų), o testavimui likę 10% duomenų (147 dienų). Taip pat tam, kad būtų galima analizuoti modelio mokymosi procesą, iš apmokymo duomenų buvo išskirta 10% duomenų (131 dienų) validavimui.

Pirmojo apmokymo metu modelis buvo apmokomas su parametrais iš [Cor18] šaltinio pagal nustatytą tikslumo matavimo strategiją, tačiau modelio rezultatai buvo blogesni palyginus su tuo, kokie rezultatai buvo gauti [Cor18] ir [RHP17] šaltiniuose (žr. 1 lentelę).

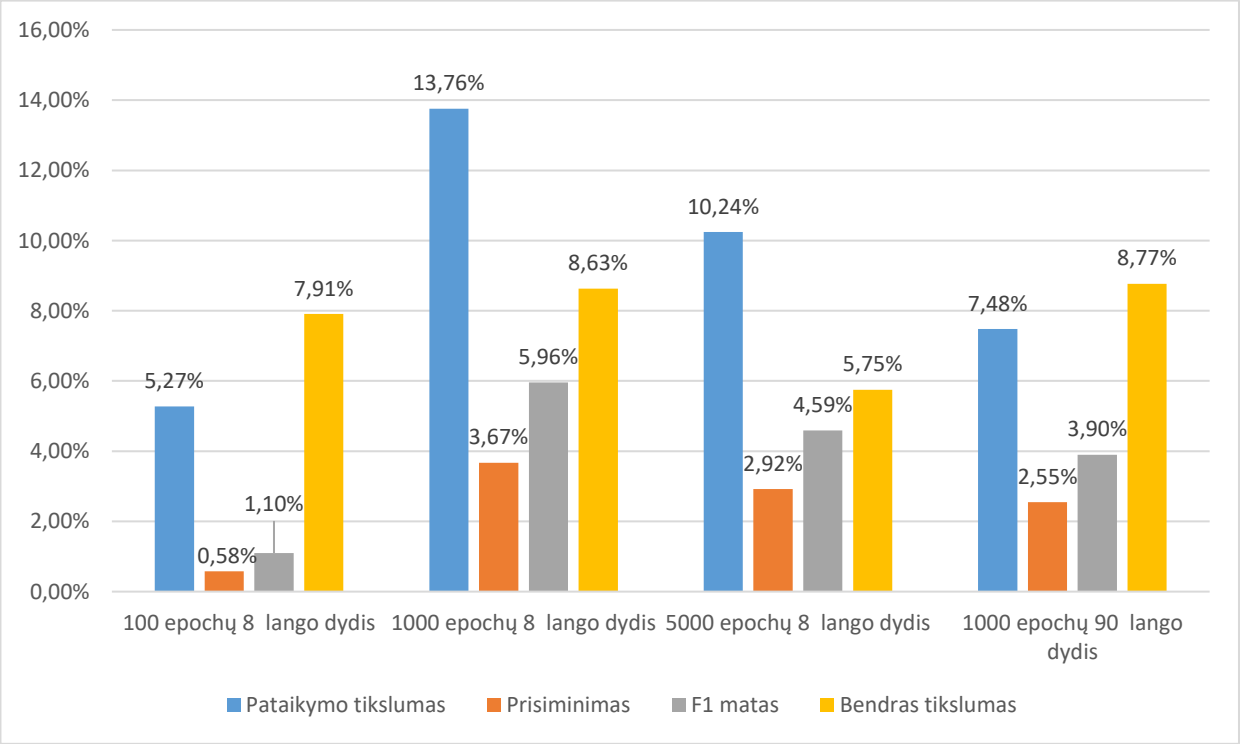
1 lentelė. Šio darbo LSTM modelio rezultatai ir kitų šaltinių prognozavimo rezultatai

Prognozavimo modelis		**Pataikymo tikslumas**	**Prisiminimas**	**F1 matas**	**Bendras tikslumas**
Daugiasluoksnis perceptronas su dviejų savaičių periodu [RHP17]		31,78%	53,19%	39,28%	-
Daugiasluoksnis perceptronas su mėnesio periodu [RHP17]	Nakties	55,68%	70,18%	61,73%	-
	Dienos	58,52%	70,48%	63,64%	-
Tarp grupių priklausomas binarinis LSTM modelis [Cor18]		60,71%	60,77%	60,73%	63,88%
Šio darbo LSTM modelis		5,27%	0,58%	1,10%	7,91%

Šio darbo LSTM modelio pataikymo tikslumas, lyginant su daugiasluoksnio perceptrono su dviejų savaičių periodu iš [RHP17] šaltinio, yra apie 6 kartus blogesnis, o palyginus su tarp grupių

priklausomų binarinių LSTM modeliu iš šaltinio [Cor18] – 11 kartų blogesnis. Prisiminimo tikslumas yra dar blogesnis, nei pataikymo tikslumas. Prisiminimas, lyginant jį su prasčiausiu modeliu iš kitų šaltinių, yra apie 86 kartus blogesnis bei 121 kartą blogesnis, lyginant jį su geriausiu modeliu iš kitų šaltinių. Modelių bendras tikslumas iš [RHP17] šaltinio nėra pateikiamas, bet palyginus šį matą su [Cor18] šaltinio rezultatu matyti, jog šio darbo LSTM modelio bendras tikslumas yra apie 8 kartus prastesnis. Kaip ir su bendru tikslumu, taip ir su F1 matu, ne visi šaltiniai jį pateikė, o jei jis ir buvo pateiktas, jame buvo naudojama kita β vertė. Šiame darbe yra naudojama β reikšmė lygi 0.95, kadangi šio darbo uždaviniams pasiekti pataikymo tikslumas yra svarbesnis už prisiminimą. Atsižvelgiant į tai, kitiems šaltiniams F1 matas buvo perskaičiuotas su šio darbo β verte, siekiant korektiškai palyginti šį matavimo vienetą. Šio darbo metu gautas F1 matas taip pat nėra didelis, jei jį lygintume su kitais šaltiniais. Lyginant šį matą su prasčiausiu modeliu iš kitų šaltinių, F1 matas yra apie 35 kartus blogesnis bei 57 kartus blogesnis lyginant jį su geriausiu modeliu iš kitų šaltinių .

Pažymėtina, jog tokius prastus rezultatus modelis išvesdavo dėl to, kad jis buvo nelabai įsitikinęs savo spėjimais. Modelio išvedamos tikimybės dažniausiai buvo apie 30%, o skaičiuojant vertinimo matus buvo laikoma, kad toje vietoje galimai bus įvykdyta nusikalstama veika tik tada, jei modelis yra įsitikinęs daugiau nei 50%. Atsižvelgiant į tai, buvo nuspręsta modelį apmokyti dar kartą, panaudojant skirtingų epochų skaičių, siekiant, jog modelis būtų labiau įsitikinęs savo prognozėmis. Taip pat, nuspręsta apmokyti modelį skirtingu slenkančio lango skaičiumi su prielaida, kad didesnis lango skaičius padės labiau įsitikinti modelio prognozėmis.



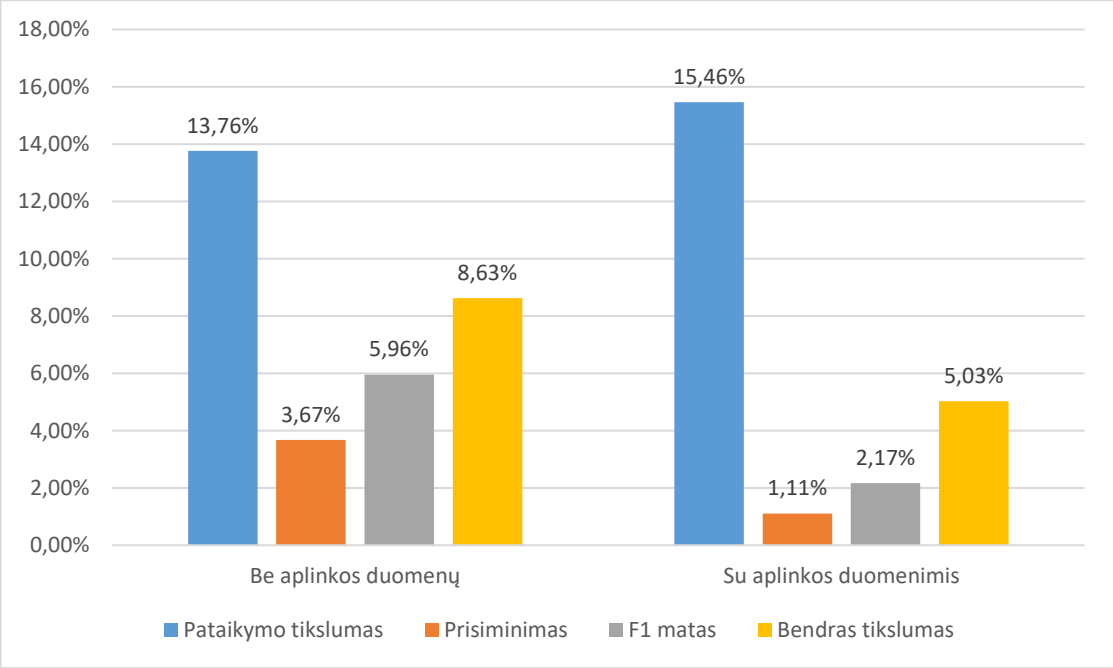
3 pav. Modelio prognozavimo rezultatai su skirtingais apmokymo parametrais

Atlikus LSTM modelio apmokymus su skirtingais parametrais, rezultatai pagerėjo. Iš 3 pav. matyti, kad geriausius rezultatus parodė modelis, kuris buvo apmokytas naudojant 1000 epochų ir kuriuo slenkančio lango skaičius buvo 8. Lyginant geriausius modelio rezultatus su pačiu pirmu apmokytu modeliu matyti, jogmodelio pataikymo tikslumas pagerėjo 2,6 karto, t. y. nuo 5,27% iki 13,76%. Tuo tarpu prisiminimas padidėjo 6,3 karto, F1 matas – 5,4 karto ir bendras tikslumas taip pat padidėjo, tačiaunežymiai – 1,09 karto.

Iš 3 pav. taip pat matyti, kad slenkančio lango dydžio padidinimas nuo 8 iki 90 modelio prognozės rezultatų nepagerino. Beveik visi matmenys pablogėjo, išskyrus bendrąji tikslumą, kuris padidėjo labai nežymiai – 1,6%. Pažymėtina, kadpataikymo tikslumas pablogėjo 1,8 karto, prisiminimas pablogėjo 1,4 karto ir F1 matas pablogėjo 1,5 karto.

Prie to, pagal 3 pav. yra matoma, kad, kai modelis buvo apmokytas 5000 epochų, visi jo matai pablogėjo, lyginant su modeliu, kuris buvo apmokytas 1000 epochų ir lango dydžiu 8. Remiantis tuo, galima pagrįstai teigti, kad modelis per daug prisitaikė prie mokymosi duomenų, t. y. buvo permokytas (*angl. overfitted*).

Nors modelio prognozės rezultatai pagerėjo, vis dėltopagerinti ir net pasiekti kitų šaltinių rezultatų nepavyko.Buvo iškelta hipotezė, jog su papildomais aplinkos duomenimis modelio prognozės tikslumas pagerės, todėl nuspręsta apmokyti modelį su geriausiais apmokymo parametrais, panaudojant papildomus aplinkos duomenis.

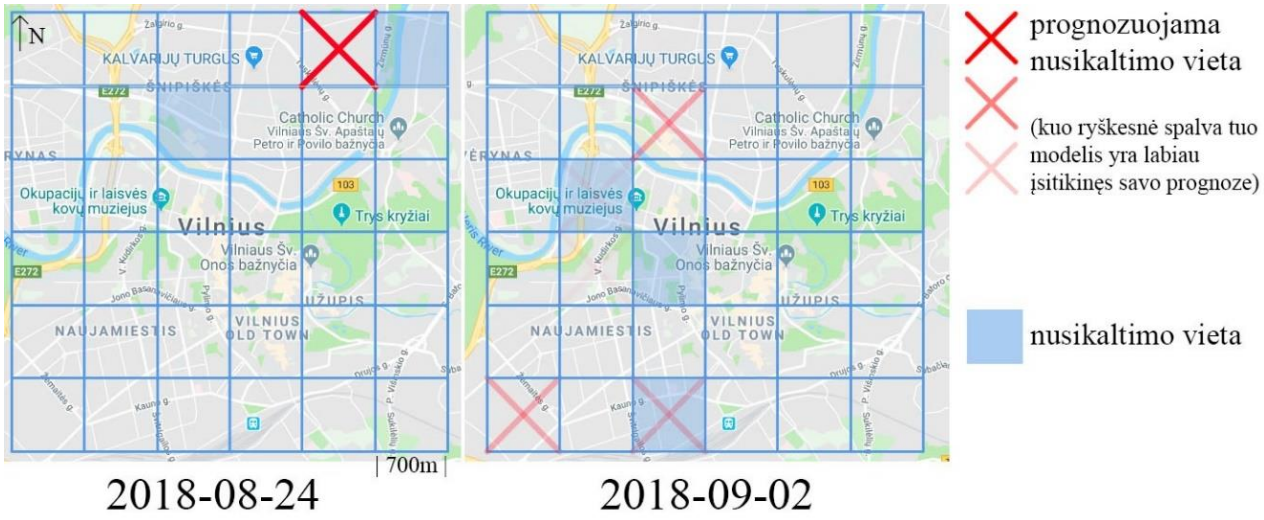


4 pav. Modelio prognozavimo rezultatai be aplinkos duomenų ir su aplinkos duomenimis

Panaudojus papildomus aplinkos duomenis, modelio rezultatai nepagerėjo, išskyrus pataikymo tikslumą (žr. 4 pav.). Pridėjus papildomus aplinkos duomenis prie apmokymo, modelio pataikymo tikslumas pagerėjo 1,12 karto, tačiau prisiminimas pablogėjo apie 3,3 karto, F1 matas

pablogėjo apie 2,74 karto ir bendras tikslumas taip pat pablogėjo apie 1,7 karto. Galimai taip įvyko dėl to, kad aplinkos duomenys buvo statiniai, t. y. jie nesikeitė laike, todėl nesvarbu, ar nusikaltimai buvo įvykdyti 2015 metais, ar 2018 metais, aplinkos duomenys buvo tokie patys. Aplinkos duomenys buvo statiniai kadangi pasinaudojus „Google Places“ paslauga, gautos vietos buvo tik iš vieno laiko momento. Laikyta, kad aplinka mieste gan lėtai keičiasi, todėl tokie statiniai duomenys turėjo padėti, tačiau kaip matyti iš rezultatų, taip nėra.

Nors modelio prognozės rezultatai nepasiekė kitų šaltinių rezultatų tikslumo, vis dėlto buvo padaryta integracija su „Google Maps“, siekiant lengviau suprasti modelio išvedimą. Atsižvelgiant į tai, galutinė modelio išvestis buvo kitos dienos žemėlapis, kur celės su nusikaltimų tikimybe buvo pažymėtos raudonu „X” ženklu (žr. 5 pav.). Be kita ko, tam, kad būtų galima vizualiai pažiūrėti ir įvertinti modelio prognozes, žemėlapyje buvo pridėti žymekliai, žymintys, kurioje celėje iš tiesų nusikaltimas buvo įvykdytas (žr. 5 pav.). Pažiūrėjus į modelio prognozes matyti, kad nors modelio tiesioginio pataikymo procentas, palyginus su kitais šaltiniais, yra labai mažas, tačiau dažniausiai spėjimas yra labai arti celės, kurioje išties buvo įvykdytas nusikaltimas (žr. 5 pav.).



5 pav. LSTM modelio prognozės rezultatai su testavimo duomenimis

REZULTATAI

1. Buvo surinkti nusikaltimai (nusikaltimai žmogaus sveikatai) Vilniaus mieste iš nusikalstamų veikų žinybinio registro duomenų žemėlapio nuo 2015 m. pradžios iki 2018 m. galo. Šie duomenys buvo sugrupuoti į tinklelį, kad juos galėtų priimti LSTM modelis. Taip pat, modelio apmokymui iš papildomų duomenų buvo gauti aplinkos duomenys, tokie kaip: barai, restoranai, parduotuvės, naktiniai klubai, parkai, bankai ir viešojo transporto stotelės. Šie duomenys buvo sujungti kartu su nusikaltimų duomenimis ir taip pat sugrupuoti į tinklelį.

2. Buvo pasirinkta LSTM modelio prognozės vertinimo strategija. Į strategiją įėjo keturi matai, pagal kuriuos buvo sprendžiamas prognozės tikslumas. Naudoti matai: bendras tikslumas, tiesioginis pataikymas, prisiminimas ir F1 matas.

3. Iš [Cor18] šaltinio buvo paimta LSTM modelio architektūra. Modelis buvo apmokytas nuo nulio ne tik su nusikaltimų duomenimis, bet ir su aplinkos duomenimis. Taip pat, atsižvelgiant į modelio rezultatus, modelio apmokymas įvyko kelis kartus, panaudojant skirtingus parametrus. Svarbiausias šio darbo rezultatas yra modelio išvestis, t. y. sekančios dienos žemėlapis, kur kiekvienai tinklelio celei yra priskirta nusikaltimo prognozės tikimybę.

IŠVADOS

1. Patyrusiems policininkams, tokie LSTM modelio prognozavimo rezultatai gali suteikti naujų įžvalgų, kurių policijos pareigūnai galbūt nežinojo ar neįvertino. Kadangi modelis dažniausiai prognozavo nusikaltimą šalia celės, kurioje iš tiesų buvo padarytas nusikaltimas, policijos pareigūnai galėtų pasinaudoti ir atsižvelgti į tokį nusikaltimų tikimybių žemėlapį, kuriant patruliavimų maršrutus. Šiame darbe buvo analizuojami tik nusikaltimai žmogaus sveikatai, tačiau yra galimybė tirti ir kitų tipų nusikaltimus. Pakeitus LSTM modelio apmokymo duomenis į vagystes, modelis išvedinėtų kitos dienos tikimybių žemėlapį, kuriame miesto sektoriuje yra labiausiai tikėtina, jog bus įvykdytas nusikaltimas. Toks žemėlapis taip pat galėtų padėti draudimo įmonėms įvertinti nekilnojamo turto draudimo kainas.

2. Papildomų aplinkos duomenų panudojimas modelio apmokymui modelio prognozės rezultatų nepagerino. Taip įvyko dėl to, kad aplinkos duomenys buvo statiniai. Aplinkos duomenys buvo surinkti pasinaudojus „Google Places“ paslauga, kurios pagalba buvo galima gauti tik dabartines (2019 metų) vietas. Buvo padaryta prielaida, kad aplinka taip greitai nesikaičia, todėl naudoti tokius pačius aplinkos duomenis per visus nusikaltimų laikotarpius galima, tačiau gauti rezultatai šią prielaidą paneigė.

Šis tyrimas tik palietė nusikaltimų prognozės srities paviršių, todėl dar yra daug vietos šio darbo vystymui ir plėtojimui. Tęsiant darbą, būtų galima pasiteirauti Vilniaus miesto savivaldybės dėl papildomų duomenų, tokių kaip: demografinių, socialinių ir ekonominių ir t.t. Pridėjus tokius duomenis prie modelio apmokymo, yra galimybė, kad šie duomenys stipriai pagerins modelio prognozę [Har15]. Taip pat, būtų galima prieš LSTM modelio apmokymą išskaidyti duomenis į darbo dienas ir šventines/savaitgalio dienas, į žiemos ir vasaros sezonus, su prielaida, kad nusikaltimų elgesys išskaidytose grupėse stipriai skiriasi ir tokiu būdu LSTM modeliui būtų lengviau atpažinti nusikaltimų sąryšius.

ŠALTINIAI

[BJ05] BOWERS, Kate J.; JOHNSON, Shane D. Domestic burglary repeats and space-time clusters: The dimensions of risk. European Journal of Criminology, 2005, 2.1: 67-92.

[BR93] BLUM, Avrim L.; RIVEST, Ronald L. Training a 3-node neural network is NP-complete. In: Machine learning: From theory to applications. Springer, Berlin, Heidelberg, 1993. p. 9-28.

[Cor18] CORTEZ, Bitzel, et al. An architecture for emergency event prediction using LSTM recurrent neural networks. Expert Systems with Applications, 2018, 97: 315-324.

[GB10] GLOROT, Xavier; BENGIO, Yoshua. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. 2010. p. 249-256.

[Har15] HARDYNS, Wim, et al. Study protocol: SWING–social capital and well-being in neighborhoods in Ghent. International journal for equity in health, 2015, 14.1: 36.

[Hoc01] HOCHREITER, Sepp, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. 2001.

[JCZ12] JANSSEN, Marijn; CHARALABIDIS, Yannis; ZUIDERWIJK, Anneke. Benefits, adoption barriers and myths of open data and open government. Information systems management, 2012, 29.4: 258-268.

[JMM96] JAIN, Anil K.; MAO, Jianchang; MOHIUDDIN, K. M. Artificial neural networks: A tutorial. Computer, 1996, 3: 31-44.

[KC14] KUCERA, Jan; CHLAPEK, Dusan. Benefits and risks of open government data. Journal of Systems Integration, 2014, 5.1: 30-41.

[LBE15] LIPTON, Zachary C.; BERKOWITZ, John; ELKAN, Charles. A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019, 2015.

[Pra18] PRADHAN, Isha. Exploratory Data Analysis And Crime Prediction In San Francisco. 2018.

[RHP17] RUMMENS, Anneleen; HARDYNS, Wim; PAUWELS, Lieven. The use of predictive analysis in spatiotemporal crime forecasting: Building and testing a model in an urban context. Applied geography, 2017, 86: 255-261.

[SL09] SOKOLOVA, Marina; LAPALME, Guy. A systematic analysis of performance measures for classification tasks. Information Processing & Management, 2009, 45.4: 427-437.

[Ste10] STEFANOWSKI, Jerzy. Artificial Neural Networks–Basics of MLP, RBF and Kohonen Networks. Institute of Computing Science Lecture 13 in Data Mining for M. Sc. Course of SE version for 2010, 2010.

[WB11] WANG, Xiaofeng; BROWN, Donald E. The spatio-temporal generalized additive model for criminal incidents. In: Proceedings of 2011 IEEE International Conference on Intelligence and Security Informatics. IEEE, 2011. p. 42-47.

[WGB12] WANG, Xiaofeng; GERBER, Matthew S.; BROWN, Donald E. Automatic crime prediction using events extracted from twitter posts. In: International conference on social computing, behavioral-cultural modeling, and prediction. Springer, Berlin, Heidelberg, 2012. p. 231-238.