Jarre Leskinen

Koneoppiminen rahoitusmarkkinoiden ennustamisessa

Tietotekniikan kandidaatintutkielma

15. toukokuuta 2019

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Jarre Leskinen

Yhteystiedot: jarre.leskinen@gmail.com

Työn nimi: Koneoppiminen rahoitusmarkkinoiden ennustamisessa

Title in English: Machine learning in financial market forecasting

Työ: Kandidaatintutkielma

Sivumäärä: 21+0

Tiivistelmä: Tutkielma käsittelee koneoppimisen soveltuvuutta rahoitusmarkkinoiden ennustamiseen käsitellen erityisesti eri algoritmeja sekä niiden yhdistelmiä ja syötteen optimointia. Tulokset osoittavat, että tehokkaiden markkinoiden hypoteesin heikot ehdot eivät ole aina toteutuneet täydellisesti ja erityisesti tukivektorikone sekä hybriditoteutukset syötteen optimointiin vaikuttavat lupaavilta. Koneoppimista voidaan hyödyntää tähän ongelmaan ja muihin satunnaisuutta sisältäviin ongelmiin. Tutkimuksessa esitetään myös parannusehdotuksia käsitellyille malleille sekä mahdollisia kohteita jatkotutkimukselle.

Avainsanat: Koneoppiminen, rahoitusmarkkinat, rahoitusmarkkinoiden ennustaminen, tekninen analyysi

Abstract: This study researches whether machine learning could be utilized in forecasting the financial markets. Different types of algorithms are researched and different combinations of those including optimizing the input data. The results suggest that the market is not always weak form efficient. Especially support vector machine and hybrid models with input optimizing show promising results. Machine learning can be utilized for this problem and other problems which include randomness by nature. The study also suggests improvements for the studied models and possible areas for further research.

Keywords: Machine learning, financial markets, forecasting financial markets, technical analysis

Kuviot

Sisältö

1 Johdanto

Rahoitusmarkkinoiden tutkiminen ja sen liikkeiden ennustaminen on ollut olemassa yhtä pitkään kuin markkinat itse. Rahoituksen näkökulmasta tällä voidaan pyrkiä tappioiden minimointiin ja markkinatuoton ylittämiseen. Tähän liittyy oleellisesti tehokkaiden markkinoiden hypoteesi (engl. *Efficient-market hypothesis*) Malkiel ja Fama 1970, joka esittää rahoitusmarkkinoiden reagoivan välittömästi uuteen tietoon ja arvopaperien senhetkinen hinta kuvaa siis täydellisesti niiden arvoa saatavilla olevan tiedon perusteella, hinnoitellen niihin liittyvät riskit ja tuotto-odotukset. Tästä seuraa, että markkinoita ei siis ole mahdollista voittaa pitkällä aikavälillä. Tähän liittyen toinen tämän tutkielman kannalta tärkeä hypoteesi on satunnaiskulku-hypoteesi (engl. *Random walk hypothesis*), joka esittää osakkeiden hinnan muutosten olevan kaoottisia satunnaisuuden takia, eikä niitä näin ollen voi ennustaa (Malkiel 2007). Näistä hypoteeseista huolimatta rahoitusmarkkinoiden pyritään aktiivisesti ennustamaan pääsääntöisesti kahdella eri tavalla. Ensimmäinen tapa on fundamenttianalyysi, joka pyrkii määrittelemään osakkeen sisäisen arvon tutkimalla esimerkiksi yrityksen kilpailuympäristöä, kassavirtoja sekä tulevaisuuden näkymiä. Toinen tapa on tekninen analyysi, joka keskittyy tutkimaan aikaisempaa kurssihistoriaa pyrkien löytämään tiettyjä malleja tai muuta käytöstä, jolla voidaan tehdä oletuksia tulevasta suunnasta. Tämä menetelmä pohjautuu oletukseen, että rahoitusmarkkinoiden kurssikäyttäytyminen reflektoi ihmisluontoa ja sen tapaa reagoida asioihin, mikä voidaan olettaa vakioksi. Näin ollen sijoittajat reagoivat politiikkaan, talouteen ja uutisiin psykologian takia tietyllä tavalla muodostaen kurssikäytöstä, jota tekninen analyysi pyrkii tunnistamaan (Pring 2014).

Tässä tutkielmassa selvitetään koneoppiminen soveltuvuutta rahoitusmarkkinoiden ennustamiseen teknisen analyysin keinoin. Kurssihistorian satunnaisuus tekee ongelmasta haastavan ja sen ratkaiseminen tarjoaisi hyötyä myös muiden ongelmien parissa, kun saatavilla oleva tieto sisältää satunnaisuutta. Tutkielman lopussa pohditaan koneoppimisen soveltuvuutta rahoitusmarkkinoiden ennustamiseen sekä markkinoiden tehokkuutta. Lisäksi esitetään mahdollisia tapoja parantaa saatuja tuloksia keskittyen erityisesti syötetyn datan valintaan sekä sen esikäsittelyyn tarjoten pohjaa mahdollisille jatkotutkimuksille.

Luvussa 2 käsitellään rahoitusmarkkinoiden määritelmää sekä erityisesti tämän tutkimuksen

kohdemarkkinoita eli osakemarkkinoita ja niiden toimintaa. Tämän jälkeen luvussa 3 esitellään tutkielman kannalta oleelliset algoritmit yleisellä tasolla ja luvussa 4 tarkastellaan näiden algoritmien ja niiden yhdistelmien suoriutumista. Lopuksi luvussa 5 esitetään tutkimuksen tulokset ja tarve mahdollisille jatkotutkimuksille.

2 Rahoitusmarkkinat

Eräs tapa määritellä rahoitusmarkkinat on kokonaisuus, joka koostuu eri instrumenteista kuten osakkeista, joukkovelkakirjoista sekä muista lainoista, jossa yhdistetään myyjät sekä ostajat (Burton, Nesiba ja Brown 2015). Tässä tutkielmassa keskitytään pelkästään osakemarkkinoihin, sillä suuri osa tutkimuksista aiheeseen liittyen koskee osakemarkkinoita ja näiden paremman vertailtavuuden takia on valittu vain tutkimuksia, jotka ovat keskittyneet samaan markkinatyyppiin. Näiden tulosten analyysi pyritään pitämään yleisellä tasolla keskittymättä erityisesti osakemarkkinoiden ominaispiirteisiin, jolloin tulokset voivat skaalautua paremmin myös muihin markkinoihin.

2.1 Osakemarkkinat

Osakkeet ovat pääomaa, joka määrittelee sen omistajalle kuuluvan osuuden yrityksen tuloista sekä omaisuudesta. Osakkeenomistajat saavat tulonsa yleensä osingoista, joka on yrityksen maksamaa osuutta sen voitoista. Ostamalla yrityksen osaketta ostaja saa oikeuden yrityksen tuleviin rahavirtoihin. Osinkojen lisäksi voi sijoittaja saada tuottoa myös osakkeen arvon nousulla (Burton, Nesiba ja Brown 2015). Tämä tutkimus keskittyy juuri osakkeen arvon muutoksen ennustamiseen, joka on oleellinen osa aktiivikauppaa.

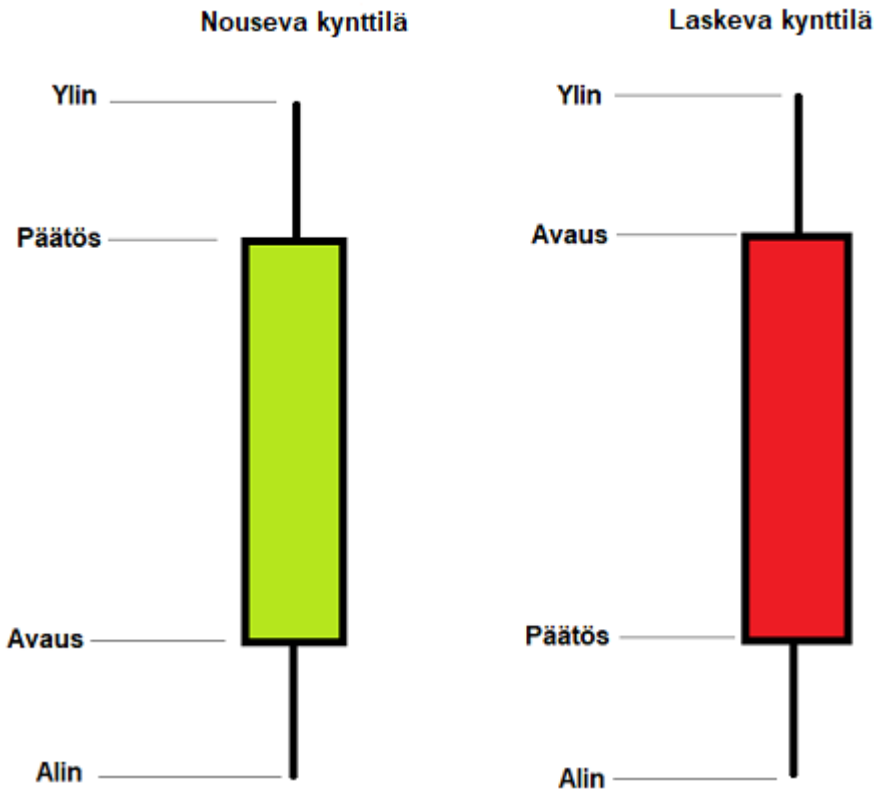
2.2 Aktiivikauppa

Aktiivista kauppaa käyvät tahot pyrkivät ennustamaan osakkeen arvon nousua sekä laskua (Pring 2014). Ideaalisesti osaketta haluttaisiin omistaa, kun sen arvon nousee sekä myydä ennen sen arvon laskua. Joissain tapauksissa kaupankäyjä voi myös myydä osaketta lyhyeksi, jolloin saadaan tuottoa osakkeen arvon laskusta (Malkiel 2007). Tässä tutkielmassa käsitellään tapausta, jossa käytetään koneoppimista ennustamaan kohde-etuuden arvon nousua ja laskua, jolloin voitaisiin myydä ja ostaa tuoton kannalta parhaisiin aikoihin. Toimivan koneoppimismallin ja kaupankäyntistrategian avulla voitaisiin vähentää omaa markkinariskiä sekä ylittää markkinatuotto.

2.3 Kurssihistorian esittäminen

Tässä tutkielmassa historiallisia kurssitietoja kohde-etuuksista esitetään kynttilätikkumallin mukaan, joka on alkuaan Japanista lähtöisin oleva ja 1990-luvulla yleistynyt esitysmenetelmä kurssihistorialle. Tämä esitystapa on erityisesti suosittu teknisessä analyysissä ja sen vuoksi soveltuu hyvin tutkielman aiheeseen. Kynttilöiden käyttäminen mahdollistaa kurssitiedoista teknisten ominaisuuksiaan helpomman tunnistamisen (Pring 2014). Yksi kynttilätikku vastaa aina yhtä aikayksikköä, joka voidaan valita vapaasti. Tämän lisäksi kynttilätikun runko näyttää sen aikana tapahtuneen kurssimuutoksen määrän ja suunnan sekä ohuet viivat ylä- ja alapuolella näyttävät sen aikana tapahtuneet korkeimmat ja matalimmat hinnat. Käytetyn aikavälin valintaan vaikuttaa kuinka lyhyen tai pitkän aikavälin kurssikäyttäytymisestä ollaan kiinnostuneita. Tässä tutkielmassa keskitytään ainoastaan yhden päivän mittaisiin kynttilöihin, jotka kuvastavat paremmin pidemmän aikavälin käytöstä. Kuviossa 1 esitetään nouseva ja laskeva kynttilä.

Kuvio 1. Nouseva ja laskeva kynttilä



2.4 Tekniset indikaattorit

Kurssikäyttäytyminen ei ole satunnaiskävelyyn (engl. *random walk*) perustuvana täysin rationaalista ja näin ollen kurssihistorian käyttäminen koneoppimisen syötteenä aiheuttaa haasteita, sillä itse syötteen laatua voidaan pitää heikkona. Tällöin pelkkien kynttilätikkujen avulla alla olevan kurssikäytöksen analysointi on haastavaa ja siksi teknisessä analyysissa suositaan paljon erilaisia indikaattoreita. Nämä indikaattorit helpottavat kurssikäytöksen tiettyjen ominaisuuksien tunnistamista, kuten sen hetkisen kurssin suuntaa ja sen vahvuutta, volyymin muutosta sekä tuki- ja vastustasojen tunnistamista (Pring 2014). Käsitellyissä tutkimuksissa käytettiin kynttilätikkujen sulkemishintaa sekä sen aikaista kaupankäynnin määrää eli volyymia. Myös kynttilän aikana olleen suurimman ja pienimmän arvon käyttäminen voisi tuoda lisähyötyä, sillä teknisessä analyysissa hyödynnetään myös näihin tietoihin liittyvää yksittäisten tai useampien kynttilöiden muodostamien kynttiläkuvioiden (engl. *Candlestick pattern)* analysointia.

Kurssihistorian satunnaisuuden takia on hyödyllistä ottaa indikaattorit avuksi koneoppimisen kanssa, sillä niillä käytettyä syötettä pystytään yksinkertaistamaan ja normalisoimaan, vähentäen taustakohinan määrää yksittäisistä kynttilöistä. Tällä tavalla pystytään oikein valituilla indikaattoreilla esikäsittelemään käytetty syöte ja parantamaan sen laatua. Tekniset indikaattorit itsessään ovat matemaattisia kaavoja, joiden tarkkaa toteutusta ei tässä tutkielmassa käsitellä.

Tarkastelluissa tutkimuksissa teknisten indikaattorien käyttäminen kurssihistorian käsittelyyn oli hyvin yleistä, mutta käytettyjen indikaattoreiden esivalintaa ei tutkimuksissa oleellisesti perusteltu. Tämä voisi olla siis mahdollinen kohde jatkotutkimukselle, jos käytettyjä indikaattoreita vaihdetaan tai luvun 2.3 mukaisia kynttiläkuvioita hyödynnettäisiin syötteessä. Tutkimukset keskittyivät käyttämään vain kynttilöiden sulkemishintaa ja joissain tapauksissa kaupankäynnin volyymia. Näillä tiedoilla ei voida vielä hyödyntää kaikkia teknisen analyysin välineitä.

3 Koneoppimiseen liittyvät algoritmit

Koneoppiminen voidaan tämän tutkielman näkökulmasta määritellä algoritmeina, joilla pyritään muodostamaan funktio, joka voidaan määritellä seuraavasti: $y = f(\vec{x})$, missä osakemarkkinoiden tapauksessa vektori $\vec{x}$ sisältää tietoa valitusta osakkeesta, kuten aikaisempaa kurssitietoa ja y kuvaa tiettyä osakkeen ominaisuutta tulevaisuudessa. Tämä voidaan valita esimerkiksi osakkeen hinnaksi tai yksinkertaistaa binääriseksi ominaisuudeksi, joka ennustaa nouseeko vai laskeeko osake tietyllä aikavälillä.

Tähän lukuun valitut algoritmit ovat niitä, jotka käsiteltyjen tutkimusten perusteella on valittu oleellisimmiksi. Luvut 3.1 ja 3.2 esittelevät kaksi tutkimuksissa käytettyä koneoppimisen algoritmia, joita käytettiin valittujen osakkeiden kurssikäytöksen ennustamiseen, sekä luvuissa 3.3 ja 3.4 esiteltävät algoritmit liittyvät tutkimusten osalta valitun syötteen optimointiin.

3.1 Tukivektorikone (SVM, engl. *Support Vector Machine*)

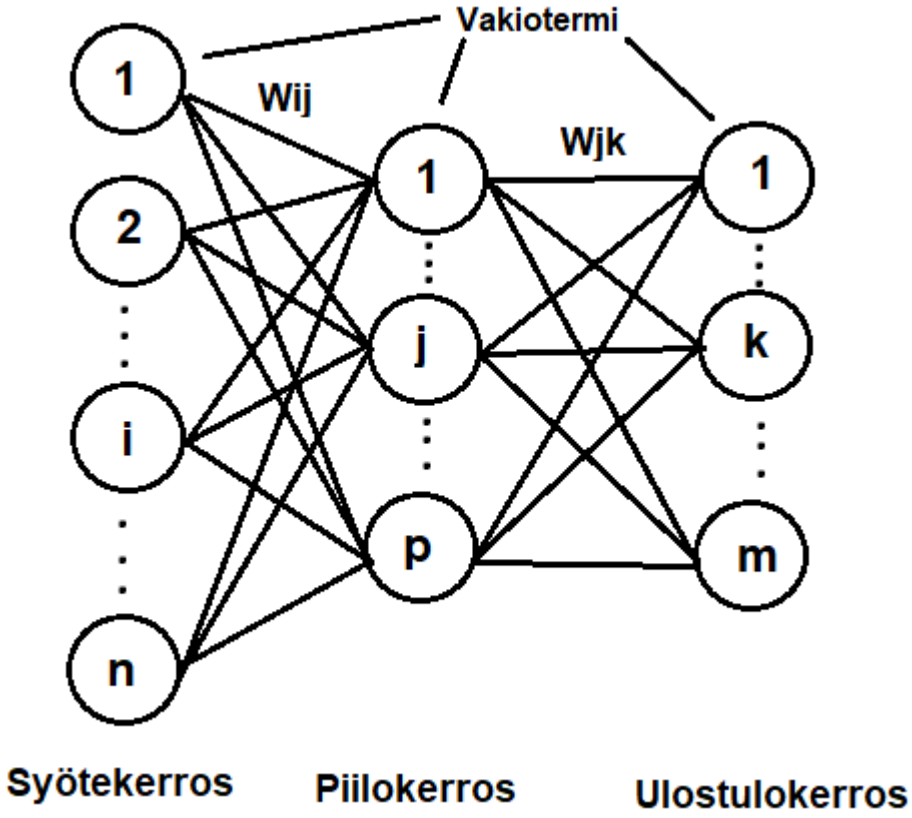
Vapnik 1999 kuvasi artikkelissaan tukivektorikoneen, joka on koneoppimisessa käytettävä algoritmi, joka soveltuu hyvin mallien tunnistamiseen (engl. *pattern recognition*) ja näin ollen intuitiivisesti voidaan sen olettaa sopivan hyvin kurssihistorian analysointiin, jos opittavia malleja on olemassa. Menetelmän ajatuksena on luoda useita funktioita, joista pyritään oppimisen kautta valitsemaan sellainen funktio, joka vastaa odotettua tulosta parhaiten. Tähän käytetään virhefunktiota, jota pyritään minimoimaan ja sen valinta riippuu opeteltavasta datasta. Vapnik esittelee artikkelissaan yksityiskohtaisen toteutuksen ja erilaisia virhefunktioita tukivektorikoneelle.

3.2 Neuroverkko (ANN, engl. *Artificial Neural Networks)*

Neuroverkko on eräs aivojen rakenteen innoittama koneoppimisen menetelmä. Ajatuksena on mallintaa neuroneita yksittäisinä prosessointiyksikköinä, jotka ovat yhteydessä toisiinsa ja näillä yhteyksillä on tietynlaiset painot. Eräs paljon käytetty malli neuroverkolle on eteen-

päin kytketty monikerrosverkko. Tässä mallissa verkko koostuu yhdestä syöte- ja ulostulokerroksesta sekä yhdestä tai useammasta piilokerroksesta. Syöte- ja piilokerroksen neuronit ovat kytkettyinä kaikkiin seuraavan kerroksen neuroneihin ja näihin kytkentöihin liittyy tietty paino, jolla neuronin lähettämä signaali kerrotaan (Moghaddam, Moghaddam ja Esfandyari 2016). Tarpeeksi isolla neuroneiden ja piilokerrosten määrällä voidaan oppia monimutkaisia malleja, jos opetusmateriaalina käytettävää dataa on riittävästi saatavilla. Kuviossa 2 esitetään neuroverkon kerrokset.

Kuvio 2. Eteenpäin kyketty monikerroksinen neuroverkko



3.3 Geneettiset algoritmit (GA, engl. *Genetic Algorithms*)

Geneettiset algoritmit ovat biologian ja erityisesti Darwinin evoluutioteorian innoittama tapa lähestyä ongelman ratkaisua. Ajatuksena on, että elinympäristö pystyy kannattamaan vain rajatun kokoista populaatiota. Näin ollen yksilöiden välillä tapahtuu luonnollista valintaa ja ne, jotka sopeutuvat parhaiten ympäristöönsä lisääntyvät. Tästä seuraten populaatio, joka selviää, sopeutuu elinympäristöönsä ja sen muutoksiin paremmin aina jokaisen sukupol-

ven myötä. Tämä logiikka pystytään siirtämään algoritmiksi koneoppimisen pariin, jossa elinympäristö määritellään ratkaistavaksi ongelmaksi ja populaation jäsenet eri ratkaisuiksi ratkaistavaan ongelmaan. Ajaen evoluutiota muistuttavaa ohjelmaa, jossa parhaiten ongelmaan sopivat ratkaisut risteytetään, saadaan optimoitua ratkaisua paremmaksi (Eiben, Smith ym. 2003). Geneettisiä algoritmeja käytetään monesti erityisesti funktioiden optimointiin ja niitä on käytetty esimerkiksi luvussa 3.2 esiteltyjen neuroverkkojen kanssa, jolloin puhutaan neuroevoluutiosta (engl. *Neuroevolution*). Tässä mallissa hyödynnetään geneettisiä algoritmeja optimoimaan neuroverkon rakennetta, kuten neuroneiden määrää eri kerroksilla, kerrosten määrää sekä käytettyä aktivointifunktiota. Tämän lisäksi tavalliseen tapaan neuroverkkoon liittyvillä algoritmeilla optimoidaan neuroneiden painoja. Etuna tässä ratkaisussa on, että voidaan välttää valitsemasta liian vähän neuroneita tai kerroksia, jolloin neuroverkko ei kykene oppimaan haluttua mallia riittävän tarkasti, tai liikaa neuroneita, jolloin neuroverkon koulutus kestää liian kauan (Hausknecht ym. 2014). Vaihtoehtoisesti geneettisiä algoritmeja voidaan hyödyntää myös neuroverkon saaman syötteen esivalintaan.

3.4 Itsenäinen komponenttianalyysi (ICA, engl. *Independent Component Analysis*)

Itsenäinen komponenttianalyysi on menetelmä, jolla pyritään esittämään käsiteltävä syöte lineaarisesti sellaisina komponentteina, jotka toisistaan tilastollisesti riippumattomia (Hyvärinen ja Oja 2000). Tämä menetelmä mahdollistaa saadusta syötteestä sen ydinrakenteen ja oleellisten ominaisuuksien erottelun. Tästä on hyötyä koneoppimisessa, kun käytettävissä oleva syöte sisältää satunnaisuutta ja on vaikea määritellä ennalta mitkä tiedot syötteessä liittyvät ratkaistavaan ongelmaan. Itsenäinen komponenttianalyysi on houkutteleva menetelmä kurssihistorian käsittelyssä, sillä sen avulla voidaan vähentää syötteen satunnaisuutta sekä erotella komponentteja, jotka liittyvät ratkaistavaan ongelmaan. Erityisesti haasteena on vaikeus intuitiivisesti valita tietoa, joka varmasti korreloisi kurssikäytöksen kanssa ja näin ollen on hyödyllistä pyrkiä karsimaan syötteestä niitä tietoja, jotka eivät lisää saatavilla olevan tiedon määrää koneoppimiselle.

4 Koneoppimismallien suoriutuminen

Tässä tutkielmassa koneoppimismallien suoriutumista arvioitiin valitsemalla joukko erilaisia tutkimuksia ja arvioimalla näissä saatuja tuloksia. Eräs tähän liittyvä oleellinen huomio on tutkimusten eriävät syötteet. Käsiteltyjen kohde-etuuksien tai niistä valittujen syötteiden poiketessa toisistaan ei eri tutkimusten tulokset ole suoraan verrattavissa keskenään, vaan ainoastaan saman tutkimuksen sisällä olevia algoritmeja on mielekästä verrata suoraan toisiinsa. Tämän takia tutkimusten tulokset arvioidaan erikseen ja yleisemmät arviot eri algoritmien ja mallien suoriutumisesta muodostetaan ilman lukujen vertailua tutkimusten välillä.

Luvut 4.1 ja 4.2 käsittelevät yksinkertaisempien mallien suoriutumista keskittyen luvussa 3 esiteltyihin algoritmeihin. Luvuissa 4.3 ja 4.4 siirrytään tarkastelemaan hybriditoteutuksia, joissa käytettyihin algoritmeihin yhdistetään syötteen optimointia arvioiden millaisia hyötyjä näillä toteutuksilla voidaan saavuttaa verrattuna yksittäisen algoritmin suoriutumiseen.

4.1 Tukivektorikone

Kuten luvussa 2.4 esitettiin, voidaan pelkän osakkeen hinnan sijaan vaihtoehtoisesti käyttää syötteenä teknisiä indikaattoreita. Patel ym. 2015 käyttivät tutkimuksessaan kymmentä erilaista teknistä indikaattoria syötteenä koneoppimiselle, mutta syytä valituille indikaattoreille ei perusteltu tarkemmin. Kim 2003 käytti tutkimuksessaan myös teknisiä indikaattoreita syötteenä algoritmeille, joita vertasi keskenään. Tutkimuksen aiheena oli tutkia tukivektorikoneen soveltuvuutta kurssikäyttäytymisen ennustamiseen, sillä tuolloin suurin osa tutkimuksista oli keskittynyt luvun 3.2 mukaisiin neuroverkkoihin. Itse vertaillut algoritmit olivat tukivektorikone, neuroverkko vastavirta (BP, engl. *Backpropagation*) -algoritmilla sekä tapauskohtainen päättely (CBR, engl. *case-based reasoning*).

Tutkimuksen tuloksissa tukivektorikone pärjäsi paremmin kuin vertailtavana olleet algoritmit yltäen 57.8 % osumatarkkuuteen, joka oli kolme prosenttiyksikköä parempi, kuin neuroverkolla. Tay ja Cao 2001 saivat tutkimuksessaan vastaavat tulokset, joiden perusteella tukivektorikone pärjäsi muita algoritmeja kuten vastavirta-algoritmilla toimivaa neuroverkkoa paremmin. Näiden tulosten perusteella tukivektorikoneen parempien tulosten lisäksi oleelli-

nen havainto on, että tulokset eroavat toisistaan. Tämä viittaa saadun syötteen sisältävän joitain malleja, mitä koneoppiminen pystyy mallintamaan eikä saatu syöte ole ollut pelkästään satunnaista.

4.2 Neuroverkko

Moghaddam, Moghaddam ja Esfandyari 2016 testasi tutkimuksessaan neuroverkkoa vastavirta-algoritmilla NASDAQ-osakeindeksin ennustamiseen. Eroavaa useaan muuhun tutkimukseen oli syöte, joka ei ollut teknisten indikaattoreiden tuloksia, vaan aikaisempien päivien kurssitietoja sekä senhetkinen viikonpäivä. Erityisesti viikonpäivän sisällyttäminen syötteeseen oli muista tutkimuksista poikkeavana valintana tulosten kannalta mielenkiintoinen.

Kurssikäytöksen voidaan olettaa olevan erilaista viikon ensimmäisenä ja viimeisenä päivänä, kuin keskellä viikkoa. Erityisesti viikonlopun aikana julkaistavat uutiset tai tapahtumat maailmalla pääsevät realisoitumaan tehokkaiden markkinoiden mukaisesti vasta ennakkomarkkinoilla (engl. *pre-market*) ja erityisesti vasta pörssin auetessa maanantaina, jolloin kurssien heilunnan määrä saattaa olla tilastollisesti isompi. Myös perjantain kurssikäyttäytymisessä on mahdollista olla tietynlaista johdonmukaisuutta, sillä aktiivikauppaa harjoittavat tahot eivät pysty realisoimaan omistuksiaan tarpeen vaatiessa viikonlopun aikana, jos maailmalla tapahtuu jotain, joka tulee maanantaina vaikuttamaan kurssiin. Lyhyemmällä aikavälillä kauppaa käyville viikonlopun yli omistusten pitäminen tuo lisää riskiä ja voi olla mieluisampaa sulkea sillä hetkellä auki olevat kaupat perjantaina, joka lisää myyntiä. Molemmat tapaukset ovat erityisesti psykologisia tekijöitä, jotka teknisen analyysin perusteella ovat niitä asioita, joista trendit ja ennakoitava kurssikäyttäytyminen muodostuvat.

Tutkitun mallin tarkkuutta mitattiin selitysasteella (R^2 engl. *Coefficient of Determination*) sekä jäännösvarianssilla (MSE, engl. *mean square error*). Tutkimuksessa testattiin useita eri rakenteita neuroverkolle, kuten piilokerrosten ja neuroneiden määrän muuttamista. Optimaalisimmat rakenteet osoittivat, että tutkimuksessa ei ollut oleellista merkitystä saiko neuroverkko syötteeksi neljän vai yhdeksän viimeisimmän päivän kurssitiedot. Esiin nostetuille malleille R^2 arvot olivat 0.9408 neljän päivän syötteelle sekä 0.9622 yhdeksän päivän syötteelle.

Tulokset vaikuttavat suurilta, mutta on oleellista huomata, että tutkimuksen malli ennusti suoraan tulevan päivän kurssia eikä binääristä arvoa siitä laskeeko vai nouseeko kohde-etuus seuraavana päivänä. Tulokset eivät siis suoraan kerro kuinka hyvin malli toimisi aktiiviseen kaupankäyntiin, vaan miten tarkasti se ennusti tulevaa kurssia. Neuroverkko voi tulosten perusteella toimia kurssikäyttäytymisen ennustamiseen, jos sen rakennetta optimoidaan tarpeeksi, sillä tutkimuksessa huomattiin tulosten heikentyminen reilusti, kun esimerkiksi piilokerroksia oli liikaa. Mahdollinen tapa rakenteen optimointiin olisi käyttää luvussa 3.3 esiteltyjä geneettisiä algoritmeja, eikä manuaalista testaamista kuten kyseisessä tutkimuksessa on tehty.

Suorien lukujen lisäksi oleellinen havainto on, että vaikka tulokset paranivat, kun neuroverkko sai useamman päivän kurssitiedot syötteeksi, oli tämän vaikutus tarkkuuteen verrattaen vähäinen. Olisi tarpeellista testata miten tulokset paranevat, jos syötteen määrää kasvatettaisiin vielä enemmän esimerkiksi 20 päivän ajalle ja katsoa onko saatu hyöty edelleen vähäinen. Tämän hetkisillä tuloksilla voidaan olettaa, että aikaisempien päivien kurssien vaikutus tulevaan vähenee hyvin nopeasti mitä kauemmas tästä hetkestä mennään. Tämä on tehokkaiden markkinoiden hypoteesin mukainen havainto, sillä aikaisemmasta kurssikäyttäytymisestä ei pitäisi voida ennustaa tulevaa. Eräs selitys saaduille tuloksille olisi, että psykologiset tekijät sekä kurssin ennustettavuus ilmenee lähellä nykyhetkeä olevassa kurssissa ja sen merkitys vanhenee nopeasti.

4.3 Hybriditoteutukset itsenäisellä komponenttianalyysilla

Ince ja Trafalis 2017 testasivat yhdistää eri algoritmeja muodostaen hybriditoteutuksia. Testattuja algoritmeja olivat luvun 3.1 tukivektorikone (SVM), kaksoistukivektorikone (TWSVM, engl. *Twin Support Vector Machine*), Fisherin erotteluanalyysi (KFDA, engl. *Kernel Fischer Discriminant Analysis*), min-max-todennäköisyyskone (MPM, engl. *minmax probability machines*) sekä luvun 3.4 itsenäinen komponenttianalyysi (IDA, engl. *Independent Component Analysis*). Jokaisen tutkitun algoritmin suoriutumista testattiin erikseen sekä yhdistettynä itsenäiseen komponenttianalyysiin. Itsenäinen komponenttianalyysi oli tutkimuksen oleellinen osa, sillä sitä käytettiin valitsemaan syötteeksi käytettyjä teknisiä indikaattoreita optimoiden muiden algoritmien saamaa syötettä.

Tutkimuksen tapauksessa alkuperäinen historiallinen data käsiteltiin 12 eri teknisellä indikaattorilla, joista itsenäisellä komponenttianalyysillä valittiin ne, jotka osoittivat vähiten tilastollista riippuvuutta. Tämä on tunnettu asia myös teknisen analyysin parissa, jossa käytettyjä indikaattoreita valitaan yleensä sen perusteella, että ne kuvastavat eri ominaisuuksia tutkitusta kurssista. Monet indikaattorit näyttävät saman asian, mutta eri tavalla. Itsenäistä komponenttianalyysia voidaan siis käyttää erottelemaan ne indikaattorit, jotka kuvastavat toisistaan riippumattomia ominaisuuksia kurssikäyttäytymisessä. Seuraavaksi näiden indikaattorien tulokset syötettiin koneoppimisalgoritmille. Tutkittujen algoritmien suoriutumista testattiin ennustamalla osakekurssin suuntaa seuraavana päivänä tehden ratkaistavasta ongelmasta binäärisen. Valittuja kohde-etuuksia olivat osakeindeksit Down Jones Industrial, joka koostuu Yhdysvaltojen suurimmista teollisuusyrityksistä, NASDAQ-komposiitti, joka painottaa erityisesti teknologiayrityksiä sekä S & P 500 -indeksi, joka koostuu Yhdysvaltojen 500 suurimmasta yrityksestä. Käytetty kurssihistoria oli vuosilta 2007 - 2015.

Tuloksista havaitaan selkeästi hybriditoteutuksien potentiaalisuus. Kaikkien algoritmien osumatarkkuudet paranivat, kun ne yhdistettiin itsenäiseen komponenttianalyysiin saadun syötteen optimointia varten. Luvussa 4.1 tutkittu tukivektorikone osoittautui myös tässä tutkimuksessa tehokkaimmaksi yltäen korkeimmillaan 86 % osumatarkkuuteen. On oleellista pitää mielessä, että osumatarkkuus on vertailukelpoinen vain muiden algoritmien kanssa, joita on testattu samalla aikavälillä. Olettaen, että markkinat nousevat tai laskevat yli puolet tarkastelluista päivistä niin vakiofunktio, joka ennustaa aina nousua tai laskua voisi myös saavuttaa yli 50 % osumatarkkuuden. Kyseisessä tutkimuksessa kerrottiin käytetty aikaväli, mutta tuloksissa ei ollut vertailukohteena vakiofunktioiden suoriutumista osumatarkkuuden osalta.

Algoritmien suoriutumista arvioitiin myös simuloimalla aktiivikauppaa, joko ostamalla kohde-etuutta päivän alussa tai pitäen omistukset käteisenä päivän ajan sen mukaan mitä testattu algoritmi suosittelee. Ennustuksen osuessa oikeaan hyödytään vain kurssien noususta, mutta ei menetetä pääomaa kurssien laskuun. Tuloksia mitattiin ylituotolla eli alfalla (α), joka on määritelty odotetun tuoton ylittävänä osuutena sekä Sharpen luvulla, joka mittaa tuottoja suhteessa riskiin. Tässä simulaatiossa odotettu tuotto on se tuotto, joka saadaan, kun kohde-etuutta omistettaisiin koko aikavälin ajan. Tehokkaiden markkinoiden hypoteesin perusteella

ylituottoa ei ole mahdollista saavuttaa pitkällä aikavälillä ilman sattumaa.

Parhaimmaksi toteutukseksi osoittautui itsenäinen komponenttianalyysi yhdistettynä tukivektorikoneeseen, joka saavutti ylituottoa 25.61 % S&P 500 indeksissä. Tulokset viittaavat siihen, että algoritmit ovat oppineet mallin, joka ennustaa kurssien käyttäytymistä seuraavana päivänä paremmin kuin satunnainen arvaus. Ylituoton ongelma suoriutumisen mittaamisessa on määritellä, muodostuiko tuotto tasaisesti onnistuneista kaupoista, vai oliko seassa pieni osa kauppoja, jotka muodostivat valtaosan tuotoista. Tällöin sattuma voisi vaikuttaa oleellisesti tuloksiin. Toinen mittari eli Sharpen luku oli kaikissa simulaatioissa alle yhden, joka kertoo riskin olleen suuri suhteessa tuottoihin. Tämä viittaa myös siihen, että saavutettu ylituotto ei ole ollut tasaista vaan se on koostunut pienemmästä joukosta onnistuneita kauppoja. Oleellinen havainto on, että eri algoritmien väleillä on isoja eroja sekä eri tutkimuksissa tukivektorikone on suoriutunut parhaiten. Tämän perusteella kurssidata ei ole täysin satunnaista vaan koneoppiminen ja erityisesti tukivektorikone oppii löytämään tiettyjä malleja kurssien käytöksestä.

Aktiivikaupassa simuloitu aikaväli oli 437 päivää eli noin 2 vuotta pörssipäiviä. Tälle ajalle saavutettuja ylituottoja voidaan pitää hyvinä, mutta jos huomioon otetaan kaupankäyntikulut niin näin aktiivinen kaupankäynti ei olisi käytännössä kannattavaa riskiin ja kuluihin nähden. Tämän tutkielman kannalta oleellinen havainto on, että kurssikäytöksen ennustaminen arvausta paremmin ei vielä johda suoraan ylituottoihin.

4.4 Hybriditoteutus geneettisillä algoritmeilla

Choudhry ja Garg 2008 testasivat hybriditoteutusta käyttäen syötteen optimointiin luvun 3.3 geneettisiä algoritmeja. Ennustetuiksi kohde-etuuksiksi valittiin kolme eri osaketta, mutta muista tarkastelluista tutkimuksista poiketen osakkeen kurssihistorian lisäksi valittiin samat tiedot myös muista osakkeista, joilla oli korkea korrelaatio kohde-etuuden kanssa. Tämä on mielenkiintoinen valinta, sillä kuten luvussa 4.2 havaittiin niin kurssihistorian hyödyllisyys laskee nopeasti mitä kauemmas nykyhetkestä mennään. Osakkeet, joiden korrelaatio on korkea voivat tarjota lisää hyödyllistä syötettä lähellä nykyhetkeä kohde-etuuden ennustamiseen. Syöte käsiteltiin muiden tutkimusten tapaan teknisillä indikaattoreilla, joita tässä

tutkimuksessa valittiin 35. Tämän jälkeen esikäsitellystä syötteestä valittiin luvun 3.3 geneettisillä algoritmeilla ne arvot, jotka lopuksi päätyvät luvun 3.1 tukivektorikoneelle. Testatun mallin suoriutumista mitattiin osumatarkkuudella.

Tuloksista havaitaan, että tukivektorikoneen osumatarkkuus parani jokaisen osakkeen kohdalla noin 3-4 prosenttiyksikköä, kun sen saamaa syötettä optimoitiin geneettisillä algoritmeilla. Saavutetut osumatarkkuudet olivat 55 % - 62 % välillä. Luvun 3.4 tapaan tämä viittaa hybriditoteutusten potentiaalisuuteen ja itse koneoppimisalgoritmin saaman syötteen valinnan tärkeyteen. Tutkimuksessa ei esitetty vertailukohteena vakiofunktion suoriutumista, mutta osumatarkkuuden parantuminen syötteen optimoinnilla viittaa siihen, että esitetty malli poikkeaisi satunnaisesta arvauksesta. Tehokkaiden markkinoiden hypoteesin heikkojen ehtojen perusteella aikaisemmalla kurssihistorialla ei voida ennustaa tulevaa, mutta saadut tulokset myös tämän tutkimuksen osalta viittaisivat siihen, että tämä ei ole aina pitänyt paikkaansa.

5 Yhteenveto

Tässä tutkielmassa käsiteltiin koneoppimisen soveltuvuutta rahoitusmarkkinoiden ennustamiseen. Saatujen tulosten perusteella luvussa 3.1 esitelty tukivektorikone, vaikuttaa lupaavimmalta algoritmilta satunnaisuutta sisältävän datan analysointiin. Luvussa 4.2 huomattiin, että pelkän kurssihistorian määrän kasvattaminen syötteessä vähensi sen hyödyllisyyttä nopeasti, mitä kauemmas tarkasteltavasta hetkestä mentiin. Tätä ongelmaa pystytään minimoimaan luvuissa 4.3 ja 4.4 käsitellyillä hybriditoteutuksilla, joilla pyritään parantamaan syötteen laatua. Näistä tuloksista havaittiin, että mallien suoriutuminen ei ole kiinni pelkästään valitusta koneoppimisen algoritmista vaan käytetyn syötteen valinta ja sen laatu vaikuttavat enemmän tuloksiin. Rahoitusmarkkinoiden ennustaminen on siis enemmän käytetyn syötteen valinnan ja optimoinnin ongelma, kuin oikean algoritmin valinnan.

Käsitellyissä tutkimuksissa kurssihistorian sisältämän satunnaisuuden muodostamaa ongelmaa lähestyttiin esikäsittelemällä kurssihistoria luvussa 2.4 esitellyillä teknisillä indikaattoreilla, joilla syötettä pystytään normalisoimaan ja vähentämään näkyvää satunnaisuutta yksittäisissä tarkastelupisteissä. Käytettyjen indikaattoreiden valintaa ei tutkimuksissa oleellisesti perusteltu ja tämä voisi olla yksi tapa pyrkiä parantamaan mallien tuloksia. Toinen oleellinen tutkimusta vaativa kohde olisi selvittää voidaanko tuloksia parantaa, jos syötteeseen lisätään kurssihistorian lisäksi muita tietoja, kuten uutisia, korkoja, tai valuuttakursseja. Tällä voitaisiin osittain pyrkiä ratkaisemaan hyödyllisen tiedon vähäisyys malleissa.

Tutkimustulokset osoittavat, että koneoppimisella on pystytty muodostamaan tekniseen analyysiin pohjautuvia malleja, jotka ovat ennustaneet kurssikäytöstä satunnaista arvausta paremmin, mikä on ristiriidassa tehokkaiden markkinoiden hypoteesin heikkojen ehtojen kanssa, johtaen kahteen johtopäätökseen:

1. Tehokkaiden markkinoiden hypoteesin heikot ehdot eivät ole aina toteutuneet täydellisesti rahoitusmarkkinoilla.
2. Koneoppiminen soveltuu rahoitusmarkkinoiden sekä satunnaista dataa sisältävien ongelmien ennustamiseen.

Lähteet

Burton, Maureen, Reynold F Nesiba ja Bruce Brown. 2015. *An introduction to financial markets and institutions.* Routledge.

Choudhry, Rohit, ja Kumkum Garg. 2008. "A hybrid machine learning system for stock market forecasting". *World Academy of Science, Engineering and Technology* 39 (3): 315–318.

Eiben, Agoston E, James E Smith ym. 2003. *Introduction to evolutionary computing.* Nide 53. Springer.

Hausknecht, Matthew, Joel Lehman, Risto Miikkulainen ja Peter Stone. 2014. "A neuroevolution approach to general atari game playing". *IEEE Transactions on Computational Intelligence and AI in Games* 6 (4): 355–366.

Hyvärinen, Aapo, ja Erkki Oja. 2000. "Independent component analysis: algorithms and applications". *Neural networks* 13 (4-5): 411–430.

Ince, Huseyin, ja Theodore B Trafalis. 2017. "A hybrid forecasting model for stock market prediction." *Economic Computation & Economic Cybernetics Studies & Research* 51 (3).

Kim, Kyoung-jae. 2003. "Financial time series forecasting using support vector machines". *Neurocomputing* 55 (1-2): 307–319.

Malkiel, Burton G. 2007. *A random walk down Wall Street: the time-tested strategy for successful investing.* WW Norton & Company.

Malkiel, Burton G, ja Eugene F Fama. 1970. "Efficient capital markets: A review of theory and empirical work". *The journal of Finance* 25 (2): 383–417.

Moghaddam, Amin Hedayati, Moein Hedayati Moghaddam ja Morteza Esfandyari. 2016. "Stock market index prediction using artificial neural network". *Journal of Economics, Finance and Administrative Science* 21 (41): 89–93.

Patel, Jigar, Sahil Shah, Priyank Thakkar ja Ketan Kotecha. 2015. “Predicting stock market index using fusion of machine learning techniques”. *Expert Systems with Applications* 42 (4): 2162–2172.

Pring, Martin J. 2014. *Technical Analysis Explained, Fifth Edition: The Successful Investor’s Guide to Spotting Investment Trends and Turning Points.* McGraw-Hill Professional.

Tay, Francis EH, ja Lijuan Cao. 2001. “Application of support vector machines in financial time series forecasting”. *omega* 29 (4): 309–317.

Vapnik, Vladimir N. 1999. “An overview of statistical learning theory”. *IEEE transactions on neural networks* 10 (5): 988–999.