TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Indrek Arandi 152904IABM

MASINÕPPE MEETODITEL PÕHINEVATE PANGAKAARTIDE TEHINGUTE PETTUSTE TUVASTAMISE ALGORITMIDE VÄLJA TÖÖTAMINE JA TESTIMINE

Magistritöö

Juhendaja: Innar Liiv
PhD

Tallinn 2017

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Indrek Arandi

07.05.2017

Annotatsioon

Käesoleva magistritöö teemaks on masinõppe meetoditel põhinevate pangakaartide tehingute pettuste tuvastamise algoritmide välja töötamine ja testimine LHV panga näitel. Töö eesmärgiks on luua masinõppel põhinevate meetoditega mudelid, mis suudaksid monitoorida pangakaartidega tehtavaid tehinguid, vähendada manuaalselt üle kontrollitavate tehingute mahtu ning kaarditehingutest saadavat kahju.

Probleemi lahendamiseks uuris autor erinevate pettuste osakaaludega treeningbaaside sobivust algoritmide treenimiseks, erinevate sisendparameetrite mõju algoritmidele, võrdles kuut masinõppe algoritmi, hindas kulupõhise lähenemise mõju algoritmide tulemuslikkusele ja testis algoritmide hinnangute kombineerimise mõju.

Töö tulemusena treeniti algoritmid, mis suudavad ligi 90% ulatuses vähendada pangakaartide pettustest saadavat kahju. Sobivaimaks treeningbaasiks tunnistati 20% pettuste osakaaluga baas. Sisendparameetrite gruppidest osutusid parimateks tehinguinfo, autentimisinfo, varasemate kaardi kohaloluta tehingute ja viimase 24 tunni sarnaste tehingute mahud. Masinõppe algoritmidest saavutati parimad tulemused juhusliku metsa, *boost´*i ja närvivõrkudega. Kuluefektiivse hindamismudeli paremust lõplike algoritmide testis kinnitada ei saanud. Algoritmide ennustuste kombineerimine tulemuslikkust ei tõstnud, sest enamus tuvastamata pettused kattusid eri algoritmide ennustuste lõikes.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 88 leheküljel, 5 peatükki, 15 joonist, 20 tabelit ja 5 lisa.

Abstract

Developing and testing of machine learning algorithms for detecting bank card transaction fraud

The research topic for this master`s thesis is training and testing of machine learning algorithms for detecting bank card trancation frauds on the example of LHV bank. The aim of the research is to develop models for monitoring card transactions, decreasing the amount of manual inspections of suspicious card transactions and decreasing losses from card transaction frauds by applying machine learning algorithms.

To solve the research questions, the author analysed the suitability of training sets with different fraud percentage, investigated the impact of different parameter groups, compared six machine learning algorithms, evaluated the effects of cost based evaluation methods and tested combinating different algoritms` predictions.

As a result, machine learning algorithms reducing loss from card transaction fraud by up to 90% were trained. For training the algorithms, a training set with 20% of fraud cases was recognised as the most suitable, the results were evenly good with different metrics. The most effective input parameter groups were transaction data, authentification data, historical transaction volume for card not present transactions and transaction volume of similar transactions within last 24 hours. The best results were achieved with random forest, boost and neural networks algorithms. The superiority of cost effective evaluation method over cost based model was rejected. While cost effective evaluation method achieved considerably better results in experiments of comparing training sets and parameters, the method did not show significantly better results in the final comparison of algorithms. Also, the superiority of combined predictions from different algorithms was rejected as undetected fraudulent transactions overlapped significantly for different algorithms.

The thesis is in Estonian and contains 88 pages of text, 5 chapters, 15 figures, 20 tables and 5 appendicies.

Lühendite ja mõistete sõnastik

AdaBoost	*Adaptive boosting*, masinõppe algoritm, mis kuulub *boosting*-meetodite perre
Alaesindamine	*Undersampling*, meetod, millega eemaldatakse osa andmeid ülekaalus olevast klassist
ATM	*Automated teller machine*, sularaha automaat
AUC	*Area under the ROC curve*, ROC kõvera alla jääv piirkond
CNP	*Cards not present*, kaarditehing ilma pangakaardi füüsilise kohaloluta
DT	*Decision tree*, masinõppe algoritm otsustuspuu
ECB	*European Central Bank*, Euroopa Keskpank
EMV standard	Tehniline standard nutikate pangakaartide ning kaarditerminalide ja pangaautomaatide jaoks, mis selliseid kaarte aktsepteerivad
Eksimismaatriks	*Confusion matrix*, maatriks, milles registreeritakse katselistele näidetele mingi reeglistiku rakendamisel saadavate õigete ja väärade liigitusjuhtude arv. EVS-ISO/IEC 2382-31:1999
Juhitud õpe	*Supervised learning*, masinõppe õppimisviis, kus sisendiks on märgistatud (määratud tulemiga) treeningandmed, mille põhjal mudel peab õppima tegema tulemi osas õigeid ennustusi
Juhtimata õpe	*Unsupervised learning*, masinõppe õppimisviis, kus sisendandmed ei ole tulemi osas märgistatud ja mudel peab leidma andmetest sarnaseid struktuure või looma üldistamise reegleid
Kinnitusega õpe	*Reinforcement learning*, masinõppe õppimisviis, kus algoritm õpib katse-eksituse meetodil, millised tegevused toovad suurimat kasu
LR	*Logistic regression*, logistiline regressioon
NN	*Neural networks*, masinõppe algoritm närvivõrgud
Pooleldi juhitud õpe	*Semi-supervised learning*, masinõppe õppimisviis, kus sisendandmed on segu märgistatud ja märgistamata näidetest ning mudel peab andmeid nii klassifitseerima kui tegema ennustusi
POS	*Point of Sale*, kaarditehingute müügiterminal

RF	*Random forest*, masinõppe algoritm juhuslik mets
ROC	*Receiver operating characteristic curve*, suhtelise toimimise karakteristikute kõver, mis hindab tõeselt positiivsete ja valepositiivsete suhet
SEPA	*Single Euro Payment Area*, ühtne euromaksete piirkond
SVM	*Support vector machines*, masinõppe algoritm tugivektor-masinad

Sisukord

Jooniste loetelu

Tabelite loetelu

1. Sissejuhatus

Käesoleva magistritöö teemaks on masinõppe meetoditel põhinevate pangakaartide tehingute pettuste tuvastamise algoritmide välja töötamine ja testimine. Töö on tehtud LHV panga andmete põhjal ning tegelikest vajadustest lähtuvalt. Töö tulemusi vaadeldakse aga ühest ettevõttest laiemas kontekstis.

1.1 Taust ja probleem

Kaarditehingute maht on järjepidevalt kasvanud ja paratamatult käivad sellega kaasas ka erinevad pangakaartidega seotud pettused. Kaarditehingute pettustest saadavad kahjud on märkimisväärsed nii maailma kui Eesti kontekstis. Seetõttu tuleb finantsasutustel pidevalt kaarditehinguid monitoorida ning võtta kasutusele vajalikud meetmed pettuste ennetamiseks ja saadavate kahjude minimeerimiseks.

Tavapärane praktika on kehtestada reeglitel põhinev süsteem, mis teavitab monitooringuga tegeleva üksuse töötajat tehingutest, mis ei ole tavapärased või mis on teatud tingimustel kahtlust äratavad. Selliste süsteemide kitsaskohad seisnevad selles, et need on staatilised ja tekitavad väga palju valehäireid, mille kontrollimiseks kulutatakse asjatult suurel määral ressurssi. Lisaks kõigele ei ole sellised süsteemid ka kõige suurema pettuste ärahoidmise suutlikkusega.

Seevastu masinõppe meetodid võimaldavad tuvastada käitumist, mis ei ole staatiliste reeglitega nii lihtsalt kirjeldatavad. Lisaks on masinõppe algoritme võimalik pidevalt uute andmetega treenida ja testida. Pidev ajakohastamine on sellisel süsteemil hädavajalik omadus, sest pettuste viisid on pidevas muutumises ning süsteem peab olema võimeline muutustega kohanema. Masinõppe abil on ühtlasi võimalik leida seoseid, mis võivad jääda inimesel märkamata või mille olemasolu avastataks pika hilinemisega.

Teema valiku põhjustas LHV panga kaarditehingute mahu ja nendega kaasnevate kahjude arvestatav kasv ning olemasoleva monitooringusüsteemi uuendamise vajadus. Seega sai eesmärgiks võetud masinõppe meetoditega probleemile läheneda ja välja töötada algoritmid, mis oleksid suutelised tuvastama kaarditehingute pettusi ning vähendama tekkivaid kahjusid.

1.2 Ülesande püstitus

Töö eesmärgiks on masinõppel põhinevate meetoditega luua mudelid, mis suudaksid monitoorida pangakaartidega tehtavaid tehinguid, vähendada manuaalselt üle kontrollitavate tehingute mahtu ja kaarditehingute pettustest saadavat kahju. Algoritme peaks saama regulaarselt ajakohastada ja treenida uutele andmetele vastavalt. Peamiseks mõõdikuks panga jaoks on kaarditehingute pettustega seotud kulude maht.

Probleemi lahendamiseks tuleb läbida erinevad masinõppe algoritmide treenimiseks ja testimiseks vajalikud etapid. Esmalt tuleb määratleda aluseks võetavad andmed ning erinevad parameetrid, mis on olemas või mida oleks võimalik genereerida. Seejärel on vajalik valida, treenida ja testida erinevaid masinõppe algoritme ning mõõta saavutatud tulemusi. Kuna peamiseks probleemiks on kaarditehingute pettustest saadav kahju, siis tuleb läbivalt keskenduda ka kulude vähendamise küsimusele. Lõpptulemusena peaks olema võimalik välja töötada sobivaimad mudelid ja testida neid reaalsele elule võimalikult lähedases olukorras.

Probleemi lahendamiseks püstitatud uurimisküsimused on järgmised:

- Millise pettuste osakaaluga treeningbaas on algoritmide treenimiseks sobivaim?
- Millised sisendparameetrid annavad algoritmide treenimisel kõige paremaid tulemusi?
- Milline masinõppe algoritm töötab kõige paremini kaarditehingute pettuste tuvastamisel?
- Kuidas mõjutab kulupõhine lähenemine algoritmide tulemuslikkust?
- Kas algoritmide ennustuste kombineerimise tulemusena on võimalik tulemuslikkust täiendavalt tõsta?

Erinevate andmete mahtude tohutu kiire kasvu ja probleemide keerukuse tõttu on masinõppe kasutamise populaarsus järjest kasvanud. Kaarditehingute pettuste tuvastamise kohta masinõppe abil on tehtud suhteliselt palju uurimustöid. Peamiselt on keskendutud mõnele üksikule masinõppe rakendamise alamprobleemile. Saadud tulemused on aga tugevalt sõltuvad kasutatud algandmetest, rakendatud algoritmidest, tulemuste mõõtmise meetoditest jne.

Antud töö erineb varasematest uurimustest selle poolest, et kaarditehingute monitoorimise probleemi on püütud lahendada terviklikult algusest lõpuni alates algandmetest, parameetritest ja erinevatest kliendi käitumist kirjeldavatest varasemate tehingute agregeerimise võtetest kuni algoritmideni välja. Seejuures on igas etapis rakendatud kulupõhiseid meetodeid, mis on teadusartiklites üha suuremat populaarsust kogunud, kuid mida pole rakendatud ühe töö raames läbivalt eri etappides.

Töö ühe eripärana võib välja tuua ka erinevate sisendparameetrite laia valikut kliendi demograafilistest näitajatest kuni erinevate agregeerimisteni, kokku 8 erinevat parameetrite gruppi. Eksperimendi viimases järgus on uue nüansina uuritud mudelite kombineerimisest tulenevat mõju kuluefektiivsuse kasvule.

Tulemuste lõplikul hindamisel on rakendatud treening- ja valideerimisbaasidest täiesti erinevaid andmeid, mis vastavad algoritmide reaalsele tööolukorrale. Varasemates töödes on jaotatud baasid pigem juhusliku valiku alusel treening- ja testbaasiks (nt [4], [15]), mis nagu töös välja tuleb, võib anda kallutatud tulemusi.

1.3 Metoodika

Probleemi lahendamiseks kasutatakse erinevate algbaaside koostamiseks kõigepealt alaesindamise (*undersampling*) meetodit. Tulemuste hindamisel lähtutakse lisaks tavapärastele statistilistele mõõdikutele kulupõhisest meetodist, mille eesmärgiks on maksimeerida kulusääst, mis tuleneb kaardipettuste monitoorimise kuludest ja pettuste kahjude ärahoidmisest. Kuluefektiivse meetodi puhul on kasutatud Bayes miinimum riski klassifitseerijat, mis kvantifitseerib kompromisse erinevate otsuste vahel, kasutades tõenäosusi ja kulusid, mis otsustega kaasnevad. Töös kasutatakse kuut erinevat masinõppe algoritmi – juhuslik mets, otsustuspuu, *boost*, tugivektor masinad, närvivõrgud ja logistiline regressioon.

1.4 Ülevaade tööst

Magistritöö esimeses osas antakse lühike ülevaade finantspettustest panganduses, kaarditehingutest ja kaarditehingutega seotud pettuste mahust nii Eestis kui Euroopas laiemalt. Lisaks antakse ülevaade varasematest töödest, mis on seotud masinõppe meetodite rakendamisega kaarditehingute pettuste tuvastamise valdkonnas.

Teooria osas antakse ülevaade masinõppest üldiselt ning kirjeldatakse töös kasutatud masinõppe algoritmide tööpõhimõtteid ning tugevusi ja nõrkusi. Oluline osa on ka tulemuste mõõdikute tutvustamisel ning kulupõhise hindamise meetodi selgitamisel. Teooria osa lõpetuseks tuuakse välja erinevad lahendused tüüpilistele probleemidele, mis võivad esineda kaarditehingute pettuste tuvastamisel masinõppe meetoditega.

Töö praktilises osas kirjeldatakse eksperimendi disaini alates andmete kogumisest ja valitud parameetritest kuni koostatud treening-, valideerimis- ja testbaaside ning valitud algoritmide rakendamiseni. Eksperimendi läbiviimisel testitakse kõigepealt erinevate pettuste osakaaludega treeningbaaside sobivust algoritmide treenimiseks ning valitakse edasiseks modelleerimiseks parim. Seejärel testitakse kaheksat erinevat parameetrite gruppi eraldiseisvalt ja kombineerituna ning valitakse välja sobivaimad parameetrid. Algoritmide lõplikuks hindamiseks kasutatakse eraldiseisvat testbaasi ning selgitatakse välja suurimat kulusäästu andvad mudelid. Viimases faasis testitakse parimate algoritmide kombineerimise mõju tulemustele. Magistritöö lõpetuseks tuuakse kokkuvõtlikult välja kõik töö olulisemad tulemused.

2. Finants- ja kaarditehingute pettused ning varasemad uurimused masinõppe rakendamisest antud valdkonnas

Käesolev peatükk annab ülevaate finantspettustest laiemalt ning kaarditehingute ning pettuste mahtudest. Eraldi analüüsitakse varasemaid töid antud valdkonnas ning olulisemaid tulemusi uurimisteemade kaupa.

2.1 Finantspettused panganduses

Finantspettus tähendab teenuse/kauba ja/või raha omandamist ebaeetiliste vahenditega. Petturlikke tehinguid on tihtipeale suurest tehingute hulgast raske tuvastada. Krediitkaardid on üks kõige sagedasemaid sihtmärke pettustele, kuid lisaks ka muud krediiditooted ja kaarditehingud laiemalt. Seoses tehnoloogia arenguga viimastel aastakümnetel on pettuste viisid dramaatiliselt muutunud ja arenenud. Ettevõtete ja finantsinstitutsioonide jaoks on kriitilise tähtsusega arendada erinevaid tehnoloogilisi lahendusi pettuste vältimiseks ja toimunud pettuste efektiivseks käsitlemiseks [2].

Pettuste liigid varieeruvad kasutatavate tehnoloogiate ja toodete lõikes. Andersen [2] toob välja kaheksa erinevat pettustega seotud dimensiooni (vt Joonis 1):

- Sihitud tooted: krediit- ja deebetkaardid, muud makseviisid
- Seotud äriprotsess: taotluste või tehingute protsessimine
- Petturi seotus kontoga: esimene (omanik), teine (nt tehingu vastaspool) ja kolmas (legitiimse rollita) osapool
- Ajastus: lühiajaline (petturi kiire rünnak) või pikaajaline (konto jälgimine ja ärakasutamine pikema perioodi jooksul)
- Erinevad identiteedi ärakasutamised: tegeliku info ilustus, vargus, väljamõeldud identiteet
- Toote omandamine petturi poolt: kaotatud või varastatud, teel kaotsi läinud, andmete kopeerimine ilma toodet omamata, kaardiandmete kopeerimine seadmega (*skimming*)
- Toote või toote detailide kasutusviis: võltsitud, ilma toote füüsilise kohaloluta, muudetud
- Seotud tehnoloogiad: pangaautomaadid (ATM), Internet, e-kaubandus

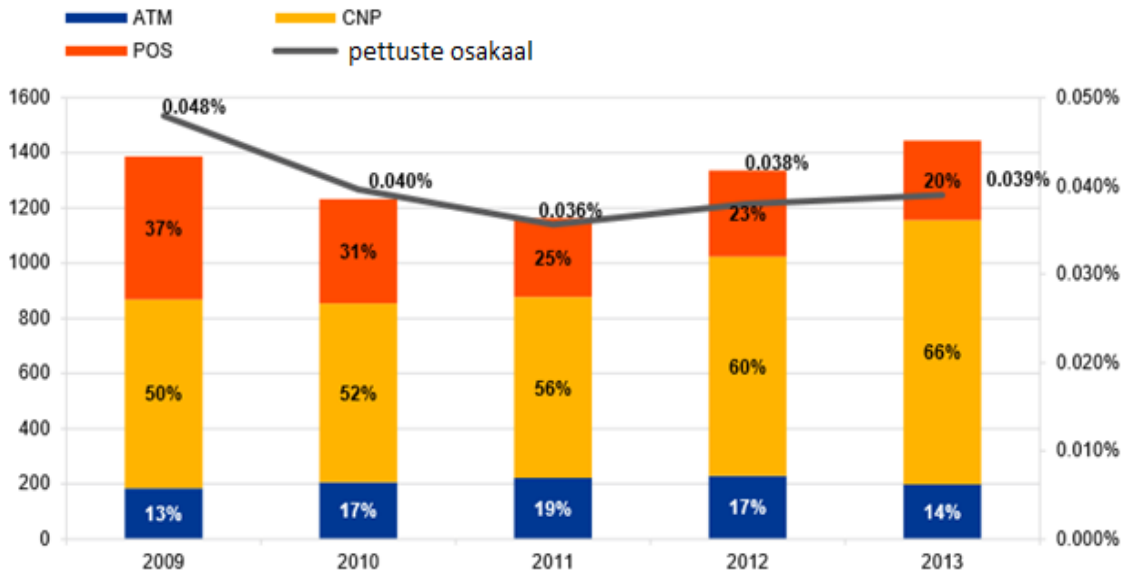
Joonis 1. Erinevad pettustega seotud dimensioonid.

Pettustega seotud vastutegevuste puhul on oluline eristada pettuste ärahoidmist ja pettuste avastamist. **Pettuste ärahoidmise** meetmete puhul on eesmärgiks hoida pettus juba eos ära, **pettuste avastamine** tegeleb juba toimunud pettuste võimalikult kiire identifitseerimisega, kui pettuste ärahoidmine on ebaõnnestunud. Reaalsuses peab võimalikke pettusi monitoorima järjepidevalt, sest sageli ei olda teadlikud, millal pettuste ärahoidmine on nurjunud. Näiteks kliendi pangakaardi andmete varastamise puhul on vaja võimalikult kiiresti tuvastada, millal hakkavad toimuma petturlikud tehingud, sest siis on võimalik järgmisi tehinguid ära hoida ja kahju kontrolli all hoida [7].

Antud töös on fookuses kaarditehingud ja nendega seotud pettused. Siinkohal saab eristada **kaardi füüsilist vargust** ning kaardi kiiret ja korduvat kasutamist petturi poolt kuni kaart jõutakse blokeerida. Mida kiiremini kaardi omanik või pank pettuse avastab, seda väiksem on tekitatav kahju. Teine oluline pettuse liik on **kaardi andmete kasutamine ilma füüsilise kaardi kohaloluta** näiteks veebipoodides [11].

2.2 Kaardipettuste mahud ja turu ülevaade

Kaardipettuste kogumaht SEPA (*Single Euro Payment Area*) piirkonnas ulatus 2013. aastal 1,44 miljardi euroni. Kaardipettuste maht on hakanud peale 2011. aastat taas kasvama (vt Joonis 2). Samal ajal on kasvanud ka kaarditehingute üldmahud, kuid pettuste osakaal on püsinud stabiilsena ca 0,04% juures. Täheldatav on aga pettuste mahu kasv olukordades, kus tehing sooritatakse füüsilist kaarti kasutamata (CNP, *cards not present*) nagu näiteks veebipoodides. CNP tehingute pettused on kasvanud 50%-lt 2009. aastal 66%-ni 2013. aastal, seejuures on müügiterminalides (POS, *point-of-sale*) ja pangaautomaatides (ATM, *automated teller machine*) tehingute osakaalud pettuste mahus vähenenud [12].

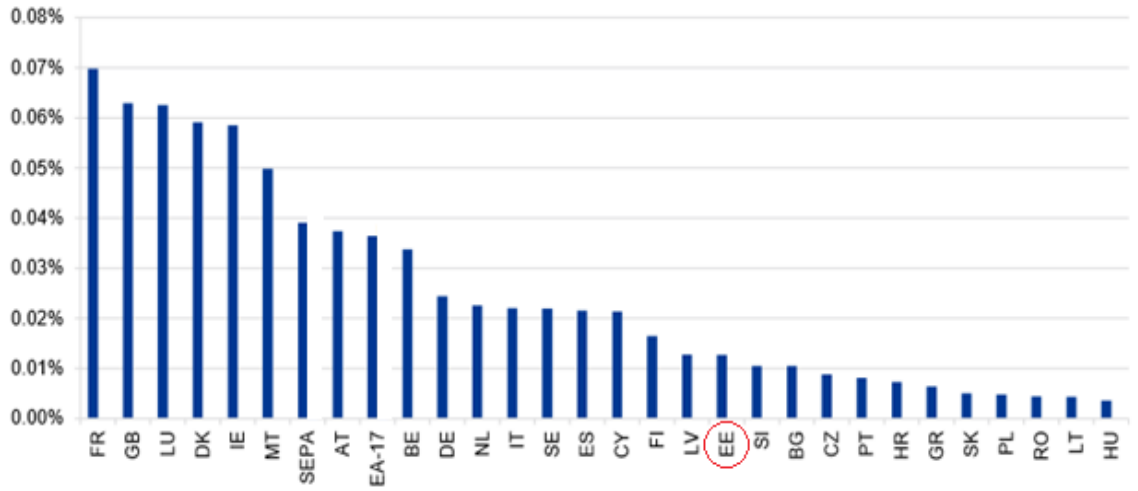


Joonis 2. Kaarditehingute pettuste mahud ja osakaalud [12].

Kaardi kohaloluga tehingute pettuste mahu vähenemist on oodata ka edaspidi, sest nii SEPA piirkonnas kui üleilmselt ollakse pea täielikult üle minemas kiibiga turvalisematele pangakaartidele (EMV standard) [12].

Kui vaadata tehingute jaotust siseriikliku ja piiriülesena, siis 2013. aastal moodustasid kohalikud tehingud 92%, piiriülesed SEPA piirkonnas 6% ja piiriülesed SEPA piirkonnast väljaspool 2%. Samal ajal olid petturlike tehingute puhul jaotused vastavalt 49%, 29% ja 22% ehk piiriülesed moodustavad poole pettustest [12]. Kaarditehingutega seotud pettuste monitoorimisel ja avastamisel tuleb seega eraldi tähelepanu pöörata piiriülestele tehingutele.

Eesti pangakaartidega seotud pettuste osakaal on oluliselt madalamal SEPA keskmisest (vt Joonis 3). Seejuures on Eesti pangakaartidega toime pandud pettuste maht perioodil 2009-2013 vähenenud ca 40%. Siinkohal on kindlasti mänginud suurt rolli Eesti pankade poolt rakendatud täiendavad meetmed internetiostude turvalisemaks muutmisel.



Joonis 3. Kaarditehingute pettuste osakaalud riigiti tehingute mahu järgi [12].

Euroopa Keskpanga statistikas paistab Eesti teiste riikidega võrreldes silma kõrge kaardikasutuse osakaaluga ja suhteliselt madala pettuste määraga. Seejuures tehakse aga küllaltki palju tehinguid piiride üleselt (ca 10%). Eesti keskmine kahjumäär aastas 1 000 kaardi kohta on 532 eurot ja iga tuhande elaniku kohta pannakse aastas toime kaardipettusi 711 euro ulatuses [12].

2.3 Varasemad uurimused masinõppe rakendamisest kaarditehingute pettuste tuvastamisel

Järgnevalt teeb autor ülevaate varasematest uurimustest ja artiklitest, kus on käsitletud masinõpet ja finantstehingute pettuste tuvastamist. Antud valdkonna kohta on avaldatud küllaltki palju erinevaid teadusartikleid. Artiklites on keskendutud erinevatele pettuste tuvastamisel esile kerkinud alamprobleemidele – millised masinõppe algoritmid sobivad antud probleemi lahendamiseks, millised peaksid olema modelleerimise aluseks võetavad algandmed, milliseid parameetreid on võimalik kasutada jne. Samuti on väga erinevaid lähenemisi tulemuste mõõtmisel alates traditsioonilistest statistilistest mõõdikutest kuni viimastel aastatel populaarsemaks muutunud kulupõhiste lähenemisteni.

2.3.1 Pettuste tuvastamiseks kasutatud masinõppe algoritmid

Varasemates töödes on kasutatud väga erinevaid masinõppe algoritme (vt Tabel 1) ja kindlasti ei ole võimalik välja tuua kindlaid, kõige paremini sobivaid algoritme. Olenevalt andmetest ja lähenemismeetoditest on tulemused olnud erinevad. Küll aga võib välja tuua mõned enamlevinud algoritmid, mis paljudes töödes korduvad, näiteks juhuslik mets, logistiline regressioon, tugivektor-masinad ja närvivõrgud.

Tabel 1. Varasemates uuringutes kasutatud masinõppe meetodid.

Autorid	Aasta	Algoritmid
Maes, S., Tuyls, K. [19]	2002	Kunstlikud närvivõrgud (ANN), Bayes närvivõrgud (BNN)
Hand, D. Adams, N. [15]	2009	Juhuslik mets (RF), logistiline regressioon (LR), tugivektor-masin (SVM), Bayes närvivõrgud (BNN), QDA, CART, KNN
Bhattacharyya, S., Jha, S., Tharakunnel, K., Westland, J.C. [6]	2011	Logistiline regressioon (LR), tugivektor-masinad (SVM), juhuslik mets (RF)
Bahnsen, A.C., Aouada, D., Sojanovic, A., Ottersten, B. [5]	2013	Logistiline regressioon (LR), C4.5, juhuslik mets (RF)
West, J., Bhattacharya, M. [25]	2016	Geneetiline programmeerimine, sipelgakoloonia, närvivõrgud, tugivektormasinad, otsustuspuu jt.
Bahnsen, A.C., Aouada, D., Sojanovic, A., Ottersten, B. [4]	2016	Otsustuspuu (DT), logistiline regressioon (LR), juhuslik mets (RF). Lisaks Bayes miinimum riski kriteerium ja kulutundlik lävend pettuste klassifitseerimisel.

Üldiselt keskendutakse piiratud arvule erinevatele algoritmidele ning algoritmi suurem keerukus ei tähenda ilmtingimata paremat tulemust. West [25] jõudis järeldusele, et parima täpsusega algoritmiks on tugivektor masinad (täpsus 91,5%), väga ligilähedasele tulemusele jõudis ka otsustuspuu. Bahnsen [4] saavutas parimaid tulemusi otsustuspuu ja logistilise regressiooniga, seejuures oli oluline kulutundliku lävendi kasutamine, mis parandas tulemusi kolmandiku võrra. Hand [15] töös oli järjepidevalt parim juhuslik mets, järgnesid tugivektor-masinad ja logistiline regressioon.

2.3.2 Treeningbaasid

Kuna pettused moodustavad kogu tehingute mahust väga väikese protsendi ning kogu tehingute kogumi kasutamine mudelite treenimiseks ei anna häid tulemusi, siis tuleb algandmeid valida. Dal Pozzolo [9] testis krediitkaartide pettuste andmete põhjal üheksat erinevat meetodit tasakaalustamata andmete probleemi lahendamiseks ja jõudis järeldusele, et enamus juhtudel andis parima tulemuse alaesindamise (*undersampling*) meetod.

Bahnsen [5] uuris alaesindamise mõju algoritmide tulemuslikkusele. Pettuste osakaal testitavates algbaasides oli vastavalt 1%, 5%, 10%, 20% ja 50%. Eesmärgiks oli uurida erinevate algoritmide tööd erinevate klassi-jaotuste korral. Tulemuste hindmisel kasutati kuluefektiivset Bayesi miinimum riski meetodit, mille puhul minimeeritakse tehingute klassifitseerimisel tekkivat kulu. Tulemusi võrreldi F_1 skoori ja kulude mahu järgi eurodes. F_1 skoori põhjal saavutati parim tulemus 5% pettuste osakaaluga algbaasi puhul. Osakaalu suurenemisel läksid tulemused järk-järgult halvemaks. Kulude järgi tekkis aga lineaarne seos pettuste osakaalu ja kulude vähenemise vahel, kus kulud olid kõige kõrgemad 1% baasi puhul ja madalaimad 50% baasi puhul.

Bahnsen [5] toob välja, et alaesindamise meetodeid saab rakendada vaid treeningandmetele, sest lõpptulemusena peavad algoritmid töötama reaalsetel andmetel, kus jaotus ongi tasakaalustamata.

2.3.3 Kulupõhine lähenemine

Hand [16] pakkus pettuste tuvastamise hindamiseks välja kulupõhise eksimismaatriksi, mis arvestab pettuseks klassifitseeritud tehingu kontrolliks administratiivkulu ja tuvastamata jäänud pettuse (FN, vale negatiivne) kuluks 100-kordse administratiivkulu.

Bahnsen [5] toob aga välja, et sellisel juhul eeldatakse ekslikult, et vale positiivne (FP) kannab sama kulu mis vale negatiivne (FN). Meetodit täiendatakse nii, et vale negatiivse kuluks määratakse tehingu summa, mis peaks kajastama suhteliselt täpselt petturlikust tehingust tingitud reaalset kulu. Samas töös jõuti järeldusele, et rakendades Bayes miinimum riski meetodit reaalsete kuludega eksimismaatriksiga, on võimalik saavutada paremaid tulemusi nii kulude kui F_1 skoori järgi. Lõpptulemusena jõuti parima tulemuseni rakendades juhusliku metsa algoritmi koos kuluefektiivse mudeliga. Meetodi

rakendamine võimaldas juhusliku metsa algoritmi puhul kulu vähendada täiendavalt 23% võrra.

Bahnseni töö kriitikaks võiks välja tuua, et lõpptulemuste F_1 skoorid on madalad, ulatudes vaid ca 0,1 piirimaile. Võrreldavateks algoritmideks on vaid C4.5, logistiline regressioon ja juhuslik mets. Eraldi tasub ka märkida, et kulusid võrreldakse absoluutsummades, mis muudab tulemused teiste uurimistöödega vähem võrreldavaks.

2.3.4 Pettuseks määratlemise piir

Tavapäraselt määratletakse tehing petturlikuks, kui tõenäosus on üle 50%, kuid see ei pruugi alati olla optimaalseim variant. Tehingute klassifitseerimise piirmäära on võimalik optimeerida arvestades kulusid, täpsemalt minimeerides kogukulu [5].

Bahnsen [5] eksperiment näitas, et rakendades Bayes miinimum riski meetodit, on võimalik kogukulu mõistes algoritmide tulemuslikkust parandada. Kulumaatriksi kasutamisel paranes kõige rohkem logistilise regressiooni tulemus (kulude mõistes). Väiksem oli efekt juhusliku metsa ja C4.5 algoritmi puhul, samas olid juba nende esialgsed tulemused oluliselt kõrgemal tasemel.

2.3.5 Varasemate tehingute agregeerimine

Hand [15] leidis, et varasemate tehingute agregeerimine kuni kolm päeva tagasi annab häid tulemusi, s.t. teeb mudelid täpsemaks. Üle ühe päeva tagasi agregeerimine oli efektiivsem kui ainult üks päev tagasi, samas enam kui kolme päevane periood tagasi enam olulist tulemusi parandavat mõju ei avaldanud.

Handi [15] tulemused näitasid, et varasemate tehingute agregeerimine töötab kõige paremini, kui kasutada võimekamaid klassifitseerimise algoritme (nt juhuslik mets ja tugivektor-masinad). Sama kehtis, kui pettuse märgistus oli algandmetes vähem täpne (nt kuupäeva täpsusega, mitte iga tehingu määratlusena) ning kui pettuste muutumine ajas on oluline faktor.

Bahnseni [4] tulemused näitasid, et algoritmide treenimisel ainult tehinguandmetega võib saavutada petturlikest tehingutest tingitud kahju vähenemist kuni 20% võrra, ainult agregeeritud näitajatega kuni 40% ja nii tehingupõhiste kui agregeeritud näitajatega kuni 60%. Parim tulemus saavutati kokkuvõttes kõiki kombineeritud näitajaid kasutanud kulutundliku otsustuspuuga, kus täiendav sääst küündis 70%-ni.

Whitrow [26] pakkus välja tehingute agregeerimise strateegia võttes arvesse kliendi varasemad kulutamisharjumused. Varasemad tehingud grupeeriti kõigepealt viimaste tundide ja päevade lõikes kaardi või konto järgi, edasi tehingu tüübi, kaupmehe grupi ja riigi järgi ning arvutati vastavad tehingute arvud ja rahalised mahud.

2.3.6 Erinevad parameetrid

Modelleerimise aluseks võetavad parameetrid on enamasti otseselt seotud tehinguga ning lisaks arvutatakse üldjuhul konto- või kaardipõhiselt mõned agregeeritud näitajad varasemate tehingute põhjal.

Kui varasemate tehingute agregeerimine on tavapärane, siis Bahnsen [4] kasutab modelleerimisel lisaks infot kliendi eelmise tehingu kohta, mille eesmärgiks on tuvastada väga erinevaid järjestikuseid tehinguid. Kasutatud näitajateks olid möödunud aeg eelmisest tehingust, eelmise tehingu summa ja riik.

Kliendi demograafiliste näitajate ja muude finantstoodete kasutuse info kaasamist mudelite väljatöötamisel varasematest töödest autor ei leidnud, kuid see võiks kindlasti olla üks täiendav uurimisteema. Kliendi profiil võiks ju olla seotud tema tehingute profiiliga ning pettuse ohvriks langemise tõenäosusega.

3. Teoreetilised aspektid

Käesolev peatükk annab ülevaate masinõppest üldiselt ning töös kasutatavatest masinõppe algoritmidest. Eraldi käsitletakse algoritmide tulemuste mõõtmise meetodeid, mis on antud töös erilise tähelepanu all. Peatüki lõpus tuuakse välja varasemates uurimustes esinenud masinõppe rakendamise kitsaskohad, millest tasub eksperimendi läbiviimisel juhinduda.

3.1 Masinõpe

Arvutiga probleemide lahendamine eeldab sobivate algoritmide kirjutamist, millega tehakse arvutile selgeks täpsed juhised, mida millises järjekorras ja tingimustel tuleb teha, et muuta sisend soovitud väljundiks. Mõnedel juhtudel on inimesel aga väga raske täpset algoritmi kirjutada. Meil on küll mingi suur hulk infot ja me teame, mida meil on vaja sealt otsida, kuid kõigi täpsete reeglite kirjapanek on hoomamatu ülesanne. Sellisel juhul soovitakse, et arvuti oleks võimeline ise looma algoritmi, mis suudaks piisava täpsusega näidisandmetest eraldada meid huvitava info. Masinõpe (ML, *machine learning*) on osalt tehisintellekt (AI, *artificial intelligence*), mis suudab kohaneda muutuva keskkonnaga ja omab seega õppimisvõimet [3].

Masinõppe aluseks võetakse teatud tasemel eeldefineeritud parameetritega mudel ja arvuti õppimine toimub näidisandmete põhjal optimeerides parameetrite ja tulemuslikkuse kriteeriumi. Mudel võib anda ennustusi tuleviku kohta või kirjeldada olemasolevaid andmeid teadmiste ammutamiseks või mõlemat [3].

Masinõppe algoritmidel võib olla erinevaid õppimisviise:

- Juhitud õpe (*supervised learning*): sisendiks on märgistatud (määratud tulemiga) treeningandmed, mille põhjal mudel peab õppima tegema tulemi osas õigeid ennustusi [8].
- Juhtimata õpe (*unsupervised learning*): sisendandmed ei ole tulemi osas märgistatud ja mudel peab leidma andmetest sarnaseid struktuure või looma üldistamise reegleid [8].
- Pooleldi juhitud õpe (*semi-supervised learning*): sisendandmed on segu märgistatud ja märgistamata näidetest ning mudel peab andmetest leidma nii sarnaseid struktuure kui tegema ennustusi [8].

- Kinnitusega õppimine (*reinforcement learning*): algoritm õpib katse-eksituse meetodil, millised tegevused toovad suurimat kasu. Komponentideks on siin agent (õppija või otsusetegija), keskkond (kõik, millega agent kokku puutub) ja tegevused (kõik, mida agent saab teha) [22].

Masinõppe algoritme on väga palju erinevaid ja enamasti grupeeritakse neid toimimismehhanismi sarnasuse järgi. Grupeerimise võimalusi on erinevaid ja üks algoritm võib kuuluda ka mitmesse erinevasse kategooriasse. Brownlee [8] toob välja seitse erinevat kategooriat kõige populaarsematest masinõppe algoritmidest.

- Juhtumil põhinevad algoritmid (k-lähima naabri algoritm (kNN), ise organiseeruv kaart (SOM), kohapeal kaalutud õppimine (LWL), õppevektori kvantimine (LVQ))
- Seaduspärasuse algoritmid (Ridge regressioon, elastne võrk, LASSO)
- Otsustuspuu algoritmid (tingimuslikud otsustuspuud, Hii-ruudu automaatne vastastikmõju avastamine (CHAID), üheastmeline otsustuspuu jne)
- Bayesi algoritmid (Naiivne Bayes, Gaussi naiivne Bayes, Bayesi võrk (BN), keskmist ühest sõltuvust ennustavad (AODE))
- Klasterdamise algoritmid (k-keskmine, k-mediaan, eelduste maksimeerimise algoritm (EM), hierarhiline klasterdamine)
- Tehislikud närvivõrkude algoritmid
- Süvanärvivõrkude algoritmid

Järgnevalt peatun täpsemalt nendel algoritmidel, mis on plaanis töö praktilises osas kasutusele võtta.

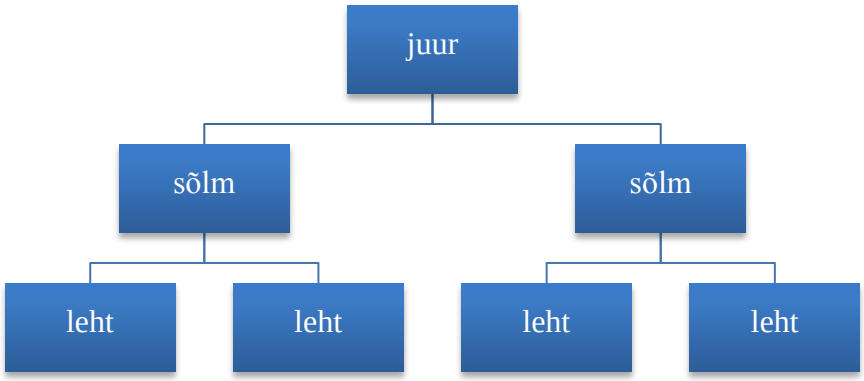
3.2 Valitud masinõppe algoritmid

Suurest hulgast erinevatest masinõppe algoritmidest on töösse valitud erinevate tööpõhimõtetega, kuid samas laialt levinud ning varasemates kaarditehingute pettuste tuvastamise uurimustes kasutatud algoritmid. Nendeks on otsustuspuu, juhuslik mets, *boost*, tugivektor-masinad, närvivõrgud ja logistiline regressioon.

3.2.1 Otsustuspuu

Otsustuspuu (nimetatakse ka klassifitseerimise ja regressiooni puu) on traditsiooniline osa andmekaevest ja masinõppe algoritmidest. Otsustuspuu atraktiivsus peitud mudeli lihtsuses ja arusaadavuses, mis võimaldab mudelit üle vaadata ja tõlgendada. Otsustuspuu ei ole alati parima täpsusega, kuid esindab kompromissi täpsuse ja mudeli arusaadavuse osas [27].

Otsustuspuu (*decision tree*) kasutab traditsioonilist struktuuri (vt Joonis 4), kus aluseks on üks sõlm, mis jaguneb erinevateks harudeks (oksteks), mis omakorda tipnevad sõlmedega, millest igaüks võib edasi hargneda või lõppeda lehega. Iga sõlme juures on test või küsimus, mis määrab, millist haru mööda edasi minna kuni jõutakse leheni ehk otsuseni [27].



Joonis 4. Otsustuspuu struktuur.

Otsustuspuu on oma lihtsuse ja arusaadavuse tõttu laialt levinud ning kaarditehingute pettuste tuvastamisel sagedasti kasutatud algoritm.

3.2.2 Juhuslik mets

Ühe otsustuspuu loomine pakub lihtsat mudelit vaadeldava probleemi lahendamiseks, kuid tihtipeale jääb see liiga lihtsaks või spetsiifiliseks. Varasemad andmekaeve kogemused on näidanud, et mitmete mudelite koostöö annab paremaid tulemusi. Mitme mudeli (nt otsustuspuu) kombineerimine ühte mudelite kogumisse annab otsustuspuude metsa [27].

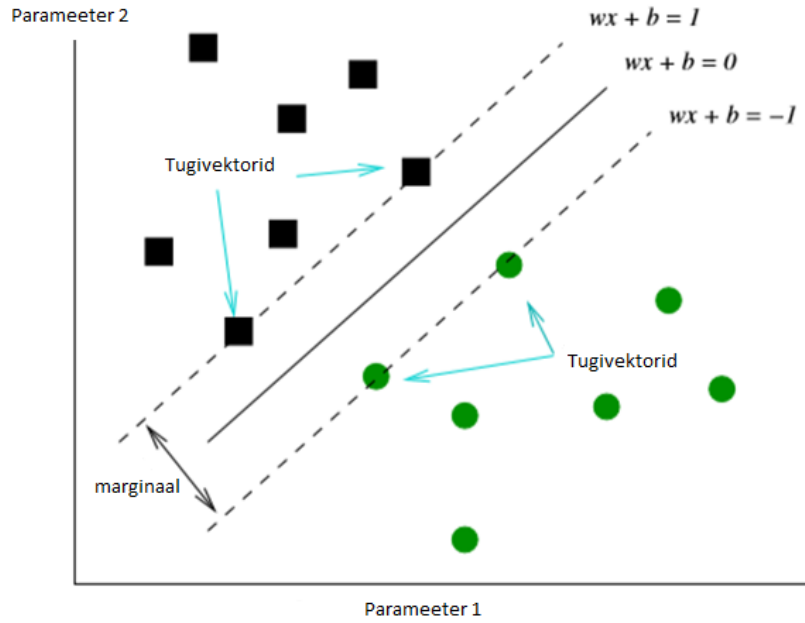
Juhusliku metsa puhul kasutatakse juhuslikku valikut nii vaatluste kui parameetrite valikul. Selle tulemusena on võimalik saavutada suurem sõltumatus andmete muutumisest, andmetes sisalduvast mürast ja ekstreemsetest vaatlustest ning

ülemäärasest sobitamisest algandmetele. Samuti on juhusliku metsa eeliseks parem toimetulek tasakaalustamata treeningandmetega. Näiteks kui binaarse klassifikatsiooni korral on petturlikke tehinguid algbaasis vaid kuni 5% ja ülejäänud on legitiimsed tehingud [27].

Juhusliku metsa kasuks räägib ka asjaolu, et see ei nõua suurt andmete eeltöötlemist, sest andmeid ei pea normaliseerima. Samuti pole vaja tegeleda parameetrite valikuga, sest algoritm teeb seda ise. Kuna mudeli paljud puud on koostatud kahe taseme juhuslikkuse alusel (vaatlused ja parameetrid), siis on iga puu eraldi sõltumatu mudel ja koondmudel on kokkuvõttes treeningandmetest vähem sõltuv [27].

3.2.3 Tugivektor-masinad

Tugivektor-masin (SVM, *support vector machine*) otsib nii-öelda tugivektoreid, mis on klasside äärealadel asuvad vaatlused, mille abil on võimalik klasse omavahel eristada. Klasside vahele jäävaid alasid nimetatakse klassidevaheliseks marginaaliks. Tugivektorite abil identifitseeritakse kahes dimensioonis klasse eraldav sirge joon (vt Joonis 5). Algoritmiga otsitakse maksimaalset marginaali erinevate klasside vahel [27].

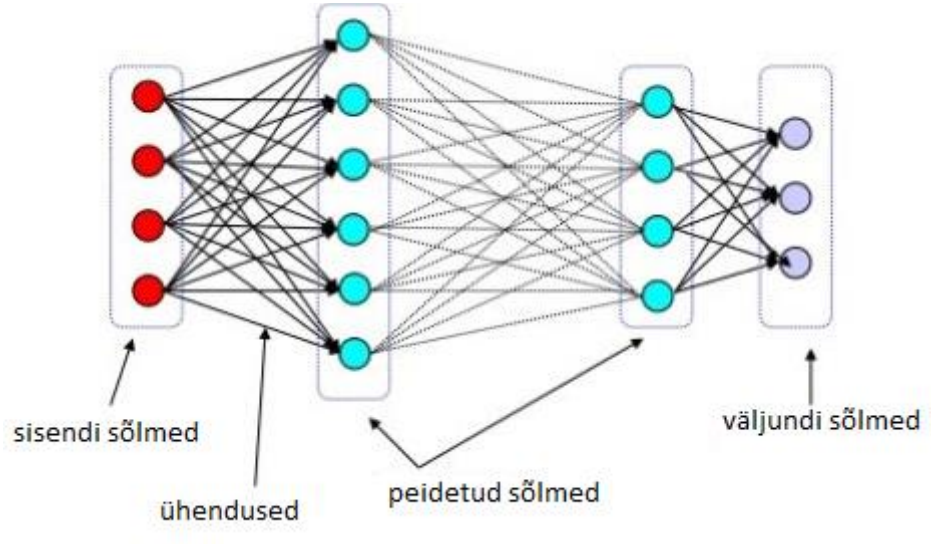


Joonis 5. Tugivektor-masina algoritmi tööpõhimõte.

Tugivektor-masinate tugevuseks on hakkama saamine probleemidega, mis on mittelineaarsed, hajusad ja kõrge dimensionaalsusega. Puuduseks on asjaolu, et algoritm on tundlik seadistamise valikutele (nt milliseid teisendusi andmetega tehakse), mis teeb kasutamise keerulisemaks ja õige mudeli leidmise ajamahukamaks. Meetodi eeliseks on asjaolu, et mudel tegeleb ainult nende tugivektoritega mitte kogu andmekogumiga ja seetõttu pole treeningvalimi suurus oluline probleem. Võttes arvesse, et tegeletakse vaid tugivektoritega, siis mudelit ei häiri äärmuslikud vaatlused [27].

3.2.4 Närvivõrgud

Närvivõrkude mudeli loomisel on inspiratsiooni saadud bioloogilistelt neuronitelt. Närvivõrgud on tunnetusliku õppimise täiendus mittelineaarsete otsuste jaoks. Tunnetusliku õppimise puhul on sisendinfo (nt pilt) otseselt seotud väljundiga (nt silt). Seda kutsutakse tihtipeale ühekihiliseks võrguks, sest eksisteerib üks kiht kaalusid. Närvivõrgu puhul on sisendi ja väljundi vahele sisestatud täiendav kiht peidetud sõlmi (otsustuspunkte), millega on saadud mitmekihiline võrk (vt Joonis 6). Sisemises kihis kasutatakse mittelineaarset algoritmi ja selle tulemusena on mudeli otsustuspiirid samuti mittelineaarsed. Sellised võrgud on võimelised väljendama peaaegu ükskõik millist funktsiooni peale lineaarse. Sellise paindlikkusega tuleb aga kaasa keerukuse kasv parameetrite häälestamisel ja mudeli disainis [10].



Joonis 6. Närvivõrgu algoritmi tööpõhimõte.

Närvivõrgu ülipeensus väljendub kihtide arvus ja keskmiste kihtide peidetud sõlmedes, mis teeb mudeli ülesehitamise raskemaks ja tõlgendamise keeruliseks. Näiteks lineaarse mudeli korral on lihtne interpreteerida kaalusid suuruse järgi, kuid mitmekihilise võrgu puhul on raske mõista, mida erinevad peidetud kihid ja sõlmed teevad [10].

3.2.5 Boosting

Boosting meta-algoritm on efektiivne, kiire ja lihtsasti kasutatav lähenemine mudelite ehitamisele. Üks populaarsemaid versioone on adaptiivne *boosting* (AdaBoost). *Boosting* algoritmid ehitavad andmestiku pealt mitu erinevat mudelit mõne teise õppimisalgoritmi põhjal, mis ei pruugi olla eriti hea õppija. *Boosting* seostab algandmetes kaalud vaatlustega ja suurendab kaalusid nendel vaatlustel, mida on raske täpselt modelleerida. Ehitatakse jada mudeleid ning peale iga mudeli ehitamist muudetakse kaalusid nii, et tõsta osatähtsust vaatlustel, mida on raskem klassifitseerida. Mudeli rakendamisel klassifitseerib iga alammudel (puu) vaatluse ning lõpptulemus leitakse kõigi alammudelite ennustuste keskmisena. Rakendamisel võib olla teatud varieeruvusi kaalude ümberarvutamisel ja iga mudeli tulemuse osakaalul, kuid üldine kontseptsioon jääb samaks. *Boosting* võib ebaõnnestuda kui andmeid pole piisavas mahus, nõrgad mudelid on liiga keerukad või andmetes on palju müra [27].

3.2.6 Algoritmide kokkuvõtlik võrdlus

Järgnevalt on kokkuvõtlikult välja toodud algoritmide peamised tugevused ja nõrkused (piirangud) (vt Tabel 2). Lisaks eelnevalt toodud materjalidele on eeskujuks West [24].

Tabel 2. Algoritmide tugevuste ja nõrkuste kokkuvõtlik võrdlus.

Meetod	Tugevused	Nõrkused (piirangud)
Logistiline regressioon	Lihtne rakendada. Pikalt kasutatud pettuste tuvastamisel.	Madalam klassifitseerimise täpsus võrreldes muude meetoditega. Raske hakkama saada pettuste tuvastamise keerukusega.
Otsustuspuu	Lihtne rakendada ja aru saada. Treenimine ja käitamine ei nõua suurt arvutusvõimsust ning omab potentsiaali reaalajas rakendamiseks.	Potentsiaalselt võib üle-sobitada treeningandmetele, kui need ei ole piisavalt esinduslikud. Nõuab pidevat taastreenimist kohanemaks muutustega andmetes.

Meetod	Tugevused	Nõrkused (piirangud)
Juhuslik mets	Suurem sõltumatus andmete muutumisest, andmetes sisalduvast mürast ja ekstreemsetest vaatlustest ning üle-sobitamisest. Ei vaja andmete eeltöötlemist ja parameetrite eelvaliku teostamist.	Mudeli koostamine nõuab rohkem arvutusressurssi.
Tugivektor-masinad (SVM)	Võimelised lahendama mittelineaarseid klassifikatsiooni probleeme, sobivad hästi pettuste tuvastamiseks. Treenimine ja käitamine ei nõua suurt arvutusvõimsust ning omab potentsiaali reaalajas rakendamiseks.	Raske mudelit tõlgendada ja hinnata.
Närvivõrgud	Laialt kasutatud pettuste tuvastamisel. Tõestanud sobivust erinevate mitte-algoritmiliste ja binaarsete klassifikatsiooni probleemide korral. Paindlik ja sobib keerukate probleemide lahendamiseks.	Nõuab suurt arvutusvõimsust nii mudeli koostamisel kui käitamisel ning seetõttu võib olla mittesobiv reaalajas rakendamiseks. Potentsiaalselt võib üle-sobitada treeningandmetele, kui need ei ole piisavalt esinduslikud. Nõuab pidevat taastreenimist kohanemaks muutustega andmetes. Keerukus parameetrite häälestamisel ja mudeli tõlgendamisel.
Boosting	Võimaldab saavutada suurt täpsust ka keerukamate probleemide puhul.	Ei sobi, kui andmeid pole piisavas mahus, nõrgad mudelid on liiga keerukad või andmetes on palju müra.

3.3 Tulemuste mõõtmine

Masinõppe algoritmide edukuse mõõtmine on oluline samm nende sobilikkuse hindamisel, eriti finantspettuste puhul, sest väike paranemine täpsuses võib viia olulise majandusliku tulemuseni. Alljärgnevalt on toodud ülevaade meetoditest, mida kasutatakse pettustega seotud masinõppe mudelite tulemuslikkuse hindamisel [24].

Klassifitseerimine: Täpsus (*accuracy*), tundlikkus (*sensitivity*), esitustäpsus (*precision*), õige positiivse suhe õige positiivse ja vale negatiivse summasse (*recall*), vale positiivne määr, F-mõõdik (*F-measure*), Fβ, kulu minimeerimine.

Statistilised mõõdikud: Z-skoor, vea ruutude summa (*sum of squared error*).

Seotuse reeglid: tugi (*support*), kindlus (*confidence*), Lift.

Visuaalne: ROC kurv, AUC (*area under the curve*)

Pettuste tuvastamise puhul on valimid üldjuhul oluliselt tasakaalust väljas, sest õigeid tehinguid on oluliselt rohkem kui petturlikke tehinguid. Samuti kaasnevad erinevad kulud tehingute klassifitseerimisel õigeks või petturlikuks. Sellisel juhul näitavad nii mõnedki mõõdikud madalamat sensitiivsust ja kõrgemat täpsust, kui see tegelikult on [24].

3.3.1 Visuaalsed mõõdikud

ROC graafik kaardistab õiget positiivset määra vastu vale positiivset määra ning AUC-d kasutatakse kui standardset mõõdikut hindamaks klassifikatsiooni tulemuslikkust [15].

Sensitiivsus (*sensitivity/specifity*) kaardistab õiget positiivset määra vastu õiget negatiivset määra. Lift graafik näitab suhtelist suurenemist ennustusvõimes positiivsete ennustuste määra suhtes [27].

Pettuste klassifitseerimise puhul ei pruugi antud näitajad olla aga sobivaimad, sest need ei arvesta erinevaid kulusid, mis kaasnevad positiivse ja negatiivse otsusega [15]. Visuaalseid mõõdikuid võib ühe osana tulemuste hindamisel kasutada, kuid kindlasti peab silmas pidama nende meetodite piiranguid ning analüüsima tulemusi koos teiste mõõdikutega, eriti pettustest tekkivat kulu arvesse võttes.

3.3.2 Klassifitseerimise mõõdikud

Eksimismaatriks (c*onfusion matrix*) (vt Tabel 3) on tüüpiline viis binaarsete klassifikatsiooni algoritmide hindamiseks [5].

Tabel 3. Binaarse klassifikatsiooni eksimismaatriks.

		Õige klass (y_i)	
		Petturlik	**Legitiimne**
Ennustatud klass (p_i)	**Petturlik**	TP	FP
	Legitiimne	FN	TN

Tabelis TP – õige positiivne (true positive), FP – vale positiivne (false positive), FN – vale negatiivne (false negative), TN – õige negatiivne (true negative)

Tabeli põhjal saab välja tuua järgnevad statistilised näitajad:

Valesti klassifitseerimine (*misclassification*) = $1 - \frac{TP+TN}{TP+TN+FP+FN}$ (1)

Recall = $\frac{TP}{TP+FN}$ (2)

Täpsus (*precision*) = $\frac{TP}{TP+FP}$ (3)

F_1-skoor = $2*\frac{Precision*Recall}{Precision+Recall}$ (4)

F_β-skoor = $\frac{(1+\beta^2)*TP}{(1+\beta^2)*TP+\beta^2*FN+FP}$ (5)

Petturlike tehingute tuvastamisel on olulisem *recall* ning sellisel juhul β=2 ja valemi tulemuseks on F_2 skoor (valem 5).

3.3.3 Kulupõhine lähenemine

Bahnsen [5] on välja pakkunud algoritmide hindamise meetodi, mis peaks realistlikult arvesse võtma rahalisi võite ja kahjusid, mis kerkivad esile pettuste tuvastamisel. Aluseks on võetud Bayes´i miinimum riski klassifikaator ning kulude hindamisel kasutatud reaalseid kulusid (vt Tabel 4).

Tabel 4. Kulumaatriks kasutades reaalseid finantskulusid.

		Õige klass (y_i)	
		Petturlik	**Legitiimne**
Ennustatud klass (p_i)	**Petturlik**	C_a	C_a
	Legitiimne	Amt_i	0

Toodud tabelis tähistab C_a tehingu kontrollimisega kaasnevaid administratiivseid kulusid ja Amt_i tehingu i summat, mis on vale negatiivse määratluse tõttu arvestatud kuluks. Selline kulumaatriks kirjeldab paremini tegelikke kulusid, sest kui pettus ei ole õigesti tuvastatud, siis vastava pettusega seotud kahjud vastavad tehingu summale [4]. Kulumõõdik on vastavalt eeltoodud kulumaatriksile defineeritud seega kui:

$C = \sum_{i=1}^{m} y_i (p_i C_a + (1 - p_i) Amt_i) + (1 - y_i) p_i C_a$ (6)

Valem (6) hindab kulude summat m transaktsiooni jaoks, kus y_i ja p_i on reaalsed ja ennustatud märgistused. Antud kulumaatriks ei võimalda üksnes hinnata kulusid, vaid

seda kasutatakse ka kulutundliku klassifikatsiooni algoritmi arendamiseks Bayesi miinimum riski põhjal [4].

Bayes´i miinimum riski klassifitseerija on otsustusmudel, mis kvantifitseerib kompromisse erinevate otsuste vahel kasutades tõenäosusi ja kulusid, mis otsustega kaasnevad. Kaarditehingute pettuste puhul on kaks võimalikku otsust – kas tehing on legitiimne (p_1) või petturlik (p_f) [13].

Kui arvestada Tabel 4 toodud kulumaatriksit, siis tehing klassifitseeritakse petturlikuks, kui järgmine tingimus on tõsi ja legitiimseks kui vale [5].

$$C_a P(p_f x) + C_a P(p_l \mid x) \leq Amt_i P(p_f \mid x) \quad (7)$$

, kus p_f – ennustab tehingu petturlikuks (*fraud*), p_l – ennustab tehingu õigeks (*legitimate*), $P(p_f)$ – tõenäosus, et tegu on petturliku tehinguga, $P(p_l)$ – tõenäosus, et tegu on õige tehinguga.

Kui pank klassifitseerib õige tehingu valeks, siis pank kaotab tehinguga seotud kasumimarginaali, mis jääb teenimata. Kui aga klassifitseeritakse vale tehing õigeks, siis kaotatakse kogu tehingusumma. Seega, et töötada panga aktsionäride parimates huvides, peaks panga eesmärk olema valesti klassifitseerimise kulude minimeerimine [11].

3.3.4 Valitud mõõdikud

Tulemuste mõõtmisel on fookuses F_2 skoor (F_1 skoor võrdluseks) ja kuluga kaalutud kahju funktsiooniga saavutatud kulusääst.

Täiendavalt vaadeldakse ka sensitiivsust, valesti klassifitseerimist, õige positiivse suhet õige positiivse ja vale negatiivse summasse (*recall*), esitustäpsust ja ROC-i. Lisaks on oluline jälgida palju tehinguid mudelid pettusteks klassifitseerivad, sest see määrab kontrollitavate tehingute mahu, mis peab olema realistlikult spetsialisti poolt ülevaadatav.

Lõpptulemuste hindamisel on siiski eriline tähelepanu kulupõhistel mõõdikutel, et minimeerida pettustest saadavat kahju.

3.4 Tüüpilised kitsaskohad kaarditehingute pettuste tuvastamise süsteemides

Järgnevalt on välja toodud tüüpilised kitsaskohad kaardipettuste tuvastamise süsteemide loomisel ja käigushoidmisel:

- Asümmeetriline jaotus (*skewed distribution*): kuna ainult väike osa kogutehingutest on petturlikud, siis algoritmide treenimisel ja kasutamisel peab süsteem sellega toime tulema [19].
- Võime toime tulla andmetes esineva müraga. Andmetes võib olla erinevaid vigu, nii sisulisi kui vormilisi. Müra reaalsetes andmetes piirab mudeli üldistamise täpsust [19].
- Kattuvad andmed (*overlapping data*): paljud tehingud võivad sarnaneda petturlikele tehingutele, kuigi tegelikult on legitiimsed. Sama kehtib ka vastupidi [19].
- Kohanemisvõime (*concept drift*): pettusi tuvastavad süsteemid peavad olema võimelised kohanema uute pettuste tüüpidega [19]. Kui teatud tüüpi pettus on avastatud, siis õige pea mõeldakse professionaalsete petturite poolt välja uus või modifitseeritud meetod, mida süsteem kohe ei suuda tuvastada [21].
- Erinevate pettuste laadide tuvastamine: samal ajal võib toimuda väga eritüübilisi pettusi, mis võivad seejuures olla regulaarse, juhusliku, sesoonse või ühekordse loomuga [21].
- Õiged klassifikatsiooni hindamise mõõdikud: tavapärased klassifikatsiooni mõõdikud nagu ROC ei tööta asümmeetrilise jaotuse puhul eriti hästi [19].
- Kulude arvesse võtmine: arvesse peab võtma nii petturliku tehinguga kaasnevaid kahjusid kui selle takistamiseks tehtavaid kulutusi [19]. Kulud ei tohi ületada tulusid ja samamoodi peab monitooringu süsteem suutma tuvastada eelkõige just neid pettusi, millest tulenevad kõige suuremad potentsiaalsed kahjud.

Eeltoodud kitsaskohtadest johtuvalt on antud töös klassifikatsiooni hindamise kõrval keskendutud kulupõhistele näitajatele, mis võtavad arvesse reaalseid kulusid. Andmete müraga ja kattuvusega seotud probleeme on raske konkreetsete meetoditega lahendada ning pigem tuleb jälgida, kuidas erinevad algoritmid erinevate andmete puhul käituvad.

Andmete asümmeetrilisusele tuleb aga treeningbaaside koostamisel eraldi tähelepanu pöörata. Teema on järjest aktuaalsem ka erinevates teadusartiklites, näiteks sellel teemal aastas avaldatud artiklite arv on viimase kümne aastaga kordades kasvanud (7-lt 118-ni) [14].

Andmete asümmeetrilisuse lahendamiseks kasutatud meetodid:

- Alaesindamine (*undersampling)* eemaldab osa andmeid ülekaalus olevast klassist. Sellist lähenemist on kasutatud väga laialdaselt erinevates pettuste tuvastamise süsteemides [1].
- Üleesindamine (*oversampling*) replikeerib alakaalus olevas klassis olevaid andmeid. Üleesindamist kasutatakse suhteliselt harva, sest see võib põhjustada mudeli liigset sobitamist andmetega, eriti kui andmetes on võrdlemisi palju müra. Eraldi meetod on SMOTE (*Synthetic Minority Oversampling*), mis genereerib alakaalus olevatest andmetest sünteetilisi uusi andmeid [1].
- Hübriidne meetod kombineerib nii üle- kui alaesindamise meetodeid [14].
- Kulutundliku algoritmi põhine meetod kasutab erinevate vea tüüpide jaoks kulumaatriksit hinnastamaks klassifikatsiooni õigsust [1]. Algoritm püüab minimeerida kulusid või maksimeerida kasumit ning väldib kulukate vigade teket (nt pettuste mitte tuvastamist) ja annab seega kõrgema kaalu petturlikele tehingutele.

Üldiselt toimivad paremini andmete meetodid kui algoritmil põhinevad meetodid. Seda seetõttu, et andmete muutmise meetodeid on lihtsam rakendada ja see ei vii treeningaja ega muude ressursside kulu suurenemiseni. Enamik pettuste tuvastamise uuringuid põhinevad andmete meetoditel [17].

Dal Pozzolo [9] krediitkaartide pettuste andmete põhjal läbi viidud väga põhjalik üheksa erineva meetodi võrdlus tunnistas parimaks alaesindamise meetodi. Suhteliselt ligilähedase tulemuse andis veel ka SMOTE, kuid teised võrreldavad meetodid olid väga võrdsel tasemel ning esimestest arvestatavalt vähemtäpsed. Samas toodi välja, et selliste võrdluste puhul sõltub väga palju võrdluse aluseks võetavatest mõõdikutest ning algandmetest.

Hulse [18] jõudis alaesindamise meetodite põhjaliku analüüsi tulemusel järeldusele, et tavaline juhuslik alaesindamine töötab paremini kui keerukamad lahendused (nn intelligentsed meetodid).

Kohanemisvõime tagamiseks võib kasutada kahte erinevat meetodit sõltuvalt sellest, millal adaptiivne funktsioon aktiveeritakse: arenemisel põhinev (*evolving based*) või reguleerimisel põhinev (*regulated based*). Esimesel juhul õppija (algoritm) kohandab automaatselt enda käitumist vastavalt jooksvatele andmetele, teisel juhul jälgivad detektorid (eraldiseisvalt klassifitseerimisest) ja teavitavad muutumisest käitumises (andmetes) ning olukorra analüüsimise järel tehakse vajalikud muudatused süsteemi [1].

4. Eksperimendid masinõppe rakendamisest kaarditehingute pettuste tuvastamisel

Käesolevas peatükis on detailsemalt kirjeldatud, kuidas eksperimendid läbi viidi, millised olid andmed ja millistele järeldustele jõuti.

4.1 Eksperimendi disain

Eksperiment koosnes viiest suuremast etapist, mis on lühidalt kirjeldatud alltoodud punktides.

1. Ajalooliste andmete kogumine

Eksperimendi ettevalmistuse aluseks oli ajalooliste andmete kogumine kaarditehingute ja petturlike tehingute kohta. Oluline oli tuvastada petturlikud tehingud ja viia need kokku kõigi andmebaasis olevate tehingutega.

2. Parameetrite valik, teisendamine ja agregeerimine

Kui andmebaasi tasemel olid tuvastatud kõik vastava perioodi legitiimsed ja petturlikud tehingud, siis järgmiseks etapiks oli välja selgitada kogu saadaolev info tehtud tehingute kohta ja kaardistada kättesaadav seotud info. Valitud parameetrid tuli vajadusel teisendada sobivasse formaati ning luua varasemate tehingute põhjal uued agregeeritud näitajad.

3. Treening, valideerimise ja test baaside loomine

Arvestades asjaolu, et oli teadmata, millise treeningbaasi pealt on võimalik saavutada parimaid tulemusi, siis kõigepealt loodi alaesindamise meetodit kasutades erinevate pettuste osakaaludega treeningbaasid. Saadud tulemuste võrdlemiseks loodi eraldi valideerimisbaas, mis oli kõigi treeningbaaside jaoks ühine. Et hinnata erinevate parameetrite gruppide kaasamist, kasutati esimeses etapis parimat tulemust andnud treeningbaasi.

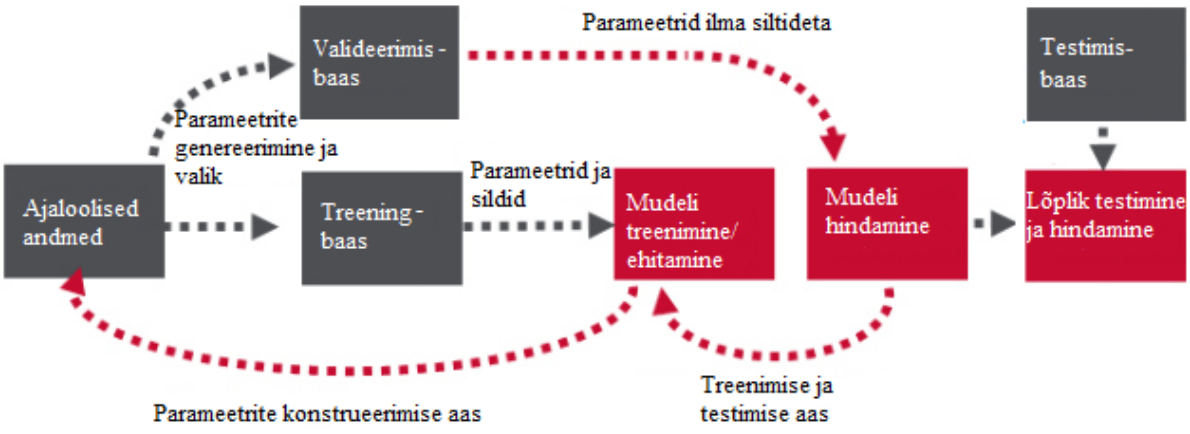
4. Mudelite treenimine ja testimine

Treeningbaaside peal rakendati kuut masinõppe algoritmi. Kõigepealt treeniti algoritme erinevate pettuste osakaaludega treeningbaasidega ja seejärel testiti valideerimisbaasi peal. Parima tulemuse andnud treeningbaasi kasutati algoritmide edasiseks treenimiseks ja erinevate parameetrite gruppide võrdlemiseks. Peale igale algoritmile sobivaimate parameetrite leidmist treeniti lõplikud mudelid.

5. Lõplike algoritmide testimine testbaasi peal

Viimases etapis testiti lõplikke algoritme uute andmete põhjal koostatud testbaasi peal. Testimise tulemusena oli võimalik anda eksperimendi edukusele lõplik hinnang.

Eelpool kirjeldatud protsess on kirjeldatud ülevaatlikult Joonis 7. Seejuures tuleb märkida, et antud protsessi sees sai mudelite treenimist ja valideerimisbaasiga hindamist korratud väga mitmeid kordi nii parima treeningbaasi leidmiseks kui parameetrite valimiseks. Viimase faasi testimisse ja hindamisse jõudsid vaid parimad treenitud algoritmid.



Joonis 7. Algoritmide treenimise ja testimise protsess.

Allikas: Eeskujuks on võetud McDonaldi [20] joonis, mida on täiendatud vastavalt autori eksperimendile.

Järgnevates alampeatükkides on selgitatud detailsemalt andmete kogumist, parameetrite loomist, testbaaside koostamist ja mudelite treenimist.

4.1.1 Andmete kogumine

Andmete kogumise esimeseks etapiks oli süsteemis eraldi paiknevate tuvastatud pettuste ülevaatamine ning tehingutele unikaalsete tunnuste loomine. Petturlikest tehingutest loodi andmebaasi eraldi tabel, mida oli võimalik kokku viia kogu muu andmebaasi infoga. See tegi võimalikuks tehingute sidumise konkreetsete klientidega ning muude lepingute ja tunnuste infoga. Kogu algandmete edasine töötlemine sai toimuda SQL serveris.

Andmete töötlemise hoidmine SQL serveris oli väga oluline, sest tehingute mahud ulatusid miljonitesse. Alaesindamise meetodi rakendamiseks oli samuti mõistlik kasutada SQL koodi.

4.1.2 Sisendparameetrid

Andmete töötluse järgmiseks etapiks oli erinevate parameetrite loomine. Modelleerimise täielikku algbaasi valiti 65 erinevat parameetrit, mis jagati kaheksasse parameetrite gruppi. Kõikide parameetrite loetelu koos andmetüüpidega on toodud lisas 1.

Parameetrite loomise protsess sisaldas otsese tehinguinfo väljavõtmist, tehingu detailandmetest koodide dešifreerimist täiendavateks tunnusteks, kliendi demograafiliste näitajate ja täiendavate lepingute info lisamist, erinevate varasemate tehingute agregeerimist vastavalt tehingu ajale, kohale, meetodile jne. Lõpuks koostati koondpäring, mis lisas igale tehingu reale juurde kõik täiendavad parameetrid.

Järgnevalt on loetletud ja lühidalt kirjeldatud loodud parameetrid gruppide kaupa.

Kaarditehingu tunnused: kaarditoode, tehingu pool (krediit/deebet), tehingu tüüp (POS, ATM), tehingu riik, kaupmehe liik, kaupmehe grupp MCC jaotuse järgi, kaupmehe riskikategooria, tehingu aeg (kuupäev, päev, tund), tehingu summa.

Kaardiomaniku profiil: kliendi tüüp (era/äri), sugu, vanus, keel, residentsus, kodakondsus, elukoht (maakond, linn), haridus, tegevusala, kliendilepingu sõlmimise kanal (kliendikontor, partnerettevõte), reklaami lubamise info, investeerimisteenuste leping, kasvukonto leping, kuldkliendi leping (Au klient), eraisiku puhul ettevõtte konto omamine samas finantsasutuses.

Kaarditehingu teostanud terminali ja kliendi/kaardi tuvastuse info: terminali autentimisvõimalused, terminali kaardi sisestamise võimalus, kaardiomaniku kohalolu

tehingu tegemisel, kaardi kohalolu tehingu tegemisel, kaardi info lugemise ja kaardiomaniku tuvastamise meetod.

Tehingule eelnenud välismaiste tehingute summad: agregeeritud erinevate perioodide lõikes (1, 2, 3, 7, 30 päeva jooksul) arvestades vaadeldavat ajaakent iga tehingu korral eraldi.

Tehingule vahetult eelnenud välismaine kaarditehing: eelnenud tehingu riik, tehingu liik, kaupmehe liik, kaupmehe riskigrupp, tehingu summa, tehingu aeg, ajavahe minutites käesoleva tehinguga.

Viimase 24 tunni jooksul tehtud sarnased tehingud: samas riigis tehtud tehingute summa ja arv, samas kaupmehe grupis tehtud tehingute summa ja arv.

Viimase 30 päeva keskmine välismaiste tehingute arv: tehingule eelnenud 30 päeva jooksul tehtud välismaiste tehingute keskmine summa ja tehingute arv.

CNP (kaardi kohaloluta) tehingud: viimase 30 päeva jooksul tehtud tehingute summa ja arv, tehingule eelnenud 24 tunni jooksul tehtud tehingute summa ja arv.

Treeningbaaside võrdlusel kasutati 14 valitud parameetrit, s.h. tehingu detailandmeid, terminali ja kaardi kohalolu infot ning osasid kliendi demograafilisi näitajaid. Parameetrite gruppide võrdlusel kaasati eelloetletud gruppe tervikuna.

4.1.3 Treening-, valideerimis- ja testbaasid

Varasemates uurimustes (nt Hand [15]) on kasutatud algbaasi juhuslikku jagamist 70% treeningbaasiks ja 30% testbaasiks. Autori andmetega tehtud esialgsed testid näitasid aga, et selline meetod ülehindab oluliselt mudelite tulemuslikkust. Seda seetõttu, et tihtipeale tehakse sarnaseid pettusi järjest ühes jadas. Näiteks kaardiga proovitakse petturlikku tehingut läbi viia järjest 5 korda samas kaarditerminalis ja isegi samades summades. Juhusliku valiku puhul on suur tõenäosus, et sellisel juhul satub sarnane tehing nii treening- kui testbaasi ja algoritmil on suhteliselt lihtne testbaasi sattunud tehing petturlikuks klassifitseerida. Lisaks räägib sellise meetodi kahjuks asjaolu, et erinevate mahtudega treeningbaasidel on erinevate mahtudega testbaasid, mis teeb tulemused raskemini võrreldavaks.

Treeningbaas mudelite väljatöötamiseks: 665 petturlikku tehingut perioodil november 2015 kuni oktoober 2016. Treeningbaasi valikul kasutatud erinevate pettuste osakaaludega baaside mahud on toodud Tabel 5.

Tabel 5. Petturlike tehingute osakaalud algbaasides.

Pettuste osakaal	Petturlikke tehinguid	Legitiimseid tehinguid	Kokku
1%	665	65 835	66 500
5%	665	12 635	13 300
10%	665	5 985	6 650
20%	665	2 660	3 325
50%	665	665	1 330

Valideerimisbaas treeningbaaside ja parameetrite võrdlemiseks: tehingud perioodil november kuni detsember 2016, sh 209 petturlikku tehingut. Kogu baasi mahuks 20 tuhat tehingut, mis teeb pettuste osakaaluks 1%. Selline jaotus peegeldab reaaleluliste andmete asümmeetrilisust, millega algoritmid peavad toime tulema, kuid samas on maht piisavalt piiratud, et oleks võimalik läbi viia märkimisväärne arv erinevaid algoritmide ja parameetrite testimisi.

Testbaas lõplike mudelite hindamiseks: tehingud perioodist jaanuar kuni veebruar 2017, sh 284 petturlikku tehingut. Baasi eesmärgiks on hinnata lõplike algoritmide sobivust võimalikult reaalelulises olukorras ja seetõttu kaasati kõik antud perioodi välismaal tehtud tehingud (üle 200 tuhande). Pettuste osakaal baasis oli selle tulemusena 0,13%.

4.1.4 Algoritmide treenimine ja testimine

Masinõppe mudelite koostamiseks kasutati R keelt. Selle kasuks rääkis asjaolu, et see on vabavaraline ning väga laialt levinud. R keeles on eraldi funktsionaalsused masinõppe algoritmide treenimiseks ja testimiseks ning genereeritud algoritme on R koodi abil võimalik rakendada ka teistes keskkondades.

Kasutatavateks masinõppe algoritmideks valis autor otsustuspuu (DT), juhusliku metsa (RF), *boost*-i, tugivektor-masinad (SVM), närvivõrgud (NN) ja logistilise regressiooni (LN).

Algoritmide treenimise ja testimise peamised etapid olid järgmised:

- Treeningbaasi laadimine R-i, parameetrite andmetüüpide ja treenimiseks kasutatavate parameetrite määramine.
- Kõigi algoritmide treenimine, vajadusel sisendite korrigeerimine.
- Valideerimisbaasi (või testbaasi) laadimine R-i ning algoritmide tulemuslikkuse hindamine (eksimismaatriks, ROC jne).
- Valideerimisbaasi (või testbaasi) kõigi algoritmide järgi iga tehingu skoorimine (hinnangu andmine pettuse tõenäosusele) ning kuluefektiivse meetodi põhjal tõenäosuste korrigeerimine. Lisaks arvutati iga mudeli jaoks eraldi kulupõhine tulem.
- Täiendavate statistiliste mõõdikute arvutamine ja tulemuste võrdlustabelite koostamine.

Sarnane protsess kordub erinevate pettuste osakaaludega baaside valikul (5 baasi korda 6 mudelit), parameetrite testimisel (8 parameetrite gruppi pluss 4 parameetrite gruppide kombinatsiooni korda 6 mudelit) ja lõplike mudelite treenimisel ning testimisel.

4.2 Eksperimendi tulemused

Eksperimendi tulemused koosnevad järgmistest alamosadest:

- Erinevate pettuste osakaaludega baasidest algoritmide treenimiseks parima baasi valimine
- Parimate parameetrite valik
- Parimate algoritmide valik
- Kulupõhiste meetodite mõju hindamine
- Mudelite kombineerimisvõimaluste hindamine

4.2.1 Erinevate treeningbaaside võrdlus

Algbaaside võrdluse aluseks on sääst protsentides võrreldes pettuste kogukahjuga tava kulumudeli ja kuluefektiivse mudeli puhul ning F_1 ja F_2 skoor. Kuna peamine fookus on tuvastada võimalikult palju pettusi ja säästa seejuures rahaliselt maksimaalsel määral, siis on kõige prioriteetsemad F_2 skoor ja kulumudelitega saavutatav sääst.

Sääst tava kulumudeli puhul on ülekaalukalt suurim 50% baasi puhul, järgnevad 20% ja 10% baasid. Sellest madalamate osakaaludega baasid annavad juba kordades halvema tulemuse (vt Tabel 6).

Tabel 6. Sääst protsentides tavamudeli kasutamisel.

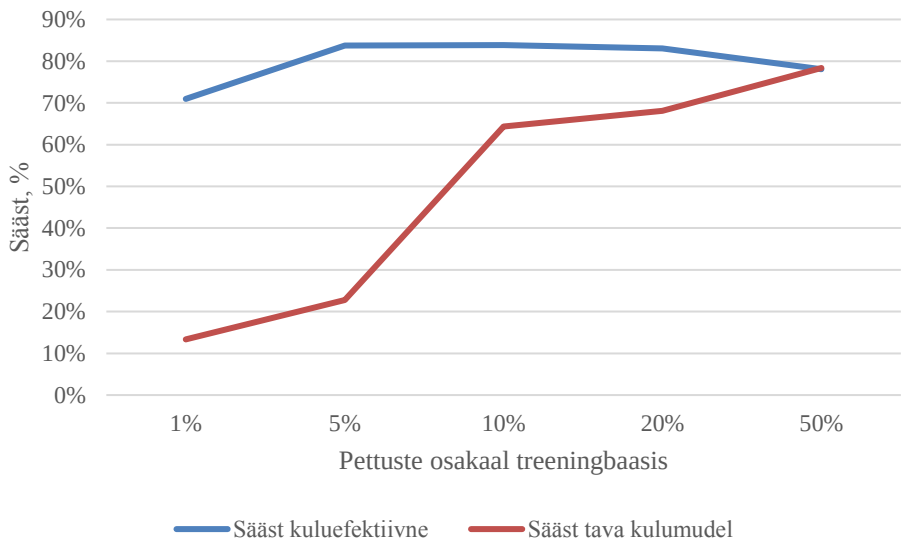
	1%	5%	10%	20%	50%
DT	10.2%	22.7%	81.9%	83.1%	79.7%
RF	0.0%	0.3%	68.5%	71.7%	88.0%
BOOST	10.7%	16.4%	73.1%	82.3%	86.4%
SVM	4.7%	44.1%	44.0%	47.6%	53.2%
NN	-	-1.6%	60.5%	64.2%	80.8%
LN	41.1%	54.8%	58.0%	59.6%	82.2%

Kuluefektiivse mudeli kasutamisel annab parima tulemuse 10% baas, 20% baasi puhul annavad kõik mudelid (va juhuslik mets) peaaegu samal tasemel hea tulemuse (otsustuspuu isegi parema). 5% baasi puhul on tulemused ebaühtlasemad. Halvimaid tulemusi saavutatakse 1% baasi puhul, samuti on võrdlemisi kesised 50% baasi tulemused (vt Tabel 7).

Tabel 7. Sääst protsentides kuluefektiivse mudeli kasutamisel.

	1%	5%	10%	20%	50%
DT	60.6%	74.7%	80.3%	84.3%	76.8%
RF	85.3%	84.9%	80.2%	75.6%	61.3%
BOOST	53.0%	88.4%	88.3%	86.5%	84.9%
SVM	84.1%	84.7%	86.4%	84.5%	79.9%
NN	-	-	84.7%	83.9%	83.6%
LN	71.8%	86.0%	83.2%	83.6%	82.1%

Kui kõrvutada tulemusi mudelite üleselt, siis kulumudeli korral on keskmine sääst 1-20% baaside puhul oluliselt suurem kui tava kulumudeli puhul. Samuti on tulemus peaaegu ühtlasel tasemel 80% juures 5%, 10% ja 20% baaside puhul, langedes mõnevõrra 50% baasi osas. Tava kulumudeliga aga saavutatav sääst järjest suureneb, mida suurem on pettuste osakaal treeningbaasis (vt Joonis 8).



Joonis 8. Tava kulumudeli ja kuluefektiivse mudeli keskmise säästu võrdlus erinevate pettuste osakaaludega treeningbaaside korral.

Tava kulumudeli järgi leitud tulemused on sarnased Bahnseni [5] uurimistöö tulemustega, kus tuvastati lineaarne seos pettuste osakaalu ja kulude vähenemise vahel, kus kõige kõrgemad kulud olid 1% baasi puhul ja madalaimad 50% baasi puhul. Seejuures erinevused 20% ja 50% pettuste osakaaludega baaside vahel olid oluliselt väiksemad kui 1% ja 5% baaside vahel. Käesolevas töös on küll mõõdetud tulemusi saavutatud säästu järgi, kuid need on otseselt seotud kuludega.

F_1 skoori puhul on parim 10% baas, 20% keskpäraselt stabiilne ja 50% baas kõikuvate tulemustega. Taaskord on halvimad tulemused 1% ja 5% baaside puhul (vt Tabel 8).

Tabel 8. F_1 skoori järgi tulemused.

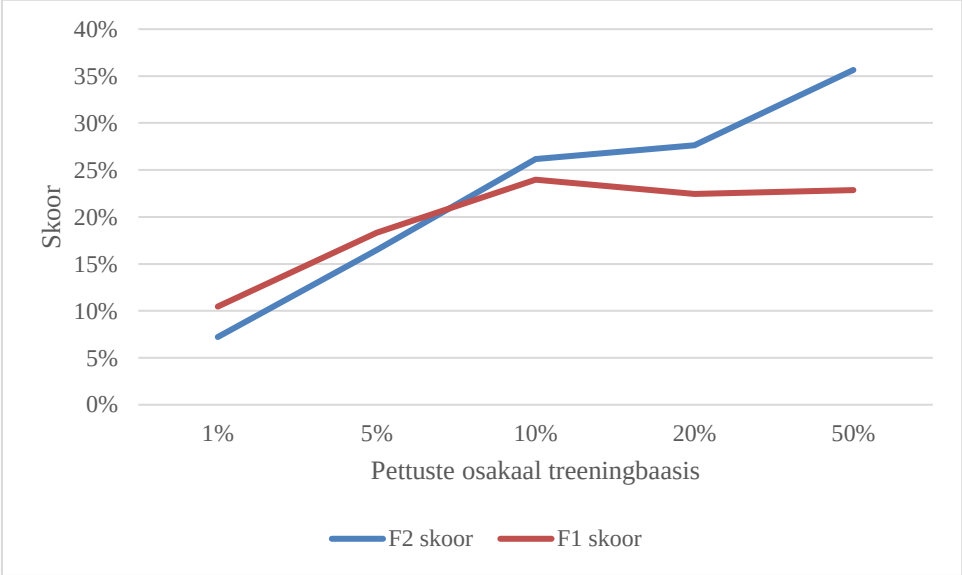
	1%	**5%**	**10%**	**20%**	**50%**
DT	13.3%	24.9%	27.2%	21.1%	18.8%
RF		3.3%	24.8%	25.7%	28.3%
BOOST	12.1%	15.9%	26.7%	21.7%	24.8%
SVM	5.4%	28.0%	27.3%	25.4%	24.7%
NN			18.0%	21.4%	17.4%
LN	11.1%	19.4%	19.8%	19.3%	22.9%

F_2 skoori puhul on ülekaalukalt parim 50% baas, 20% baas stabiilne aga märkimisväärselt halvem (vt Tabel 9). Sarnaselt eelnevate tulemustega on kõige madalamad skoorid 1-5% pettuste osakaaludega treeningbaaside korral.

Tabel 9. F_2 skoori järgi tulemused.

	1%	5%	10%	20%	50%
DT	9.2%	22.1%	33.1%	26.5%	33.4%
RF		2.3%	22.1%	28.0%	41.0%
BOOST	8.2%	13.3%	27.3%	25.4%	38.0%
SVM	3.5%	24.4%	30.5%	33.0%	37.9%
NN			20.5%	28.1%	27.7%
LN	8.0%	20.4%	23.5%	24.8%	35.9%

Kui võrrelda keskmisi skoore, siis F_1 skoor saavutab kõrgeima taseme 10% baasi juures ja peale seda mõnevõrra langeb, F_2 skoor aga paraneb järjepidevalt pettuste osakaalu kasvades ja on kõrgeimal tasemel 50% baasi korral (vt Joonis 9).



Joonis 9. Keskmiste F_1 ja F_2 skooride võrdlus erinevate pettuste osakaaludega treeningbaaside korral.

Kui võrrelda saadud tulemust Bahnseni [5] F_1 skoori põhjal tehtud võrdlusega, siis võib ka siin leida sarnasusi. Nimelt F_1 skoori järgi saavutati parim tulemus samuti 10% baasi juures, kuid edasise pettuste osakaalu kasvuga hakkas skoor kukkuma, mitte ei saavutanud platood nagu käesolevas töös. Kui võrrelda skooride väärtusi, siis käesoleva töö F_1 skoorid on Bahnseni saavutatud tulemustest ca 10% võrra kõrgemad. F_2 skoori nimetatud töös ei kasutatud.

Kokkuvõetult võib järeldada, et kindlasti saab välistada 1% ja 5% baasid, sest nende puhul olid tulemused läbi kõigi näitajate kõige madalamad. 10% baas võidab marginaalse vahega kuluefektiivse mudeli ning F_1 skoori puhul.

50% baas annab küll tava kulumudeli ja F_2 skoori alusel parimaid tulemusi, kuid kuluefektiivse mudeli järgi on tulemused halvemad, kui madalamate osakaaludega baaside korral.

Erinevate näitajate puhul paistab 20% osakaaluga baas silma oma stabiilselt hea taseme poolest. Samuti jääb see baas vaid marginaalselt parimale alla prioriteetse kuluefektiivse meetodi puhul. Seega võib 20% baasi pidada antud eksperimendi kompromissvõitjaks.

4.2.2 Parameetrite võrdlus

Parameetrite võrdlusel on treeningandmetesse alati sisendina kaasatud tehinguinfo ning sellele on eraldi juurde lisatud mingi kindel parameetrite grupp. Parameetrite võrdlusel on vaadeldud kõigepealt saavutatud säästu tava kulumudeli puhul (vt Tabel 10). Ainult tehinguinfo puhul on saavutatud sääst küllaltki madal, kõigest keskmiselt 24,1%. Kui lisati profiili info, siis see parandas oluliselt otsustuspuu tulemust (64,5%-ni), profiili info üllatuslikult halvendab aga RF, *boost* ja LN tulemust.

Autentimisinfo lisamine omab väga suurt positiivset mõju. Nelja mudeli puhul kerkib sääst üle 90%, parima tulemuse annab *boost* algoritm 96,3%. RF puhul toimub aga hoopis tulemuse halvenemine.

Tabel 10. Sääst protsentides tava kulumudeli kasutamisel.

	T	**TP**	**T auth**	**teh 1_30**	**CNP**	**prev teh**	**24h sama**	**keskm V**
DT	39.8%	64.5%	95.8%	44.6%	50.4%	42.7%	63.2%	39.9%
RF	26.3%	9.0%	16.2%	67.1%	46.7%	1.0%	54.5%	50.5%
BOOST	14.9%	7.2%	96.3%	45.5%	45.6%	2.9%	39.7%	15.7%
SVM	6.4%	27.6%	39.8%	9.9%	38.0%	39.8%	55.9%	35.1%
NN	21.2%	33.7%	91.8%	-4.3%	78.2%	-29.8%	32.5%	73.3%
LN	36.1%	27.0%	94.9%	65.9%	72.4%	39.5%	69.7%	69.3%
Keskmine	**24.1%**	**28.2%**	**72.5%**	**38.1%**	**55.2%**	**16.0%**	**52.6%**	**47.3%**

Lühendite tähendused: **T** – ainult tehingupõhine info, **TP** – tehingu ja kliendiprofiili info, **T auth** – tehingu ning kaarditehingu teostanud terminali ja kliendi/kaardi tuvastuse info, **teh 1 30** – tehingu ja eelnenud tehingute agregeeritud summad perioodidel 1 kuni 30 päeva, **CNP** – tehingu ja viimaste kaardi kohaloluta tehingute info, **prev teh** – tehingu ja eelneva tehingu detailinfo, **24h sama** – tehingu ja viimase 24 tunni jooksul tehtud sarnased tehingud, **keskm V** – tehingu ja viimase 30 päeva keskmiste tehingumahtude info

Viimase 30 päeva agregeeritud tehingud aga parandavad oluliselt just RF tulemust. NN puhul muutub sääst aga hoopis negatiivseks. Agregeeritud tehingud paistavad positiivselt mõjuvat just nendele algoritmidele, millele oli profiili infol negatiivne mõju (RF, *boost*, LN). RF ja LN puhul saavutatud üle 65% kulusääst on vägagi arvestatav.

CNP (kaardi kohaloluta) tehingute parameetrite grupp mõjub kõigi algoritmide puhul positiivselt ja keskmine sääst ulatub 55,2%-ni. Suurim positiivne mõju on NN algoritmile (78,2%), samuti LN algoritmile (72,4%), teistel juhtudel jääb tulemus 40-50% vahemikku.

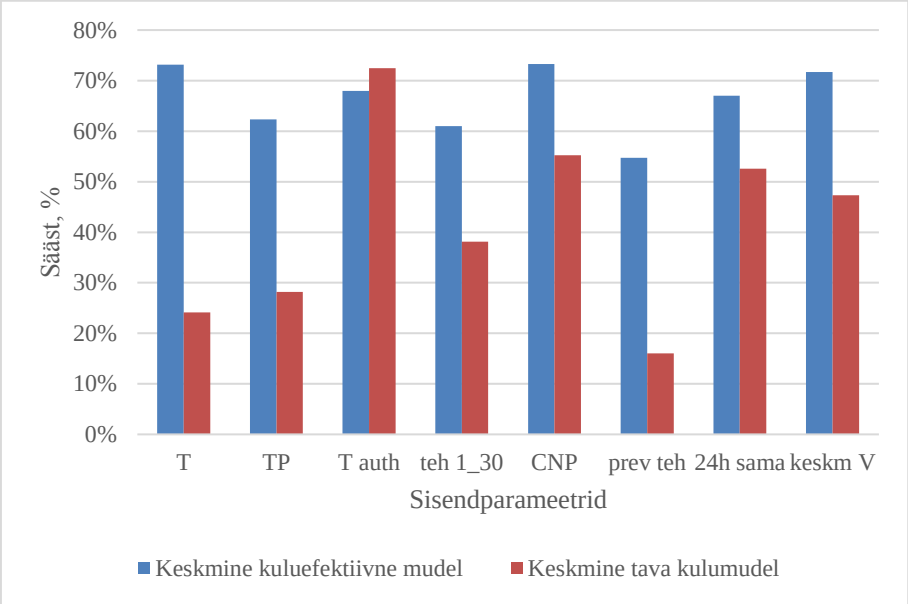
Eelmise tehingu info omab vastuolulist mõju, olles oluline näitaja ehk vaid SVM mudeli puhul. NN puhul tekib aga lausa -29,8% negatiivne sääst, RF ja *boost* puhul nullitakse põhimõtteliselt kogu mudeli tulemus. Seega antud näitaja tundub algoritmide tööd pigem segavat, kui tulemuse paranemisele kaasa aitavat. Selline tulemus on mõneti isegi ootuspärane, sest algbaasis on sellisel juhul nii vaadeldava tehingu kui lihtsalt sellele eelnenud tehingu detailinfo ja algoritm ei suuda tegelikult eristada kummale keskenduda. Võib juhtuda, et aluseks võetakse eelkõige just eelmine tehing, kuid paljudel juhtudel on see täiesti legaalne tehing.

Viimase 24 tunni sarnaste tehingute info on suhteliselt ühtlaselt oluline näitaja, pakkudes keskmiselt 52,6% suurust säästu ning omades üle 60% mõju DT ja LN puhul. SVM algoritmi puhul on see kõige olulisem parameetrite grupp (55,9%), samas kui SVM algoritm jääb üldiselt teistele alla.

Keskmised välismaa tehingud omavad NN ja LN puhul ca 70% mõju, mis on vägagi arvestatav. RF puhul saavutatakse 50% sääst, SVM puhul 35% ning DT ja *boost* puhul see tulemuse paranemist ei too, kui võrrelda seda ainult tehinguinfoga treenitud algoritmidega.

Kuluefektiivse mudeli kasutamine toob pea kõigil juhtudel oluliselt suurema keskmise säästu. Juba baasvariandi puhul, kui aluseks on võetud vaid tehingupõhine info, on kulumudeli sääst keskmiselt 73,1% võrreldes tavamudeli 24,1% (vt Joonis 10). Põhjus peitub suuresti selles, et kuluefektiivne mudel määrab summast sõltuvalt rohkem tehinguid petturlikeks ning hoiab sellega suurema tõenäosusega ära ka suuremad kahjumid. Kulusäästlik mudel määras keskmiselt 1655 tehingut petturlikuks (ca 8,3%), samal ajal kui tavamudel määras keskmiselt 631 tehingut petturlikuks (ca 3,2%). Kui

tegelik pettuste osakaal on alla ühe protsendi, siis tuleb arvestada, et petturlikuks määratud tehingute mahtu tuleb piirata nii, et reaalselt jätkuks ressurssi tehingute kontrolliks.



Joonis 10. Keskmised kulusäästud parameetrite gruppide lõikes tava kulumudeli ja kuluefektiivse mudeli puhul.

Mitte ühegi algoritmi puhul ei too profiili info lisamine kaasa täiendavat kulusäästu. Teatud juhtudel toimub isegi märgatav halvenemine, nt LN ja NN algoritmide puhul (vt Tabel 11).

Autentimise info omab ka kuluefektiivse mudeli puhul väga suurt mõju. Nelja algoritmi (DT, *boost*, NN ja LN) puhul saavutatakse üle 95% kulusääst. Samas on see ligikaudu samal tasemel kui tavamudeli puhul. Eraldi tasub välja tuua, et nende algoritmide puhul märgiti ka kõige vähem tehinguid petturlikuks, keskmiselt ca 240 tehingut, mis on sarnasel tasemel tavamudeliga. Üllatuslikult mõjus aga autentimise info RF ja SVM algoritmidele väga negatiivselt.

Viimase 30 päeva tehingute info parandab vaid *boost* ja LN algoritmide tulemust ning sedagi vaid vastavalt 5,1% ja 3,0% võrra. Huvitaval kombel NN algoritmi kulusääst muutub täiesti olematuks ehk see parameetrite grupp NN algoritmi puhul kindlasti ei sobi.

CNP tehingute info on olulise tähtsusega, kuid kui tavamudeli puhul andis CNP tehingute info märkimisväärse kulusäästu kasvu, siis kuluefektiivse mudeli puhul on täiendav mõju

väike. *Boost* ja LN algoritmide kulusääst kasvas vastavalt 6,9% ning 3,8% võrra. Võib siiski öelda, et tulemused on ühtlaselt head.

Eelmise tehingu info halvendab pea kõigi algoritmide (va DT) tulemust. Olukord on siiski märgatavalt parem kui tavamudeli puhul.

Viimase 24 tunni sarnased tehingud parandavad *boost* algoritmi tulemust 5,6% võrra, LN tulemust 3,7% võrra ja DT 1,4% võrra. NN algoritmi puhul tekkis aga -44,9% negatiivne efekt, millist tavamudeli puhul ei tekkinud.

Keskmised välismaa tehingud annavad kõikide algoritmide puhul suhteliselt stabiilse tulemuse, märkimisväärset paranemist kulusäästule aga ei ole ühelgi juhul.

Tabel 11. Sääst protsentides kuluefektiivse mudeli kasutamisel.

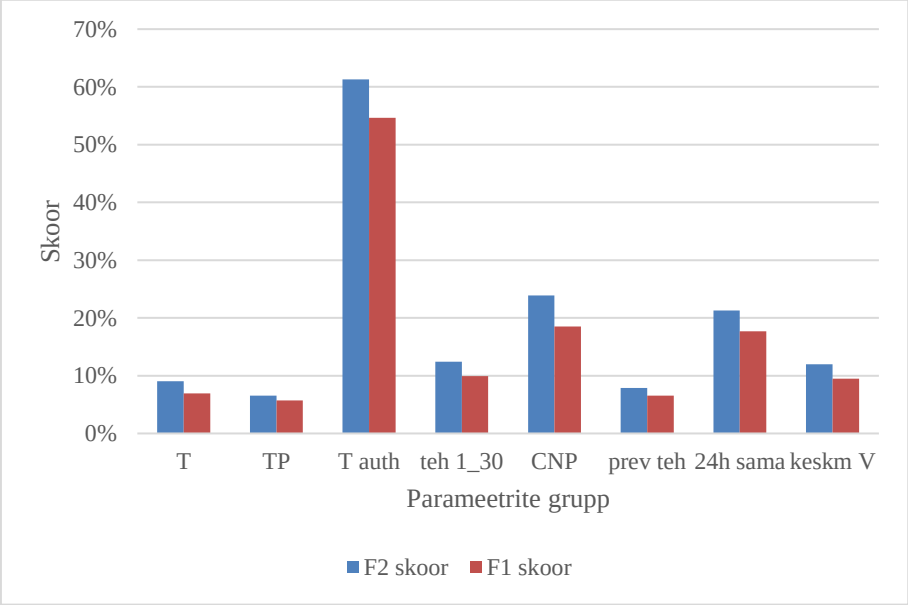
	T	**TP**	**T auth**	**teh 1_30**	**CNP**	**prev teh**	**24h sama**	**keskm V**
DT	71.1%	71.0%	96.1%	71.7%	66.7%	72.3%	72.6%	69.9%
RF	66.7%	59.6%	-0.1%	62.6%	64.4%	65.2%	65.0%	56.1%
BOOST	75.5%	72.7%	96.6%	80.6%	82.3%	74.2%	81.1%	78.6%
SVM	76.8%	73.4%	23.2%	76.6%	76.8%	75.8%	75.8%	75.4%
NN	77.8%	58.0%	96.4%	0.6%	74.8%	-21.9%	32.8%	77.5%
LN	70.9%	39.2%	95.5%	74.0%	74.7%	62.7%	74.7%	72.7%
Keskmine	**73.1%**	**62.3%**	**68.0%**	**61.0%**	**73.3%**	**54.7%**	**67.0%**	**71.7%**

Parameetrite võrdluse põhjal võib järeldada, et võrreldes tavamudeliga parandab kulumudeli rakendamine tulemusi kolmekordselt ning parameetrite gruppide lisamine omab oluliselt väiksemat mõju (vt Tabel 12). Suuremat kulusäästu annavad samad parameetrite grupid, mis tavamudeli puhulgi ehk ülekaalukalt olulisim on autentimisinfo, järgnevad CNP tehing, keskmised välismaised tehingud ja viimase 24 tunni sarnased tehingud.

Tabel 12. Täiendav sääst kulumudeli kasutamisel (võrreldes ainult tehinguinfo põhjal koostatud mudeliga).

	TP	T auth	teh 1_30	CNP	prev teh	24h sama	keskm V
DT	-0.2%	25.0%	0.6%	-4.5%	1.2%	1.4%	-1.2%
RF	-7.1%	-66.8%	-4.1%	-2.3%	-1.4%	-1.7%	-10.5%
BOOST	-2.8%	21.1%	5.1%	6.9%	-1.3%	5.6%	3.1%
SVM	-3.4%	-53.6%	-0.2%	0.0%	-1.0%	-1.1%	-1.4%
NN	-19.8%	18.6%	-77.2%	-3.0%	-99.7%	-44.9%	-0.3%
LN	-31.8%	24.6%	3.0%	3.8%	-8.2%	3.7%	1.8%
Keskmine	**-10.8%**	**-5.2%**	**-12.1%**	**0.1%**	**-18.4%**	**-6.2%**	**-1.4%**

Parameetrite gruppide võrdlemisel tuleb kindlasti tähelepanu pöörata ka klassifitseerimise täpsusele. Mudelite eesmärgiks on eelkõige tuvastada võimalikult täpselt petturlikud tehingud, mida mõõdab F_2 skoor. Nii F_1 kui F_2 skoorid muutuvad väga sarnaselt läbi erinevate parameetrite gruppide ja algoritmide. Läbivalt on F_2 skoor kõrgemal tasemel, keskmiselt 3,1%. (vt Joonis 11)



Joonis 11. Keskmised F_1 ja F_2 skoorid parameetrite gruppide lõikes.

Ainult tehinguinfot arvesse võttes on skoorid väga madalad, jäädes alla 10% taseme. Seejuures võib välja tuua, et pea kõik parameetrite grupid, välja arvatud profiil ja eelmine tehing, parandavad tulemust, eriti F_2 skoori järgi mõõdetuna.

F_1 skoori järgi on tulemused toodud Tabel 13.

Tabel 13. Parameetride võrdlus F_1 skoori põhjal.

	T	TP	T auth	teh 1_30	CNP	prev teh	24h sama	keskm V
DT	9.5%	8.0%	79.6%	10.0%	17.1%	10.5%	8.3%	9.6%
RF	9.9%	7.4%	1.6%	14.2%	17.2%	0.9%	24.1%	9.6%
BOOST	5.8%	6.8%	86.7%	16.4%	14.0%	7.0%	14.9%	7.3%
SVM	6.7%	6.7%	7.1%	6.3%	19.0%	12.1%	20.0%	11.3%
NN	4.8%	2.9%	83.5%	4.8%	23.1%	2.0%	22.2%	10.8%
LN	4.8%	2.7%	69.3%	7.9%	20.5%	6.7%	16.5%	8.2%
Keskmine	**6.9%**	**5.7%**	**54.6%**	**9.9%**	**18.5%**	**6.5%**	**17.7%**	**9.5%**

Nii F_1 kui F_2 skoori järgi on tulemused sarnased (vt Tabel 14). Kõige märkimisväärsem mõju on autentimisinfol, mille tulemusena tõuseb keskmine F_2 skoor üle 60%. Kui välja arvata RF ja SVM algoritmid, siis teiste algoritmide puhul ulatuvad F_2 skoorid lausa 80-90% vahemikku. Arvestatav kasv on ka CNP ja 24h sarnase tehingu parameetrite puhul.

Tabel 14. Parameetrite võrdlus F_2 skoori põhjal.

	T	TP	T auth	teh 1_30	CNP	prev teh	24h sama	keskm V
DT	13.6%	11.2%	87.8%	15.5%	25.6%	16.2%	12.7%	13.7%
RF	11.5%	6.3%	2.9%	14.3%	17.5%	0.6%	26.4%	9.6%
BOOST	6.8%	7.3%	90.8%	17.7%	16.0%	5.5%	16.4%	7.8%
SVM	7.5%	6.4%	15.9%	6.5%	20.7%	12.9%	21.4%	12.7%
NN	7.4%	4.1%	88.3%	9.3%	33.6%	4.1%	27.4%	15.7%
LN	7.4%	4.0%	82.1%	11.4%	29.9%	8.0%	23.4%	12.5%
Keskmine	**9.0%**	**6.5%**	**61.3%**	**12.4%**	**23.9%**	**7.9%**	**21.3%**	**12.0%**

F_1 ja F_2 skoorid kinnitavad samu järeldusi, mis sai tehtud kulumudelite puhul. Skooride puhul on parameetritest tulenev paranemine pea igal juhul märkimisväärne, isegi kui kuluefektiivse mudeli puhul olid tulemuste paranemised väheolulised.

Eraldi tasub analüüsida parameetrite omavahelise kombineerimise mõju tulemustele. Eelnevad testid näitasid, et kõik parameetrid ei pruugi mudelite täpsusele kasuks tulla. Seejuures töötasid erinevad parameetrid eri algoritmidega erinevalt, mis sobis ühele

algoritmile hästi ei pruukinud teise puhul tulemust parandada. Teatud juhtudel uue parameetri lisamine hoopis halvendas väga olulisel määral mudeli tulemuslikkust.

Järgnevalt on kaasatud erinevaid eelmises testis hästi töötanud parameetrite gruppe, et välja selgitada, kas nende omavaheline kombineerimine võib anda täiendava positiivse efekti.

Esimesel juhul on kasutatud sisendiks kogu andmebaasi koos kõigi parameetrite gruppidega. *Boost*, DT ja LN algoritmide puhul andis see kokkuvõttes väga häid tulemusi, kus sääst kulumudeli puhul ulatus vastavalt 96,6%, 96,1% ja 91,5% (vt Tabel 15). Seejuures oli pettusteks määratud tehingute arv nende mudelite puhul madal, varieerudes 200 ja 400 vahel. Kõigi parameetrite puhul ei töötanud aga SVM, RF ja NN algoritmid. Nende algoritmide puhul tuleks kindlasti eemaldada need parameetrid, mis eraldi testides tulemusi halvendasid.

Tabel 15. Parameetrite kombineerimise tulemused kuluefektiivse mudeli puhul.

	all	**A_C_24**	**CNP_24h**	**1_30_keskm**
DT	96.1%	96.1%	68.2%	71.7%
RF	7.9%	0.5%	64.8%	62.2%
BOOST	96.6%	96.7%	83.2%	82.4%
SVM	21.0%	10.9%	77.1%	76.1%
NN	-4.3%	91.0%	4.6%	59.1%
LN	91.5%	93.5%	76.1%	73.9%
Keskmine	**51.5%**	**64.8%**	**62.3%**	**70.9%**

Lühendite tähendused: **all** – kaasatud kõik parameetrite grupid, **A C 24** – tehingu, autentimisinfo, CNP ja 24 tunni sarnaste tehingute parameetrid, **CNP 24h** – tehingu, CNP ja 24 tunni sarnaste tehingute parameetrid, **1 30 keskm** – tehingu, 1-30 päeva agregeeritud tehingute ja viimase 30 päeva keskmiste tehingute parameetrid

Nelja algoritmi puhul saavutati suurim kulusääst, kui kombineeritud olid tehinguinfo, autentimisinfo, CNP tehingu info ja viimase 24h sarnaste tehingute info. Sellisel juhul olid välistatud kliendi profiili, kuni 30 päeva agregeeritud tehingud, eelmise tehingu detailinfo ja keskmised varasemad välismaa tehingud. Kõige parem tulemus oli *boost* algoritmil 96,7% kulusäästuga, järgnesid DT 96,1%, LN 93,5% ja NN 91%. RF ja SVM algoritmide puhul selline kombinatsioon aga üldse ei töötanud.

Kui lisaks tehinguinfole kaasati ainult CNP ja viimase 24h tehingud, siis tulemused jäid eelnevale oluliselt alla. Erandiks oli aga SVM mudel, mis saavutas just sellise kombinatsiooni korral oma parima tulemuse 77,1%. Sarnane tulemus saavutati ka 1-30 päeva tehingute ja keskmiste välismaiste tehingute kombineerimisel. Kulusääst oli suhteliselt ühtlaselt 60-80% vahel ning väga suuri äärmusi ei esinenud.

Järgnevalt on toodud täiendav sääst võrreldes mudelitega, mis olid koostatud ainult tehinguinfot arvesse võttes (vt Tabel 16). Selle põhjal on selgelt näha, et kolmel juhul kõigi parameetrite kaasamine tugevalt parandas (20-25%) kulusäästu, kuid samal ajal kolme algoritmi puhul veel suuremal määral kahandas tulemust (-55% kuni -80%). Autentimis-, CNP ja 24h tehingute info andis sarnase tulemuse, kuid NN puhul oli see kõige parem kombinatsioon. Teised kaks parameetrite kombinatsiooni enamus juhtudel tulemusi märgatavalt ei muutnud, välja arvatud *boost* ja LN algoritmid, mille kulusääst suurenes vastavalt 7-8% ja 3-5%.

Tabel 16. Täiendav sääst kuluefektiivse mudeli kasutamisest % (võrdluseks ainult tehingupõhise mudeliga).

	all	**A_C_24**	**CNP_24h**	**1_30_keskm**
DT	25.0%	25.0%	-2.9%	0.6%
RF	-58.8%	-66.2%	-1.9%	-4.4%
BOOST	21.1%	21.2%	7.7%	6.9%
SVM	-55.8%	-66.0%	0.3%	-0.8%
NN	-82.1%	13.2%	-73.2%	-18.7%
LN	20.6%	22.6%	5.1%	3.0%
Keskmine	**-21.7%**	**-8.4%**	**-10.8%**	**-2.3%**

Klassifitseerimise täpsus F_2 skoori järgi andis sarnase tulemuse (vt Tabel 17). Parimatel juhtudel ulatusid F_2 skoorid 90% taseme juurde ning parimad algoritmid olid *boost*, DT ja LN. Küll aga on erinevused parimate ja halvimate tulemuste vahel F_2 skoori järgi keskmiselt suuremad.

Tabel 17. F2 skoori tulemused parameetrite kombinatsioonide korral.

	all	A_C_24	CNP_24h	1_30_keskm
DT	87.8%	87.8%	18.8%	15.5%
RF	3.9%	14.2%	31.7%	13.1%
BOOST	90.9%	92.2%	21.5%	17.6%
SVM	14.2%	10.0%	25.9%	6.2%
NN	1.9%	60.3%	25.6%	20.0%
LN	72.1%	75.5%	29.8%	11.3%
Keskmine	**45.1%**	**56.7%**	**25.5%**	**13.9%**

Võrreldes ainult tehinguinfo põhjal koostatud mudelitega andsid esimesed kaks parameetrite kombinatsiooni enamus juhtudel kõige suurema täpsuse suurenemise (vt Tabel 18). Kolmanda kombinatsiooni puhul oli võit 15-20% ning viimase kombinatsiooni puhul tulemused oluliselt ei paranenud.

Tabel 18. F2 skoori täiendav paranemine (võrreldes ainult tehinguinfo põhjal koostatud mudeliga).

	all	A_C_24	CNP_24h	1_30_keskm
DT	74.2%	74.2%	5.1%	1.9%
RF	-7.6%	2.7%	20.2%	1.6%
BOOST	84.1%	85.4%	14.7%	10.8%
SVM	6.7%	2.5%	18.4%	-1.3%
NN	-5.5%	52.9%	18.2%	12.6%
LN	64.7%	68.1%	22.4%	3.9%
Keskmine	**36.1%**	**47.6%**	**16.5%**	**4.9%**

Kokkuvõtlikult võib järeldada, et parameetrite valikul on suur roll sellel, millist algoritmi kasutatakse. Parameetrid, mis sobivad ühe algoritmi puhul ei pruugi sobida teisega. Kõige suurema efekti enamus mudelite puhul andis autentimisinfo parameetrite grupp, kus nelja algoritmi (DT, *boost*, NN, LN) puhul saavutati kuluefektiivse mudeliga üle 95% sääst. Samas autentimisinfo omas lausa negatiivset mõju RF ja SVM algoritmidele. Enamus juhtudel omasid arvestatavat positiivset mõju kaardi kohaloluta (CNP) tehingud ja viimase 24 tunni sarnased tehingud. Lihtsalt varasemate tehingute agregeerimine kuni 30 päeva tagasi ja keskmiste tehingumahtude lisamine olulist mõju ei avaldanud. Parameetritest üldse ei toiminud kliendi profiili info ning eelmise tehingu detailinfo. Need

näitajad tunduvad algoritmide tööd pigem segavat, kui tulemuse paranemisele kaasa aitavat.

Kui Hand [15] ja Bahnsen [4] tõid välja varasemate tehingute üldise agregeerimise positiivse mõju algoritmide tulemuslikkusele, siis antud töös oli näha, et kõigi tehingute agregeerimise asemel toimib palju efektiivsemalt eritüübiliste tehingute eristamine ning nende eraldi agregeerimine (nt CNP ja viimase 24 tunni sarnased tehingud).

4.2.3 Algoritmide hindamine

Algoritmide lõplik hindamine toimus 2017. aasta esimese kahe kuu tehingutest moodustatud testbaasi põhjal. Pettuste osakaal on sellel baasil reaalelule vastav ehk tuntavalt all 1%. Sellisel baasil testimise põhjal on võimalik teha järeldusi, kas algoritmid on võimelised töötama reaalelulistel andmetel. Parim algoritm peab olema täpne klassifitseerimises, pakkuma suurimat kulusäästu ja suutma tekitada võimalikult vähe valehäireid.

Algoritmide võrdlus statistiliste näitajate baasil on toodud Tabel 19. *Recall* määr on oluline kuna näitab palju pettusi algoritmi poolt tuvastati. Tulemused on siinkohal väga tasavägised kõikudes 71,8% ja 73,6% vahel. Kõige parema tulemuseni jõudis SVM algoritm, järgnesid otsustuspuu, logistiline regressioon, *boost*, juhuslik mets ja madalaima tulemusega närvivõrgud.

Tabel 19. Algrotimide võrdluse statistilised näitajad.

	DT	**RF**	**BOOST**	**SVM**	**NN**	**LN**
Recall	73.2%	71.8%	72.9%	73.6%	70.8%	73.2%
Täpsus	18.7%	57.1%	32.8%	22.4%	27.1%	13.0%
F1 skoor	29.8%	63.7%	45.2%	34.4%	39.2%	22.0%
F2 skoor	46.3%	68.3%	58.5%	50.6%	53.5%	37.9%
AUC	85.8%	98.8%	85.4%	92.5%	85.6%	85.5%

Täpsuse põhjal, mis arvestab ka valesti pettuseks määratud tehingute arvu, saavutas parima tulemuse juhuslik mets 57,1%, millele järgnes *boost* 32,8% ja närvivõrgud 27,1%. Halvima tulemusega (13,0%) oli logistiline regressioon. Mida madalam on see näitaja, seda rohkem on vaja teha „tühja tööd“ ja kontrollida tehinguid, mis ei ole tegelikult pettused.

F_2 skoor on kõigil mudelitel arvestatavalt kõrgem (4,7-16,5%) kui F_1 skoor ehk algoritmid suudavad paremini tuvastada just petturlikke tehinguid. Parim tulemus F_2 skoori järgi on juhuslikul metsal (68,3%), järgnevad *boost* (58,5%) ja närvivõrgud (53,5%). Halvim on taaskord logistiline regressioon (37,9%).

ROC kõvera järgi on täpseimad juhuslik mets ja SVM ning halvimad logistiline regressioon ja *boost*. ROC kõverad on eraldi toodud lisas 4.

Võrdluse kõige tähtsamaks parameetriks võib siiski pidada algoritmide rakendamise tulemusena saavutatavat kulusäästu pettuste ärahoidmisest (vt Tabel 20). Parima tulemuse tava kulumudeli järgi saavutas juhuslik mets (90%), kuid kuluefektiivse mudeli järgi oli selle algoritmi tulem kõige madalam (25%). Põhjuseks on kuluefektiivse mudeli puhul väga suur pettuseks klassifitseeritud tehingute arv (10 811 tk), mis tõstis tehingute kontrollimise kulusid. Lahenduseks võiks olla tõenäosuste skaala eraldi korrigeerimine juhusliku metsa algoritmi jaoks.

Tabel 20. Algoritmide võrdlus kulumudelite järgi.

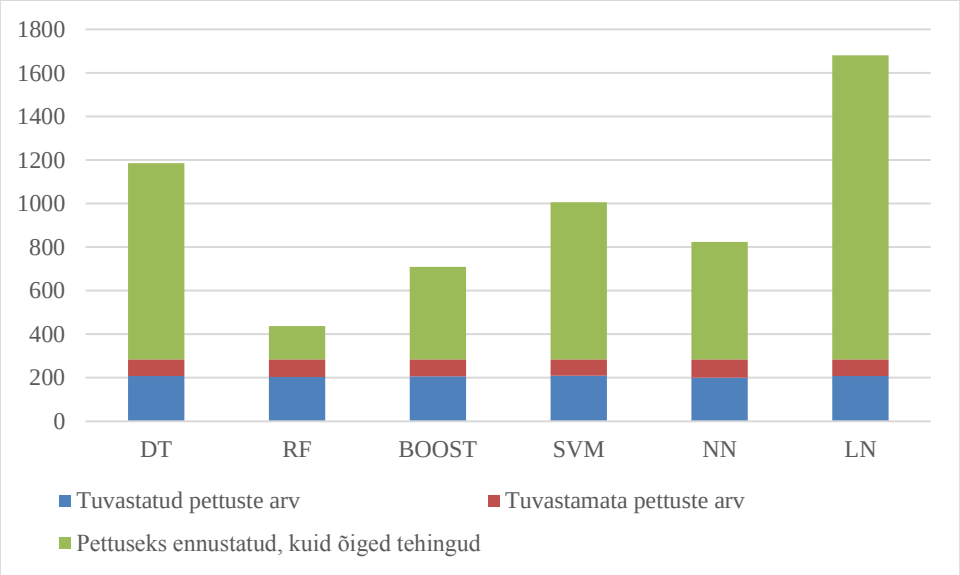
	DT	**RF**	**BOOST**	**SVM**	**NN**	**LN**
Sääst tava kulumudeli kasutamisest %	85%	90%	88%	67%	83%	82%
Sääst kuluefekt.mudel kasutamisest %	84%	25%	87%	79%	84%	83%
Pettuseks klassifits. tavamudel	1 104	356	632	563	741	1 585
Pettuseks klassifits. kuluefekt. mudel	1 304	10 811	793	2 385	787	1 433

Boost algoritm saavutas aga 88% ja 87% kulusäästu ning seejuures püsisid pettuseks klassifitseeritud tehingute arvud väga mõistlikul tasemel (vastavalt 632 ja 793). See tähendab, et tehakse mõstlikus mahus tehingute kontrolle ning saavutatakse maksimumi lähedane kulusääst pettuste ärahoidmisest.

Suhteliselt lähedase tulemuse kulusäästu mõistes saavutasid ka otsustuspuu, närvivõrkude ja logistilise regressiooni algoritmid. Neist kolmest parimaks võiks pidada närvivõrke, sest kontrollitavate tehingute maht oli sel juhul teistest arvestatavalt väiksem, mis tähendab, et tehingute kontrollid on suurema tõenäosusega ressursi mõistes reaalselt teostatavad.

Kui SVM algoritm oli klassifitseerimise täpsuse järgi ühtlaselt heal tasemel, siis kulusäästu mõttes oli see halvim. Seega libisesid algoritmi kontrollist läbi just suuremate summadega pettused ning olukorda ei parandanud ka kuluefektiivse lävendi kasutamine, sest siis tõusis pettuseks klassifitseeritud tehingute arv liiga kõrgeks.

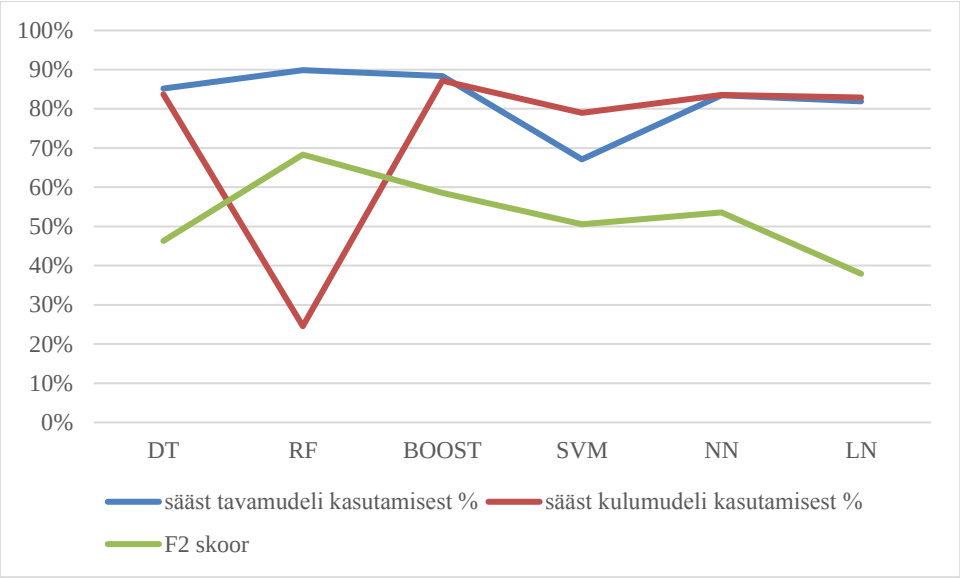
Joonis 12 kirjeldab ilmekalt, kuidas tuvastatud ja tuvastamata jäänud pettuste arvud on väga sarnasel tasemel, küll aga erineb väga oluliselt õigete tehingute maht, mis mudeli poolt määrati pettuseks. See on tehingute maht, mida peab täiendavalt üle kontrollima ning seejuures asjatult ressursse kulutama. Juhusliku metsa puhul on see teistest mudelitest kordades madalam, tehes selle mudeli tõeliselt efektiivseks.



Joonis 12. Tuvastatud ja tuvastamata pettuste ning valehäirete arv.

Kui vaadata kokkuvõtlikult tulemusi kulusäästu ja klassifitseerimise (F_2 skoori) järgi (vt Joonis 13), siis parima algoritmi valik ei ole sugugi lihtne ülesanne. Tulemused on tasavägised ning otsus sõltub valitud mõõdikust.

Nii klassifitseerimise kui saavutatud kulusäästu järgi on parimateks algoritmideks juhuslik mets, *boost* ja närvivõrgud. Nende algoritmide puhul on kõige madalamal tasemel ka valehäirete arv. Juhuslik mets on seejuures väga heal tasemel nii kulumudelite kui klassifitseerimise järgi, kuid kuluefektiivne mudel selle algoritmi ja algandmete puhul ei toimi, ilma et tõenäosuse hinnanguid eraldi ei korrigeeritaks.



Joonis 13. Kulusääst ja F2 skoor algoritmide lõikes.

Juhuslik mets oli parim ka näiteks Handi [15] uurimistöös. Kui aga Westi [25] ja Bahnseni [4] tulemused tõid välja otsustuspuu, siis antud töös oli see kulusäästu järgi küll heal tasemel, kuid klassifitseerimise ja valehäirete nõrkade tulemuste tõttu see parimate hulka ei kuulunud. West [25] ja Hand [15] tõid välja ka tugivektor-masinate algoritmi, kuid käesolevas töös jäi see teistele algoritmidele alla.

4.2.4 Kulupõhise lähenemise mõju

Kulupõhine lähenemine sobib tulemuste hindamiseks hästi, sest annab tulemustele väga selgelt arusaadava ja võrreldava hinnangu. Lõplike algoritmide testbaasi peal saavutatud keskmine kulusääst 80-90% on seejuures muljetavaldav ning annab kinnitust, et masinõppe algoritmide rakendamine kaardipettuste tuvastamisel on väga hästi toimiv ning majanduslikult mõttekas tegevus.

Tava kulumudeli ja kuluefektiivse mudeli tulemused olid mõnevõrra vastuolulised. Treeningbaaside võrdlemisel, eriti just madalamate pettuste osakaalude puhul, aitas kuluefektiivne mudel algoritmide tulemuslikkust oluliselt parandada. Samuti oli parameetrite võrdlusel kuluefektiivse mudeli poolt saavutatud kulusääst oluliselt kõrgemal tasemel võrreldes tava kulumudeliga. Kuluefektiivse mudeli rakendamine ei parandanud algoritmide tulemuslikkust lõpliku testbaasi peal. Kuluefektiivse mudeliga saavutatud tulemused olid kas samal tasemel või halvemadki võrreldes tava kulumudeliga. Kuluefektiivse mudeli mõte on eelkõige mitte läbi lasta suurema summaga

petturlikke tehinguid ning sellega ära hoida suuremad kahjud. Kui vaadata aga konkreetseid testbaasis olevaid tehinguid, siis juba tavamudel suutis kõik suuremate summadega pettused tuvastada ning seetõttu ei tulnudki kuluefektiivse mudeli eelis testbaasi peal välja.

Seega ei pea kuluefektiivse meetodi rakendamine olema ilmtingimata eesmärk omaette, sest mudeli eelise mõjule pääsemine sõltub väga palju konkreetsetest petturlikest tehingutest. Töö autori nägemuse kohaselt võiks see olla paralleelselt jooksev mudel, mida võiks rakendada täiendava vaba tehingute kontrollressursi olemasolul. Kindlasti tasuks ka edaspidi mõlema lähenemise tulemusi jälgida ning võimalik, et tehingute muutudes (nt tehingusummade kasvades) muutub selle rakendamine mõttekamaks.

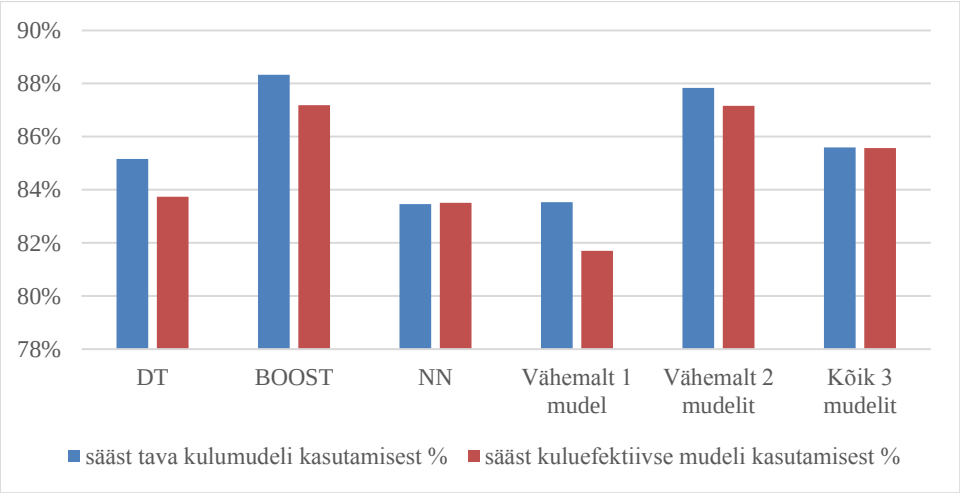
Kuluefektiivse mudeli kindlat paremust sarnaselt Bahnsen 2013. a [5] ja 2016. a [4] uurimustele käesoleva töö tulemustega kinnitada ei saa. Samas ei saa väita ka vastupidist, sest eksperimendi esimestes etappides oli kuluefektiivsel mudelil tava kulumudeliga võrreldes arvestatav eelis.

4.2.5 Algoritmide kombineerimine

Kui algoritmid olid eraldi võimelised arvestatavat kulusäästu saavutama, siis tekib küsimus, kas nende kombineerimine võiks anda veelgi parema efekti. Selleks sai läbi viidud eraldi eksperiment, kus kombineeriti kolme parima algoritmi tulemusi nii tava kulumudeli kui kuluefektiivse mudeli puhul.

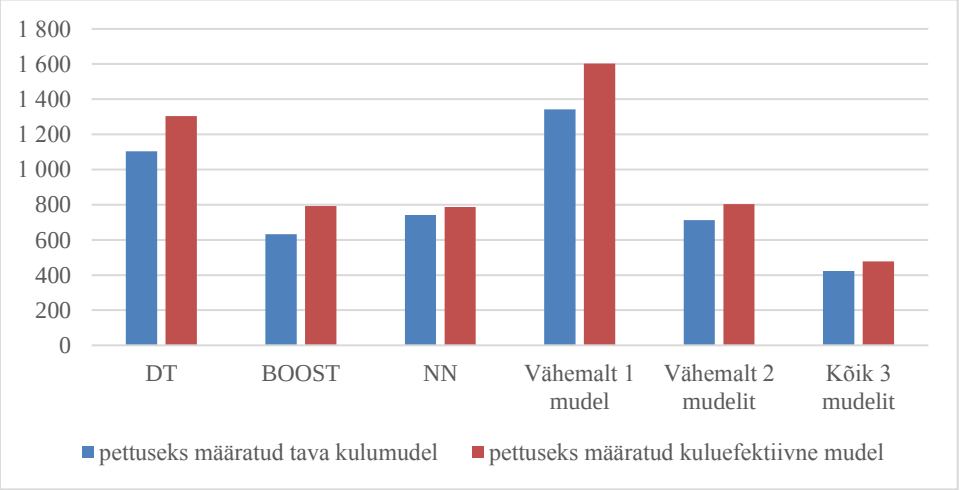
Aluseks on võetud enamuse hääletuse põhimõte. Selleks, et mingi tehing petturlikuks määrata, peaks vähemalt kaks mudelit sellise ennustuse tegema. Võrdluseks on toodud veel kaks täiendavat varianti – petturlikuks määratakse tehing kui kolmest üks algoritm sellise ennustuse teeb või kui kõik kolm algoritmi sellise ennustuse teevad.

Tulemused on näha Joonis 14, kus on koos nii algoritmide eraldiseisvad kui kombineerimise teel saavutatud kulusäästud. Algoritmide kombineerimise parimaks versiooniks osutus enamuse otsus (ehk vähemalt kahe algoritmi ennustus). Samas ei ole see parem kui parima eraldiseisva algoritmi (*boost*) poolt saavutatud kulusääst.



Joonis 14. Kulusääst mudelite kombineerimisel.

Kui vaadata juurde pettuseks määratud tehingute arvu, siis kombineeritud tulemuste korral väheneb loogiliselt pettuseks määratud tehingute arv vastavalt sellele kui mitu mudelit pidid konsensusele jõudma. Enamuse otsuse tulemus on ka siinkohal pea täpselt sama, mis parima *boos*t algoritmi oma (vt Joonis 15).



Joonis 15. Pettuseks määratud tehingute arv mudelite kombineerimisel.

Eraldi sai üle vaadatud need tehingud, mis jäid algoritmide poolt pettuseks määramata. Selle tulemusena selgus, et tuvastamata pettused kattusid kõigi algoritmide puhul ca 90% ulatuses. Seejuures olid need tehingud oma olemuselt väga sarnased, nimelt enamus neist ühes kindlas kaupmehe grupis. Treeningandmetes aga sama kaupmehe grupiga tehinguid ei olnud ning tundub, et algoritmide jaoks oli see uut tüüpi tehingu liik, mida seetõttu pettuseks määratleda ei suudetud. See toob selgelt välja vajaduse algoritmide pidevaks

uuesti treenimiseks võimalikult värske pettuste infoga, et kõik uut tüüpi pettused saaksid võimalikult kiiresti kaasatud.

Kokkuvõttes võib järeldada, et mitme mudeli kombineerimine loodetud täiendavat kulusäästu ei pakkunud. Olulisem tundub hoopis algoritmide pidev ümber treenimine ja erinevate pettuste tüüpide kaasamine treeningandmetesse. Seega pole erinevate algoritmide enamushääletuse lähenemine antud juhul põhjendatud, sest teeb süsteemi rakendamise ressursimahukamaks ja täiendavat kulusäästu ei anna.

5. Käesoleva töö peamised tulemused

Antud magistritöös andis autor tervikliku ülevaate masinõppe rakendamisest kaarditehingute pettuste tuvastamisel, püstitades eri etappides probleemid ja leides neile lahendused. Lõpptulemusena jõuti töötavate algoritmideni, mida on võimalik reaalsetes süsteemides rakendada. Sama väärtuslik on ka kogu protsessi läbimisel saadud kogemused.

Masinõppe rakendamisel mängivad väga olulist rolli konkreetsed aluseks võetud andmed, rakendatud algoritmid ja tulemuste mõõtmisel kasutatud mõõdikud. Seetõttu on raskendatud ka teiste varasemate uurimuste tulemuste automaatne ülevõtmine. Näiteks kui varasemad uurimused [4, 15, 25] tõid välja ühe parema algoritmina tugivektor-masinad, siis käesolevas töös ei pääsenud see algoritm üldse parimate hulka.

Treeningbaaside puhul tuli välja alaesindamise olulisus ning üldjoontes on tulemused sarnased ühe eeskujuna kasutatud Bahnseni [5] tööga. Samas tuleb tähelepanu pöörata asjaolule, et erinevate pettuste osakaaludega treeningbaaside sobivus sõltub väga palju rakendatavast algoritmist.

Parimate parameetrite otsingute osas võib esile tõsta vajaduse eristada erinevad tehingutüübid kaardi varasemast tehinguajaloost ning leida agregeeritud tehingud just tehinguliikide järgi mitte üldisel tasemel nagu näiteks Hand [15] või Bahnsen [4]. Kuluefektiivse meetodi rakendamisel olid esialgsed tulemused väga paljulubavad sarnaselt Bahnseni töödele [4, 5], kuid viimase etapi hindamises kuluefektiivse meetodi eelis kinnitust ei leidnud.

Järgnevalt on välja toodud magistritöö käigus saavutatud olulisemad tulemused lähtuvalt töö alguses püstitatud uurimisküsimustest.

5.1 Sobivaim treeningbaas

Sobivaima treeningbaasi valimine sõltub suhteliselt palju aluseks võetavast mõõdikust ning konkreetsest algoritmist. Tava kulumudeli puhul saavutatav sääst järjest suureneb, mida suurem on pettuste osakaal treeningbaasis. Seevastu kuluefektiivse mudeli puhul on parimad tulemused pettuste 10-20% osakaalu juures ning pettuste osakaalu kasvades tulemused langevad. 1-5% baasid on tavamudeli puhul täiesti mitterahuldavate

tulemustega, kuluefektiivse mudeli puhul võib aga 5% baasi pidada ka rahuldavaks, kui välja arvata närvivõrkude algoritm.

Kui võrrelda keskmisi skoore, siis F_1 skoor saavutab kõrgeima taseme 10% baasi juures ja peale seda kergelt langeb, F_2 skoor aga paraneb järjepidevalt pettuste osakaalu kasvades ja on kõrgeimal tasemel 50% baasi korral.

Kokkuvõtvalt võib järeldada, et kindlasti pole treenimiseks sobilikud kõige madalamate ehk 1% ja 5% pettuste osakaaludega baasid, sest nende puhul on tulemused läbi kõigi näitajate kõige madalamal tasemel. Parimaks treeningbaasiks võib pidada 20% osakaaluga baasi, mille puhul on tulemused stabiilselt head kõigi oluliste mõõdikute puhul. Sellisel juhul saavutatakse maksimumi lähedane kulusääst ja ka klassifitseerimise täpsus on heal tasemel.

5.2 Parimad parameetrid

Suurimat kulusäästu pakkuvaks parameetrite grupiks on kaarditehingu teostanud terminali ja kliendi/kaardi tuvastuse info, millele järgnesid kaardi kohaloluta tehingud (CNP), keskmised välismaised tehingud ja viimase 24 tunni sarnased tehingud. F_1 ja F_2 skoorid kinnitasid samu järeldusi, mis sai tehtud kulumudelite puhul. Eraldi võib välja tuua, et parameetrite lisandumine parandas algoritmide klassifitseerimise täpsust oluliselt rohkem kui kulumudeli poolt saavutatavat täiendavat kulusäästu.

Eksperimendid näitasid veel, et kõik parameetrid ei pruugigi mudelite täpsusele kasuks tulla. Seejuures töötasid erinevad parameetrid eri algoritmidega erinevalt, s.t. mis sobis ühele algoritmile hästi ei pruukinud teise puhul tulemust parandada. Teatud juhtudel uue parameetri lisamine hoopis halvendas väga olulisel määral mudeli tulemuslikkust.

Enamus algoritmide puhul saavutati parim kulusääst ja klassifitseerimise täpsus kui kombineeriti tehinguinfo, terminali ja autentimisinfo, CNP tehingute ning viimase 24h sarnaste tehingute info. Kõrvale jäeti kliendi profiil, kuni 30 päeva agregeeritud tehingud, eelmise tehingu detailinfo ja keskmised varasemad välismaised tehingud.

5.3 Kulupõhise lähenemise mõju

Algbaaside võrdluses andis kuluefektiivne mudel oluliselt suuremat kulusäästu kui tava kulumudel. Samuti parameetrite võrdluses tõi kuluefektiivne mudel pea kõigil juhtudel oluliselt suurema keskmise säästu. Näiteks juba baasvariandi korral, kui aluseks oli vaid tehingupõhine info, oli kulumudeli sääst keskmiselt 73,1% võrreldes tavamudeli 24,1%-ga.

Viimase testi põhjal aga kuluefektiivne mudel ei parandanud algoritmide tulemuslikkust ning saavutatud tulemused olid kas samal tasemel või veidi halvemad kui tava kulumudeliga saavutatu. Kuluefektiivse mudeli eelise mõjule pääsemine sõltub väga palju konkreetsetest petturlikest tehingutest. Töö autori nägemuse järgi võiks see olla paralleelselt jooksev mudel, mida võiks rakendada täiendava vaba tehingute kontrollressursi olemasolul.

Üldiselt sobib aga kulupõhine lähenemine tulemuste hindamiseks hästi, sest annab tulemustele väga selgelt arusaadava ja võrreldava hinnangu. Lõplike algoritmide testbaasi peal saavutatud keskmine kulusääst 80-90% on seejuures vägagi arvestatav.

5.4 Sobivaimad masinõppe algoritmid ja algoritmide kombineerimine

Parima tulemuse kulumudeli järgi saavutas juhuslik mets (90%), millele järgnes otsustuspuu (85%) ja *boost* algoritm 88% kulusäästuga. Suhteliselt lähedase tulemuse kulusäästu mõistes saavutas ka närvivõrkude algoritm (83%). Nende kõigi puhul püsis ka kontrollitavate tehingute maht küllaltki madalal tasemel, nimelt umbes iga kolmas kontrollitav tehing oli ka reaalselt petturlik (otsustuspuu puhul iga viies). Kui SVM algoritm oli klassifitseerimise täpsuse järgi ühtlaselt heal tasemel, siis kulusäästu mõttes oli see halvim.

Nii klassifitseerimise kui saavutatud kulusäästu järgi võib parimateks algoritmideks pidada juhuslikku metsa, *boost*-i ja närvivõrke. Nende algoritmide puhul on kõige madalamal tasemel ka valehäirete arv.

Mitme algoritmi kombineerimine loodetud täiendavat täpsuse paranemist ja kulusäästu ei toonud ning käesoleva töö kontekstis tundub, et selline lähenemine pole antud juhul põhjendatud. Rõhku võiks panna hoopis treeningandmete maksimaalsele

mitmekesisusele pettuste erinevate tüüpide mõistes ja pidevale algoritmide taastreenimisele.

5.5 Edasised tegevused ja uurimisteemad

Kõige suuremat kasu kaarditehingute pettuste monitoorimise algoritmidest on võimalik saavutada siis, kui süsteem töötab reaalajaliste andmete peal. Kui algoritmi usaldusväärsus on piisavalt kõrgel tasemel, siis võiks süsteemil olla volitused konkreetse kaarditehingu otse läbi laskmiseks, täiendavasse kontrolli suunamiseks või tagasi lükkamiseks. Tehingute automaatse tagasi lükkamise õigus kiirendaks kogu protsessi oluliselt ning aitaks kindlasti ära hoida olulisel määral kahjusid. Seda seetõttu, et kontrolli suunatud tehingute ülevaatamine võtab aega ning kontrolli ei teostata 24 tundi ööpäevas (sh nädalavahetustel). Seejuures on eraldi uurimiskoht, et kuidas hinnata nendest tehingutest saadavat kahju, mis automaatselt süsteemi poolt blokeeriti, kuid mis tegelikult olid olemuselt legitiimsed. Sellest võib tuleneda klientide pahameel ning samuti jääb pangal teenimata tehingutasude tulu. Sellest tulenevad ka kõrgendatud nõudmised algoritmide täpsusele. Sellise süsteemi väljatöötamine nõuab eraldi täiendavat uurimist ja sobivate IT lahenduste väljatöötamist.

Kaarditehingute pettuste tuvastamisele sarnast lähenemist on võimalik rakendada finantsasutuses mitmetes erinevates valdkondades. Näiteks on finantsasutustel kohustus monitoorida klientide makseid ning tuvastada rahapesu kahtlusega tehinguid. Seejuures on rahapesu tuvastamine fikseeritud reeglite abil isegi keerulisem ülesanne. Pigem on oluline tuvastada erinevaid mustreid ja käitumismalle, mis võivad viidata rahapesule. Siinkohal võib masinõppel olla väga oluline eelis, sest võimaldab oluliselt täpsemalt kahtlaseid tehinguid tuvastada ning täiendavasse kontrolli suunata. Käesoleva töö käigus omandatud kogemused on väga väärtuslikud ning aitavad kindlasti lahendada näiteks ka rahapesu monitooringuga seotud probleeme.

Kokkuvõte

Käesoleva magistritöö eesmägiks oli masinõppel põhinevate meetoditega luua mudelid, mis suudaksid monitoorida pangakaartidega tehtavaid tehinguid, vähendada manuaalselt üle kontrollitavate tehingute mahtu ning pettustega kaasnevaid kahjusid. Probleemi lahendamiseks läbiti erinevad masinõppe algoritmide treenimiseks ja testimiseks vajalikud etapid. Magistritöö teoreetilises osas tutvustas autor finantspettuste liike ja kaardipettuste mahte nii Eestis kui Euroopas laiemalt, lisaks anti ülevaade antud valdkonna kohta tehtud varsematest uurimustest.

Töö käigus testiti erinevate pettuste osakaaludega treeningbaase, vaadeldi kaheksa erineva parameetrite grupi mõju algoritmide tulemuslikkusele ja võrreldi kuut erinevat masinõppe algoritmi. Lisaks hinnati kulupõhise lähenemise mõju masinõppe tulemuste hindamisele ning katsetati algoritmide kombineerimist tulemuste täiendavaks parandamiseks.

Töö tulemuste põhjal saab järeldada, et väga madala pettuste osakaaluga treeningbaasid ei ole sobilikud algoritmide treenimiseks ning parimaks treeningbaasiks võib pidada 20% pettuste osakaaluga baasi, mille puhul olid tulemused stabiilselt heal tasemel kõigi oluliste mõõdikute puhul.

Enamus algoritmide puhul saavutati parim kulusääst ja klassifitseerimise täpsus kui kombineeritud olid tehinguinfo, terminali ja autentimisinfo, kaardi kohaloluta tehingu ja viimase 24h sarnaste tehingute info. Kliendi profiil ja vahetult eelnenud tehingu info algoritmide tulemuslikkust ei parandanud.

Kuluefektiivse mudeli eelis tava kulumudeli ees jäi kindla kinnituseta ning nõuab täiendavat uurimist pikema perioodi ja erinevate andmete peal. Üldiselt sobis aga kulupõhine lähenemine tulemuste hindamiseks hästi, sest andis tulemustele väga selgelt arusaadava ja võrreldava hinnangu. Lõplike algoritmide testbaasi peal saavutatud keskmine kulusääst 80-90% oli seejuures vägagi arvestatav.

Masinõppe algoritmidest võib kaarditehingute pettuste tuvastamise puhul sobivaimaks pidada juhuslikku metsa, mille kulusääst ulatus 90%-ni. Heal tasemel olid ka *boost* ja närvivõrkude algoritmid. Nende kõigi puhul püsis ka kontrollitavate tehingute maht

küllaltki madalal tasemel, nimelt ligikaudu iga kolmas kontrollitav tehing oli reaalselt petturlik.

Mitme algoritmi kombineerimine loodetud täiendavat täpsuse paranemist ja kulusäästu ei toonud ning käesoleva töö kontekstis tundub, et selline lähenemine pole põhjendatud. Rõhku võiks panna hoopis treeningandmete maksimaalsele mitmekesisusele pettuste erinevate tüüpide mõistes ja pidevale algoritmide taastreenimisele.

Magistritöös püstitatud eesmärgid said täidetud ja töö tulemusena sai loodud masinõppe algoritmid, mis on võimelised väga heal tasemel kaarditehingute pettusi tuvastama ning vähendama pettustest tulenevat kahju hinnanguliselt kuni 90% ulatuses. Töö autor tutvustas tulemusi ka LHV pangas ning hetkel käib tihe töö uurimaks tehnilisi võimalusi, kuidas töö käigus saadud teadmisi ning välja töötatud algoritme reaalajaliste andmete peal rakendada. Lisaks ollakse huvitatud sarnase süsteemi väljatöötamisest muudes valdkondades nagu näiteks maksete ja rahapesu monitooring.

Kasutatud kirjandus

[1] Abdallah, A., Maarof, M., Zainal, A. Fraud detection system: A survey. – *Elsevier. Journal of Network and Computer Applications*, 2015, 68, 90-113.

[2] Anderson, R. The Credit Scoring Toolkit: theory and practice for retail credit risk management and decision automation. New York : Oxford University Press, 2007.

[3] Alpaydin, E. Introduction to Machine Learning, Second Edition. London : The MIT Press Cambridge, 2010.

[4] Bahnsen, A.C., Aouada, D., Sojanovic, A., Ottersten, B. Feature engineering strategies for credit card fraud detection. – *Elsevier. Expert Systems With Applications*, 2016, 51, 134-142.

[5] Bahnsen, A.C. Aouda, D., Sojanovic, A, Ottersten, B. Cost Sensitive Credit Card Fraud Detection using Bayes Minimum Risk. – *12th International Conference on Machine Learning and Applications*, 2013.

[6] Bhattacharyya, S., Jha, S., Tharakunnel, K., Westland, J.C. Data mining for credit card fraud: A comparative study. – *Elsevier. Decision Support Systems*, 2011, 50, 602-613.

[7] Bolton, J.R., Hand, J.D. Statistical Fraud Detection: A Review. – *Statistical Science*, 2002, 17-3; 235–255.

[8] Brownlee, J. A Tour of Machine Learning Algorithms. Machine Learning Mastery. [WWW] http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/ (25.11.2013)

[9] Dal Pozzolo, A. Adaptive Machine Learning for Credit Card Fraud Detection. Brussels, Universite Libre de Bruxelles, 2015.

[10] Daumé, H. A Course in Machine Learning. Maryland : TODO, 2015.

[11] Delamaire, L., Abdou, H., Pointon, J. Credit card fraud and detection techniques: a review. – *Banks and Bank Systems*, 2009, 4-2.

[12] European Central Bank (ECB). Fourth report on card fraud. 2015.

[13] Elkan, C. The Foundations of Cost-Sensitive Learning. – *Seventeenth International Joint Conference on Artificial Intelligence*, 2001, 973–978.

[14] Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., Bing, G. Learning from class-imbalanced data: Review of methods and applications. – *Elsevier. Expert Systems With Applications*, 2017, 73, 220–239

[15] Hand, D. Adams, N. Transaction aggregation as a strategy for credit card fraud detection. – *Data Mining and Knowledge Discovery*, 2009, 18:30–55.

[16] Hand, D.J., Whitrow, C., Juszczak, P., Weston, D.J., Adams, N.M. Performance criteria for plastic card fraud detection tools. – *Journal of the Operational Research Society*, 2007, 59(7), 956–962.

[17] Hofmann, M., Brenann, P. A comprehensive survey of methods for overcoming the class imbalance problem in fraud detection. Dublin, Institute of Technology Blanchardstown, 2012.

[18] Hulse, J., Khoshgoftaar, T., Napolitano, A. Experimental Perspectives on Learning from Imbalanced Data. Florida Atlantic University, 2007.

[19] Maes, S., Tuyls, K. Credit card fraud detection using Bayesian and neural networks. Brussels, Vrije Universiteit Brussel, 2002.

[20] McDonald, C. Real Time Credit Card Fraud Detection with Apache Spark and Event Streaming. [WWW] *https://www.mapr.com/blog/real-time-credit-card-fraud-detection-apache-spark-and-event-streaming* (03.05.2016)

[21] Phua, C., Lee, V., Smith, K., Gayler, R. A Comprehensive Survey of Data Mining-based Fraud Detection Research. Australia, Victoria, Monash University, 2010.

[22] SAS. Machine Learning – What it is & why it matters. [WWW] *http://www.sas.com/it_it/insights/analytics/machine-learning.html* (12.02.2017)

[23] Statsoft. Naive Bayes Classifier Introductory Overview [WWW] *http://www.statsoft.com/textbook/naive-bayes-classifier* (12.02.2017)

[24] West, J., Bhattacharya, M. Intelligent financial fraud detection: A comprehensive review. – *Elsevier. Computers and Security*, 2016, 57; 47-66.

[25] West, J., Bhattacharya, M. Some Experimental Issues in Financial Fraud Mining. – *Elsevier. Procedia Computer Science*, 2016, 80, 1734-1744.

[26] Whitrow, C., Hand, D.J., Juszczak, P., Weston, D.J., Adams, N.M. Transaction aggregation as a strategy for credit card fraud detection. – *Data Mining and Knowledge Discovery*, 2008, 18(1), 30-55.

[27] Williams, G. Data Mining with Rattle and R. The Art of Excavating Data for Knowledge Discovery. New York : Springer, 2011.

Summary

The aim of this master's thesis was to develop machine learning algorithms that would be able to monitor card transactions, decrease the amount of manual inspections of suspicious card transactions and reduce the losses from card transaction fraud. To solve the research problem, the author went through different stages of training and testing of machine learning algorithms. In theoretical part of the thesis, the author introduced different types of financial fraud, provided an overview of card transaction volumes in Estonia and Europe, and introduced previous research carried out in the field.

In this study, the author analysed training sets with different fraud percentage, investigated the impact of eight different parameter groups on algorithm's perfomances while comparing six machine learning algorithms. In addition, the effect of applying cost based evaluation method on assessment of machine learning output was measured, also, combining different algoritms for improved predictions was tested.

The result of the work shows that training sets with very low fraud percentage are not suitable for training the algorithms. For training the algorithms, a training set with 20% of fraud cases was recognised as the most suitable, the results were evenly good with different metrics.

The best performing input parameter groups were transaction data, authentification data, historical transaction volume for card not present transactions and transaction volume of similar transactions within last 24 hours. Client profile and preceding transaction data did not improve the algorithms´ results.

The advantage of cost effective evaluation method over regular cost method remained unconfirmed. Further investigation over a longer period of time and with various datasets is needed. However, in general, cost based approach was suitable for evaluating the results as it made comparison easier to understand and quantifiable. The results were impressive as the trained machine learning algorithms achieved 80-90% reduction of loss from fraudulent card transactions.

The best machine learning algorithm was a random forest with a 90% reduction rate of loss from fraudulent card transactions. Boost and neural network achieved high results as well. These algorithms also maintained the number of transactions sent to manual

inspection at a reasonable level, more specifically every third transaction sent for inspection was in fact fraud.

The superiority of combined predictions from different algorithms was rejected. The emphasis should be put on the diversification of different fraud cases in training sets and continuous retraining of algorithms.

In conclusion, the research objectives were achieved and as a result machine learning algorithms were trained to be able to detect card transaction frauds at a very high level, reducing loss from card transaction fraud up to 90%. The author has introduced the findings to LHV bank which is now focusing on finding ways to implement the gained knowledge and trained algorithms on real-time data. In addition, there is a strong interest in developing similar systems for other areas, such as monitoring payments and preventing money laundering.

Lisa 1. Sisendparameetrite täielik loetelu

Parameetri nimetus	Andmetüüp	Täiendav selgitus
Kliendi profiili info:		
era_äriklient	kategooriline	Era või äriklient
sugu	kategooriline	
keel	kategooriline	
keel_grupp	kategooriline	
residentsus	kategooriline	
res_grupp	kategooriline	
kodakondsus	kategooriline	
kod_grupp	kategooriline	
maakond	kategooriline	
linn	kategooriline	
kliendi_kanal	kategooriline	Kliendilepingu sõlmimise kanal
vanus	numbriline	
vanus_grupp	kategooriline	
haridus	kategooriline	
tegevusala	kategooriline	
reklaami_info	kategooriline	Kliendi luba saata reklaami
inv_leping	kategooriline	Investeerimiskonto leping
kasvukonto	kategooriline	Kasvukonto leping
au_leping	kategooriline	Kuldkliendi leping
ettevotja	kategooriline	Klient omab ka ettevõtte kontot
Tehingu info:		
kaardi tootenimetus	kategooriline	
kaardi grupp	kategooriline	Kaarditoote liik
tehingu_pool	kategooriline	Krediit või deebet tehing
tehingu_tüüp	kategooriline	Kaardimakse, pangaautomaat
tehingu_riik	kategooriline	
riik grupp	kategooriline	
kaupmene_tüüp	kategooriline	
kaupmehe_grupp	kategooriline	
MCC_nimetus	kategooriline	Tehingu liik MCC koodi järgi

kaupmehe_risk	kategooriline	Riskigrupp Mastercard järgi
tehingu kuupäev	kuupäev	
tehingu_päev	numbriline	
tehingu_tund	numbriline	
tehingu_summa	numbriline	
Terminali ja tuvastuse meetodi info:		
Terminal_card_input	kategooriline	Kaardi kasutuse meetod terminalis
Terminal_auth_possibilities	kategooriline	Terminali tuvastuse võimalused
Cardholder_present	kategooriline	Kaardiomaniku kohalolu tehingu juures
Card_present	kategooriline	Kaardi kohalolu tehingu juures
Card_read_mode	kategooriline	Kaardi info lugemise meetod
Cardholder_auth	kategooriline	Kaardiomaniku tuvastuse meetod
Tehingule eelnenud välismaiste tehingute summad agregeerituna:		
day_sum1	numbriline	1 päeva tehingute summa
day_sum2	numbriline	2 päeva tehingute summa
day_sum3	numbriline	…
day_sum7	numbriline	…
day_sum30	numbriline	…
Tehingule vahetult eelnenud tehingu info:		
prev_tehingu_riik	kategooriline	
prev_riik_grupp	kategooriline	
prev_tehingu_tüüp	kategooriline	
prev_kaupmehe_tüüp	kategooriline	
prev_kaupmehe_grupp	kategooriline	
prev_kaupmehe_risk	kategooriline	
prev_tehingu_summa	numbriline	
prev_tehingu_tund	numbriline	
prev_tehingu_kuupäev	kuupäev	
prev_time_diff	numbriline	Käesoleva ja eelnenud tehingu vaheline aeg minutites
Samas riigis või samas riigis ja kaupmehe grupis tehtud tehingute summa ja arv viimase 24 tunni jooksul:		
sama_riik_sum	numbriline	
sama_riik_tk	numbriline	

sama_riik_MCC_sum	numbriline	
sama_riik_MCC_tk	numbriline	
Tehingule eelnenud viimase 30 päeva keskmine välismaiste tehingute arv:		
kuu_keskm_teh_sum	numbriline	
kuu_keskm_teh_arv	numbriline	
CNP (kaardi kohaloluta) tehingute summa ja arv:		
CNP_sum30	numbriline	Viimase 30 päeva jooksul
CNP_tk30	numbriline	…
CNP_sum1	numbriline	Viimase 24 tunni jooksul
CNP_tk1	numbriline	…

Lisa 2. Algbaaside võrdluse tulemused

Pettuste osakaal baasis **50%**

	DT	RF	BOOST	SVM	NN	LN
Vale positiivne määr	5.9%	2.7%	3.3%	3.3%	4.0%	3.6%
Vale negatiivne määr	80.2%	100.0%	86.0%	93.5%	100.0%	86.4%
Sensitiivsus	0.8%	0.6%	0.6%	0.6%	0.5%	0.6%
Valesti klassifitseerimine	6.2%	3.1%	3.7%	3.7%	4.6%	4.0%
Recall (TP/(TP+FN))	68.9%	58.4%	58.9%	58.9%	45.9%	57.4%
Keskmine klassi viga	18.0%	22.0%	22.0%	22.0%	29.0%	24.0%
Täpsus	10.9%	18.7%	15.7%	15.7%	10.7%	14.3%
F_1 skoor	18.8%	28.3%	24.8%	24.7%	17.4%	22.9%
F_2 skoor	33.4%	41.0%	38.0%	37.9%	27.7%	35.9%
Area under Recall	91.4%	91.5%	91.0%	88.3%	73.6%	91.8%
Area under ROC (AUC)	91.3%	91.4%	91.0%	88.2%	73.5%	91.7%
Sääst tavamudeli %	79.7%	88.0%	86.4%	53.2%	80.8%	82.2%
Sääst kulumudel %	76.8%	61.3%	84.9%	79.9%	83.6%	82.1%

Pettuste osakaal baasis **20%**

	DT	RF	BOOST	SVM	NN	LN
Vale positiivne määr	1.8%	1.1%	1.4%	1.9%	2.1%	2.0%
Vale negatiivne määr	67.9%	70.3%	71.3%	58.9%	64.6%	69.4%
Sensitiivsus	0.3%	0.3%	0.3%	0.4%	0.4%	0.3%
Valesti klassifitseerimine	2.5%	1.8%	2.2%	2.5%	2.7%	2.7%
Recall (TP/(TP+FN))	32.1%	29.7%	28.7%	41.1%	35.4%	30.6%
Keskmine klassi viga	35.0%	36.0%	36.0%	30.0%	34.0%	36.0%
Täpsus	15.7%	22.7%	17.4%	18.3%	15.4%	14.1%
F_1 skoor	21.1%	25.7%	21.7%	25.4%	21.4%	19.3%
F_2 skoor	26.5%	28.0%	25.4%	33.0%	28.1%	24.8%
Area under Recall	82.9%	89.5%	89.9%	90.0%	89.8%	91.1%
Area under ROC (AUC)	82.8%	89.4%	89.9%	90.0%	89.8%	91.0%
Sääst tavamudeli %	83.1%	71.7%	82.3%	47.6%	64.2%	59.6%
Sääst kulumudel %	84.3%	75.6%	86.5%	84.5%	83.9%	83.6%

Pettuste osakaal baasis **10%**

	DT	RF	BOOST	SVM	NN	LN
Vale positiivne määr	1.5%	0.5%	0.8%	1.1%	1.3%	1.5%
Vale negatiivne määr	61.2%	79.4%	72.2%	67.0%	77.5%	73.2%
Sensitiivsus	0.4%	0.2%	0.3%	0.4%	0.2%	0.3%
Valesti klassifitseerimine	2.2%	1.3%	1.6%	1.8%	2.1%	2.3%
Recall (TP/(TP+FN))	38.8%	20.6%	27.8%	33.0%	22.5%	26.8%
Keskmine klassi viga	32.0%	40.0%	36.0%	34.0%	40.0%	38.0%
Täpsus	20.9%	31.2%	25.8%	23.3%	15.1%	15.6%
F_1 skoor	27.2%	24.8%	26.7%	27.3%	18.0%	19.8%
F_2 skoor	33.1%	22.1%	27.3%	30.5%	20.5%	23.5%
Area under Recall	66.5%	88.7%	83.9%	88.0%	90.3%	90.7%
Area under ROC (AUC)	66.4%	88.6%	83.8%	87.9%	90.3%	90.7%
Sääst tavamudeli %	81.9%	68.5%	73.1%	44.0%	60.5%	58.0%
Sääst kulumudel %	80.3%	80.2%	88.3%	86.4%	84.7%	83.2%

Pettuste osakaal baasis **5%**

	DT	RF	BOOST	SVM	NN	LN
Vale positiivne määr	0.5%	0.1%	0.4%	0.4%	0.6%	1.0%
Vale negatiivne määr	79.4%	98.1%	88.0%	77.5%	100.0%	78.9%
Sensitiivsus	0.2%	0.0%	0.1%	0.2%	0.0%	0.2%
Valesti klassifitseerimine	1.3%	1.2%	1.3%	1.2%	1.7%	1.8%
Recall (TP/(TP+FN))	20.6%	1.9%	12.0%	22.5%	0.0%	21.1%
Keskmine klassi viga	40.0%	49.0%	44.0%	39.0%	50.0%	40.0%
Täpsus	31.6%	13.3%	23.8%	37.0%	0.0%	18.0%
F_1 skoor	24.9%	3.3%	15.9%	28.0%	-	19.4%
F_2 skoor	22.1%	2.3%	13.3%	24.4%	-	20.4%
Area under Recall	65.5%	90.9%	84.5%	90.2%	49.9%	90.7%
Area under ROC (AUC)	65.3%	90.9%	84.4%	90.1%	49.7%	90.7%
Sääst tavamudeli %	22.7%	0.3%	16.4%	44.1%	-1.6%	54.8%
Sääst kulumudel %	74.7%	84.9%	88.4%	84.7%	0.0%	86.0%

Pettuste osakaal baasis **1%**

	DT	RF	BOOST	SVM	NN	LN
Vale positiivne määr	0.1%	0.0%	0.0%	0.0%	0.0%	0.2%
Vale negatiivne määr	92.3%	100.0%	93.3%	97.1%	100.0%	93.3%
Sensitiivsus	0.1%	0.0%	0.1%	0.0%	0.0%	0.1%
Valesti klassifitseerimine	1.0%	1.1%	1.0%	1.1%	1.0%	1.1%
Recall (TP/(TP+FN))	7.7%	0.0%	6.7%	2.9%	0.0%	6.7%
Keskmine klassi viga	46.0%	50.0%	46.0%	48.0%	50.0%	46.0%
Täpsus	50.0%	0.0%	63.6%	40.0%	-	31.8%
F_1 skoor	13.3%	-	12.1%	5.4%	-	11.1%
F_2 skoor	9.2%	-	8.2%	3.5%	-	8.0%
Area under Recall	64.9%	82.7%	78.5%	90.1%	0.0%	90.3%
Area under ROC (AUC)	64.8%	82.6%	78.4%	90.1%	50.0%	90.3%
Sääst tavamudeli %	10.2%	0.0%	10.7%	4.7%	0.0%	41.1%
Sääst kulumudel %	60.6%	85.3%	53.0%	84.1%	0.0%	71.8%

Lisa 3. Algbaaside võrdluse tulemused algoritmide järgi

Algoritm:	**DT**				
	Baasid pettuste osakaalu järgi				
	1%	**5%**	**10%**	**20%**	**50%**
Vale positiivne määr	0.1%	0.5%	1.5%	1.8%	5.9%
Vale negatiivne määr	92.3%	79.4%	61.2%	67.9%	80.2%
Sensitiivsus	0.1%	0.2%	0.4%	0.3%	0.8%
Valesti klassifitseerimine	1.0%	1.3%	2.2%	2.5%	6.2%
Recall (TP/(TP+FN))	7.7%	20.6%	38.8%	32.1%	68.9%
Keskmine klassi viga	46.0%	40.0%	32.0%	35.0%	18.0%
Täpsus	50.0%	31.6%	20.9%	15.7%	10.9%
F_1 skoor	13.3%	24.9%	27.2%	21.1%	18.8%
F_2 skoor	9.2%	22.1%	33.1%	26.5%	33.4%
Area under Recall	0.649	0.655	0.665	0.829	0.914
Area under ROC (AUC)	0.648	0.653	0.664	0.828	0.913
Sääst tavamudeli %	10.2%	22.7%	81.9%	83.1%	79.7%
Sääst kulumudel %	60.6%	74.7%	80.3%	84.3%	76.8%

Algoritm:	**RF**				
	Baasid pettuste osakaalu järgi				
	1%	**5%**	**10%**	**20%**	**50%**
Vale positiivne määr	0.0%	0.1%	0.5%	1.1%	2.7%
Vale negatiivne määr	100.0%	98.1%	79.4%	70.3%	100.0%
Sensitiivsus	0.0%	0.0%	0.2%	0.3%	0.6%
Valesti klassifitseerimine	1.1%	1.2%	1.3%	1.8%	3.1%
Recall (TP/(TP+FN))	0.0%	1.9%	20.6%	29.7%	58.4%
Keskmine klassi viga	50.0%	49.0%	40.0%	36.0%	22.0%
Täpsus	0.0%	13.3%	31.2%	22.7%	18.7%
F_1 skoor	-	3.3%	24.8%	25.7%	28.3%
F_2 skoor	-	2.3%	22.1%	28.0%	41.0%
Area under Recall	0.827	0.909	0.887	0.895	0.915
Area under ROC (AUC)	0.826	0.909	0.886	0.894	0.914
Sääst tavamudeli %	0.0%	0.3%	68.5%	71.7%	88.0%
Sääst kulumudel %	85.3%	84.9%	80.2%	75.6%	61.3%

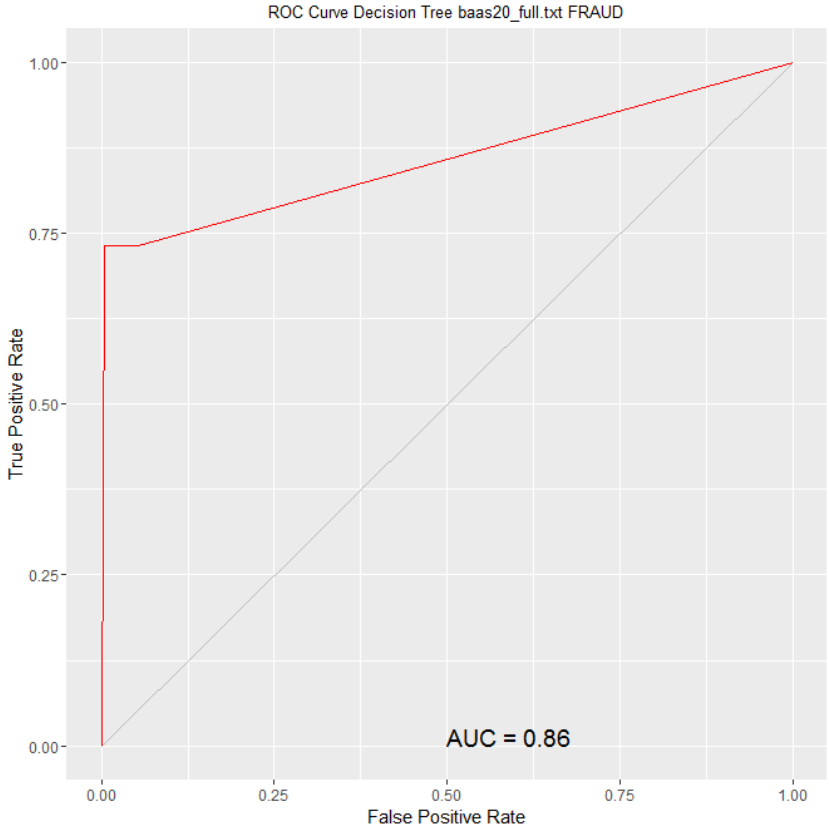
Algoritm:	**BOOST**				
	Baasid pettuste osakaalu järgi				
	1%	**5%**	**10%**	**20%**	**50%**
Vale positiivne määr	0.0%	0.4%	0.8%	1.4%	3.3%
Vale negatiivne määr	93.3%	88.0%	72.2%	71.3%	86.0%
Sensitiivsus	0.1%	0.1%	0.3%	0.3%	0.6%
Valesti klassifitseerimine	1.0%	1.3%	1.6%	2.2%	3.7%
Recall (TP/(TP+FN))	6.7%	12.0%	27.8%	28.7%	58.9%
Keskmine klassi viga	46.0%	44.0%	36.0%	36.0%	22.0%
Täpsus	63.6%	23.8%	25.8%	17.4%	15.7%
F_1 skoor	12.1%	15.9%	26.7%	21.7%	24.8%
F_2 skoor	8.2%	13.3%	27.3%	25.4%	38.0%
Area under Recall	0.785	0.845	0.839	0.899	0.910
Area under ROC (AUC)	0.784	0.844	0.838	0.899	0.910
Sääst tavamudeli %	10.7%	16.4%	73.1%	82.3%	86.4%
Sääst kulumudel %	53.0%	88.4%	88.3%	86.5%	84.9%

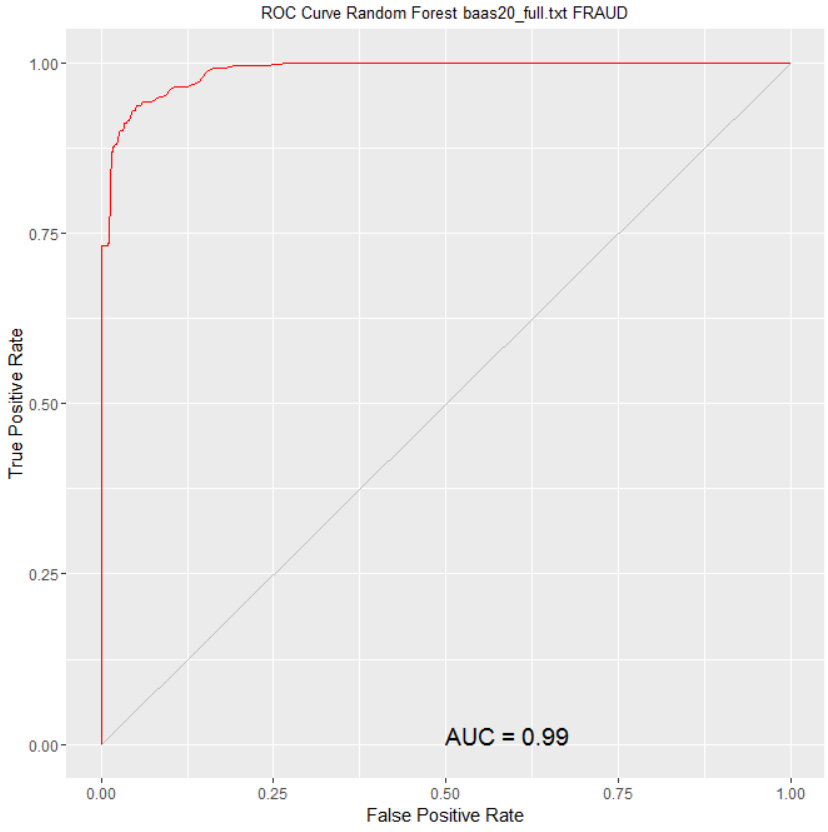
Algoritm:	**SVM**				
	Baasid pettuste osakaalu järgi				
	1%	**5%**	**10%**	**20%**	**50%**
Vale positiivne määr	0.0%	0.4%	1.1%	1.9%	3.3%
Vale negatiivne määr	97.1%	77.5%	67.0%	58.9%	93.5%
Sensitiivsus	0.0%	0.2%	0.4%	0.4%	0.6%
Valesti klassifitseerimine	1.1%	1.2%	1.8%	2.5%	3.7%
Recall (TP/(TP+FN))	2.9%	22.5%	33.0%	41.1%	58.9%
Keskmine klassi viga	48.0%	39.0%	34.0%	30.0%	22.0%
Täpsus	40.0%	37.0%	23.3%	18.3%	15.7%
F_1 skoor	5.4%	28.0%	27.3%	25.4%	24.7%
F_2 skoor	3.5%	24.4%	30.5%	33.0%	37.9%
Area under Recall	0.901	0.902	0.880	0.900	0.883
Area under ROC (AUC)	0.901	0.901	0.879	0.900	0.882
Sääst tavamudeli %	4.7%	44.1%	44.0%	47.6%	53.2%
Sääst kulumudel %	84.1%	84.7%	86.4%	84.5%	79.9%

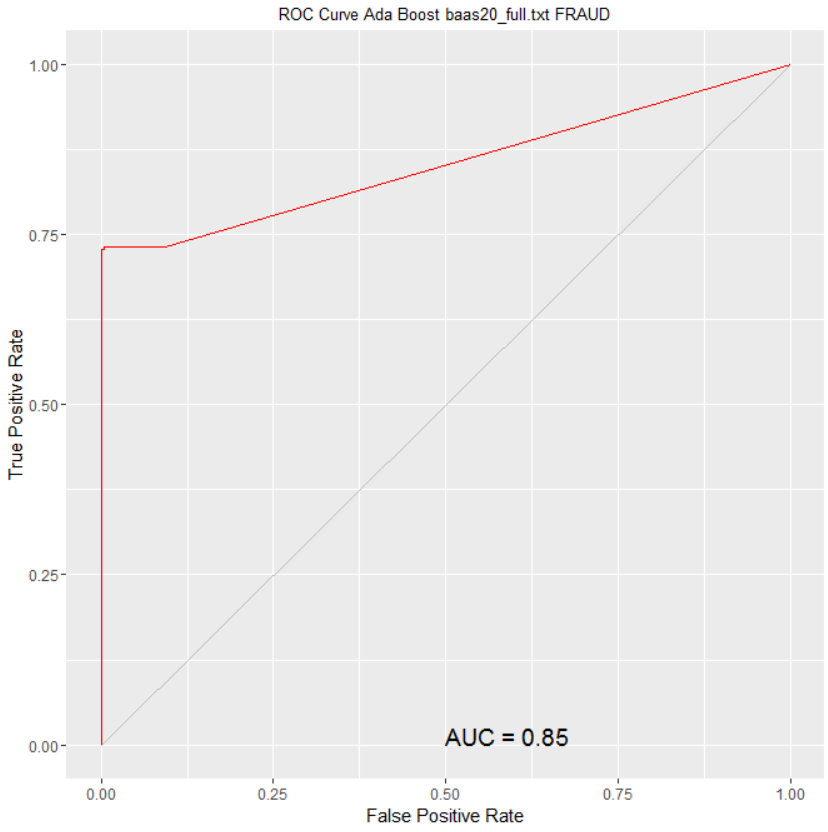
Algoritm:	**NN**				
	Baasid pettuste osakaalu järgi				
	1%	**5%**	**10%**	**20%**	**50%**
Vale positiivne määr	0.0%	0.6%	1.3%	2.1%	4.0%
Vale negatiivne määr	100.0%	100.0%	77.5%	64.6%	100.0%
Sensitiivsus	0.0%	0.0%	0.2%	0.4%	0.5%
Valesti klassifitseerimine	1.0%	1.7%	2.1%	2.7%	4.6%
Recall (TP/(TP+FN))	0.0%	0.0%	22.5%	35.4%	45.9%
Keskmine klassi viga	50.0%	50.0%	40.0%	34.0%	29.0%
Täpsus	-	0.0%	15.1%	15.4%	10.7%
F_1 skoor	-	-	18.0%	21.4%	17.4%
F_2 skoor	-	-	20.5%	28.1%	27.7%
Area under Recall	0.000	0.499	0.903	0.898	0.736
Area under ROC (AUC)	0.500	0.497	0.903	0.898	0.735
Sääst tavamudeli %	0.0%	-1.6%	60.5%	64.2%	80.8%
Sääst kulumudel %	0.0%	0.0%	84.7%	83.9%	83.6%

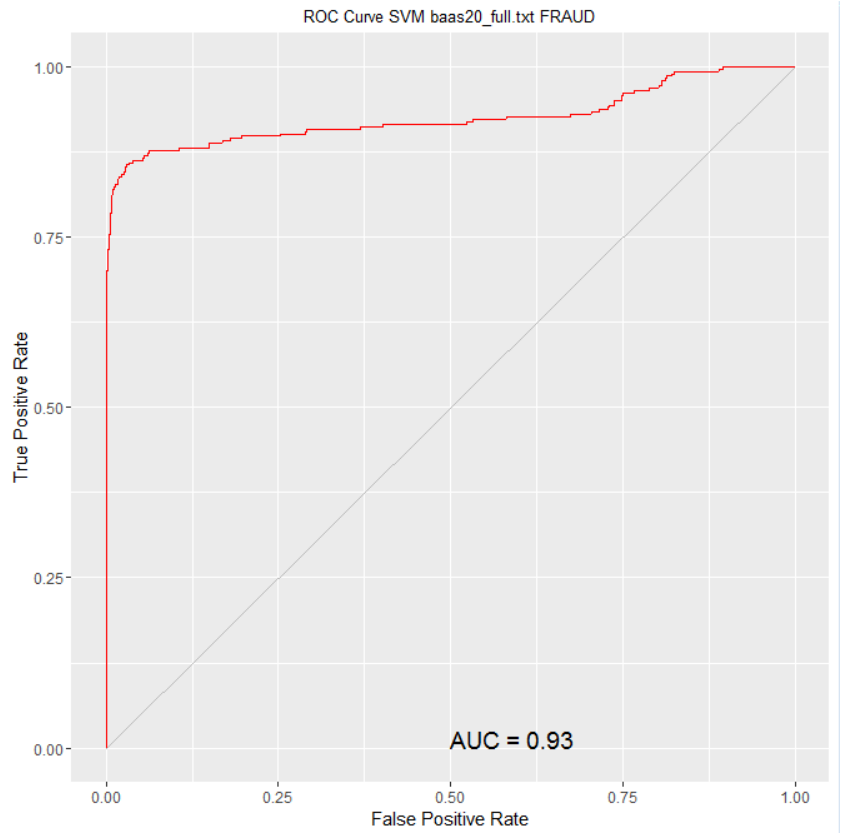
Algoritm:	**LN**				
	Baasid pettuste osakaalu järgi				
	1%	**5%**	**10%**	**20%**	**50%**
Vale positiivne määr	0.2%	1.0%	1.5%	2.0%	3.6%
Vale negatiivne määr	93.3%	78.9%	73.2%	69.4%	86.4%
Sensitiivsus	0.1%	0.2%	0.3%	0.3%	0.6%
Valesti klassifitseerimine	1.1%	1.8%	2.3%	2.7%	4.0%
Recall (TP/(TP+FN))	6.7%	21.1%	26.8%	30.6%	57.4%
Keskmine klassi viga	46.0%	40.0%	38.0%	36.0%	24.0%
Täpsus	31.8%	18.0%	15.6%	14.1%	14.3%
F_1 skoor	11.1%	19.4%	19.8%	19.3%	22.9%
F_2 skoor	8.0%	20.4%	23.5%	24.8%	35.9%
Area under Recall	0.903	0.907	0.907	0.911	0.918
Area under ROC (AUC)	0.903	0.907	0.907	0.910	0.917
Sääst tavamudeli %	41.1%	54.8%	58.0%	59.6%	82.2%
Sääst kulumudel %	71.8%	86.0%	83.2%	83.6%	82.1%

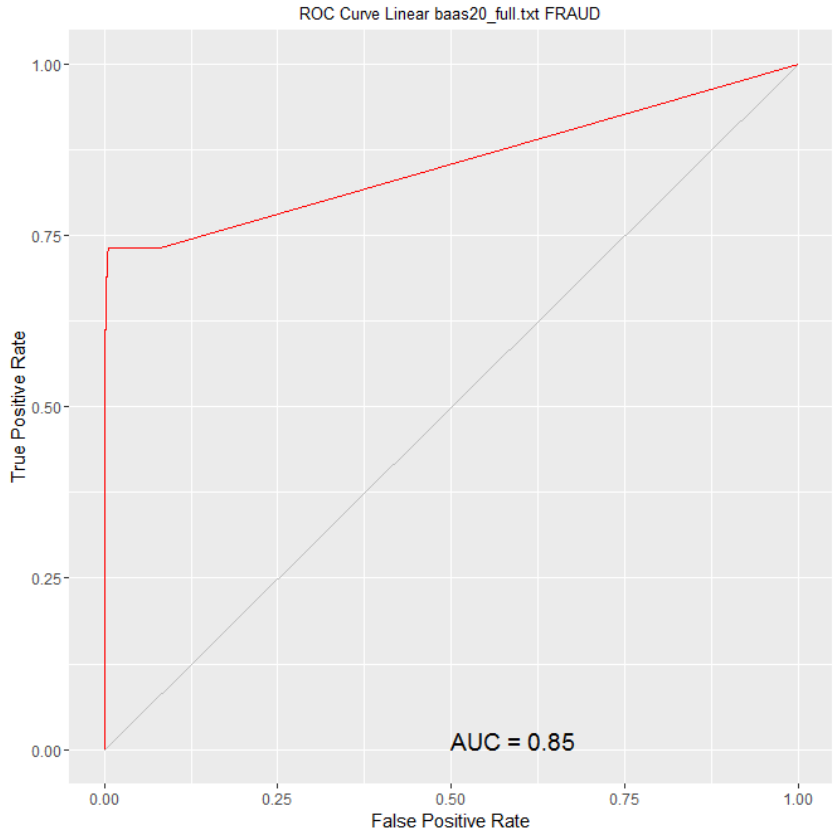
Lisa 4. Lõplike algoritmide ROC kõverad

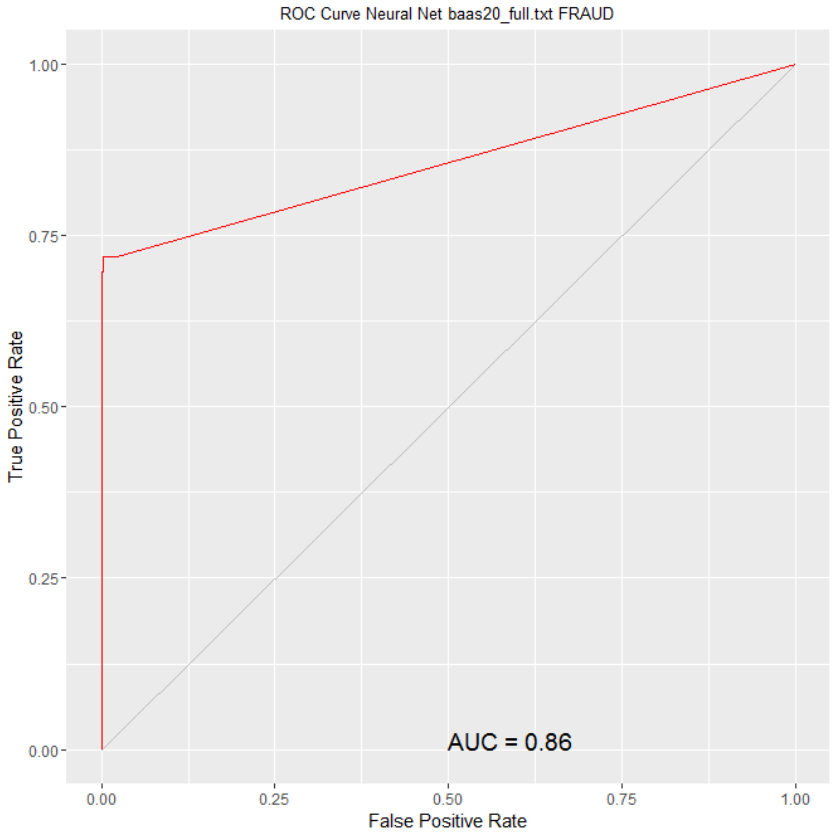





ROC Curve Ada Boost baas20_full.txt FRAUD
True Positive Rate
False Positive Rate
AUC = 0.85
0.00
0.25
0.50
0.75
1.00


ROC Curve SVM baas20_full.txt FRAUD
True Positive Rate
False Positive Rate
AUC = 0.93
0.00
0.25
0.50
0.75
1.00


ROC Curve Linear baas20_full.txt FRAUD
True Positive Rate
False Positive Rate
AUC = 0.85
0.00
0.25
0.50
0.75
1.00


ROC Curve Neural Net baas20_full.txt FRAUD
True Positive Rate
False Positive Rate
AUC = 0.86
0.00
0.25
0.50
0.75
1.00

Lisa 5. Lõplike algoritmide statistilised näitajad

	Mudel					
	DT	**RF**	**BOOST**	**SVM**	**NN**	**LN**
Vale positiivne määr	0.4%	0.1%	0.2%	0.3%	0.2%	0.6%
Vale negatiivne määr	26.8%	28.2%	27.1%	26.4%	29.2%	26.8%
Sensitiivsus	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%
Valesti klassifitseerimine	0.4%	0.1%	0.2%	0.4%	0.3%	0.7%
Recall (TP/(TP+FN))	73.2%	71.8%	72.9%	73.6%	70.8%	73.2%
Keskmine klassi viga	44.0%	14.0%	14.0%	13.0%	14.0%	14.0%
Täpsus	18.7%	57.1%	32.8%	22.4%	27.1%	13.0%
F_1 skoor	29.8%	63.7%	45.2%	34.4%	39.2%	22.0%
F_2 skoor	46.3%	68.3%	58.5%	50.6%	53.5%	37.9%
Area under Recall	0.858	0.000	0.854	0.925	0.856	0.855
Area under ROC (AUC)	0.858	0.988	0.854	0.925	0.856	0.855
Sääst tavamudeli %	85.2%	89.9%	88.3%	67.1%	83.5%	81.9%
Sääst kulumudel %	83.7%	24.5%	87.2%	79.0%	83.5%	82.9%
Pettuseks määratud tavamudel	1 104	356	632	563	741	1 585
Pettuseks määratud kulumudel	1 304	10 811	793	2 385	787	1 433
Kulumudel vs tavamudel	18.1%	2 936%	25.5%	323.6%	6.2%	-9.6%