



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

VERSLO VADYBOS FAKULTETAS

FINANSŲ INŽINERIJOS KATEDRA

Greta Pratuzaite

**DIRBTINIO INTELEKTO TAIKymo BANKŲ SEKTORIJE
TYRIMAS**

**RESEARCH OF ARTIFICIAL INTELLIGENCE APPLICATION IN
THE BANKING SECTOR**

Magistro baigiamasis darbas

Finansų inžinerijos studijų programa, valstybinis kodas 6211LX060

Finansinių technologijų (FinTech) specializacija

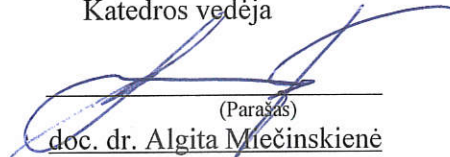
Finansų studijų kryptis

Vilnius, 2020

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
VERSLO VADYBOS FAKULTETAS
FINANSŲ INŽINERIJOS KATEDRA

TVIRTINU

Katedros vedėja


(Parašas)
doc. dr. Algita Mitečinskienė
(Vardas, pavardė)
2020-01-07
(Data)

Greta Pratuzaite

**DIRBTINIO INTELEKTO TAIKymo BANKŲ SEKTORIJE
TYRIMAS**
**RESEARCH ON ARTIFICIAL INTELLIGENCE APPLICATION IN
THE BANKING SECTOR**

Baigiamasis magistro darbas

Finansų inžinerijos studijų programa, valstybinis kodas 6211LX060
Finansinių technologijų (FinTech) specializacija
Finansų studijų kryptis

Vadovas

dr. Nijolė Maknickienė

(Moksl. laipsnis/pedag. vardas, vardas, pavardė)



(Parašas)

2020 01 02

(Data)

Lietuvių kalbos konsultantas

doc. dr. Rasuolė Vladarskienė

(Moksl. laipsnis/pedag. vardas, vardas, pavardė)



(Parašas)

2019 12 16

(Data)

Vilnius, 2020

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
VERSLO VADYBOS FAKULTETAS
FINANSŲ INŽINERIJOS KATEDRA

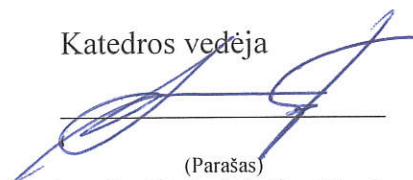
Finansų studijų kryptis

Finansų inžinerijos studijų programa, valstybinis kodas 6211LX060

Finansinių technologijų (FinTech) specializacija

TVIRTINU

Katedros vedėja



(Parašas)

doc. dr. Algita Miečinskienė

(Vardas, pavardė)

2018-10-05

(Data)

**BAIGIAMOJO MAGISTRO DARBO
UŽDUOTIS**

2018-11-05 Nr.

Vilnius

Studentui (ei) Gretai Pratuzaitei
- (Vardas, pavardė)

Baigiamojo darbo tema:

Dirbtinio intelekto taishymas bankų sektoriuje tyrimas

patvirtinta 2018 m. spalio 31 d. dekanų potvarkiu Nr. 238vv

Baigiamojo darbo užbaigimo terminas 2020 m. sausio 2 d.

BAIGIAMOJO DARBO UŽDUOTIS:

1. Apibrėžti dirbtinio intelekto sąvoką ir išanalizuoti konkrečius pavyzdžius, kokiose srityse dirbtinis intelektas yra naudojamas, pateikti jo galimas naudas, rizikas ir įtaką.
2. Atlikti mokslinės literatūros analizę apie dirbtinį intelektą, jo pritaikymą ir mokymosi algoritmus, kurie gali būti taikomi bankų sektoriuje.
3. Išanalizuoti kreditinių kortelių sukčiavimo problemą ir galimybes, kaip ši problema gali būti išspręsta, kai yra naudojamas dirbtinis intelektas.
4. Išnagrinėti pasirinktų metodų taikymą kreditinių kortelių sukčiavimo problemai spręsti.
5. Atlikti tyrimą apie kreditinių kortelių sukčiavimo tikslumo nustatymą, remiantis keliais pasirinktais metodais ir palyginti gautus rezultatus.
6. Apskaičiuoti, kokią finansinę naudą gali gauti finansinė institucija tinkamai suspėjus atpažinti sukčiavimus.

Vadovas
(Parašas)

dr. Nijolė Maknickienė
(Moksl. laipsnis/pedag.vardas, vardas, pavardė)

Užduotį gavau

.....
(Parašas)

(Vardas, pavardė)

2018 m. lapkričio 5 d.

(Data)

Vilniaus Gedimino technikos universitetas
Verslo vadybos fakultetas
Finansų inžinerijos katedra

ISBN ISSN
Egz. sk. 1.....
Data 2020.01.02

Antrosios pakopos studijų **Finansų inžinerijos** programos magistro baigiamasis darbas

Pavadinimas	Dirbtinio intelekto taikymo bankų sektoriuje tyrimas
Autorius	Greta Pratuzaitė
Vadovas	Nijolė Maknickienė

Kalba: lietuvių

Anotacija

Baigiamajame magistro darbe nagrinėjamas dirbtinio intelekto taikymas bankų sektoriuje, vertinama jo svarba ir galimos rizikos. Pateikiami pagrindiniai pavyzdžiai, kokiose srityje bankų sektoriuje dirbtinis intelektas naudojamas, kokios perspektyvos ateičiai. Išnagrinėti paplitę algoritmai, kurie yra naudojami sukčiavimo atpažinimo uždaviniams spręsti. Pateikti sukčiavimo metodų vertinimai, lyginami mašininio mokymosi ir giliojo mokymosi algoritmai pagal tai, kuris metodas atpažįsta sukčiavimą geriausiai. Apskaičiuojama metodų finansinė nauda. Išnagrinėjus teorinius ir praktinius dirbtinio intelekto taikymo bankų sektoriuje aspektus, pateikiamos baigiamojo darbo išvados ir siūlymai.

Darbą sudaro 5 dalys: įvadas, teorinė dalis, praktinė dalis, išvados ir siūlymai, literatūros sąrašas.

Darbo apimtis – 78 p. teksto be priedų, 36 iliustr., 11 lent., 86 bibliografiniai šaltiniai.

Atskirai pridedami darbo priedai.

Prasminiai žodžiai: dirbtinis intelektas, bankai, taikymas, sukčiavimai, mokėjimai.

Vilnius Gediminas Technical University
Faculty of Business Management
Department of Financial Engineering

ISBN ISSN
Copies No. 1
Date 2020.01.02

Master Degree Studies **Financial Engineering** study programme Master Graduation Thesis

Title **Research on Artificial Intelligence Application in the Banking Sector**

Author **Greta Pratuzaitė**

Academic supervisor **Nijolė Maknickienė**



Thesis language: Lithuanian

Annotation

The thesis describes the artificial intelligence concept in the financial sector considering its importance and risk. The use of artificial intelligence and perspectives for the future are presented in the thesis. Common algorithms which are the main tasks of fraud detection are tested and modified. Provide fraudulent methodological assessments, compare machine learning and deep learning initial priorities. The financial benefits of methods are calculated. After analyzing theoretical and practical artificial intelligence application in the banking sector, presenting conclusions and solutions.

Structure: introduction, theoretical and practical part, conclusions and suggestions, references.

Thesis consist of: 78 p. text without appendixes, 36 pictures, 11 tables, 86 bibliographical entries.

Appendixes included.

Keywords: artificial intelligence, banks, fraud, application, payments.

Vilniaus Gedimino technikos universiteto
egzaminų sesijų ir baigiamųjų darbų
rengimo bei gynimo organizavimo tvarkos
aprašo
2 priedas

(Baigiamojo darbo sąžiningumo deklaracijos forma)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Greta Pratužaitė, 20183005
(Studento vardas ir pavardė, studento pažymėjimo Nr.)
Verslo vadybos fakultetas
(Fakultetas)
Finansų inžinerija, FTfm-18
(Studijų programa, akademinė grupė)

BAIGIAMOJO DARBO (PROJEKTO)

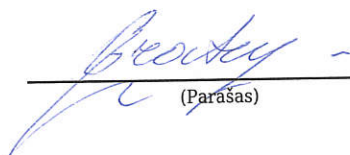
SĄŽININGUMO DEKLARACIJA

2020 m. sausio 02 d.

Patvirtinu, kad mano baigiamasis darbas tema „Dirbtinio intelekto taikymo bankų sektoriuje tyrimas“ patvirtintas 2018 m. spalio 31 d. dekanų potvarkiu Nr. 238vv, yra savarankiškai parašytas. Šiame darbe pateikta medžiaga nėra plagijuota. Tiesiogiai ar netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Mano darbo vadovas daktaras Nijolė Maknickienė.

Kitų asmenų indėlio į parengtą baigiamąjį darbą nėra. Jokių įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs (-usi).


(Paršas)

Greta Pratužaitė
(Vardas ir pavardė)

TURINYS

IVADAS.....	13
1. DIRBTINIO INTELEKTO FINANSŲ SEKTORIUJE TEORINIAI ASPEKTAI ..	16
1.1. Dirbtinis intelektas ir jo veikimas	16
1.1.1. Mašinos mokymasis	18
1.1.2. Neuroniniai tinklai.....	19
1.1.3. Gilusis mokymasis	22
1.1.4. Neuroninių tinklų architektūros.....	22
1.1.5. Algoritmai.....	24
1.2. Dirbtinio intelekto panaudojimas.....	27
1.3. Dirbtinio intelekto nauda ir rizikos	34
1.4. Dirbtinio intelekto įtaka	35
1.5. Mokėjimo kortelių sukčiavimo aptikimo technikos	39
2. SUKČIAVIMO ATPAŽINIMO TYRIMO METODAI	41
2.1. Anomalijos ir jų aptikimas.....	42
2.2. Duomenų klasifikavimas	43
2.3. Klaidų matrica.....	43
2.4. AUC - ROC kreivė	45
2.5. Apibendrintas momentų metodas ir jo taikymas sukčiavimui.....	46
2.6. K-artimiausių kaimynų metodo pritaikymas sukčiavimui.....	47
2.7. Naivusis Bajeso metodas ir jo naudojimas sukčiavimui atpažinti.....	47
2.8. Aktyvavimo funkcijos giliam mokymuisi atlikti	48
3. KREDITINIŲ KORTELIŲ MOKĖJIMŲ SUKČIAVIMO TYRIMAS.....	53
3.1. Apibendrintas momentų metodas	56
3.2. K-artimiausio kaimyno metodas	59
3.3. Naivusis Bajeso klasifikatoriaus metodas.....	61
3.4. Giliojo mokymosi algoritmas	62
3.5. Finansinė sukčiavimo atvejus atpažįstančių modelių nauda.....	68

IŠVADOS IR PASIŪLYMAI	70
LITERATŪROS SĄRAŠAS.....	72

LENTELIŲ SĄRAŠAS

1 lentelė. Sukčiavimo atvejai Europoje ir tam tikrose šalyse	33
2 lentelė. GMM modelio tikslingumas	57
3 lentelė. Tikslumo rezultatai pagal k-artimiausio kaimyno metodą.....	60
4 lentelė. Tikslumo rezultatai pagal Naive Bayes metodą.....	61
5 lentelė. Giliojo mokymosi testavimo rezultatai - 1 bandymas	63
6 lentelė. Giliojo mokymosi testavimo rezultatai - 2 bandymas	65
7 lentelė. Giliojo mokymosi testavimo rezultatai - 3 bandymas	66
8 lentelė. Giliojo mokymosi testavimo rezultatai - 4 bandymas	66
9 lentelė. Giliojo mokymosi testavimo rezultatai - 5 bandymas	67
10 lentelė. Tikslumo rezultatai pagal Giliojo mokymosi metodą.....	67
11 lentelė. Finansinė modelių nauda	69

ILIUSTRACIJŲ SĄRAŠAS

1 pav. Pasitikėjimas dirbtiniu intelektu dabar ir ateityje	17
2 pav. Dirbtinio intelekto struktūra.....	18
3 pav. Mašininio mokymosi struktūra	19
4 pav. Žmogaus neuronas	20
5 pav. Dirbtinio neurono modelis	20
6 pav. Giliojo mokymosi grafikas	22
7 pav. Rekurentiniai tinklai be savarankiško grįžimo kilpų ir be paslėptų neuronų	23
8 pav. Rekurentiniai tinklai be paslėptų neuronų	23
9 pav. Vektorių palaikymo mašinos modelio pavyzdys	25
10 pav. K-artimiausio kaimyno modelio pavyzdys	26
11 pav. Naive Bayes klasifikavimo metodo pavyzdys	26
12 pav. Sukčiavimo statistika	32
13 pav. Mokėjimų pasiskirstymas tarp skirtingų kanalų	34
14 pav. Banko darbuotojų skaičių pokyčiai Europoje	36
15 pav. Susidomėjimas bankų virtualiais asistentais internete.....	37
16 pav. Tyrimo eigos grafikas	41
17 pav. Klaidų matrica.....	44
18 pav. AUC-ROC kreivės.....	45
19 pav. GMM modelio duomenų skaidymas.....	46
20 pav. Sigmoidinės funkcijos grafikas.....	50
21 pav. Tangento funkcijos grafikas.....	51
22 pav. RELU funkcijos grafikas	52
23 pav. Mokėjimo klasių paskirstymas	53
24 pav. Mokėjimų skaičius pagal klases	54
25 pav. Mokėjimo judėjimai laiko atžvilgiu pagal klases	55
26 pav. Komponentų koreliacija.....	55
27 pav. Bruožų pasiskirstymas	56
28 pav. AUC kreivės pagal GMM modelį.....	58
29 pav. GMM modelio klaidų matricos.....	59
30 pav. Klaidų matricos pagal k-artimiausio kaimyno metodą.....	60
31 pav. Klaidų matrica pagal Naive Bayes metodą.....	61
32 pav. Neuroninių tinklų architektūros pagal skirtingą sluoksnių ir neuronų pasiskirstymą .	64

33 pav. Mokymosi grafikas - 2 bandymas.....	65
34 pav. Mokymosi grafikas - 3 bandymas.....	66
35 pav. AUC kreivė pagal Gilijo mokymosi modelį.....	68
36 pav. Klaidų matrica pagal Giliojo mokymosi metodą.....	68

PRIEDŲ SĄRAŠAS

1 PRIEDAS. Bruožų pasiskirstymas	79
2 PRIEDAS. GMM modelio „Python“ kodas	83
3 PRIEDAS. K-artimiausio kaimyno modelio „Python“ kodas	85
4 PRIEDAS. Naive Bayers modelio „Python“ kodas.....	86
5 PRIEDAS. Giliojo mokymosi „Python“ kodas	87
6 PRIEDAS. Straipsnis - Kreditinių kortelių sukčiavimo atpažinimo tymas.....	91

IVADAS

Dabar inovacijos ir technologijos yra stipriai augančios ir besikeičiančios. Bet kad jų sėkmė būtų pasiekta, išties svarbus yra pasitikėjimas jomis. Žinomos kompanijos, tokios kaip „Apple“, „Amazon“, „Netflix“ stengiasi protingai ir apdairiai rizikuoti, darbuotojams leidžia ieškoti išmanių ir išsiskiriančių idėjų, bet taip pat jos neužmiršta ir pasitikėjimo, kurį turi kelti klientui, svarbos (Botsman, 2017).

Pasitikėjimas padidina galimybes, žmonių suvokimą apie išradimus, plečia bendradarbiavimo galimybes, kūrybines galias. Tačiau jei yra kalbama apie pasitikėjimą robotais, kurie geba atlikti tam tikras funkcijas, pasitikėjimas jais gali greitai sumažėti dėl menkiausios klaidos. Neurologas ir psichologas Kreuger'is mano, kad daugiau šia mašina žmogus nebepasikliaus, jei ji jį nuvils. Bet jei ta mašina ims ir atsiprašys, didelė tikimybė, kad pasitikėjimas bus susigrąžintas ir suteiktas kitas šansas mašinai vykdyti paskirtą užduotį. Visgi todėl tam tikri robotai yra užprogramuoti tiek šypsotis, tiek susiraukti ar kitaip skleisti žmogiškus signalus ir pavergti vartotoją į savo pusę.

Taip pat ir bankai siekia sukurti programas, kurios veiktų lyg tikros žmogaus smegenys ir galėtų atlikti žmonėms prilygstančias užduotis, tokias kaip: mokėjimų sukčiavimų atpažinimas, ateities įvykių prognozavimas ir netgi tobulėti, prisitaikyti, kai yra atliekama daugiau tam tikro pobūdžio užduočių. Vis dažniau robotų sąvoka yra prilyginama dirbtiniam intelektui, kuris bankuose plinta kiek lėčiau nei kituose sektoriuose. Priežasčių šiam reiškiniui gali būti daug, tiek senesnės sistemos, dideli duomenų srautai, atsakomybės, apribojimai, darbuotojų kaita ir baimės, kad bus prarastas klientų pasitikėjimas. Bet reikia nepamiršti, kad dirbtinis intelektas dar nėra tobulas, programos klysta, nes pati rinka yra neprognozuojama, todėl vis dar yra svarbus žmogaus įsikišimas.

Įvairūs tyrimai rodo, kad bankai investuoja į dirbtinį intelektą ir į žmonių - mašinų bendradarbiavimo priemones, o šios technologinės perspektyvos didina jų pajamas ir mažina išlaidas. Tačiau bankas stipriai patiria konkurencinį spaudimą iš aplinkos - vienas iš jų susijęs su finansinių technologijų įmonėmis (toliau - Fintech). Kai kurie bankai bando su tokiomis įmonėmis bendradarbiauti, taip patenkinti žmonių lūkesčius ir bendrai kurti inovatyvius sprendimus.

Taigi, dirbtinis intelektas vis labiau yra pritaikomas ir dar daugiau sektorių bando jį įtraukti į savo vykdomas veiklas, kad darbas būtų efektyvesnis ir našesnis, kaštai sutaupomi ir pritraukiama daugiau klientų. Bet kyla klausimas, koks dirbtinio intelekto taikymas egzistuoja tradiciniuose bankuose, kai Fintech įmonės vis kalba apie savus dirbtinius intelektus ir pristato

vis naujesnius būdus, pritraukiančius klientus ir kartais net nuviliojančius nuo tradicinių bankų pasiūlymų.

Darbo problema: ar dirbtinio intelekto taikymas bankų sektoriuje atneša finansinę naudą?

Darbo tikslas: nustatyti, kaip dirbtinis intelektas yra taikomas bankų sektoriuje ir kokius rezultatus jo panaudojimas lemia.

Darbo uždaviniai:

1. Apibrėžti dirbtinio intelekto sąvoką ir išanalizuoti konkrečius pavyzdžius, kokiose srityse dirbtinis intelektas yra naudojamas, pateikti jo galimas naudas, rizikas ir įtaką.
2. Atlikti mokslinės literatūros analizę apie dirbtinį intelektą, jo prisitaikymą ir mokymosi algoritmus, kurie gali būti taikomi bankų sektoriuje.
3. Išanalizuoti kreditinių kortelių sukčiavimo problemą ir galimybes, kaip ši problema gali būti išspręsta, kai yra naudojamas dirbtinis intelektas.
4. Išnagrinėti pasirinktų metodų taikymą kreditinių kortelių sukčiavimo problemai spręsti.
5. Atlikti tyrimą apie kreditinių kortelių sukčiavimo tikslumo nustatymą, remiantis keliais pasirinktais metodais ir palyginti gautus rezultatus.
6. Apskaičiuoti, kokią finansinę naudą gali gauti finansinė institucija tinkamai suspėjus atpažinti sukčiavimus.

Tyrimo metodai: lietuvių ir užsienio mokslinės literatūros šaltinių analizė, atliktų tyrimų aprašymas, palyginimas, interpretavimas ir grafinis pavaizdavimas, naudojant „Python“ programą analizuojami kreditinių kortelių sukčiavimai įtraukiant apibendrintą momentų, naivųjį Bajeso, k-artimiausio kaimyno ir giliojo mokymosi algoritmus, rezultatų analizė keičiant pagrindinius parametrus, kurie turi įtakos tikslingesniai kreditinių kortelių sukčiavimų atpažinimui.

Darbo vertė: šį darbą sudarys trys dalys: teorijos pagrindimas, tyrimo metodika ir eksperimentiniai tyrimai. Pirmoje baigiamojo darbo dalyje bus pristatoma dirbtinio intelekto sąvoka, jos panaudojimas, pagrindiniai metodai ir svarba finansinėse įstaigose. Antroje baigiamojo darbo dalyje išsamiau aptariami metodiniai tyrimo aspektai, skirti kreditinių kortelių sukčiavimų atpažinimui, aprašomi taikomi metodai, kurie bus panaudoti vykdant eksperimentinius tyrimus. Trečioje baigiamojo darbo dalyje atliekami eksperimentiniai tyrimai, skirti nagrinėti konkrečius kreditinių kortelių sukčiavimus pagal pasirinktus modelius. Tikslas gauti kuo tikslesnį sukčiavimų atpažinimą, keičiant pagrindinius parametrus ir pritaikant

skirtingus metodus. Gauti rezultatai palyginami tarpusavyje ir apskaičiuojama finansinė nauda bankui.

1. DIRBTINIO INTELEKTO FINANSŲ SEKTORIUJE TEORINIAI ASPEKTAI

Bankų sektorius susiduria su iššūkiais prisitaikyti prie naujų technologijų ir inovacijų. Technologiniai pokyčiai vis dažniau yra siejami su dirbtinio intelekto sąvoka ir jos integravimu į vykdomą veiklą. Tačiau dėl didelių duomenų srautų ir senesnių sistemų tai nėra taip lengvai įgyvendinama kaip visai naujoms įmonėms, kurios iškart gali įtraukti dirbtinį intelektą į savo vykdomas veiklas. Dirbtinio intelekto įsitraukimas gali pagreitinti įvairius procesus, juos automatizuoti, aptikti anomalijas, prognozuoti ateities įvykius ir kurti klientų ar savo kasdienybę lengvinančius sprendimus, kurie bus pristatyti šiame darbe.

1.1. Dirbtinis intelektas ir jo veikimas

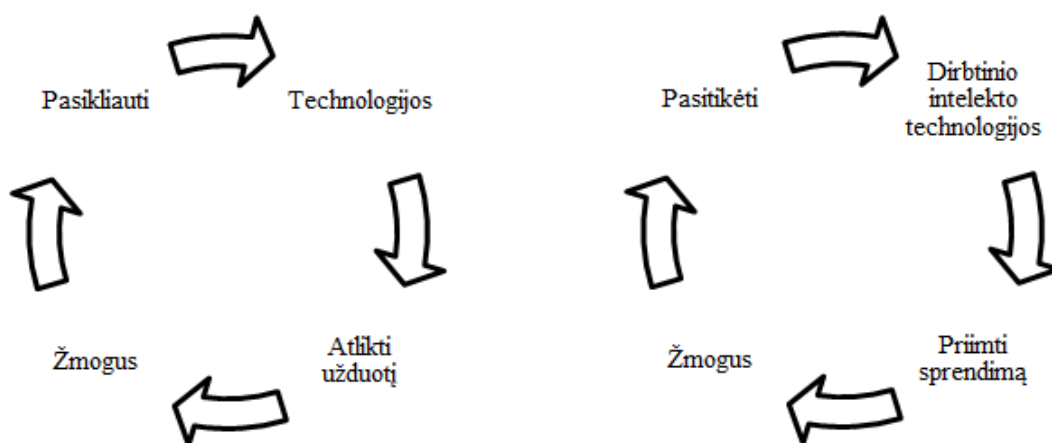
Daugiau nei prieš 90 metų, tiksliau 1921 metais, pirmą kartą pasaulis išgirdo žodį robotas. Dramaturgas K. Čapek'as savo premjeroje vaidavo robotų kūrimo procesą gamykloje, jų vergovę, jiems priskiriamus darbus, bet staiga įvyko karas tarp žmonių ir robotų, kuriame išgyveno tik vienas žmogus. Šio dramaturgo pėdomis seka ir vis daugiau kūrėjų, kurie vaizduoja metalinius robotus, bekūnius, skleidžiančius kompiuterinius, neaiškius ir kartais net bauginančius garsus. Mokslinės fantastikos kūriniuose ypač vaizdžiai pateikiama įtampa dėl pasitikėjimo robotais, žmonių išstūmimo ir gebėjimo juos pranokti. Bet ar tai pavyks realiame gyvenime, viskas priklauso nuo žmogaus ir jo sugebėjimo tvirtai valdyti šias naujas technologijas (Botsman, 2017).

20 amžiaus viduryje garsus mokslininkas ir kompiuterio pradininkas A. Turing'as parašė tuomet gana išsiskiriantį straipsnį pavadinimu „Ar mašinos geba mąstyti“ (Hutchens, 1996). Straipsnyje buvo pristatytas Tiuring'o testas, kuris byloja apie tai, ar galima sukurti tokią intelektualią mašiną, kurios veiksmai nebūtų atskiriami nuo paties žmogaus elgesio. Jo tikslas, kad tokia mašina gebėtų komunikuoti taip lyg tai prilygtų bendravimui su pačiu žmogumi. Su Tiuring'u dirbo ir kitas mokslininkas J. Good'as, kuris teigė, kad intelektinį testą praėjusios mašinos taps protingesnės už žmonių ir pradės kurti pačios dar protingesnes mašinas, kas gali sukelti ir didelį pavojų. Tad iki pat savo mirties jis laikėsi tos nuomonės, kad žmonės neturi persistengti kurdami per daug protingas mašinas, kurios pranoktų žmogų (Botsman, 2017). Tačiau ne tik šis žmogus kalba būtent taip apie protingas mašinas, tos panašios nuomonės ir E. Musk'as, B. Gates'as bei S. Hawking'as, kurie mano, kad kai sukurtas dirbtinis intelektas bus

savarankiškas, tobulins save pats vis greičiau, tai žmogus nespės varžytis su juo ir prognozuoti jo galimų veiksmų (Botsman, 2017).

Dirbtinis intelektas (angl. *Artificial Intelligence*), tai tam tikra mokslo šaka, kai yra kuriamos protingos programos, analizuojami vadinamieji protingi agentai (angl. *Intelligent Agents*). Protingi agentai, tai tokios mašinos, kurios mąsto ir bando pakelti savo šansus įveikti tam paskirtas užduotis (Mills & Stuffebeam, 2015; Ghosh, Tweedale, Nafalski, & Dollard, 2012). Agentų programos geba imituoti žmonių bei organizacijų tarpusavio santykių schemą - bendravimą, emocijas, bendradarbiavimą, tikslų siekimą. Dirbtinio intelekto pagalba yra kuriamos kompiuterinės sistemos, kurios geba imituoti procesus, kurie būdingi žmogui. Tokios mašinos siekia atkartoti žmogaus smegenų veikimą, bando atlikti veiksmus, kuriuos geba atlikti pats žmogus, pavyzdžiui, atskirti objektus, garsus ar žmogaus jausmus. Dirbtinis intelektas yra suvokiamas dažnai kaip kompiuterinė programa, veikianti kaip žmogus arba prilygstanti žmogaus mąstymui. Tam tikri šaltiniai teigia, kad kompiuterinė programa gali būti priskirta dirbtiniam intelektui, jei ji atpažįsta informaciją tiek prieš bandymą, tiek po jo, geba daryti naujas išvadas iš turimų duomenų ar vykstant apsimokymo procesui ji prisitaiko prie naujausių aplinkybių.

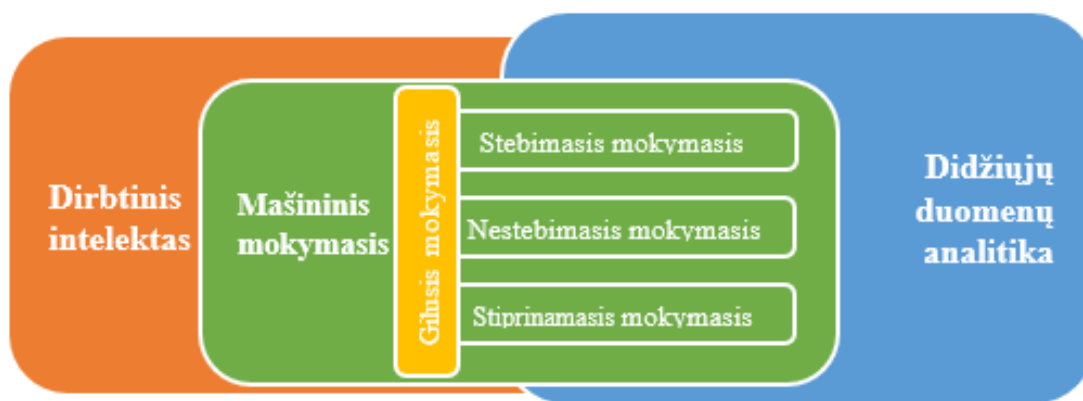
Visgi yra manoma, kad ateityje pasitikėjimas dirbtiniu intelektu tik augs, tai rodo grafikas, kuris pateiktas 1 pav. Dabar žmogus yra linkęs pasikliauti technologijomis, įvairios technologijos atlieka tam tikras užduotis, o ateityje žmogus taip pat pasitikės, bet pasitikės jau dirbtinio intelekto technologijomis, kurios pačios priims sprendimą ir pateiks reikiamą informaciją žmogui be jo pastangų. Tad ateityje dirbtinis intelektas priims vis daugiau sprendimų ir taps dar sumanesnis.



1 pav. Pasitikėjimas dirbtiniu intelektu dabar ir ateityje

Šaltinis: Pratuzaite ir Maknickienė (2019)

Dirbtinį intelektą galima būtų įvardyti ir kaip kompiuterinių sistemų plėtrą, kai yra atliekamos užduotys, susijusios su žmogaus intelektu, pavyzdžiui, kalbos supratimas ar automobilio vairavimas (Batson, 2018). Šis terminas gali dažnai klaidinti, bet jokių būdu tai nėra žmogaus protas, jis tik imituojamas ir intelektiniai rezultatai yra pritaikomi konkrečiose situacijose. Gana svarbus pokytis įvyko 2014-2015 metais, kai tam tikros programinės įrangos jau galėjo viršyti žmonių veiklos rezultatus, o seniau - negalėjo. Pavyzdžiui: balso, veido atpažinimas, ištartų žodžių perrašymas ir pan.



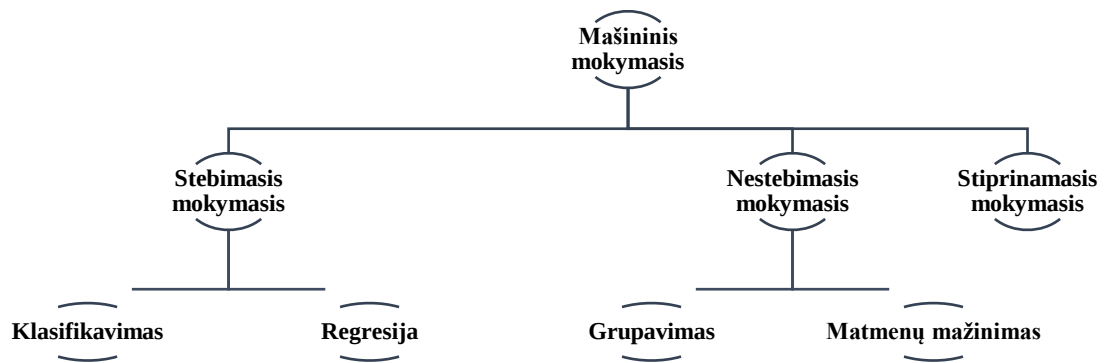
2 pav. Dirbtinio intelekto struktūra

Šaltinis: Financial Stability Board (2017)

Pavaizduota dirbtinio intelekto struktūra pateikia komponentus iš ko susideda ir nuo ko priklauso dirbtinis intelektas (žr. 2 pav.). Atsiranda ir naujų sąvokų, kurios bus pristatomos kituose poskyriuose. Visgi dirbtinio intelekto sąvoka dažnai yra siejama su mašininio mokymusi, o jo konkretūs įrankiai yra paremti statistikos metodais, kurie geba įvertinti didelį duomenų kiekį (Financial Stability Board, 2017).

1.1.1. Mašinos mokymasis

Mašininis mokymasis dažnai yra siejamas su optimizavimu ir automatizavimu, suskirstymu į konkrečias kategorijas. Egzistuoja keletas mašininio mokymosi algoritmų kategorijų - stebimasis mokymasis (angl. *Supervised learning*), nestebimasis mokymasis (angl. *Unsupervised learning*) ir stiprinamasis mokymasis (angl. *Reinforcement learning*), kurias sujungianti sąvoka yra gilusis mokymasis, o visą tai kartu sujungia didžiųjų duomenų analitika (Schmidhuber, 2015).



3 pav. Mašininio mokymosi struktūra

Šaltinis: Jha (2017)

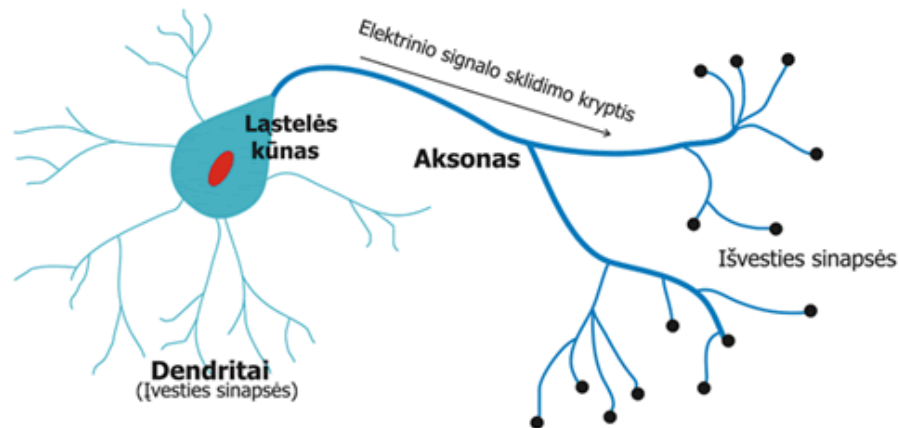
Dirbtinio intelekto nagrinėjimas dažniausiai prasideda nuo sąvokos mašinos mokymasis (angl. *Machine Learning*). Tai yra mokslo šaka, kai naudojant dirbtinį intelektą kuriamos programos, kurios pačios gerina savo rezultatus, pačios apsimoko, kai yra naudojama vis daugiau duomenų. Gana greitai tai tapo kaip dažniausiai naudojamas dirbtinio intelekto mokymosi būdas (Shwartz & David, 2014; Chollet, 2017). Mašininis mokymasis gali būti skirstomas įvairiais aspektais (žr. 3 pav.). Gana dažnai jis skaidomas į stebimąjį mokymąsi, nestebimąjį mokymąsi ir stiprinamąjį mokymąsi (Love, 2002; Hexagon, 2018).

Stebimasis mokymasis yra išskaidomas į šias kategorijas - klasifikaciją arba regresiją. Klasifikacijos tikslas yra naujus stebėjimus priskirti tam tikrai kategorijai, o regresijos - prognozuoti duomenis. Nestebimasis mokymasis neturi duomenų pažymėjimo ir būtent dėl tos priežasties yra galimos tik tokios funkcijos kaip grupavimas (angl. *Clustering*) ir matmenų mažinimas. Grupavimas gali būti naudingas, kai yra grupuojami tam tikri objektai į sluoksnius (klasterius), pavyzdžiui, klientai skirstomi pagal savo galimybes ar įpročius. Taisyklių ieškojimas gali turėti naudos, kai pavyzdžiui klientai įsigijo vieną prekę ar paslaugą, bet atitinkamai tuo pačiu įsigijo ir dar kitą. Kitas mokymosi metodas - stiprinamasis dažniau yra naudojamas akademiniiais tikslais ar kuriant kompiuterinius žaidimus.

1.1.2. Neuroniniai tinklai

Dirbtinio intelekto siekis yra imituoti žmogaus neuroninius tinklus (angl. *Neural networks*). Smegenys turi gebėjimą organizuoti savo struktūrines sudedamąsias dalis (neuronus) taip, kad būtų galima atlikti tam tikras funkcijas (žr. 4 pav.). Pavyzdžiui, žmogaus

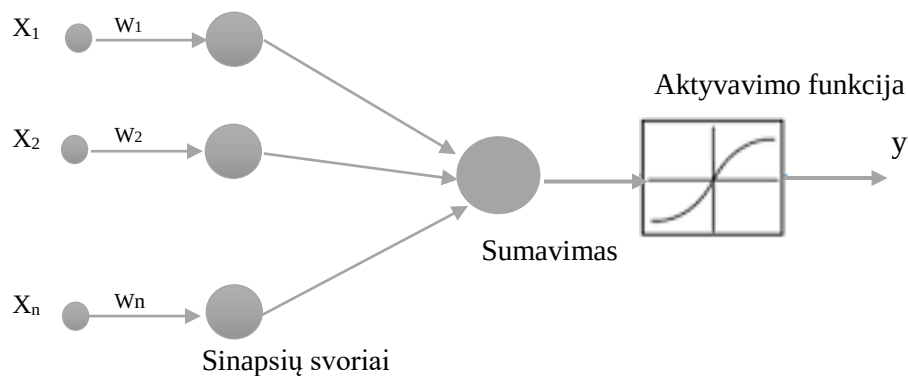
regėjimas atlieka informacijos apdorojimo funkciją tam, kad apibūdintume mus supančią aplinką.



4 pav. Žmogaus neuronas

Šaltinis: Makūnaitė (2006)

Dirbtinis neuroninis tinklas (angl. *Artificial neural network*) - tai tokia struktūra, kuri yra sudaryta iš daugiau dirbtinių neuronų (Cilimkovic, 2015). Šie tinklai išties primena žmogaus smegenis, nes informacija yra gaunama per mokymosi procesą ir patys neuronai geba prisiminti pateiktą informaciją, ją kaupti, prisitaikyti ir mokytis (Haykin, 2009; Serrano, 2012). Galima sakyti, kad paties roboto smegenys yra sudarytos iš programinės įrangos tinklo, kurio tikslas vis atkartoti žmogaus smegenyse esančias neuronų jungtis.



5 pav. Dirbtinio neurono modelis

Šaltinis: Haykin (2009), Lu (2017)

Neuroniniai tinklai yra sudaryti iš neuronų, kurie vienas su kitu ir sujungti, ir pagal jungtis perduoda signalus (Johnston, 2018). Veikimas susideda iš to, kad tinklas priima įvesties

signalą ir kelis kartus apdoroja, kas suteikia gilesnį ir naudingesnį mokymąsi. Šių metodų naudojimas reiškia, kad algoritmas gali pagerinti užduoties atlikimą vis kartodamas naudojamus duomenis. 5 paveiksle vaizduojamas dirbtinio neurono modelis, iš kurio galima sudaryti dar didesnę neuroninę tinklą.

Kiekviena įėjimo jungtis (sinapsė) turi savo svorį, žymimą w_{kj} , kuris parodo įėjimo jungties svarbą. Signalas x_j prie sinapsės j įvesties yra prijungtas prie neurono k , kai yra padauginama iš sinapsės svorio w_{kj} . Galiausiai po to visi įvesties signalai yra sudedami ir pasiekama aktyvavimo funkcija, kuri skirta apriboti neurono išėjimo amplitudę ir įgyvendinti išėjimą, žymimą y_k (žr.(1) formulę.).

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (1)$$

Neuroniniai tinklai geba suteikti šias galimybes (Haykin, 2009):

1. Netiesiškumas (angl. *Non-linearity*). Ši savybė reiškia, kad dirbtinis neuronas gali būti arba linijinis, arba ne. Neuroninis tinklas yra sudarytas iš netiesinių neuronų susijungimo ir pats savaime yra nelinijinis.
2. Įvesties - išvesties susiejimas (angl. *Input-Output Mapping*). Kiekviena situacija susideda iš unikalaus įvesties signalo ir atitinkamo norimo atsakymo.
3. Adaptacija (angl. *Adaptivity*). Neuroniniai tinklai turi gebėjimą pritaikyti savo sinapses, keičiantis aplinkai.
4. Įrodantis atsakymas (angl. *Evidential Response*). Neuroninis tinklas gali būti suprojektuotas taip, kad būtų pateikta informacija ne tik kokių konkretų modelių pasirinkti, bet ir apie pasitikėjimą gautu sprendimu. Tai gali būti naudojama siekiant atmesti dviprasmiškus modelius ir taip patobulinti klasifikaciją.

Tad šiuo metu dirbtiniai neuroniniai tinklai yra daug kur taikomi, pavyzdžiui, duomenų klasifikavimui, optimizavimui ir prognozei atlikti. Nors dirbtinis intelektas turi gebėjimą mokytis tarsi tikros smegenys, bet labiau vis tiek yra orientuotas į atkartojimą, duomenų apdorojimą ir bandymą mokytis iš to. Pavyzdžiui, telefonai „iPhone“ turi kalbos vertimo ir atpažinimo programą „Siri“, tačiau ji tikrai ne visada gali teisingai suprasti pasakomus žodžius. Pirmiausiai dėl to, kad žmonės turi skirtingus akcentus, gali kalbėti skirtingu garsumu ir pati aplinka gali skleisti pašalinį triukšmą, kas trukdo programai teisingai atsakyti į užduodamus klausimus. Būtent dėl to mokslininkai ir programuotojai nuolat tobulina ir siekia pagerinti

neuroninių tinklų veikimą, kad būtų pasiekiami vis geresnių rezultatų ir programos vis mažiau klystų.

1.1.3. Gilusis mokymasis

Gilusis mokymasis (angl. *Deep learning*), tai tam tikras mašininio mokymosi tipas (Chollet, 2017). Šis mokymasis yra vadinamas giliuoju vien dėl to, kad yra sudarytas iš daugiau neuroninių sluoksnių ir kuo daugiau duomenų yra naudojama, tuo geresni rezultatai yra pasiekiami (žr. 6 pav.). Būtent tai leido sukurti be pilotės mašinas, garsų ir vaizdų atpažinimo programas, geresnę rezultatų paiešką sistemose.



6 pav. Giliojo mokymosi grafikas

Šaltinis: Brownlee (2016)

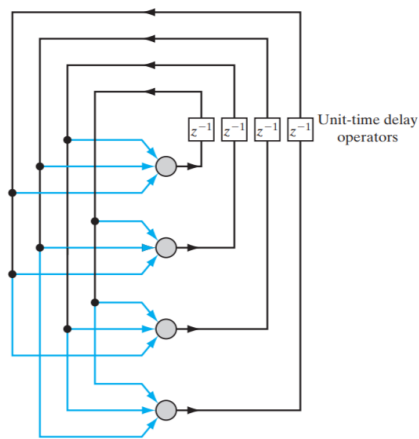
1.1.4. Neuroninių tinklų architektūros

Dirbtinio neuroninio tinklo architektūra gali apibrėžti, kaip veikia keli neuronai, kurie yra išdėstomi vienas kito atžvilgiu (Pumsirirat & Yan, 2018). Tokios architektūros yra struktūrizuotos kompozicijos. Tam tikros architektūros mokymas apima taikymą suderinti veiksmus siekiant sureguliuoti neuronų svorius ir slenksčius. Neuroninių tinklų architektūros pagal disponavimo funkcija gali būti skirstomos į vieno sluoksnio, daugiasluoksnius, pasikartojančius ir apraizgytus tinklus (Silva, Spatti & Flauzino, 2017). Taip pat dirbtiniai neuroniniai tinklai gali būti suskirstyti į tam tikrus sluoksnius, kurie yra įvardijami kaip (Silva et al., 2017):

1. Įvesties sluoksnis (angl. *Input layer*), kuris yra atsakingas už informacijos, signalų, funkcijų arba informacijos priėmimą.

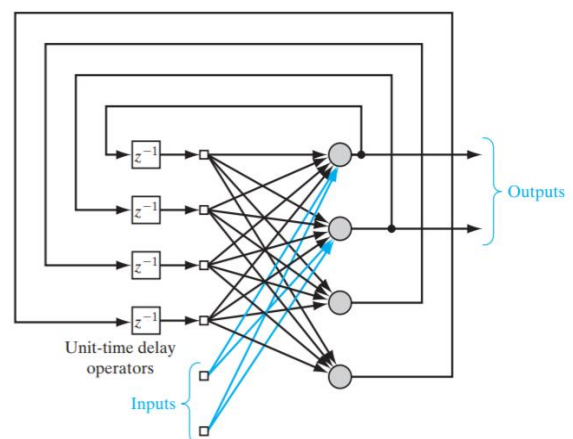
2. Paslėpti, tarpiniai arba nematomi sluoksniai (angl. *Hidden layer*), kurie yra atsakingi už duomenų ištraukimą, susiję su nagrinėjamu procesu ar konkrečia sistema. Šie sluoksniai atlieka daugybę vidinių procesų.
3. Išvesties sluoksnis (angl. *Output layer*), kuris yra atsakingas už gamybą ir pateikiami galutiniai tinklo išėjimai, kurie gaunami apdorojant atliekami ankstesniuose sluoksniuose iš neurono.

Remiantis J. Brownlee'o straipsniu, yra teigiama, kad algoritmai gali pasiekti tokią ribą, kai jų rezultatai nebe gerės. Tačiau giliojo mokymosi algoritmai, pavyzdžiui, gilūs neuroniniai tinklai (angl. *Deep Neural Networks*) ir konvoliuciniai neuroniniai tinklai (angl. *Convolutional Neural Network*) veikia kiek kitaip, nes jie gali pateikti geresnius rezultatus, kai dirbtinis neuroninis tinklas gauna vis daugiau ir daugiau duomenų.



7 pav. Rekurentiniai tinklai be savarankiško grįžimo kilpų ir be paslėptų neuronų

Šaltinis: Haykin (2009)



8 pav. Rekurentiniai tinklai be paslėptų neuronų

Šaltinis: Haykin (2009)

Konvoliuciniai neuroniniai tinklai yra giliojo mokymosi rūšis, kuri dažniausiai naudojama siekiant atpažinti vaizdus, rasti objektus ir atskirti garso įrašus (Veen, 2016; Maršalkytė, 2018). Nuolat yra siekiama patobulinti mokymosi metodus, naudoti vis kitokias technikas, dar daugiau duomenų, kad mokymasis vyktų vis efektyviau. Žodis „konvoliuciniai“ reiškia, kad šie neuroniniai tinklai vykdo tam tikrą funkciją, kuri yra vadinama konvoliucija (Goodfellow, Bengio & Courville, 2016). Šie neuroniniai tinklai gali naudoti bendruosius svorius, skirtus duomenims, naudoti vietines jungtis, daugybę sluoksnių ar keisdami parametrų skaičių, atitinkamai sumažinti pačio mokymosi trukmę. Šie tinklai išsiskiria ir tuo, kad nekaupia atminties, todėl visus duomenis reikia pateikti iš karto ne taip, kaip su rekurentiniais tinklais.

Rekurentiniai neuroniniai tinklai (angl. *Recurrent Neural Network*) - tai mašininio mokymosi dar vienas tipas, kai neuronai yra sujungiami taip, kad sudaro orientuotą grafą (Haykin, 2009). Rekurentinis reiškia, kad tinklas yra sudarytas iš rekursyvių neuronų, o juose esantys algoritmai gali kreiptis patys į save. Kiek žemiau yra pateiktos dvi skirtingos neuroninių tinklų architektūros, kurios vaizduoja kaip rekurentiniuose tinkluose duomenys įeina į neuroną, kur įvykdomas algoritmas su tais duomenimis ir kaip išeina (žr. 7, 8 pav.). Rekurentiniai neuroniniai tinklai geba nuolatos kaupti ir atnaujinti visą informaciją.

1.1.5. Algoritmai

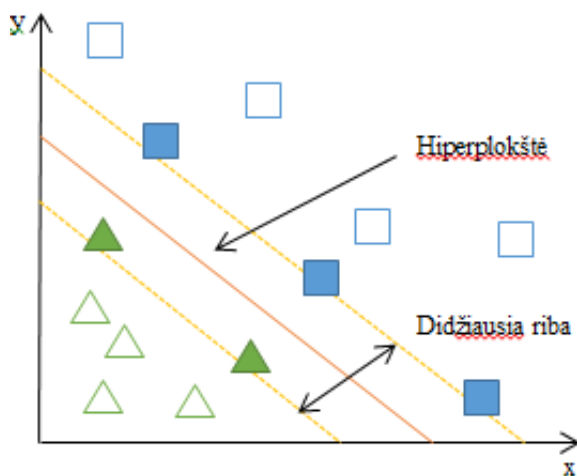
Nėra vieno konkretaus algoritmo, kuris išspręstų visas problemas, todėl yra daugybę kitų, pritaikomų pagal konkrečią problemą. Tinkamo algoritmo pasirinkimą dažnai lemia tokie veiksniai kaip duomenų apimtis, kintamųjų tipas, norimi tikslai, trukmės apribojimai ir struktūra, tačiau vis tiek gana sunku nusakyti, kuris algoritmas yra pats geriausias, todėl ypač svarbu yra tinkamai apsibrėžti norimą problemą. Pagal įvairius mokslinius straipsnius buvo pasirinkti tokie algoritmai, kurie sprendžia problemas, susijusias su bankų sektoriais ir jiems aktualiomis problemomis, pavyzdžiui, sukčiavimų atpažinimas.

Sprendimų medis (angl. *Decision tree*) vienas iš dažniausiai naudojamų klasifikavimo algoritmų, kurio tikslas duomenis išskirstyti į du ar daugiau panašių rinkinių (grupių) pasirenkant tinkamiausius požymius (Song & Lu, 2015). Sprendimų medį sudaro trys pagrindiniai komponentai: medžio viršūnė, kuriai parenkamas tinkamiausias požymis prognozei; šaka, kurios sekantys požymiai yra testuojami; lapas, kuris atspindi galutinį rezultatą (Omar, 2013). Sprendimų medžio sudarymo eiga:

- visi objektai (įrašai) įtraukiami į medžio viršūnę;
- objektai toliau skirstomi į grupes pagal požymius;
- objektai grupuojami tol, kol pasiekiami tam tikri medžio sudarymo pabaigos kriterijai.

Sprendimų medžiai sudaromi iš viršaus į apačią kiekviename žingsnyje pasirenkant tokį požymį (metriką), kuris padalintų objektus geriausiu būdu. Yra naudojami skirtingi skaičiavimo būdai matuojant „geriausius“ paskirstymus, o įprastai grupėje yra išmatuojamas priklausančių objektų homogeniškumas. Sprendimų medis tikrina visus požymius norint nustatyti geriausią objektų suskaidymo būdą pagal pasirinktas metrikas.

Atraminų vektorių metodas (angl. *Support vector machine*) - tai algoritmas, priskiriamas prižiūrimajam mašininio mokymosi tipui. Jis atlieka klasifikavimo užduotis ir priklauso nuo sprendimo planų, kurie atskiria duomenis į skirtingas klases (dažnai į teigiamas ir neigiamas) (Seeja et al., 2014).

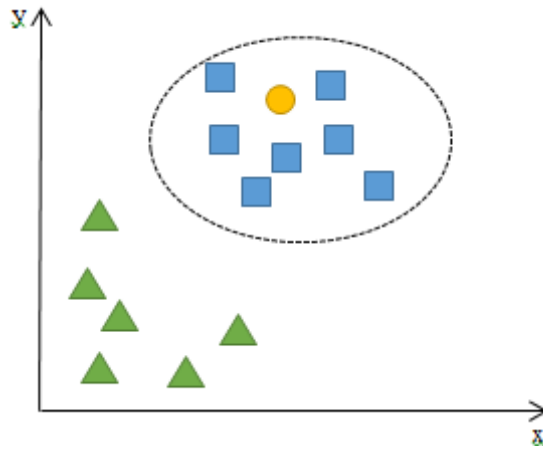


9 pav. Vektorių palaikymo mašinos modelio pavyzdys

Šaltinis: Pratuzaite ir Maknickienė (2019)

Šio metodo tikslas yra atrasti n -dimensijų erdvėje hiperplokštes (angl. *Hyperplane*), kurios klasifikuoja duomenis (žr. 9 pav.) (Omar, 2013). Siekiant atskirti dvi duomenų klases, yra daugybė galimų hiperplokščių, kurias galima pasirinkti, o tikslas yra surasti tokią plokštę, kuri turėtų didžiausią ribą tarp abiejų duomenų klasių (Gandhi, 2018). Tai yra svarbu tam, kad būtų nustatyta tiksli duomenų klasė.

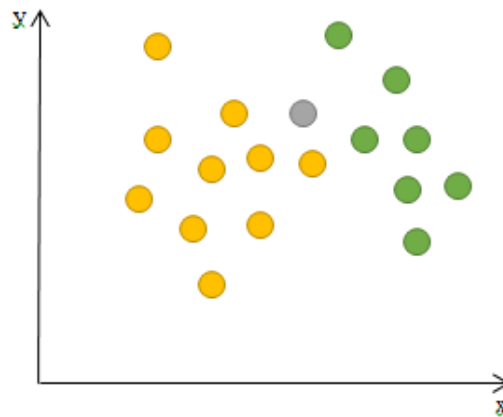
K-artimiausio kaimyno metodas (angl. *K-nearest neighbor*) - tai toks algoritmas, kuris klasifikuoja pagal artimiausius k (nurodytų kaimynų skaičius) kaimynus (žr. 10 pav.) (Shwartz & David, 2014). Šis metodas kaip ir atraminų vektorių metodas priklauso prižiūrimajam mokymuisi. Pirmiausiai yra suklasifikuojami artimiausi kaimynai į naujus, pagrįstus ir panašius rezultatus. Kiekvienas naujas duomuo lyginamas su esamais duomenimis naudojant atstumo metriką ir artimiausią esamą duomenį siekiama priskirti tam tikrai klasei (Seeja, 2014).



10 pav. K-artimiausio kaimyno modelio pavyzdys

Šaltinis: Pratužaitė ir Maknickienė (2019)

Naivusis Bajeso klasifikavimo metodas (angl. *Naive Bayes*), tai dar vienas algoritmas, kuris naudoja mokymosi duomenų rinkinį tikslinių klasių prognozavimui (Seeja, 2014; Xu, 2018). Šis klasifikavimo metodas yra tarsi tikimybinis metodas, naudojantis klasės informaciją iš mokymosi duomenų rinkinio ir yra skirtas numatyti būsimas klases pagal tikimybes (Morrison & Rao, 2017). Pavyzdžiui, toliau pateiktame grafike yra pavaizduoti skirtingų klasių objektai ir kai atsiranda naujas objektas be klasės, yra skaičiuojama tikimybė, kuriai klasei jis yra priskiriamas (žr. 11 pav.).



11 pav. Naive Bayes klasifikavimo metodo pavyzdys

Šaltinis: Pratužaitė ir Maknickienė (2019)

Paslėptas Markovo modelis (angl. *Hidden Markov model*) - tai toks stochastinis modelis, kai modelio objektai yra paslėpti ir gali grąžinti rezultatus, kurie yra stebimi ir analizuojami (Ghahramani, 2001). Šis modelis išsiskiria iš kitų, nes nesprenžia klasifikavimo

problemų. Jis naudoja esamus rezultatus, gautus iš paslėptų duomenų ir naudodamas tikimybinės funkcijas, nusprendžia apie objekto būseną (Morrison & Rao, 2017). Kaip pavyzdį galime įsivaizduoti, kad mums reikia nuspręsti, koks yra dabar oras, bet viskas iš ko galime nuspręsti tai iš draugo, kuris ateina į namus ir jis turi skėtį arba ne. Įsivaizduokime, kad yra trys galimos baigtis: saulėta, apsiniaukę ir lyja. Mes kaupiame stebėjimų duomenis, iš jų mes galime apsiskaičiuoti reikalingus parametrus ir pagal (2) formulę galime apsiskaičiuoti sekančios dienos oro tikimybę.

$$P(q_1, \dots, q_t | o_1, \dots, o_t) \propto \prod_{i=1}^t P(o_i | q_i) \prod_{i=1}^t P(q_i | q_{i-1}) \quad (2)$$

Ši formulė yra skirta gauti tikimybę iš stebėtų duomenų ir galimų baigčių.

čia:

$P(q_i | q_{i-1})$ - perėjimo tikimybė;

$P(o_i | q_i)$ - emisijos tikimybė;

$P(q_i)$ - pradinės būsenos tikimybė;

o - stebėjimo duomenys (pavyzdžiui - atėjo su skėčiu arba be jo);

q - baigtys (pavyzdžiui - saulėta, apsiniaukę, lyja);

t - stebėjimų skaičius.

Šis modelis naudojamas ir mokėjimų sukčiavimams aptikti, kai yra nagrinėjamos kortelės savininko išlaidos (Bhusari & Patil, 2011). Kiekvienas kortelės savininkas turi savo unikalų modelį, kuriame atsispindi visi sandoriai, išsami informacija apie įsigytas prekes ir paslaugas tam tikromis dienomis. Tai veiksmingas būdas kovoti su sukčiavimu, o jei įvyksta nukrypimas, tada pati sistema gali sustabdyti sandorį. Šiame metode yra naudojami skirtingi lygmenys, vienas yra paslėptas, o kitas yra matomas visiems. Pavyzdžiui, bankinis pavedimas tai yra paslėpti duomenys, bet bankinis pavedimo rezultatas yra matomas ir iš to mes galime nuspręsti, ar tai sukčiavimo atvejis ar ne.

1.2. Dirbtinio intelekto panaudojimas

Dauguma jau net nebesusimąstome, tačiau dirbtinis intelektas yra gana plačiai naudojamas mūsų kasdieniame gyvenime. Pavyzdžiui, elektroninių parduotuvių sistemos, kurios seka pirkėjų paiešką ir tai, kas yra perkama, atitinkamai pagal tai, vėliau pateikia tam

tikrus pasiūlymus. Taip pat įvairūs asistentai, kurie bendrauja tarsi žmonės, gali atsakyti į reikiamus klausimus ar atlikti tam tikrus veiksmus, susijusius su programa. Pavyzdžiui, „Sir“ naujojiama „Apple“ įrenginiuose, „Cortana“ naudojama „Microsoft“ įrenginiuose, o „Alexa“ naudojama „Amazon“. Tad dirbtinis intelektas gali būti panaudojamas įvairiausiose sektoriuose: transporte, mažmeninėje prekyboje, reklamoje, finansuose, gamyboje, klientų aptarnavime, medicinoje ar kariuomenėje.

Vienas iš dažniausių dirbtinio intelekto panaudojimo būdų - automatizuoti veiklas ir įvairiausias užduotis, apdoroti didelius kiekius duomenų. Tai gali padaryti ir žmonės, bet dėl dirbtinio intelekto yra taupomi kaštai ir darbuotojų laikas, susikoncentruojama labiau į kitas sritis. Tačiau dirbtinio intelekto įsiskverbimas į bankų sektorių iki šiol yra šiek tiek ribotas. Atskirų duomenų rinkinių ir konfidencialių duomenų rizika lemia lėtesnę integraciją. Bet kai internetinė bankininkystė ir mobilioji bankininkystė tampa vis populiareesnė, galime tikėtis, kad dirbtinis intelektas dar sėkmingiau įsitrauks į vartotojų kasdienę veiklą.

Naujųjų technologijų atsiradimas davė svarbių pakeitimų finansų sektoriui ypač per pastarąjį dešimtmetį (Shook et al., 2018). Dėl mašininio mokymosi ir dirbtinio intelekto investuotojai ir vartotojai gauna galimybę naudotis novatoriškomis priemonėmis, naujomis produktų rūšimis. Finansų srityje dirbtinis intelektas dažnai apima tokias sritis kaip įvairių finansinių procesų prognozavimą, produktų kūrimą ir investicinių portfelių sudarymą. Tam yra kuriami imituojantys modeliai ir agentų sistemos, kurios geba pakartoti ir prognozuoti finansinių rinkų elgseną, pritaikyti tinkamą strategiją. Bet pačios funkcijos įtraukus dirbtinį intelektą nesikeičia, siekiama ir toliau kuo efektyviau vykdyti esamas ar būsimas veiklas ir minimizuoti galimas rizikas. Dirbtinis intelektas gali būti panaudojamas ir šiose srityse: buhalterija, auditas, reklama, valiutų keitimas, investicijos, klientų aptarnavimas, korupcija, kredito analizė, pinigų plovimas, sukčiavimo aptikimas ir pan.

Duomenys bankui yra svarbūs beveik visose srityse, susijusiose tiek su indėliais, tiek su investicijomis ar turto valdymu. Tačiau duomenų valdymas be žmogaus dalyvavimo bankui suteikia galimybes pagerinti procesų greitį, tikslumą ir efektyvumą. Dirbtinio intelekto taikymas gali būti skirstomas į šias kategorijas (Kaya, 2019):

- į klientus orientuotos „front office“ programos;
- į operacijas orientuotos „back office“ paraiškos;
- prekyba ir portfelio valdymas;
- normų laikymasis.

Remiantis audito kompanijos „Deloitte“ straipsniu dirbtinis intelektas geba pagerinti įvairias galimybes siekiant:

- atpažinti modelius;
- numatyti ateities įvykius;
- sukurti tinkamas taisykles;
- priimti tikslingus sprendimus;
- komunikuoti su tiksline auditorija.

Tad tokie įsivyraujantys pokyčiai keičia bankų veiklos plėtrą, labiau reikia prisitaikyti ir efektyviau išnaudoti technologijas.

Gana dažnai dirbtinio intelekto naudojimas finansų srityje yra susiję su pinigų apsauga, siekiant apsisaugoti nuo sukčiavimo atvejų. Kompiuterinės sistemos, kurios kovoja su sukčiavimais, veikia taip, kad yra stebimos kliento išlaidos ir gaunami signalai, jei tik įvyksta kokie nors netikėti ir įtarimus keliantys mokėjimai. Pasak Oksfordo universiteto profesoriaus D. Kroening'o, bankai patiria didelį spaudimą iš aplinkos, įmonių, kurios neturi senos programinės įrangos ir tiek duomenų kaip bankai (Allan, 2018). Pavyzdžiui, Fintech konkurentai tokie kaip „Revolut“ ar „Monzo“ greičiau įtraukia dirbtinį intelektą į savo darbus, siūlo klientams paprastesnes programas, o bankai susiduria su begaline rizika patenkinti savo klientus ir jų neprarasti. Šiaurės Amerika yra laikoma konkurencingiausia ir sparčiausiai besivystančia dirbtinio intelekto technologijų rinka finansų pramonėje. Nuo 2011 iki 2018 metų buvo įkurtos daugiau kaip 3330 naujų, technologijomis paremtų, įmonių, teikiančių finansines paslaugas, iš kurių apie 40 % yra orientuotos į bankus ir kapitalo rinkas.

Procesų automatizavimas yra vienas iš pagrindinių dirbtinio intelekto plėtrą skatinančių veiksnių. Dėl dirbtinio intelekto sistemos gali atlikti dar sudėtingesnius automatizavimo procesus ir efektyviau išnaudoti resursus. Bet sukčiavimas yra kone didžiausia finansų įstaigoms nerimą kelianti sritis ir ateityje gali sudaryti didžiausias išlaidas informacinių technologijų kompanijoms. Taigi, tikimasi, kad siekiant išvengti sukčiavimų, bus įdėbinta daugiau Fintech įmonių arba bus didesnis jų bendradarbiavimas su bankais. Nesąžininga veikla finansų srityje vystosi jau kelis dešimtmečius. Tačiau augant kibernetinių atakų skaičiui, sukčiavimas įgavo daugiau virtualių formų, tad nuo to apsisaugoti darosi tik sunkiau. Augančios technologijos ir skaitmeniniai kanalai tampa svarbiausiais bankų klientų pasirinkimais, tad bankams yra didesnis poreikis naudoti priemones nuo sukčiavimo ir taip apsaugoti savo klientus.

Vienas dažniausių dirbtinio intelekto įrankių bankuose yra automatizuoti asistentai, kurie klientams teikia greitą pagalbą norint išspręsti klausimus per internetinę pranešimų sistemą, naudojantis asmeniniais kompiuteriais ar išmaniais telefonais. Virtualus asistentas yra

dirbtiniu intelektu valdoma sistema, padedanti pateikti klientams automatizuotus atsakymus. Naudodamiesi mašininio mokymosi technologijomis, pokalbių svetainės nuolat tobulėja, atsižvelgiant į jų sugebėjimą tiksliai nustatyti kliento problemą ir atsakyti į klausimus, todėl šiandien jie geba atpažinti daugybę bendrų klausimų, kurių gali paklausti klientas. Siekiant užtikrinti dirbtinio intelekto veikimą, reikia nemažo kiekio duomenų, nes dėl jų bus grindžiami tam tikri sprendimai. Tad vis dažniau finansinės kompanijos stengiasi kuo geriau pažinti savus klientus, darydamos apklausas ir pagal tai sprendamos, kokias paslaugas gali jiems suteikti.

Įvairūs bankai pasaulyje jau naudoja arba ruošiasi ateityje plėsti dirbtinio intelekto technologijas, pavyzdžių:

1. „JPMorgan Chase“ bankas naudoja programą pavadintą „LOXM“, kuri efektyviau ir greičiau vykdo klientų mokėjimus ir naudoja „CoiN“ platformą, kuri analizuoja teisinius dokumentus ir sugeneruoja kredito sutartis net per kelias sekundes. 2019 metais jie padidino savo technologijoms skirtą biudžetą iki 11,4 milijardo JAV dolerių, o tai tik parodo, kad jie stengiasi tobulėti, plėsti savo produktus ir projektus (Sennaar, 2019).
2. Vokietijos skaitmeninis bankas „Number26“ naudoja programą pavadintą „Pulse26“, kuri analizuoja klientų istoriją ir prognozuoja jų galimus veiksmus.
3. Britanijos bankas „Atom Bank“ naudoja „Virtual Agent“ bendraujančią su klientais ir atsakančią į jų užduodamus klausimus iki tam tikro lygio. Jei nebegali atsakyti į užduodamus klausimus, nukreipia tiesiogiai pas banko specialistą.
4. Amerikos bankas taip pat naudoja asistentą pavadinimu „Erica“, kuris atlieka kasdienes sandorius, susijusius su finansiniais poreikiais, padeda geriau suprasti aspektus apie pinigų valdymą ir finansinį švietimą, planuoja susitikimus tiesiogiai su banko specialistais, gali peržiūrėti sąskaitas ir planuoti mokėjimus. Jis naudoja dvi dirbtinio intelekto formas - natūralios kalbos apdorojimą, kad būtų suprantama kalba ir mašininį mokymąsi, kad būtų galima rinkti įžvalgas pagal gautus kliento duomenis, kuriuos galima būtų paversti naudingais patarimais (Crosman, 2018). „Bank of America“ virtualių asistentų srityje yra pirmaujantis tarp savo konkurentų. „Erica“ gali pasigirti turinti net 6 milijonus vartotojų nuo 2019 metų (Sennaar, 2019).
5. Indijos banke yra naudojamas virtualus asistentas „SIA“, kuris atlieka kasdienes bankines užduotis ir užklausas. Šis asistentas vienas iš didžiausių pasaulyje, orientuotų į dirbtinio intelekto diegimą ir plėtrą.

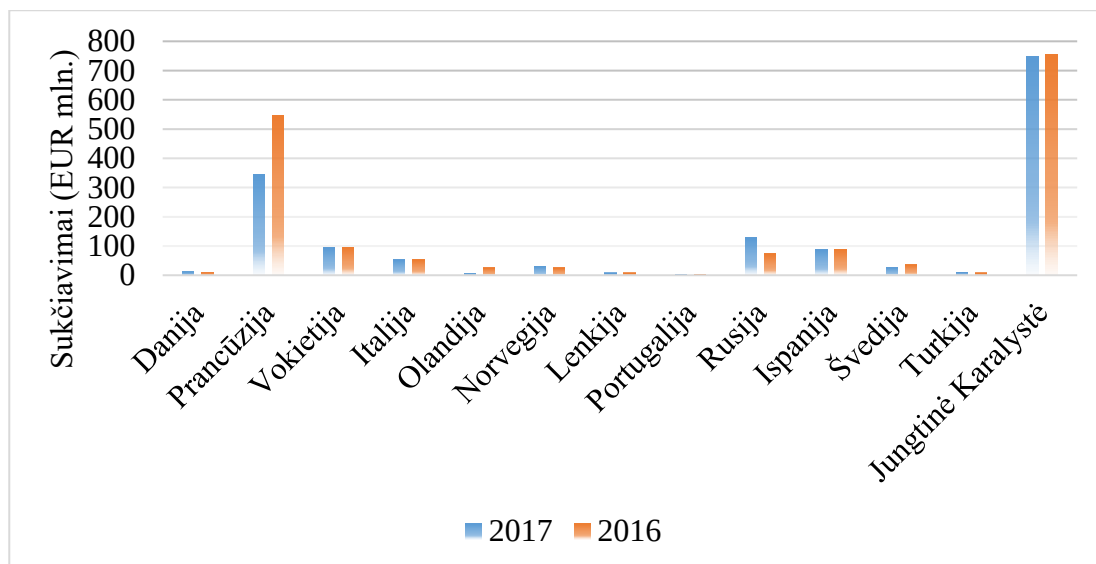
6. Skandinavijos bankai taip pat naudoja dirbtinį intelektą, kad būtų sumažintas krūvis darbuotojams. Pavyzdžiui: „Aida“ naudojami „SEB“ banko klientai, „Nina“ – „Swedbank“, „Nova“ - „Nordea“ banke. Visi jie yra pavadinti moterų vardais, o to priežastis, kad klientai jaučiasi patogiau, kai bendrauja su moterimis finansiniais klausimais.

Įsitraukimas naudoti dirbtinį intelektą bankų sektoriuose tikrai yra gana sparčiai didėjantis. Pagal atliktą „Global Digital IQ Survey“ apklausą, net 54 proc. organizacijų investuoja į dirbtinį intelektą, o šis rodiklis taip išaugo per tris metus net iki 63 proc. Tačiau bankai susiduria su dideliais iššūkiais, tai ir senesnės sistemos, kurias reikia pritaikyti, didžiulės duomenų bazės, klientų jautrumas, procesų patvirtinimas, pasitikėjimas ir darbuotojų pasikeitimai, kurie gali sutrikdyti darbų eigą.

Pastaraisiais metais vis didėjanti vartotojų paklausa skatina bankus vis labiau įdiegti skaitmenizavimą. Internetas ir mobilioji bankininkystė tampa įprastu įrankiu (kanalu) klientams. Vis dažniau naudojant šiuos kanalus sudaroma galimybė greitų mokėjimų įvykdymui, patogesniai ir greitesniai įvairių sprendimų ir galimybių naudojimui, tačiau tai didina tikimybę nusikaltėliams pasinaudoti klientu ir jo duomenimis.

Europoje yra vykdomas prevencijos mechanizmas nuo sukčiavimo (Fico, 2017). Sukčiavimo išvengimui yra naudojamas bendradarbiavimas su kitais pramonės atstovais ir jų duomenimis, taip pat naudojamas mašinų mokymasis ir dirbtinis intelektas. Įvairūs algoritmai skaičiuoja galimus kliento veiksmus pagal jo vartojimą, stengiamasi atpažinti tokias situacijas, kurios yra nebūdingos tam klientui ir taip atskleisti, kad tai yra sukčiavimas. Tokių technologijų įsitraukimas į šią sritį turi didžiulės naudos, nes klientų duomenys yra apsaugomi, sumažinama rizika pinigų plovimui vykti ir sutaupomi kaštai.

Gana opi problema bankų sektoriuje yra susijusi su mokėjimo kortelių sukčiavimais, kai yra klastojami kortelių duomenys, kortelės yra vagiamos ar vykdomi nesąžiningi atsiskaitymai internetu. Tyrimus atliekanti kompanija „Fico“ pateikia statistiką, vaizduojančią pasikeitimus Europoje dėl mokėjimo kortelių sukčiavimo. Sumos, kiek pinigų yra prarandama, yra ganėtinai aukštos, o didžiausi pradimai egzistuoja Didžiojoje Britanijoje, Prancūzijoje ir Rusijoje, atitinkamai 747, 346 ir 128 milijonai eurų (žr. 12 pav.).



12 pav. Sukčiavimo statistika

Šaltinis: sudarytas autorės, remiantis Fico (2017)

Pagal Europos centrinio banko ataskaitą, bendri sukčiavimo nuostoliai, susiję su mokėjimo kortelėmis 2016 metais siekia 1,8 milijardo eurų. Kortelių sukčiavimo vertė sumažėjo 0,4 procento palyginus su 2015 metais, o 35 procento lyginant su 2012 metais. Tačiau visos sandorių vertės dalis sumažėjo nuo 0,042 procentų 2015 metais iki 0,041 procento 2016 metais. Kortelių sukčiavimo nuostoliai pasauliniu lygmeniu siekia 22,8 milijardo JAV dolerių 2016 metais, o tai yra 4,4 procento daugiau nei 2015 metais. Visi šie duomenys tik SEPA erdvei priklausančių šalių rezultatai 2012-2016 metų laikotarpiu.

Didžiausi iššūkiai dėl kortelių sukčiavimo yra susiję su CNP (angl. *Card not present*) sukčiavimais, tarpvalstybiniais sukčiavimais ir klastojimais ne EMV kortelėmis. CNP yra toks sukčiavimas, kai yra netiesiogiai naudojama mokėjimų kortelė, kuri daugiausiai apima internetinį sukčiavimą. Tai yra didžiausia kortelių sukčiavimo rūšis, sudaranti 73 proc., remiantis 2016 m. duomenimis. EMV, tai technologijų standartas, susijęs su mokėjimais, kai yra atliekamos operacijos su debeto ir kredito kortelėmis. Šis standartas suteikia operacijų saugumą ir mažina apgavysčių pavojų. Operacijoms atlikti yra naudojamas unikalus kodas, po to vykdomas patikrinimas yra siunčiamas bankui su kortelės informacija. Šis tapatybės patvirtinimo būdas sunkina galimybes nusikaltėliams pasinaudoti kortelės savininko duomenimis.

1 lentelė. Sukčiavimo atvejai Europoje ir tam tikrose šalyse

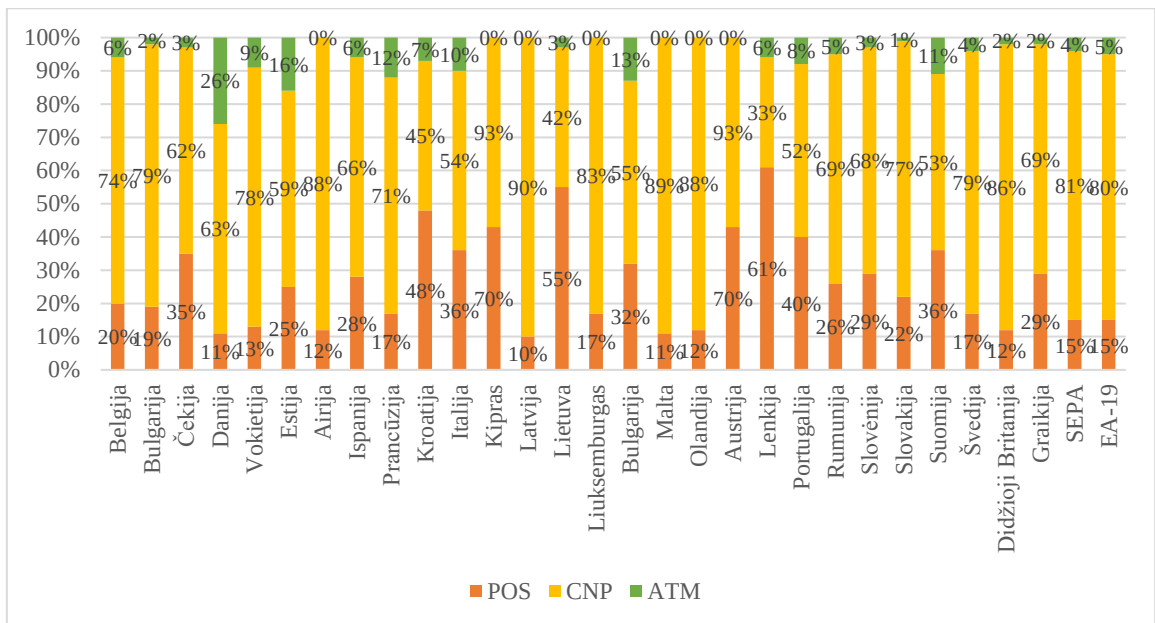
	Europos sąjunga (2016)	Prancūzija	Jungtinė Karalystė	Olandija	Danija	Norvegija	Švedija
Populiacija (mln.)	512,5	67,1	66,0	17,1	5,8	5,3	10,1
Kortelių skaičius (mln.)	812,4	86,0	180,2	32,3	9,0	16,1	20,3
Mokėjimo kortelių vertė (bn.)	3053,5	527,8	1143,2	133,9	88,8	88,2	107,6
Nuostolis dėl sukčiavimų (mln.)	1800,0	360,7	690,7	11,7	13,3	15,6	25,9

Šaltinis: sudaryta autorės, remiantis ECB ir nets duomenimis

Kalbant apie kreditinių kortelių sukčiavimus, kurių būna daugiau nei debetinių kortelių, su jomis susiję sukčiavimai padidėjo 27,2 procento daugiau nei 2015 metais ir 92 procento palyginti su 2012 metais. Bendras sandorių skaičius padidėjo tik 9,6 procento palyginti su 2015 metais (Seidler, 2018).

Olandija ir Šiaurės šalys vienos iš tų, kurios geriausiai kontroliuoja sukčiavimus ir kovoja su jais visoje Europoje (žr. 1 lentelę). Nuostoliai nuo sukčiavimų siekia nuo 11,7 iki 25,9 milijonų eurų. Jungtinėje Karalystėje ir Prancūzijoje yra didesni kortelių sukčiavimo atvejai, daugiausia dėl CNP sukčiavimo perkant internetu, pamestų ir pavogtų kortelių ar tarpvalstybinio sukčiavimo nuostolių, kai užsienyje yra naudojamos vidaus kortelės. Prancūzijoje nuostoliai sudaro virš 360 milijonų, o Jungtinėje Karalystėje beveik dviem kartais daugiau.

Kiekvienoje šalyje pasiskirstymai mokėjimų tarp kanalų yra labai skirtingi (žr. 13 pav.). Bankomatuose įvykstantys sukčiavimai (ATM) sudaro nuo 0 % iki 26 %, mediana siekia 4 %, operacijos, kurios atliekamos nepateikus fiziškai kortelės (CNP) sudaro nuo 33 % - 93 %, mediana - 24 %, o terminaluose (POS) - 7 % - 61 %, mediana - 24 %.



13 pav. Mokėjimų pasiskirstymas tarp skirtingų kanalų

Šaltinis sudaryta autorės, remiantis ECB ir nets duomenimis

Gauti rezultatai tik parodo, kad problema yra opi, ją reikia spręsti. Suradus tinkamus modelius ir juos pritaikius galima sumažinti finansinių institucijų nuostolius, gerinti jų reputaciją ir tada klientai jaustųsi mažiau pažeidžiami. Jeigu CNP operacijos Europos sąjungos lygmeniu sudaro 73 procentus visų sukčiavimo nuostolių, tai maksimaliai išnaudojus šias spragas ir tinkamai panaudojus dirbtinį intelektą, galima būtų sutaupyti 1,314 milijonus eurų.

1.3. Dirbtinio intelekto nauda ir rizikos

Dirbtinio intelekto naudojimas iš tiesų teikia daug naudos įvairiose srityse. Jo dėka yra apdorojami labai dideli kiekiai duomenų, mažinamas žmoniškųjų klaidų faktorius. Taip pat vyrauja efektyvesnis išlaidų panaudojimas, suprantami realūs klientų poreikiai pagal įvairiausius kriterijus, šablonus. Tačiau neišvengiama rizika ir pavojai, kuriuos gali sukelti ši technologija. Tikrai stinga pilno pasitikėjimo, kuris jau buvo minėtas kiek anksčiau, pasitaiko neteisingo vystymo ir reikia daug duomenų, kad dirbtinis intelektas tinkamai apsimokytų. Susiduriama ir su teisinių aktų trūkumu ir klaidų atsakomybės dilema - kas kaltas, kai įvyksta didelė žala. Taip pat gali kilti sunkumų nustatant rezultatus ir klaidas (angl. *Black box*).

Vienas iš didžiausių dirbtinio intelekto privalumų yra mažinimas rankinio darbo. Algoritmai geba pakeisti darbuotoją, gali sutaupyti tam tikras sumas, priklausomai nuo srities, ir stipriai sumažinti išlaidas ir mokesčius, kuriuos moka vartotojai. Daugelis finansų sektoriaus darbo vietų buvo laikomos nepakeičiamos, tačiau dabar viskas kinta, kai yra įtraukiamas

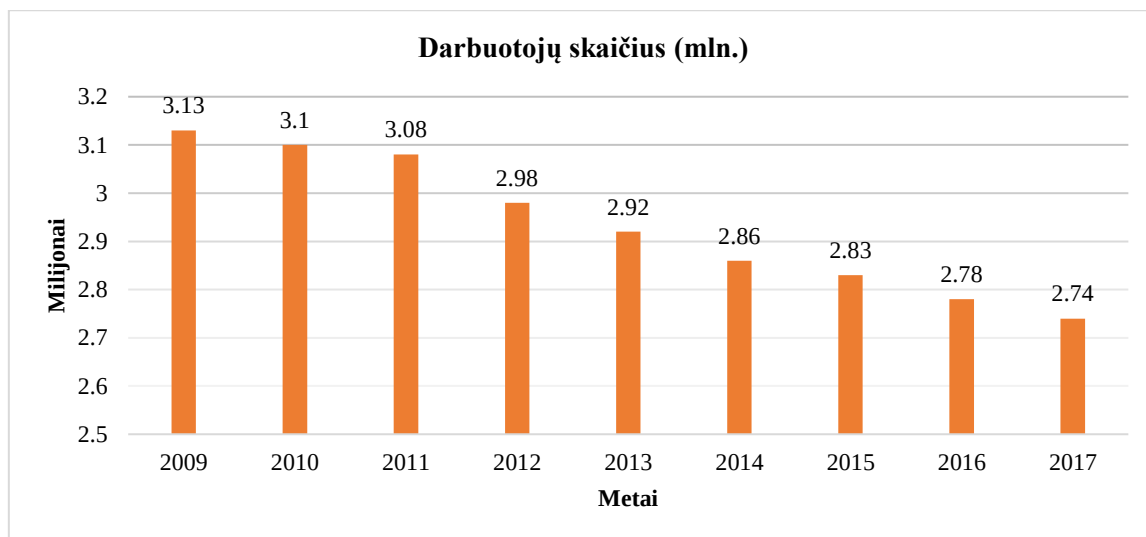
dirbtinis intelektas ir automatizuojami procesai, darbuotojas gali atlikti daugiau aukštesnio pobūdžio užduočių, sutelkti dėmesį į svarbesnius dalykus. Pasak Japonijos įmonės „Mizuho Financial Group“, iki 2027 metų dirbtinis intelektas pakeis net 19 tūkstančių dirbančiųjų.

Dirbtinis intelektas leidžia bankams kurti labiau asmeninius vartojimo produktus, pritaikytus konkrečiam klientui. Tik surinkę pakankamai duomenų apie klientų išlaidas, įpročius, bankai gali rekomenduoti tam tikras paskolų rūšis, draudimus, lizingus ar kitas paslaugas. Ankstyvajame bankininkystės amžiuje bankininkai turėjo asmeninius ryšius su savo klientais, kad padėtų jiems priimti sprendimus. Tačiau skaitmeniniame pasaulyje šis asmeninis ryšys menksta. Dirbtinis intelektas gali būti panaudotas tam, kad sugrąžintų tą ryšį.

Vienas svarbiausių dirbtinio intelekto vaidmenų bankininkystės srityje - gerinti klientų aptarnavimą. Milijonai sandorių atliekami internetu nepriklausomai nuo laiko ir vietos. Automatizuotas dirbtinis intelektas gali lengvai aptarnauti, surinkti duomenis ir apdoroti juos per mokymosi procesą, kad suteiktų reikiamą informaciją ir nukreiptų naudotojus į informacijos šaltinį. Jį paprasta integruoti į bankininkystės programą, rodyti paslaugas, pasiūlymus ir įžvalgas pagal naudotojo elgesį. Taip pat gali padėti vartotojams finansinio planavimo srityje naudojant įvairiai strategijas.

1.4. Dirbtinio intelekto įtaka

Vis dažniau yra aptariama dirbtinio intelekto įtaka darbo rinkai, poveikis visuomenei, ateities perspektyvoms. Vieni džiaugiasi naujovėmis ir jų keliamais patogumais, kiti jaučia baimę dėl galimo darbo praradimo (Marria, 2018). Tiesa, daug darbo vietų gali išnykti, bet taip pat dėl dirbtinio intelekto ir atsirasti. Galbūt žmonės gali atlikti protingesnius ir naudingesnius darbus, mažiau techninius, skatinančius daugiau mąstyti, kurti inovatyvias ir prasmingas idėjas. Galbūt dirbtiniam intelektui vis labiau įsitraukiant į mūsų gyvenimus nebeturėtų egzistuoti standartinė darbo savaitė, gal tą laiką būtų galima išnaudoti kur nors produktyviau. Pavyzdžiui, įvykusi skaitmenizacija iš tiesų sumažino personalo poreikį padaliniuose, vis daugiau jų yra uždaroma. Remiantis Europos banko federacijos duomenimis, 2017 metų pabaigoje banke dirbo apie 2,7 milijonai žmonių, maždaug 40 tūkstančių mažiau negu 2016 metais (žr. 14 pav.).



14 pav. Banko darbuotojų skaičių pokyčiai Europoje

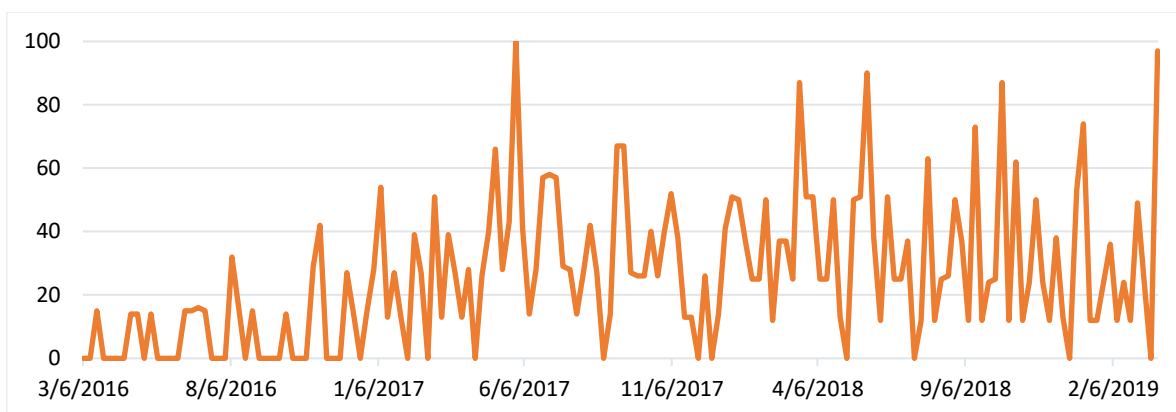
Šaltinis: sudarytas autorės, remiantis European Banking Federation (2017)

Virtualūs asmeniniai asistentai, prie kurių egzistavimo prisideda dirbtinis intelektas, turi didelę svarbą klientų aptarnavimo ir verslo komunikacijų srityse. Ši priemonė padeda žmonėms atlikti kasdienes užduotis, išsprendžia įvairias problemas greitai ir efektyviai. Ji sukuria automatizavimą ir supaprastina procesą, nuolat yra aptarnaujami klientai, atsakoma į paprastus klausimus, o jei reikia, nukreipiama į internetinę banko svetainę. Virtualus asistentas gali gerokai sumažinti klientų aptarnavimo padalinių darbuotojų krūvį. Dabar vyrauja toks laikotarpis, kai finansų įstaigos stengiasi pagerinti virtualių asistentų galimybes, sukurti savo klientams kitokią patirtį ir patogumą (Rao & Verweij, 2018). Tai išties kuria visai naują pokalbių bankininkystės lygį, kai rezultatai yra pristatomi tą pačią akimirką, kas palengvina ir pagreitina kliento sprendimo priėmimą ir klientui nebereikia laukti eilių padaliniuose ar mėginant prisiskambinti konsultantams.

Kuo daugiau virtualiais asistentais žmonės naudosis, tuo daugiau jie patys apsimokys ir gerins savo gebėjimus pagelbėti klientams (Marria, 2018). Šis derinys keičia sąveiką tarp kliento ir paslaugų, vis daugiau klientų renkasi gauti greitus atsakymus, naudodamiesi virtualiais asistentais (Galert, 2018). Pokalbio lange klientas gali įrašyti klausimą ir virtualus asistentas, naudodamas natūralios kalbos apdorojimą ir mašininio mokymosi technologiją, suras reikiamą atsakymą. Tai naudinga bankui, nes tikimasi, kad bus sumažintas klientų aptarnavimo specialistų darbo krūvis, kurį sudaro atsakymai į paprasčiausius klausimus, o virtualus asistentas gali būti pasiekiamas visą parą (Morijs, 2019). Mašinoms iš tiesų yra gana sunku suprasti natūralią žmogaus kalbą. Pavyzdžiui, jei klientas nori užsisakyti naują kortelę, jis tai gali suformuluoti begale skirtingų būdų: „Kur galiu kreiptis dėl naujos kortelės?“,

„Nerandu, kur kreiptis dėl naujos kortelės“, „Man reikia naujos kortelės“. Virtualus asistentas turės visa tai suprasti ir kuo greičiau pateikti teisingą atsakymą.

Daug sumanių asistentų yra naudojama Amerikoje, kurie detalai atsako į didelius srautus klausimų, planuoja susitikimus ar net formuoja pavedimus pagal kliento nurodymus. Virtualūs asistentai yra labai perspektyvūs bankams ir kitoms finansinėms institucijoms, tačiau jei jie neveiks tinkamai, gali kelti ir blogą patirtį klientams (Manning, 2018). Jei yra neatsakoma į vartotojų užklausas, tai gali net padidinti skambučių skaičių klientų aptarnavimo centruose, o ne sumažinti, kaip yra tikimasi. Kita sritis pasak Manning'o yra asmens duomenų saugumo užtikrinimas - turi būti įdiegtos tinkamos autentifikaciją patvirtinančios priemonės. Nuolat stengiantis sukurti kuo geresnę patirtį banko klientams, virtualus asistentas yra svarbus įrankis, kuris gali suteikti realią vertę klientams ir užtikrinti galimybę paprastai paklausti kasdinių klausimų ir gauti akimirksniu greitus atsakymus.



15 pav. Susidomėjimas bankų virtualiais asistentais internete

Šaltinis: sudaryta autorės, remiantis Google Trend duomenimis

Naudojantis „Google Trend“ sistema buvo pastebėta, kad susidomėjimas bankų virtualiais asistentais interneto naršyklėje vis didėjo per keturis metus (žr. 15 pav.). Galima teigti, kad žmonėms yra aktualus paprastumas ir efektyvumas bankuose. Taip pat galima prognozuoti, kad susidomėjimas virtualiais asistentais ateityje tik augs. Tačiau įvertinus rezultatus pagal regionus, didžiausias susidomėjimas vyrauja Amerikoje ir Indijoje, kur virtualių asistentų šiuo metu yra daugiausia. Remiantis „2018 State of Chatbots Report“ duomenimis, komunikacijų kanaluose žmonės yra linkę bendrauti interaktyviai arba naudojantis virtualiais asistentais, kai nori gauti tą pačią akimirką atsakymus į reikiamus klausimus, o tai tik parodo, kad ši priemonė yra viena iš labiausiai naudojamų, kai yra siekiama gauti reikiamą informaciją.

Yra žinomi ir kiti dirbtinio intelekto būdai, kurie mažiau kelia grėsmę darbo vietoms. Dirbtinis intelektas naudojamas atpažinti sukčiavimus ir pinigų plovimui. Būtent tokios įdiegtos sistemos bankuose vis labiau neša sėkmę ir sutrumpina darbuotojų laiką, kai galima efektyviau ir išsamiau nagrinėti konkrečius atvejus.

Dirbtinis intelektas išties supaprastina daugybę sričių, padeda gyventi lengviau, saugiau ir patogiau, bet kelia ir tam tikrą „revoliuciją“ ekonomikoje, kultūroje ir pačioje visuomenėje. Įvairūs algoritmai, kurie apdoroja mūsų duomenis, gali byloti apie mūsų norus, mintis, sprendimus, jausmus ir taip mus paveikti bei kontroliuoti (Kniežienė, 2017). Dirbtinis intelektas kelia baimes, kas bus ateityje, kiek darbų bus prarasta, kaip žmogus sugebės prisitaikyti ir atsispirti naujų technologijų keliamai manipuliacijai. Surinkę pakankamai duomenų apie klientų išlaidų įpročius, bankai gali rekomenduoti konkrečius produktus ar kitas paslaugas. Kaip rodo „Nacionalinio verslo tyrimų instituto“ apklausa, daugiau nei 32 proc. finansų įstaigų naudoja dirbtinio intelekto balso atpažinimo ir nuspėjamosios analizės priemonėmis (Riley, 2018). Pavyzdžiui, „millennials“ labai priklauso nuo mobiliosios bankininkystės, o tai tik reiškia, kad bankinės mobiliosios programos su dirbtiniu intelektu gali jaunimą dar labiau pritraukti (Riley, 2018). Tokios programos gali lengvai atitikti naudotojo lūkesčius su asmeninėmis, kontekstinėmis ir nuspėjamosiomis paslaugomis. Tai yra protingos programos, kurios gali stebėti naudotojo elgesį, suteikti jiems asmeninius patarimus, išvalgas apie taupymą ir išlaidas.

Pasak Nwazor'o (2019) dirbtinis intelektas jau keičia bankų industriją ir tampa labiau prieinamas vartotojams. Finansiškai raštinga ir technologiškai išmintinga jaunoji karta tampa turtingesnė, o bankai stengiasi pasiūlyti daugiau produktų, kad pagerintų kreditingumą, matytų geresnes investicines grąžas. Kitas dirbtinio intelekto privalumas yra sumažėjęs pasitikėjimas rankiniu darbu. Įvairūs algoritmai po truputi keičia žmogiškąjį darbuotoją, kad ateityje būtų galima sutaupyti dideles sumas, sumažinti išlaidas ir mokesčius. Seniau buvo manyta, kad daugelis darbo vietų yra nepakeičiamos dirbtinio intelekto, tačiau dabar vis labiau viską automatizuojant buvo apskaičiuota, kad iki 2025 metų 90 000 iš 300 000 dabartinių darbo vietų šioje srityje išnyks (Nwazor, 2019).

Taip pat dirbtinis intelektas gali prisidėti prie banko pelningumo dviem būdais: perimdamas pasikartojančias darbuotojų užduotis, kas galėtų sumažinti mažiau kvalifikacijos reikalaujantį darbą ir pagerinti banko efektyvumą, o kitas būdas - pajamų generavimas, kuris gali padėti bankams kurti naujus produktus ar pritaikyti esamus, kurie tinkamiau atitiktų kliento pageidavimus. Dirbtinis intelektas daro teigiamą įtaką Europos bankų grąžai (ROA).

Padidindami darbo našumą, dirbtiniu intelektu paremtos technologijos galėtų struktūriškai sumažinti išlaidas bankų sektoriuje.

1.5. Mokėjimo kortelių sukčiavimo aptikimo technikos

Egzistuoja du sukčiavimo aptikimo būdai - sukčiavimo analizė (angl. *Fraud analysis*) ir naudotojo elgesio analizė (angl. *User behaviour analysis*) (Zojaji, Atani & Monadjemi, 2016). Sukčiavimo analizė yra atliekama, kai istoriniai duomenys yra analizuojami, stebima, ar įvykę sandoriai yra įprasti ir niekuo neišsiskiriantys, ar vykdoma nesąžininga veikla. Tokie duomenų rinkiniai yra sudaromi pagal klasifikavimo modelį ir yra naudojamas dviejų klasių klasifikavimas - taisyklių indukcija, sprendimų medžiai ir neuroniniai tinklai. Būtent šios priemonės aptinka daugiausiai sukčiavimo gudrybių. Kitas metodas yra susijęs su nekontroliuojama metodika, kuri priklauso nuo elgesio. Mokėjimo nustatymas gali būti nustatomas apgaulingai, jei prieštarauja vartotojo elgesiui ir dėl to reiktų turėti teisėtą ir tikslų vartotojo elgesio modelį. Sukčiavimo analizė išsiskiria tuo, kad sukčiavimo analizės sistemos mokymasis ženkiai priklauso nuo sukčiavimo įrašų, bet ji negali aptikti naujų įrašų, o analizuojant elgesio analizę yra nukrypstama į naujų sukčiavimų atpažinimą. Tad nors elgesio analizė yra stipri, bet ji dažnai kenčia nuo melagingų signalų.

Naujų technologijų atsiradimas suteikia papildomų galimybių nusikaltėliams sukčiauti. Kredito kortelių naudojimas yra plačiai paplitęs, o šios priemonės sukčiavimas pastaruoju laikotarpiu nuolat kyla. Finansiniai nuostoliai sukčiaujant paveikia ne tik bankus, bet ir individualius klientus. Jei bankas praranda pinigus, klientai galiausiai moka didesnę palūkanų normą ar didesnius mokesčius. Sukčiavimas taip pat gali turėti įtakos banko reputacijai, gali sukelti nefinansinius nuostolius, kuriuos kiek sunkiau kiekybiškai įvertinti per trumpą laiką, o ilgainiui gali tapti labiau pastebimi. Klientas nebegalės pasitikėti savo banku ir pasirinks kitą patikimesnį konkurentą.

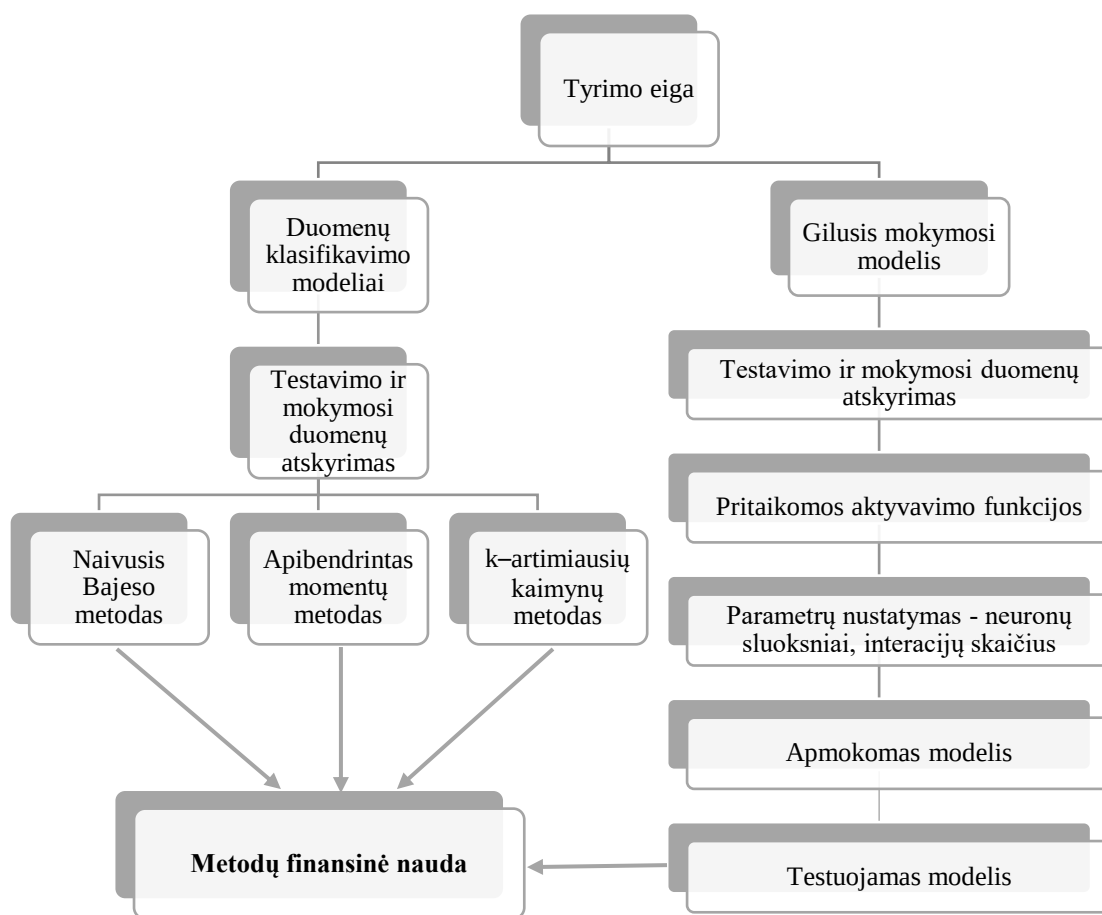
Žinoma, egzistuoja prevencija nuo sukčiavimo, kurios pagalba bandoma blokuoti nesąžiningus sandorius ir aptikti sukčiavimus. Technologijos, kurios buvo naudojamos siekiant užkirsti kelią sukčiavimui, yra adreso tikrinimo sistemos (angl. *Address Verification System*), kortelės tikrinimo metodas (angl. *Card Verification Method*) ir asmens identifikavimo numeris (angl. *Personal Identification Number*) (Bhatla et al., 2003). Sukčiavimas yra aptinkamas, kai yra atsižvelgiama į kredito kortelių vykdytų operacijų rinkinį, procesų identifikavimą, ar įvykusi operacija priklauso suklastotai, ar tikrai sandorių klasei (Maes, Tuyls, Vanschoenwinkel & Manderick, 2002).

Pirmiausias yra svarbu patikrinti esmines sąlygas: ar tinkamas balansas ir įvertinti pagal parinktą modelį, kuris įvertina kiekvieną sandorį, ar jis yra su didele, ar su maža sukčiavimo rizikos tikimybe. Jei rizika yra didelė, asmenys yra informuojami perspėjimais. Tyrėjai tikrina šiuos įspėjimus ir pateikia kiekvieno perspėjimo atsakymą, ar tikras sukčiavimas, ar ne.

Galime efektyviai naudotis mašininio mokymosi metodikomis nustatant sukčiavimus ir remtis įvairiomis duomenų bazėmis. Dažniausiai modelis yra parametrų funkcija, kuri leidžia prognozuoti sandorio sukčiavimo tikimybę. Sukčiavimo aptikime mokymosi metodų naudojimas yra patogus įrankis, nes leidžia aptikti modelius didelės apimties duomenų bazėse, o kiekvieną sandorį apibrėžia daugybė kintamųjų. Taip pat, nesąžiningi sandoriai dažnai koreliuoja tiek laiko, tiek erdvės atžvilgiu. Vienas iš tokių pavyzdžių, kai sukčiavimus bando atlikti toje pačioje parduotuvėje per trumpą laiką. Kai aptinkamas sukčiavimo metodas, nusikaltėliai pritaiko savo strategijas ar bando visai kitas. Tuo pačiu metu kasdien nauji nusikaltėliai taip pat bando sukčiauti, bet jau naudodami naujas strategijas. Keitimasis idėjomis apie aptikimo įrankius nėra plačiai paplitęs, finansinės institucijos irgi nėra linkusios jų atskleisti. Dėl tos pačios priežasties taip pat ir duomenų bazės nėra dažnai viešai prieinamos moksliniams tyrimams atlikti.

2. SUKČIAVIMO ATPAŽINIMO TYRIMO METODAI

Saugumas yra bene svarbiausia sritis bankams. Tai svarbi sąlyga klientų pasitikėjimui išsaugoti bei sėkmingos veiklos vykdymui. Nuo seno klientai savo turtą ir pinigus patikėdavo bankams, nes manė, kad jų seifai bus apsaugoti nuo piktadarių. Tačiau skaitmeninių technologijų amžiuje, atsiranda kita sąvoka kaip kibernetinis saugumas, dėl ko yra svarbu užtikrinti pinigų saugojimą sąskaitose, atliekamų mokėjimų saugumą ir užtikrinti konfidencialumą. Pasitelkus dirbtinio intelekto įrankius, stengiamasi su tokiais pavojais kovoti kuo efektyviau ir pasiekti kuo geresnį sukčiavimų atpažinimą. Tyrimui atlikti buvo pasirinkti keli duomenų klasifikavimo modeliai ir jų rezultatai sulyginami su giliuoju mokymusi (žr. 16 pav.).



16 pav. Tyrimo eigos grafikas

Šaltinis: sudaryta autorės

Tiek duomenų klasifikavimo modeliams, tiek giliojo mokymosi modeliui pirmiausiai svarbu apdoroti pasirinktą duomenų bazę ir ją suformatuoti. Sukuriama mokymosi ir testavimo duomenys iš esamos duomenų bazės pagal pasirinktas strategijas. Duomenys turi būti vienos dimensijos ir normalizuoti, o tai padeda atlikti tam tikros funkcijos ir algoritmai. Dažnai giliojo mokymosi atveju yra kuriama sigmoidinė, relu, linijinė ir tangento funkcija, nustatomi parametrai - neuronų ir iteracijų skaičius, sluoksniai. Modelis apmokomas mokymosi duomenimis, o tada yra testuojamas. Duomenų klasifikavimo ir giliojo mokymosi modelių tikslas tas pats - pasiekti kuo didesnę tikslumą.

2.1. Anomalijos ir jų aptikimas

Bandant aptikti mokėjimų sukčiavimą, viskas prasideda nuo anomalijų paieškos. Anomalijų aptikimas yra toks metodas, kuris yra naudojamas siekiant nustatyti neįprastus modelius, kurie yra neatitinkantys tikėtino kliento elgesio (Chandola, Banerjee & Kumar, 2009). Jų būna įvairiausių, tiek nuo įsibrovimo, tiek iki kreditinių kortelių sukčiavimo aptikimo.

Pačios anomalijos tai yra reti įvykiai, kurie kelia įtarimus, nes tam tikri rodikliai ženkiai išsiskiria nuo kitų duomenų. Anomalijos yra skirstomos į (Hayes & Capretz, 2014; Perera, 2015):

- **Taško anomalijas** (angl. *Point Anomalies*): kai tam tikras rodmuo yra atitolęs nuo kitų rodmenų. Sukčiavimų atveju galima atsižvelgti į mokėtinas sumas, kurios gali būti stipriai didesnės lyginant su kitomis.

- **Kontekstines anomalijas** (angl. *Contextual Anomalies*): kurio priklauso nuo konteksto ir yra įprastas reiškinys laiko eilučių duomenyse. Vienas iš pavyzdžių galėtų būti tam tikru laiko momentu išleidžiama vienoda suma ar vienoda operacija tam tikrame regione.

- **Kolektyvines anomalijas** (angl. *Collective Anomalies*): kai duomenų rinkinys padeda nustatyti anomalijas ir vaizduoja netikėtus derinius. Pavyzdžiui, kai yra perkama daugybė brangių daiktų.

Anomalijos yra nustatomos pagal paprastuosius statistinius mašininio mokymosi metodus. Vienas iš paprastųjų statistinių metodų būdų yra aprašyti taisykles, kai tokios anomalijos yra aptinkamos ir jei iškelta hipotezė yra teisinga pagal tam tikras taisykles, galima interpretuoti, kad tai yra anomalija. Tačiau paprastieji statistiniai modeliai ne visada suveikia. Duomenys gali vaizduoti anomalinį reiškinį, bet riba tarp elgsenos nėra tiksliai nusakyta arba vyraujantis sezoniškumas yra neįvertintas, todėl rezultatai yra iškreipiami. Bet mašininio

mokymosi algoritmai anomalijų nagrinėjime vis plačiau yra naudojami ir tolimesniame tyrime bus palyginami keli metodai ir jų tikslumai mokėjimų sukčiavimuose.

2.2. Duomenų klasifikavimas

Vienas iš pagrindinių mašininio mokymosi uždavinių yra duomenų klasifikavimas. Šiame uždavinyje įvesties duomenys duodami mašininio mokymosi modeliui, kurio tikslas pagal pateiktus duomenis prognozuoti kintamojo reikšmę. Duomenims turi būti apibrėžtos tam tikros kategorijos, o tai priklauso nuo pačio uždavinio paskirties. Pavyzdžiui, duomenys gali būti klasifikuojami į dvi kategorijas ar net daugiau.

Modelio sudarymui yra svarbūs etapai - modelio apmokymas ir testavimas. Modelio apmokymas išsiskiria tuo, kad jo parametrai yra vis koreguojami naudojant apmokymo duomenis. Modelio testavimui yra taikomi nauji duomenys, likusi turimų duomenų dalis, kurie nebuvo panaudoti siekiant sudaryti modelį. Pagrindinis tikslas - sudaryti tokį klasifikavimo modelį (klasifikatorių), kuris tinkamai atliktų duomenų klasifikavimą naudojant tiek apmokymo, tiek testavimo duomenis.

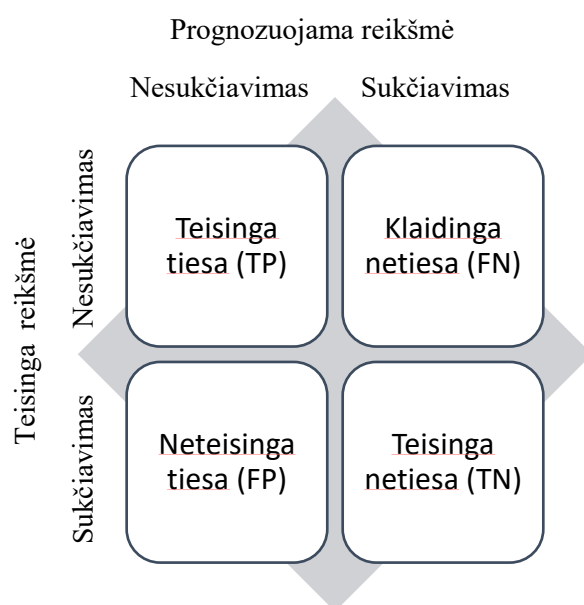
Klasifikavimas geba surasti dėsningumą istoriniuose duomenyse ir juos suskirstyti į klases. Tikslas analizuoti turimus duomenis ir sukurti modelį, kurio pagalba būtų galima prognozuoti ateities galimus poelgius, mokytis priskirti kitokius duomenis reikiamai klasei. Klasifikavimas yra skirstomas į daug technikų, bet nagrinėjamos bus šios: naivusis Bajeso metodas, apibendrintas momentų metodas, k-artimiausias kaimynų metodas ir gilusis mokymasis.

Egzistuoja daugybė tiek minėtų, tiek nemintų modelių, kurie geba klasifikuoti duomenis, bet tolimesnio tyrimo metu bus pasirenkami tik minėti ir palyginami jų tarpusavio rezultatai. Tikslas visus šiuos modelius pritaikyti sukčiavimo atpažinimo uždavinyje ir gauti kuo mažesnę klaidingo klasifikavimo tikimybę.

2.3. Klaidų matrica

Siekiant įvertinti pasirinkto modelio tikslumą yra lyginamos tarpusavyje prognozuojamos ir tikrosios reikšmės pagal klaidų matrica (angl. *Confusion matrix*). Tai mašininio mokymosi klasifikavimo matavimo būdas, kai išėjimas (angl. *Output*) gali būti dviejų ar daugiau klasių (žr. 17 pav.). Ši matrica vizualiai pavaizduoja gautus duomenis pagal 4 skirtingas prognozuojamas ir faktines (teisingas) verčių kombinacijas (Narkhede & Shinde, 2018):

- **neteisinga tiesa** (FP) (angl. *False Positive*): naudojama apibūdinti teigiamas prognozes, kurios iš tikrųjų yra neigiamos;
- **teisinga tiesa** (TP) (angl. *True Positive*): naudojama apibūdinti teigiamas prognozes, kurios iš tikrųjų yra teigiamos;
- **klaidinga netiesa** (FN) (angl. *False Negative*): naudojama norint apibūdinti neigiamas prognozes, kurios iš tikrųjų yra teigiamos;
- **teisinga netiesa** (TN) (angl. *True Negative*): naudojama apibūdinti neigiamas prognozes, kurios iš tikrųjų yra neigiamos.



17 pav. Klaidų matrica

Šaltinis: sudaryta autorės, remiantis Narkhede (2018)

Aukščiau pateikta matrica parodo, kaip galėtų būti interpretuojami skaičiai, jei matricos duomenys būtų susiję su mokėjimų sukčiavimu pagal prognozuojamas ir faktines reikšmes (žr. 17 pav.). Siekiant nustatyti klasifikavimo tikslumą yra naudojami ir kiti matai (Tichonov & Yee, 2018):

- **bendras klasifikavimo tikslumas** (angl. *Accuracy*) rodo teisingai suklasifikuotų ir visų duomenų santykį, ir gali būti naudojamas tada, kai esančios klasės subalansuotos;

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

- **tikslumas** (angl. *Precision*) - gali parodyti, kiek prognozuotų reikšmių yra teisingi rezultatai;

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

- **jautrumas** (angl. *Recall*) - gali parodyti, kokia dalis buvo teisingai prognozuota teigiamų reikšmių;

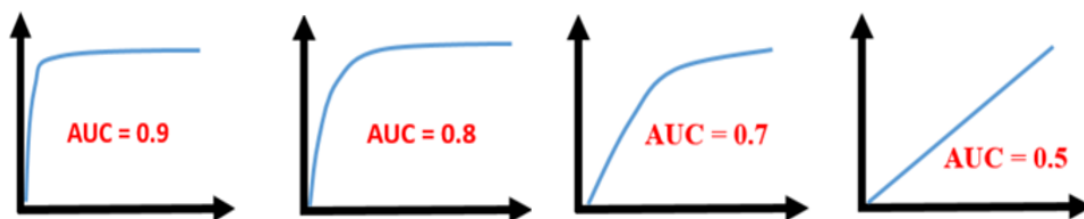
$$Recall = \frac{TP}{TP + FN} \quad (5)$$

- **F1 rodiklis** (angl. *F1 score*) yra reikalingas, kai norime rasti pusiausvyrą tarp tikslumo ir jautrumo, ir kai klasių paskirstymas yra netolygus - didelis skaičius faktinių neigiamų reikšmių (Flach et al., 2015).

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (6)$$

2.4. AUC - ROC kreivė

AUC (angl. *Area Under The Curve*) - ROC (angl. *Receiver Operating Characteristic*) kreivė yra naudojama klasifikavimo uždaviniams spręsti ir parodyti jų našumą (Narkhede, 2018; Draelos, 2019). AUC yra plotas po ROC kreive, o ROC - tikimybinė kreivė. Siekiant įvertinti sukurto modelio rezultatus, blogiausias AUC egzistuoja, kai siekia 0, o geriausias AUC, kai yra arti 1 (žr. 18 pav.).



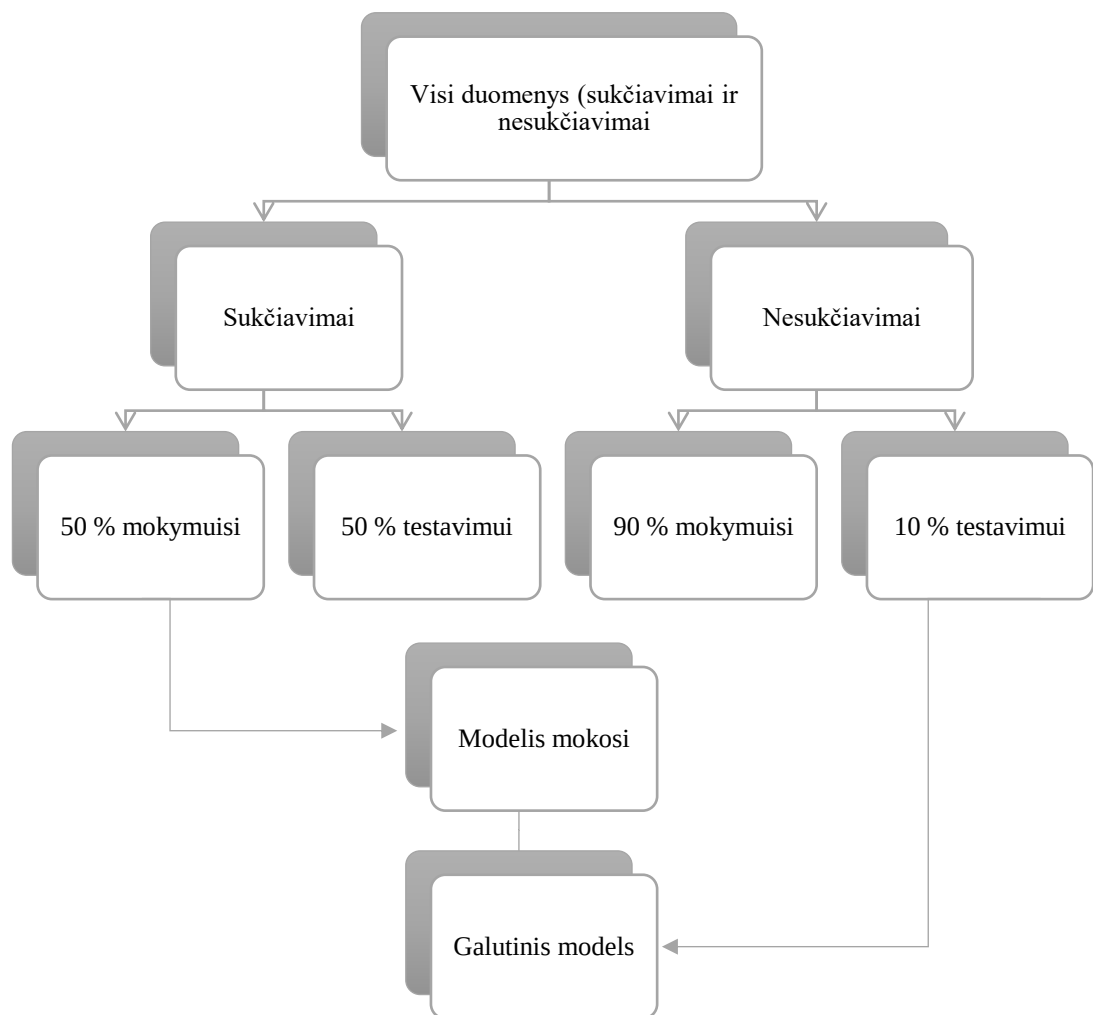
18 pav. AUC-ROC kreivės

Šaltinis: D'Souza (2018)

ROC kreivė (angl. *Receiver Operating Characteristic*) yra naudojama tada, kai siekiama įvertinti kaip modelis geba atskirti esamas klases ir parodyti kaip teisingai klasifikuojamų teigiamų pavyzdžių skaičius priklauso nuo neteisingai klasifikuojamų neigiamų pavyzdžių skaičiaus (Fawcett, 2005; Davis, 2006). Esantis plotas po kreivėmis yra skirtas įvertinti metodą ir jo veiksmingumą. Kuo užimamas plotas yra didesnis, tuo metodas tinkamesnis. AUC taip pat labiau yra vertinamas nei bendras klasifikavimo rodiklis (Ling, Huang & Zhang, 2003).

2.5. Apibendrintas momentų metodas ir jo taikymas sukčiavimui

Apibendrintas momentų metodas (GMM) yra įverčių, analogiškų maksimaliai tikimybei, konstravimo metodas (Scherrer, 2007). Šis metodas naudoja prielaidas apie konkrečius atsitiktinių kintamųjų momentus, o ne prielaidas apie visą paskirstymą. Prielaidos vadinamos momentinėmis sąlygomis.



19 pav. GMM modelio duomenų skaidymas

Šaltinis: sudaryta autorės

Giliojo mokymosi ir mašininio mokymosi kūrėjams iš visų tikimybinių modelių dažnai tenka susidurti su Gauso paskirstymo modeliu. Jis dažnai prilyginamas varpo formos kreivei ir dar yra vadinamas normaliu pasiskirstymu. Būtent pagal šias kreives tyrimo dalyje bus pasirenkami tam tikri komponentai ir įvertinami tikslumą nusakantys rodikliai: bendras klasifikavimo tikslumas, jautrumas, tikslumas, F1 rodiklis ir AUC kreivė. Tačiau pirmiausiai duomenys bus išskaidomi pagal schemą, pateiktą kiek žemiau (žr. 19 pav.). Sukčiavimo duomenys paskirstomi po lygiai, tiek mokymuisi, tiek testavimui, o nesukčiavimo - 90 proc. skirta mokymuisi, o 10 proc. testavimui. Modelis apmokomas su mokymosi duomenimis, o po to yra testuojamas su testavimo duomenimis.

Tiek šis, tiek kiti modeliai yra apskaičiuojami, remiantis „Python“ programavimo kalba. Pagal „mixture.GaussianMixture“ klasę yra sprendžiamas apibendrintas momentų metodo algoritmas ir taip atpažįstami sukčiavimą nusakantys tikslumo rodikliai, kurie bus detaliau aptarti tiriamojoje dalyje.

2.6. K-artimiausių kaimynų metodo pritaikymas sukčiavimui

Kaip jau buvo minėta teorinių aspektų dalyje, k-artimiausio kaimyno metodas pradedamas nagrinėti nuo atsitiktinio elemento, nuolat renkantis iš nepasirinktų patį artimiausią kaimyną. Šis algoritmas gana paprastas, bet jo rezultatai gali būti nuviliantys. Algoritmas geba greitai pateikti rezultatus, bet dėl neefektyvumo yra rečiau naudojamas (Kumar, 2019).

Šiam metodui atlikti pagal „KNeighborsClassifier“ klasę yra sprendžiamas artimiausio kaimyno algoritmas ir taip atpažįstami sukčiavimą nusakantys tikslumo rodikliai ir sudaroma klaidų matrica. Bandymų metu yra keičiamas artimiausių kaimynų skaičius. Visi duomenys yra išskaidomi į 80 proc. duomenų, skirtų mokymuisi, o testavimui - 20 proc.

2.7. Naivusis Bajeso metodas ir jo naudojimas sukčiavimui atpažinti

Naivusis Bajeso metodas yra tikimybinis klasifikavimo metodas, kuris yra grindžiamas Bayes taisykle ir yra pavadintas matematiko Thomas Bayes garbei. Šis klasifikatorius bando numatyti klasę, vadinamą rezultatų klase, remiantis tikimybėmis (Kiran et al., 2018). Toks mokymasis yra labai efektyvus, greitas ir tikslus. Be to, jis įvertina parametrus, kai yra naudojama labai mažai mokymosi duomenų, kurie naudojami klasifikavimui.

Tyrimo rezultatams gauti pagal „GaussianNB“ klasę yra sprendžiamas naivusis Bajeso klasifikatoriaus algoritmas, apskaičiuojami tikslumo rodikliai ir klaidų matrica. Šio metodo

metu visi duomenys yra išskaidomi į 80 proc. mokymuisi, o testavimui - 20 proc. Kai yra paduodama „predict()“ funkcija su testiniais duomenimis, galima gauti tokį duomenų masyvą, kuriame bus matyti visi sukčiavimo ir nesukčiavimo atvejai, o su „predict_proba()“ funkcija prognozuojama tikimybė, kaip tiksliai bus atpažinti sukčiavimai ir nesukčiavimai. Visi šie rezultatai bus aptariami ir įgyvendinti tiriamojoje dalyje.

2.8. Aktyvavimo funkcijos giliajam mokymuisi atlikti

Siekiant naudoti metodą, kuris atpažintų sukčiavimus, kai yra įtraukiamas dirbtinis intelektas, verta paminėti dirbtinio neurono sandarą ir aktyvavimo funkcijas. Dirbtinis neuronas yra neuroninio tinklo elementas, kurį sudaro šie elementai: svoriai, slenksčiai ir tam tikra aktyvavimo funkcija. Svorio koeficientai w vaizduoja atskirų įvesčių stiprumą, aprašytą vektoriumi x . Tada įvesties signalas yra dauginamas iš svorio koeficiento ir gaunama neuroninė jungtis xw . Jeigu svorio koeficientas teigiamas, xw sužadina signalą išvestyje y , o jei neigiamas - xw slopina išvesties signalą (žr.(7) formulę.).

$$y = f\left(\sum_{i=1}^n XW - \theta\right) \quad (7)$$

Aktyvavimo funkcija (angl. *Activation function*) yra matematinė funkcija su išvesties signalu (Mahmood, 2018). Jos sudėtingumo pasirinkimas priklauso nuo neuroninio tinklo sprendžiamo uždavinio. Perceptronas (angl. *Perceptron*), tai smegenų modelis, kuris geba tirti intelektą tiek fizikiniais, tiek matematiniais įrankiais, bet jis taip pat yra įvardijamas kaip dirbtinis neuronas (Lee, 2018). Dirbtinius neuronus galima sujungti tiek į vieną sluoksnį (angl. *Single layer perceptron*), tiek į kelis (angl. *Multilayer perceptron*). Visi neuronų tinklo perceptronai turi dvi funkcijas: įvesties ir aktyvinimo. Įvesties funkcija surenka visas įvestis ir atlieka sumos funkciją, po kurios perkelia rezultatą į aktyvinimo funkciją. Aktyvinimo funkcija atlieka keletą veiksmų ir tik tada yra vykdomas kitas etapas. Aktyvinimo funkcijos yra sudarytos iš trijų tipų (Gupta, 2017):

- **Binarinės pakopos funkcija** (angl. *Binary step function*) yra slenksčiu pagrįsta aktyvinimo funkcija. Jei įvesties vertė yra virš arba žemesnė už tam tikrą slenkstį, neuronas aktyvuojamas ir siunčia tiksliai tą patį signalą į kitą sluoksnį. Tačiau ši funkcija neleidžia gauti daug verčių turinčių rezultatų. Ji negali palaikyti įvesties klasifikavimo į vieną iš kelių kategorijų.

- **Linijinė aktyvavimo funkcija** (angl. *Linear activation function*) neuroniniuose tinkluose paima įėjimus, padaugintus iš kiekvieno neurono svorio ir sukuria išėjimo signalą, kuris yra proporcingas įėjimui. Linijinė funkcija yra geresnė už binarinės pakopos funkciją, nes ji turi daugiau išėjimų. Tačiau linijinio aktyvinimo funkcija turi dvi pagrindines problemas. Pirmiausiai negalima naudoti „*backpropagation*“ (gradiento nusileidimo) funkcijos modeliui mokytis, nes funkcijos išvestis yra pastovi ir neturi jokio ryšio su įėjimu. Taigi nėra galimybės grįžti ir suprasti, kurie svoriai įvesties neuronuose gali suteikti geresnį prognozavimą. Kita problema, kad linijinio aktyvinimo funkcija neuroninį tinklą paverčia vienu sluoksniu. Neuroninis tinklas, turintis linijinę aktyvinimo funkciją, yra linijinis regresijos modelis ir turi ribotą galią ir gebėjimą valdyti sudėtingus įvesties duomenų parametrus.
- Šiuolaikiniai neuronų tinklo modeliai naudoja **nelinijines aktyvinimo funkcijas** (angl. *Non-linear activation function*). Jie leidžia modeliui sukurti sudėtingus žemėlapių įvesties ir išvesties duomenis, kurie yra būtini sudėtingiems duomenims, pavyzdžiui, tokiems kaip: vaizdams, vaizdo įrašams, garsui ir duomenų rinkiniams. Beveik bet koks įsivaizduojamas procesas gali būti pateikiamas kaip funkcinis skaičiavimas neuroniniame tinkle, su sąlyga, kad aktyvinimo funkcija yra nelinijinė. Nelinijinės funkcijos sprendžia linijinio aktyvinimo funkcijos problemas: leidžia sukurti gilųjį neuroninį tinklą. Norint išmokyti sudėtingus duomenų rinkinius, turinčius didelį tikslumo lygį, reikalingi keli paslėpti neuronų sluoksniai.

Vieni dažniausių aktyvavimo funkcijų pavyzdžių:

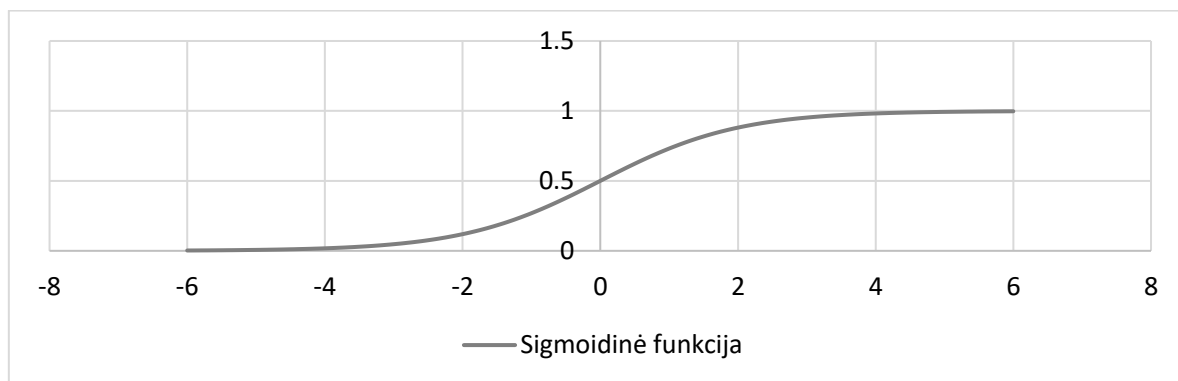
Tiesinė funkcija išsiskiria tuo, kad jos reikšmė proporcingai didėja pagal v reikšmę (žr. (8) formulę.). Ši funkcija išsprendžia dvejetainės pakopos funkcijos problemą, kai ji pateikia tik 0 ir 1 reikšmę (Sharma, 2017). Siekiant modelį apmokyti, ši funkcija nėra tinkama ir dažniau yra naudojamos netiesinės funkcijos.

$$f(v) = v \tag{8}$$

Sigmoidinės funkcijos naudojimo priežastis dažnai yra siejama su tuo, kad jos intervalas yra tarp 0 ir 1, todėl dažnai ji yra naudojama su modeliais, kurie yra susiję su klasifikavimu ir modelio apmokymu (Karlik & Olgac, 2011; Sharma, 2017). Kaip galima

matyti iš (9) formulės, funkcija yra diferencijuota, o tai rodo, kad galima surasti sigmoidinės kreivės nuolydį bet kuriuose dviejuose taškuose (žr. 20 pav.).

$$f(v) = \frac{1}{1 + e^{-v}} = \frac{e^v}{e^v + 1} \quad (9)$$

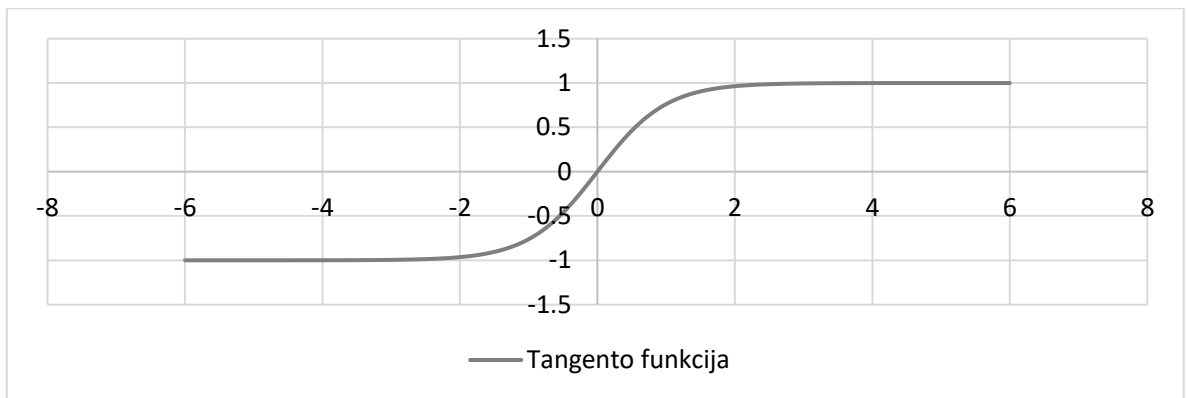


20 pav. Sigmoidinės funkcijos grafikas

Šaltinis: sudaryta autorės

Hiperbolinė tangento funkcija taip pat gali būti naudojama kaip aktyvavimo funkcija (Lu, 2017). Privalumas yra tas, kad neigiamos įvestys yra pažymėtos neigiamai, o nulinės įvestys - pažymėtos šalia nulio (Khan, 2014). Taip pat ši funkcija yra diferencijuota ir monotonišė, o daugiausia naudojama yra dviejų klasių klasifikacijai (Sharma, 2017). Hiperbolinės tangento funkcijos rezultatai yra nuo -1 iki 1. Kaip vaizduoja grafikas jie yra apriboti ir diferencijuojami (Karlik & Olgac, 2011). Funkcija ir jos formulė yra pateikta žemiau (žr. 21 pav.).

$$f(v) = \tanh(v) = \frac{e^v - e^{-v}}{e^v + e^{-v}} = \frac{e^{2v} - 1}{e^{2v} + 1} \quad (10)$$



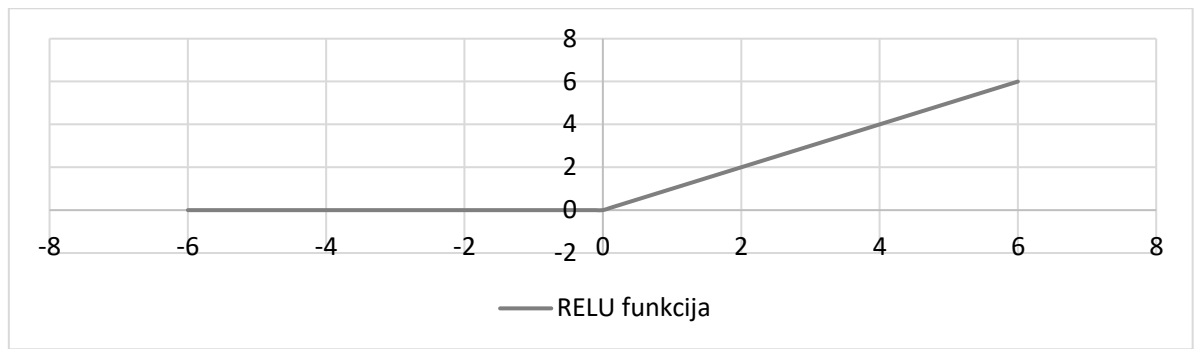
21 pav. Tangento funkcijos grafikas

Šaltinis: sudaryta autorės

Detektorinė linijinio vieneto (toliau - **ReLU**) funkciją (angl. *The Rectified Linear Unit*) plačiai naudojama kaip aktyvinimo funkcija, kuri yra skirta giliojo mokymosi uždaviniams atlikti (Sharma & Lu, 2017). Tai greitesnis mokymasis, kuris pasirodė esantis sėkmingiausias ir plačiausiai naudojamas. Jis siūlo spartesnį neuroninio tinklo mokymąsi, geresnius rezultatus ir apibendrina gilųjų mokymąsi lyginant su sigmoidine ir tangento aktyvavimo funkcijomis (Nwankpa, Ijomah, Gachagan & Marshall, 2017). Jis atstovauja linijinei funkcijai ir išsaugo linijinių modelių savybes, dėl kurių jas galima lengviau optimizuoti (žr. 22 pav.). Pagrindinis ReLU funkcijos panaudojimo pranašumas yra tas, kad ji garantuoja greitesnį skaičiavimą, nes jis yra atliekamas neapskaičiuojant eksponentų ir padalijimų, o bendras skaičiavimo greitis tampa sustiprintas.

$$f(v) = \max(0, v) \quad (11)$$

Dėl giliojo neuroninio tinklo atsiranda tokia problema, kad nepakanka vien sigmoidinės ar hiperbolinės tangento funkcijos. Dėl nykstančių gradientų problemos, kuri atsiranda, kai išvestinės aktyvavimo funkcijų yra $(-1;1)$, tada neuroninis tinklas sumažina savo mokymąsi arba jį baigia (Browlee, 2019; Moon, Park, Rho & Hwang, 2019). Pasak Marchisio et al. (2018) būdas to išvengti - ReLU funkcija, kurios išvestinė lygi 1, kai $x > 0$ bei lygi 0, kai $x < 0$, o dėl to neuroninis tinklas veikia greičiau (žr. (11) formulę.).



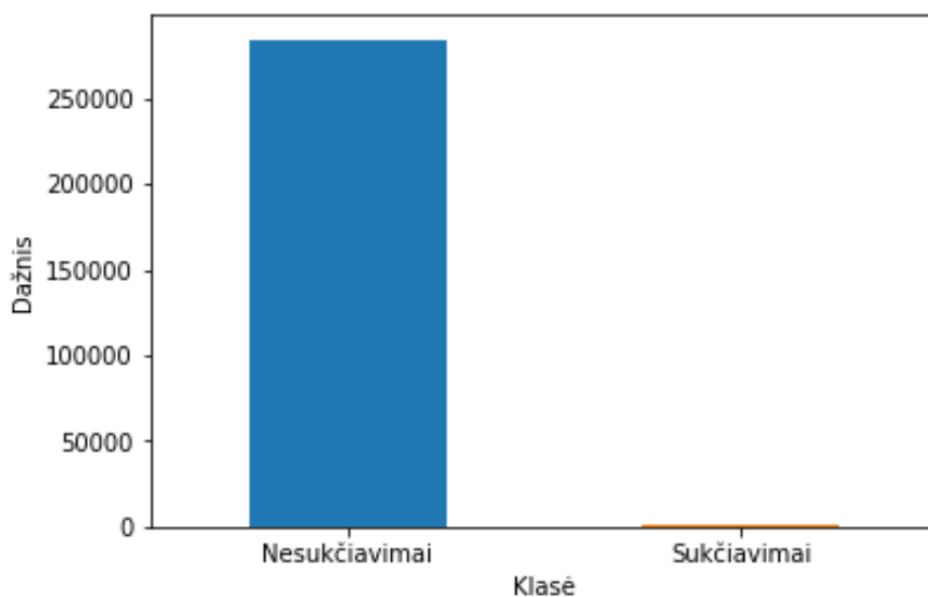
22 pav. RELU funkcijos grafikas

Šaltinis: sudaryta autorės

Tad tiriamojoje dalyje bus pritaikomos kai kurios aktyvavimo funkcijos, sudaromos neuroninio tinklo architektūros ir atlikus įvairius testavimus bus mėginama pasiekti kuo didesnį sukčiavimų atpažinimą keičiant pagrindinius parametrus. Bus pritaikoma sigmoidinė aktyvavimo funkcija, nes ji padeda greitai suklasifikuoti duomenis, nes jos rėžiai yra nuo 0 iki 1, kas yra pritaikoma sukčiavimų tematikoje. Aktyvavimo funkcija ReLU yra tinkama naudoti dėl greitesnio mokymosi proceso ir to, kad ji nesuaktyvina visų neuronų tuo pačiu metu (Walia, 2017). Jei jos įvestis yra neigiama, ji bus paversta nuliu, o neuronas nebus suaktyvintas. Tai tik rodo, kad vienu metu yra suaktyvinami tik keli neuronai, todėl neuroninis tinklas yra nedidelis, bet efektyvesnis. Dirbtinių neuroninių tinklų mokymuisi svarbus didelis duomenų kiekis, tinkamai nustatyta struktūra ir pagrindiniai hiperparametrai. Šio tinklo parametrai yra vis keičiami tam, kad būtų gaunamas kuo geresnis atpažinimo rezultatas. Taip pat prie tikslesnio rezultato gavimo prisideda ir kitas algoritmas „backpropagation“, kuris padeda rasti mažiausią paklaidą, kai yra naudojamas mokymosi greitis, kuris dažniausiai siekiama, kad būtų kuo mažesnis (Mercioni, Tiron. & Holban, 2019). Galiausiai visi tiek mašininio mokymosi, tiek giliojo mokymosi modeliai palyginami tarpusavyje pagal tikslumą nusakančius rodiklius ir apskaičiuojama finansinė nauda, kurią gali gauti finansinės institucijos, jei sukčiavimai būtų tinkamai atpažįstami.

3. KREDITINIŲ KORTELIŲ MOKĖJIMŲ SUKČIAVIMO TYRIMAS

Bankai siekia atpažinti nesąžiningus mokėjimo kortelių sandorius, kad klientai nebūtų apmokestinami pirkiniais, kurių, pavyzdžiui, jie nepirko, kad nebūtų pasinaudota jų duomenimis ir vykdoma nesąžininga veikla. Šiai problemai išspręsti yra naudojama daug dirbtinio intelekto algoritmų, kurie gali naudoti mokėjimo kortelių duomenų bazes ir atpažinti sukčiavimą. Iš Kaggle duomenų bazių sistemos buvo pasirinkta duomenų bazė, susijusi su kreditinių kortelių sukčiavimais. Šie duomenys yra apie kreditinių kortelių savininkų operacijas Europoje 2013 metais rugsėjo mėnesį. Tačiau šis duomenų rinkinys rodo operacijas, kurios yra įvykusios per dvi dienas. Iš viso yra įvykdyti 492 apgaulingi veiksmai iš 284 807 operacijų. Eksperimentinio tyrimo metu buvo analizuojami mokėjimų sukčiavimai naudojant „Python“ programavimo kalbą. Importuojamos reikiamos bibliotekos, pritaikomi algoritmai pagal pasirinktus modelius ir vizualizuojami gauti duomenys.

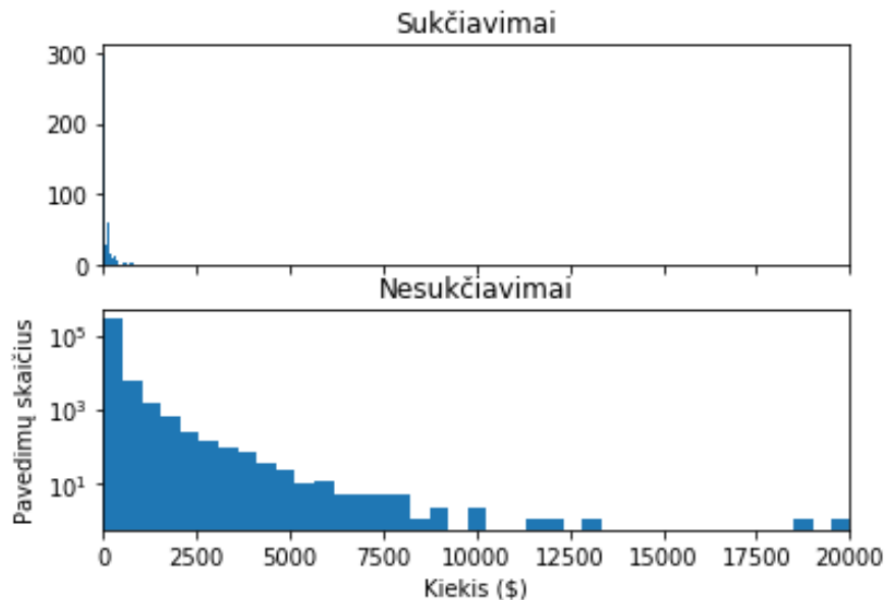


23 pav. Mokėjimo klasių paskirstymas

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze

Duomenų bazėje egzistuoja dvi mokėjimo klasės - pavedimai be sukčiavimų ir pavedimai su sukčiavimais (žr. 23 pav.). Duomenų rinkinys yra nesubalansuotas, nes teigiama klasė (sukčiavimai) sudaro 0,172 % visų sandorių. Dėl konfidencialumo negalima pateikti pagrindinių funkcijų ir kitos papildomos informacijos apie įvykusius sandorius, todėl duomenys buvo transformuoti pagal pagrindinę komponentų analizę (angl. *Principal component analysis*) ir dėl šios priežasties duomenys turi tik skaitinę vertę (Lepoivre et al., 2016).

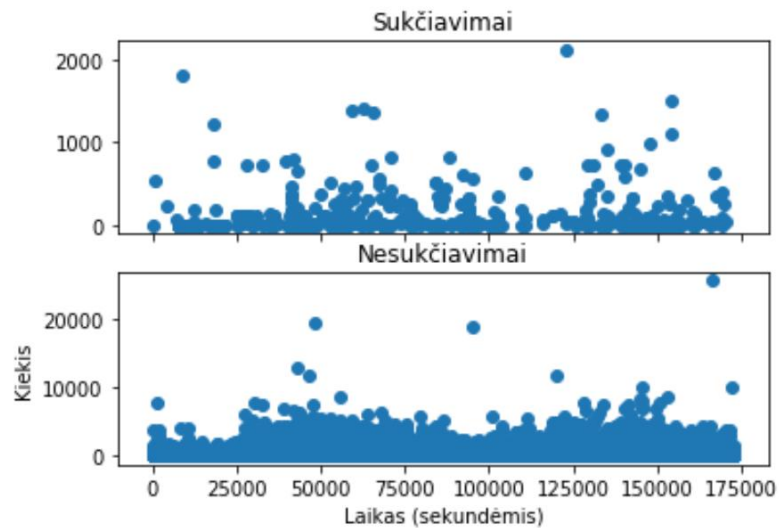
Ši duomenų bazė susideda iš V1-V28 kintamųjų, kurie buvo gauti naudojant pagrindinę komponentų analizės funkciją. Konkretus bankas, kuris taikytų specialius algoritmus sukčiavimo atpažinime, turėtų pilną informaciją ir galėtų identifikuoti, kurie požymiai geriausiai atspindi sukčiavimą. Tai leistų bankui imtis papildomų saugumo priemonių pagal konkretų požymį, bet šiuo atveju dėl konfidencialumo duomenys buvo transformuoti.



24 pav. Mokėjimų skaičius pagal klases

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze

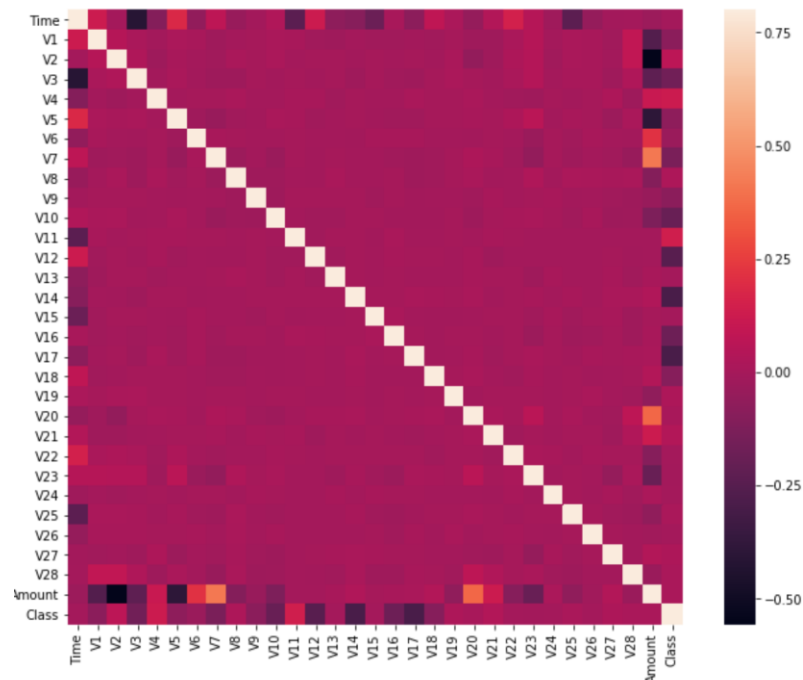
24 paveikslas vaizduoja pavedimų pasiskirstymą pagal sukčiavimus ir nesukčiavimus, kas akivaizdžiai rodo, kad sukčiavimų skaičius yra ženkliai mažesnis. Jei buvo įvykdytas sukčiavimas, tada pavedimas duomenų bazėje yra žymimas - 1, jei ne - 0. Taip pat svarbus vienas komponentas tai laikas, kuris šioje duomenų bazėje yra pavaizduojamas sekundžių tikslumu po kiekvieno įvykusio mokėjimo (žr. 25 pav.).



25 pav. Mokėjimo judėjimai laiko atžvilgiu pagal klases

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze

Toliau pavaizduota koreliacijos matrica rodo, kad nė vienas iš V1-V28 komponentų neturi tarpusavio koreliacijos. Tačiau jei atkreipsime dėmesį į klasę, ji turi tam tikrą formą teigiamą ir neigiamą koreliaciją su visais V komponentais, bet neturi ryšio su laiku ir suma (žr. 26 pav.)

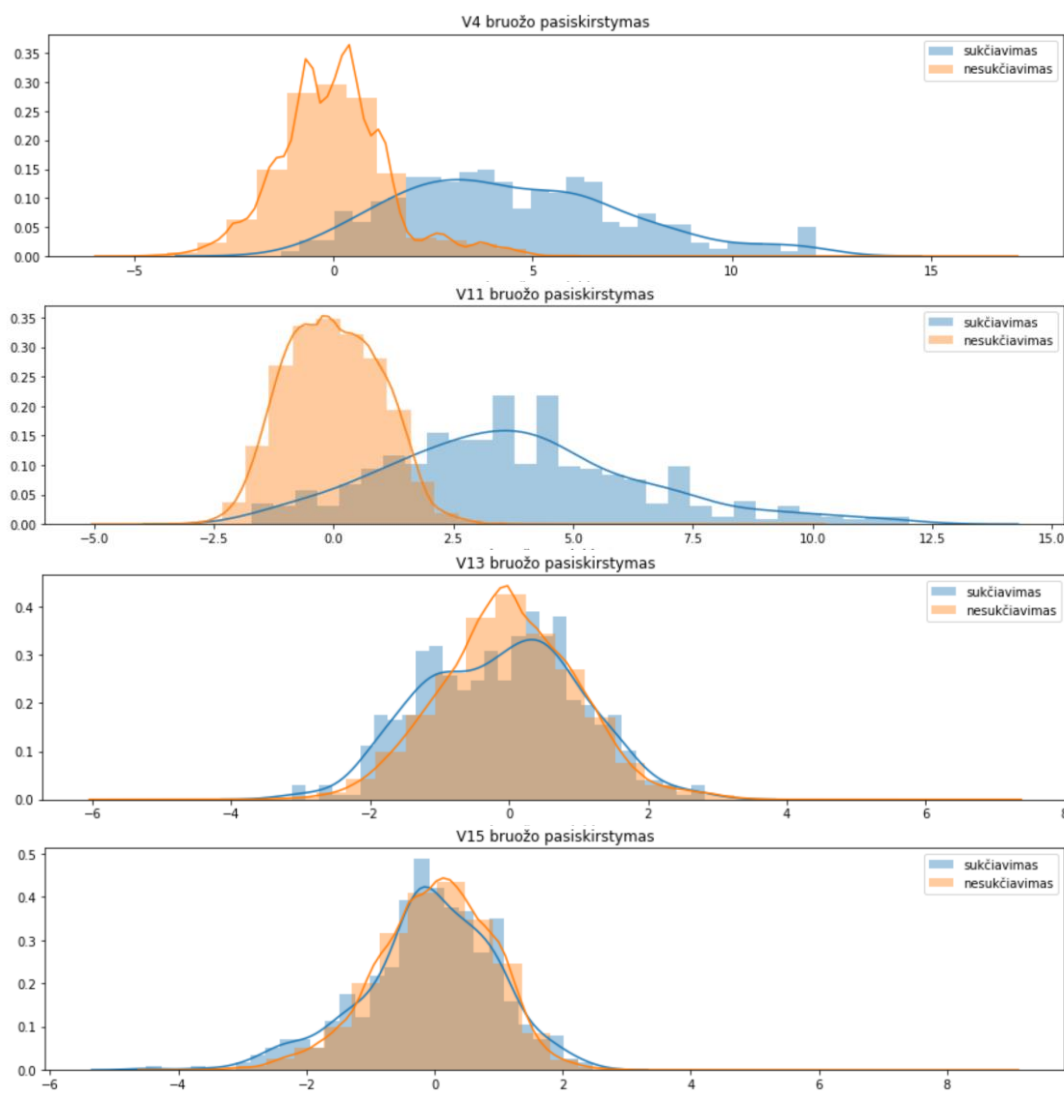


26 pav. Komponentų koreliacija

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze

3.1. Apibendrintas momentų metodas

Apibendrintam momentų metodui atlikti yra naudojamos pasiskirstymo (Gauso) kreivės tam, kad efektyviai būtų pasirenkami du bruožai iš V1-V28 ir nagrinėjamas jų pasiskirstymas (žr. 27 pav.). Buvo pasirinkti keli grafikai, vaizduojantys sukčiavimų ir nesukčiavimų pasiskirstymą, ir pagal jų duomenis atliekamas tolimesnis modelio pritaikymas.



27 pav. Bruožų pasiskirstymas

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze

Tyrimo metu apibendrintas momentų metodas (toliau - GMM) yra užpildomas mokymosi duomenimis, kai yra naudojami pasirinktieji bruožai. Sukuriamas toks duomenų rinkinys, kuriame bus tik nesąžiningos operacijos ir duomenų rinkinys su sąžiningomis operacijomis. Nesąžiningos operacijos išskaidomos po lygiai - 0,5 mokymuisi ir tiek pat

testavimui, o sąžiningos - 0,9 duomenų mokymuisi, o 0,1 testavimui atlikti. Kai modelis yra paleidžiamas su reikiamais testavimo duomenimis, apskaičiuojamos tikimybės. Jei tikimybė nesiekia ribos, sakysime, kad tai yra nesąžiningas sandoris. Tai yra todėl, kad modelis pagrįstas sąžiningais sandoriais. Maža tikimybė reiškia, kad nėra tikėtina, jog tam tikras sandoris yra nesąžiningas.

Galiausiai atsižvelgus į Gauso kreives ir sukčiavimų bei nesukčiavimų pasiskirstymus, buvo pasirinktos keturios tam tikrų bruožų kombinacijos ir apskaičiuoti jų tikslumą nusakantys rodikliai. 2 lentelėje yra pateikti pasirinktų efektyvumą nusakančių matų rezultatai: tikslumas, jautrumas, bendras tikslumo rodiklis ir F1 matas, kuris sujungia tikslumo ir jautrumo rezultatus. Siekiant įvertinti kaip modelis atskiria klases yra naudojama ROC kreivė, rodanti specifiškumą ir jautrumą, bet šiuo atveju, kai yra aktualus klasifikavimo atlikimas, galima ROC kreivę pritaikyti tikslumui nustatyti, jei yra apskaičiuojamas plotas po ja - AUC (angl. *Area Under Receiver Operating Characteristic Curve*).

2 lentelė. GMM modelio tikslingumas

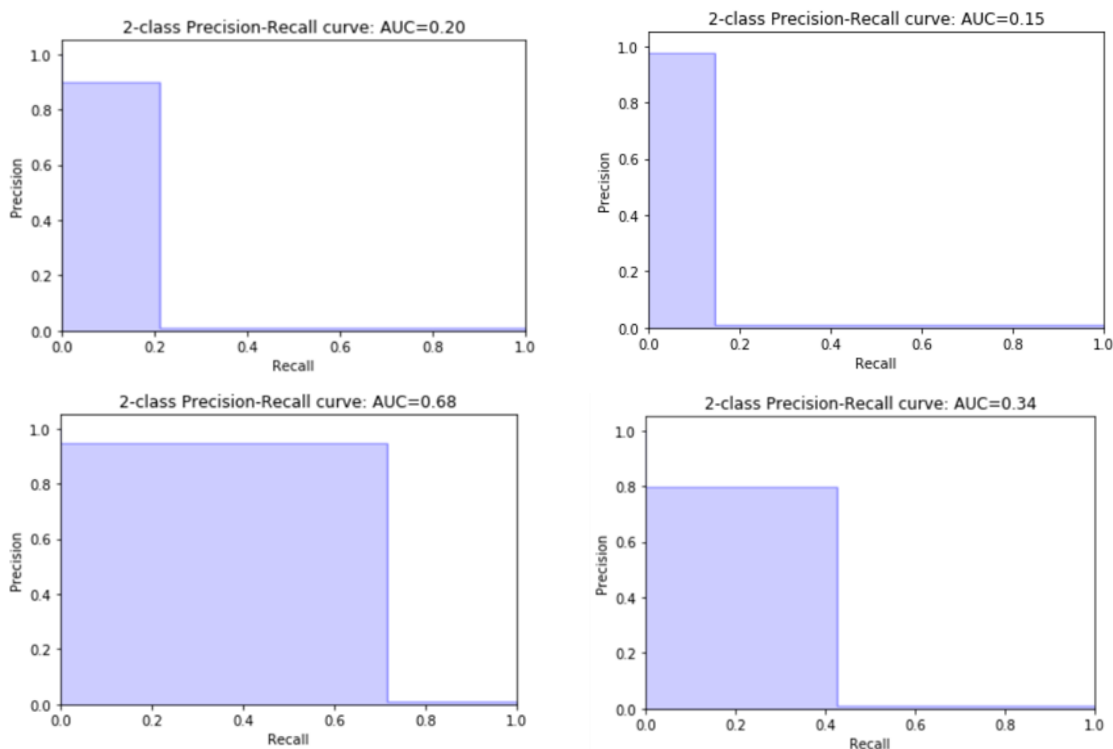
	Bruožai			
Tikslumo matai	V3 ir V4	V9 ir V11	V14 ir V17	V10 ir V12
Tikslumas (nesukčiavimas)	0,99	0,99	1,00	1,00
Tikslumas (sukčiavimas)	0,90	0,97	0,95	0,80
Jautrumas (nesukčiavimas)	1,00	1,00	1,00	1,00
Jautrumas (sukčiavimas)	0,21	0,15	0,72	0,43
F1 (nesukčiavimai)	1,00	1,00	1,00	1,00
F1 (sukčiavimai)	0,34	0,25	0,81	0,56
AUC	0,20	0,15	0,68	0,34

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze

Gauti rezultatai parodo, kad didžiausiu tikslumu pasižymi V14 ir V17 bruožai (žr. 2 lentelę). Pagal tikslumo rodiklį beveik visų bruožų rezultatai yra gana aukšti, o pagal jautrumo rodiklį rezultatai ženkliai mažesni. F1 matas, jungiantis šiuos rodiklius, parodo, kad jei tikslumas ir jautrumas yra išsiskiriantis, jis bus mažesnės reikšmės, ką šiuo atveju ir rodo gautas rezultatas. Jei tikslumas ir jautrumas sutaptų, F1 būtų lygus aritmetiniam vidurkiui.

Pagal AUC matą V14 ir V17 bruožai turi didžiausią rezultatą, siekiantį 0,68. Sudarytos kreivės pagal pasirinktus bruožus taip pat parodo savo rezultatą pagal didžiausią užpildytą plotą

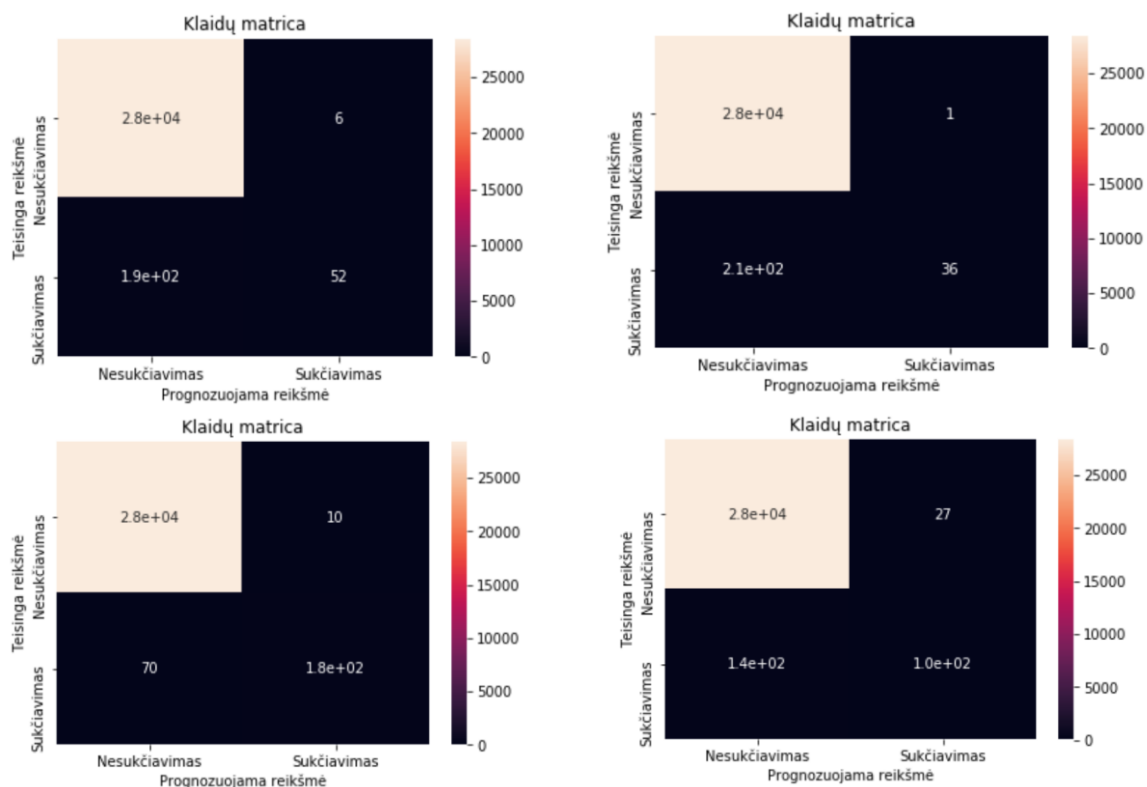
(žr. 28 pav.). Mažiausią AUC turi V9 ir V11 bruožai, bet šių bruožų F1 ir jautrumo matai yra patys žemiausi lyginant su kitomis bruožų kombinacijomis.



28 pav. AUC kreivės pagal GMM modelį

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze

Šiame tyrime taip pat buvo apskaičiuotos ir klaidų matricos pagal visus pasirinktus komponentus, nusakančius modelio tikslumą pagal prognozuojamas ir teisingas reikšmes (žr. 29 pav.). Visose matricose šviesūs kvadratai vaizduoja didžiausias reikšmes, kurios rodo, kad prognozės buvo įvykdytos teisingai ir pavaizduotas skaičius nesukčiavimų priskirtas nesukčiavimams pagal prognozę. Ten kur buvo nesukčiavimai, o po prognozės buvo priskirta sukčiavimams, daugiausiai suklysta buvo paskutinėje matricoje, kuri sudaryta iš V10 ir V12 bruožų. Įvertinus sukčiavimų atpažinimą, prasčiausiai atpažinimą rodo antroji matrica, sudaryta iš V9 ir V11 bruožų ir kurios tikslumą nusakantys rodikliai yra patys prasčiausi, o geriausią rezultatą vaizduoja trečioji matrica ir būtent jos AUC rodiklis ir kiti tikslumo matai yra patys didžiausi.



29 pav. GMM modelio klaidų matricos

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze

3.2. K-artimiausio kaimyno metodas

K-artimiausio kaimyno metodui įvertinti taip pat yra naudojami tokie patys tikslumą nusakantys matai, kaip ir prieš tai minėtame modelyje. Tačiau šiame metode jau yra įtraukiami visi bruožai, išskyrus laiką ir klasę.

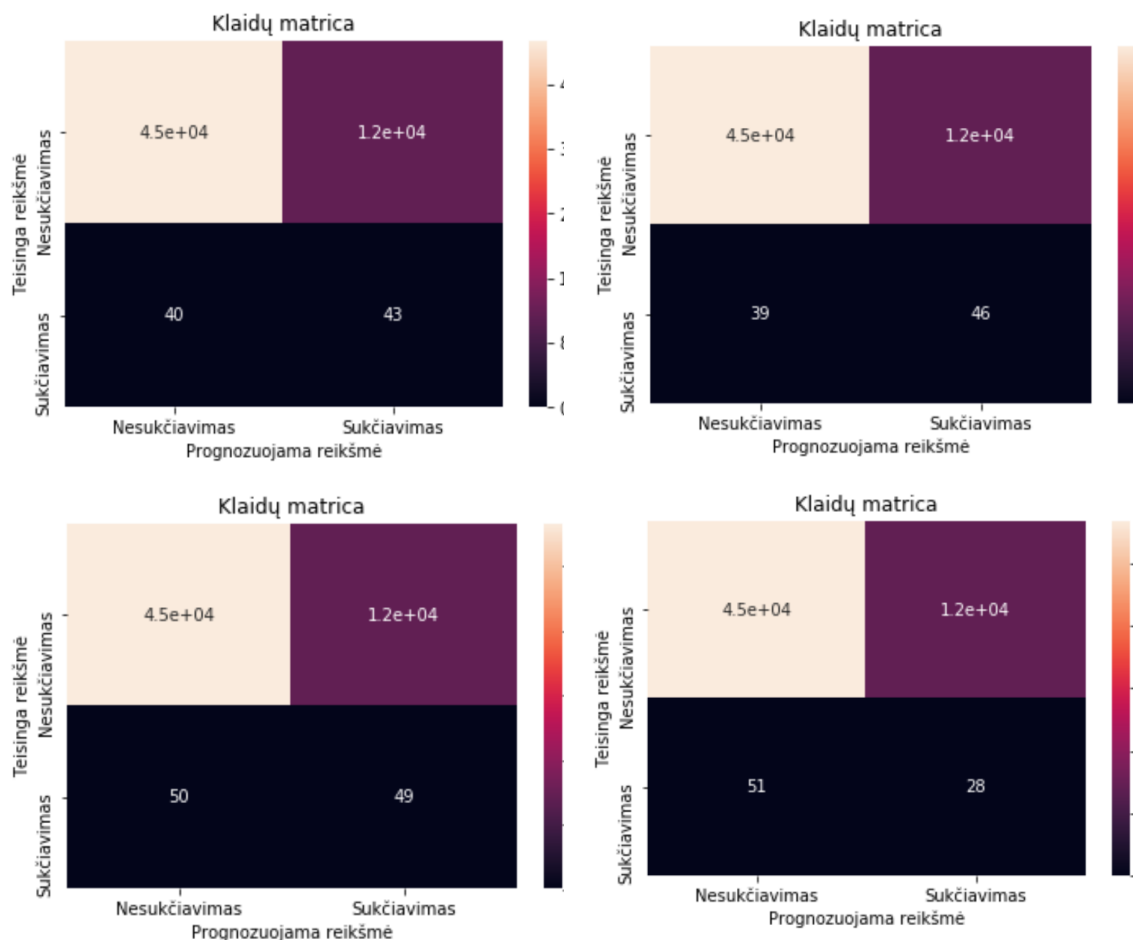
Mokymuisi yra skirta 0,8 dalis duomenų, o testavimui - likusi dalis. Tyrimo metu buvo keičiamas artimiausių kaimynų skaičius ir stebima, kiek tai turi įtakos tikslumą įvertinantiems rodikliams. Bendras klasifikavimo tikslumas yra labai aukštas atlikus visus keturis testavimus su skirtingais kaimynų skaičiais. Jis yra lygus 0.99, kas yra puikus rezultatas, tačiau juo nereikėtų visiškai pasikliauti (žr. 3 lentelę). Jautrumo rezultatas visais atvejais yra mažesnis nei tikslumo, bet vis tiek bendrai gana aukštas.

AUC rezultatas yra didžiausias, kai yra pasirenkamas vienas kaimynas, o mažiausias, kai yra pasirenkama dešimt kaimynų. O F1 matas, sujungiantis tikslumą ir jautrumą, yra siekiantis virš 0,7, kas vaizduoja vidutinį klasifikavimo įvertinimą. Įdomu tai, kad stipriai padidinus kaimynų skaičių, F1 matas ir tikslumas palaipsniui mažėja.

3 lentelė. Tikslumo rezultatai pagal k-artimiausio kaimyno metodą

	Kaimynų skaičius			
Tikslumo matai	1 kaimynas	5 kaimynai	10 kaimynų	20 kaimynų
Bendras klasifikavimo rodiklis	0,999	0,999	0,999	0,999
Tikslumas	0,780	0,873	0,9	0,794
Jautrumas	0,771	0,729	0,636	0,683
F1	0,778	0,795	0,746	0,735
AUC	0,895	0,822	0,701	0,838

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze



30 pav. Klaidų matricos pagal k-artimiausio kaimyno metodą

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze

Šiam metodui įvertinti taip pat yra pritaikomos klaidų matricos, kurios parodo, kaip modelis atpažįsta mokėjimus su sukčiavimais ir be jų. Rezultatai parodo, kad nesukčiavimai buvo atpažinti puikiai, bet visi sukčiavimai nebuvo tinkamai priskirti sukčiavimų klasei ir lyginant su prieš tai buvusi modeliu, rezultatai nėra tokie aukšti (žr. 30 pav.). Tad šiuo modeliu nelabai galima pasikliauti, nes kyla daug priešpriešų tarp tikslumo matų ir klaidų matricos.

3.3. Naivusis Bajeso klasifikatoriaus metodas

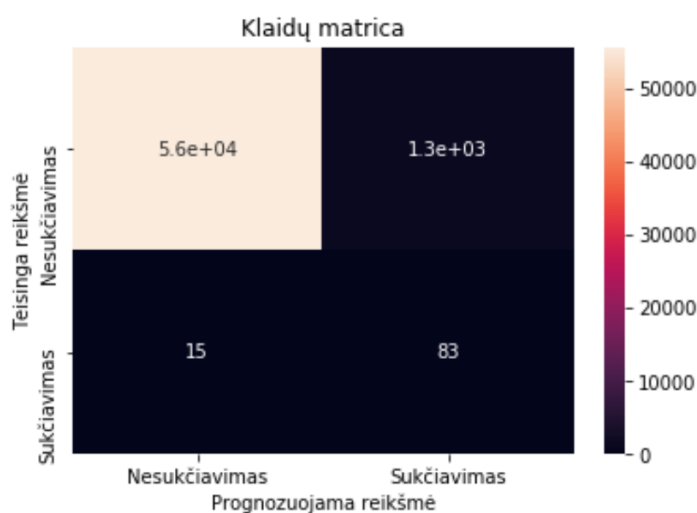
Kitas klasifikavimo metodas, kuris buvo pasirinktas atpažinti sukčiavimus, yra naivusis Bajeso metodas. Nuo pat pradžių šiame modelyje taip pat yra stebima, kokia sąsaja yra su laiku ir mokėjimų sumomis, bet galiausiai tolimesnei analizei atlikti yra naudojami tik V1-V28 bruožai. Kaip ir kiti modeliai, šio duomenys yra padalijami į 0,2 dalį testavimui ir 0,8 dalį mokymuisi. Šio modelio tikslumui įvertinti buvo pasirinkti tie patys matai ir klaidų matrica.

4 lentelė. Tikslumo rezultatai pagal Naive Bayes metodą

Tikslumo matai	
Bendras tikslumo matas	0,9764
Tikslumas	0,0587
Jautrumas	0,8469
F1	0,1099
AUC	0,9632

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze

Bendras modelio klasifikavimo tikslumas gana aukštas, siekiantis 0,98 tikslumą, bet tikslumo matas ypač žemas (žr. 4 lentelę). F1 rodiklis taip pat rodo labai nedidelį tikslumo atpažinimą, siekiantį vos 0,11. Nors AUC rezultatas yra labai aukštas, lygus 0,96, tai tik parodo, kad metodas yra nepatikimas kaip prieš tai analizuoti metodai.



31 pav. Klaidų matrica pagal Naive Bayes metodą

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze

Šio metodo klaidų matrica pagal testavimo duomenis rodo didesnius nuokrypius ir klaidingus mokėjimų priskyrimus sukčiavimams ir nesukčiavimams (žr. 31 pav.). Ženkliai daugiau nesukčiavimų yra priskiriami nesukčiavimams pagal prognozę nei prieš tai minėti metodai, bet gana daug nesukčiavimų yra priskiriami klaidingai sukčiavimams.

3.4. Giliojo mokymosi algoritmas

Modelis sudaromas iš mokymosi duomenų, o spėjimas iš testavimo. 20 proc. duomenų yra naudojami testavimui, likę 80 proc. - mokymuisi. Negalima mokyti ir testuoti tų pačių duomenų, nes tada modelis atpažintų 100 proc., atsirastų netikslumai. Būtent todėl ir reikia atskirti šias duomenų dalis. Prieš sudarant klasifikavimo modelį (klasifikatorių), būtina buvo atlikti pradinį duomenų apdorojimą: pašalinti nereikalingus kintamuosius ir įrašus, kuriuose yra praleistų reikšmių. Tam, kad būtų sudaromas klasifikavimo modelis, būtina padalinti turimą duomenų rinkinį į dvi nesikertančias dalis: modelio apmokymo duomenys - įprastai imama 80 proc. pradinio duomenų dalis (didžioji dalis) ir modelio testavimo duomenys - likusi nepanaudota 20 proc. duomenų dalis. Šio proceso tikslas naudojant apmokymo duomenis sudaryti klasifikatorių, kuris būtų tinkamas naudoti ir testavimo duomenims. Naudojant giliųjų neuroninių tinklų algoritmą ir apmokymui skirtus duomenis yra sudaromas klasifikatorius. Modelis yra pritaikomas ir testavimo duomenims. Kai klasifikatorius sudarytas ir išbandytas naudojant testinius duomenis, būtina patikrinti jo tikslumą.

Pirminiai neuronų svoriai yra nežinomi ir parenkami atsitiktinai, naudojant „random“ funkciją, o sluoksnių skaičius ir neuronų skaičius keičiamas automatiškai. Mokymosi norma taip pat yra nustatoma pagal tai, kuo ji yra mažesnė, tuo geriau mokosi neuroninis tinklas. Tyrimo metu

bandoma iteracijų skaičių didinti tam, kad būtų mokymąsi ilgiau ir galbūt efektyviau. Daugiau neuronų yra geriau, bet kartais gali įvykti persimokymas, todėl gali pradėti blogėti rezultatas su tokiais duomenimis.

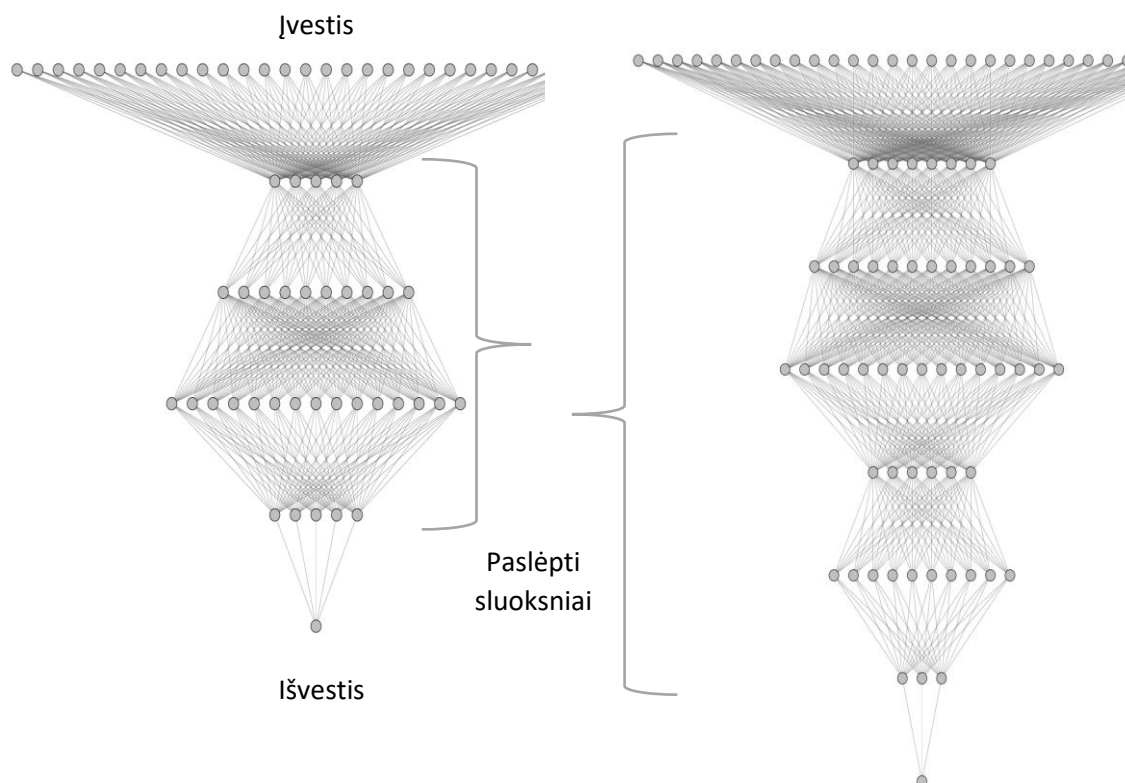
5 lentelė. Giliojo mokymosi testavimo rezultatai - 1 bandymas

Sluoksnių skaičius	Mokymosi greitis	Iteracijų skaičius	Tikslumas
1 sluoksnis - įvesties: 30 neuronų; 2 sluoksnis - paslėptas: 10 neuronų; 3 sluoksnis - paslėptas: 4 neuronų; 4 sluoksnis - išvesties: 1 neuronas	0,003	200	0,99090
1 sluoksnis - įvesties: 30 neuronų; 2 sluoksnis - paslėptas: 5 neuronų; 3 sluoksnis - paslėptas: 10 neuronų; 4 sluoksnis - paslėptas: 15 neuronų; 5 sluoksnis - paslėptas: 5 neuronai; 6 sluoksnis - išvesties: 1 neuronas.	0,001	400	0,99833
1 sluoksnis - įvesties: 30 neuronų; 2 sluoksnis - paslėptas: 8 neuronai; 3 sluoksnis - paslėptas: 12 neuronų; 4 sluoksnis - paslėptas: 15 neuronų; 5 sluoksnis - paslėptas: 6 neuronai; 6 sluoksnis - paslėptas: 10 neuronų; 7 sluoksnis - paslėptas: 3 neuronai; 8 sluoksnis - išvesties: 1 neuronas.	0,001	300	0,99810

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze

Giliajam mokymuisi atlikti yra naudojamos dvi aktyvavimo funkcijos - sigmoidinė ir ReLU. Modelis testuojamas su skirtingais hiperparametrais, kai yra keičiamas neuronų sluoksnių skaičius, neuronų skaičius tam tikruose sluoksniuose, epochų (iteracijų) skaičius, mokymosi norma (žr. 5 lentelę).

Pirmiausiai buvo testuojama vis keičiant sluoksnių skaičių ir neuronų skaičių, mokymosi greitį bandyta išlaikyti kuo mažesnę. Mokymosi greitis geba reaguoti į apskaičiuotą klaidą atsinaujinant svoriams (Brownlee, 2019). Kaip geba veikti neuroninis tinklas priklauso nuo pasirinktos architektūros. Jų gali būti tiek labai paprastų, tiek sudėtingesnės struktūros su skirtingomis funkcijomis. Bandymų su šešiais ir aštuoniais sluoksniais neuroninių tinklų architektūros yra pavaizduotos kiek žemiau (žr. 32 pav.). Gauti rezultatai parodė, kad didžiausią tikslumą turėjo tas testavimas, kurio mokymosi greitis buvo mažiausias, o iteracijų skaičius - didžiausias.



32 pav. Neuroninių tinklų architektūros pagal skirtingą sluoksnių ir neuronų pasiskirstymą
Šaltinis: sudaryta autorės, remiantis NN-architecture schematics programa

Atlikus kitą bandymą ir siekiant nustatyti modelio tikslumą, buvo keičiamas mokymosi greitis, iteracijų skaičius ir sudaromi penki sluoksniai:

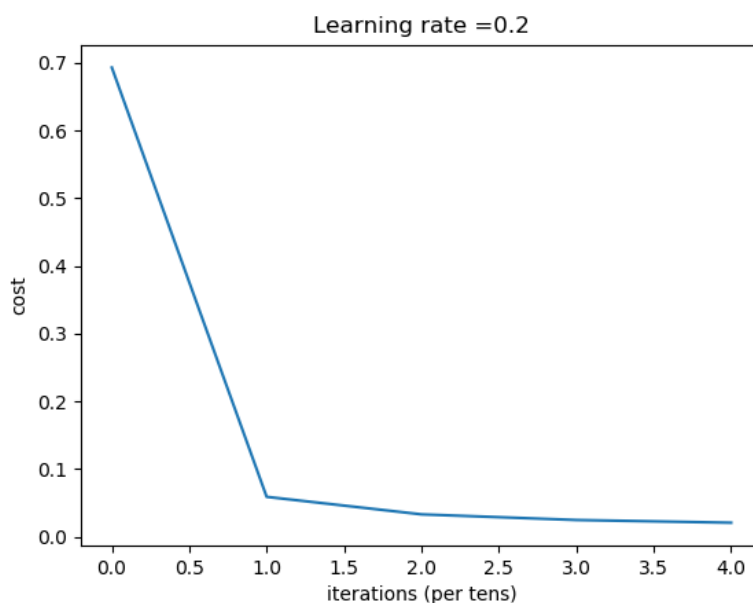
- 1 sluoksnis - įvesties: 30 neuronų;
- 2 sluoksnis - paslėptas: 20 neuronų;
- 3 sluoksnis - paslėptas: 10 neuronų;
- 4 sluoksnis - paslėptas: 5 neuronai;
- 5 sluoksnis - išvesties: 1 neuronas.

Kuo daugiau iteracijų neuroninis tinklas praeina, atitinkamai kaina (pastangos) turi mažėti (reikia vis mažiau pastangų atpažinti sukčiavimus). Gauti tikslumai yra gana panašūs, bet geriausi tie, kurių mokymosi greitis ir iteracijų skaičius yra mažiausias (žr. 6 lentelę). Taip pat, daugiau neuronų yra geriau, bet kartais gali įvykti persimokymas, todėl gali pradėti blogėti rezultatas. 33 paveiksle pavaizduota, kad bėgant laikui iteracijų skaičius kinta ir didėjant iteracijų skaičiui kaina didėja.

6 lentelė. Giliojo mokymosi testavimo rezultatai - 2 bandymas

Mokymosi greitis	Iteracijų skaičius	Tikslumas
0,99	200	0,99833
0,1	200	0,99844
0,05	200	0,99833
0,05	500	0,99803
0,2	500	0,99837
0,15	300	0,99839

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze



33 pav. Mokymosi grafikas - 2 bandymas

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze

Kitam bandymui atlikti buvo vėl keičiamas mokymosi greitis, iteracijų skaičius ir sudaromi penki sluoksniai su skirtingais neuronų kiekiais lyginant su prieš tai atliktu bandymu:

- 1 sluoksnis - įvesties: 30 neuronų;
- 2 sluoksnis - paslėptas: 50 neuronų;
- 3 sluoksnis - paslėptas: 80 neuronų;
- 4 sluoksnis - paslėptas: 7 neuronai;
- 5 sluoksnis - išvesties: 1 neuronas.

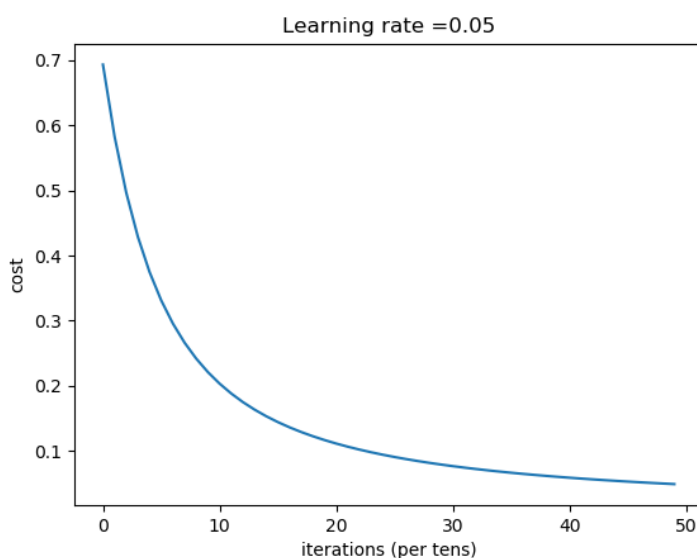
Įdomu tai, kad atliekant šį bandymą kyla hipotezė, kad įvyksta neuroninio tinklo persimokymas ties 1000 iteracijų ir mažiausiu mokymosi greičiu (žr. lentelę 7). Tad šiai hipotezei patvirtinti arba paneigti yra atliekamas papildomas testavimas nekeičiant mokymosi greičio, bet keičiant tik iteracijų skaičių.

7 lentelė. Giliojo mokymosi testavimo rezultatai - 3 bandymas

Mokymosi greitis	Iteracijų skaičius	Tikslumas
0,9	100	0,99805
0,1	200	0,99823
0,02	300	0,99832
0,05	500	0,99842
0,01	1000	0,99828

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze

Lyginant prieš tai minėtą grafiką su žemiau esančiu, matyti, kad mokymosi greitis turi didelę svarbą ir kuo jis yra mažesnis, tuo yra geriau (žr. 34 pav.). Pasirinkus atlikti papildomą tyrimą dėl kylančios hipotezės, gauti rezultatai tik patvirtino, kad neuroninis tinklas persimokė ties 1000 iteracijų ir todėl tikslumas pradėjo mažėti, nors mokymosi greitis išlieka gana žemas (žr. lentelę 8).

**34 pav. Mokymosi grafikas - 3 bandymas**

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze

8 lentelė. Giliojo mokymosi testavimo rezultatai - 4 bandymas

Mokymosi greitis	Iteracijų skaičius	Tikslumas
0,01	600	0,998069
0,01	700	0,998191
0,01	800	0,998367
0,01	900	0,998490

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze)

Testavimas buvo atliktas ir keičiant bruožų kiekį tam, kad suprasti, kurie kintamieji turi įtakos geriausiam modelio tikslumui atpažinti. Mokymosi greitis visais atvejais buvo

pasirinktas 0,001, o iteracijų skaičius 300. Gauti rezultatai parodė, kad didžiausias tikslumas gaunamas su V14 ir V17 bruožais taip pat kaip ir su apibendrintų momentų metodu.

9 lentelė. Giliojo mokymosi testavimo rezultatai - 5 bandymas

Bruožai	Sluoksnių skaičius	Tikslumas
V3 ir V4	2,15,3,1	0,99829
V9 ir V11	2,12,6,2,1	0,99805
V14 ir V17	2,10,6,4,2,1	0,99834
V10 ir V12	2,10,5,1	0,99821
V1-V28	28,12,8,5,1	0,99827

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze

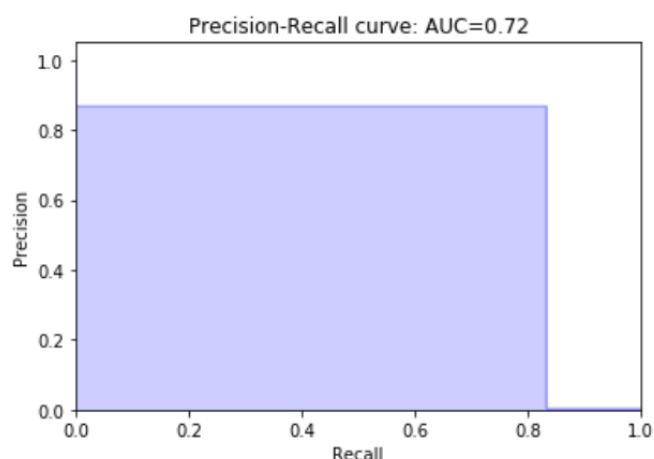
Istoriškai įprasta sukčiavimą atpažinti ir atpažinimą vertinti pagal mašininio mokymosi algoritmus. Tačiau giliojo mokymosi algoritmai įveda naują požiūrį. Modelis geba mokytis efektyviau ir turėti didesnę tikimybę atpažinti sukčiavimus. Giliojo mokymosi algoritmas yra labiau gebantis prisitaikyti prie naujų duomenų, anomalijų ir atlikti ne tik klasifikavimo funkciją kaip kiti metodai.

10 lentelė. Tikslumo rezultatai pagal Giliojo mokymosi metodą

Tikslumo matai	
Bendras tikslumo matas	0,9995
Tikslumas	0,87
Jautrumas	0,83
F1	0,85
AUC	0,72

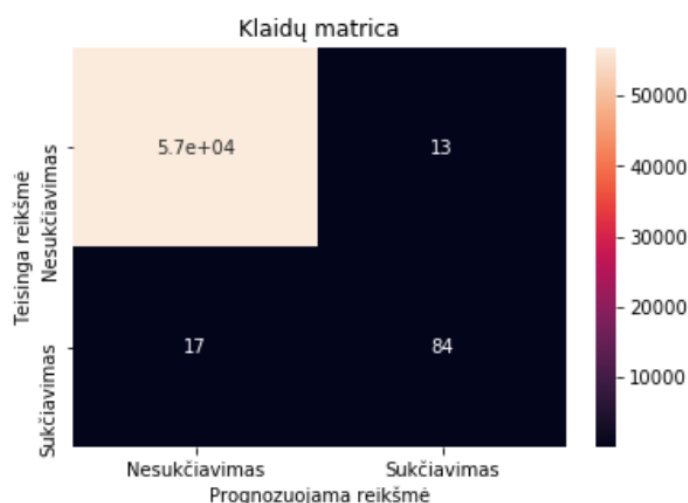
Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze)

Taip pat buvo suskaičiuoti ir kiti tikslumo rodikliai, kai yra pasirenkami keturi sluoksniai po 29, 15, 10, 1 neuroną ir pasirinkus mažiausią mokymosi rodiklį (žr. lentelę 10). Apskaičiuotas bendras tikslumo matas turi labai aukštą tikslumą, kiti matai taip pat. F1 rodiklis taip pat yra gana aukštas lyginant su klasifikavimo metodais. Žemiau pateiktas grafikas vizualizuoja AUC rezultata, siekiantį 0,72, o tai taip pat leidžia teigti, kad jo tikslumas yra geras.



35 pav. AUC kreivė pagal Gilijo mokymosi modelį

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze



36 pav. Klaidų matrica pagal Giliojo mokymosi metodą

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze

Buvo sudaroma ir klaidų matrica, pagal kurią galima spręsti, kaip tiksliai buvo atpažinti sukčiavimai ir nesukčiavimai, kiek kartų ir kur tiksliai modelis suklydo (žr. 36 pav.). Lyginant su kitais metodais, ši matrica vaizduoja geriausią rezultatą. Buvo suklysta 30 kartų priskiriant neteisingai sukčiavimus ir nesukčiavimus.

3.5. Finansinė sukčiavimo atvejus atpažįstančių modelių nauda

Modelių, pritaikomų atpažinti sukčiavimams, yra įvairiausių. Tiriamojoje dalyje buvo pasirinkti tik keli iš jų, kurie buvo pritaikyti tai pačiai duomenų bazei. Daugumos iš jų bendri

tikslumą nusakantys rodikliai turėjo gana gerus rezultatus, bet reikia atsižvelgti ir į kitus kriterijus, pagal kuriuos lengviau nuspręsti, kuris modelis yra geriausias.

Bankams yra svarbu tokius modelius turėti ir jais naudotis, nes tai kelia klientų pasitikėjimą, sumažina riziką dėl galimų nesąžiningų reiškinių, kurių pasekmės būtų skaudžios tiek klientams, tiek finansinei institucijai. Lentelėje žemiau yra matyti bendri visų modelių tikslumą nusakantys rodikliai bei galima finansinė nauda (žr. lentelę 10). Finansinė nauda buvo apskaičiuota pagal sukčiavimų atpažinimą ir bendrą jų sumą. Sukčiavimo suma sudauginama su tikslumo rodikliu ir gautas rezultatas parodo finansinę naudą, kiek pinigų gali būti sutaupyta, jei teisingai modelis atpažins sukčiavimus. Kuo didesnis modelio atpažinimas, tuo finansinė nauda yra didesnė. Jei modelio atpažinimas būtų žemas, didelė grėsmė, kad bankas patirs finansinius nuostolius ir nepastebės sukčiavimų.

11 lentelė. Finansinė modelių nauda

	Bendras tikslumas	Finansinė nauda
Apibendrintas momentų metodas	0,81	36918,6
K-artimiausio kaimyno metodas	0,765	47801,7
Naivusis Bajeso metodas	0,110	6614,1
Gilusis mokymasis	0,999	60097,9

Šaltinis: sudaryta autorės, remiantis Kaggle duomenų baze

Apibendrintą momentų metodą nebūtų teisinga lyginti su kitais metodais, nes ne visi bruožai buvo įtraukti į tikslumo rodiklių skaičiavimą. Bet siekiant įvertinti visus metodus, geriausiai atpažįsta dirbtiniu intelektu paremtas giliojo mokymosi algoritmas, kurio atpažinimas geriausias. Visiems šiems metodams buvo atitinkamai paskaičiuota finansinė nauda, kurią jie gali suteikti. Didžiausią finansinę naudą turi gilusis mokymasis, o mažiausią - naivusis Bajeso metodas, kurio tikslumas ypač žemas. Verta nepamiršti, kad visi analizuoti mokėjimai buvo tik kelių dienų rezultatas. Sukčiavimų įvyksta ženkliai daugiau ir pritaikius tikslingus giliojo mokymosi algoritmus finansinė nauda gali būti didžiulė. Juolab išlaikytas klientų pasitikėjimas ir finansinės įstaigos reputacija.

IŠVADOS IR PASIŪLYMAI

1. Dirbtinis intelektas įgauna vis didesnę svarbą ir dėl klestinčios jo koncepcijos, įmonės gali padaryti savo įrenginius galingesnius ir protingesnius, kad jie tinkamiau tarnautų savo klientams ir sėkmingai vykdytų užduotis. Dirbtinį intelektą galima pritaikyti įvairiausiose srityse: tiek portfelių sudarymui, tiek rizikos apskaičiavimui, tiek bendraujant su klientu apie kasdienes klausimus. Finansų sektoriuje gana dažnai yra naudojami virtualūs asistentai, kurie bendrauja su klientais visą parą ir atsako į jiems rūpimus klausimus. Taip pat kuriamos programos, kurios palengvina daug procesų, juos automatizuoja, leidžia rasti anomalijas ir jas išspręsti, pavyzdžiui, mokėjimų sukčiavimų atpažinimas.

Viena didžiausių baimių, apie kurias yra kalbama, kai išgirstama sąvoka dirbtinis intelektas - darbo rinkos situacija. Tiesa, daug darbo vietų gali išnykti, bet taip pat dėl dirbtinio intelekto ir atsirasti. Galbūt žmonės gali atlikti protingesnius ir naudingesnius darbus, mažiau techninius, skatinančius daugiau mąstyti, kurti, galvoti daugiau idėjų ateičiai. Galbūt dirbtiniam intelektui vis labiau įsitraukiant į mūsų gyvenimus nebeturėtų egzistuoti standartinė darbo savaitė, gal tą laiką būtų galima išnaudoti kur nors produktyviau.

Bankai diegdami naujas technologijas turi prisitaikyti. Tai padaryti nėra lengva, kai jų sistemos yra senesnės, didelis skaičius duomenų ir klientų. Bankai turi atsižvelgti į pasaulyje vykstančių technologijų pokyčius, nes daugybė sektorių tampa vis labiau automatizuoti ir priklausantys nuo robotų ir juos valdančių algoritmų. Dirbtinio intelekto įsitraukimas į bankų veiklą palengvina daug procesų, juos automatizuoja, leidžia rasti anomalijas ir jas išspręsti. Susiduriama ir su baimėmis, kad dirbtinis intelektas gali pakeisti daugybės žmonių darbą, bet taip pat žmonės galės efektyviau dirbti, atlikti kūrybingesnes užduotis, kurti naujas idėjas.

2. Egzistuoja daugybė algoritmų, kurie yra susiję su dirbtiniu intelektu, o jų panaudojimas priklauso nuo paties uždavinio ir siektinų tikslų. Kadangi pats tyrimas buvo susijęs su sukčiavimo atpažinimu, buvo pasirinkti labiausiai paplitę algoritmai, gebantys šią problemą išspręsti. Algoritmų, atpažįstančių sukčiavimus, yra gana daug, bet patys bankai nelinkę atskleisti jų dėl konfidencialumo ir netinkamo panaudojimo grėsmės. Daug iš jų atlieka klasifikavimo funkciją, pavyzdžiui, sprendimų medis, k-artimiausio kaimyno metodas, bet pasirinktas gilusis mokymasis išsiskyrė tiek savo kiek kitokia struktūra, sudaryta iš svarių, sluoksnių, neuronų skaičiaus ir kitų parametrų, bet pateikė ir geriausią tyrimo rezultatą

3. Šiomis dienomis ypač daug yra kalbama apie sukčiavimų atpažinimą naudojant dirbtinį intelektą ar kitokius algoritmus, nes, deja, ne visus procesus gali atpažinti pats žmogus. Sukčiavimai gali būti aptinkami pagal sukčiavimo ir naudotojo analizę. Sukčiavimo analizė

išsiskiria tuo, kad yra nagrinėjami istoriniai duomenys, stebimi išsiskiriantys mokėjimai. Tam yra naudojami klasifikavimo metodai ir neuroniniai tinklai, kurie aptinka daugiausiai sukčiavimų. Naudotojo analizę sunkiau pasikliauti, nes reikia tikslaus vartotojo elgesio modelio ir dažnai vyrauja didesnė paklaida. Sukčiavimo aptikimas teikia didelę pridėtinę vertę, o įvykęs greitas reagavimas gali sumažinti didžiulius nuostolius. Galiausiai taip pat yra didinamas kliento pasitikėjimas, išsaugomas geras banko vardas ir sumažėja rizika prarasti vertingus klientus.

4. Tyrimo metu buvo pasirinkti apibendrinto momentų, k-artimiausio kaimyno, naivusis Bajeso ir giliojo mokymosi metodai. Tiriant pasirinktus metodus buvo siekiama pritaikyti tinkamiausius parametrus, nustatyti tinkamas architektūras, kad atpažinimo rezultatas būtų kuo geresnis. Visi modeliai buvo apmokomi su tam tikru kiekiu mokymosi duomenų, o po to testuojami su testavimui skirtais duomenimis. Tada apskaičiuojami ir įvertinami tikslumą nusakantys rodikliai: bendras klasifikavimo tikslumas, jautrumas, tikslumas, F1 rodiklis, AUC kreivė ir klaidų matrica.

5. Gauti rezultatai parodė, kad didžiausią atpažinimą pasiekė gilusis mokymasis, naudojantis neuroninius tinklus. Jo atpažinimas siekė 99,9 proc. Gilusis mokymasis yra linkęs labiau prisitaikyti prie naujų duomenų ir atlikti ne tik klasifikavimo funkciją kaip kiti metodai. Su šiuo metodu buvo keičiama daug parametrų, bet tikslumas visais bandymais buvo vis tiek aukštas. Prasčiausią sukčiavimų atpažinimą turėjo naivusis Bajeso metodas. Jo tikslumo ir F1 rodikliai buvo itin žemi, o klaidų matrica vaizdavo didelius nuokrypius.

6. Sukčiavimų atpažinimas bankams teikia finansinę naudą, kuri priklauso nuo to, kaip tiksliai bus atpažįstami sukčiavimai ir laiku imtasi veiksmų juos sustabdyti. Tyrimo metu pasirinkti duomenys vaizdavo kelių dienų mokėjimus, todėl gauta finansinė nauda, naudojant pasirinktą dviejų dienų rinkinį, yra nedidelė. Tačiau jei duomenys būtų pusmetiniai ar metiniai ir pritaikomi tiksliniai metodai, finansinė nauda būtų ženkliai didesnė, išlaikytas klientų pasitikėjimas ir banko reputacija.

LITERATŪROS SĄRAŠAS

- Allan, K. (2018). *How banking is adopting and using AI technology*. Prieiga per internetą: <https://www.idgconnect.com/idgconnect/analysis-review/1018981/banking-adopting-ai-technology>
- Bhatla, T. P., Prabhu, V., & Dua, A. (2003). *Understanding credit card frauds*. *Cards business review*, 1(6).
- Botsman, R. (2017). *Kuo galite pasitikėti? Kaip technologijos mus suartino ir kodėl gali išskirti*. Vilnius: Eugrimas
- Bhusari, V., & Patil, S. (2011). *Application of Hidden Markov Model in Credit Card Fraud Detection*. *International Journal of Distributed and Parallel Systems*, 2(6), 203.
- Brownlee, J. (2016). *What is Deep Learning?* Prieiga per internetą: <https://machinelearningmastery.com/what-is-deep-learning/>
- Brownlee. (2018). *How and When to Use ROC Curves and Precision-Recall Curves for Classification in Python*. Prieiga per internetą: <https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/>
- Brownlee, J. (2019). *Understand the Impact of Learning Rate on Neural Network Performance*. Prieiga per internetą: <https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/>
- Browlee, J. (2019). *A Gentle Introduction to the Rectified Linear Unit (ReLU)*. Prieiga per internetą: <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>
- Browlee J. (2014). *Classification Accuracy is Not Enough: More Performance Measures You Can Use*. Prieiga per internetą: <https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/>
- Chandola, V., Banerjee, A., & Kumar, V. (2009). *Anomaly detection: A survey*. *ACM computing surveys (CSUR)*, 41(3), 15.
- Chollet, F. (2017). *The impossibility of intelligence explosion*. Prieiga per internetą: <https://medium.com/@francois.chollet/the-impossibility-of-intelligence-explosion-5be4a9eda6ec>
- Cilimkovic, M. (2015). *Neural Networks and Back Propagation Algorithm*. Institute of Technology Blanchardstown, Blanchardstown Road North Dublin, 15.

- Crosman, P. (2018). *Where Bank of America uses AI, and where its worries lie*. Prieiga per internetą: <https://www.americanbanker.com/news/where-bank-of-america-uses-artificial-intelligence-and-where-its-worries-lie>
- Davis, J., & Goadrich, M. (2006). *The relationship between Precision-Recall and ROC curves*. In Proceedings of the 23rd international conference on Machine learning (pp. 233-240). ACM.
- Deloitte. *The new physics of financial services. How artificial intelligence is transforming the financial ecosystem*. Prieiga per internetą: <https://www2.deloitte.com/content/dam/Deloitte/uk/Documents/financial-services/deloitte-uk-world-economic-forum-artificial-intelligence-summary-report.pdf>
- Draeos, R. (2019). *Measuring Performance: AUPRC*. Prieiga per internetą: <https://glassboxmedicine.com/2019/03/02/measuring-performance-auprc/>
- D'Souza, S. (2018). *Let's learn about AUC ROC Curve!* Prieiga per internetą: <https://medium.com/greyatom/lets-learn-about-auc-roc-curve-4a94b4d88152>
- European Banking Federation. (2018). *Banking in Europe: EBF Facts & Figures 2018*.
- European Banking Federation. (2018). *Financial technology*. Prieiga per internetą: <https://www.ebf.eu/facts-andfigures/financial-technology/>
- European Central Bank. (2018). *Card fraud report*. Prieiga per internetą: <https://www.ecb.europa.eu/pub/cardfraud/html/ecb.cardfraudreport201809.en.html#toc1>
- Fawcett, T. (2005). *An introduction to ROC analysis*. Pattern recognition letters, 27(8), 861-874.
- Fico. (2017). *Evolution in card fraud in Europe 2017. Data Compromise Overshadows Improvements Across Europe*. Prieiga per internetą: <https://www.fico.com/europeanfraud/index>
- Flach, P., & Kull, M. (2015). *Precision-recall-gain curves: PR analysis done right*. In Advances in neural information processing systems (pp. 838-846).
- Gandhi, R. (2018). *Support Vector Machine - Introduction to Machine Learning Algorithms*. Prieiga per internetą: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>
- Ghahramani, Z. (2001). *An introduction to hidden Markov models and Bayesian networks*. In Hidden Markov models: applications in computer vision (pp. 9-41).
- Ghosh, A., Tweedale, J. W., Nafalski, A., & Dollard, M. (2012). *Multi-Agent Based System for Analysing Stress using the StressCafé*. In KES (Vol. 243, pp. 1656-1665).
- Goertzel, K. & Pennachin, C. (2007). *Artificial General Intelligence*.

- Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep Learning*.
- Gupta, G. (2017). *Fundamentals of Deep Learning - Activation Functions and When to Use Them?* Prieiga per internetą: <https://www.analyticsvidhya.com/blog/2017/10/fundamentals-deep-learning-activation-functions-when-to-use-them/>
- Hayes, M. A., & Capretz, M. A. (2014). *Contextual anomaly detection in big sensor data*. In 2014 IEEE International Congress on Big Data (pp. 64-71). IEEE.
- Haykin, S. (2009). *Neural networks and learning machines* (3th ed.).
- Hexagon, C. (2018). *How did you benefit from machine learning today?* Prieiga per internetą: <https://www.crimsonhexagon.com/blog/how-did-you-benefit-from-machine-learning-today/>
- Hutchens, J. L. (1996). How to pass the Turing test by cheating. School of Electrical, Electronic and Computer Engineering research report TR97-05. Perth: University of Western Australia.
- Yee, O. S., Sagadevan, S., & Malim, N. H. A. H. (2018). *Credit Card Fraud Detection Using Machine Learning As Data Mining Technique*. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 10(1-4), 23-27
- Jha, V. (2017). *Machine Learning Algorithm - Backbone of emerging technologies*. Prieiga per internetą: <https://www.techleer.com/articles/203-machine-learning-algorithm-backbone-of-emerging-technologies/>
- Johnston, M. (2018). *Seizing the moment for artificial intelligence*. Prieiga per internetą: <https://blog.evergreengavekal.com/seizing-the-moment-for-artificial-intelligence/>
- Karlik, B., & Olgac, A. V. (2011). *Performance analysis of various activation functions in generalized MLP architectures of neural networks*. International Journal of Artificial Intelligence and Expert Systems, 1(4), 111-122.
- Khan, A. U. S., Akhtar, N., & Qureshi, M. N. (2014). *Real-time credit-card fraud detection using artificial neural network tuned by simulated annealing algorithm*. In Proceedings of International Conference on Recent Trends in Information, Telecommunication and Computing, ITC (pp. 113-21).
- Kiran, S., Kumar, N., Guru, J., Katariya, D., Kumar, R., & Sharma, M. (2018). *Credit card fraud detection using Naïve Bayes model based and KNN classifier*. International Journal of Advance Research, Ideas and Innovations in Technology, 4(3).

- Kumar, N. (2007). *Advantages and Disadvantages of KNN Algorithm in Machine Learning*. Prieiga per internetą: <http://theprofessionalspoint.blogspot.com/2019/02/advantages-and-disadvantages-of-knn.html>
- Lee, M. K. (2018). *Understanding perception of algorithmic decisions: Fairness, trust, and emotion in response to algorithmic management*. *Big Data & Society*, 5(1), 2053951718756684.
- Lepoivre, M. R., Avanzini, C. O., Bignon, G., Legendre, L., & Piwele, A. K. (2016). *Credit card fraud detection with unsupervised algorithms*. *Journal of Advances in Information Technology*, 7(1).
- Ling, C. X., Huang, J., & Zhang, H. (2003). *AUC: a statistically consistent and more discriminating measure than accuracy*. In *Ijcai* (Vol. 3, pp. 519-524).
- Love, B. C. (2002). *Comparing supervised and unsupervised category learning*. *Psychonomic bulletin & review*, 9(4), 829-835.
- Lu, Y. 2017. *Deep neural networks and fraud detection*.
- Mahmood, H. (2018). *Activation Functions in Neural Networks*. Prieiga per internetą: <https://towardsdatascience.com/activation-functions-in-neural-networks-83ff7f46a6bd>
- Maes, S., Tuyls, K., Vanschoenwinkel, B., & Manderick, B. (2002). *Credit card fraud detection using Bayesian and neural networks*. In *Proceedings of the 1st international nairo congress on neuro fuzzy technologies* (pp. 261-270).
- Makūnaitė, R. (2006). *Neuroniniai tinklai*. Prieiga per internetą: <http://www.elektronika.lt/teorija/kompiuterija/4342/neuroniniai-tinklai/>
- Manning, J. (2018). *How AI is disrupting the banking industry*. Prieiga per internetą: <https://internationalbanker.com/banking/how-ai-is-disrupting-the-banking-industry/>
- Marchisio, A., Hanif, M. A., Rehman, S., Martina, M., & Shafique, M. (2018). *A Methodology for Automatic Selection of Activation Functions to Design Hybrid Deep Neural Networks*. arXiv preprint arXiv:1811.03980.
- Marria, V. (2018). *Is Artificial Intelligence Replacing Jobs In Banking?* Prieiga per internetą: <https://www.forbes.com/sites/vishalmarria/2018/09/26/is-artificial-intelligence-replacing-jobs-in-banking/#72eebc73c55>
- Maršalkytė, G. (2018). *Dirbtinis intelektas, mašininis mokymasis ir gilieji tinklai: kas slepiasi už šių savokų?* Prieiga per internetą: <http://techo.lt/dirbtinis-intelektas-masininis-mokymasis-ir-gilieji-tinklai-kas-slepiasi-uz-siu-savoku/>

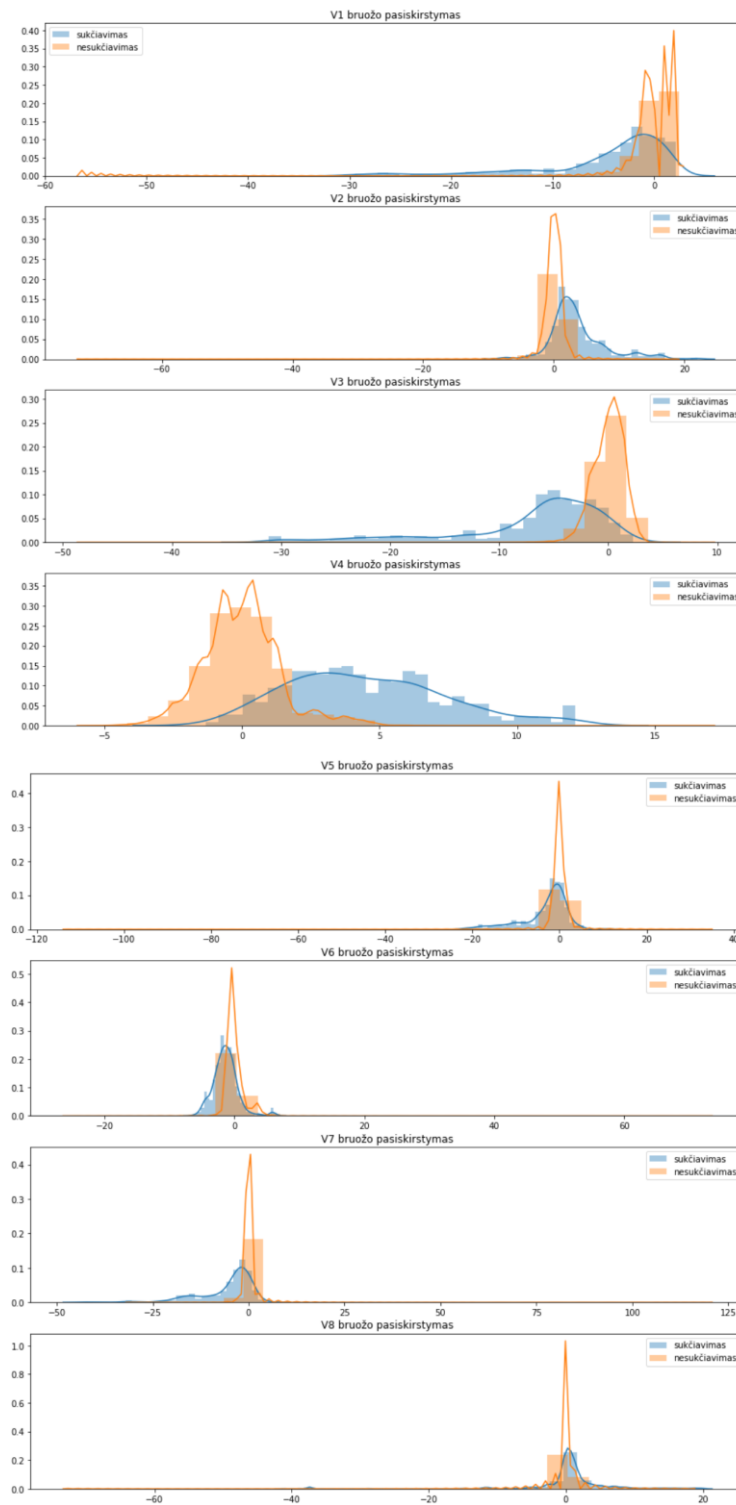
- Mathworks. (n.d.). *Supervised Learning Workflow and Algorithms*. Prieiga per internetą: <https://www.mathworks.com/help/stats/supervised-learning-machine-learning-workflow-and-algorithms.html>
- Mercioni, M. A., Tiron, A. & Holban, S. (2019). *Dynamic Modification of Activation Function using the Backpropagation Algorithm in the Artificial Neural Networks*. International Journal of Advanced Computer Science and Applications, 10(4).
- Mills, F. & Stuffebeam, R. (2005). *Introduction to intelligent agents*. Illinois State University.
- Moon, J., Park, S., Rho, S., & Hwang, E. (2019). *A comparative analysis of artificial neural network architectures for building energy consumption forecasting*. International Journal of Distributed Sensor Networks, 15(9), 1550147719877616.
- Morrison, A. & Rao, A. (2017). *Machine learning methods (infographic)*. Prieiga per internetą: <http://blog.else-corp.com/2017/10/machine-learning-methods-infographic-pwc/>
- Narkhede, S. (2018). *Understanding Confusion Matrix*. Prieiga per internetą: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>
- Narkhede, S. (2018). *Understanding AUC - ROC Curve*. Prieiga per internetą: <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>
- Nets. (2018). *European Fraud Report - Payments Industry Challenges*. Payments Industry Intelligence Payments Cards & Mobile.
- Nwazor, T. (2019). *Artificial Intelligence Is Changing Banking and Wall Street. Here's What to Expect in 2019*. Prieiga per internetą: <https://www.entrepreneur.com/article/327660>
- Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). *Activation functions: Comparison of trends in practice and research for deep learning*. arXiv preprint arXiv:1811.03378.
- Omar, S., Ngadi, A., & Jebur, H. H. (2013). *Machine learning techniques for anomaly detection: an overview*. International Journal of Computer Applications, 79(2).
- Patil, V., & Lilhore, U. K. (2018). *A Survey on Different Data Mining & Machine Learning Methods for Credit Card Fraud Detection*.
- Perera, S. (2015). *Introduction to Anomaly Detection: Concepts and Techniques*. Prieiga per internetą: https://iwringer.wordpress.com/2015/11/17/anomaly-detection-concepts-and-techniques/?fbclid=IwAR2e3Q_2fCVl2WYvzSd8teojVyu0tq9jff9vywGiA5-Bli6VqM67ELsOsno

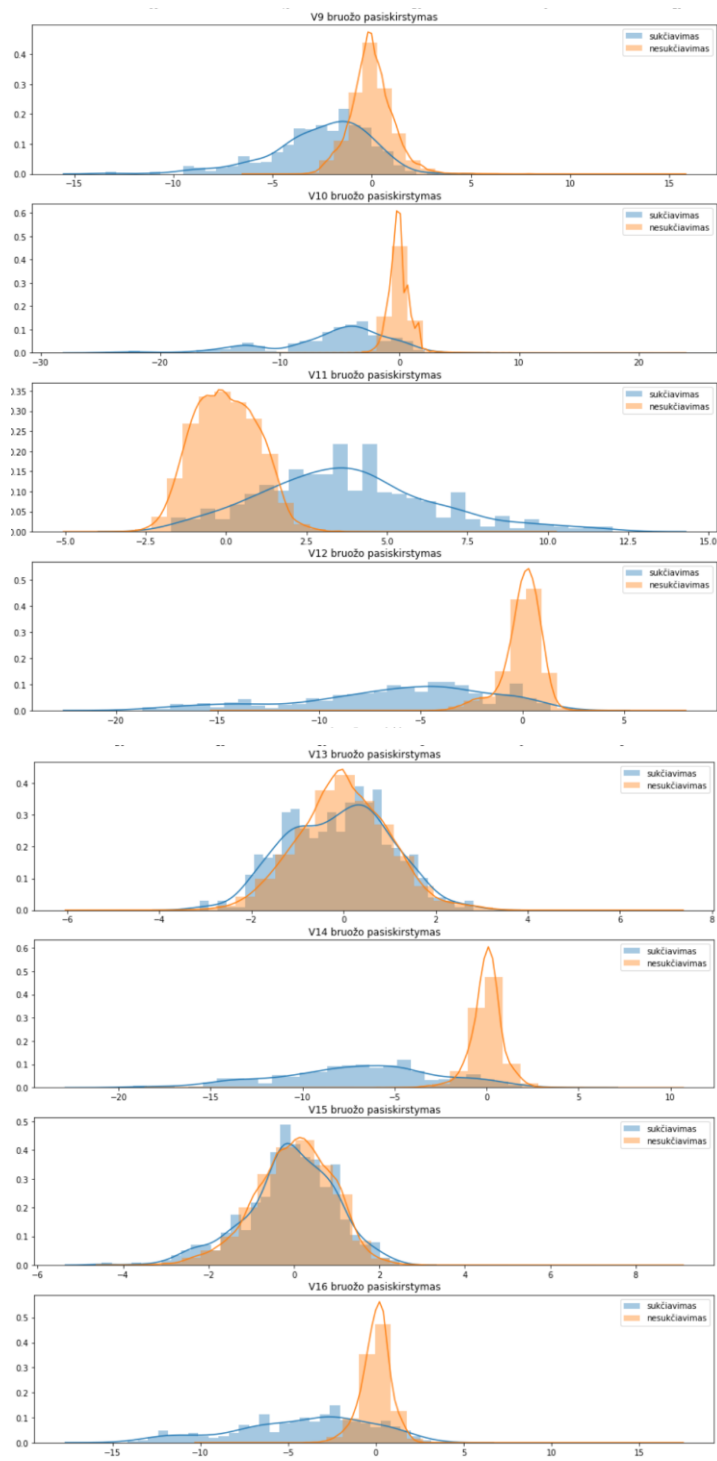
- Pumsirirat, A. & Yan, L. (2018). *Credit Card Fraud Detection using Deep Learning based on Auto-Encoder and Restricted Boltzmann Machine*. International Journal of Advanced Computer Science and Applications, 9(1), 18-25.
- Prakash, A. (2017). *Analysis of the modern techniques and methods on credit card fraud detection*. International Journal for Research in Science Engineering & Technology, 4(6), 12-18.
- Pratuzaite, G. & Maknickienė, N. (2019). *Kreditinių kortelių sukčiavimo atpažinimo tyrimas*. Iš Lietuvos jaunųjų mokslininkų konferencija „Mokslas - Lietuvos ateitis“. Ekonomika ir vadyba.
- Rao, A. & Verweij, G. (2018). *Sizing the prize what's the real value of AI for your business and how can you capitalise?* Prieiga per internetą: <https://www.pwc.com/AI>
- Riley, B. (2018). *The 18 Top Use Cases of Artificial Intelligence in Banks*. Prieiga per internetą: <https://www.fintechnews.org/the-18-top-use-cases-of-artificial-intelligence-in-banks/>
- Scherrer, B. (2007). *Gaussian mixture model classifiers*. Lecture Notes, February.
- Schmidhuber, J. (2015). *Deep learning in neural networks: An overview*. Neural networks, 61, 85-117.
- Seidler, K. (2018). *The Latest Credit Card Fraud Statistics and Insights*. Prieiga per internetą: <https://losspreventionmedia.com/credit-card-fraud-statistics-and-insights>
- Sennaar, K. (2019). *AI in Banking - An Analysis of America's 7 Top Banks*. Prieiga per internetą: <https://emerj.com/ai-sector-overviews/ai-in-banking-analysis/>
- Serrano, A. M. R., da Costa, J. P. C. L., Cardonha, C. H., Fernandes, A. A., & de Sousa Júnior, R. T. (2012). *Neural Network Predictor for Fraud Detection*. ICoFCS 2012, 61.
- Sharma, A. (2017). *Understanding Activation Functions in Neural Networks*. Prieiga per internetą: <https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0>
- Shinde, P. (2018). *Credit Card Fraud Detection using Deep learning models*. Prieiga per internetą: <https://mc.ai/credit-card-fraud-detection-using-deep-learning-models/>
- Shook, E. J., Knickrehm, M. A., McIntype, A., Woolf, A., Browne, Y. & Lavelle, K. (2018). *Future workforce survey - banking realizing the full value of AI*.
- Shwartz, S. S. & David, S. B. (2014). *Understanding Machine Learning: From Theory to Algorithms*. New York: Cambridge University Press
- Silva, I. N., Spatti, D. H. & Flauzino, R. A. (2017). *Artificial Neural Network Architectures and Training Processes. Chapter 2*.

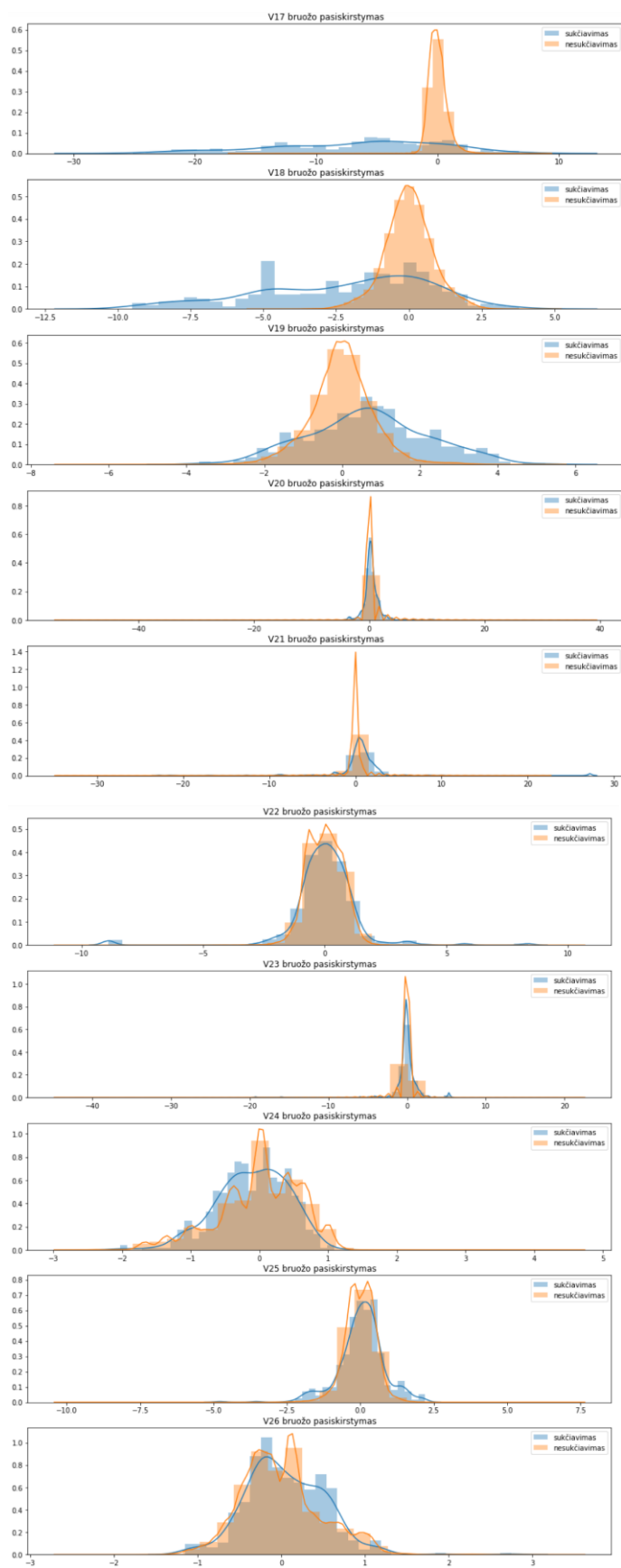
- Song, Y. & Lu, Y. (2015). *Decision tree methods: applications for classification and prediction*. Shanghai archives of psychiatry, 27(2), 130.
- Veen, V. F. (2016). *The neural network zoo*. Prieiga per internetą: <http://www.asimovinstitute.org/neural-network-zoo/>
- Tichonov, J. (2018). *Klasifikavimo metodais grindžiami skaitmeninių vaizdų glaudinimo sprendimai*: Daktaro disertacija. Vilnius: Vilniaus universitetas.
- Walia, A. S. (2017). *Activation functions and it's types-Which is better?* Prieiga per internetą: <https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f>
- Xu, S. (2018). *Bayesian Naïve Bayes classifiers to text classification*. Journal of Information Science, 44(1), 48-59.
- Zojaji, Z., Atani, R. E., & Monadjemi, A. H. (2016). *A survey of credit card fraud detection techniques: Data and technique oriented perspective*. arXiv preprint arXiv:1611.06439.

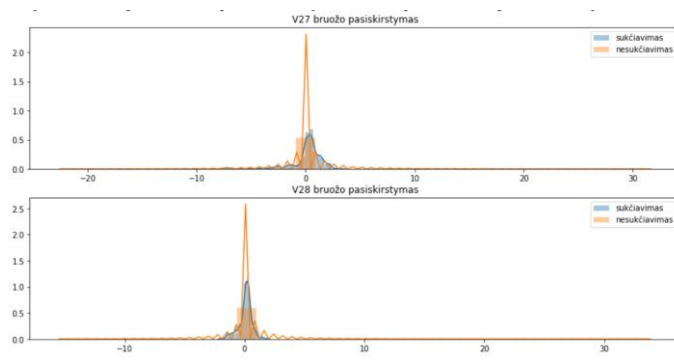
PRIEDAI

1 PRIEDAS. Bruožų pasiskirstymas









2 PRIEDAS. GMM modelio „Python“ kodas

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv(r"C:\Users\XXX\Desktop\CREDITCARD.csv")
import matplotlib.gridspec as gridspec
features=['V1','V2','V3','V4','V5','V6','V7','V8','V9','V10','V11','V12','V13','V14','V15','V16','V17']
nplots=np.size(features)
plt.figure(figsize=(15,4*nplots))
gs = gridspec.GridSpec(nplots,1)
for i, feat in enumerate(features):
    ax = plt.subplot(gs[i])
    sns.distplot(df[feat][df.Class==1], bins=30)
    sns.distplot(df[feat][df.Class==0],bins=30)
    ax.legend(['sukčiavimas', 'nesukčiavimas'],loc='best')
    ax.set_xlabel('')
    ax.set_title(feat + ' bruožo pasiskirstymas')

from sklearn.model_selection import train_test_split
from matplotlib.colors import LogNorm
from sklearn import mixture

df_0=df[df.Class==0] #Dataset with non-fraudulent only
df_1=df[df.Class==1] #Dataset with fraudulent only
df_0=df_0[['V3', 'V4','Class']] #Select two most correlated features for now
df_1=df_1[['V3', 'V4', 'Class']]

X_train, X_test, y_train, y_test = train_test_split(df_0.drop(['Class'],axis=1), df_0['Class'], test_size=0.2, random_state=42)
X_train_1, X_test_1, y_train_1, y_test_1 = train_test_split(df_1.drop(['Class'],axis=1), df_1['Class'], test_size=0.2, random_state=42)

# Fit a Gaussian Mixture Model with the two components 'V14' and 'V17'
clf = mixture.GaussianMixture()
clf.fit(X_train)

from sklearn.metrics import classification_report
from sklearn.metrics import average_precision_score
from sklearn.metrics import precision_recall_curve
T=-40
y_test_proba[y_test_proba>=T]=0
y_test_proba[y_test_proba<T]=1

y_test_orig=np.hstack([y_test, y_test_1])
print('Classification report')
print(classification_report(y_test_orig, y_test_proba))
print('Test AUCPR = ' + str(average_precision_score(y_test_orig, y_test_proba)))

precision, recall, _ = precision_recall_curve(y_test_orig, y_test_proba)
plt.step(recall, precision, color='b', alpha=0.2, where='post')
plt.fill_between(recall, precision, step='post', alpha=0.2, color='b')
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.ylim([0.0, 1.05])
plt.xlim([0.0, 1.0])
plt.title('2-class Precision-Recall curve: AUC={0:0.2f}'.format( average_precision_score(y_test_orig, y_test_proba)))
```

```

import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test_orig, y_test_proba)

ax = plt.subplot()
sns.heatmap(cm, annot=True, ax = ax);

ax.set_xlabel('Prognozuojama reikšmė');ax.set_ylabel('Teisinga reikšmė');
ax.set_title('Klaidų matrica');
ax.xaxis.set_ticklabels(['Nesukčiavimas', 'Sukčiavimas']); ax.yaxis.set_ticklabels(['Nesukčiavimas', 'Sukčiavimas']);

```

3 PRIEDAS. K-artimiausio kaimyno modelio „Python“ kodas

```
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import normalize
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
import time

tic=time.time()
full_data=pd.read_csv(r"C:\Users\XXX\desktop\creditcard.csv")
full_data=full_data.sample(frac=1)#randomize the whole dataset
full_features=full_data.drop(["Time","Class"],axis=1)
full_labels=pd.DataFrame(full_data[["Class"]])
full_features_array=full_features.values
full_labels_array=full_labels.values
train_features,test_features,train_labels,test_labels=train_test_split(full_features_array,full_labels_array,train_size=0.80)
train_features=normalize(train_features)
test_features=normalize(test_features)

#k_nearest_neighbours_classification
knn=KNeighborsClassifier(n_neighbors=20,algorithm="kd_tree",n_jobs=-1)
knn.fit(train_features,train_labels.ravel())
knn_predicted_test_labels=knn.predict(test_features)

tn,fp,fn,tp=confusion_matrix(test_labels,knn_predicted_test_labels).ravel()

knn_accuracy_score=accuracy_score(test_labels,knn_predicted_test_labels)
knn_precision_score=precision_score(test_labels,knn_predicted_test_labels)
knn_recall_score=recall_score(test_labels,knn_predicted_test_labels)
knn_f1_score=f1_score(test_labels,knn_predicted_test_labels)

print("")
print("K-Nearest Neighbours")
print("Confusion Matrix")
print("tn =",tn,"fp =",fp)
print("fn =",fn,"tp =",tp)
print("Scores")
print("Accuracy -->",knn_accuracy_score)
print("Precision -->",knn_precision_score)
print("Recall -->",knn_recall_score)
print("F1 -->",knn_f1_score)

|
toc=time.time()
elapsedtime=toc-tic
print("")
print("Time Taken : "+str(elapsedtime)+"seconds")

import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(test_labels,kmeans_predicted_test_labels)

ax= plt.subplot()
sns.heatmap(cm, annot=True, ax = ax);

ax.set_xlabel('Prognozuojama reikšmė');ax.set_ylabel('Teisinga reikšmė');
ax.set_title('Klaidų matrica');
ax.xaxis.set_ticklabels(['Nesukčiavimas', 'Sukčiavimas']); ax.yaxis.set_ticklabels(['Nesukčiavimas', 'Sukčiavimas']);

import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(test_labels,knn_predicted_test_labels)
|
ax= plt.subplot()
sns.heatmap(cm, annot=True, ax = ax);

ax.set_xlabel('Prognozuojama reikšmė');ax.set_ylabel('Teisinga reikšmė');
ax.set_title('Klaidų matrica');
ax.xaxis.set_ticklabels(['Nesukčiavimas', 'Sukčiavimas']); ax.yaxis.set_ticklabels(['Nesukčiavimas', 'Sukčiavimas']);
```

4 PRIEDAS. Naive Bayers modelio „Python“ kodas

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, auc, roc_auc_score
from sklearn.metrics import recall_score, precision_score, accuracy_score, f1_score

df = pd.read_csv(r"C:\Users\XXX\desktop\creditcard.csv")
```

```
def split_data(df, drop_list):
    df = df.drop(drop_list, axis=1)
    print(df.columns)

    from sklearn.model_selection import train_test_split
    y = df['Class'].values
    X = df.drop(['Class'], axis=1).values
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
                                                        random_state=42, stratify=y)

    print("train-set size: ", len(y_train),
          "\ntest-set size: ", len(y_test))
    print("fraud cases in test-set: ", sum(y_test))
    return X_train, X_test, y_train, y_test
```

```
def get_predictions(clf, X_train, y_train, X_test):

    clf = clf

    y_pred = clf.predict(X_test)

    y_pred_prob = clf.predict_proba(X_test)
    |
    train_pred = clf.predict(X_train)
    print('train-set confusion matrix:\n', confusion_matrix(y_train, train_pred))
    return y_pred, y_pred_prob
```

```
def print_scores(y_test, y_pred, y_pred_prob):
    print('y_pred', y_pred)
    print('y_pred_prob', y_pred_prob)
    print('test-set confusion matrix:\n', confusion_matrix(y_test, y_pred))
    print("recall score: ", recall_score(y_test, y_pred))
    print("precision score: ", precision_score(y_test, y_pred))
    print("f1 score: ", f1_score(y_test, y_pred))
    print("accuracy score: ", accuracy_score(y_test, y_pred))
    print("ROC AUC: {}".format(roc_auc_score(y_test, y_pred_prob[:, 1])))
```

```
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
```

```
drop_list = []
X_train, X_test, y_train, y_test = split_data(df, drop_list)
y_pred, y_pred_prob = get_predictions(GaussianNB(), X_train, y_train, X_test)
print_scores(y_test, y_pred, y_pred_prob)
```

5 PRIEDAS. Giliojo mokymosi „Python“ kodas

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from subprocess import check_output

df = pd.read_csv("creditcard.csv")

columns = "V14 V17".split()
X = pd.DataFrame.as_matrix(df,columns=columns)
Y = df.Class
Y = Y.values.reshape(Y.shape[0],1)
X.shape
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2)

print("No of training Examples : "+str(X_train.shape[0])) # 80% data
print("No of test Examples : "+str(X_test.shape[0])) # 20% data
print("Shape of training data : "+str(X_train.shape))
print("Shape of test data : "+str(X_test.shape))
print("Shape of Y test data : "+str(Y_test.shape))

X_train_flatten = X_train.reshape(X_train.shape[0],-1).T
Y_train_flatten = Y_train.reshape(Y_train.shape[0],-1).T
X_test_flatten = X_test.reshape(X_test.shape[0],-1).T
Y_test_flatten = Y_test.reshape(Y_test.shape[0],-1).T

print("No of training Examples : "+str(X_train_flatten.shape))
print("No of test Examples : "+str(Y_train_flatten.shape))
print("No of X_test Examples : "+str(X_test_flatten.shape))
print("No of Y_test Examples : "+str(Y_test_flatten.shape))
print("No of Sanity_test : "+str(X_train_flatten[0:5,0]))

X_train_set = preprocessing.normalize(X_train_flatten)
Y_train_set = Y_train_flatten

print("No of X_train_set shape : "+str(X_train_set.shape))
print("No of Y_train_set shape : "+str(Y_train_set.shape))

def initialize_parameters(layer_dims):
    parameters = {}
    L = len(layer_dims)
    for l in range(1,L):
        parameters['W'+str(l)] = np.random.randn(layer_dims[l],layer_dims[l-1])*0.01
        parameters['b'+str(l)] = np.zeros((layer_dims[l],1))

    return parameters
```

```

# create the sigmoid function
def sigmoid(z):

    s = 1/(1+np.exp(-z))
    cache = z
    return s,cache

# create the relu function
def relu(z):

    r = np.maximum(0,z)
    cache = z
    return r,cache

# Relu Backward and Sigmoid Backward
def relu_backward(dA, cache):

    Z = cache
    dZ = np.array(dA, copy=True)
    dZ[Z <= 0] = 0
    assert (dZ.shape == Z.shape)
    return dZ

def sigmoid_backward(dA, cache):

    Z = cache
    s = 1/(1+np.exp(-Z))
    dZ = dA * s * (1-s)
    assert (dZ.shape == Z.shape)
    return dZ

# Linear_forward
def linear_forward(A, W, b):

    Z = np.dot(W,A)+b
    cache = (A, W, b)
    return Z, cache

#linear_activation_forward
def linear_activation_forward(A_prev, W, b, activation):

    if activation == "sigmoid":
        Z, linear_cache = linear_forward(A_prev,W,b)
        A, activation_cache = sigmoid(Z)

    elif activation == "relu":
        Z, linear_cache = linear_forward(A_prev,W,b)
        A, activation_cache = relu(Z)

    cache = (linear_cache, activation_cache)

    return A, cache

# L layers forward propagation
def forward_propagation(X, parameters):
    caches = []
    A = X
    L = len(parameters) // 2

```

```

    for l in range(1, L):
        A, cache = linear_activation_forward(A, parameters["W" + str(l)], parameters["b" + str(l)], activation="relu")
        caches.append(cache)

    AL, cache = linear_activation_forward(A, parameters["W" + str(L)], parameters["b" + str(L)], activation="sigmoid")
    caches.append(cache)

    return AL, caches

# Cost function
def cost_function(AL, Y):
    m = Y.shape[1]

    cost = (-1/m)*np.sum(Y*np.log(AL)+(1-Y)*np.log(1-AL))

    cost = np.squeeze(cost)
    return cost

# linear_backward
def linear_backward(dZ, cache):
    A_prev, W, b = cache
    m = A_prev.shape[1]

    dW = (1/m)*np.dot(dZ, A_prev.T)
    db = (1/m)*np.sum(dZ, axis=1, keepdims=True)
    dA_prev = np.dot(W.T, dZ)

    return dA_prev, dW, db

# linear_activation_backward
def linear_activation_backward(dA, cache, activation):
    linear_cache, activation_cache = cache

    if activation == "relu":
        dZ = relu_backward(dA, activation_cache)
        dA_prev, dW, db = linear_backward(dZ, linear_cache)

    elif activation == "sigmoid":
        dZ = sigmoid_backward(dA, activation_cache)
        dA_prev, dW, db = linear_backward(dZ, linear_cache)

    return dA_prev, dW, db

# backward propagation
def backward_propagation(AL, Y, caches):
    grads = {}
    L = len(caches) # the number of layers
    Y = Y.reshape(AL.shape)

    # Initializing the backpropagation
    dAL = - (np.divide(Y, AL) - np.divide(1 - Y, 1 - AL))

```

```

current_cache = caches[L-1]
grads["dA" + str(L)], grads["dW" + str(L)], grads["db" + str(L)] = linear_activation_backward(dAL,current_cache,activation="sigmoid")

for l in reversed(range(L-1)):
    # lth layer: (RELU -> LINEAR) gradients.
    # Inputs: "grads["dA" + str(l + 2)], caches". Outputs: "grads["dA" + str(l + 1)], grads["dW" + str(l + 1)], grads["db" + str(l + 1)]
    current_cache = caches[l]
    dA_prev_temp, dW_temp, db_temp = linear_activation_backward(grads["dA"+str(l+2)],current_cache,activation="relu")
    grads["dA" + str(l + 1)] = dA_prev_temp
    grads["dW" + str(l + 1)] = dW_temp
    grads["db" + str(l + 1)] = db_temp

return grads

# update parameters
def update_parameters(parameters, grads, learning_rate):

    L = len(parameters) // 2 # number of layers in the neural network
    for l in range(1,L+1):
        parameters["W"+str(l)]=parameters["W" + str(l)]-learning_rate*grads["dW" + str(l)]
        parameters["b"+str(l)]=parameters["b" + str(l)]-learning_rate*grads["db" + str(l)]
    return parameters

# setting the size of the network
layer_dims = [2,10,6,4,2,1]

# Deep Learning network to classify frauds and normal
def nn_model(X,Y,layer_dims,learning_rate, num_iterations,print_cost):
    costs = []

    parameters = initialize_parameters(layer_dims)

    for i in range(0,num_iterations):
        #forward propagation
        AL, caches = forward_propagation(X, parameters)

        #compute cost
        cost = cost_function(AL, Y)

        #backward propagation
        grads = backward_propagation(AL, Y, caches)

        #update parameters
        parameters = update_parameters(parameters,grads,learning_rate)

        if print_cost and i % 100 == 0:
            print ("Cost after iteration %i: %f" %(i, cost))
        if print_cost and i % 100 == 0:
            costs.append(cost)

    plt.plot(np.squeeze(costs))
    plt.ylabel('cost')
    plt.xlabel('iterations (per tens)')
    plt.title("Learning rate =" + str(learning_rate))
    plt.show()

    return parameters

X_train_set.shape
Y_train_set.shape

# running a model
parameters = nn_model(X_train_set,Y_train_set,layer_dims,learning_rate=.001,num_iterations = 300, print_cost = True) # su tais 94% duomenų

# predict Function
def predict(X, y, parameters):

    m = X.shape[1]
    p = np.zeros((1,m))

    # Forward propagation
    probas, caches = forward_propagation(X, parameters)

    for i in range(0, probas.shape[1]):
        if probas[0,i] > 0.5:
            p[0,i] = 1
        else:
            p[0,i] = 0

    print("predictions: " + str(p))
    print("true labels: " + str(y))
    print("Accuracy: " + str(np.sum((p == y)/m)))

    return p

print("Predicting data from mixed data...")
pred_test = predict(X_test_flatten, Y_test_flatten, parameters)

```

6 PRIEDAS. Straipsnis - Kreditinių kortelių sukčiavimo atpažinimo tymas

KREDITINIŲ KORTELIŲ SUKČIAVIMO ATPAŽINIMO TYRIMAS

Greta PRATUZAITĖ¹, Nijolė MAKNIČKIENĖ²

¹ Vilniaus Gedimino technikos universitetas, verslo vadybos fakultetas, finansų inžinerijos katedra, Saulėtekio al. 11, LT-10223 Vilnius, Lietuva

² Vilniaus Gedimino technikos universitetas, verslo vadybos fakultetas, finansų inžinerijos katedra, Saulėtekio al. 11, LT-10223 Vilnius, Lietuva

2018 m. gruodžio 10d.

Santrauka. Įvairios institucijos susiduria su iššūkiu prisitaikyti prie naujų technologijų, įtraukti dirbtinį intelektą į savo vykdomą veiklą ir reaguoti į vykstančius pokyčius. Dirbtinis intelektas gali ženkliai palengvinti įvairių anomalijų, tokių kaip kreditinių kortelių mokėjimų sukčiavimas, stebėjimą bei rasti būdus kaip galima būtų išvengti vyraujančių sukčiavimų. Šio tyrimo tikslas yra aprašyti dirbtinio intelekto sąvoką finansų sektoriuje, naudojamus algoritmus bei iširti duomenų bazę, kuri rasta Kaggle duomenų bazėje. Tyrimo metu buvo aprašyta, kas yra dirbtiniai neuroniniai tinklai, keli dažnai naudojami algoritmai finansų sektoriuje, pavyzdžiui, vektorių palaikymo mašinos, K-artimiausio kaimyno modeliai. Taip pat buvo pristatyta ir išanalizuota kreditinių kortelių transakcijų duomenų bazės duomenys, naudojami apmokant dirbtinį neuroninį tinklą. Analizuojant duomenų bazę, buvo naudota Python programavimo kalba, kad būtų braižomi grafikai. Tyrimo metu buvo susipažinta su dirbtiniais neuroniniais tinklais, buvo išsiaiškinta, kad tirama duomenų bazė yra sudaryta iš realių transakcijų ir dėl konfidencialumo buvo transformuoti duomenys, tačiau duomenų bazė tinkama, norint apmokyti neuroninį tinklą.

Reikšminiai žodžiai: kreditinės kortelės, bankai, mokėjimai, sukčiavimai, dirbtinis intelektas.

Įvadas

Turbūt kiekvienas pritartų, kad dabar inovacijos ir technologijos yra stipriai augančios ir besikeičiančios. Bet kad jų sėkmė būtų pasiekta, ištis svarbus yra pasitikėjimas jomis. Žinomos kompanijos, tokios kaip „Apple“, „Amazon“, „Netflix“ stengiasi protingai ir apdairiai rizikuoti su naujomis technologijomis, darbuotojams leidžia ieškoti išmanių ir kitokių idėjų, bet taip pat jos neužmiršta ir pasitikėjimo, kurį turi kelti klientui, svarbos (Botsman, 2017). Įvairios institucijos susiduria su iššūkiu prisitaikyti prie naujų technologijų, įtraukti dirbtinį intelektą į savo vykdomą veiklą ir reaguoti į vykstančius pokyčius. Pasaulis tampa vis labiau automatizuotas ir vis daugėja užduočių, priklausančių nuo robotų ir juos valdančių algoritmų. Žmonės ar pačios įmonės siekia sukurti programas, kurios veiktų lyg tikros žmogaus smegenys bei galėtų atlikti žmonėms prilygstančias užduotis.

Kiekviena įmonė stengiasi sukurti kažką geresnio bei efektyvesnio, o nuolatinis tobulėjimas skatina pokyčius pasaulyje. Atkreipus dėmesį į tai kas vyksta pasaulyje technologijose, dirbtinis intelektas jau mažai ką bestebina. Tačiau finansų sektoriuje, konkrečiau bankuose, dirbtinis intelektas plinta, bet ne taip stipriai kaip norėtųsi. Priešasčių šiam reiškiniui gali būti daug, tiek senesnės sistemos, dideli duomenų srautai, atsakomybės, apribojimai, darbuotojų kaita ir baimės, kad bus prarastas klientų pasitikėjimas. Jau kuris laikas ypač dideli nuostoliai yra susiję su kreditinių kortelių sukčiavimu. Bankai siekia atpažinti nesąžiningus mokėjimo kortelių sandorius, kad klientai nebūtų apmokestinami pirkiniais, kurių, pavyzdžiui, jie nepirko, kad nebūtų pasinaudota jų duomenimis ir vykdoma nesąžininga veikla. Ši problema ištis svarbi ir yra naudojama daug dirbtinio intelekto algoritmų, kurie mokosi ant mokėjimo kortelių duomenų bazių. Šiame straipsnyje bus nagrinėjamas dirbtinis intelektas ir jo panaudojimas bankuose, pristatyti dažniausiai naudojami algoritmai bei kreditinių kortelių sukčiavimo duomenų bazė iš Kaggle duomenų bazės.

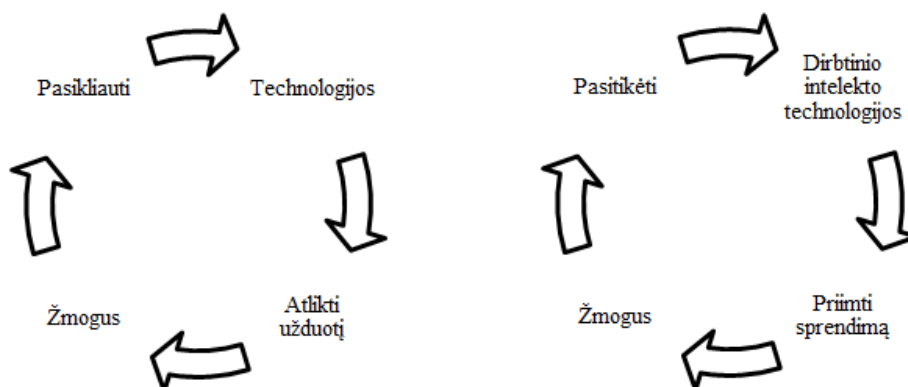
Dirbtinis intelektas finansiniame sektoriuje

Dirbtinis intelektas (angl. Artificial Intelligence), tai tam tikra mokslo šaka, kai yra kuriamos nuovokios programos, analizuojami taip vadinami „protingi agentai“ (angl. Intelligent Agents). Protingi agentai, tai tokios mašinos, kurios mąsto ir bando padidinti savo šansus įveikti tam paskirtas užduotis. Tokios mašinos siekia atkartoti žmogaus smegenų veikimą, bando atlikti veiksmus, kuriuos geba atlikti pats žmogus, pavyzdžiui, atskirti objektus ar žmogaus jausmus. Dirbtinį intelektą galima būtų įvardyti ir kaip kompiuterinių sistemų plėtrą, kai yra atliekamos užduotys, susijusios

¹ Greta Pratuzaite. El. paštas greta.pratuzaite@stud.vgtu.lt

² Nijolė Maknickienė. El. paštas nijole.maknickiene@vgtu.lt

su žmogaus intelektu, pavyzdžiui, supratimas kalbos ar automobilio vairavimas (Botsman, 2018). Šis terminas gali dažnai klaidinti, bet jokių būdu tai nėra žmogaus protas, jis tik imituojamas ir tik intelektualiai rezultatai yra taikomi dirbtiniam intelektui konkrečiose situacijose. Ateityje dirbtinio intelekto staigus šuolis pasitikėjime turėtų tik augti, tai byloja paveikslas, kuris pateiktas žemiau (1 paveikslas). Dabar žmogus vis dažniau pasikliauna technologijomis, kurios atlieka tam tikras užduotis, o ateityje šis pasitikėjimas tik augs, tačiau vis labiau bus įtraukiamas dirbtinis intelektas, kuris pats priims sprendimą, pateiks reikiamą informaciją žmogui ir taps vis protingesniu.



1 paveikslas. Pasitikėjimas dirbtiniu intelektu dabar ir ateityje (Šaltinis: Botsman 2017)

1.1. Dirbtinio intelekto panaudojimas ir svarba finansų sektoriuje

Finansų sektoriuje dirbtinis intelektas gali būti panaudojamas šiose srityse: buhalterija, auditas, reklama, valiutų keitimas, investicijos, klientų aptarnavimas, korupcija, kredito analizė, pinigų plovimas, sukčiavimo aptikimas ir pan. Bet gana dažnai tai yra susiję su pinigų apsauga, siekiant apsisaugoti nuo sukčiavimo. Būtent tokios kompiuterinės sistemos veikia taip, kad yra stebimos kliento išlaidos ir gaunami signalai, jei tik įvyksta kokie nors netikėti ir įtarimus keliantis mokėjimai. Pasak Oksfordo universiteto profesorius D. Kroening, bankai patiria didelį spaudimą iš aplinkos, įmonių, kurios neturi senos programinės įrangos ir tiek duomenų kaip bankai (Allan, 2018). Bankuose naudojami ir automatizuoti asistentai klientams, kurie teikia greitą pagalbą išspręsti savo klausimus per internetinę pranešimų sistemą, naudodami savo asmeninius kompiuterius ar išmaniuosius telefonus. Naudodamiesi mašininio mokymosi technologijomis, pokalbių svetainės nuolat tobulėja, atsižvelgiant į jų sugebėjimą tiksliai nustatyti kliento problemą ir atsakyti į klausimus. Todėl šiandien jie geba atpažinti daugybę bendrinių klausimų, kurių gali paklausti klientas.

Įvairūs bankai pasaulyje jau naudoja arba ruošiasi ateityje plėsti dirbtinio intelekto technologijas. Štai keletas pavyzdžių:

- „JPMorgan Chase“ bankas naudoja programą pavadintą „LOXM“, kuri efektyviau ir greičiau vykdo klientų pavedimus bei „CoiN“ platformą, kuri analizuoja teisinius dokumentus ir sugeneruoja kredito sutartis ženkliai greičiau - per kelias sekundes;
- Vokietijos skaitmeninis bankas „Number26“ naudoja programą pavadintą „Pulse26“, kuri analizuoja klientų istoriją ir prognozuoja jų galimus veiksmus;
- Britanijos bankas „Atom Bank“ naudoja „Virtual Agent“, bendraujančią su klientais ir atsakančią į jų užduodamus klausimus iki tam tikro lygio;
- Amerikos bankas taip pat naudoja asistentą pavadinimu „Erica“, kuris atlieka kasdienes sandorius, susijusias su finansiniais poreikiais. Jis naudoja dvi formas dirbtinio intelekto - natūralios kalbos apdorojimas, kad būtų suprantama kalba bei mašininį mokymąsi, kad būtų įmanoma surinkti įžvalgas pagal gautus kliento duomenis, kuriuos galima paversti naudingais patarimais (Crosman, 2018);
- Skandinavijos bankai irgi naudoja dirbtinį intelektą, kad būtų sumažintas krūvis darbuotojams. Dirbtiniai intelektai, kurių pavidalas yra kaip asistentų: „Aida“ - SEB banke, „Nina“ - Swedbank 'e, „Nova“ - Nordea banke. Visi jie yra pavadinti moterų vardais iš dalies gal todėl, nes klientai jaučiasi patogiau, kai bendrauja su moterimis.

Pagal 2017 metais atliktą „PwC“ kompanijos „Global Digital IQ Survey“ apklausą, net 54% organizacijų investuoja į dirbtinį intelektą ir toks skaičius taip išaugo per tris metus net iki 63%. Tačiau bankai susiduria išties su dideliais iššūkiais, tai ir senesnės sistemos, kurias reikia pritaikyti, didžiulės duomenų bazės, klientų jautrumas, procesų patvirtinimas ir pasitikėjimas bei darbuotojų pasikeitimai, kurie gali sutrikdyti darbų eigą.

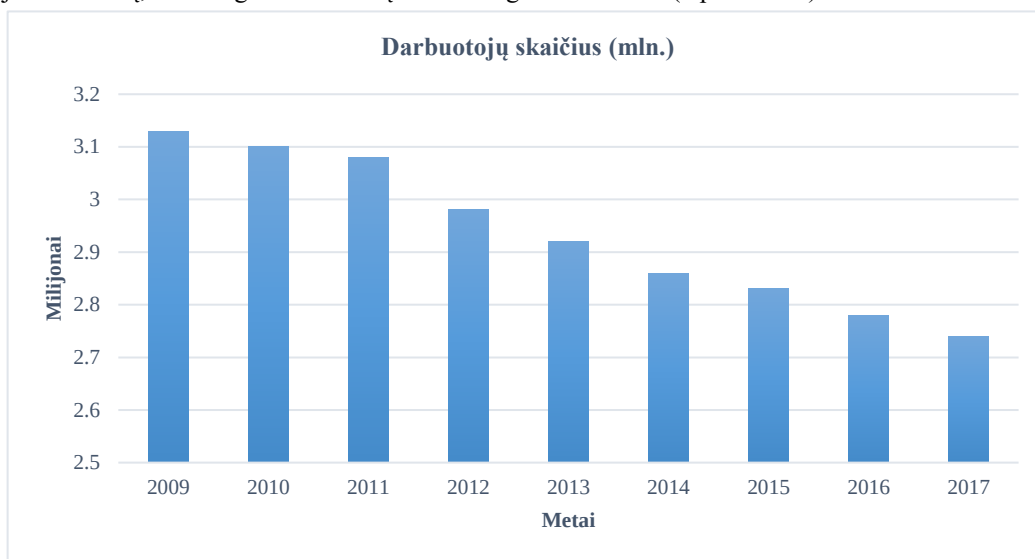
Taip pat „Accenture“ kompanija pateikė ataskaitą apie bankines technologijų vizijas 2018, kur apskaičiavo, kad tokios naujosios technologijos kaip dirbtinis intelektas, turi būti naudojamos tiksliai, išauga būtinybė įgyti žmogiškųjų žinių, kad tikslai būtų įgyvendinami, todėl potencialiai gali augti organizacijų užimtumo lygis apie 10

procentų nuo 2018 iki 2022 metų. Be to net 77 procentai bankų planuoja naudoti dirbtinį intelektą savo veikloje, kad per ateinančius metus automatiškai atliktų reikiamas užduotis įvairiausiais mastų dydžiais.

Daug perspektyvų, susijusių su dirbtiniu intelektu yra pastebima kovoje su pinigų plovimu, kredito sprendimais ir įvairia analize, kur dirbtinio intelekto naudojimas būtų paprasčiausias, tačiau nėra plačiai naudojamas. Dažniausiai yra naudojami „Chat Box“, kur įvairūs asistentai komunikuoja su klientais iki tam tikro lygio.

Dirbtinio intelekto pagalba gali būti apdorojami labai dideli kiekiai duomenų, mažinamas žmogiškųjų klaidų faktorius. Taip pat vyrauja efektyvesnis išlaidų panaudojimas, suprantami realūs klientų poreikiai pagal įvairiausius kriterijus, šablonus, kuriuos arba sunkiau pastebėti žmonėms, arba iš vis yra nepastebima. Tačiau neišvengiama ir rizikos bei pavoju, kuriuos kelia ši technologija. Tikrai stinga pilno pasitikėjimo, kuris jau buvo minėtas kiek anksčiau, pasitaiko neteisingo vystymo ir tikrai reikia daug duomenų, kad būtų galima iš ko mokytis. Susiduriama ir su teisinių aktų trūkumu bei klaidų atsakomybė kelia daug dilemų, kas kaltas, kai įvyksta didelė žala.

Įsitraukimas į banko sektorių gali atnešti ne tik džiugesį, bet ir neišvengiamą rūpestį, kad bus mažinamas darbuotojų skaičius (Marria, 2018). Pavyzdžiui, įvykusi skaitmenizavimą iš tiesų sumažino personalo poreikį padaliniuose, vis daugiau jų buvo uždarama. Remiantis Europos banko federacijos duomenimis 2017 metų pabaigoje banke dirbo apie 2,7 milijonai žmonių, maždaug 40 tūkstančių mažiau negu 2016 metais (2 paveikslas).



2 paveikslas. Darbuotojų skaičių pokyčiai Europoje (Šaltinis: sudaryta darbo autorių, remiantis European Banking Federation 2017)

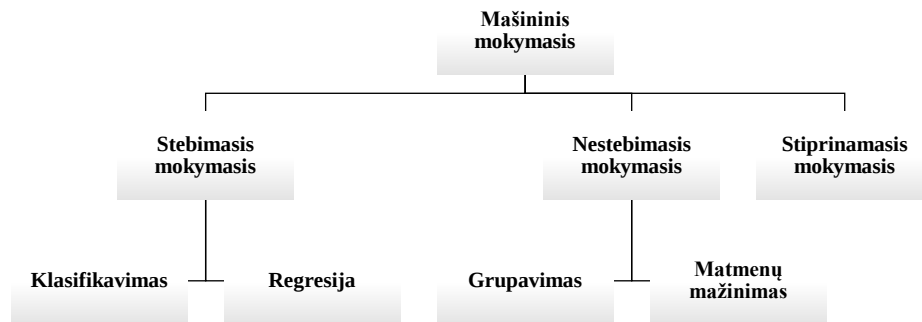
Dirbtinis intelektas ypač yra naudojamas kaip pokalbių dėžutės bei virtualūs asistentai. Jie keičia kai kuriuos darbus ir kuo daugiau jais bus naudojama, tuo daugiau jie patys apsimokys ir gerins savo gebėjimus pagelbėti klientams (Marria, 2018). Taip pat dirbtinis intelektas naudojamas atpažinti sukčiavimus bei pinigų plovimui. Būtent tokios įdiegtos sistemos bankuose atnešė daug sėkmės ir sutrumpina darbuotojų laiką bei padidina efektyvumą. Kibernetinės grėsmės yra viena iš svarbesnių rizikų, priklausančių finansinių paslaugų grupei, o pagal įvairius tyrimus kibernetinių duomenų ir duomenų saugumo didinimas - tai pirmasis pasaulinių bankų prioritetas. Dirbtinis intelektas ir pažangios duomenų analizės bus pagrindinis užkertant kelią kibernetinėms atakoms.

Remiantis „ThreatMetrix“ duomenimis, skaitmeniniai sandoriai Europoje 2018 metų pradžioje įvyko 30% daugiau elektroninių išpuolių nei praėjusių metų pirmąjį ketvirtį. Tad be viso saugumo valdymo, siekiant išvengti finansų sistemos pažeidžiamumo, svarbu mokytis ir itin gerai suprasti technologinius sprendimus. Taip pat kompanija „BBVA“ nurodė, kad 69 procentai Europos kompanijų stinga pagrindinių žinių apie įtaką kibernetinių atakų, nepaisant to, kad 80 procentai jų turėjo tam tikrą kibernetinį incidentą per pastaruosius metus.

1.2. Neuroniniai tinklai

Gana dažnai dirbtinio intelekto nagrinėjimas prasideda nuo sąvokos mašininis mokymasis (angl. Machine Learning). Tai yra mokslo šaka, kuri dirbtinio intelekto pagalba kuria programas, kurios pačios gerina savo rezultatus, pačios apsimoko, kai yra naudojama vis daugiau duomenų. Gana greitai tai tapo kaip dažniausiai naudojamas dirbtinio intelekto mokymosi būdas (Chollet, 2017). Mašininis mokymasis gali būti skirstomas įvairiais aspektais, bet pagal

įvairius šaltinius galima rasti, kad jis yra skirstomas į stebimąjį, nestebimąjį bei stiprinamąjį mokymąsi (Jha, 2017; Hexagon, 2018).



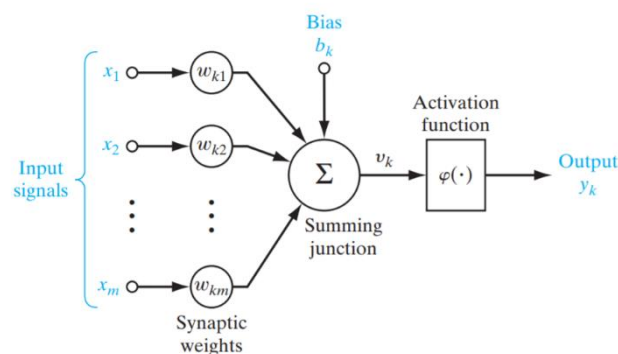
3 paveikslas. Mašininio mokymosi struktūra (Šaltinis: sudaryta darbo autorių, remiantis Jha 2017)

Stebimasis mokymasis išsiskiria tuo, kad tam tikri duomenys yra pažymėti ir iš to yra ieškoma taisyklių - būdų kaip turėtų būti prognozuojami nauji duomenys. Tad pirmas žingsnis iš turimų duomenų suformuoti modelį jau su naujais duomenimis ir iš kurių po to yra duotas naujas rezultatas. Stebimasis mokymasis yra išskaidomas į šias kategorijas - klasifikaciją arba regresiją. Klasifikacijos tikslas yra naujus stebėjimus priskirti tam tikrai kategorijai, o regresijos - prognozuoti duomenis.

Nestebimasis mokymasis neturi duomenų pažymėjimo ir būtent dėl tos priežasties yra galimos tik tokios funkcijos: grupavimas (angl. Clustering) ir matmenų mažinimas. Grupavimas gali būti naudingas, kai yra grupuojami tam tikri objektai į sluoksnius (klasterius), pavyzdžiui, klientai skirstomi pagal savo galimybes ar įpročius. Taisyklių ieškojimas gali turėti naudos, kai, pavyzdžiui, klientai įsigyja vieną prekę ar paslaugą, bet atitinkamai tuo pačiu įsigyja ir dar kitą. Taip pat egzistuoja ir kitas mokymosi metodas - stiprinamasis (angl. Reinforcement learning), kuris dažniau yra naudojamas akademiniams tikslams ar kuriant žaidimus.

Dirbtiniai neuroniniai tinklai išties primena žmogaus smegenis, nes informacija yra gaunama per mokymosi procesą ir patys neuronai geba prisiminti pateiktą informaciją ir ją kaupti (Haykin, 2009). Galima sakyti, kad paties roboto smegenys yra sudarytos iš programinės įrangos tinklo, kurio tikslas vis atkartoti žmogaus smegenyse esančias neuronų jungtis.

Neuroniniai tinklai yra sudaryti iš neuronų, kurie vienas su kitu yra sujungti ir pagal jungtis geba perduoda signalus. Veikimas susideda iš to, kad tinklas priima įvesties signalą ir kelis kartus apdoroja, kas suteikia gilesnį ir naudingesnį mokymąsi. Šių metodų naudojimas reiškia, kad algoritmas gali pagerinti užduoties atlikimą vis kartojant naudojamus duomenis. 4 paveikslas vaizduoja neurono modelį, iš kurio galima sudaryti dar didesnę neuroninį tinklą.



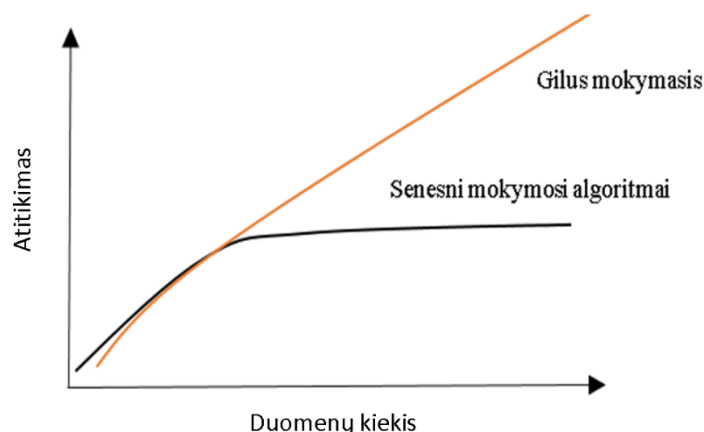
4 paveikslas. Neurono modelis (Haykin 2009)

Kiekviena sinapsė turi savo svorį, žymimą w_{kj} . Signalas x_j prie sinapsės į įvesties yra prijungtas prie neurono k , kai yra padauginama iš sinapsės svorio w_{kj} (žr.(1)). Galiausiai po to visi įvesties signalai yra sudedami ir pasiekia aktyvavimo funkciją, skirta apriboti neurono išėjimo amplitudę ir įgyvendinti išėjimą, žymimą y_k (Haykin, 2009).

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (1)$$

Gilus mokymasis (angl. Deep learning), tai tam tikras mašininio mokymosi tipas (Chollet, 2017). Šis mokymasis yra vadinamas gilusis vien dėl to, kad yra sudarytas iš daugiau neuroninių sluoksnių ir kuo daugiau duomenų yra

naudojama, tuo geresni rezultatai yra pasiekiami (Brownlee, 2016). Būtent tai leido sukurti be pilotes mašinas, garsų ir vaizdų atpažinimo programas, geresnę rezultatų paiešką sistemose ir pan.



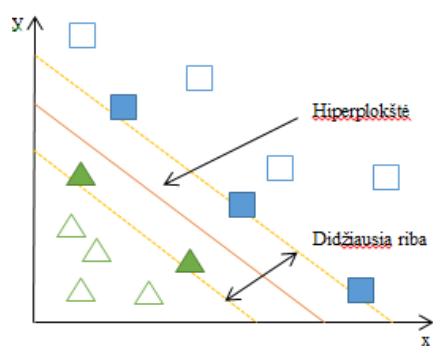
5 paveikslas. Gilaus mokymosi grafikas (Šaltinis: sudaryta darbo autorių, remiantis Brownlee 2016)

Dirbtinio neuroninio tinklo architektūra gali apibrėžti kaip veikia keli neuronai, kurie yra išdėstomi vienas kito atžvilgiu. Tokios architektūros yra struktūrizuotos kompozicijos. Tam tikros architektūros mokymas apima taikymą suderinti veiksmus siekiant sureguliuoti neuronų svorius bei slenksčius. Neuroninių tinklų architektūros pagal disponavimo funkciją, gali būti skirstomi į vieno sluoksnio, daugiasluoksnius, pasikartojančius bei apraizgytus tinklus (Silva et. al, 2017).

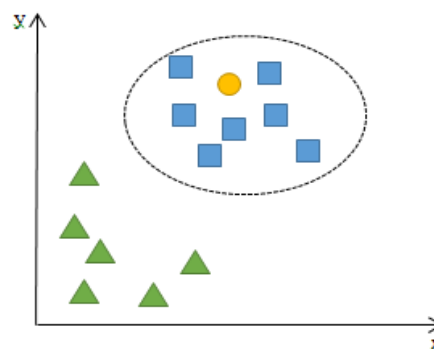
2. Mokėjimo kortelių sukčiavimo atpažinimas taikant algoritmus

Nėra vieno konkretaus algoritmo, kuris išspręstų visas problemas, todėl yra daugybę kitų. Tinkamo algoritmo pasirinkimą dažnai lemia tokie veiksniai kaip duomenų apimtis, kintamųjų tipas, tikslai, trukmės apribojimai ir struktūra, tačiau vis tiek gana sunku nusakyti, kuris algoritmas yra pats geriausias, todėl svarbu yra apsibrėžti norimą problemą. Pagal įvairius straipsnius kiek žemiau buvo pasirinkti tokie algoritmai, kurie sprendžia problemas, susijusias su bankų sektoriais, kas šiuo atveju ir yra aktualiausia.

Atraminų vektorių metodas (angl. Support vector machine) - tai algoritmas, kuris atlieka klasifikavimo užduotį ir priklauso nuo sprendimo planų, kurie atskiria duomenis į skirtingas klases (dažnai į teigiamas ir neigiamas) (Seeja et.al., 2014). Tikslas metodo, atrasti N-dimensijų erdvėje hiperplokštes (angl. Hyperplane), kurios klasifikuoja duomenis. Siekiant atskirti dvi duomenų klases, yra daugybė galimų hiperplokščių, kurias galima pasirinkti, o tikslas yra surasti tokią plokštę, kuri turėtų didžiausią ribą tarp abiejų duomenų klasių (Gandhi, 2018). Tai svarbu tam, kad būtų nustatyta tiksli duomenų klasė. Šį metodą vaizduoja 6 paveikslas.



6 paveikslas. Vektorių palaikymo mašinos modelio pavyzdys (Šaltinis: sudaryta darbo autorių, remiantis Dnl Institute 2015)

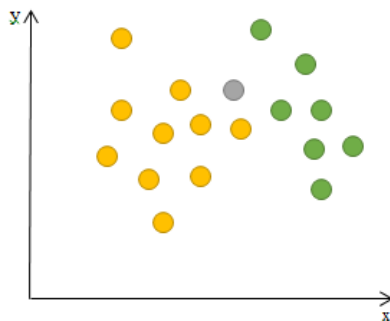


7 paveikslas. K-artimiausio kaimyno modelio pavyzdys (Šaltinis: sudaryta darbo autorių, remiantis Bernd Klein 2018)

K-artimiausio kaimyno metodas (angl. K-nearest neighbor) - tai toks algoritmas, kuris klasifikuoja pagal artimiausius k (nurodytų kaimynų skaičius) kaimynus (7 paveikslas). Tada juos suklasifikuoja į naujus, pagrįstus bei panašius

rezultatus. Kiekvienas naujas duomuo lyginamas su esamomis duomenimis naudojant atstumo metriką ir artimiausią esamą duomenį siekiama priskirti tam tikrai klasei (Seeja et al., 2014).

Naivusis Bajeso klasifikavimo metodas (angl. Naive Bayes), tai dar vienas algoritmas, kuris naudoja mokymosi duomenų rinkinį tikslinių klasių prognozavimui (Seeja et al., 2014). Šis klasifikavimo metodas yra tarsi tikimybinis metodas, naudojantis klasės informaciją iš mokymosi duomenų rinkinio ir yra skirtas numatyti būsimas klases pagal tikimybes. Pavyzdžiui, žemiau pateiktame grafike yra pavaizduoti skirtingų klasių objektai ir kai atsiranda naujas objektas be klasės, yra skaičiuojama tikimybė, kuriai klasei jis yra priskiriamas.



8 paveikslas. Naive Bayes klasifikavimo metodo pavyzdys (Šaltinis: sudaryta darbo autorių, remiantis Holczer 2018)

Paslėptas Markovo modelis (angl. Hidden Markov model) - tai toks stochastinis modelis, kur modelio objektai yra paslėpti ir modelio objektai gali grąžinti rezultatus, kurie yra stebimi ir analizuojami (Ghahramani, 2001). Šis modelis išsiskiria iš kitų, nes nesprenžia klasifikavimo problemų. Jis naudoja esamus rezultatus, gautus iš paslėptų duomenų ir naudodamas tikimybinės funkcijas nusprendžia apie objekto būseną.

2.1. Mokėjimo kortelių sukčiavimas

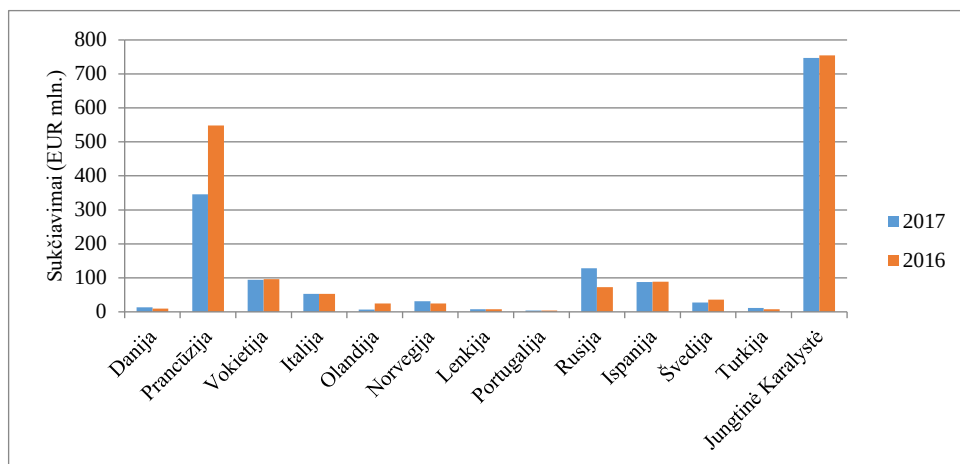
Pastaraisiais metais vis didėjanti vartotojų paklausa skatina bankus vis labiau įdiegti skaitmenizavimą. Internetas ir mobilioji bankininkystė tampa įprastu įrankiu (kanalu) klientams. Vis dažniau naudojant šiuos kanalus sudaroma galimybė greitų mokėjimų įvykdymui, patogesniai ir greitesniai įvairių sprendimų ir galimybių naudojimui, tačiau tai didina galimybę nusikaltėliams pasinaudoti klientu ir jo duomenimis.

Kredito kortelių naudojimas yra plačiai paplitęs, o šios priemonės sukčiavimas pastaruoju laikotarpiu nuolat kyla. Finansiniai nuostoliai sukčiaujant paveikia ne tik bankus, bet ir individualius klientus. Jei bankas praranda pinigus, klientai galiausiai moka didesnes palūkanų normą ar didesnius mokesčius. Sukčiavimas taip pat gali turėti įtakos banko reputacijai, gali sukelti nefinansinius nuostolius, kuriuos kiek sunkiau kiekybiškai įvertinti per trumpą laiką, o ilgaiui gali tapti labiau pastebimi. Klientas nebegalės pasitikėti savo banku ir pasirinks kitą - patikimesnį konkurentą.

Europoje yra vykdomas prevencijos mechanizmas nuo sukčiavimo (Fico, 2017). Sukčiavimo išvengimui yra naudojamas bendradarbiavimas su kitais pramonės atstovais ir jų duomenimis, taip pat naudojamas mašininis mokymasis ir dirbtinis intelektas. Įvairūs algoritmai skaičiuoja galimus kliento veiksmus pagal jo vartojimą, stengiamasi atpažinti tokias situacijas, kurios yra nebūdingos tam klientui ir taip atskleisti, kad tai yra sukčiavimas. Tokių technologijų įsitraukimas į šią sritį turi didžiulės naudos, nes klientų duomenys yra apsaugomi, sumažinama rizika pinigų plovimui vykti bei sutaupomi kaštai darbuotojams. Šie įrankiai turi veikti įmonės lygiu, turi integruoti duomenis į tinkamą sprendimą, automatizuoti klientų autentifikavimą bei turi nebenaudoti žmogaus įsikišimo.

Tyrimus atliekanti kompanija „Fico“ pateikia statistiką, vaizduojančią pokyčius Europoje dėl mokėjimo kortelių sukčiavimo. Sumos, kiek pinigų yra prarandama, yra ganėtinai aukštos, o didžiausi praradimai egzistuoja Didžiojoje

Britanijoje, Prancūzijoje ir Rusijoje, atitinkamai 747, 346 ir 128 milijonai eurų, kas yra pateikta kiek žemiau ir pagal kitas šalis.



9 paveikslas. Sukčiavimo statistika (Šaltinis: sudaryta darbo autorių, remiantis Fico 2017)

Egzistuoja du sukčiavimo aptikimo būdai - sukčiavimo analizė (angl. Fraud analysis) ir naudotojo elgesio analizė (angl. User behaviour analysis) (Sorournejad et. al., 2016). Sukčiavimo analizė yra atliekama, kai istoriniai duomenys yra analizuojami, stebima, ar įvykę sandoriai yra įprasti ir niekuo neišsiskiriantys, ar vykdoma nesąžininga veikla. Tokie duomenų rinkiniai yra sudaromi pagal klasifikavimo modelį ir yra naudojamas dviejų klasių klasifikavimas - taisyklių indukcija, sprendimų medžiai bei neuroniniai tinklai. Būtent šios priemonės aptinka daugiausiai sukčiavimo gudrybių. Kitas metodas yra susijęs su nekontroliuojama metodika, kuri priklauso nuo naudotojo elgesio. Mokėjimo nustatymas gali būti nustatomas apgaulingai, jei prieštarauja vartotojo elgesiui ir būtent dėl to reiktų turėti teisėtą ir tikslų vartotojo elgesio modelį. Sukčiavimo analizė išsiskiria tuo, kad sukčiavimo analizės sistemos mokymasis ženkliai priklauso nuo sukčiavimo įrašų, bet ji negali aptikti naujų įrašų, o analizuojant elgesio analizę yra nukrypstama į naujų sukčiavimų atpažinimą. Tad nors elgesio analizė yra stipri, bet ji dažnai kenčia nuo melagingų signalų.

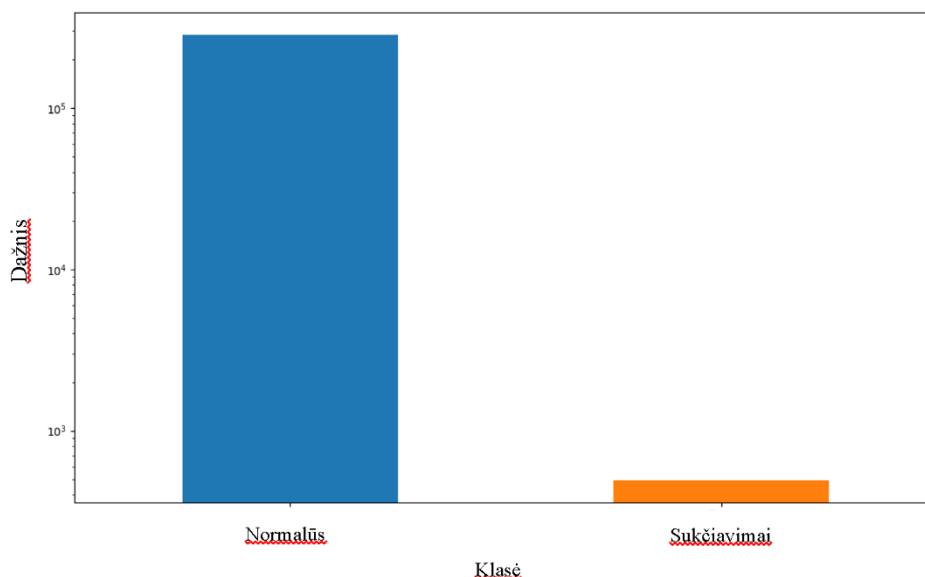
Technologijos, kurios buvo naudojamos siekiant užkirsti kelią sukčiavimui yra adreso tikrinimo sistemos (angl. Address Verification System), kortelės tikrinimo metodas (angl. Card Verification Method) ir asmens identifikavimo numeris (angl. Personal Identification Number) (Bhatla et.al., 2003). Sukčiavimas yra aptinkamas, kai yra atsižvelgiama į kredito kortelių vykdytų operacijų rinkinį, procesų identifikavimą, ar įvykusi operacija priklauso suklustotai ar tikrai sandorių klasei (Maes et. al., 2002). Pirmiausias yra svarbu patikrinti esmines sąlygas: ar tinkamas balansas ir įvertinti pagal parinktą modelį kiekvieną sandorį, ar jis yra su didele ar maža sukčiavimo rizikos tikimybe. Jei rizika yra didelė, asmenys yra informuojami perspėjimais. Tyrėjai tikrina šiuos išpėjimus ir pateikia kiekvieno perspėjimo atsakymą, ar tikras sukčiavimas, ar ne. Tada modelį reikia tobulinti ir jis gali būti pagrįstas taisyklėmis, kurioms reikia žmonių priežiūros.

Galime efektyviai naudotis mašinų mokymosi metodika nustatant apgaulės atvejus, kurie yra nustatomi pagal modelį, remiantis duomenų bazėmis. Dažniausiai modelis yra parametrų funkcija, kuri leidžia prognozuoti sandorio sukčiavimo tikimybę. Sukčiavimo aptikime mokymosi metodų naudojimas yra patogus įrankis, nes leidžia aptikti modelius didelės apimties duomenų bazėse, o kiekvieną sandorį apibūdina daugybė kintamųjų. Taip pat, nesąžiningi sandoriai dažnai koreliuoja tiek laiko, tiek erdvės atžvilgiu. Mokymosi metodai gali būti naudojami nustatant ir modeliukite egzistuojančias nesąžiningas strategijas bei nustatant naujas strategijas, susijusias su neįprastais kortelių savininkų elgesiais. Kai aptinkamas sukčiavimo metodas, nusikaltėliai pritaiko savo strategijas ar bando visai kitas. Tuo pačiu metu kasdien nauji nusikaltėliai žaidime dalyvauja ir bando naujas ir senas strategijas.

3. Mokėjimo kortelių atpažinimui naudojamų duomenų bazių tyrimas

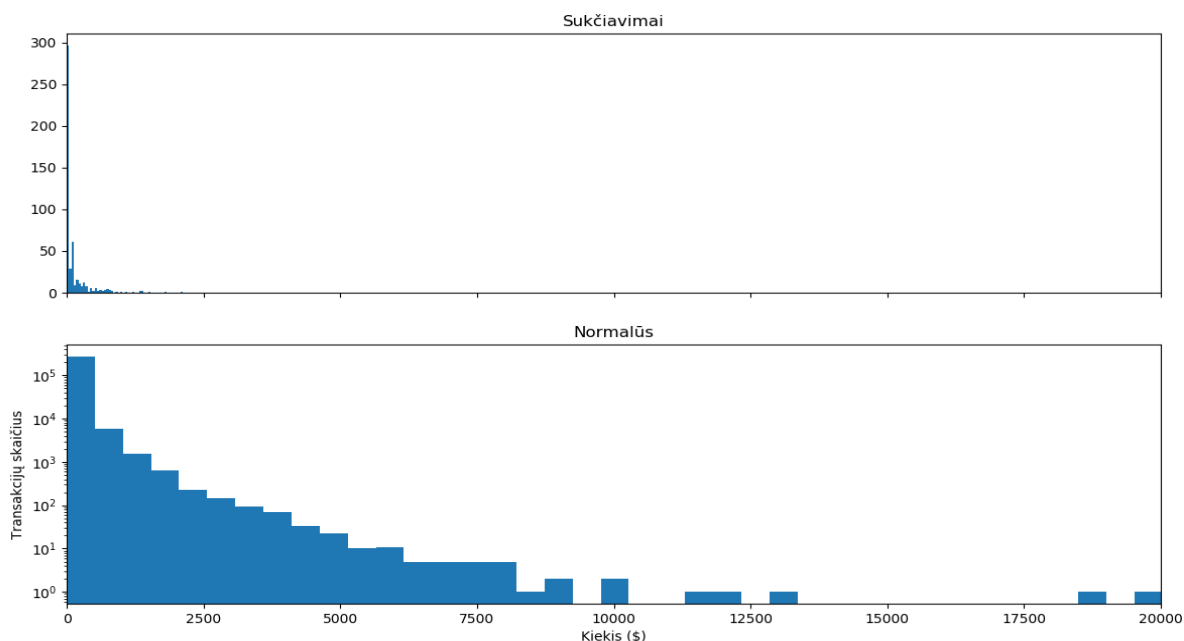
Yra naudojama daug dirbtinio intelekto algoritmų, kurie gali naudoti mokėjimo kortelių duomenų bazes. Iš Kaggle duomenų bazių sistemos, buvo pasirinkta duomenų bazė, susijusi su kreditinių kortelių sukčiavimu. Šie duomenys apie kreditinių kortelių savininkų operacijas Europoje yra 2013 metų rugsėjo mėnesio. Tačiau šis duomenų rinkinys

rodo operacijas, kurios yra įvykusios per dvi dienas bei yra įvykdyti 492 apgaulingi veiksmai iš 284 807 operacijų. Duomenų bazės analizei ir grafikų braižymui buvo naudota Python programavimo kalba.



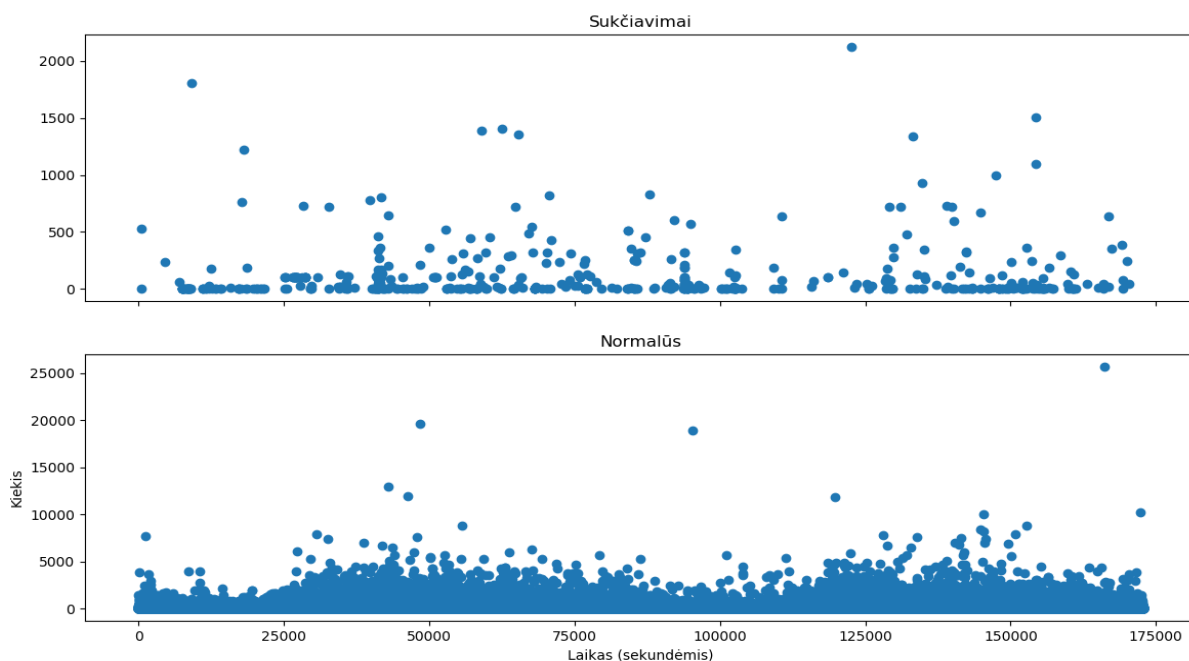
10 paveikslas. Mokėjimo kortelių paskirstymas (Šaltinis: sudaryta darbo autorių, remiantis Kaggle duomenų baze)

Remiantis 10 paveikslu, kuriame yra pavaizduotos dvi mokėjimo klasės (normalios transakcijos ir sukčiavimai), duomenų rinkinys yra nesubalansuotas, nes teigiama klasė (sukčiavimai) sudaro 0,172% visų sandorių. Dėl konfidencialumo negalima pateikti pagrindinių funkcijų ir kitos papildomos informacijos apie įvykusius sandorius, todėl duomenys buvo transformuoti pagal pagrindinę komponentų analizę (angl. Principal component analysis) ir būtent dėl šios priežasties duomenys turi tik skaitinę vertę. Ši duomenų bazė susideda iš V1-V28 kintamųjų, kurie buvo gauti naudojant pagrindinę komponentų analizės funkciją. 11 paveikslas vaizduoja transakcijų pasiskirstymą pagal sukčiavimus ir ne sukčiavimus, kas akivaizdžiai byloja, sukčiavimų skaičius yra ženkliai mažesnis.



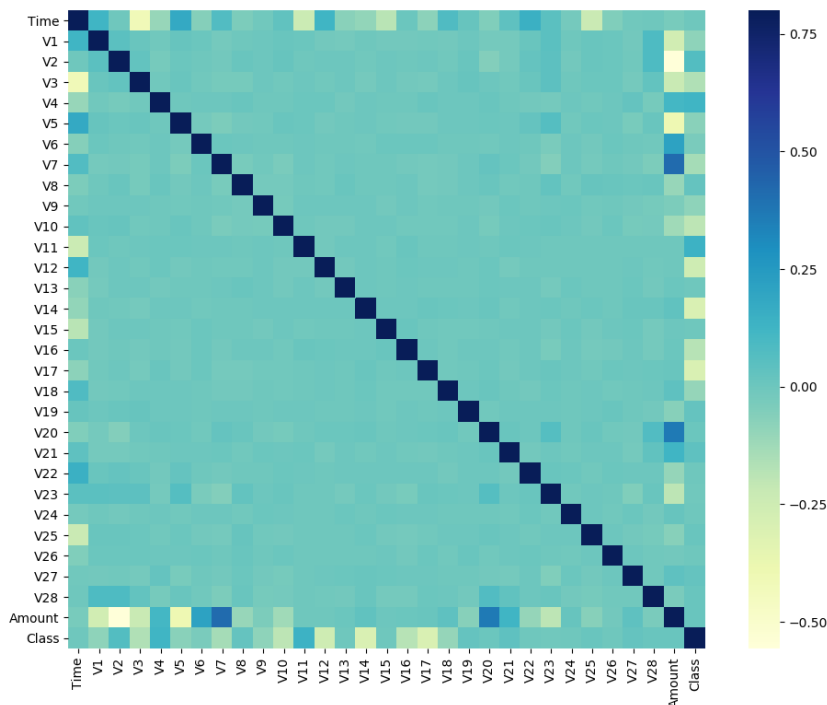
11 paveikslas. Transakcijų skaičius pagal klases (Šaltinis: sudaryta darbo autorių, remiantis Kaggle duomenų baze)

Taip pat šiame duomenų rinkinyje yra nurodyta sandorio suma bei klasė, kuri vaizduoja, ar sandoris buvo sukčiavimas ar ne. Jei sukčiavimas, tada transakcija yra žymima -1, jei ne - 0. Taip pat svarbus vienas komponentas tai laikas, kuris šioje duomenų bazėje yra atvaizduojamas sekundžių tikslumu po kiekvieno įvykusio mokėjimo (12 paveikslas).



12 paveikslas. Mokėjimo judėjimai laiko atžvilgiu pagal klases (Šaltinis: sudaryta darbo autorių, remiantis Kaggle duomenų baze)

Žemiau pavaizduota koreliacijos matrica rodo, kad nė vienas iš V1-V28 komponentų neturi tarpusavio koreliacijos. Tačiau, jei atkreipsime dėmesį į klasę, ji turi tam tikrą formą teigiamą ir neigiamą koreliaciją su visais V komponentais, bet neturi ryšio su laiku ir suma.



13 paveikslas. Komponentų koreliacija (Šaltinis: sudaryta darbo autorių, remiantis Kaggle duomenų baze)

Duomenų nesikoreliavimas tarpusavyje reiškia, kad duomenys vienas nuo kito nepriklauso, pačios transakcijos yra unikalios ir nepriklausančios viena nuo kitos. Norint apmokyti neuroninius tinklus šis duomenų rinkinys yra kiek per mažas, nes sukčiavimo pavyzdžių nėra tiek daug, tačiau algoritmų analizei ši duomenų bazė yra tinkama.

Išvados

Naujausios technologijomis daro didžiulę įtaką finansų sektoriui. Ir jei pasitikėjimas senosiomis institucijomis imtų ir sumažėtų, žmogui rasti kuom pasitikėti ištis pasidaro ganėtinai sunku. Bet šios institucijos nebūtinai turi išnykti, jos gali ir turi prisitaikyti prie pasaulyje vykstančių technologinių pokyčių. Pasaulis tampa vis labiau automatizuotas, bet klausimas kada gali įvykti tai, kad viskas priklausys nuo robotų ir juos valdančių algoritmų. Toks dalykas sukeltų sąlygas vis mažesnėms abejonėms, bet taip pat gali tapti ir itin pavojingas. Tad iš tiesų svarbu suprasti ir nuspręsti, kur, kiek ir kada kompiuterinis kodas gali kontroliuoti situaciją.

Darbo metu buvo susipažinta su dirbtiniais neuroniniais tinklais, bei apžvelgta kokie yra dažniausiai naudojami algoritmai, norint sukurti dirbtinį neuroninį tinklą, kuris geba atpažinti tinkamas ir klaidingas transakcijas. Buvo ištirta duomenų bazė ir nubraižyti tam tikri grafikai. Matyti, kad duomenys tarpusavyje nesikoreliuoja ir jei būtų siekiama ateityje apmokyti neuroninius tinklus yra kiek per mažai pateiktą sukčiavimo pavyzdžių, tačiau algoritmų skaičiavimams ši duomenų bazė yra tinkama, nes naudojama iš tikrų duomenų ir duomenys tarpusavyje vienas nuo kito nepriklauso, kas reiškia yra daugybe unikalų mokėjimų, kur jų tinkamumas nepriklauso nuo kitų transakcijų. Kreditinių kortelių sukčiavimas yra opi problema ir būtent todėl yra įtraukiamas dirbtinis intelektas su įvairiausiais algoritmais, kad ši problema būtų išspręsta. Tačiau siekiant nustatyti nesąžiningas strategijas, vis sukuriama kiti modeliai ir kitos nusikaltėlių strategijos. Tad analizuojant įvairiausias duomenų bazes ir algoritmus yra susiduriama su iššūkiais, nes tokios duomenų bazės yra labai konfidencialios bei duomenys būna transformuojami, todėl tai analizuoti pasidaro kiek sudėtingiau.

Literatūra

- Accenture. (2018). *Building the future - ready bank. Banking technology vision 2018*. Retrieved from <https://www.accenture.com/gb-en/acnmedia/PDF-78/Accenture-Banking-Technology-Vision-2018.pdf>
- Allan, K. (2018). *How banking is adopting and using AI technology*. Retrieved from <https://www.idgconnect.com/idgconnect/analysis-review/1018981/banking-adopting-ai-technology>
- Bhatla, T. P., Prabhu, V & Dua, A. (2003). *Understand credit cards fraud*. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.431.7770&rep=rep1&type=pdf>
- Botsman, R. (2017). *Kuo galite pasitikėti? Kaip technologijos mus suartino ir kodėl gali išskirti*.
- Brownlee, J. (2016). *What is Deep Learning?* Retrieved from <https://machinelearningmastery.com/what-is-deep-learning/>
- Chollet, F. (2017). *The impossibility of intelligence explosion*. Retrieved from <https://medium.com/@francois.chollet/the-impossibility-of-intelligence-explosion-5be4a9eda6ec>
- Crosman, P. (2018). *Where Bank of America uses AI, and where its worries lie*. Retrieved from <https://www.americanbanker.com/news/where-bank-of-america-uses-artificial-intelligence-and-where-its-worries-lie>
- Dnl Institute. (2015). *Support Vector Machine: Simplified*. Retrieved from <http://dni-institute.in/blogs/support-vector-machine-simplified/>
- European Banking Federation. (2018). *Banking in Europe: EBF Facts & Figures 2018*.
- European Banking Federation. (2018). *Financial technology*. Retrieved from <https://www.ebf.eu/facts-and-figures/financial-technology/>
- Fico. (2017). *Evolution in card fraud in Europe 2017*. Retrieved from <https://www.fico.com/europeanfraud/index>
- Gandhi, R. (2018). *Support Vector Machine - Introduction to Machine Learning Algorithms*. Retrieved from https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms934a444fca47?fbclid=IwAR3JEigdujakaNpbHrffwI9pevjEYMV9PfiRsiAPx0Yf-5_wp4GqdglnRRA
- Ghahramani, Z. (2001). *An introduction to Hidden Markov Models and Bayesian Networks*. Retrieved from <http://mlg.eng.cam.ac.uk/zoubin/papers/ijprai.pdf>
- Haykin, S. (2009). *Neural networks and learning machines*.
- Hexagon, C. (2018). *How did you benefit from machine learning today?* Retrieved from <https://www.crimsonhexagon.com/blog/how-did-you-benefit-from-machine-learning-today/>
- Holczer, B. (2018). *Naive Bayes Classifier Explained Step by Step*. Retrieved from <https://www.globalsoftwaresupport.com/naive-bayes-classifier-explained-step-step/>
- Jha, V. (2017). *Machine Learning Algorithm - Backbone of emerging technologies*. Retrieved from <https://www.techleer.com/articles/203-machine-learning-algorithm-backbone-of-emerging-technologies/>
- Kaggle. (2018). *Credit Card Fraud Detection*. Retrieved from <https://www.kaggle.com/mlg-ulb/creditcardfraud>
- Klein, B. (2018). *k-Nearest-Neighbor Classifier*. Retrieved from https://www.python-course.eu/k_nearest_neighbor

[classifier.php](#)

- Maes, S, Tuyls, K, Vanschoenwinkel, B & Manderick, B. (2002). *Credit card fraud detection using bayesian and neural networks*. Retrieved from https://www.researchgate.net/profile/Karl_Tuyls/publication/248809471_Credit_Card_Fraud_Detection_Applying_Bayesian_and_Neural_networks/links/0deec52519708c5f7a000000/Credit-Card-Fraud-Detection-Applying-Bayesian-and-Neural-networks.pdf
- Marria, V. (2018). *Is Artificial Intelligence Replacing Jobs In Banking?* Retrieved from <https://www.forbes.com/sites/vishalmarria/2018/09/26/is-artificial-intelligence-replacing-jobs-in-banking/#72eebc73c55>
- Pwc. (2017). *Consumer Intelligence Series Bot.Me: A revolutionary partnership*. Retrieved from <http://pwcartificialintelligence.com/>
- Seeja, K. R. & Zareapoor, M. (2014). *FraudMiner: A Novel Credit Card Fraud Detection Model Based on Frequent Itemset Mining*. Retrieved from <https://www.hindawi.com/journals/tswj/2014/252797/>
- Silva, I. N., Hernane Spatti, D., & Flauzino, R. A. (2017). *Artificial Neural Network Architectures and Training Processes. Chapter 2*.
- Sorournejad, S., Zojaji, Z., Atani, R. E., & Monadjemi, A. H. (2016). *A Survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective*. Retrieved from https://www.researchgate.net/publication/310610856_A_Survey_of_Credit_Card_Fraud_Detection_Techniques_Data_and_Technique_Oriented_Perspective

CREDIT CARDS FRAUD IDENTIFICATION INVESTIGATION

Greta PRATUZAITĖ , Nijolė MAKNIČIENĖ

Abstract.

Different institutions face the challenge of adapting to new technologies, incorporating artificial intelligence into their own activities and monitoring the changes. Artificial intelligence can make easier monitoring of various anomalies, as well as counterfeiting credit card payments. The aim of this paper is to describe the artificial intelligence concept in the financial sector using the algorithms and to investigate the database that was found in the Kaggle database. During the work, artificial neural networks were described, and algorithms are often used in the financial sector (eg: Vector Support Machines, K-Neighbor Models). A credit card database was also introduced and analyzed, which can be used to teach an artificial neural network. When analyzing the database, Python programming language was used to draw graphs. During my work I managed to learn about artificial neural networks, it became clear that the database under study was made up of real transactions and the data was transformed due to confidentiality, but still the database is suitable for training the neural network.

Keywords: credit cards, fraud, investigation, banks, transactions, artificial intelligence.