TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Daniel Aju 192018IABM

Aktsiahindade ennustamine generatiivse võistlusvõrguga

Magistritöö

Juhendaja: Tõnn Talpsepp
PhD

Tallinn 2021

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: Daniel Aju

09.05.2021

Annotatsioon

Käesoleva töö eesmärgiks on luua sügavõppe tehnikal põhinev generatiivse võistlusvõrgu (GAN) mudel ning rakendada seda aktsiahindade ennustamisel. Töö käigus uuritakse tehniliste indikaatorite ja sentimentaalanalüüsi kasutamise mõju ennustusjõudlusele ning vaadeldakse ka mudeli treenimisprotsessi. Loodud mudelit simuleeritakse ajaloolistel andmete peal ning tulemusi võrreldakse osta ja hoia investeerimisstrateegiaga. Analüüsiti ka teisi teadustöid, et teada saada, milline on GAN mudelite ennustusjõudlus võrreldes traditsioonilisemate mudelitega, mida aktsiahindade ennustamisel kasutatakse.

Loodud mudel oli suuteline osta ja hoia investeerimisstrateegiat ületama, kuid sellel esines ka teatud puudujääke ja probleeme, mis on töö käigus välja toodud. Samuti jõuti järeldusele, et GAN mudelid suudavad aktsiahindu paremini ennustada kui traditsioonilisemad mudelid

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 49 leheküljel, 5 peatükki, 30 joonist, 14 tabelit.

Abstract

Forecasting stock prices with generative adversarial networks

The aim of this work is to create a generative competitive network (GAN) model based on deep learning techniques and to apply it in stock price forecasting. In the course of this work, the impact of the use of technical indicators and sentimental analysis on the prediction performance of the model is studied, the training process of the model is also examined. The model is simulated on historical stock data and the results are compared with the buy and hold investment strategy. Other similar research was also examined to determine the predictive performance of GAN models compared to more traditional models that are used to predict stock prices.

The model created in this work was able to beat the buy and hold investment strategy. However, it also had certain shortcomings and problems that have been identified and addressed in the course of the work. It was also concluded that GAN models are better able to predict stock prices than more traditional models. It was also concluded that GAN models can predict stock prices more accurately than other traditional models.

The thesis is in Estonian and contains 49 pages of text, 5 chapters, 30 figures, 14 tables.

Lühendite ja mõistete sõnastik

ANN	*Artificial neural network*, tehnisnärvivõrk
API	*Application programming interface*, rakendusliides
ARIMA-GARCH	Masinõppe mudel
BERT	*Bidirectional Encoder Representations from Transformers*, loomuliku keele töötluse keelemudel
Bs	*Batch size*, ploki suurus
DJI	*Dow Jones Industrial*, börsiindeks
GAN	*Generative adversarial network*, generatiivne võistlusvõrk
GRU	*Gated recurrent unit*, rekurrentse närvivõrgu arhitektuur
LSTM	*Long short term memory*, rekurrentse närvivõrgu arhitektuur
Naive Bayes	Klassifitseerimismudel
NLP	*Natural language processing*, loomuliku keele töötlus
PNN	*Probabilistic neural network*, tõenäosuslik närvivõrk
RNN	*Recurrent neural network*, rekurrentne närvivõrk
SVM	*Support-vector machine*, masinõppe mudel
TDNN	*Time delay neural network*, aegviitega närvivõrk
TextCNN	Konvolutsiooniline närvivõrgu mudel teksti klassifitseerimiseks

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

1.1 Taust ja probleemitõstatus

Tuleviku ettenägemine on inimkonnale huvi pakkunud juba aastatuhandeid. Tänapäeval kasutatakse erinevates valdkondades teatud sündmuste ennustamiseks kõikvõimalike signaale, sisendeid, mustreid ja tehnikaid.

Efektiivse turu hüpoteesi kohaselt kajastab väärtpaberi hind juba kogu olemasolevat informatsiooni ning väärtpaberi hinna ennustamine on praktiliselt võimatu. Teooria on empiiriliselt kohati toetust leidnud, kuid laialdast akadeemilist kinnitust pole see saanud. Teooriale räägivad vastu ka näiteks börsimullid ning -krahhid, samuti ka investorid ja investeerimisfondid, kes turgu järjepidevalt lüüa suudavad. [1], [2]

1970. aastal ilmunud ülevaateartiklis [3] jagas Fama efektiivse turu hüpoteesi empiirilised testid kolmeks: nõrgaks, pooltugevaks ja tugevaks vormiks. Testide jaotamise eesmärk oli vaadelda, millisel tasemel efektiivse turu hüpotees n-ö laguneb. Eelnimetatud vormid viitavad hüpoteesis “kogu olemasoleva informatsiooni” all mõeldud informatsiooni hulgale. Nõrk vorm kasutab informatsioonina vaid ajaloolisi hinnaandmeid ning kontrollib nende abil hüpoteesi vettpidavust. Pooltugev vorm kaasab informatsioonina ka avalikult saadaval olevaid andmed, näiteks uudised, majandusaasta aruanded või ettevõtte teadaanded ning tugev vorm käsitleb lisaks ka privaatset teavet. Fama järeldas, et nii nõrga kui ka pooltugeva vormi ümberlükkamiseks piisavalt tõendeid ei olnud.

Üks esimestest närvivõrgu rakendamist börsiturul käsitlevatest töödest avaldati aastal 1988 White’i poolt [4]. White väitis, et kuigi efektiivse turu teooria nõrka vormi pole veenvalt suudetud ümber lükata, siis tegu on siiski teooriaga ja asjakohaste tõendite abil on seda võimalik teha. Töö eesmärgiks oli otsida lihtsa närvivõrgu abil seaduspärasusi, mis kajastuvad ajaloolistes börsihinna andmetes ning oleks ühtlasi tõestuseks teooria ümber lükkamisel. Kuigi White ei leidnud piisavalt tõendeid, siis olulise järeldusena tõi ta välja, et isegi lihtsad närvivõrgud võivad oma olemuselt olla väga dünaamilised.

1990. aastal ilmunud töös [5] kirjeldatakse Tokyo börsihinnaindeksi ennustamise süsteemi, mis kasutas modulaarseid närvivõrke. Eesmärk oli närvivõrkude üldistusvõimele tuginedes ennustada ostu- ja müügisignaale kuu aega ette. Tulemuste valideerimiseks viidi läbi simulatsiooni ning loodud süsteem tõi 33 kuu pikkuses ajaraamis ligi 18% võrra suurema kasumi kui osta ja hoia investeerimisstrateegia. 1998. aastal ilmunud uurimistöös [6] võrreldi omavahel TDNN, PNN ning RNN arhitektuuriga närvivõrke lühiajaliste trendide ennustamisel. Autorid klassifitseerisid aktsia hinna 2%, 5% või 10% tõusu kui kasumi teenimise võimalust. Järeldustena toodi välja, et kõigi kolme võrguga on võimalik lühiajalisi trende edukalt ennustada. Samuti märkisid autorid, et võrkude vale-positiivsete arvu on võimalik kontrollida ja konservatiivsemate ennustuste soovi korral ka nullini vähendada.

Viimastel aastatel on börsiennustamisel sagedamini uuritud otsustuspuude ja tugivektor-masinate kõrval kiirelt arenevaid sügavõppel põhinevaid tehnikaid [7], [8]. Võimalikuks põhjuseks sügavõppe populaarsuse kasvu taga võib pidada tehnika ja arvutusjõudluse kiiret arengut [8]. Finantsmaailmas laiemalt on samuti mõistetud sügavõppe potentsiaali ning üldiselt saavutavad sügavõppel põhinevad mudelid paremaid tulemusi kui traditsioonilisemad tehnikad [9].

Aktsiahindade ennustamise probleem seisneb aktsiate stohhastilises olemuses ning nende ennustamise keerukuses. Kuivõrd efektiivse turu hüpoteesi üle on võimalik pikalt diskuteerida ning paljudes allikates on seda ka tehtud, siis antud töö puhul on eelduseks, et turg on mõnevõrra ebaefektiivne ning seda on võimalik teatud määral ette ennustada. Olgugi, et ilmselt ei ole aktsiahindu mitte kunagi võimalik sajaprotsendiliselt ette ennustada, siis investori vaatenurgast on oluline saavutada täpsus, mis annab suurima kasumi teenimise võimaluse. Sellest tulenevalt on antud töö keskmeks probleemiks aktsiahindade stohhastilisuse probleemi lahendamine, kasutades selleks sügavõppel põhinevat uudset meetodit – generatiivset võistlusvõrku.

1.2 Eesmärk

Töö peamine eesmärk on uudse generatiivse võistlusvõrgu (GAN) arhitektuuril põhineva mudeli loomine ja selle katsetamine aktsiahindade ennustamisel. Uuritakse nii tehniliste indikaatorite kui ka sentimentaalanalüüsi mõju mudeli ennustusjõudlusele ning võetakse vaatluse alla ka mudeli üldisem käitumine treeningprotsessil. Mudelile tuginedes luuakse

lihtne investeerimisstrateegia ja seda katsetatakse simulatsioonis, kus peamiselt võrreldakse seda osta ja hoia strateegiaga. Autor on püstitanud järgmised uurimisküsimused:

1. Kui hästi on võimalik aktsiahindasid ennustada kasutades GAN mudelit võrreldes teiste hetkel levinumate sügavõppe tehnikatega?

2. Millist mõju avaldab mudeli ennustusjõudlusele tehniliste indikaatorite kasutamine ühe sisendina?

3. Millist mõju avaldab mudeli ennustusjõudlusele sentimendi kasutamine ühe sisendina?

4. Kas lihtsa investeerimisstrateegia kasutamine loodud mudeli ennustustele tuginedes on efektiivsem kui osta ja hoia strateegia?

1.3 Struktuur

Töö teises peatüks antakse ülevaade kasutatud metoodikast ning põhilistest tööriistadest. Selgitatakse lähemalt kasutatud tehnikate põhimõtteid ning antakse ka ülevaade valideerimiseks kasutatud meetoditest.

Kolmas osa keskendub töö tulemustele. Kirjeldatakse täpsemalt andmete ettevalmistamist, mudeli arhitektuuri ning mudelit kasutades läbi viidud eksperimente, mille käigus valitakse välja parim mudel. Mudelile tuginedes luuakse investeerimisstrateegia ning simulatsioon selle testimiseks ajaloolistel andmetel.

Neljandas osas analüüsitakse loodud mudelit ja põhjendatakse sisendite valikuid/loomist. Samuti kõrvutatakse lahendust teiste sarnaste teadustööde tulemustega, et välja selgitada põhilised erinevused käesoleva töö mudeli ja teiste lahenduste vahel. Uuritakse ka seda, kus paikneb GAN oma ennustusjõudlusega võrreldes teiste levinumate mudelitega. Analüüsi käigus pakutakse välja muid võimalike lähenemisviise ning aspekte, mida antud töö puhul edasi uurida ja/või täiendada.

2 Metoodika

2.1 Masinõpe

Masinõpet on üldjoontes võimalik kirjeldada kui arvutuslike meetodeid, mis kasutavad kogemusi, et teatud ülesandeid paremini lahendada või teha täpseid ennustusi. Kogemusteks on üldjuhul elektroonilisel kujul ajalooline informatsioon. Kuna masinõppe algoritmide edukus sõltub olulisel määral andmetest, siis on masinõpe väga tihedalt seotud andmeanalüüsi ja statistikaga. Masinõpet kasutatakse laialdaselt mitmetes valdkondades, näiteks loomuliku keele töötlusel, arvutinägemises, meditisiiniliste diagnooside määramisel või krediitkaardipettuste avastamisel. [10]

Masinõppe süsteeme saab jaotada vastavalt sellele, mis koguses ja mis tüüpi juhendamist nad treenimise käigus kasutavad. Peamiselt liigitatakse masinõpet nelja kategooriasse: juhendatud õpe (*supervised learning*), juhendamata õpe (*unsupervised learning*), pooljuhendatud õpe (*semi-supervised learning*) ja stiimulõpe (*reinforcement learning*). [11]

Juhendatud õppe puhul kasutatakse treeningandmetes lisaks sisendandmetele ka märgendit õigest väärtusest. Tüüpilised juhendatud õppe ülesanded on klassifikatsioon ja regressioon. Klassifikatsiooni korral õpitakse andmete pealt, mis on kõrvutatud vastava klassi väärtusega, ning mudel peab õppima kuidas uute andmete korral neid klassifitseerida. Regressiooni puhul on klassi asemel ennustatavaks väärtuseks numbriline väärtus. Regressiooni algoritme kasutatakse tihtipeale samuti klassifitseerimiseks, näiteks on logistilise regressiooni puhul võimalik väljundina anda teatud objekti tõenäosus mingisse klassi kuulumise kohta. [11]

Juhendamata õppe puhul puudub sisendandmetel kindel väärtus, mida üritatakse ennustada. Algoritm peab iseseisvalt õppima ja andmete vahel seoseid leidma. Sellest tulenevalt on juhendamata õppe peamisteks ülesanneteks klasterdamine, anomaaliate tuvastamine, visualiseerimine ja andmete dimensioonide vähendamine. Klasterdamise eesmärk on objekte rühmitada erinevate omaduste ja omavaheliste sarnasuste alusel.

Visualiseerimine aitab andmeid kujutada – mõista, kuidas andmed on organiseeritud ja tuvastada teatud mustreid. Dimensioonide vähendamise eesmärk on andmeid lihtsustada ja kompaktsemale kujule viia ilma liialt informatsiooni kaotamata. Anomaaliate tuvastamise puhul üritatakse leida objekte, mis on teistest selgelt erinevad, olgu nendeks näiteks krediitkaardipettused või tootmisdefektid. [11]

Pooljuhendatud õppe puhul kasutatakse treeningandmetes nii märgenditega kui ka ilma märgenditeta andmeid. Märgenditeta andmete puhul ennustab algoritm sellele mingi kindla väärtuse. Ehk kui märgend puudub, siis ennustab algoritm märgendiga andmete alusel sellele märgendi. Pooljuhendatud õpet kasutatakse tavapäraselt siis, kui on ligipääs märgenditeta andmetele, kuid märgendite loomine on ajaliselt või majanduslikult liialt kulukas. [10]

Stiimulõppe puhul on uurimisobjektiks tarkvaral põhinev agent. Agendi ülesanne on kindlas keskkonnas õppida, kuidas kaardistada erinevaid tegevusi ja situatsioone, et seeläbi saavutada optimaalne tulemus. Agendile pole ette öeldud, milliseid tegevusi ta tegema peab, vaid ta peab tegevusi läbi proovides leidma sellised, mis kõige rohkem kasu annavad. Õppimise protsess on mõnevõrra sarnane katse-eksituse meetodile, kus agendi tegevuste tagajärjel premeeritakse või karistatakse teda vastava skooriga ning lõplik eesmärk on õppimise käigus saavutada parim võimalik tulemus. [12]

Antud töö raames kasutatakse juhendatud õpet. Allolev Tabel 1 annab ülevaate põhilistest masinõppega seotud terminitest ja põhimõtetest, mida antud töö raames on kasutatud.

Tabel 1. Põhilised töös kasutatud masinõppe terminid ja põhimõtted.

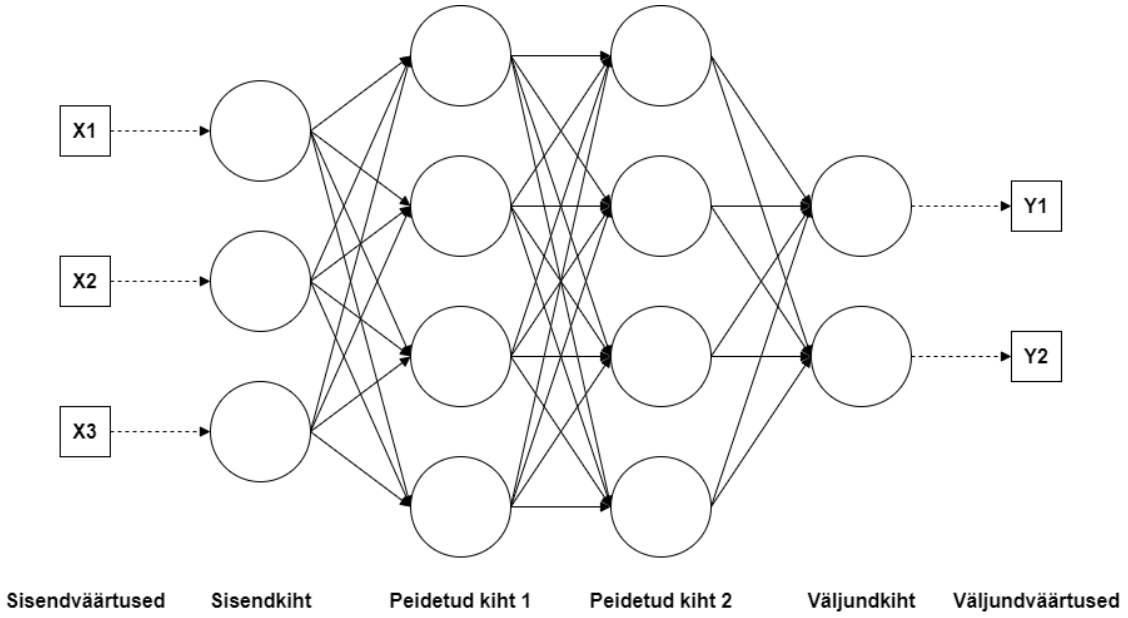
Termin	**Selgitus**
Tunnused (*feature*s)	Tunnused on andmestiku puhul näitajad, mida on võimalik mõõta. Tüüpiliselt on tabulaarsel kujul andmestiku tunnused esitatud päisena.
Eksemplar (*example*)	Tunnuste väärtused. Näiteks tabulaarsel kujul andmestiku puhul üks rida. Töös nimetatakse eksemplare ka X-väärtuseks.
Märgendid (*labels*)	Eksemplaridele omistatud väärtused või kategooriad. Kõrvutatakse ühe või mitme eksemplariga, mille ennustatavaks väärtuseks see on. Töös nimetatakse märgendeid ka Y-väärtuseks.

Termin	Selgitus
Hüperparameetrid (*hyperparameters*)	Parameetrid, mille väärtustega kontrollitakse mudeli õppimisprotsessi.
Epohh (*epoch*)	Hüperparameeter, mis määrab ära mitu läbimist õppimisalgoritm andmestiku peal kokku teeb.
Kaalud (*weights*)	Mudeli sisemised õpitavad parameetrid, mis kontrollivad neuronitevaheliste ühenduste tugevusi. Kaal määrab ära kui palju neuroni sisend väljundit mõjutab.
Ploki suurus (*batch size*)	Hüperparameeter, mis määrab ära mitu eksemplari õppimisalgoritm kasutab enne mudeli sisemisi parameetrite muutmist.
Kaofunktsioon (*loss function*)	Kadu on ennustusviga. Kaofunktsioon mõõdab päris ja ennustatud märgendi vahelist viga.
Gradient	Vektorväli, mille väärtuseid kasutatakse närvivõrgu kaalude uuendamiseks. Arvutatakse kadude pealt.
Optimiseerija (*optimizer*)	Algoritm, mida kasutatakse närvivõrgu atribuutide muutmiseks (näiteks kaalud), et minimeerida kadusid.
Ülesobitamine (*overfitting*)	Ülesobitamine tähendab, et mudel omab suurt täpsust treeningandmete puhul, kuid valideerimisandmete puhul on täpsus oluliselt kehvem. Ehk sisuliselt ei oma mudel üldistusvõimet.

2.2 Tehisnärvivõrgud ja sügavõpe

Tehisnärvivõrgud (edaspidi närvivõrgud) on matemaatilised mudelid masinõppe valdkonnas, mis on inspireeritud bioloogilistest neuronite võrgustikest inimese peaajus [13]. Närvivõrgud koosnevad neuronitest ning neuronitevahelistest seostest. Kõigil seostel on kaalud, mis on põhiline moodus informatsiooni salvestamiseks närvivõrgus – kaalude uuendamise käigus õpib võrk uut informatsiooni. Närvivõrkude käitumist kujundab selle arhitektuur, mida üldjuhul määratletakse neuronite arvu, kihtide ja seoste tüüpide järgi.

Üheks tuntumaks närvivõrgu arhitektuuriks on pärilevivõrk [13]. Nagu ka teised närvivõrgud, koosneb see sisendkihist, ühest või mitmest peidetud kihist ning väljundkihist. Iga kiht on täielikult ühendatud järgneva kihiga ning igal kihil võib olla erinev arv neuroneid. Joonisel 1 on kujutatud lihtsat pärilevivõrku.



Joonis 1. Mitmekihiline pärilevivõrk kahe peidetud kihiga.

Sügavõpe on üks masinõppe meetoditest, see on ühtlasi ka kaasaegne nimetus tehisnärvivõrkudele, mis koosnevad paljudest kihtidest ning on oma olemuselt keerukad [14], [15]. Masinõppe tehnikate puhul on üldjuhul oluline, et andmed koosneksid õigetest tunnustest ning sageli konstrueeritakse neid tunnuseid käsitsi, et algoritmidega paremaid tulemusi saavutada. Sügavõppe puhul ei ole vajalik käsitsi tunnuseid luua, kuna sügavad närvivõrgud suudavad iseseisvalt tuvastada ja õppida andmete keerulisi seoseid tänu oma mitmetele kihtidele ja keerukusele.

2.3 Rekurrentsed närvivõrgud

Rekurrentsed närvivõrgud on olemuselt sarnased pärilevivõrkudega, kuid põhiline erinevus seisneb selles, et rekurrentsel võrgul on sisemine mälu. Tänu sisemisele mälule on võrk võimeline töötlema jada kujul olevat järjestikust informatsiooni. Kui tavaline pärilevivõrk kasutab sisendina 2D-kujul andmeid, siis rekurrentses võrgus võetakse arvesse ka ajasamme, mille tulemusena kasutab rekurrentne võrk andmeid 3D-kujul ehk aegridadena. Aegridade kasutamine ja võrgu sisemine mälu võimaldavad võrgul sõltuda eelnevatest ajasammudest. Samuti on aegridade järjestusel on oluline roll väljundväärtuse kujundamisel, kuna väljundväärtus sõltub otseselt varasematel ajasammudel saavutatud väärtustest. [13]

2.3.1 LSTM

LSTM (*Long Short-Term Memory*) on 1997. aastal väljapakutud edasiarendus rekurrentsetele närvivõrkudele. Tavalise rekurrentse võrgu puhul esineb mitmeid probleeme, mis muudavad võrgu treenimise aeglaseks ja treenimise käigus kipuvad ka esimeste sisendite mälu olekud kaduma, mis tähendab, et iga ajasammuga läheb teatud osa informatsiooni kaduma. LSTM on jõudluse poolest oluliselt võimekam, mis võimaldab stabiilsemat ja kiiremat treenimist ning võimaldab ka andmetes tuvastada pikaajalisi sõltuvusi. LSTM-i elemendil on samasugused sisendid ja väljundid nagu seda on tavalisel rekurrentse võrgu neuronil, kuid elemendi sisemine arhitektuur on oluliselt keerukam ja kasutab rohkem parameetreid ja ka väravate süsteemi. LSTM-i element suudab väravate süsteemi abil õppimise käigus tuvastada olulisi sisendeid ning salvestada need oma pikaajalisse mälusse. Niisamuti suudab ta õppida kui kaua seda sisendit on vaja oma sisemises mälus hoida ning mis hetkel seda enam vaja teha ei ole. [16]

2.3.2 GRU

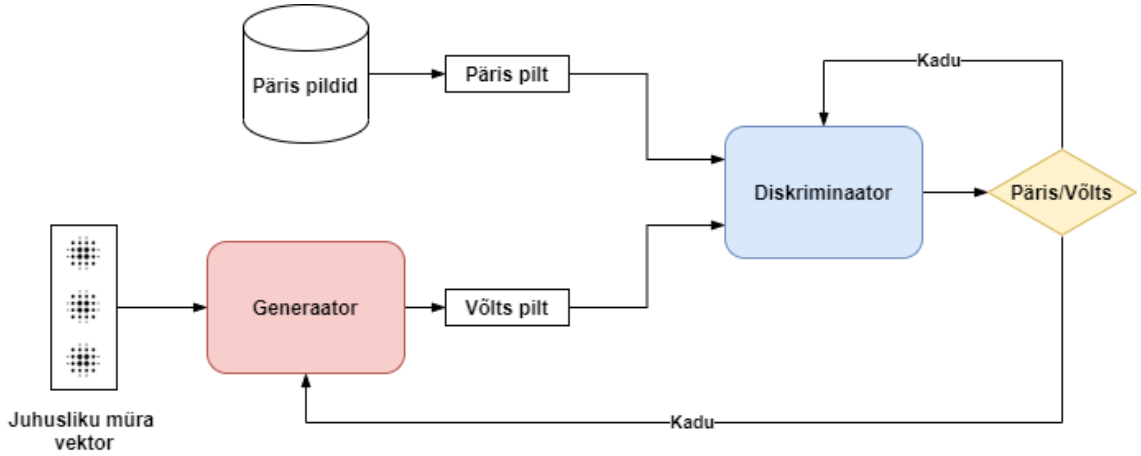
GRU (*Gated Recurrent Unit*) on 2014. aastal väljapakutud lihtsustatud versioon LSTM elemendist. Kolme värava asemel kasutab GRU vaid kahte väravat ning omab ka vähem parameetreid. Mitmed uuringud on leidnud, et GRU jõudlus on sarnane LSTM-iga, kuid tänu lihtsamale arhitektuurile vajab GRU treenimisel vähem arvutusjõudlust, mis teeb treenimisprotsessi kiiremaks kui LSTM-i puhul. [17]

2.4 Generatiivne võistlusvõrk (GAN)

Generatiivsed võistlusvõrgud on võimsad generatiivsed mudelid, mida algselt kasutati pilditöötluse ning tehisnägemise valdkonnas, kuid on nüüdseks leidnud laiemat kasutust ka teistes valdkondades. Mudel töötati välja 2014. aastal Goofellow poolt [10]. Esialgu oli tegemist mudeliga, mis lahendas juhendamata õppega seotud probleeme, kuid mudelit on edukalt rakendatud ka pooljuhendatud õppe probleemide lahendamisel [18], täielikult juhendatud õppe [19] ning ka stiimulõppe puhul [20].

Generatiivse võistlusvõrgu (*GAN*) [21] näol on tegu sügavaõppe raamistikuga, mis koosneb kahest omavahel võistlevast tehisnärvivõrgust – genereerivast ja diskrimineerivast võrgust. Kahe võrgu omavahelist võistlust saab pidada nullsummamänguks, kus generatiivne võrk on vastandatud diskrimeerivale võrgule.

Generaatori eesmärk on diskriminaatorit petta luues selleks müra alusel genereeritud pilte. Diskriminaator peab päris ja generaatori poolt loodud võlts-piltidel vahet tegema ja need vastavalt klassifitseerima. Võrkudevahelise võistluse käigus korrigeeritakse pidevalt mõlema võrgu kaale, et mõlemad võrgud õpiks seeläbi oma ülesannet paremini tegema. Ideaalne tulemus saavutatakse kui diskriminaator ei suuda enam vahet teha päris ning genereeritud piltidel. GAN-i arhitektuur on kujutatud Joonisel 2. Kuigi mudelit rakendati algselt piltide peal, siis samu põhimõtteid on võimalik kasutada ka teiste andmete peal, mida on antud töö raames ka tehtud.



Joonis 2. GAN mudeli arhitektuur.

Traditsioonilise GAN mudeli puhul võib treenimise käigus esineda mitmeid probleeme ja puudujääke. Peamiselt on nendeks *Vanishing Gradients*, *Mode Collapse* ja *Failure to Converge* probleemid. *Vanishing Gradients* probleem seisneb selles, et kui treenimise käigus mudeli kaale uuendatakse, siis võib juhtuda et gradient on liiga väike ning see takistab mudeli kaalude muutmist. Halvimal juhul võib see täielikult lõpetada mudeli edasise õppimise. *Mode Collapse* tähendab, et mudel ei suuda enam korralikult üldistada. Kuna generaator üritab alati leida diskriminaatorile sisendit, mis viimasele õige tundub, siis võib juhtuda, et generaator õpibki genereerima ühte ja sama väljundit põhjustades generaatori läbikukkumise. *Failure to Converge* tähendab, et kui treenimise käigus generaator piisavalt hea jõudluse saavutab ja diskriminaator ei suuda enam vahet teha päris ja genereeritud andmetel, siis muutub ennustuste tegemine juhuslikuks. Ehk diskriminaator ei anna enam sisulist tagasisidet generaatorile ja kuna generaator õpib selle juhusliku tagasiside põhjal, siis generaatori kvaliteet võib seetõttu langeda. [22]

Wasserstein GAN (WGAN) [23] on alternatiiv traditsioonilise GAN mudelile, mis üritab lahendada *Vanishing Gradients* ja *Mode Collapse* probleeme. Võrreldes traditsioonilise GAN-iga, mis kasutab Minimax kaofunktsiooni, kasutab WGAN Wasserstein'i kaofunktsiooni. Kui traditsioonilises GAN mudelis annab diskriminaator MinMax funktsiooni kasutades tõenäosuse, siis WGAN mudelis diskriminaator mõõdab päris ja genereeritud andmete jaotuste vahelist Wasserstein kaugust. Kuna WGAN mudeli diskriminaator ei tee vahet päris ja genereeritud andmetel, siis kutsutakse seda vahel ka diskriminaatori asemel kriitikuks. WGAN aitab leevendada probleeme, mis esinevad traditsioonilise GAN mudeli treenimisel, kuid autorid on märkinud, et kasutatav kaalude lõikamise (*weight clipping*) meetod tekitab endiselt treenimisel ebastabiilsust.

WGAN-GP [24] on väljapakutud lahendus, mis aitab lahendada *Failure to Converge* probleemi. WGAN-GP kasutab kaalude lõikamise asemel gradient karistust (*gradient penalty*), mis teeb treeningprotsessi lihtsamaks ja stabiilsemaks. Antud töö raames on kasutatud WGAN-GP mudelit, hoidmaks ära potentsiaalseid probleeme, mis esinevad traditsioonilise GAN ja WGAN mudelite puhul.

2.5 FinBERT

BERT (*Bidirectional Encoder Representations from Transformers*) [25] on sügavõppe tehnikal põhinev keelemudel, mis on arendatud Google teadlaste poolt ning avaldati esmakordselt 2018. aastal. BERT on loomuliku keele töötluse (NLP) eel-treenimise meetod, mis tähendab, et seda treenitakse tekstikorpustel mingit kindlat keelt või keeli mõistma. Mudeli raamistik koosneb kahest põhilisest sammust: eel-treenimine ning peenhäälestamine.

Eeltreenimisel kasutatakse suuri tekstikorpuseid, et mudelil tekiks arusaam või mõistmine õpitavast keelest või keeltest. Kuna tekstikorpuste maht on suur ja üldjuhul nendel puuduvad tähised ja õiged väljundid, siis eeltreenimisel kasutakse juhendamata õpet. Kogu eeltreenimise protsess võib ajaliselt võtta mitu päeva ning selle käigus omandab mudel arusaama õpitava keele omadustest. Peenhäälestamine toimub juhendatud õppega – tekstikorpused on tähistatud. Eesmärgiks on mudeli kohandamine erinevate NLP ülesannete täitmiseks. Iga ülesande jaoks tuleb sisestada vastava ülesandega seotud sisend- ja väljundparameetrid. Samuti vajab peenhäälestamine võrreldes eeltreenimisega

oluliselt vähem arvutusjõudlust ning mõne tunniga on võimalik mudel kohandada kindla NLP probleemi lahendamiseks.

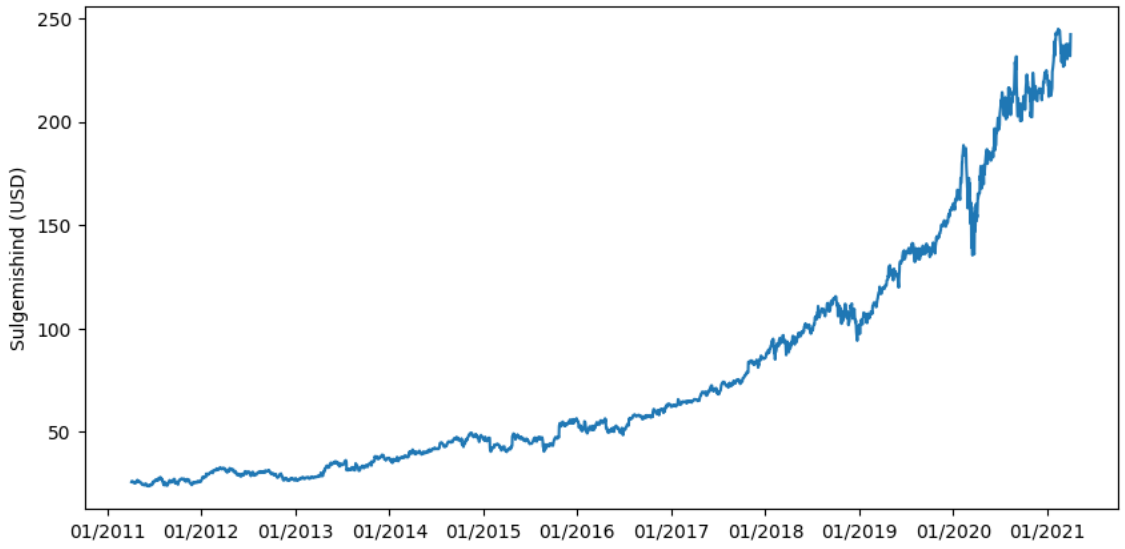
FinBERT [26] on eeltreenitud BERT mudel, mis on sobilik finantsteemaliste NLP probleemide lahendamiseks. Andmed, millega mudelit treeniti on pärit kolmest erinevast finantskommunikatsiooni tekstikorpusest, mille kogusuurus on 4,9 miljardit tekstisõna. Lisaks eeltreenitud mudelile on autorid avaldanud ka FinBERT-i peenhäälestatud mudeli, mida saab kasutada sentimendi klassifitseerimiseks. Antud töö raames on kasutatud eelnevalt nimetatud FinBERT peenhäälestatud mudelit.

2.6 Andmed

Töö käigus kasutatud aktsiahindade andmed on võetud Yahoo Finance [27] portaalist. Aktsiahinna andmed koosnevad seitsmest tunnusest ning alamhulk andmetest on näitena kujutatud Tabelis 2. Töös peamiselt kasutatud Microsofti sulgemishinnad (*Close*) on graafiliselt kujutatud Joonisel 3.

Tabel 2. Microsofti aktsiahindade andmestik.

Date	**Open**	**High**	**Low**	**Close**	**Adj Close**	**Volume**
25/07/2019	140.43	140.61	139.32	140.19	137.4745	18356900
26/07/2019	140.37	141.68	140.3	141.34	138.6023	19037600
29/07/2019	141.5	141.51	139.37	141.03	138.2983	16605900
30/07/2019	140.14	141.22	139.8	140.35	137.6315	16846500



Joonis 3. Microsofti aktsia sulgemishinnad.

Sentimentaalanalüüsi jaoks kasutatud Microsoftiga seotud uudiste andmestik on päritud TickerTick-APIst [28] – API, mis tagastab ettevõttega seotud uudiste pealkirju. Microsoftiga seotud uudised on API kaudu päritud vahemikus 09.07.2018 – 06.04.2021. Varasemaid Microsoftiga seotud uudiseid API ei paku. API-st päritud andmetest kokku pandud andmestikus on 30915 rida andmeid, mis sisaldavad uudise pealkirja, uudisteportaali veebisaiti, kuupäeva ja muid vähem olulisemaid andmeid.

2.7 Tulemuste valideerimine

Valideerimine toimub kahes põhilises etapis: mudeli treenimisel ning järeltestides. Mudeli treenimise käigus vaadeldakse nii treening- kui ka testandmete peal ruutkeskmist viga (RMSE). RMSE eelis näiteks keskmise absoluutse vea (MAE) ees on see, et RMSE annab kõrgema kaalu suurematele vigadele mistõttu on selle kasutamine kasulikum kui on soov suuremaid vigu vältida. Valideerimisel on RMSE puhul on eelistatud võimalikult väikest väärtust.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (1)$$

Järeltestimine (*backtesting*) on meetod, millega simuleeritakse investeerimisstrateegiat ajalooliste andmete peal, et analüüsida selle riskitaset ja kasumlikkust. Järeltestimise etapis luuakse eelnevalt valitud mudeli ümber lihtne investeerimisstrateegia ning simuleeritakse seda ajaloolistel andmetel. Eelkõige on vaadeldud mudeli kasumlikkust võrreldes osta ja hoia strateegiaga, kuid on ka teisi olulisi mõõdikuid. Tabelis 3 on välja toodud kõik järeltestimisel vaadeldud mõõdikud ning nende selgitused.

Tabel 3. Järeltestimisel kasutatavad mõõdikud ja nende selgitused.

Mõõdik	**Selgitus**
Equity Final [$]	Lõplik omakapitali väärtus.
Equity Peak [$]	Kõrgeim omakapitali väärtus.
Return [%]	Lõplik tootlus.
Buy & Hold Return [%]	Osta ja hoia strateegia tootlus.
Sharpe Ratio	Sharpe suhtarv – mõõdik, mis võrdleb investeeringu tootlust ja riski. Näitab kuidas vara tootlikkus kompenseerib investori riske. Mida kõrgem väärtus seda suurem tootlikkus.

Mõõdik	Selgitus
Max. Drawdown [%]	Suurim kaotus kõrgeimast omakapitali väärtusest järgmise madalaima punktini.
Number of trades	Tehtud tehingute arv.
Win Rate [%]	Kasumit teeninud tehingute protsent.
Best Trade [%]	Parima tehingu kasum/kahjum protsentides.
Worst Trade [%]	Kõige kehvema tehingu kasum/kahjum protsentides.
Avg. Trade [%]	Keskmise tehingu kasum/kahjum protsentides.
Avg. Trade Duration	Keskmine tehingu pikkus päevades.

2.8 Kasutatud tööriistad

2.8.1 Python

Python on üldotstarbeline, objektorienteeritud programmeerimiskeel, mille oluliseks omaduseks on selle loetavus ja lihtne süntaks. Pythonil on mitmeid kasutusalasid, kuid seda kasutatakse laialdaselt ka teaduslike arvutuste tegemisel, mida toetavad ka arvukad raamistikud ning teegid, mis just selleks otstarbeks on loodud.

2.8.2 TensorFlow

TensorFlow [29] on avatud lähtekoodiga masinõppe tarkvarateek, mis pakub erinevaid kaasaegseid masinõppega seotud tööriistu ja teeke ning pakub stabiilset API-t Pythoni ja C++ programmeerimiskeeltele.

2.8.3 Keras

Keras [30] on sügavõppe API, mis on kirjutatud Pythoni programmeerimiskeeles ning jookseb TensorFlow mootori peal. Peamiseks fookuseks on pakkuda kasutajasõbralikku ja lihtsalt kasutatavat liidest lahendamaks sügavõppega seotud probleeme.

2.8.4 NumPy

NumPy [31] on üks põhilisemaid teeke, mida kasutatakse Pythonis teaduslike arvutuste tegemisel. NumPy pakub tuge tegelemaks suurte mitmemõõtmeliste andmemassiivide ning maatriksitega ning sisaldab erinevaid tööriistu nende massiivide töötlemiseks ja manipuleerimiseks.

2.8.5 Pandas

Pandas [32] on andmetöötluse ja -analüüsi teek, mis on suures osas ehitatud NumPy teegi peale. Pandase pakub võimalust kasutada kiireid ning paindlikke andmestruktuure, mis muudavad struktureeritud ning aegridade andmetega töötamise lihtsaks ja mugavaks.

2.8.6 Pandas TA

Pandas TA [33] on teek, mis võimaldab erinevaid tehnilise analüüsiga seotud indikaatoreid ning funktsioone kasutada. Olemuselt on see Pandas teegi laiendus ehk indikaatorite ja funktsioonide lisamine toimub mugavalt ja kiirelt Pandas DataFrame andmestruktuuris. Saadaval on üle 130 erineva tehnilise analüüsiga seotud indikaatori ja funktsiooni.

2.8.7 Backtesting.py

Backtesting.py [34] on kasutajasõbralik raamistik, mis võimaldab investeerimisstrateegiaid luua ning neid ajaloolistel andmetel simuleerida. Simulatsiooni saab lisada erinevaid sätteid nagu näiteks tehingutasude arvestamine. Väljundina antakse hulk erinevaid mõõdikuid ning on võimalus ka simulatsiooni graafiliselt kujutada.

2.8.8 Google Colaboratory

Antud töö raames kasutati Google Colab [35] keskkonda, mis on Google poolt pakutud tasuta pilveteenus. Colabis on võimalik meelevaldset Pythoni koodi veebibrauseris kirjutada ning käivitada. Kasutajate jaoks teeb Colabi atraktiivseks see, et see on juba eelnevalt seadistatud ning teekide installimine ja importimine käib lihtsalt. Samuti annab võimaluse kasutada taasuta arvutusjõudlust, sealhulgas võimaldades ka GPU-de kasutamist.

2.9 Töö protsess

Töö algab GAN mudeli programmeerimisega ning selle kohandamisega, et piltide genereerimise asemel aktsia sulgemishindu ennustada. Kohandamise käigus tegeletakse ka andmete töötlemise ning vajalikule kujule viimisega, et neid oleks võimalik mudeli sisendina kasutada. Samuti kasutatakse FinBERT mudelit, et uudiste pealkirjade järgi konstrueerida semantilisuse skoorid, mida hiljem kasutatakse ühe tunnusena andmestikus. Loodud mudeli alusel viiakse läbi eksperimendid, et leida vastused

uurimisküsimustele 2 ja 3, mis uurivad sentimendi ja tehniliste indikaatorite mõju ennustusjõudlusele. Neljanda uurimisküsimuse raames rakendatakse eksperimentide käigus välja valitud mudeli alusel järeltestimise metoodikat, et võrrelda mudeli sooritusvõimet osta ja hoia investeerimisstrateegiaga. Esimesele uurimisküsimusele, mis võrdleb GAN-i võimekust teiste levinumate sügavõppe mudelitega, leitakse vastus töö analüüsi ja järelduste osas, kus analüüsitakse teisi sarnaseid teadustöid.

3 Töö tulemused

3.1 Andmete ettevalmistus ja töötlus

Et eksperimente läbi viia tuleb andmeid eelnevalt töödelda ning ka vajalike tunnuseid juurde lisada. Esialgu viiakse andmed tabeli kujule – indekseeritud kuupäeva järgi kasvavas järjekorras, tabeli päis koosneb tunnustest ning tabeli sisu vastava kuupäeva ning tunnuse väärtustest. Kuna aktsiahindade näol on tegu aegridadega, siis hiljem viiakse andmestik rekurrentsele närvivõrgule sobivale libiseva akna kujule. Kasutatud on Microsofti aktsiahinna andmeid ning sentimentaalanalüüsi osas on samuti Microsoftiga seotud andmed.

Tehnilist analüüsi kasutatakse väärtpaberi hinna muutumise prognoosimiseks toetudes minevikus toimunud aktsiahindade liikumisele. Investorid kasutavad ostu- ja müügiotsuste tegemisteks graafikuid ning indikaatoreid, mis on konstrueeritud aktsia hindadest, käibest, nõudlusest ja pakkumisest. Kuna tehnilised indikaatorid tuletatakse aktsiahinna andmete pealt, siis on välja valitud üldtuntud indikaatorid, need algandmete alusel konstrueeritud ja lõplikus andmestikus tunnustena kasutatud. Indikaatoriteks on valitud RSI, MACD, *Bollinger Band*, SMA5, SMA10, EMA5 ning EMA10. Indikaatorid on täpsemalt lahti seletatud Tabelis 4.

Tabel 4. Tehnilised indikaatorid ja nende selgitused.

Tehniline indikaator	**Selgitus**
RSI	RSI (*Relative Strength Index*) mõõdab ajalooliste hinnamuutuste suurusjärke, et hinnata, kas aktsia on üleostetud või ülemüüdud. RSI on vahemikus 0 kuni 100. Kui RSI väärtus on üle 70-ne, siis peetakse väärtpaberit üleostetuks ehk ülehinnatuks. Kui väärtus on alla 30-ne, siis peetakse väärtpaberit alaostetuks, ehk alahinnatuks.
MACD	MACD (*Moving Average Convergence Divergence*) näitab kahe erineva pikkusega väärtpaberi hinna liikuvte keskmiste suhet. Tavaliselt on lühema perioodi pikkus 12 päeva ja pikem on 26 päeva ning nende põhjal arvutatakse MACD-joon. Lisaks kasutatakse ka signaaljoont, mis arvutatakse MACD-joone pealt. Tüüpiliselt funktsioneerivad MACD-joon ja signaaljoon kui ostu

Tehniline indikaator	Selgitus
	või müügi signaalidena, olenevalt sellest, kas MACD-joon ületab signaaljoone või mitte.
Bollinger Band	*Bollinger Band* mõõdab, kas väärtpaber on ülemüüdud või üleostetud. See koosneb kolmest joonest – ülemine, keskmine ja alumine joon. Ülemine ja alumine joon on väärtpaberi hinna eelmiste päevade positiivsed ja negatiivsed standardhälbed Keskmine joon on väärtpaberi hinna liikuv keskmine. Kui keskmine joon on lähedal ülemisele, siis seda peetakse märgiks, et väärpaber on üleostetud. Kui keskmine joon on lähedal alumisele, siis peetakse väärtpaberit ülemüüduks.
SMA5	SMA5 (*Simple Moving Average*) on aritmeetiline liikuv keskmine, mis kasutab 5 päeva pikkust perioodi.
SMA10	SMA10 on analoogne SMA5-le, kuid kasutab 10 päeva pikkust perioodi.
EMA5	EMA5 (*Exponential Moving Average*) on liikuv keskmine, mis paneb suurema kaalu ja tähtsuse hiljutisematele väärtustele. Kasutab 5 päeva pikkust perioodi.
EMA10	EMA10 on analoogne EMA5-le, kuid kasutab 10 päeva pikkust perioodi.

Sentimentaalanalüüsiks API kaudu päritud andmete hulk oli esialgu 30915 rida. Kuna andmestik sisaldas palju müra blogipostituste ja muude taoliste artiklite näol, siis filtreeriti andmed üldtuntud finantsuudiste saitide järgi. Peale filtreerimist koosnes andmestik 4206-st reast ning oli vahemikus 10/07/2018 - 06/04/2021. Kuna API ei tagasta uudiste sisu ega lühikokkuvõtet, siis sentiment tuvastati uudiste pealkirjade järgi. Kõigi 4206 uudise pealkirja puhul klassifitseeriti FinBERT mudeli põhjal sentimentide tõenäosused.

Tabelis 5 on näha mõningate FinBERT'i poolt klassifitseeritud pealkirjade sentimentide tõenäosused. Pealkirjad on andmestikust välja valitud visuaalse vaatluse käigus keskendudes just sellistele, mis demonstreerivad FinBERT'i võimekust ning ka andmete kvaliteeti.

Tabel 5. FinBERTi poolt klassifitseeritud uudiste pealkirjad.

Pealkiri	**Positiivne**	**Negatiivne**	**Neutraalne**
As US-China trade tensions rise, shares of Apple, Microsoft, Amazon, Alphabet, and Facebook dropped between 3% and 5.2% Monday, losing a combined $162B in value	0.0073	0.9752	0.0175
3 Stocks To Watch This Coming Week: Microsoft, Apple, Amazon	0.0509	0.0344	0.9147
Microsoft revenue grew 13% despite coronavirus	0.9567	0.0150	0.0283
Microsoft wins $21.9B contract with U.S. Army to supply AR headsets	0.9306	0.0161	0.0533
Microsoft Says Users Hit 30 Billion Minutes a Day On Its Cloud-Based Apps	0.1177	0.0193	0.8629
Oracle Wins Bid for TikTok in U.S., beats Microsoft	0.8880	0.0197	0.0923
LinkedIn co-founder: These tweaks will make your job application stand out	0.8656	0.0089	0.1256
Does Microsoft's (NASDAQ:MSFT) CEO Salary Compare Well With Industry Peers?	0.3742	0.0239	0.6019
: Intel stock drops 6% after report of Microsoft developing its own chips	0.0091	0.9673	0.0236

Visuaalse vaatluse käigus oli näha, et pärast filtreerimist esines endiselt teatud hulgal müra. Kohati tekivad ka väärpositiivsed ning tõsinegatiivsed tulemused, kuna mudel ei tea, mis ettevõtte kohta ta ennustusi teeb. Kuid see pole mudeli süü, vaid pigem andmete kvaliteedist tulenev probleem. Mudeli klassifikatsioonid koondati üheks sentimendiskoori väärtuseks, lahutades positiivsest tõenäosusest negatiivne tõenäosus. Et sentimendiskoore põhiandmestikuga siduda, siis grupeeriti skoorid päevade kaupa ning võeti päeva keskmine sentimendiskoor. Juhul, kui turg oli uudise ilmumise kuupäeval suletud, siis võeti skooriks suletud päeva(de) ja esimese avatud päeva keskmine sentimendiskoor. Ehk kui laupäev ja pühapäev on turg suletud ning esmaspäeval on turg avatud, siis esmaspäeva sentimendiskoor on nende kolme päeva keskmine.

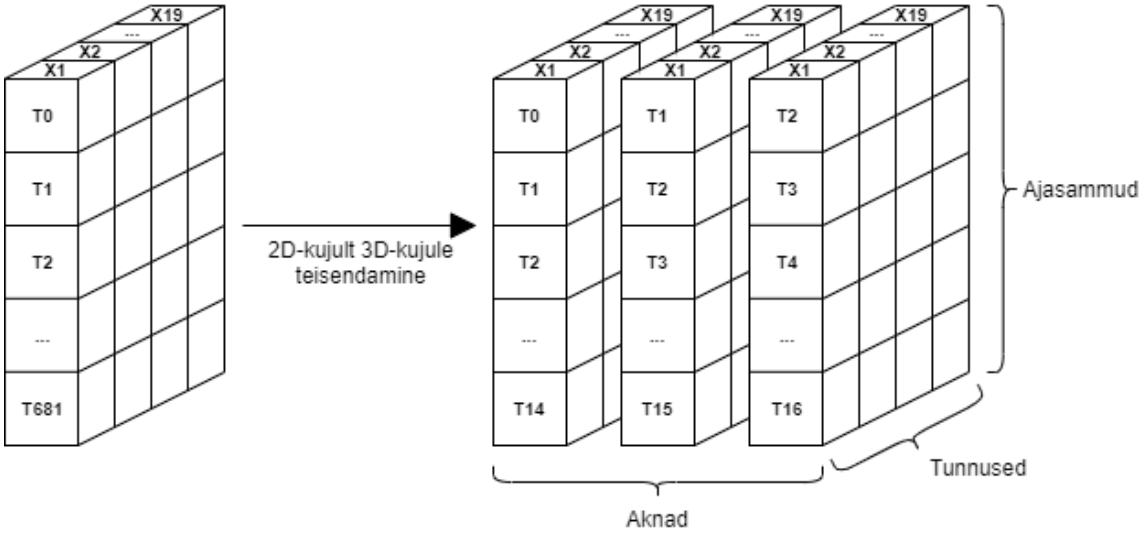
Kuna tehniliste indikaatorite lisamisel tekkisid andmestikus puuduolevad väärtused, siis on andmestik selle võrra lühemaks lõigatud, et puuduolevad väärtused eemaldada. Lõplik andmestik on vahemikus 19.07.2018 – 05.04.2021, koosneb 682-st reast ehk

kauplemispäevast ning 19-st tunnusest. Nimekiri lõpliku andmestiku kõigist tunnustest koos selgitustega on välja toodud Tabelis 6.

Tabel 6. Töödeldud andmestiku tunnuste nimekiri.

Tunnus	**Selgitus**
Open	Hind, millega alustatakse aktsiaga kauplemist antud päeval.
High	Kõrgeim hind, millega päeva jooksul kaubeldi.
Low	Madalaim hind, millega päeva jooksul kaubeldi.
Close	Hind, millega lõpetatakse aktsiaga kauplemine antud päeval.
Adj Close	Sulgemishind, mis võtab arvesse korporatiivseid tegevusi.
Volume	Päeva jooksul kaubeldud väärtpaberite kogus.
RSI_14	RSI indikaator (*Relative Strength Index)* 14 päevase tagasivaate perioodiga.
MACD_12_26_9	MACD indikaator (*Moving Average Convergence Divergence*) 12 ja 26 päevaste perioodidega.
MACDh_12_26_9	MACD indikaatori histogramm.
MACDs_12_26_9	MACD indikaatori signaaljoon.
BBL_5_2.0	Bollinger Band indikaatori (*Bollinger Bands*) alumine joon, 5 päevase perioodiga ning 2,0 suuruse standardhälbega.
BBM_5_2.0	Bollinger Band indikaatori keskmine joon 5 päevase perioodiga ning 2,0 suuruse standardhälbega.
BBU_5_2.0	Bollinger Band indikaatori ülemine joon 5 päevase perioodiga ning 2,0 suuruse standardhälbega.
BBB_5_2.0	Bollinger Band indikaatori ülemise ja alumise joone vaheline laius.
SMA5	SMA indikaator (*Simple Moving Average*) 5 päevase perioodiga.
SMA10	SMA indikaator 10 päevase perioodiga.
EMA5	EMA indikaator (*Exponential Moving Average*) 5 päevase perioodiga.
EMA10	EMA indikaator 10 päevase perioodiga.
Sentiment	Keskmine sentimendiskoor.

Et andmeid rekurrentses närvivõrgus ehk generaatoris sisendina kasutada, siis on sisendandmed kahe dimensiooni asemel teisendatud kolmeks dimensiooniks ehk n-ö "libiseva akna" kujule. See tähendab, et 3D-kujule teisendatud andmestik koosneb 682-st aknast ning iga aken koosneb 19-st tunnusest ja 15-st päevast. Generaatori sisendandmete, ehk X-väärtuste teisendus on graafiliselt kujutatud Joonisel 4.



Joonis 4. Andmestiku teisendamine libiseva akna kujule.

Ennustatavaks väärtuseks ehk Y-väärtuseks on vaid üks tunnus – sulgemishind. Y-väärtus viiakse eelnevalt kirjeldatud põhimõtte alusel samamoodi libiseva akna kujule, kuid erinevus on selles, et lisaks 15 ajasammule on lisatud ka järgmise päeva samm ehk ennustatav väärtus. Samuti on kasutatud abistavat Z-väärtust, mis on sisuliselt sama nagu Y-väärtus, kuid ilma ennustatava väärtuseta. Selle vajalikkus on pikemalt lahti seletatud mudeli arhitektuuri kirjeldava peatüki all. Libiseva akna kujule teisendatud väärtused on välja toodud allolevas Tabelis 7.

Tabel 7. Libiseva akna kujule teisendatud väärtused ning nende kujud.

	Kuju	**Selgitus**
X-väärtus	(667, 15, 19)	667 eksemplari sisendandmeid. Iga eksemplar koosneb 15-st ajasammust ning 19-st tunnusest.
Y-väärtus	(667, 16, 1)	667 eksemplari sulgemishindu. Iga eksemplar koosneb 15-st ajasammust ning järgmisest ehk ennustatavast sammust. Tunnuseks on ainult sulgemishind ehk ennustatav väärtus.
Z-väärtus	(667, 15, 1)	Abiväärtus 667 eksemplariga. Sisuliselt sama, mis Y-väärtus, kuid ilma ennustatava sammuga. Kasutatakse generaatori väljundiga sidumisel, et diskriminaatori sisendina kasutada.

Andmed on jagatud treening- ning testandmeteks suhtega 7:3, mille tulemusel jääb treeningandmete suuruseks 467 akent ning testandmed koosnevad 200-st aknast. Samuti on kõik väärtused skaleeritud vahemikus -1 kuni 1.

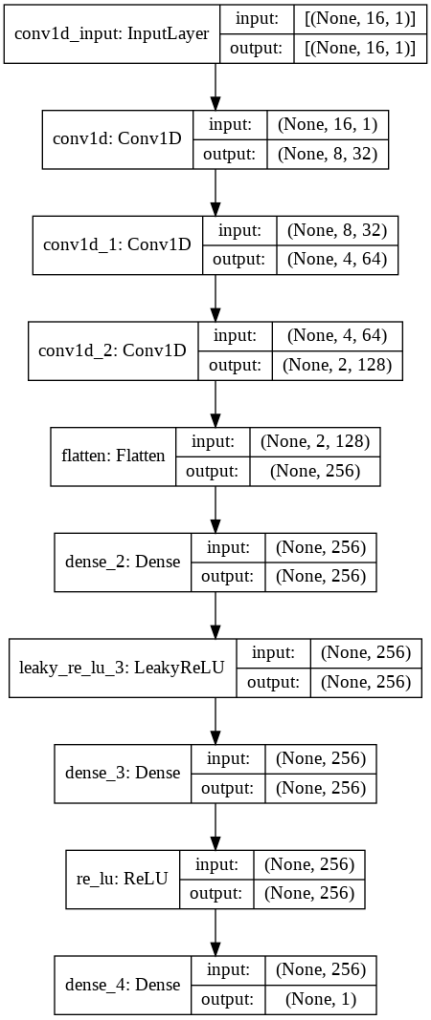
3.2 Mudeli arhitektuur

Lõpliku mudeli treenimisel on kasutatud ploki suurust 64. See tähendab, et andmed tükeldatakse sedasi, et iga plokk koosneb 64-st järjestikusest aknast. Originaalses GAN mudeli uurimistöös on soovitatud ühe iteratsiooni käigus diskriminaatorit treenida 10 korda ning generaatorit ühe korra. Antud töö käigus treenitakse nii generaatorit kui ka diskriminaatorit iteratsiooni käigus vaid üks kord, seda põhjusel, et diskriminaatori sisendandmed on oluliselt väiksemate dimensioonidega. Samuti aitab see kaasa treeningu stabiilsusele ning on väiksem võimalus, et generaator hakkab andmeid ülesobitama. *Gradient penalty* kaalu väärtuseks on jäetud 10 nagu WGAN-GP uurimistöös [24] on soovitatud. Kasutatud on ka samu kaofunktsioone. Nii generaatori kui ka diskriminaatori puhul on kasutatud Adami optimeerijat, mille parameetrid ja nende väärtused on välja toodud Tabelis 8.

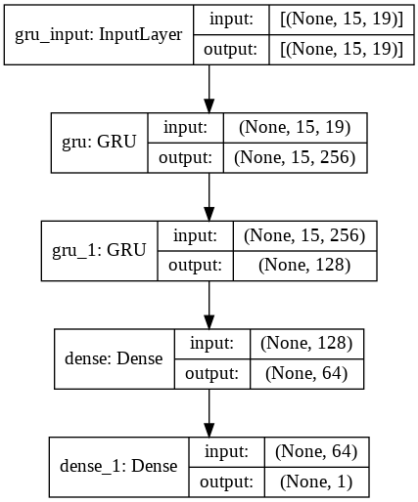
Tabel 8. Adami optimeerijas kasutatud parameetrid ja nende väärtused.

Parameeter	**Väärtus**
learning_rate	0.0002
beta_1	0.9
beta_2	0.999

Diskriminaatori mudel koosneb 10-st kihist ning võtab sisendiks andmeid, mis on Y-väärtuse kujul (ploki suurus, ajasammude arv, tunnuste arv) ning annab väljundi dimensioonidega (ploki suurus, 1). Väljundväärtuseks on tõenäosus, ehk kui suure tõenäosusega on tegemist päris andmetega. Diskriminaatori mudeli arhitektuur on kujutatud Joonisel 5. Generaatori mudel koosneb 5-st kihist, millest 2 on rekurrentsed GRU kihid. Sisendiks on X-väärtused kujul (ploki suurus, ajasammude arv, tunnuste arv) ning väljundi dimensioonideks on (ploki suurus, 1). Väljundväärtuseks on ennustatud sulgemishind. Generaatori mudeli arhitektuur on kujutatud Joonisel 6. Nii generaatori kui ka diskriminaatori mudeli täpsemaid parameetreid on näha vastavalt Lisades 2 ja 3, mis esitavad mõlemat mudelit programmikeeles.



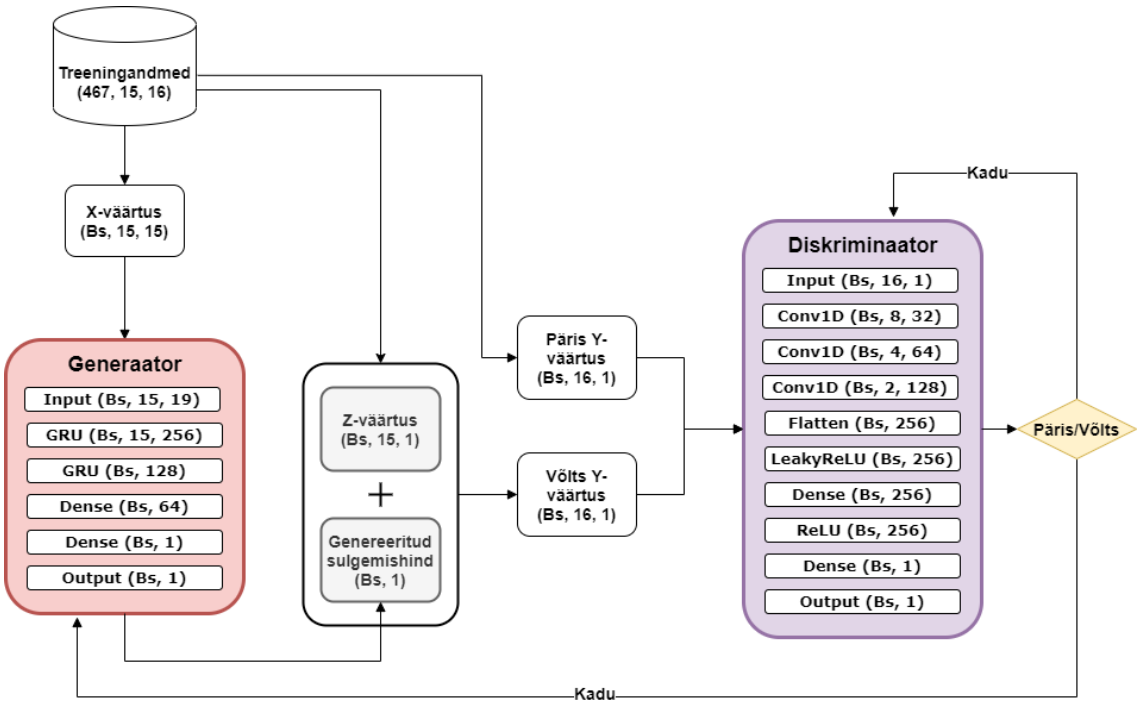
Joonis 5. Diskriminaatori mudeli arhitektuur.



Joonis 6. Generaatori mudeli arhitektuur.

Mudeli üldine arhitektuur erineb mõnevõrra klassikalisest GAN mudelist. Põhiline erinevus seisneb selles, et generaatori sisendina ei kasutata juhusliku müra vektorit. Erinevus seisneb ka andmete olemuses, GAN mudeli algne kasutusala oli piltide genererееrimine, kus generaator genereeris juhusliku müra alusel pildi ning diskriminaator pidi vahet tegema päris ning genereeritud pildil.

Generaator kasutab sisendina eelnevalt mainitud X-väärtusi, mille alusel ennustatakse järgmise päeva sulgemishinda. Ennustatud hind ehk generaatori väljundväärtus liidetakse Z-väärtusega kokku ehk 15 varasema päeva sulgemishinnale liidetakse otsa ennustatud sulgemishind, mis teeb seeria pikkuseks 16 päeva. See kõrvutatakse algandmetest võetud päris sulgemishindadega ehk Y-väärtusega, mis koosneb samast 15 päeva sulgemishinnast nagu Z-väärtus, kuid 16. päev on päris, mitte genereeritud väärtus. Mõlemad seeriad edastatakse diskriminaatorile ning diskriminaator annab hinnangu, kas tegu on päris või valeandmetega. Seejärel arvutatakse kaofunktsioonidega kaod ning nende alusel korrigeeritakse vastavalt generaatori või diskriminaatori kaale. Mudeli arhitektuuri on visuaalselt kujutatud Joonisel 7.



Joonis 7. Loodud GAN mudeli arhitektuur. Lühed Bs tähendab ploki suurust (*batch size*).

On oluline vahet teha, et mudeli treenimisel kasutatakse eelnevalt kirjeldatud arhitektuuri täies ulatuses, kuid valideerimisel ja hilisemates töö etappides ennustuste tegemisel

kasutatakse treenitud generaatori mudelit eraldiseisvalt. Treenimise eesmärk on luua võimalikult hea generaator. Üks oluline omadus, mis generaatoril olema peaks on üldistusvõime. Kuna mudelit treenitakse treeningandmete peal, siis on loogiline, et mudel on treeningandmete peal pisut ülesobitunud ning parimad tulemused tulevadki just treeningandmeid kasutades. Üldistusvõime seisnebki selles, kuidas mudel võõraste andmetega hakkama saab, antud juhul on nendeks valideerimisandmed. Kõik tööga seotud andmestikud ja mudel on saadaval GitHub'i repositooriumis.[1]

3.3 Eksperimendid

Mudeliga viidi läbi kaks eksperimenti, et kindlaks teha, kuidas teatud tunnuste kaasamine ennustusjõudlust mõjutab. Esimeses eksperimendis vaadeldakse, kuidas sentimendi ja tehnilinste indikaatorite kasutamine ennustusjõudlusele mõjub. Teises eksperimendis vaadeldakse tehniliste indikaatorite mõju ennustusjõudlusele põhjalikumalt pikema perioodi andmeid kasutades. Kuna esimeses eksperimendis oli sentimendi andmete vähesus piiranguks pikema ajaperioodi valimisel, siis teises eksperimendis seda probleemi ei ole ning kasutatud andmete puhul on valitud pikem ajaperiood. Mõlema eksperimendi puhul on kõikide mudelite treenimiseks kasutatud eelnevalt kirjeldatud treeningparameetreid ja treeningu pikkuseks on 750 epohhi.

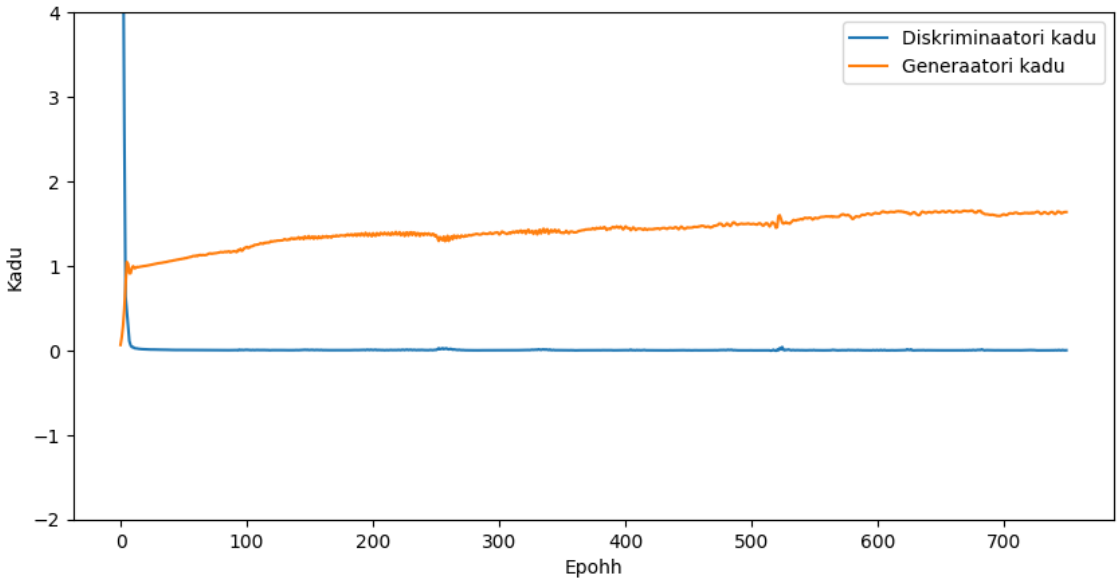
Eksperimentide käigus vaadeldakse mudelite treenimise protsesse ja valitakse välja parim mudel, mida järeltestimise etapis kasutatakse. Treenimise käigus on oluline eristada treening- ja valideerimisandmeid. Treeningu käigus mudel valideerimisandmetele ligi ei pääse, valideerimist kasutatakse iga epohhi lõpus veendumaks, et generaator on saavutanud optimaalse üldistusvõime ning suudab võõraste andmete korral sulgemishinda edukalt ennustada. Treening-RMSE väärtusele erilist tähelepanu ei pöörata, oluline on vaid see, et vastav väärtus oleks madalam kui valideerimis-RMSE. Sellega on tagatud see, et mudel on treeningandmete puhul pisut ülesobitatud, kuid on sellegipoolest piisava üldistusvõime saavutanud, et ka võõraste andmetega hea tulemus saavutada.

[1] https://github.com/danaju/Aktsiahindade-ennustamine-GAN

3.3.1 Esimene eksperiment

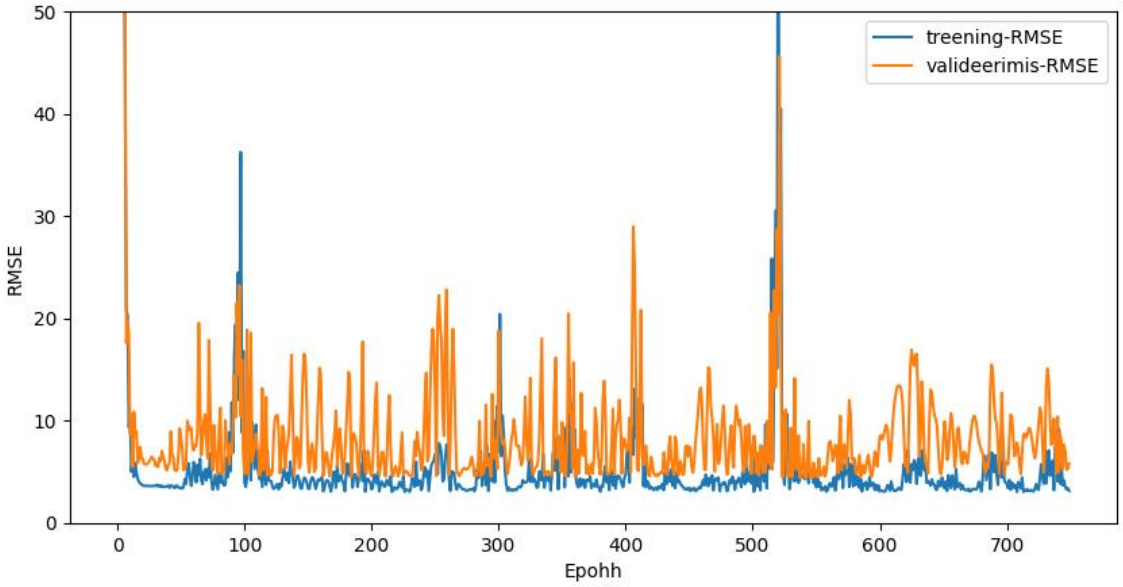
Esimeses eksperimendis on vaatluse all sentimentaalanalüüsi ja tehniliste indikaatorite mõju. Lähtepunktiks on vaid aktsiahinna andmete peal treenitud mudel ning selle tulemusi võrreldakse mudelitega, mis kasutavad aktsiahinna andmeid koos sentimendiga, aktsiahinna andmeid koos tehniliste indikaatoritega ja kõiki sisendeid korraga.

Esimene mudel on lähtepunktiks võrdlemaks sentimendi ning tehniliste indikaatorite mõju ennustuse kvaliteedile. Parim valideerimis-RMSE saavutati 538. epohhil ning selle väärtuseks oli 4,4238. Treening-RMSE sama epohhi kohta oli 4,2231. Joonis 8 peal on näha, et diskriminaatori kadu käitub ootuspäraselt – saavutab nullilähedase väärtuse. Kuid kuna diskriminaator saavutab selle nullilähedase väärtuse üsna kiirelt, siis võib tegu olla metoodika osas kirjeldatud *Failure to Converge* probleemiga. Ehk diskriminaator ei suuda enam vahet teha päris ning genereeritud andmetel ja ei anna enam generaatorile sisulist tagasisidet, mille tulemusena generaatori kvaliteet hakkab vähehaaval langema. Kuna parim tulemus saavutati 538. epohhil, siis võib järeldada, et generaatori kvaliteet oluliselt langenud ei ole. Samas võib tegu olla ka asjaoluga, et diskriminaator endiselt vähehaaval õpib paremini vahet tegema õigetel ja valedel andmetel ning seetõttu ka generaatori kadu suureneb. Kuna GAN mudel on olemuselt selline, kus võrgud võistlevad üksteise vastu, siis ühe võrgu kao vähenedes teise oma suureneb. Tüüpiliselt peaksid mõlemad kaod kindla väärtuse saavutama ning stabiliseeruma. Generaatori kao suurus võib mingil määral varieeruda – suurem vahe diskriminaatori kaoga ei pruugi tähenda kehvemat mudelit.



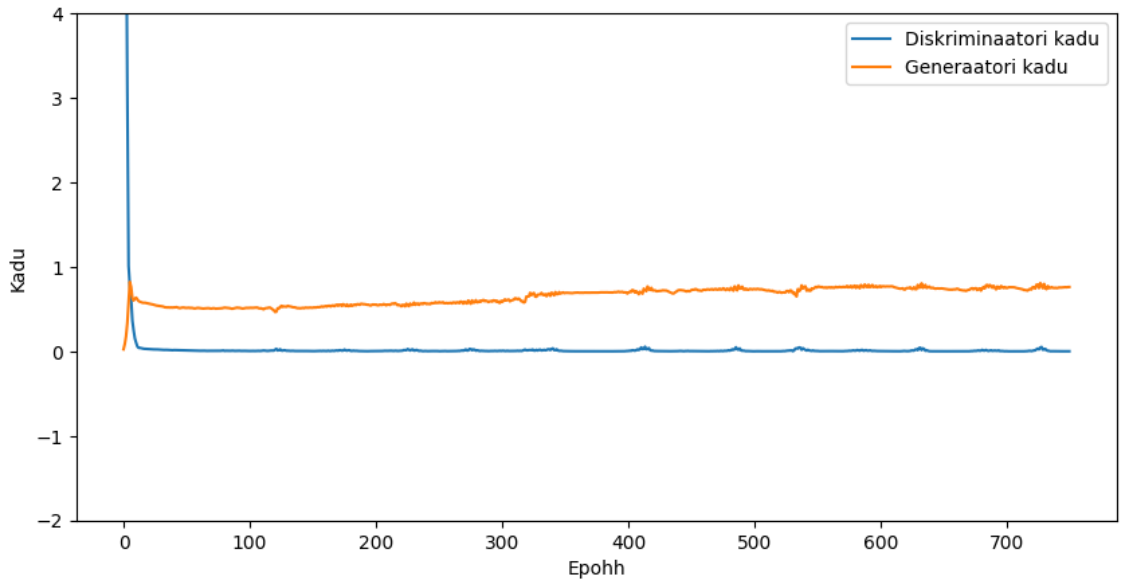
Joonis 8. Kaod aktsiahinna andmeid kasutades esimeses eksperimendis.

Joonis 9 pealt on näha, et mudeli treenimisel on treeningandmete RMSE-väärtuste puhul olnud vaid mõned üksikud suuremad kõikumised. Valideerimisandmetel on kõikumised suuremad, kuid üldiselt saab treeningu kulgu pidada stabiilseks.



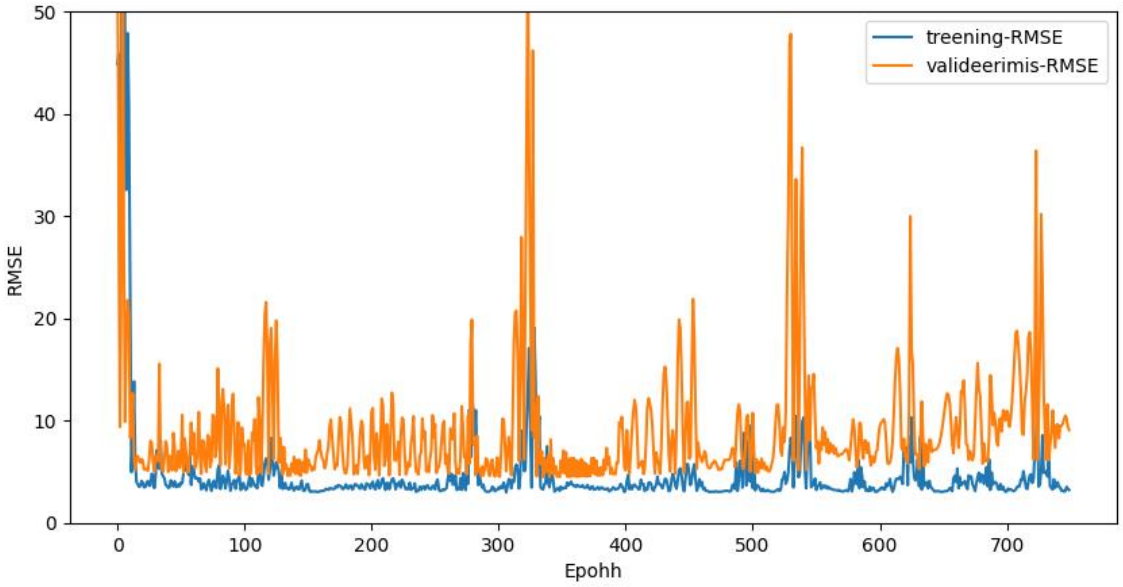
Joonis 9. RMSE aktsiahinna andmeid kasutades esimeses eksperimendis.

Teine mudel on treenitud kasutades aktsiahinna andmeid koos sentimendiga. Parim RMSE väärtusega 4,4691 saavutati 336. epohhil. Treening-RMSE väärtus sama epohhi puhul oli 3,2655. Joonis 10 näitab, et generaatori kadu on treeningu keskel stabiliseerinud ja kindla väärtuse juurde jäänud, mis võib olla märk sellest, et mudel on leidnud optimaalse punkti ja ei suuda ennast enam parandada.



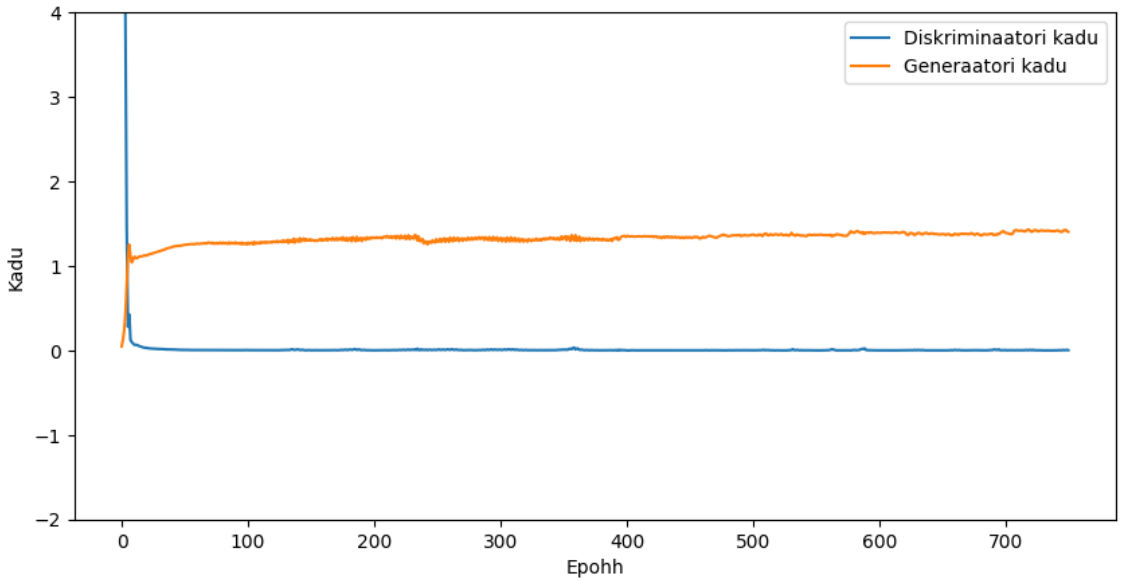
Joonis 10. Kaod aktsiahinna andmeid ja sentimenti kasutades esimeses eksperimendis.

Joonis 11 näitab, et mudel on hilisemate epohhide puhul hakanud liialt ülesobitama ehk treening- ja valideerimis-RMSE vahe on hakanud suurenema. Ehk paistab, et kui mudel leiab optimaalse punkti, kus ta enam juurde ei suuda õppida, siis ta hakkab ülesobitama.



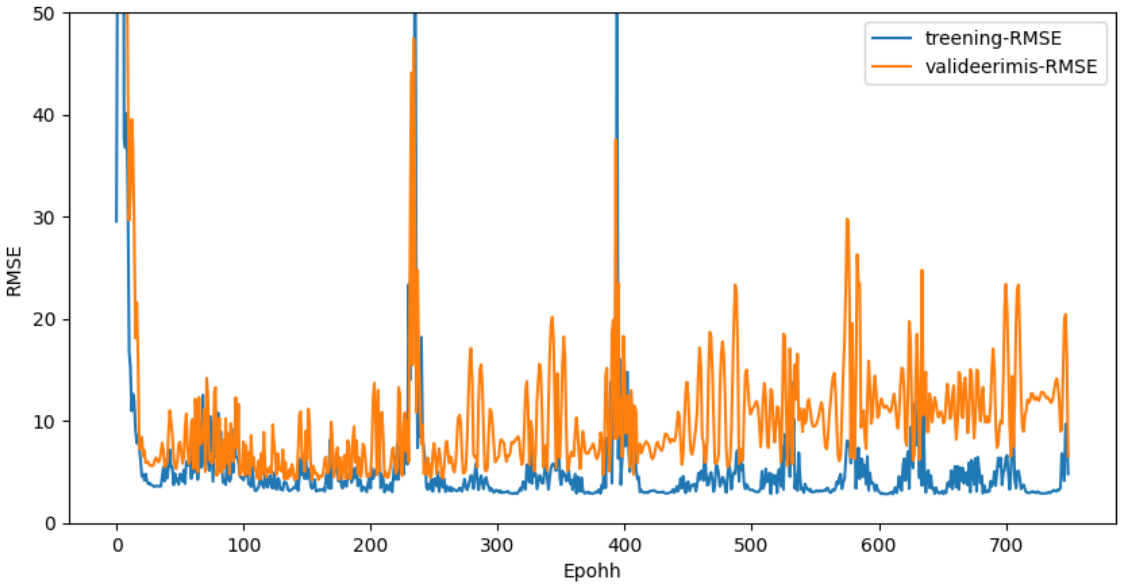
Joonis 11. RMSE aktsiahinna andmeid ja sentimenti kasutades esimeses eksperimendis.

Kolmas mudel kasutab aktsiahinna andmeid koos tehniliste indikaatoritega, sentimenti kasutatud ei ole. Parim valideerimis-RMSE saavutati 160. epohhil väärtusega 4,2259, treening-RMSE samal epohhil oli 3,1793. Joonis 12 peal on näha, et generaatori kadu on eelmiste eksperimentidega võrreldes oluliselt kiiremini mingi kindla väärtuse juurde koondunud, mis eeldaks et mudel on kiiremini optimaalse punkti leidnud ja ka kiiremini ülesobitama hakanud.



Joonis 12. Kaod aktsiahinna andmeid ja tehnilisi indikaatoreid kasutades esimeses eksperimendis.

Joonis 13 pealt on näha, et treeningu teises pooles on rohkem ülesobitamist esinenud, teiste mudelite puhul nii palju ülesobitamist ei esinenud. Saab järeldada, et tänu tehnilistele indikaatoritele leiab mudel kiiremini üldistamiseks sobivad kaalud üles ning seejärel hakkab andmeid ülesobitama, mille tõttu ka valideerimis-RMSE tõuseb. Märgiks taolisest käitumisest on ka see, et parim tulemus saavutati 160. epohhil, teiste mudelite puhul saavutati tulemus vastavalt 538. ja 336. epohhil ehk oluliselt hilisemalt.



Joonis 13. RMSE aktsiahinna andmeid ja tehnilisi indikaatoreid kasutades esimeses eksperimendis.

Kokkuvõtvalt on esimese eksperimendi tulemused kujutatud Tabelis 9 – välja on toodud parima generaatori mudeli treening- ja valideerimis-RMSE ning ka epohhi number, kus parim tulemus saavutati.

Tabel 9. Esimese eksperimendi tulemused.

Mõõdikud	**Aktsiahinna andmed**	**Aktsiahinna andmed + sentiment**	**Aktsiahinna andmed + tehnilised indikaatorid**
Treening-RMSE	4,2230	3,2655	3,1793
Valideerimis-RMSE	4,4238	4,4691	4,2259
Epohh	538	336	160

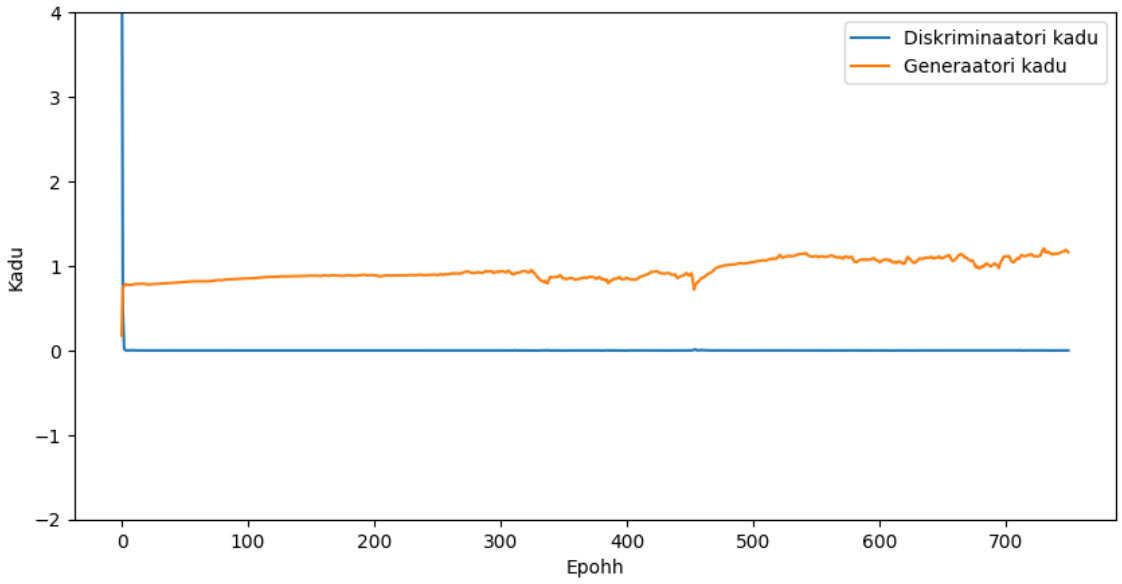
Parim valideerimis-RMSE saavutati tehnilisi indikaatoreid kasutades ning kõige kehvem tulemus tuli sentimenti kasutades. Vaid aktsiahinna andmeid kasutades saavutati sentimenti kasutanud mudelile üsna sarnane valideerimis-RMSE, kuid huvitaval kombel oli vaid aktsiahinna andmeid kasutanud mudeli treening-RMSE oluliselt kõrgem kui seda oli sentimenti kasutanud mudelil. Antud tulemuste põhjal saab järeldada, et sentimendi kasutamine mingit eelist ei anna. Samuti võiks järeldada, et tehnilised indikaatorid annavad mudeli treenimisel paremad tulemused, kuid täpsemalt on seda vaadeldud järgmises eksperimendis.

3.3.2 Teine eksperiment

Teise eksperimendi puhul sentimendi andmeid kaasatud ei ole. Kuna sentimentaalanalüüsi jaoks olevaid andmeid oli saadaval vaid piiratud koguses ning aktsiahinna andmetest puudust ei tule, siis on kasutatud pikemat perioodi. Kasutatud on samu Microsofti aktsiahindade andmeid, kuid 10 aastase perioodiga ehk vahemikus 05.04.2011 – 01.04.2021. Pikema perioodi kasutamine võimaldab andmed kolmeks jagada – treening-, valideerimis- ja testandmed. Treening- ja valideerimisandmeid kasutatakse samamoodi nagu eelmises eksperimendis, kuid testandmeid teise eksperimendi raames ei kasutata. Testandmeid kasutatakse hilisemas järeltestimise etapis, et ei tekiks olukorda, kus järeltestimine toimub samade andmete peal (valideerimisandmed), mille alusel parim mudel välja valiti. Sellest tulenevalt on andmed jagatud treening-, valideerimis-, ning testandmeteks suhtega 6:2:2, ehk treenimisel

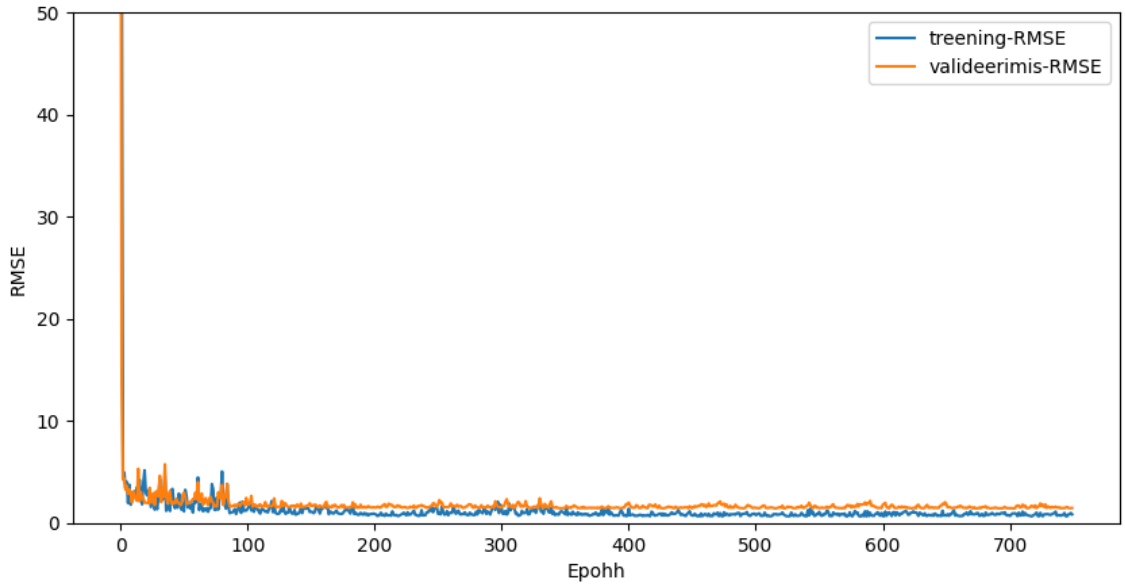
kasutatakse umbes 6 aasta andmeid, valideerimisel 2 aasta andmeid ning hilisemas järeltestimise faasis samuti 2 aasta andmeid. Muid parameetreid mudeli juures muudetud ei ole.

Teise eksperimendi esimene mudel on treenitud kasutades vaid aktsiahindade andmeid. Parim valideerimis-RMSE väärtusega 1,4231 saavutati 523. epohhil ning treening-RMSE sama epohhi jaoks oli 0,8221. Joonisel 14 on näha, et diskriminaatori kadu läheneb kiiremini nullile kui esimese eksperimendi mudelite puhul. See on tingitud sellest, et antud juhul on kasutatud oluliselt rohkem andmeid, mis lubab mudelil epohhi jooksul rohkem õppida. Samuti on näha, et generaatori kadu on esimeses treeningu pooles stabiilsem kui teises.



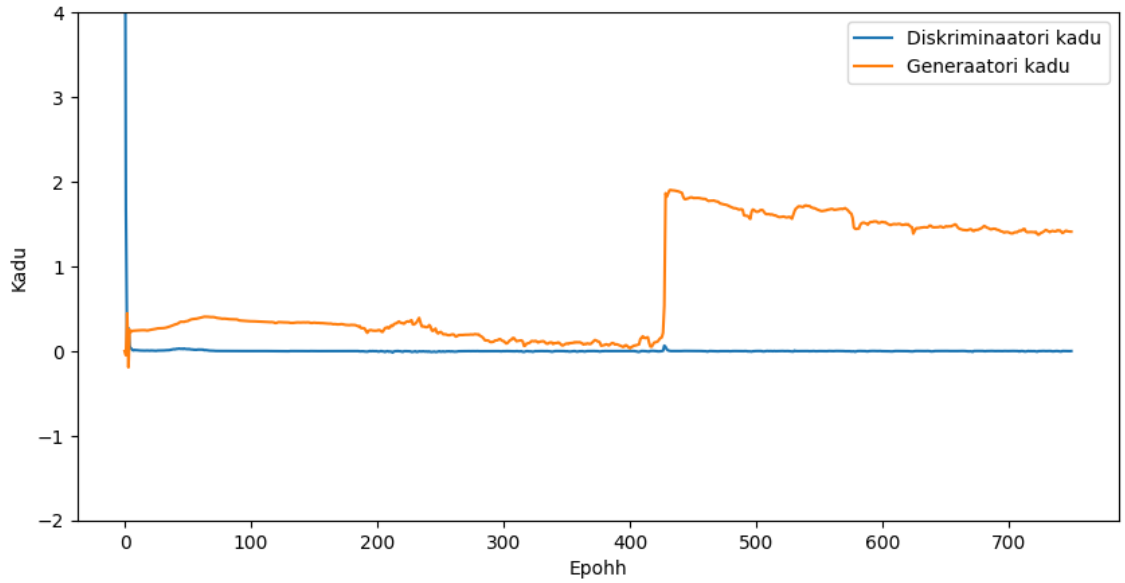
Joonis 14. Kaod aktsiahinna andmeid kasutades teises eksperimendis.

Joonis 15 näitab, et esimese 100 epohhi jooksul on mudel mõnevõrra ebastabiilsem kui hilisemas treeningu faasis. Samas, kui võrrelda, et generaatori kadu on treeningu teises pooles ebastabiilsem olnud, siis RMSE-de poolest seda väita ei saa. On raske öelda, millest selline käitumine tingitud võib olla.



Joonis 15. RMSE aktsiahinna andmeid kasutades teises eksperimendis.

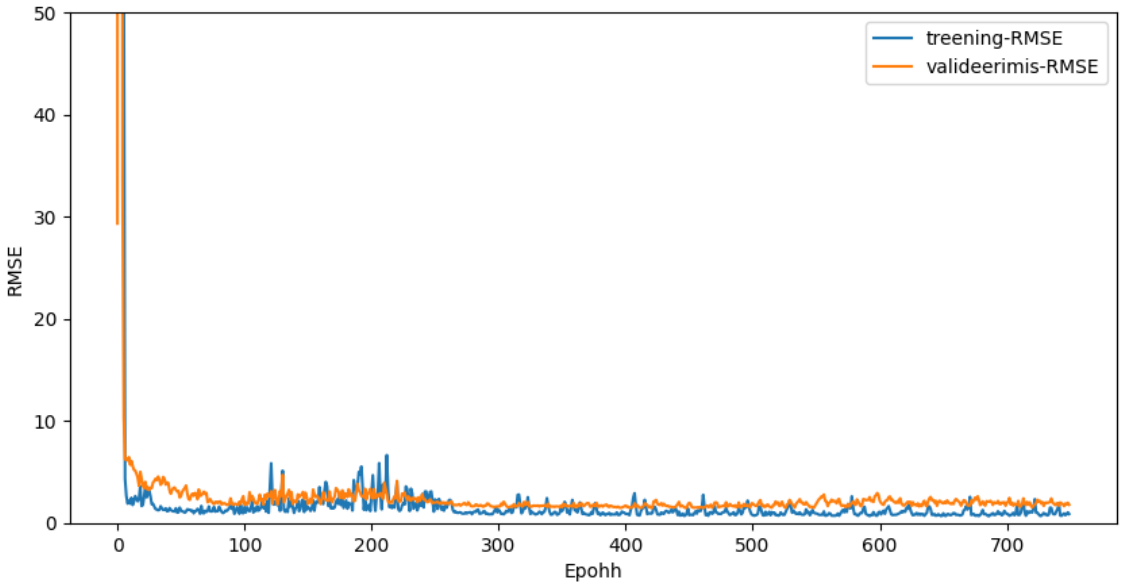
Teine mudel on treenitud kasutades aktsiahinna andmeid koos tehniliste indikaatoritega. Parim valideerimis-RMSE saavutati 549. epohhil väärtusega 1,4701. Treening-RMSE sama epohhi jaoks oli 1,0237. Joonisel 16 on näha, et generaatori kadude käitumine on mõnevõrra ootamatu.



Joonis 16. Kaod aktsiahinna andmeid ja tehnilisi indikaatoreid kasutades teises eksperimendis.

Kuid Joonisel 17 ei paista, et sama epohhi juures suurem kõikumine või ebastabiilsus oleks olnud. Siiski võrreldes eelneva mudeliga on näha, et treening- ja valideerimis-

RMSE vahe on treeningu lõpu poole suurem. Sellegipoolest jääb arusaamatuks, millest on tingitud taoline generaatori kadude kõikumine.



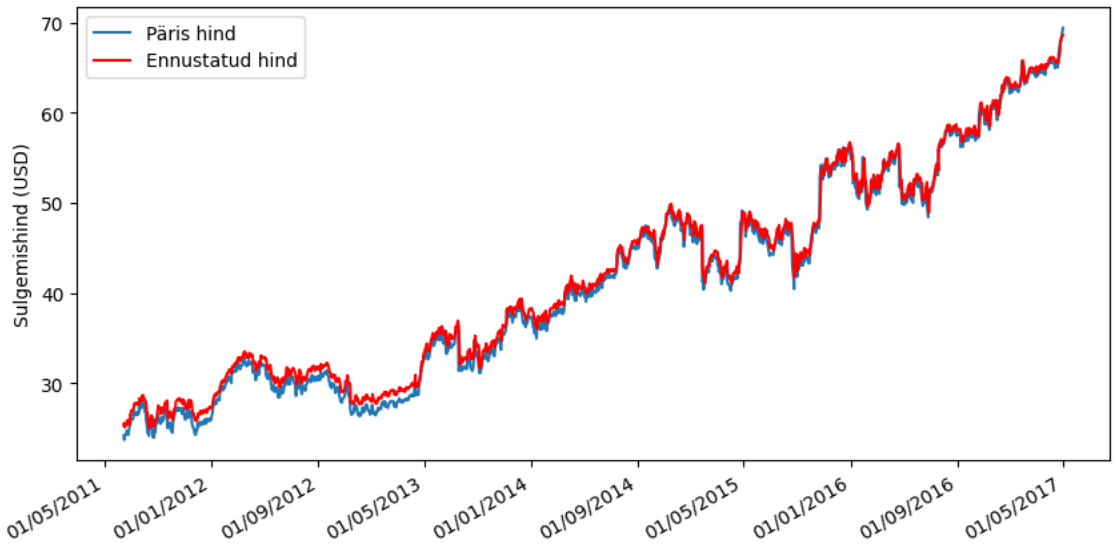
Joonis 17. RMSE aktsiahinna andmeid ja tehnilisi indikaatoreid kasutades teises eksperimendis.

Tabelis 10 on välja toodud teise eksperimendi tulemused. Nii treening-RMSE kui ka valideerimis-RMSE poolest saavutas parima tulemuse mudel, mis kasutas vaid aktsiahinna andmeid. Kuid arvestades, et mõlema mudeli kadude puhul esines ebatavalist käitumist, siis ei saa eksperimenti edukaks lugeda.

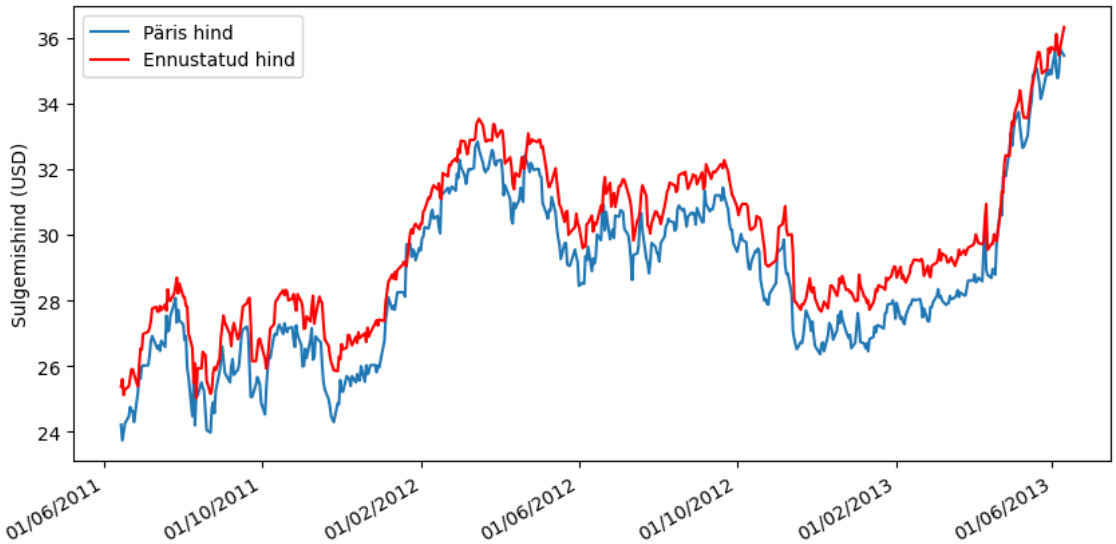
Tabel 10. Teise eksperimendi tulemused.

Mõõdikud	**Aktsiahinna andmed**	**Aktsiahinna andmed + tehnilised indikaatorid**
Treening-RMSE	0,8221	1,0236
Valideerimis-RMSE	1,4231	1,4701
Epohh	523	549

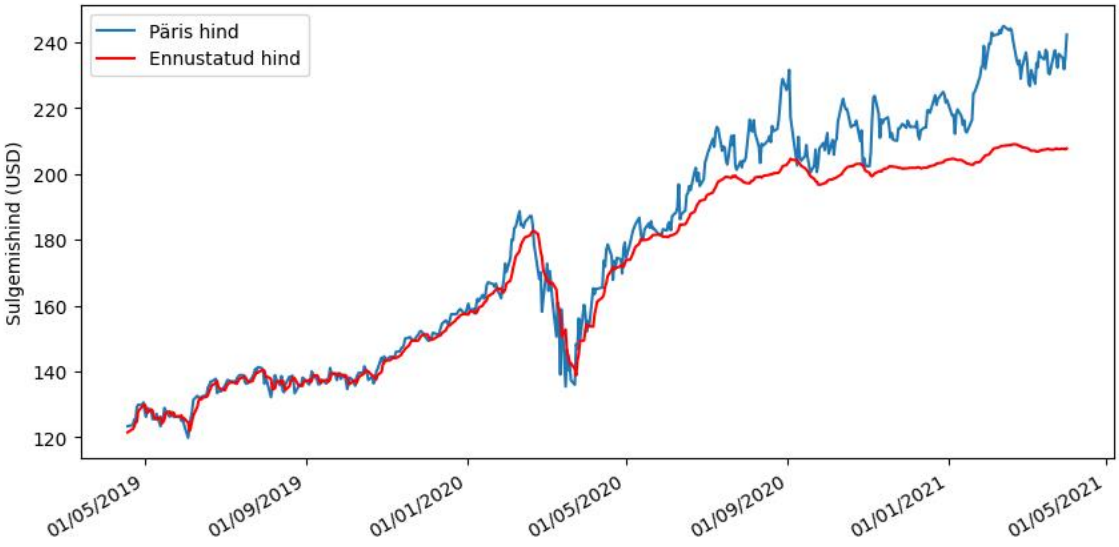
Kui lähemalt vaadelda parima mudeli ennustusi treeningandmetel (Joonis 18 ja 19), siis on näha, et madalamata sulgemishindade juures esineb olukord, kus ennustatud hind on pidevalt kõrgem päris hinnast. Ning kui vaadelda seda, kuidas mudel käitub testandmetel (Joonis 20), siis on näha, et mudel ei saa hakkama ka kõrgema väärtustega sulgemishindade ennustamisega ehk järeltestimiseks antud mudel sobilik ei ole.



Joonis 18. Teise eksperimendi parima mudeli ennustused treeningandmetel.



Joonis 19. Teise eksperimendi parima mudeli ennustused treeningandmetel suurendatuna.

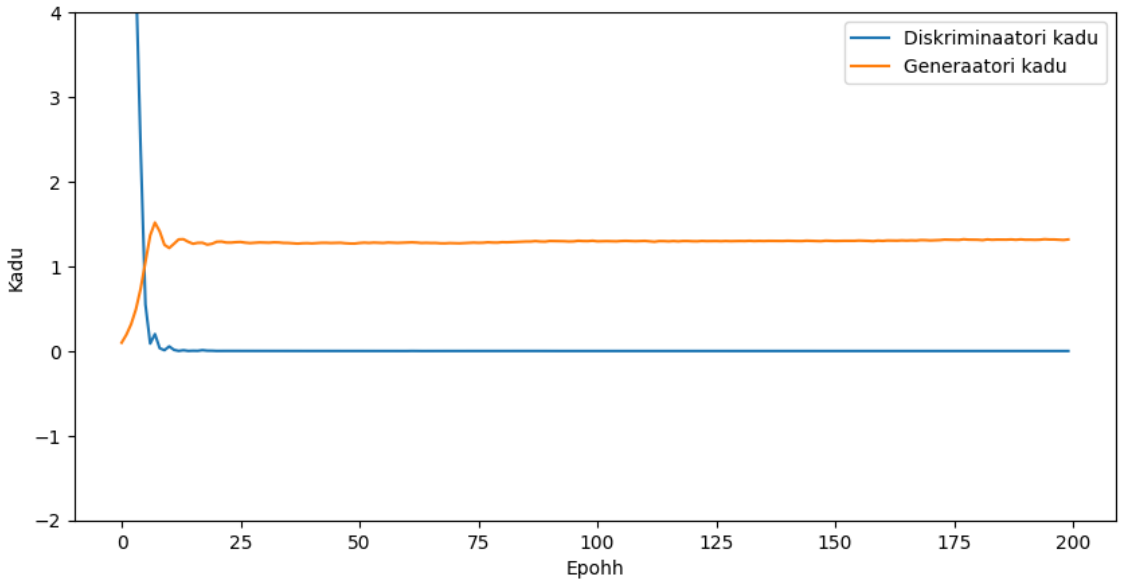


Joonis 20. Teise eksperimendi parima mudeli ennustused testandmetel.

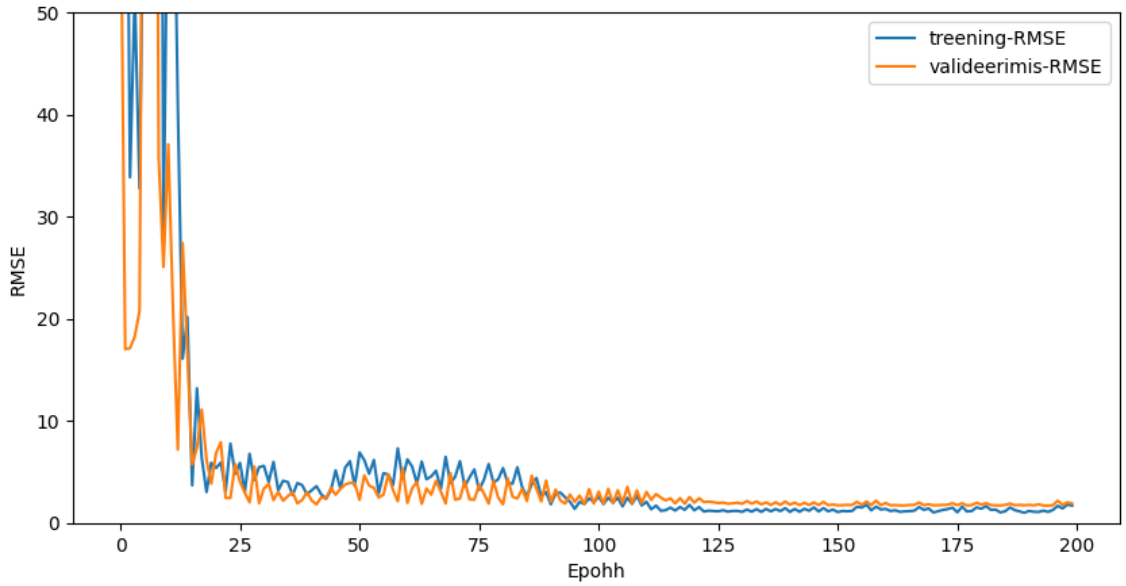
3.3.3 Kolmas eksperiment

Kolmas eksperiment on sisuliselt sama nagu teine eksperiment – eesmärk on vaadelda tehniliste indikaatorite mõju ning leida sobiv mudel järeltestimiseks. Kuna eelnevas eksperimendis käitusid mudelid treenimisel ebatavaliselt ning ennustuste kvaliteet oli kehv, siis otsustas autor viia läbi ka kolmanda eksperimendi, kus on üritatud eelnevaid probleeme vältida. Probleemide vältimiseks on muudetud mõningaid hüperparameetreid – ploki suurust on suurendatud 64 pealt 256 peale ning epohhide arv on 750 pealt langetatud 200 peale. Kõik muu on sama nagu eelmises eksperimendis.

Esimese mudeli puhul, mis kasutab vaid aktsiahinna andmeid on Joonisel 21 märgata oluliselt stabiilsemat käitumist kui seda oli teiste eksperimentide mudelitel. Parim valideerimis-RMSE saavutati 195. epohhil ning selle väärtuseks oli 1,7050. Treening-RMSE samal epohhil oli 1,0823. Joonis 22 näitab samuti, et generaator on treeningu teises pooles optimaalsed kaalud leidnud.

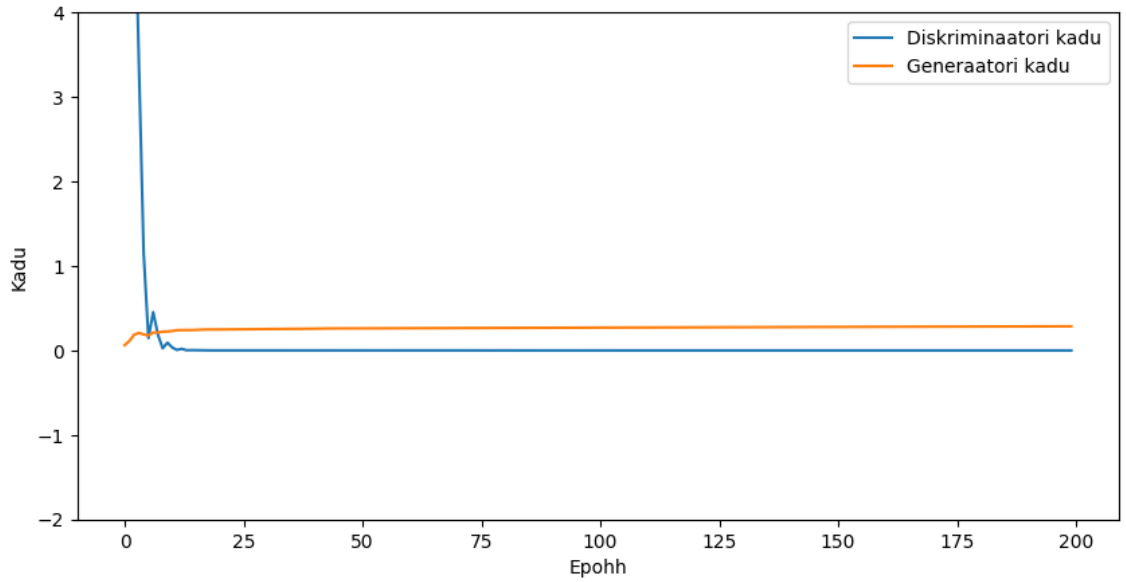


Joonis 21. Kaod aktsiahinna andmeid kasutades kolmandas eksperimendis.

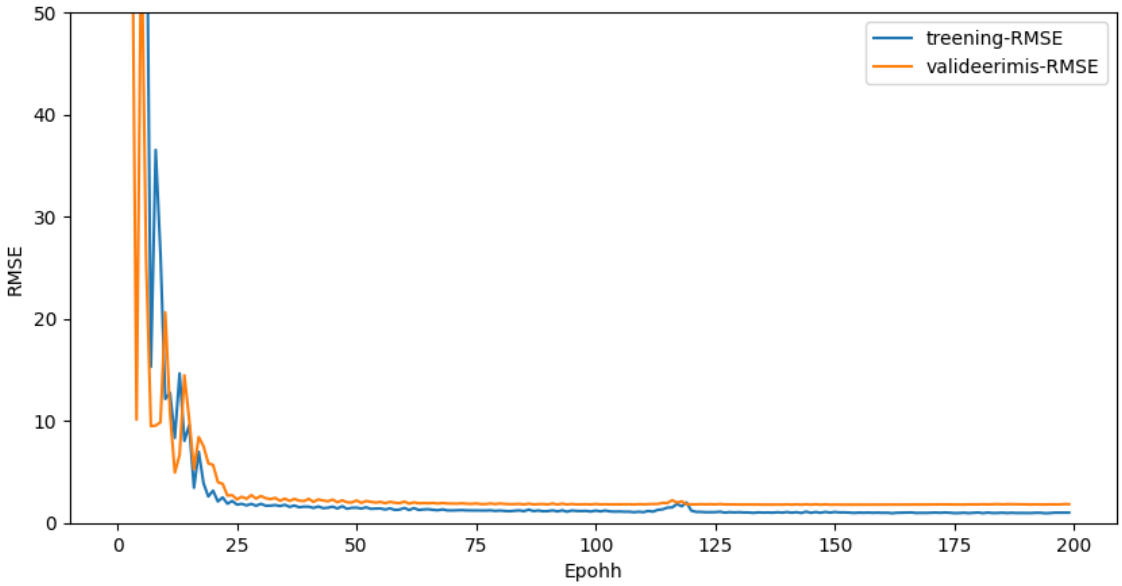


Joonis 22. RMSE aktsiahinna andmeid kasutades kolmandas eksperimendis.

Teine mudel, mis kasutas ka tehnilisi indikaatoreid, saavutas parima valideerimis-RMSE 150. epohhil väärtusega 1,7895. Treening-RMSE sama epohhi jaoks oli 1,0205. Sarnaselt eelmisele mudelile, on Joonis 23 peal näha, et treenimisel on kiirelt õiged kaalud leitud ning kaod on stabiilsed väärtused leidnud. Suurim erinevus võrreldes eelneva mudeliga ilmneb Joonisel 24, kus on näha, et treening- ja valideerimis-RMSE stabiliseeruvad oluliselt kiiremini. See tähendab, et tehnilisi indikaatoreid kasutades jõuab mudel treenimisel kiiremini optimaalsesse punkti.



Joonis 23. Kaod aktsiahinna andmeid ja tehnilisi indikaatoreid kasutades kolmandas eksperimendis.



Joonis 24. RMSE aktsiahinna andmeid ja tehnilisi indikaatoreid kasutades teises eksperimendis.

Läbiviidud eksperiment on näide sellest, kui oluline on õigete parameetrite valik GAN arhitektuuril põhineva mudeli treenimisel. Kolmanda eksperimendi tulemused on kujutatud Tabelis 11.

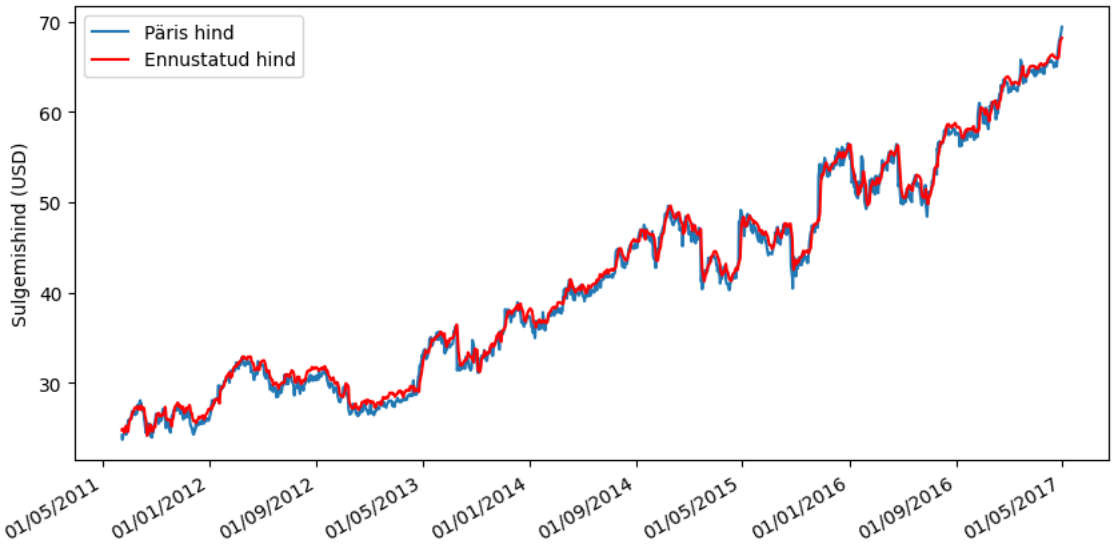
Tabel 11. Kolmanda eksperimendi tulemused.

Mõõdikud	**Aktsiahinna andmed**	**Aktsiahinna andmed + tehnilised indikaatorid**
Treening-RMSE	1,0823	1,0205
Valideerimis-RMSE	1,7050	1,7895
Epohh	195	150

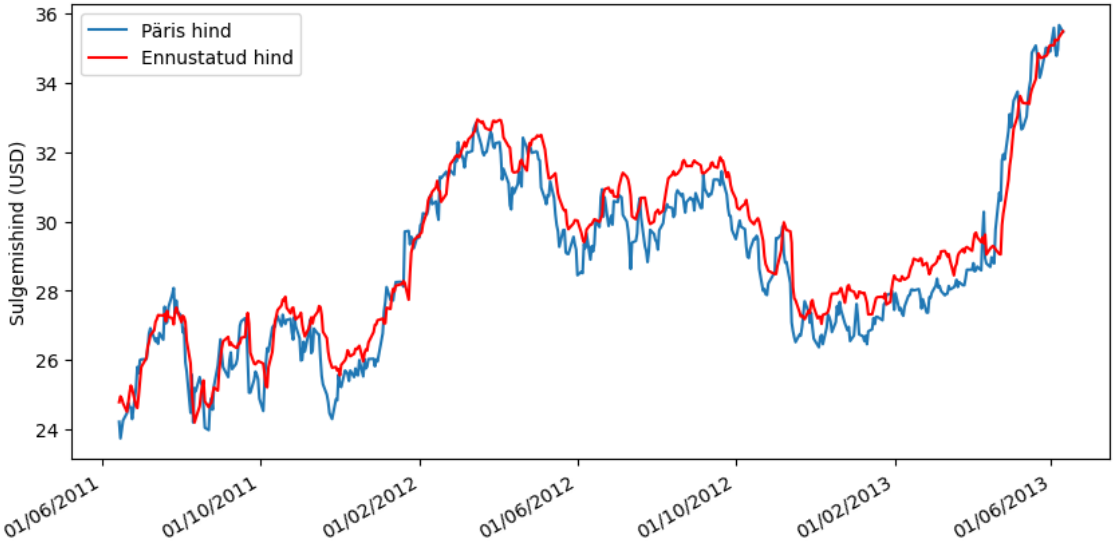
Parima tulemuse valideerimis-RMSE poolest saavutas ilma tehniliste indikaatoriteta mudel, kuid treening-RMSE poolest oli tehnilisi indikaatoreid kasutanud mudel parem. Kuna oli näha, et tehniliste indikaatorite kasutamine kiirendab mudelil optimumi jõudmist, siis võib põhjuseks olla see, et mudel hakkab kiiremini ülesobitama ning ei suuda selle ajaga paremat valideerimis-RMSE väärtust leida ning seetõttu ongi valideerimis-RMSE poolest kehvem kui ilma tehniliste indikaatoriteta mudel. Sellest tulenevalt ei saa kindlalt väita, et tehniliste indikaatorite kasutamine parandaks ennustuste kvaliteeti. Kui leida mudelile sobivad parameetrid, mis ülesobitamist niivõrd ei

soodustaks, saaks täpsema ülevaate. Küll aga oli kõigi kolme eksperimendi puhul näha, kuidas tehnilisi indikaatoreid kasutavad mudelid hakkasid teistest kiiremini ülesobitama, ehk nende kasutamine aitab kiirendada treeningu protsessi ja sobivate kaalude leidmist.

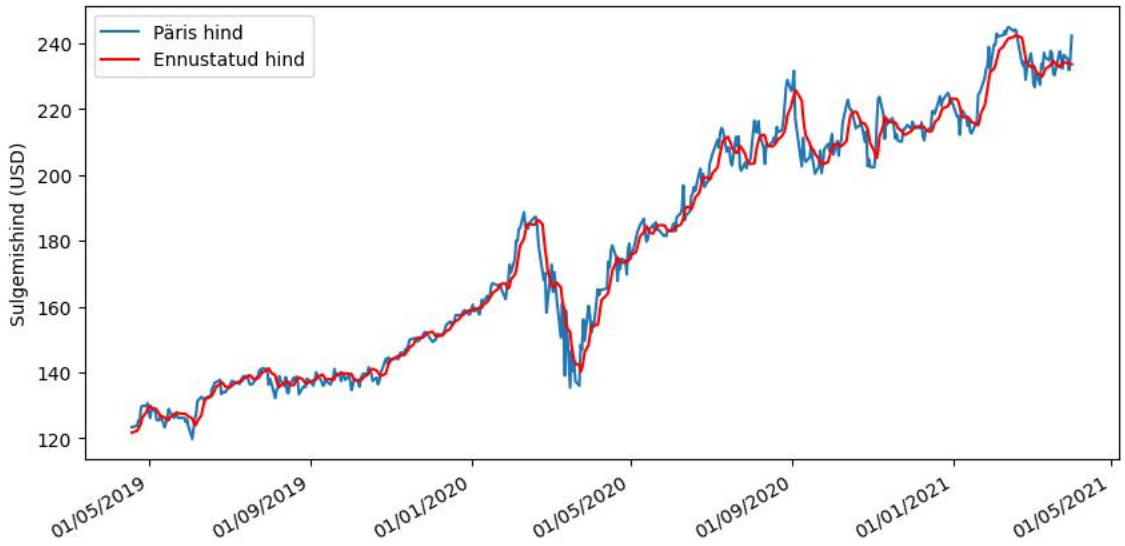
Joonistel 25 ja 26 on näha et treeningandmete madalate väärtuste puhul on olukord võrreldes teise eksperimendi tulemustega paranenud ning mudel saab madalamate hindade ennustamisega paremini hakkama. Kui vaadelda ennustusi testandmetel (Joonis 27), siis on näha, et mudel on nüüd võimeline hakkama saama ka kõrgemate sulgemishindadega.



Joonis 25. Kolmanda eksperimendi parima mudeli ennustused treeningandmetel.



Joonis 26. Kolmanda eksperimendi parima mudeli ennustused testandmetel.



Joonis 27. Kolmanda eksperimendi parima mudeli ennustused testandmetel.

3.4 Järeltestimine

Järeltestimisel on kasutatud kolmanda eksperimendi parimat mudelit (ilma tehniliste indikaatoriteta), ning test on läbiviidud eelnevalt nimetatud testandmete peal, mille vahemikuks on 18.04.2019 – 05.04.2021 ehk kokku 714 päeva. Kõigi järeltestide puhul on algkapitaliks määratud 10000 USA dollarit.

Loodud strateegia on olemuselt lihtne ning kasutab peamise muutujana lävendit, mida testide jooksul muudetakse. Teise muutujana kasutatakse muutu, mis mõõdab mitu protsenti suurem/väiksem on ennustatud hind sulgemishinnast. Strateegias arvestatakse ka positsiooni – peale väärtpaberite ostmist määratakse positsioon pikaks (*long*) ning ja peale müümist lühikeseks (*short*). Tingimusel, et muut on lävendist suurem ja positsioon ei ole pikk, ostetakse kogu olemasoleva kapitali eest väärtpabereid. Ning kui muut on negatiivsest lävendist väiksem ja positsioon ei ole lühike, siis müüakse kõik omatud väärtpaberid. Ehk kui lävendiks oleks 0,5%, siis ostutehing tehakse kui ennustatud hind on sulgemishinnast vähemalt 0,5% võrra kõrgem. Ning müügitehing tehakse, kui ennustatud hind on sulgemishinnast vähemalt 0,5% võrra madalam.

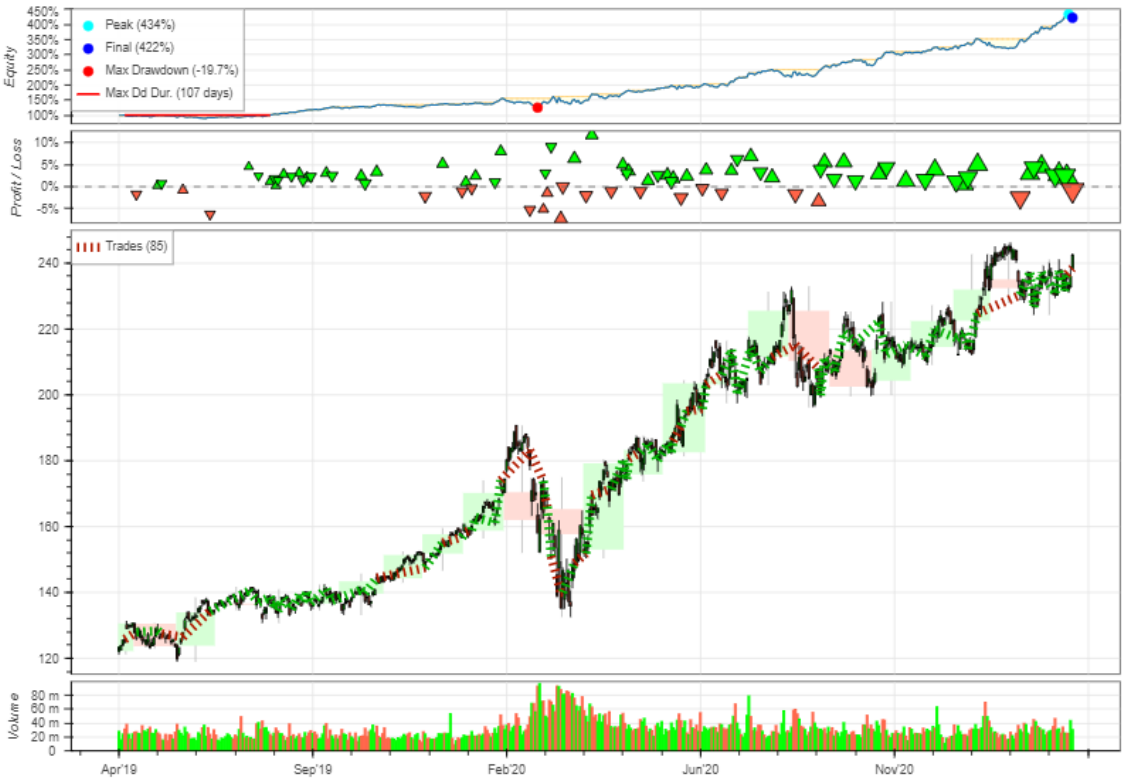
Mudelite puhul on lävenditeks valitud viis erinevat väärtust. Esimeses järeltestis on kasutatud kolmanda eksperimendi parima valideerimis-RMSE väärtusega mudelit (ilma tehniliste indikaatoriteta). Tabelis 12 on kujutatud järeltestimise tulemused.

Tabel 12. Järeltestid testandmestiku (Microsoft) peal.

Mõõdik	**Lävend 0%**	**Lävend 0,25%**	**Lävend 0,5%**	**Lävend 0,75%**	**Lävend 1%**
Equity Final [$]	23183	24327	29496	**42174**	33769
Equity Peak [$]	23183	24327	29496	**43359**	34709
Return [%]	131.83	143.28	194.96	**321.75**	237.69
Buy & Hold Return [%]	96.44	96.44	96.44	**96.44**	96.44
Sharpe Ratio	1.046	1.090	1.268	**1.506**	1.336
Max. Drawdown [%]	-20.03	-19.98	-20.19	**-19.72**	-19.74
Number of trades	151	125	111	**85**	63
Win Rate [%]	66.89	66.40	72.97	**75.29**	69.84
Best Trade [%]	7.53	7.53	7.53	**11.68**	13.51
Worst Trade [%]	-7.97	-9.05	-10.50	**-7.28**	-7.28
Avg. Trade [%]	0.563	0.718	0.985	**1.715**	1.958
Avg. Trade Duration	5 days	6 days	7 days	**9 days**	12 days

Tulemuste põhjal on näha, et lihtsat strateegiat kasutades on mudel võimeline lööma osta ja hoia strateegiat. Osta ja hoia strateegia saavutas antud perioodi jooksul 96,4% suuruse tootluse. Kasutades lävendit 0,75% suutis mudel seda rohkem kui kolmekordselt ületada saavutades tootluseks 321,7%. Positiivne on ka see, et kõigi lävendite puhul oli kasumlike tehingute arv üle 66%, parima lävendi puhul lausa 75%. Samas on ka näha, et suurim kaotus (*Max Drawdown*) on kõigi lävendite puhul ligi 20%, mis võiks kindlasti madalam olla.

Joonisel 28 on kujutatud parima lävendiga järeltest ka graafiliselt. On näha, et simulatsiooni alguses on vähe tehinguid ning see võib olla märk sellest, et mudel teatud hindade juures väga hästi ei tööta. Kuid seejärel on tehingute arv ühtlustunud. Suurim kaotus esineb 2020. aasta alguses, mis on tingitud COVID-19 kriisist ning ka joonisel on samas perioodis näha aktsiahinna suurt langust. Näha on ka seda, et peale kriisi algust oli turg palju volatiilsem ning suuremad kasumid tehingute pealt tulid just volatiilsemal perioodil.



Joonis 28. Järeltesti tulemused testandmestiku (Microsoft) peal (lävend = 0,75%).

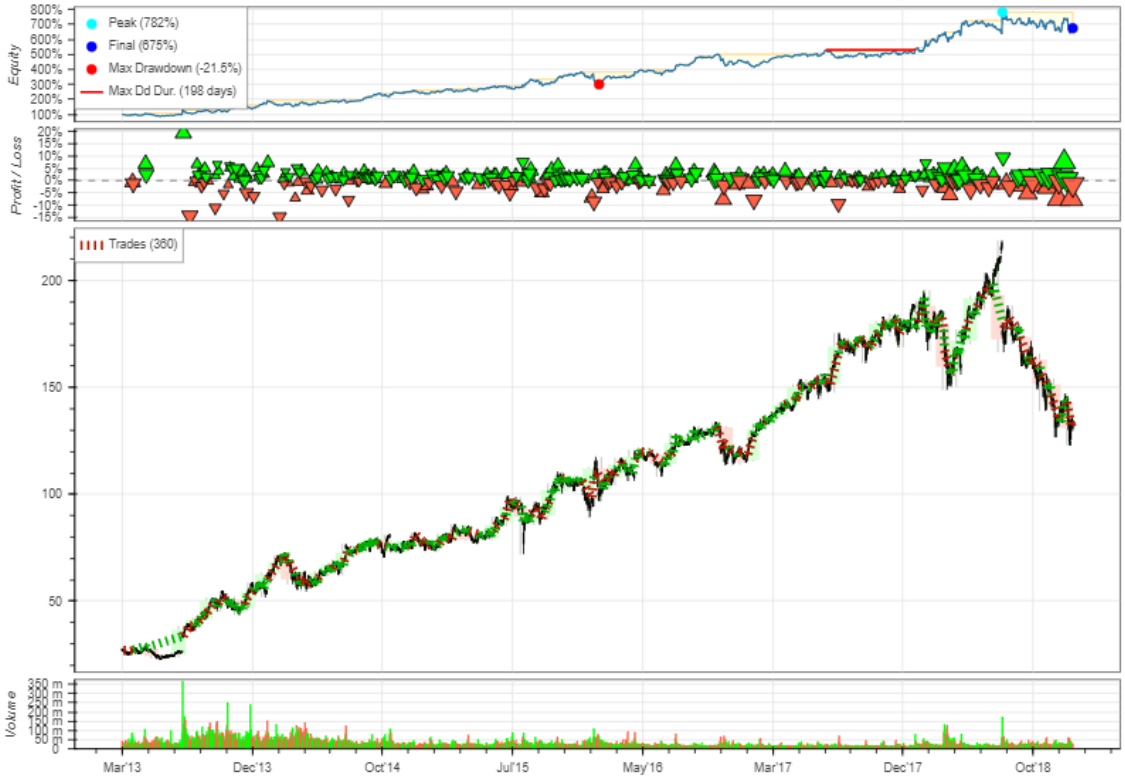
Et vaadelda, kuidas käitub mudel teiste aktsiate puhul ning enne COVID19 kriisi, siis on tehtud järeltestid ka Facebook'i ja Apple aktsiate peal. Facebooki aktsia puhul on kasutatud andmeid perioodist 01.01.2013 – 01.01.2019, kokku 6 aastat. Tulemused on kujutatud Tabelis 13.

Tabel 13. Järeltestide tulemused Facebooki aktsiate peal.

Mõõdik	**Lävend 0%**	**Lävend 0,25%**	**Lävend 0,5%**	**Lävend 0,75%**	**Lävend 1%**
Equity Final [$]	**67525**	59981	58212	50625	54654
Equity Peak [$]	**78209**	71222	67002	55758	60197
Return [%]	**575.25**	499.8	482.1	406.25	446.54
Buy & Hold Return [%]	**384.08**	384.08	384.08	384.08	384.08
Sharpe Ratio	**0.805**	0.748	0.741	0.692	0.723
Max. Drawdown [%]	**-21.52**	-23.74	-24.47	-22.57	-25.68
Number of trades	**360**	294	242	190	156
Win Rate [%]	**67.22**	71.43	73.55	70.00	68.59
Best Trade [%]	**19.20**	19.20	19.20	19.20	19.20
Worst Trade [%]	**-14.67**	-14.67	-14.89	-14.89	-14.08

Mõõdik	**Lävend 0%**	**Lävend 0,25%**	**Lävend 0,5%**	**Lävend 0,75%**	**Lävend 1%**
Avg. Trade [%]	**0.533**	0.612	0.732	0.860	1.096
Avg. Trade Duration	**6 days**	8 days	9 days	12 days	14 days

Tulemustest on märgata, et kõik lävendid on osta ja hoia strateegiat ületanud, küll aga on Sharpe suhtarv alla ühe ning see tähendab, et riskitase on kõrge. Ka suurim kaotus (Max Drawdown) on kõigi testide puhul üle 20%, mis on märk sellest, et Facebooki aktsia hindade peal mudel nii hästi ei tööta. Näha on ka, et järeltesti vältel oli kõrgeim kapital, mis saavutati 78209 USD, kuid lõppsumma oli ligi 11000 USD madalam. Joonis 29 pealt on näha lävendiga 0% järeltesti tulemusi graafiliselt. On märgata, et testi lõpuosas on kapital oluliselt langenud. See võib olla tingitud sellest, et aktsiahind tegi päevaga suure kukkumise (ligi 20%) ning jätkas langust ka järgneval perioodil.



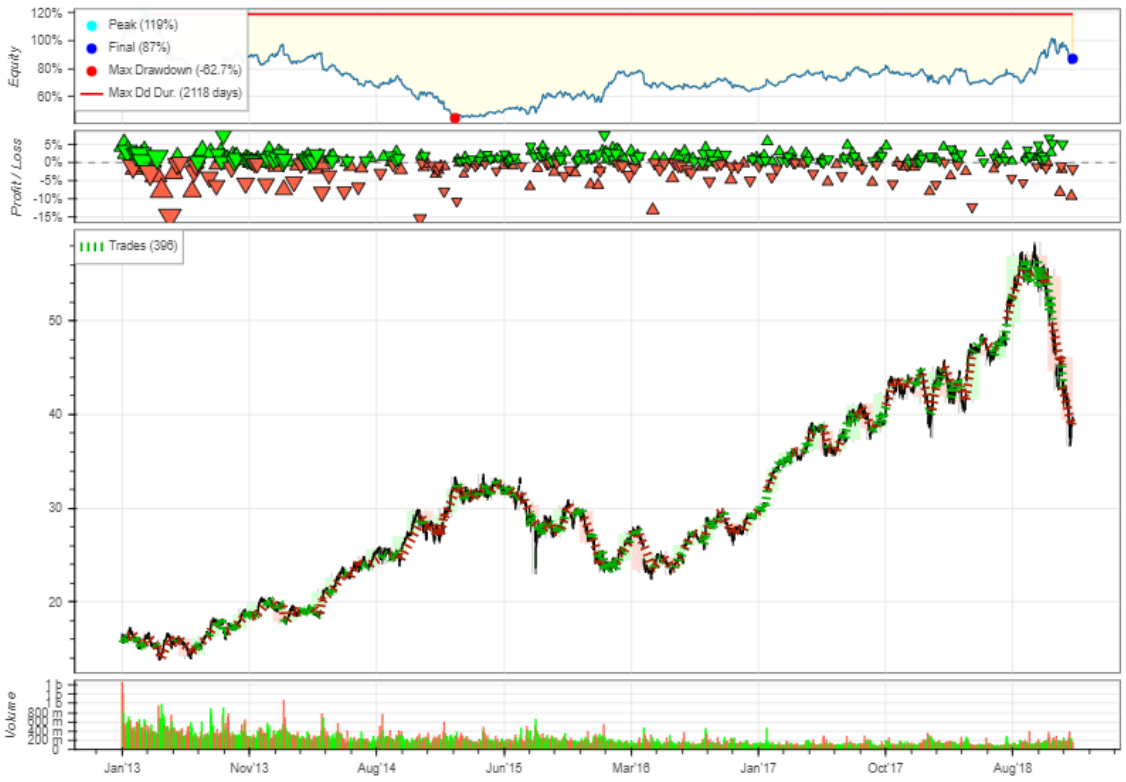
Joonis 29. Järeltesti tulemused Facebooki aktsiate peal (lävend = 0%).

Viimane järeltest on läbiviidud Apple aktsiate peal samas vahemikus nagu eelmine (01.01.2013 – 01.01.2019). Tulemused on kujutatud Tabelis 14.

Tabel 14. Järeltestide tulemused Apple aktsiate peal.

Mõõdik	**Lävend 0%**	**Lävend 0,25%**	**Lävend 0,5%**	**Lävend 0,75%**	**Lävend 1%**
Equity Final [$]	8692	9174	9266	7296	8531
Equity Peak [$]	11894	11276	11066	11407	10803
Return [%]	-13.07	-8.26	-7.33	-27.03	-14.68
Buy & Hold Return [%]	145.10	145.10	145.10	145.10	145.10
Sharpe Ratio	0	0	0	0	0
Max. Drawdown [%]	-62.71	-60.39	-51.76	-51.40	-41.63
Number of trades	396	294	234	180	150
Win Rate [%]	62.37	62.59	64.53	63.33	60.67
Best Trade [%]	7.61	8.52	8.52	8.52	10.51
Worst Trade [%]	-15.44	-15.44	-22.85	-21.25	-21.25
Avg. Trade [%]	-0.035	-0.029	-0.032	-0.174	-0.105
Avg. Trade Duration	6 days	8 days	10 days	13 days	15 days

On näha, et mudel ei saa Apple aktsiate puhul ennustamisega väga hästi hakkama ning jääb kõigi lävendite puhul kahjumisse. Joonisel 30 on 0% lävendiga järeltesti tulemus kujutatud graafiliselt, mis näitab samuti, et mudel jääb kohe alguses kahjumisse. Suure tõenäosusega on kehva tulemuse põhjuks Apple aktsia madalad hinnad, millega mudel nii hästi hakkama ei saa. Et leida universaalsem mudel, mis suudaks igas vahemikus aktsiahindu ennustada, tuleks mudelit treenida erinevate ettevõtete aktsiahindade peal.



Joonis 30. Järeltesti tulemused Apple aktsiate peal (lävend = 0%).

4 Analüüs ja järeldused

4.1 Sentimentaalanalüüs

Eksperimentide käigus saavutatud tulemustest järeldub, et kasutatud kujul sentimendianalüüs ennustusjõudlust ei paranda. Saavutatud tulemus ei erinenud oluliselt tulemusest, kus kasutati vaid aktsiahinna andmeid. Suure tõenäosusega on probleem kasutatud API madalas kvaliteedis. API valikul osutus määravaks asjaolu, et finantsuudiste andmestike ja API-sid, mis on seotud mingi konkreetse ettevõttega, on saadaval väga piiratud koguses. Enamik saadaolevaid API-sid pakuvad maksimaalselt vaid aasta jagu finantsuudiseid. Kuna antud töös kasutatud API oli ainuke, mis pakkus enam kui aasta jagu uudiseid, siis otsustas autor selle kasuks. Puuduseks oli ka see, et pärida oli võimalik vaid uudiste pealkirju, nende sisu ega lühikokkuvõtet ei pakutud. Autor usub, et kvaliteetsete finantsuudiste põhjal tehtud sentimentaalanalüüsiga oleks võimalik mudeli ennustusjõudlust parandada.

2019. aastal ilmunud töös [36] on BERT mudelit kasutatud *Dow Jones Industrial* (DJI) börsiindeksi tõusude ja languste ennustamiseks. Andmeteks on kasutatud finantsuudiseid ning andmestik koosneb kokku 582-st uudisest. Olulise tulemusena on välja toodud, et sentimendi klassifitseerimises on BERT märkimisväärselt parem kui teised levinud teksti klassifitseerimise meetodid nagu näiteks SVM, TextCNN ja Naive Bayes. Samuti saavutati 69% täpsus DJI hinna suundade ennustamisel kasutades vaid BERT-i poolt klassifitseeritud sentimente.

Lisaks finantsuudistele võib kaaluda ka sotsiaalmeedia andmete kasutamist sentimendi määramiseks. Üks levinum viis selleks on Twitteri säutsude kasutamine. 2016. aastal ilmunud uurimistöös [37] on uuritud säutsude sentimentide põhjal aktsiaturu liikumissuuna ennustamist. Autorid olid valinud aasta jagu säutse, mis koosnesid erinevatest arvamustest, mis puudutasid Microsofti aktsiat, tooteid ja teenuseid. Kuigi antud töös pole kasutatud BERT-i, siis on välja toodud, et säutsude sentimendi ja aktsia hinna suuna vahel esineb tugev korrelatsioon. Loodud klassifikaatormudeliga, mis kasutas aktsiahindu ja sentimenti, saavutati ligi 72% täpsus.

Eelnevalt kirjeldatud tööde puuduseks saab pidada andmete vähesust. Töödes kasutatud andmete ajalised perioodid on üsna lühikesed, mille tõttu võib oletada, et aktsiate puhul on mahuka ja kvaliteetse sentimentaalanalüüsi läbiviimiseks sobivatest andmetest puudus. Kuigi saadaval on mitmeid API-sid, mille kaudu on võimalik reaalajas finantsuudiseid pärida, siis probleem seisneb ajaloolistes andmetes, mida on vaja mudeli treenimiseks. Kui leida lahendus ajalooliste andmete puudusele, siis võib hästi läbiviidud sentimentaalanalüüs investorile osutuda oluliseks tööriistaks ja eeliseks teiste investorite ees.

4.2 GAN teistes teadustöödes

Kuna käesoleva töö raames loodud mudelit teiste mudelitega ei võrreldud, siis antud küsimusele on vastust otsitud kirjandusest. 2019. aastal ilmunud töös [38] on kasutatud GAN mudelit, mis erineb traditsioonilisest GAN mudelist selle poolest, et autorid on kombineerinud traditsioonilise GAN mudeli generaatori kaofunktsiooni enda loodud kaofunktsiooniga. Sarnaselt käesoleva tööga, kasutavad autorid generaatori sisendina juhusliku müra asemel päris andmeid ning generaator väljastab järgmise päeva sulgemishinna. Sulgemishind liidetakse eelmisele 5-le päevale ning seda kasutatakse päris andmete kõrval diskriminaatori sisendina. Mudeli treenimisel kasutati 20 aasta jagu 5 erineva aktsia ja börsiindeksi andmeid USA ja Hiina turgudelt. Loodud GAN mudeli tulemusi võrreldi teiste klassikaliste mudelitega, milleks olid LSTM, ANN ja SVR. Kõigi 5 andmestike keskmiste tulemuste põhjal oli GAN selgelt teistest parem, paremuselt teine mudel oli LSTM. Olgugi, et autorid kasutasid loodud GAN mudeli generaatorina LSTM mudelit, siis saavutas see ikkagi märkimisväärselt paremaid tulemusi kui tavaline LSTM mudel.

2018. aastal ilmunud töös [39] kasutati GAN mudelit kõrgsageduslike börsiandmete peal. Andmeteks kasutati 42 erineva Hiina turu aktsiahindasid ühe aastase perioodiga. Andmed olid üheminutiliste intervallidena ning mudeli ülesanne oli ennustada väärtpaberi järgmise minuti hinda. Generaatorina kasutati LSTM võrku ning autorid kasutasid spetsiaalset kaofunktsiooni, mis arvestas päris hinna ja genereeritud hinna vea suurusega ning ka mõlema hinna liikumissuunaga. Mudeli hindamiseks kasutati kahte mõõdikut. Esimeseks mõõdikus oli RMSRE, mis mõõdab ruutkeskmist suhtelist viga ning on üsna sarnane ka käesolevas töös kasutatud RMSE mõõdikule. Teine mõõdik oli spetsiaalselt

loodud hindamaks päris ja ennustatud hinna liikumise suundasid. Autorid viisid läbi ulatuslikud testid ja võrdlesid loodud mudelit teiste levinud mudelitega, milleks olid ARIMA-GARCH, ANN, LSTM ja SVM. Kusjuures kasutas LSTM mudel sama kaofunktsiooni nagu autorite loodud GAN mudeli generaator. Autorite poolt loodud GAN-FD mudel saavutas mõlema mõõdiku puhul suurema osa ajast parimad tulemused. Paremuselt teine mudel oli LSTM mudel, mis suutis vaid üksikutel kordadel GAN-FD mudelit lüüa.

4.3 Loodud mudel ja järeltestimine

GAN mudelid on tuntud oma ebastabiilsuse poolest ning eksperimentide käigus oli seda ka märgata. Eksperimendid näitasid, kui oluline on õigete parameetrite valik GAN mudeli treenimisel. Asjaolu ei lihtsusta ka see, et GAN-il on oluliselt rohkem parameetreid, mida muuta kui teistel traditsioonilistematel mudelitel, nagu näiteks tavalisel LSTM mudelil. Mudeli loomise käigus pani autor tähele, et lisaks hüperparameetritele oli mudeli stabiilsust teatud määral võimalik kontrollida ka generaatori ja diskriminaatori võrgukihtide parameetreid muutes. Diskriminaatori puhul parameetreid niivõrd muutma ei pidanud, kuna diskriminaatori sisendiks olid oluliselt väiksemate dimensioonidega andmed. Et saavutata treenimisel veelgi parem stabiilsus ning seeläbi ka paremad tulemused tuleks põhjalikult hüperparameetreid tuunida ning optimeerida. Oluline oleks ka katsetada erinevaid võrkude arhitektuure nii generaatori kui ka diskriminaatori puhul. Samuti oli märgata, et treenimisel koondub mudel kiirelt punkti, kus diskriminaator ei suuda vahet teha päris ja genereeritud andmetel ehk saavutab optimumi. Kuid optimumi saavutades oli mudel liialt ülesobitunud ning ei omanud piisavat üldistusvõimet, et valideerimisandmete peal häid tulemusi saavutada.

Töö käigus jõuti ka järeldusele, et valideerimise käigus kasutatud RMSE väärtus ei tähenda järeltestimisel ilmtingimata parimat tulemust. Mitmel korral esines olukordi, kus mudel A oli mudel B-st RMSE poolest vähemalt 0,5 punkti võrra parem, kuid järeltestimisel andis mudel B sama strateegiat kasutades kordades suurema kasumi. RMSE ja sarnaste mõõdikute kasutamine on hea viis võrdlemaks erinevate mudelite täpsust arvulise väärtuse ennustamisel, kuid investeerimisstrateegia loomise korral tuleks kaaluda mudelis kasutatava Y-väärtuse konstrueerimist kui hinna tõusu/languse klassifikatsiooni. Taolise Y-väärtusega oleks lihtsam treenimise käigus välja valida parim

mudel, kuid see kitsendab võimalusi investeerimisstrateegia loomisel hinna muutumise suurusjärkudega arvestamiseks. Keerukam variant oleks konstrueerida Y-väärtused osta/müü/hoia signaalidena, mis lihtsustaks hilisemat investeerimisstrateegia loomist ja rakendamist, kuid see-eest tuleb oluliselt rohkem vaeva näha Y-väärtuse konstrueerimisel.

Käesoleva töö nõrkuseks saaks pidada generaatori kaofunktsiooni. Kaaluda tuleks eelnevalt kirjeldatud teadustööde eeskujul generaatori jaoks spetsiaalse kaofunktsiooni loomist, mis vastab paremini lahendatava probleemi vajadustele ning võtab arvesse ka hinna liikumissuunda. Kaofunktsioon võimaldaks efektiivsemalt mudelit treenida ning lihtsustaks parima mudeli leidmise protsessi. Eelduseks on see, et ka valideerimiseks kasutatakse samuti spetsiaalseid mõõdikuid, mis tuginevad kasutatavale kaofunktsioonile. Kuid julgustavaks asjaoluks võib pidada, et ka mitte-optimaalse kaofunktsiooniga suudeti välja valida mudel, mis andis järeltestimisel häid tulemusi. Sellest tulenevalt võib eeldada, et mudeli ennustusjõudlust on võimalik veelgi parandada.

Kuigi eksperimentide käigus saavutati RMSE poolest parimad tulemused tehnilisi indikaatoreid mitte kasutades, siis tulemuste vahe oli üsna väike ning ei saa kindlalt järeldada, et tegu ei olnud lihtsalt juhusega. Küll aga oli märgata, et tehniliste indikaatorite kasutamine kiirendab mudeli optimumi jõudmist ehk mudel leiab treenimisel kiiremini sobivad kaalud ja treeningprotsess on kiirem. Kuid selle tulemusel hakkab mudel ka kiiremini ülesobitama. Kuna tehnilised indikaatorid on tuletatud aktsiahinna andmete pealt, siis võiks eeldada, et tehniliste indikaatorite kasutamine ennustusjõudlusele juurde ei anna. Antud töös on kasutatud vaid tuntumaid tehnilisi indikaatoreid, kuid on ka arvukalt muid ja vähemtuntumaid, mida saaks kasutada. Et täpsemalt välja selgitada, kas ja millist mõju indikaatorite kasutamine mudeli ennustusjõudlusele avaldab, siis tuleks läbi viia põhjalik ja mahukas uurimistöö. Kuid investori vaatenurgast võib see osutada ebaoluliseks, sest nende kasutamine ei pruugi olulisel määral ennustusjõudlusele kaasa aidata ning selliste uurimistööde läbiviimine võib olla ajaliselt kulukas.

Järeltestimise käigus saadud tulemused olid autori jaoks oodatust oluliselt paremad. Kuid oli selgelt näha, et mudelil esineb teatud puuduseid. Kui Microsofti ja Facebooki aktsiate puhul saavutati osta ja hoia strateegiast oluliselt paremad tulemused, siis Apple aktsiahindade peal tehtud järeltestid olid kõik kahjumis. Sellest ilmnes, et mudel suudab

sulgemishindu vaid teatud vahemikes edukalt ennustada. Olgugi, et andmed on standardiseeritud, siis tõenäoliselt jäävad erinevate aktsiate puhul siiski mõningad eripärad ning mudel nendega hakkama ei saa. Et antud probleemi elimineerida tuleks mudelit treenida mitte ainult ühe kindla ettevõtte aktsiahindadega, vaid oluliselt rohkemate. See peaks tagama mudelile nii-öelda universaalsuse, kus ta saab edukalt hakkama ka madalamate või kõrgemate hindade ennustamisega. Samuti tuleks täpsemalt hinnata mudeli kasutamisega seotud riske, et ei tekiks olukorda, kus mudel suure kahjumi toob.

Loodud investeerimisstrateegia puhul oli märgata, et lävendi muutmisel hakkavad tulemused varieeruma ning on raske öelda, milline lävend reaalses olukorras parima tulemuse annaks. Tegu on olemuselt üsna lihtsa strateegiaga ning kogenumad investeerijad oskaks kindlasti ka teistsuguseid ja optimaalsemaid strateegiad antud mudeli põhjal konstrueerida.

4.4 Generatiivne võimekus ja sünteetilised andmed

Kuigi loodud mudeli eesmärk on aegridade, täpsemalt aktsia sulgemishinna ennustamine, siis mudelit on võimalik modifitseerida ka näiteks realistlike sünteetiliste aegridade genereerimiseks. Selle jaoks oleks vajalik treenimisel generaatori sisendiks määrata päris andmete asemel juhuslike väärtustega vektor, mis ei tohiks olla konstantne, vastasel juhul genereeritaks vaid ühte kindlalt eksemplari. Samas, kui on soov näha, kuidas ühe ja sama eksemplari genereerimine epohhide jooksul muutub, siis on treenimisel seda võimalik vaadelda kasutades teatud ajahetkedel konstantset sisendvektorit. Samuti tuleks meeles pidada, et kuna sünteetiliste aegridade genereerimine on olemuselt teistsugune probleem kui pelgalt ühe väljundväärtuse ennustamine, siis tuleks muudatusi teha ka mudeli parameetrites, võrkude arhitektuurides ning tuleks muuta ka kasutatavate andmete kuju.

Sünteetilistel andmetel on mitmeid potentsiaalseid eeliseid päris andmete ees. Päris andmeid on tihtipeale saadaval vaid piiratud koguses ning nende kogumine on nii majanduslikult kui ka ajaliselt kulukas, samuti võivad nende kasutamisel olla takistuseks andmekaitsepiirangud. GAN mudelitega on sünteetilisi andmeid genereeritud ning kasutatud mitmetes valdkondades. Näiteks on katsetatud sünteetiliste peaaju Magnetresonantstomograafia (MRT) piltide genereerimiseks [40], et läbi andmehulga suurendamise saavutada paremad tulemusi ajukasvajate diagnoosimisel või koolitada

arste erinevaid haigusi paremini mõistma ning seeläbi ennetada valede diagnooside määramist. Visuaalse Turingi testi käigus oli isegi kogenud arstidel keeruline vahet teha päris ning sünteetilistel MRT piltidel.

GAN arhitektuuril põhinevate mudelite võimekust näitab ka töö [41], kus genereeriti inimese kõne abil pilt tema näost. Töö erilisus seisneb selles, et pilt isiku näost genereeritakse pikslite tasemel vaid selle sama isiku kõne helisignaalist. Autorid lõid YouTube'i platvormi tuntumate kasutajate videote põhjal andmestiku ning kasutasid seda mudeli treenimisel. Valideerimiseks kasutati teisi eeltreenitud mudeleid. Genereeritud näopiltide täpsus oli 90,25%, mis tähendab, et mudel suutis enamikel kordadest jäädvustada nägude omadused. Kõneleja identiteeti suudeti genereeritud piltide põhjal tuvastada 50% täpsusega.

Generatiivsed võistlusvõrgud on võimsad ning mitmekülgsed tööriistad sünteetiliste andmete genereerimisel. Niisamuti võiks GAN mudeli kasutamine olla investori jaoks kasulik, et sünteetilisi aegridu genereerida. Selle tulemusel oleks võib-olla võimalik elimineerida probleemi, kus sentimentaalanalüüsiks sobivate andmete vähesus takistab treenimisel suurema perioodiga andmete kasutamise. Lisaks sellele on ka muid väga erinevaid domeene, kus GAN mudelite generatiivset võimekust on võimalik rakendada ning nende abil on ka väga keerulisi probleeme võimalik lahendada. Eeliseks on ka see, et GAN mudeleid on lihtne kohandada vastavalt lahendatavast probleemist tulevatele vajadustele.

4.5 Edasine töö

Et mudelit reaalselt investeerimiseks rakendada tuleks tehtud vigadest õppida. Peamiseks puuduseks loodud mudeli puhul on kaofunktsioon ning see puudus tuleks kindlasti likvideerida. Samuti tuleks mudelit treenida oluliselt rohkemate ja erinevate ettevõtete aktsiahindade peal, mis eeldatavasti tagaks selle, et mudelil oleks igas hinnavahemikus aktsepteeritav sooritusvõime. Tuleks uurida, kuidas olemasolevat investeerimisstrateegiat optimeerida või milliseid teisi strateegiaid on võimalik luua ja mis tulemusi nendega on võimalik saavutada. Enne investeerimist tuleks kindlasti ka põhjalikult mudelist ja strateegiast tulenevaid riske hinnata, et investeerimisel ei tekiks ebasoodsaid olukordi.

5 Kokkuvõte

Aktsiahindasid on erinevate masinõppe meetoditega ennustatud juba aastakümneid, kuid tänu tehnoloogia arengule ja arvutusjõudluse kasvule on üha populaarsemaks muutunud sügavõppel põhinevad meetodid. Aktsiahindade ennustamise probleem seisneb nende stohhastilises loomuses, mis muudab nende ennustamise keeruliseks. Käesoleva töö eesmärk oli kasutada aktsiahindade ennustamiseks uudset sügavõppel põhinevat mudelit – generatiivset võistlusvõrku (GAN). Uuriti, millised puudused GAN mudeli puhul esinevad ning milliseid edasiarendusi kasutades on neist võimalik hoiduda. Sentimentaalanalüüsi jaoks kasutati eeltreenitud ja finantsandmetel peenhäälestatud keelemudelit. Töö tulemustena kirjeldati andmete töötlemise protsessi, loodud mudeli arhitektuuri ning viidi läbi ka eksperimendid selgitamaks välja sentimentaalanalüüsi ja tehniliste indikaatorite kasutamise mõju ning vaadeldi ka mudeli üldist käitumist treeningul. Treenimise käigus valiti välja sobivaim mudel ning järeltestimisel suudeti lihtsa investeerimisstrateegiaga osta ja hoia strateegiat edukalt lüüa. Küll aga ilmnesid mudeli juures mõningad probleemid, mille põhjal tehti järeldused ning kirjeldati, mida tuleks teha, et edasises töös nendest hoiduda. Teisi teadustöid analüüsides jõuti järeldusele, et GAN arhitektuuril põhinevad mudelid edestavad aktsiahindade ennustamisel traditsioonilisemaid masinõppe mudeleid.

Kasutatud kirjandus

[1] AS LHV Pank, „Investeerimisõpik,“ [Võrgumaterjal]. Available: https://fp.lhv.ee/academy/investmentguide. [Kasutatud 22.02.2021].

[2] W. contributors, „Efficient-market hypothesis,“ 2021. [Võrgumaterjal]. Available: https://en.wikipedia.org/w/index.php?title=Efficient-market_hypothesis&oldid=1007776989. [Kasutatud 22 Veebruar 2021].

[3] E. F. Fama, „Efficient Capital Markets: A Review of Theory and Empirical Work.,“ *The Journal of Finance,* kd. 25, nr 2, pp. 383-417, 1970.

[4] White, „Economic prediction using neural networks: the case of IBM daily stock returns,“ *IEEE 1988 International Conference on Neural Networks,* kd. II, pp. 451-458, 1988.

[5] T. Kimoto, K. Asakawa, M. Yoda ja M. Takeoka, „Stock market prediction system with modular neural networks,“ *1990 IJCNN International Joint Conference on Neural Networks,* kd. I, pp. 1-6, 1990.

[6] E. W. Saad, D. V. Prokhorov ja D. C. Wunsch, „Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks,“ *IEEE Transactions on Neural Networks,* kd. 9, nr 6, pp. 1456-1470, 1998.

[7] W. Jiang, „Applications of deep learning in stock market prediction: recent progress,“ 2020.

[8] O. Bustos ja A. Pomares-Quimbaya, „Stock market movement forecast: A Systematic review,“ *Expert Systems with Applications,* kd. 156, nr 0957-4174, 2020.

[9] A. M. Ozbayoglu, M. U. Gudelek ja O. B. Seze, „Deep learning for financial applications : A survey,“ *Applied Soft Computing,* kd. 93, nr 1568-4946, 2020.

[10] M. Mohri, A. Rostamizadeh ja A. Talwalkar, %1 *Foundations of Machine Learning, second edition*, MIT Press, 2018, pp. 1-7.

[11] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition, 2019.

[12] R. S. Sutton ja A. G. Barto, %1 *Reinforcement Learning, second edition: An Introduction*, MIT Press, 2018, pp. 1-7.

[13] J. Patterson ja A. Gibson, Deep Learning, O'Reilly Media, Inc., 2017.

[14] F. Chollet, Deep Learning with Python, Manning Publications, 2017.

[15] S. Ravichandiran, „What is deep learning?,“ %1 *Hands-On Deep Learning Algorithms with Python*, Packt Publishing, 2019.

[16] A. Géron, „Recurrent Neural Networks,“ %1 *Neural networks and deep learning*, O'Reilly Media, Inc., 2018.

[17] R. Shanmugamani, A. Fandango ja G. Bonaccorso, „GRU,“ %1 *Python: Advanced Guide to Artificial Intelligence*, Packt Publishing, 2018.

[18] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford ja X. Chen, „Improved Techniques for Training GANs," *Proceedings of the 30th International Conference on Neural Information Processing Systems,* p. 2234–2242, 2016.

[19] P. Isola, J.-Y. Zhu, T. Zhou ja A. A. Efros, „Image-to-Image Translation with Conditional Adversarial Networks," 2018.

[20] A. Kuefler, J. Morton, T. Wheeler ja M. Kochenderfer, „Imitating driver behavior with generative adversarial networks," *2017 IEEE Intelligent Vehicles Symposium (IV),* pp. 204-211}, 2017.

[21] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville ja Y. Bengio, „Generative Adversarial Nets," *Proceedings of the 27th International Conference on Neural Information Processing Systems,* kd. 2, p. 2672–2680, 2014.

[22] „Machine Learning Crash Course," [Võrgumaterjal]. Available: https://developers.google.com/machine-learning/gan/problems.

[23] M. Arjovsky, S. Chintala ja L. Bottou, „Wasserstein GAN," 2017.

[24] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin ja A. Courville, „Improved Training of Wasserstein GANs," %1 *Curran Associates Inc.,* 2017.

[25] J. Devlin, M.-W. Chang, K. Lee ja K. Toutanova, „BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2018.

[26] Y. Yang, M. Christopher Siy UY ja A. Huang, „FinBERT: A Pretrained Language Model for Financial Communications," 2020.

[27] „Yahoo Finance," [Võrgumaterjal]. Available: https://finance.yahoo.com/. [Kasutatud 13.03.2021].

[28] „TickerTick-API," [Võrgumaterjal]. Available: https://github.com/hczhu/TickerTick-API. [Kasutatud 04.04.2021].

[29] „TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems}," [Võrgumaterjal]. Available: https://www.tensorflow.org/. [Kasutatud 13.03.2021].

[30] „Keras," [Võrgumaterjal]. Available: https://keras.io/. [Kasutatud 12.03.2021].

[31] „NumPy," [Võrgumaterjal]. Available: https://numpy.org/doc/stable/contents.html. [Kasutatud 12.03.2021].

[32] „pandas," [Võrgumaterjal]. Available: https://pandas.pydata.org/. [Kasutatud 13.03.2021].

[33] „Pandas TA - A Technical Analysis Library in Python 3," [Võrgumaterjal]. Available: https://github.com/twopirllc/pandas-ta. [Kasutatud 27.03.2021].

[34] „Backtesting.py," [Võrgumaterjal]. Available: https://kernc.github.io/backtesting.py/. [Kasutatud 12.03.2021].

[35] „Google Colab," [Võrgumaterjal]. Available: https://colab.research.google.com/. [Kasutatud 12.03.2021].

[36] M. G. Sousa, K. Sakiyama, L. d. S. Rodrigues, P. H. Moraes, E. R. Fernandes ja E. T. Matsubara, „BERT for Stock Market Sentiment Analysis," %1 *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI),* 2019.

[37] V. S. Pagolu, K. N. Reddy, G. Panda ja B. Majhi, „Sentiment analysis of Twitter data for predicting stock market movements," %1 *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES),* 2016.

[38] K. Zhang, G. Zhong, J. Dong, S. Wang ja Y. Wang, „Stock Market Prediction Based on Generative Adversarial Network,“ *Procedia Computer Science,* kd. 147, pp. 400-406, 2019.

[39] X. Zhou, Z. Pan, G. Hu, S. Tang ja C. Zhao, „Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets,“ 2018.

[40] C. Han, H. Hayashi, L. Rundo, R. Araki, W. Shimoda, S. Muramatsu, Y. Furukawa, G. Mauri ja H. Nakayama, „GAN-based synthetic brain MR image generation,“ *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018),* pp. 734-738, 2018.

[41] A. Duarte, F. Roldan, M. Tubau, J. Escur, S. Pascual, A. Salvador, E. Mohedano, K. McGuinness, J. Torres ja X. Giro-i-Nieto, „Wav2Pix: Speech-conditioned Face Generation using Generative Adversarial Networks,“ 2019.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[1]

Mina, Daniel Aju

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Aktsiahindade ennustamine generatiivse võistlusvõrguga", mille juhendaja on Tõnn Talpsepp
   1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
   1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

09.05.2021

1 Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Generaatori mudel programmikoodina

```
def Generator(input_dim, output_dim, feature_size) -> tf.keras.models.Model:
  model = Sequential()
  model.add(GRU(units=256,
                return_sequences=True,
                recurrent_dropout=0.1,
                recurrent_regularizer=regularizers.l2(0.001)))
  model.add(GRU(units=128,
                recurrent_dropout=0.1,
                recurrent_regularizer=regularizers.l2(0.001)))
  model.add(Dense(64, kernel_regularizer=regularizers.l2(0.001)))
  model.add(Dense(units=output_dim))
  return model
```

Lisa 3 – Diskriminaatori mudel programmikoodina

```
def Discriminator(input_dim, feature_size) -> tf.keras.models.Model:
  model = tf.keras.Sequential()
  model.add(Conv1D(32, input_shape=(input_dim, feature_size), kernel_size=3,
strides=2, padding="same", activation=LeakyReLU(alpha=0.01)))
  model.add(Conv1D(64, kernel_size=3, strides=2, padding="same",
activation=LeakyReLU(alpha=0.01)))
  model.add(Conv1D(128, kernel_size=3, strides=2, padding="same",
activation=LeakyReLU(alpha=0.01)))
  model.add(Flatten())
  model.add(Dense(256, use_bias=True))
  model.add(LeakyReLU())
  model.add(Dense(256, use_bias=True))
  model.add(ReLU())
  model.add(Dense(1))
  return model
```