LUT University

TUOTANTOTALOUDEN KOULUTUSOHJELMA

Koneoppimisen käyttö markkinoiden ennustamisessa

Predicting stock markets with machine learning

Kandidaatintyö

Juuso Ahlroos

TIIVISTELMÄ

Tekijä:
Juuso Ahlroos
Työn nimi: Koneoppimisen käyttö markkinoiden ennustamisessa

Vuosi: 2019 **Paikka:** Espoo

Kandidaatintyö. LUT-yliopisto, Tuotantotalous.
34 sivua, 7 kuvaa ja 6 liitettä
Tarkastaja: Lasse Metso

Hakusanat: Koneoppiminen, osakemarkkinat, ennustaminen
Keywords: Machine learning, stock markets, forecasting

Työ selvittää, pystyykö osakemarkkinoita mallintamaan koneoppimisen avulla. Työssä käydään läpi yleisimmät algoritmit ja niiden toimintaperiaatteet, sekä mallintamisessa käytettyjä muuttujia. Työ on kirjallisuuskatsaus.

Osakemarkkinoiden ennustaminen on kiinnostanut akateemikkoja ja alalla työskenteleviä ihmisiä jo pitkään. Markkinoiden voittamiseen sovelletaankin yhä hienostuneempia menetelmiä ja uusien, toimivien mallien, löytämiseksi on käytettävä yhä enemmän aikaa. Koska osakemarkkinat on monisyinen ongelma, koneiden käyttö mahdollistaa erilaisten syy-seuraussuhteiden tunnistamisen.

Tietokoneiden kehittynyt laskentateho, internetin mahdollistama datan keräys ja edistykset algoritmeissa ovat luoneet pohjan koneoppimisen hyödyntämiseen useissa erilaisissa ongelmissa. Koneoppimisen avulla voidaan löytää datasta yhteyksiä, joita ihminen ei pystyisi mitenkään havaitsemaan. Koneoppimisen käyttö vaatii kuitenkin datan sekä algoritmien tarkkaa valitsemista, datan esikäsittelyä ja parametrien muokkausta. Tämä johtaa siihen, että mahdollisia malleja on tarjolla lähes rajattomasti. Parhaan mallin löytäminen ei siis ole helppoa, ja sen eteen on tehtävä paljon vertailua. Tässä työssä pyritään selvittämään kuinka hyvin mallit voivat suoriutua ja kehittämään perusteet hyvän mallin luomiseksi, sekä mallien vertailuksi.

SISÄLLYSLUETTELO

1. JOHDANTO

Markkinoiden ennustaminen on ollut kysymys, jonka vastausta on pyritty selvittämään niin kauan, kuin markkinat ovat olleet olemassa. Tämä johtuu siitä, että markkinoilla liikkuu valtavia määriä rahaa ja ennustuksessa onnistunut voikin kasvattaa omaisuuttaan merkittävästi. Jokainen saattaa onnistua saamaan markkinoita parempaa tuottoa, sillä satunnaismenetelmällä todennäköisyys voittaa markkinat yhtenä vuotena on noin 50%. Kahtena peräkkäisenä 25%, kolmena 12.5% ja niin edelleen. Ilman toimivaa ennustemenetelmää esimerkiksi kymmenen peräkkäisen voittavan vuoden saaminen onkin erittäin epätodennäköistä, 0.5^10 = 0.1%.

Aikojen saatossa osakemarkkinoille on kehittynyt useita teorioita mahdollisista ennustusmenetelmistä, mutta samalla myös monet tutkimukset ovat osoittaneet markkinoiden käyttäytymisen olevan satunnaista. Ennustamisessa on mahdollista käyttää useita erilaisia syötteitä ja esimerkiksi Patel, Shah, Thakkar, Kotecha (2015a) ovat onnistuneet siinä teknisten indikaattorien avulla, kun taas Rikkinen (2019) on löytänyt yhteyksiä Twitterin käyttäjien aktiivisuuteen.

Työssä käydään läpi yleisimmät koneoppimisen algoritmit ja niiden toimintaperiaatteet, sekä mallintamisessa käytettyjä muuttujia. Tutkimus vastaa kysymyksiin:

- *Minkälaisia tuloksia koneoppimisella on saatu markkinoiden ennustamisessa?*
- *Mitkä ovat yleisimmin koneoppimisen kanssa käytetyt syötteet*?

Työ on kirjallisuuskatsaus. Työ alkaa teoriaosuudella, jossa selitetään markkinoiden toiminta ja miksi ennustaminen on ongelmallinen käsite. Markkinateoriassa käydään myös läpi tapoja mitata ennustamisessa onnistumista. Sen jälkeen koneoppimisosuudessa käydään yleisperiaatteita koneoppimisen toiminnasta ja kuvataan yleisimpiä osakemarkkinoiden ennustamisessa käytettyjä algoritmeja. Osakemarkkinat ja koneoppiminen –osiossa perehdytään tutkimuksiin, joissa ollaan käytetty koneoppimista markkinoiden ennustamiseen. Siinä pyritään vertailemaan erilaisia algoritmeja ja listaamaan ennustamisessa käytettyjä syötteitä.

Rajaan työn koskemaan osakemarkkinoihin, mutta käytän myös kirjallisuutta, joissa käsitellään johdannaisia, indeksejä ja valuuttoja. En näe järkevänä karsia näitä, sillä vaikka kyseisiä tuotteita voidaan pitää eri asioina kuin osakkeita, ovat ne silti toiminnallisesti hyvin

samanlaisia. Koneoppimisen osalta rajaan työn koskemaan tunnetuimpia algoritmeja, enkä käsittele tutkimuskohtaisia pieniä muokkauksia, vaan keskityn toiminnallisuuteen. Samojen algoritmien erilaiset versioit ja erityisesti niiden matemaattinen tausta ovat liian laajoja käsiteltäviksi työhön, joten jätän ne tutkimuksesta pois.

Työ kokoaa aikaisempaa kirjallisuutta osakemarkkinoiden ennustamisesta koneoppimisella ja luo pohjan tutkimuksen laajentamiselle rahoitusteorian suuntaan. Työssä kritisoin erityisesti aikaisemmassa kirjallisuudessa käytettyjä mittausmenetelmiä ja pyrin kertomaan, miksi niitä on kehitettävä. Työ pyrkiikin yhdistämää koneoppimisesta mallintamisen ja mallien luomisen rahoitusteorian mukaiseen tuottojen mittaamiseen ja mallien arviointiin.

2. MARKKINAT

Markkinat määräävät osakkeen hinnan. Ne toimivat kysynnän ja tarjonnan perusteella, eli kysynnän kasvu nostaa hintaa, kun taas tarjonnan kasvu laskee sitä. Rahoitus on omana tieteenalana suhteellisen uusi mutta osakemarkkinoiden toimintaa on pyritty kuvaamaan usealla erilaisella teorialla, kuten yleisesti hyväksytyllä Random Walk –teorialla (Fama, 1995), jonka mukaan edellisten päivien muutoksilla on vain erittäin vähän ennustusvoimaa seuraavien päivien muutoksiin, jolloin päiviä voi pitää toisistaan riippumattomina. Vaikka hinta määrittyy kysynnän ja tarjonnan perusteella, satunnaisuuden vaikutus onkin silti merkittävä erityisesti lyhyen aikavälin toiminnassa, joka tekee ennustamisesta hankalaa.

2.1 Efficient Market Hypothesis

Tehokkaiden markkinoiden hypoteesi, Efficient market hypothesis, (EMH) väittää markkinoiden olevan tehokkaita joko heikosti, keskivahvasti tai vahvasti. Markkinoiden tehokkuus tarkoittaa sitä, että markkinat kuvaavat jo kaikkea olemassa olevaa tietoa. EMH:n mukaan markkinat voivat olla tehokkaita eri tasoisesti. Tasot heikko, keskivahva ja vahva kuvaavat, mitä kaikkea tietoa hintoihin on jo sisällytetty, eli mitä tietoa hyödyntämällä markkinoita ei voi enää voittaa. Heikko tehokkuus tarkoittaa, että markkinoilla oleva hinta sisältää kaiken historiallisen osakkeen hintatiedon. Keskivahva tehokkuus tarkoittaa, että markkinahintoihin on sisällytetty kaikki yrityksien fundamentaaliset tiedot ja vahva tehokkuus tarkoitta, että hinnoissa on myös kaikki sisäpiiritieto. (Malkiel & Fama 1970)

Tutkimuksessani pyrin selvittämään, voiko koneoppimiselle ennustaa markkinoita. Jos tämä on mahdollista, markkinoiden tehokkuus on kyseenalaista. Kaiken tiedon ollessa sisällettynä hintaan, ei markkinoita pitäisi pystyä myöskään ennustamaan. Jos ennustaminen on mahdollista, se rikkoo mahdollisesti jotain EMH:n tasoista. *” Tehokkailla markkinoilla yksikään sijoittaja ei pysty millään investointistrategialla tai -tyylillä saavuttamaan ylisuuria tuottoja”* (Knüpfer & Puttonen 2018).

2.2 CAPM

Random Walk ja EMH selittävät markkinoiden toimintaa teoriassa. Käytännössä asia on kuitenkin haastavampi. Ensimmäisiä käytäntöön sovellettavia viitekehyksiä osakemarkkinoiden toiminnasta on Capital Asset Pricing model (CAPM) (Merton 1973). CAPM mukainen yhtälö 1:

$$E_r = R_f + \beta * (E_{rm} - R_f) \quad (1)$$

E_r = Odotettu tuotto

R_f = Riskitön korko

β = Yrityksen riskisyys verrattuna markkinoihin

$E_{rm} - R_f$ = Markkinoiden riskipreemio

Yhtälön mukaan eri osakkeiden odotettu tuotto riippuu pelkästään osakkeen riskisyydestä. Näin suurimman odotetun tuoton saa sijoittamalla riskisimpään osakkeeseen. Ylituotot ovat CAPM:n mukaan tuottoja, joita ei siis pysty selittämään osakkeen riskisyydellä. Ne esitetään CAPM yhtälössä alfalla, yhtälö 2.

$$E_r = \alpha + R_f + \beta * (E_{rm} - R_f) \quad (2)$$

α = vakiotermi tai niin sanottu ylituotto

EMH mukaan regressiosta saatu CAPM:n yhtälö, jossa alfa ei ole nolla on mahdoton, sillä se implikoisi ylituottoja.

2.3 Three -factor model

Eugene Fama ja Kenneth French (1992) tutkivat yrityksen riskisyyden ja tuottojen suhteita tarkemmin ja he huomasivat, että osakkeen riskisyys ei ole ainoa tekijä, joka vaikuttaa osakkeen tuottoihin. He kehittivät siten 3-factor –mallin, jossa tuottoja on riskisyyden lisäksi selittämässä yrityksen koko ja kirja-arvon ja markkina-arvon suhde, yhtälö 3.

$$E_r = \alpha + R_f + \beta * (E_{rm} - R_f) + \beta_{smb} * SMB + \beta_{hml} * HML \quad (3)$$

SMB = Small minus big, kokoefekti, miten yrityksen koko vaikuttaa tuottoon

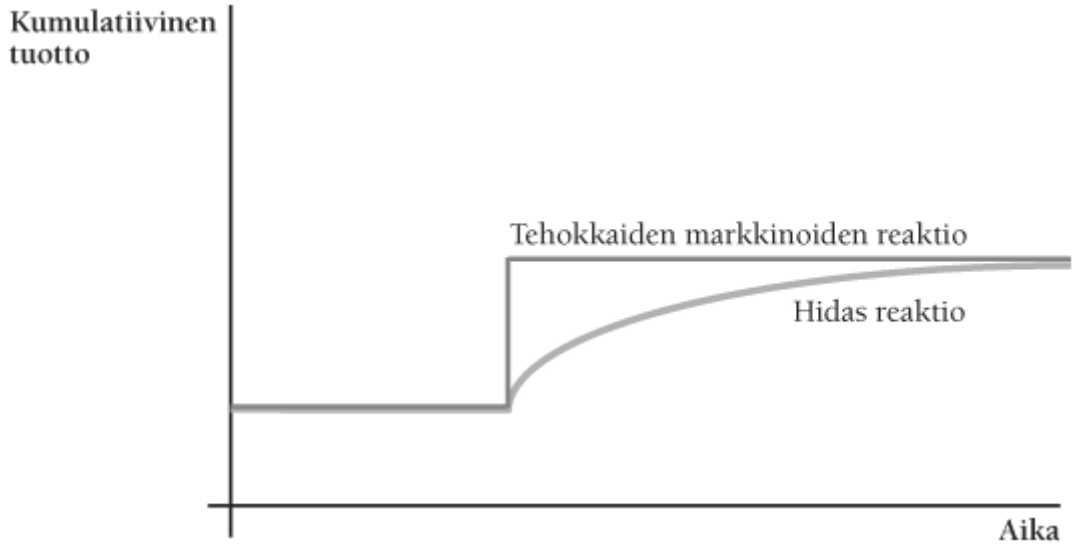
β_{smb} = kokoefektin kerroin yritykselle

HML = High minus low, arvoefekti, miten yrityksen kirja-arvo vaikuttaa tuottoon

β_{hml} = arvoefektin kerroin yritykselle

2.4 Markkinoiden ennustaminen

Markkinoiden ennustaminen ei ole yksinkertaista, sillä kysyntään ja tarjontaan perustuvat markkinat estävät tehokkaasti ennusteisiin perustuvan ylituoton saamisen. Tämä johtuu siitä, että ennuste hinnan noususta lisää osakkeen kysyntää. Kysynnän ja tarjonnan lain mukaan tämä nostaa osakkeen hintaa, kunnes tasapainopiste saavutetaan. Ennuste onkin sen julkaisun jälkeen osa tietoa, joka on jo sisällytetty osakkeen hintaan ja EMH:n mukaisesti ylituottoja ei enää ole saatavilla. Kuvassa 1 nähdään, kuinka tehokkaat markkinat reagoivat teorian mukaan välittömästi, kun taas todellisuudessa reaktio on hitaampi. Koneoppimisen avulla reaktionopeutta näihin tapahtumiin voidaan kuitenkin nopeuttaa, jolloin tehokkaalle hyödyntäjälle on tarjolla käyrien väliin jäävän pinta-alan verran ylituottoja.



Kuva 1: EMH teoria (tehokkaiden markkinoiden reaktio) ja käytäntö (hidas reaktio), (Knüpfer & Puttonen 2018, s. 169).

Puhtaalla lineaarisella regressiolla, esimerkiksi käyttäen CAPM:ia tai three-factor -mallia pystytään selittämään suuri osuus osakkeen oletetusta tuotosta, noin 70% CAPM:lla ja 90% 3 factor -mallilla. Niiden perusteella kaupankäynti ei ole kuitenkaan mielekästä, sillä mallit tunnetaan laajasti, jolloin osakkeet ovat jo EMH:n mukaan hinnoiteltu huomioimaan three-factor -malli ja CAPM. Tämä ei kuitenkaan tarkoita, että mallit olisivat hyödyttömiä, sillä niitä voidaan käyttää strategioiden arvioimiseen. Kun koneoppimisella saadun portfolion sisältöä ja tuottoja käytetään three-factor -regressiossa, saadaan yhtälön vakioterminä alfa, ylituotto. Jos koneoppimismalli tuottaa three-factor -mallilla mitattuna ylituottoa, voidaan sen sanoa voittavan markkinat. Erilaisten rahoitusteorioiden tunteminen onkin tärkeää koneoppimismallien vertailun kannalta. (Fama & French 1992)

Kun kaupankäyntistrategian tuottoja arvioidaan three-factor -mallilla, alfan saavuttaminen ei kuitenkaan automaattisesti tarkoita EMH:n rikkomista. Tämä johtuu siitä, että kun tuottoja arvioidaan jollain mallilla, testaan samalla myös arviointimallin pätevyyttä. Voi ollakin, että three-factor –mallilla mitattuna koneoppimisstrategialla saadaan ylituottoja, mutta paremmalla mallilla ylituotot häviävät. Tämän kaksoisluonteen takia markkinoiden voittamisen mittaaminen ja EMH:n rikkominen on hankalaa. Three-factor –malli selittää kuitenkin suuren osan tuotoista, joten sitä voi pitää kohtuullisen hyvänä mittarina.

3. KONEOPPIMINEN

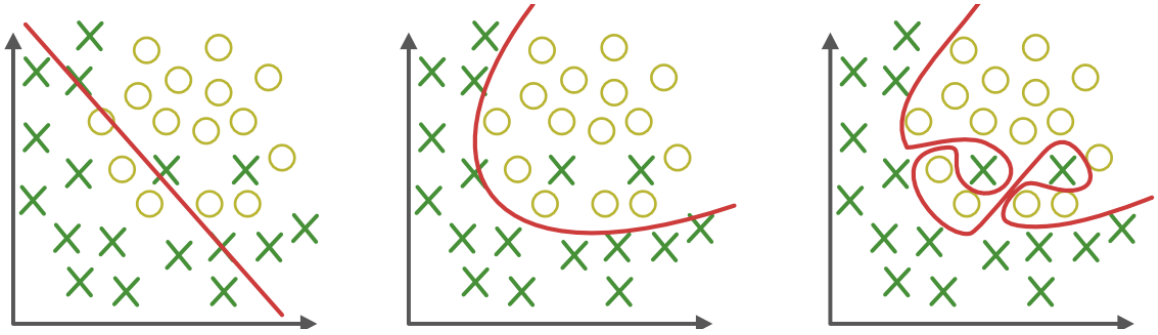
Christopher M. Bishop (2006) kuvaa koneoppimisen hyödyllisyyttä kertomalla, kuinka ihmiset ovat hyviä havaitsemaan aineistossa esiintyviä säännöllisyyksiä. Esimerkiksi listassa [1, 2, 3, 1, 2, 3, 1, 2, 3, …] on nopeasti havaittavissa jakso [1, 2, 3]. Ihmisten kyky on kuitenkin "laskentateholtaan" erittäin rajallinen ja perustuukin ensisijaisesti tuttujen kuvioiden tunnistamiseen sekä lineaarisiin yhteyksiin. Kyky häviää nopeasti, kun parametrien määrä kasvaa tai arvojen välinen yhteys on muuta kuin lineaarinen.

Ohjelmilla on taas erilaisia haasteita näiden jaksojen tunnistamisessa (pattern recognition). Esimerkkitilanteeseen sopivan ohjelman kirjoittaminen on vielä suhteellisen helppoa, mutta isommissa ongelmissa yksinkertainen ratkaisutapa alkaa sisältää paljon sääntöjä, poikkeuksia ja poikkeuksia sääntöihin. Kun sääntöjen ja poikkeuksien määrä kasvaa, se ei ole enää järkevä ratkaisutyyli. Paljon tehokkaammin tuloksia saadaankin näissä tilanteissa käyttämällä koneoppimista (machine learning). (Bishop 2006, Michie, Spiegelhalter & Taylor 1994)

Koneoppimisessa mukautuvaan ohjelmaan syötetään suuri harjoitteluaineisto, jonka jokaisella alkiolla on usein myös kohdevektori. Alkioiden ja kohdevektorien suhdetta voi esittää esimerkiksi funktio $y(x) = t$, jossa x on harjoitteluaineiston alkio ja t on alkion kohdevektori. Funktio $y(x)$ ei ole ensin määritelty mitenkään mutta se saa tietyn muodon ohjelman oppimisvaiheen aikana. Oppimisvaiheessa pyritään optimoimaan funktion $y(x)$ parametreja koko harjoitteluaineiston alueella niin, että lopulta funktio pystyisi tehokkaasti ennustamaan, mikä on uuden, aiemmin kohtaamattoman alkion kohdevektori. Oppimisvaiheen jälkeen mallin toimintaa testataan aineistolla, johon se ei aikaisemmin ole törmännyt (test set). Tässä testataan erityisesti mallin yleistyskykyä (generalization). Käytännössä harjoitteluaineistoon kuuluva data on vain pieni osa kaikista mahdollisista syötteistä, joten mallin on opittava sen perusteella yleistämään ratkaisu muille syötteille. Tämän takia harjoitteluaineiston läpi käynyt malli ei saa olla liian tarkka, koska muuten se menettää kyvyn yleistää ratkaisua uusiin tilanteisiin ja on siten hyödytön. (Hastie, Tibshirani & Friedman 2009)

Mallin testaaminen tehdään usein eri tilastollisilla menetelmillä, esimerkiksi pienin neliöiden summa (ordinary least squares, OLS). Menetelmillä voidaan verrata, kuinka paljon mallin

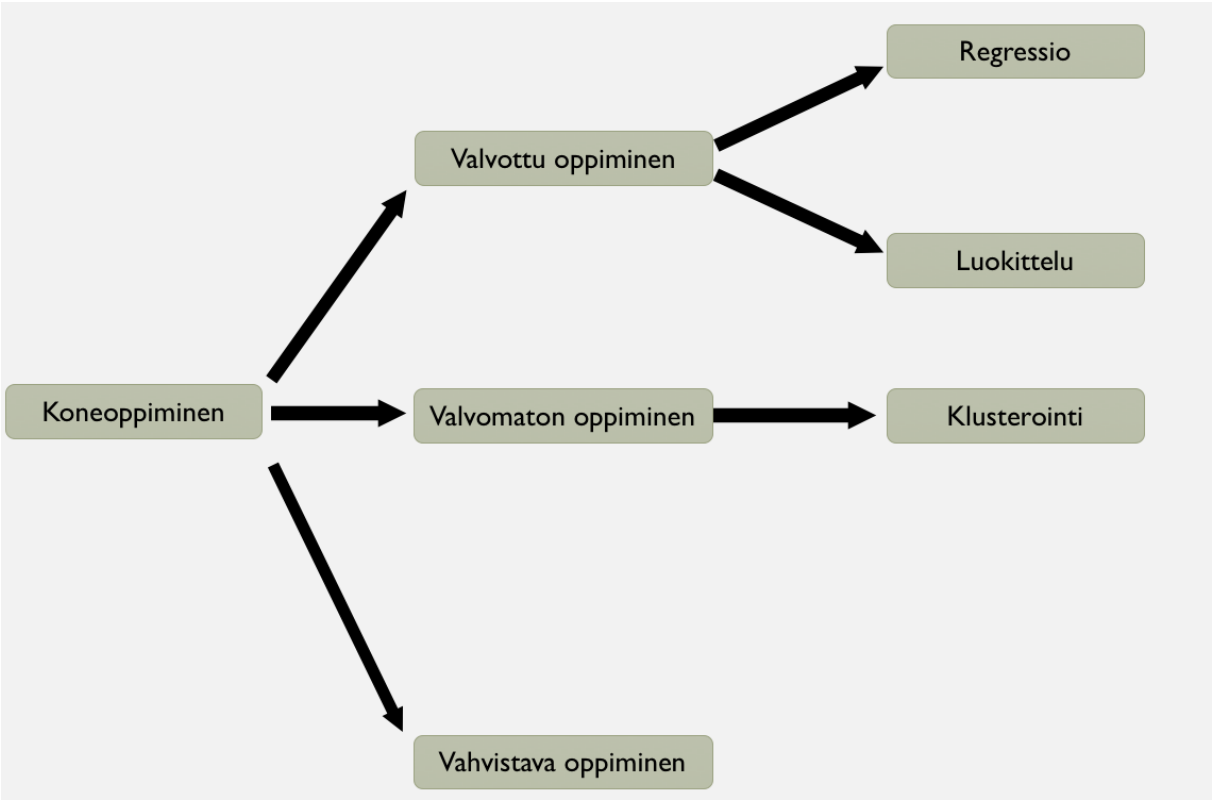
antama arvio poikkeaa kohdevektorin oikeista arvoista ja siten selvittämään mikä malleista antaa tarkimman arvion. Vain yksittäisen tilastollisen menetelmän käyttö voi kuitenkin johtaa erinäisiin ongelmiin, sillä esimerkiksi OLS:llä mitattuna täydellisesti aineistoon sopiva malli ei välttämättä kuvaakaan parhaiten datan kehittänyttä prosessia, vaan kuvaa sitä ns. liian hyvin (over fitting). Kun tällainen malli viedään harjoitteluaineiston ulkopuolelle ja sitä kokeillaan test settiin, osuvuus (fit) häviää ja malli ei kuvaakaan näitä samasta datalähteestä olevia alkioita. Liian hyvin sovitettu malli menettää siis kyvyn yleistää ja se toimiikin vain harjoitteluaineiston kuvaamisessa. (Duda et al. 2012) Kuva 2 esittää vasemmalta oikealle alisovitetun-, oikein sovitetun- ja ylisovitetun mallin. Alisovitettu malli ei juuri jaa alkioita eri ryhmiin, kun taas ylisovitettu tekee sen täydellisesti. Ylisovitettu on kuitenkin huono malli siksi, että se on todennäköisesti menettänyt kykynsä yleistää, ja jakaa vain juuri kyseisen datan hyvin, eikä onnistu jaossa enää uuden datan kanssa. Keskimmäinen, oikein sovitettu malli jakaa datan hyvin ja keskittyy yleiskuvaan.



Kuva 2: Mallien sovitus (Kasturi 2019, s.1)

Koneoppimismallin käyttämä data on usein prosessoitu etukäteen niin, että jaksojen tunnistaminen siitä olisi helpompaa. Esimerkiksi reaaliaikaisessa kasvojentunnistuksessa syötteenä on valtava määrä pikseleitä sekunnissa ja näitten käyttäminen kasvojentunnistusalgoritmissa johtaa mahdollisesti ongelmiin laskentatehon kanssa. Näissä tapauksissa data on jaoteltava osiin, joilla algoritmin ajaminen olisi mahdollisimman tehokasta. Tätä kutsutaan esikäsittelyksi (pre-processing) ja se on tehokas tapa helpottaa lopullisen algoritmin työtä. Esikäsittelyssä on kuitenkin huomioitava, että osa informaatiosta häviää ja jos hävinnyt informaatio on mallille hyödyllistä, se saattaa vaikuttaa loppuratkaisuun. (Kotsiantis et al. 2006)

3.1 Erilaisia koneoppimisen ongelmia



Kuva 3: koneoppimisen jakaminen alakäsitteisiin (Duda et al. 2012)

Kuvassa 3 kuvataan koneoppimisen jakautumista erilaisiin ongelmiin. Koneoppimisen algoritmit hyödyntävätkin lähes aina joko valvottua oppimista, valvomatonta oppimista tai vahvistavaa oppimista erilaisten tehtävien ratkaisemiseen. (Bishop 2006)

Yllä käytetty esimerkkiä, jossa harjoitteluaineisto ja sen kohdevektorit ovat tunnettuja, kutsutaan valvotun oppimisen ongelmaksi (supervised learning problem). Siinä pyritään saamaan syötteiden tuloksiksi tarkka arvo ja mallin vaatimuksena onkin hyvä yleistyskyky. (Kotsiantis et al. 2006)

Valvomattoman oppimisen ongelma (unsupervised learning problem) on lähes samanlainen kuin valvotun oppimisen ongelma, mutta siinä syötteenä on vain alkiot x, jolla ei ole kohdevektoreita t. Tällaisen ongelman tavoitteena on esimerkiksi pyrkiä löytämään datasta ryhmittymiä (clustering) joihin kuuluvilla alkiolla on samanlaisia ominaisuuksia. Tätä voidaan käyttää myös datan visualisoinnissa, jolloin moniulotteista dataa pyritään tuomaan kaksi- tai kolmiulotteiseen muotoon. (Le 2013)

Vahvistava oppiminen (reinforcement learning) on ongelma, jossa kone tekee kunkin valinnan maksimoidakseen siitä saatavan ”palkkion”. Myöskään tässä syötteillä ei ole kohdevektoria, vaan algoritmin on löydettävä tilanne mihin se pyrkii itsenäisesti kokeilemalla (trial and error). Esimerkiksi shakkiin sovellettuna vahvistava oppiminen kokeilisi ensin itseään vastaan pelaten minkälaiset siirrot johtavat lähemmäs tavoiteltua tilaa eli voittoa. Siirrot, jotka vievät algoritmin lähemmäs voittoa saavat suuremman pistearvon, kun muut siirrot. Pistearvojen hakemisen jälkeen algoritmi jatkaa itseään vastaan pelaamista nyt pyrkien maksimoimaan siirtojen pisteet teoriassa päätyen lopulta voittavaan ratkaisuun. Vahvistavan oppimisen algoritmi ei keskity vain nykytilanteeseen vaan ajattelee myös tulevia siirtoja. Esimerkiksi siirto 1 on arvoltaan 100 pisteen arvoinen, kun taas siirto 2 on arvoltaan 10 pistettä. Siirto 2 kuitenkin avaa 1000 pisteen siirron, kun taas siirto 1 jälkeinen paras siirto on vain 10 pistettä. Algoritmi pyrkii ymmärtämään nämä tilanteet ja siten valitsemaan tässä tulevaisuutta ajatellen parhaan vaihtoehdon. Vahvistavan oppimisen pointtina on, ettei algoritmi ensin edes tiedä mihin se pyrkii mutta lopulta oppii valitsemaan liikkeet, jotka tuovat sen lähemmäs voittoa. (Sutton & Barto 1998)

Hankaluuksia vahvistavassa oppimisessa tuottaa erityisesti ongelmien laajuus. Esimerkkinä käytetty shakki sisältää valtavan määrän siirtoja, eikä välttämättä ole edes toteutettavissa tällä tavalla. Olettaen, että se voisi onnistua, ongelmana on erityisesti kuinka paljon algoritmin kannattaa tutkia uusia vaihtoehtoja (explore) verrattuna kuinka paljon sen kannattaa pyrkiä hyödyntämään jo opittuja arvoja (exploit). Jos tutkimisen kerroin on liian suuri, tehokkuus heikkenee ja algoritmi pyrkii kokeilemaan lopulta kaikki mahdolliset siirrot, mikä ei ole mahdollista, koska shakissa mahdollisia pelejä on paljon, 10^27586 (Fate, 2015). Toisaalta, jos exploit –kerroin on liian suuri, algoritmi ei tutustu vaihtoehtoisiin reitteihin vaan pyrkii jokaisella pelikerralla pelkästään maksimoimaan lopputuloksen. Tämä johtaa siihen, ettei algoritmi tutustu kunnolla mahdollisiin vaihtoehtoisiin ratkaisuihin, vaan jää pyörimään samojen siirtojen kanssa kerrasta toiseen. (Silver et al. 2017)

Luokittelu ja klusterointi ovat lähes samoja asioita, jotka eroavat vain sillä, että luokittelussa luokat ovat annettu algoritmille valmiiksi, kun taas klusterointi luo ne datan perusteella. Yksinkertaisena esimerkkinä voisi olla esimerkiksi syötteinä käytetty sademäärää ja

lämpötiloja, jonka jälkeen algoritmi arvioi, mikä kuukausi on kyseessä, eli luokittelee datan kuuluvan todennäköisimmin tietylle kuukaudelle. Ilman pyyntöä jakaa data kuukausiksi, kyse on klusteroinnista, jonka tuloksena data saatetaan jakaa esimerkiksi vuodenaikojen mukaisesti. (Hastie et al. 2009)

Jos lopputuloksessa on yksi tai useampi jatkuva muuttuja, kyseessä on regressio (regression). Esimerkki regressioista voisi olla esimerkiksi kemiallinen valmistusprosessi, jossa muuttujina on raaka-aineiden määrä, paine ja lämpötila ja pyritään koneoppimisen avulla arvioimaan lopputuotteen määrää näiden perusteella. (Harrington 2012)

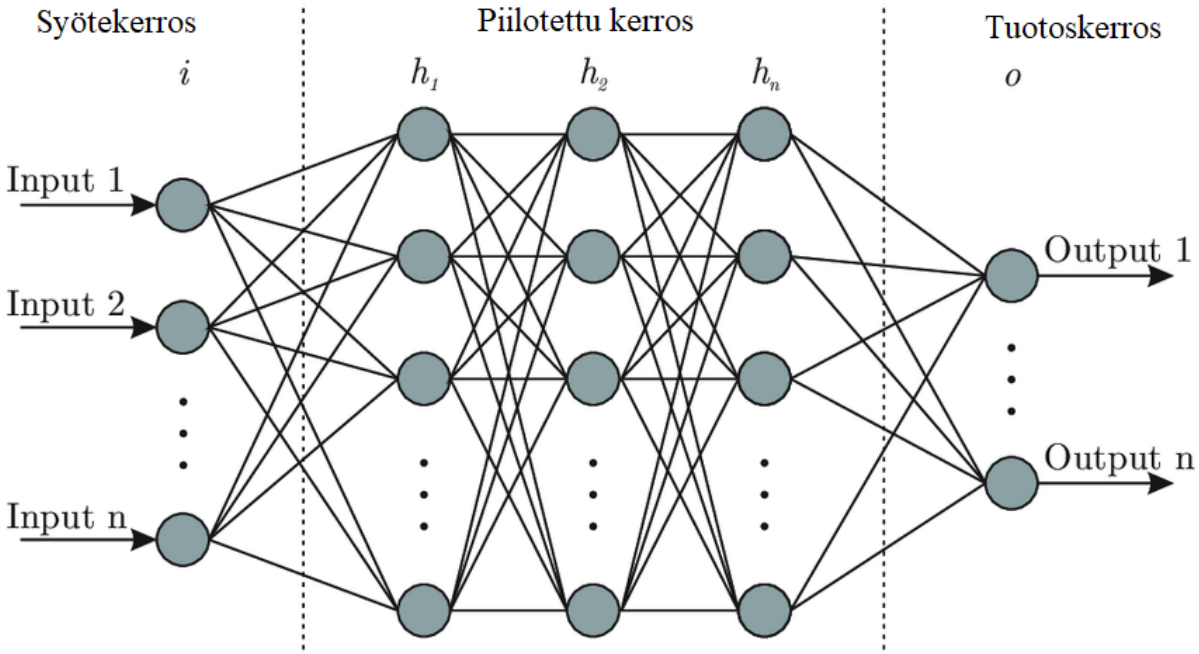
Osakemarkkinoiden ennustamisessa voidaan soveltaa useita eri ongelmia riippuen siitä, mitä käytännössä halutaan ennustaa. Yksinkertaisin vaihtoehto on varmasti valvotun oppimisen ongelma, sillä historiallisesta datasta pystytään kertomaan, menikö osake näillä arvoilla ylös vai alas. Syötteinä siinä voisivat toimia valitut arvot, kuten esimerkiksi hinta edellisen kuukauden aikana ja kohdevektorina on seuraavan päivän hinta. Osakkeisiin voidaan käyttää joko regressioita tarkan hinnan ennustamiseen, tai luokittelua muutoksen suunnan ennustamiseen. Tutkimuksissa näin olikin tehty, ja vaihtoehtoja kutsuttiin suunta- ja tasomalleiksi (Enke & Thawornwong 2005).

3.2 Koneoppimisen algoritmeja

Yllä kuvattujen ongelmien toteuttamiseen voidaan käyttää erilaisia algoritmeja. Kaikki algoritmit eivät sovellu jokaiseen ongelmaan, mutta ymmärrys myös niiden toiminnasta on tärkeää koneoppimisen toiminnan ymmärtämiseksi. Alla kuvataan useimmin kirjallisuudessa esiintyviä algoritmeja ja niiden toimintaa. Perusrakenteisiin on myös vaihtoehtoina paljon erilaisia muotoja, mutta toimintaperiaatteet pysyvät samoina.

Neuroverkko (neural networks, NN) on suosittu algoritmi, jota on usein käytetty osakemarkkinoiden ennustamisessa. Neuroverkkojen toiminta perustuu rakenteellisesti aivojen toimintaan, mutta käytännössä ne ovat erilaisia. Vaikka neuroverkko terminä kuulostaa hankalalta, toiminta on käytännössä suhteellisen yksinkertaista ja neuroverkko koostuukin useista kerroksista lineaarisia regressioita. Neuroverkkoja käytetään esimerkiksi kuvan- ja

puheentunnistamisessa tai kääntämisessä. Neuroverkkojen opettaminen on laskentatehollisesti raskasta, koska lopputulosta lähestytään hidastuvaa vauhtia, mutta niiden etuna on erityisesti se, että oppimisvaiheen jälkeen käyttö ennustamisessa on erittäin tehokasta. (Bishop, 2006)

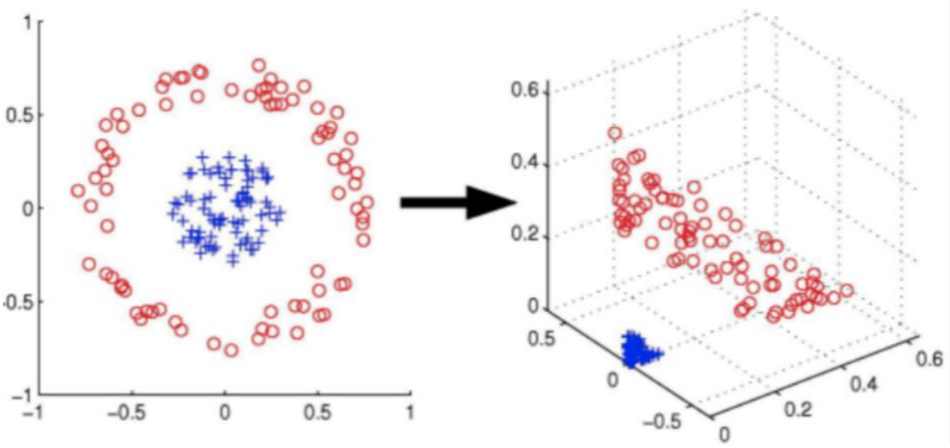


Kuva 4: Neuroverkko (Bre, Gimenez & Fachinotti 2018)

Kuva 4 on esimerkki yhdestä neuroverkosta, jossa on kolme piilotettua kerrosta (hidden layer). Yllä olevassa neuroverkossa on myös n syötettä (input) ja n tuotosta (output). Jokainen kerros koostuu alkioista (node, eli ympyrät).

Käytännössä tämän verkon opettaminen toimii niin, että kukin piilotetuista alkioista ja lopputuotteista saa ensin oman satunnaisen kertoimensa. Syötteenä saadut luvut kulkevat viivoja pitkin ja niitä muokataan kunkin alkion kertoimella. Lopulta muokkausten jälkeen verkon ennuste tulee esiin tuotoskerroksella. Jos kyseessä on vasta opetusvaihe, saatua ennustetta verrataan oikeaan tulokseen ja kertoimia muokataan niin, että ennusteen ja tuotoksen ero pienenee. Tätä prosessia toistetaan useita kertoja (esimerkiksi 100 000) useilla eri syöteillä ja lopputuloksilla, jonka jälkeen mallin alkiot saavat optimaaliset kertoimet ja sen pitäisi osata tehdä ennusteita datasta, johon se ei ole aikaisemmin törmännyt. (Hansen & Salamon 1990, Lawrence 1997).

Toinen usein osakemarkkinoiden ennustamisessa käytetty algoritmi on support vector machine (SVM). SVM on suosittu algoritmi, kun pyritään ratkaisemaan luokittelu- ja regressio-ongelmia. SVM etu on se, että sen avulla pystytään ratkaisemaan muitakin kuin lineaarisia yhteyksiä, jolloin saadut optimit ovat myös globaaleja optimeja, eivätkä pelkästään paikallisia optimeja. Yksinkertaisuudessaan luokitteluongelmalla kuvattuna SVM painottaa datapisteitä niin, että ne pystytään jakamaan omiin luokkiinsa. SVM tekee tätä usealla eri tasolla (dimensions), jolloin lineaarisen jaon epäonnistuessa se nostaa datan seuraavaan potenssiin kernel-funktion avulla ja käyttää sitä niin kuin se olisi ylemmässä potenssissa. Ylemmän potenssin data on eri näköistä, jolloin SVM saattaa löytää yhteyden. SVM jatkaa dataa ylempään potenssiin nostamista, kunnes se löytää järkevän jaon. Kuvassa 5 on esimerkki, kuinka kaksiulotteisen datan nostetaan kolmiulotteiseksi käyttämällä kernelfunktiota. Noston jälkeen data on kolmiulotteisessa koordinaatistossa ja jako onnistuu yhdellä tasolla (hyperplane), koska punaiset ja siniset pisteet ovat selkeästi eri ryhmissä. (Bedell 2018, Tay & Cao 2001)



Kuva 5: Support vector machine (Bedell 2018, Medium, s.1)

Naiivi bayesilainen algoritmi (Naive Bayes/Bayesian) on todennäköisyyksiin perustuva algoritmi. Naiivi tulee nimeen siitä, että se käsittelee kutakin yksittäistä ominaisuutta riippumattomana muista ominaisuuksista. Naiivi Bayesilainen perustuu Thomas Bayesin kehittämään Bayesin teoreemaan, jonka mukaan A:n todennäköisyys ehdolla B on yhtä suuri kuin B:n todennäkösyys ehdolla A kertaa A:n todennäköisyys jaettuna B:n todennäköisyydellä. Bayesin teoreema esitetään yhtälössä 4. (Zhang 2004).

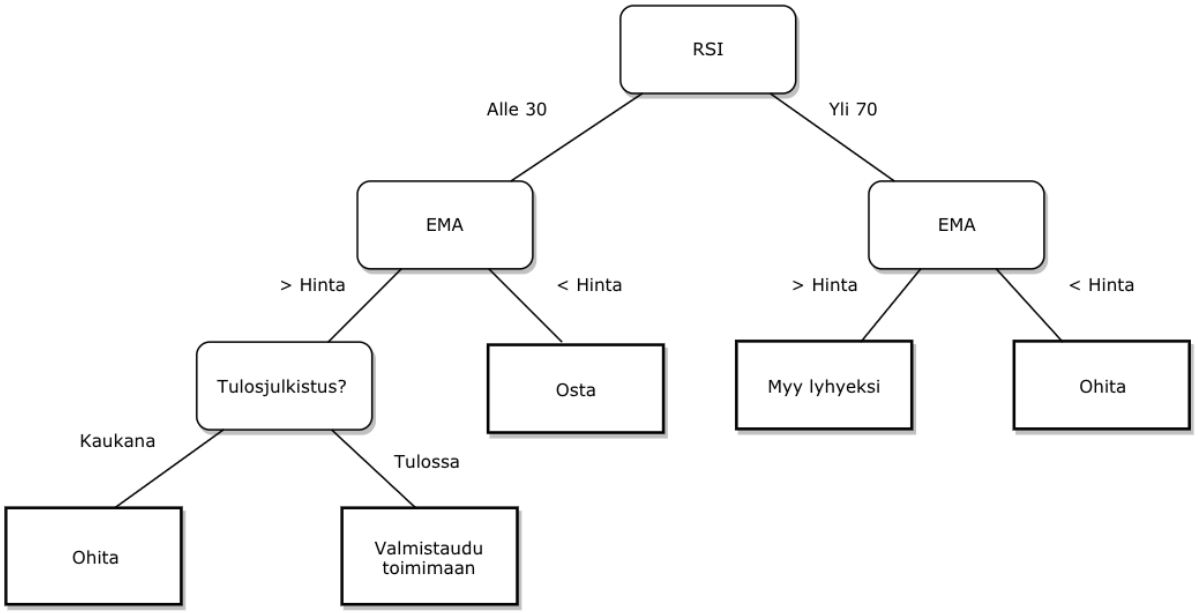
$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)} \quad (4)$$

P(A) = todennäköisyys tapahtumalle A

P(A|B) = todennäköisyys A ehdolla B

Naiivi bayesilainen algoritmi tekeekin ennustuksia juuri bayesin teoreemaan perustuen, hankkien todennäköisyydet oppimisdatasta. Sen jälkeen oppimisdatasta saadut todennäköisyydet voidaan sijoittaa Bayesin teoreemaan, jolloin saadaan arvio tuloksesta. Lopullinen kaava ei kuitenkaan ole yhtä yksinkertainen, koska siinä on useita eri ominaisuuksien todennäköisyyksiä. (Kumar 2017)

Random Forest, tai suomeksi satunnaismetsä on päätöksentekopuihin (decision trees) perustuva koneoppimisen malli. Kuvassa 6 on esimerkki päätöksentekopuusta. Siinä datasta valitaan satunnaisia arvoja, joiden perusteella luodaan useita päätöksentekopuita, joista metsä lopulta koostuu. Sen jälkeen metsän puita testataan siihen dataan, jota ei käytetty puiden tekemiseen, jolloin selviää luodun metsän tarkkuus. Oppiminen mallissa tapahtuu niin, että metsän luomiseen annetaan vakioarvoja, jotka kuvaavat puiden luomiseen valittujen näytteiden määrää ja sitä, montako muuttujaa valitaan kerralla puun jokaisen kerroksen luomisen. Kun näitä vakioita käydään läpi useita kokeillen aikaan saatua metsää datan käyttämättömiin arvoihin, saadaan selville, minkä metsän tarkkuus on paras ja se valitaan ennustemalliksi (Gupta 2015, Liaw & Wiener 2002).



Kuva 6: Esimerkki päätöksentekopuusta koneoppimisessa osakemarkkinoilla.
RSI = relative strenght index, suhteellisen voimakkuuden indeksi
EMA = eksponentiaalinen liukuva keskiarvo

Li et al. (2016) käyttävät tutkimuksessaan Extreme Learning Machine (ELM) nimistä neuroverkkoa. Se on heidän mukaansa paljon nopeampi alustaa verrattuna tavallisiin neuroverkkoihin, joissa oppiminen vaatii useita edestakaisin ajoja oikeiden painotusten löytämiseksi. ELM:ssä sen sijaan asettaa vain satunnaiset painot kullekin alkiolle, eikä niitä muuteta. Algoritmi löytää parhaan ratkaisun lisäämällä alkioita piilotetulle kerrokselle, jolloin virheen suuruus pienenee.

4. OSAKEMARKKINAT JA KONEOPPIMINEN

Useat sijoitusrahastot käyttävät sijoitusstrategianaan koneoppimista, matematiikkaa tai tilastollisia menetelmiä. Ehkä tunnetuin näistä rahastoista on Renaissance Technologies ja heidän Medallion Fund, jonka matemaatikko Jim Simmons perusti 1982. Vuodesta 2001 vuoteen 2013 rahaston huonoin vuosi oli +21% ja esimerkiksi vuonna 2008, kun S&P 500 putosi 38.5%, Renaissance Technologiesin Medallion -rahasto tuotti +98.2%. (Rubin & Collins 2015)

Monet ovat pyrkineet kopioimaan tätä menestysreseptiä ja koneoppimisen käyttö osakemarkkinoilla on suosittua useasta muustakin syystä. Kirjallisuus koneoppimisesta perustuukin monessa tapauksessa osakemarkkinadataan, sillä niistä on saatavilla paljon numeerista dataa ja sitä on luonteva käyttää koneoppimisen testaamiseen. Markkinoilla ei myöskään ole liian yksinkertaista ratkaisua, joten ongelma ei varmasti ole kenellekään liian helppo. (Martin 2007)

Vaikka osakemarkkinadataa hyödyntäviä koneoppimisalgoritmeja tehdään paljon, parhaiten onnistuneita ennustusalgoritmeja tuskin edes julkaistaan missään, vaan niiden kehittäjät haluavat käyttää niitä itse. Julkaisun jälkeen kaikki pääsevät algoritmeihin käsiksi ja pystyvät käymään kauppaa samalla strategialla, pienentäen tarjolla olevia voittoja, sillä niiden tuottama tieto on nyt osa hintoihin sisällytettyä tietoa ja tehokkaat markkinat poistavat ylituotot. (Malkiel & Fama 1970)

4.1 Markkinoiden ennustamiseen käytettyjen algoritmien vertailu

Tay ja Cao (2001) ovat ensimmäisiä, ketkä testasivat support vector machine -algoritmia (SVM) osakemarkkinoiden aikasarjojen ennustamiseen. Heidän pyrkimyksensä oli verrata SVM:n ennustuskykyä jo aikaisemmin ennustamisessa käytettyihin neuroverkkoihin. Osakkeiden tai indeksien ennustamisen sijaan, he pyrkivät ennustamaan futuurien hinnoittelu. Futuurit ovat instrumentti, jonka avulla voidaan suojautua riskejä, kuten esimerkiksi raaka-aineiden hinnan nousua, vastaan. Toiminnallisesti kuitenkin ne ovat verrattavissa osakkeisiin, ja hinta määräytyy kysynnän ja tarjonnan perusteella. He vertaavat neuroverkoa ja SVM:a

usealla eri tilastollisella mittarilla ja näillä kaikilla SVM oli parempi algoritmi. Tutkimuksessa esitetään neljä perustelua SVM:n paremmuudelle:

1. SVM minimoi rakenteellisen riskin, eli se minimoi yleistämisessä tulevan riskin ylärajan, kun taas neuroverkko minimoi opetusvaiheen ylärajan.
2. SVM:ssä on vain kolme parametria, joita voidaan muokata, kun neuroverkoissa parametreja ovat kerrokset (layers), piilotetut alkiot (hidden nodes), oppimistahti, ajokerrat ja painojen asettaminen. Näiden optimaalinen valinta neuroverkkoon on hankalaa.
3. Neuroverkko pyrkii pääsemään ratkaisuun kulmakertoimien avulla, jolloin se saattaa löytää vain paikallisia minimeitä globaalin minimin sijaan. SVM:n ratkaisut ovat globaaleja
4. Neuroverkon opettaminen niin, ettei se sovi dataan liian hyvin (over fitting) on hankalaa ja sen välttämiseksi opettaminen on osattava lopettaa ajoissa. Tämä vaatii paljon kokemusta tai testaamiseen sopivan datan.

Patel, et al. (2015a) vertailevat tutkimuksessaan neljää eri algoritmia käyttämällä niitä kahteen osakkeeseen ja kahteen Intian indeksiin. Algoritmeiksi he ovat valinneet neuroverkon, SVM:n, satunnaismetsään ja naiivin bayesilainen, joita he käyttävät aikaisempaan Kara, Boyacioglu ja Baykanin (2011) tutkimukseen perustuvien indikaattoreiden kanssa. He vertailevat sekä algoritmeja, että eroja jatkuvien ja diskreettien arvojen käyttöä. Taulukossa 1 on heidän saamien tuloksiensa keskiarvot eri osakkeiden ja indeksien kesken eri algoritmeille. Suurempi ennustetarkkuus tarkoittaa parempaa tulosta ja yli 55% ennustetarkkuus onkin osakemarkkinoiden tapauksessa jo hyvä tulos. Tuloksien mukaan jokainen algoritmityyppi onnistuu ennustamisessa hyvin ja saavuttaa yli 70% ennustetarkkuuden. Ennustetarkkuudet kuitenkin riippuvat paljon syötteiden esikäsittelystä. Parhaan ennusteen tuotti jatkuvilla muuttujilla satunnaismetsä ja diskreeteillä muuttujilla naiivi bayesilainen.

Taulukko 1: Ennustetarkkuuksien keskiarvot muuttujien tyypin perusteella

	Jatkuva	Diskreetti
Neuroverkko	0.7494	0.8669
SVM	0.7871	0.8933
Satunnaismetsä	**0.8359**	0.8998
Naiivi bayesilainen	0.7331	**0.9019**

Toisessa tutkimuksessaan Patel et al. (2015b) pyrkivät yhdistelemään algoritmeja ja luomaan niistä fuusiomalleja. Näiden fuusiomallien tarkoituksena on heidän mukaansa mahdollistaa datan käyttö pidemmältä aikaväliltä. Esimerkiksi, kun pyritään ennustamaan päivän t hintaa käyttäen dataa päivältä *(t - n),* aikaisemmin käytetyt yksivaiheiset mallit eivät pysty kunnolla käsittelemään näitä vanhempia arvoja n:n kasvaessa suuremmaksi, kun taas fuusiomallit eivät törmää tähän ongelmaan. Käytännössä tutkimuksessa käytetyt fuusiomallit toimivat niin, että ne pyrkivät ennustamaan ensin annetun datan tulevaisuuteen, jonka perusteella saadaan tulevaisuuden arvot. Tulevaisuuteen ennustetuilla arvoilla ennustetaan sen jälkeen osakekurssia. Esimerkiksi t:n kuvatessa aikaa, jolloin indikaattorit saavat arvot *I(t)* ja kohdeindeksi saa arvon *O(t).* Yksikerroksinen algoritmi pystyisi suhteellisen hyvin kuvaamaan funktiota *f*, joka ennustaa lyhyen ajan päähän tulevaisuuteen *f(I(t)) = O(t+1)*. Kun pyritään ennustamaan kauemmas tulevaisuuteen, n = 15 päivää, koneoppimisella saatu funktio *h* pyrkii toteuttamaan *h(I(t)) = O(t+15).* Tämä toimii huonommin kuin lyhyemmän aikavälin ennustaminen esimerkiksi suuremman epävarmuuden takia. Fuusioalgoritmi pyrkikin ennustamaan ensin syötedatan tulevaisuuden arvoja ja pyrkii niiden avulla muodostamaan “lyhyen ajan” hintaennusteen. Esimerkiksi fuusioalgoritmin ensimmäinen kerroksen muodostama funktio *f1* muuttaa ensin datan tulevaisuuteen *f1(I(t)) = I(t+n).* Tästä toisen kerroksen funktio pystyy ennustamaan paremmin, koska ennuste perustuu ”uudempaan” dataan. Saadaankin siis *f2(f1(I(t))) = O(t+n)*. Tämä ennuste tuottaa tutkimuksen mukaan parempia tuloksia kuin *h(I(t)) = O(t + n).*

Algoritmeina he käyttävät neuroverkkoa, SVM ja satunnaismetsää (Random forest, RF). Kussakin fuusiomallissa he esikäsittelevät dataa käyttäen SVM:ä, jolloin testatut fuusioalgoritmit ovat SVM - NN, SVM - SVM ja SVM - RF. Tässä ensimmäinen SVM vastaa esimerkissä käytettyä funktiota *f1* ja jälkimmäinen *f2*. Alla olevassa taulukossa on virhetermit erilaisilla tilastollisilla mittareille algoritmeille ja niiden fuusioille testattuna CNX Nifty -indeksiin, joka kuvaa Intian pörssin liikkeitä. Pienempi arvo viittaa kullakin mittarilla parempaan tarkkuuteen.

Taulukko 2: Algoritmien ja fuusioalgoritmien virhetermit keskimäärin eri aikaväleillä

	MAPE	MAE	RMSE	MSE
NN	3,01	160	3,79	43875
SVM - NN	**2,66**	**140**	3,41	**34207**
SVM	2,71	142,5	3,43	37530
SVM - SVM	**2,66**	**140**	**3,39**	34975
RF	3,02	160	3,87	46992
SVM -RF	2,72	143,6	3,44	36166

Selitteet liitessä 1

Taulukosta 2 nähdään, että fuusioalgoritmit saavat kaikissa tapauksissa pienemmät virhearvot verrattuna pelkästään yhden koneoppimisalgoritmin käyttöön, eli ne suoriutuvat ennustamisessa paremmin. SVM -malli hyötyy fuusiosta vähiten saaden siitä vain pienen parannuksen verrattuna SVM:n käyttöön yksittäin. Parhaiten kaikkien testien kesken Patel et al. (2015b) mukaan suoriutuu SVM - NN malli.

Li et al. (2016) pyrkivät ensisijaisesti testaamaan algoritmien nopeuttamista ja nopeuden vaikutusta suorituskykyyn Tutkimuksessaan he vertaavat extreme learning machine (ELM), SVM ja neuroverkon tuottamia ennustearvoja, sekä niiden toiminnan tehokkuutta. Taulukossa

3 kuvataan algoritmien eroja. Suurempi tarkkuus ja pienemmät ennuste- sekä opetusajat ovat parempia.

Taulukko 3: algoritmien tarkkuus, ennustenopeus ja opetusnopeus sekunneissa

Algoritmi	Tarkkuus	Ennusteaika (s)	Opetusnaika(s)
Neuroverkko	0.537	0.190	3383
SVM	**0.644**	0.266	164
ELM	0.607	**0.155**	**14**

Tutkimuksen mukaan SVM voittaa ennustetarkkuudessa ELM:n kaikilla annetuilla parametrien arvoilla ja on myös tilastollisesti merkittävästi parempi. Aikojen perusteella mitattuna ELM pärjäsi parhaiten. Tutkimuksessa mainitaan kuitenkin, ettei algoritmin opetusnaika ole osakekaupan kannalta kovinkaan merkittävää, sillä opettaminen tehdään yleensä ennen kaupankäynnin aloittamista, eikä sen aikana. (Li et al. 2016). Ennusteaika sen sijaan on tärkeää esimerkiksi korkean frekvenssin kaupankäynnissä ja arbitraasien hyödyntämisessä (Martin 2007).

Enke ja Thawornwong (2005) käyttivät neuroverkkoa Standard & Poor's 500 (S&P 500) ennustamiseen. He testasivat erilaisten algoritmien välisiä eroja, sekä eroja erilaisten tavoitteiden kanssa. He tutkivat miten ennustetarkkuus muuttuu, kun ennustetaan pelkän hinnan (tasomallit, ennustavat hinnan tasoa esimerkiksi euroissa) sijaan suuntaa markkinan suuntaa (suuntamallit, ennustavat hinnan muutoksen suuntaa ylös tai alas). Heillä oli algoritmeina käytössään erilaisia neuroverkkojen versioita ja vertailukohtana regressiomalli. Heidän tuloksensa löytyvät liitteestä 2.

Malleilla oli alhainen korrelaatio markkinatuottojen kanssa, joka heidän mukaansa kertoo, ettei mikään malleista pystyisi tarkasti ennustamaan ylimääräisiä tuottoja. Kuitenkin markkinoiden hinnan muutoksella mitattuna he löysivät tilastollisesti merkittävän ennustetarkkuuden 95% varmuusasteella kaikilla muilla malleilla paitsi regressiolla. Heidän mukaansa tavallinen

regressio pärjäsikin vertailussa selkeästi huonoiten. Tämä viittaa siihen, että historiallisilla indikaattoreilla olisi epälinearinen suhde osaketuottoihin (Enke & Thawornwong 2005)

4.2 Erilaiset syötteet

Datana koneoppimismalleissa voidaan käyttää lähes mitä vain. Ongelmaksi kuitenkin muodostuu nopeasti, miten esimerkiksi teksti saadaan koneelle ymmärrettävään muotoon, josta se pystyy tunnistamaan eroja. Yleisimmät käytetyt indikaattorit voidaan jakaa teknisiin, fundamentaalisiin ja sentimentaalisiin. Tekniset indikaattorit ovat yleisimpiä, sillä ne ovat usein valmiiksi numeromuodossa ja siten helppo käsitellä. Dataa on myös tarjolla laajasti, joten ne ovat myös sen takia suosittuja. Useat työkalut myös piirtävät tekniset indikaattorit suoraan osakekurssigraafiin, jolloin niiden perusteella voidaan käydä kauppaa myös manuaalisesti. (Brock, Lakonishok & LeBaron 1992)

Fundamentaaliset indikaattorit ovat esimerkiksi tilinpäätöksestä saatavia arvoja, kuten liikevaihto ja tulos. Niiden käyttö koneoppimisen kanssa on hankalampaa, sillä ne päivittyvät harvoin, eivät ole verrannollisia aikavälien kesken, poikkeavat merkittävästi eri toimialojen ja erityisesti eri yritysten välillä ja eivät ole usein yhtä helposti saatavilla automaattisesti, kuin hintadata. (Summers 1986)

Sentimenttidata kuvaa ”keskiarvomielipidettä” jostain aiheesta ja se voidaan kerätä esimerkiksi Twiiteistä tai uutisista. Sentimenttidatan käyttö ei ole helppoa ja jo sen prosessoimisessa joudutaan usein käyttämään koneoppimista. Prosessointi on tärkeää, että kone pystyy ymmärtämään tekstin sisältöä ja siten selvittämään tekstistä populaation ajatuksia. (Engelberg & Gao 2011).

Shen, Jiang ja Zhang (2012) pyrkivät tutkimuksessaan käyttämään hyödyksi globalisaation lisääntymisestä johtuvaa korrelaatiota eri markkina-alueiden välillä. Lisääntyneen globalisaation takia yhden pörssin liikkeet vaikuttavaa myös muhin. Pörssit eivät kuitenkaan liiku täysin käsi kädessä ja ne eivät myöskään ole auki täysin samoihin aikoihin, vaan esimerkiksi Saksan pörssi on ollut auki jo useita tunteja Yhdysvaltain pörssien auetessa. Shen, Jiang ja Zhang käyttävät SVM -algoritmia selvittämään, voiko pörssin liikkeillä ennustaa toisen

pörssin tulevia liikkeitä, sillä esimerkiksi Saksan DAX -indeksillä oli 70.8% korrelaatio NASDAQ -indeksiin. He käyttivät lopullisessa mallissaan seuraavia syötteitä: indekseistä Hang Seng, Nikkei, FTSE ja ASX, valuutoista euro, Australian dollari, Japanin yen, sekä raaka-aineista kulta, platina, öljy ja hopea. Näillä he saivat ennustetarkkuuksiksi 74.4% NASDAQ:iin, 77.6% Dow Jones Industrial Average:n ja 76% S&P 500:n. He kehittivät koneoppimismallin pohjalta myös kaupankäyntistrategian. Strategiassa kaupankäyntiriskin minimoimiseksi he kävivät kauppaa vain päivinä, joina ennustemalli oli suhteellisen varma markkinan suunnasta. He kokeilivat malliin perustuvaa strategiaa viidellä 50 päivän aikajaksolla ja saivat $10 000 alkusijoituksella tuottoa keskimäärin $814, kun markkinat tuottivat vastaavilla aikaväleillä keskimäärin $132. Heidän tuottonsa oli noin 8% 50 päivän ajalta, joka vastaa 30% vuosituottoa, kun markkinat ovat vuosina 1928-2018 tuottaneet keskimäärin 9.49% ja vuosina 2009-2018 12.98% (Damidaran, 2019). Malli pärjäsi markkinoita paremmin erityisesti huonoina aikoina, vaikka se ei mahdollistanut lyhyeksi myyntiä, jonka implementointi heidän mukaansa maksimoisi tuoton. On kuitenkin huomioitava, ettei kaupankäyntikuluja tai veroja ollut otettu huomioon, eli tilanne ei ole täysin identtinen reaalimaailmassa. (Shen et al. 2012)

Patel et al. (2015a) käyttivät tutkimuksessaan useita eri teknisiä indikaattoreita, jotka löytyvät liitteestä 3. He esikäsittelivät myös dataansa ja korostavat erityisesti indikaattorien normalisoinnin merkitystä, sillä ilman sitä suuret vaihteluvälit arvojen välillä aiheuttavat ongelmia. Tutkimuksessaan he käyttävät kahta eri metodia datan normalisointiin. Ensimmäinen on yksinkertaisesti se, että kukin arvo normalisoidaan välille [-1, 1], niin, että kunkin indikaattorin aineiston pienin arvo saa normalisoinnin jälkeen arvon -1 ja suurin 1. Normalisointi tapahtuu käyttäen yhtälöä 5. Tätä tutkimuksessa kutsutaan jatkuvien arvojen käytöksi.

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (5)$$

Toinen tutkimuksessa käytetty normalisointitapa on käyttää indikaattoreiden omaisuuksia ja antaa niille sen perusteella diskreetit arvot. Jos indikaattori ennustaa laskevaa trendiä, se saa arvon -1 ja ennustaessaan nousevaa trendiä se saa arvon 1. Esimerkiksi liukuva keskiarvo

(simple moving average, SMA) ennustaa nousevaa trendiä, kun osakkeen hinta on suurempi kuin se. Tämä johtuu momentum-ilmiöstä, eli siitä, että osakkeen hinnan muuttuessa on todennäköisempää, että se jatkaa samaan suuntaan trendin mukaisesti. (Patel et al. 2015a). Muiden tutkimuksessa käytettyjen indikaattoreiden muokkaaminen esitetään liitteessä 4.

Tutkimuksessa havaitaan, että indikaattorien sisäisten ominaisuuksien hyödyntäminen datan normalisoinnissa tuottaa merkittävästi parempia tuloksia, kuin tavallinen normalisoinnin yhtälö. Eri muuttujatyyppien diskreeteillä arvoilla ennusteet olivat luotettavampia koko näytteen laajuudelta ja eri algoritmeilla. Tämä johtuu tutkimuksen mukaan siitä, että he pyrkivät ennustamaan tarkan arvon sijaan markkinan suuntaa, kuten aiemmin mainituissa suuntamalleissa. Diskreetit muuttujat kuvaavat kunkin indikaattorin ennustusta hinnan muutoksen suunnasta, kun taas jatkuvia arvoja käytettäessä tämä indikaattorin sisäinen ominaisuus häviää. Tutkimuksessa arviointiinkin, että suunnan sijaan markkinoiden tarkkojen hintojen ennustamisessa (tasomalli) jatkuvat arvot tuottaisivat parempia tuloksia. Lopputuloksena tutkimukselle mainittiin datan esikäsittelyn tärkeyden koneoppimisen käytössä, viitaten siinä indikaattoreiden normalisointiin. Siinä myös ehdotettiin mahdollisia tulevaisuudessa käytettäviä indikaattoreita: vaihtokurssit, inflaatio, lainsäädäntömuutokset ja korkotaso. Lopuksi he vielä lisäsivät, että tutkimus keskittyi erityisen lyhyelle aikavälille, ja aikavälin pidentäminen saattaisi muuttaa lopputuloksia. Pidemmän aikavälin malleihin he korostivat erityisesti fundamentaalisten tekijöiden merkitystä, sillä osakkeen pidemmän aikavälin muutostrendi perustuu enemmän yrityksen toiminnan kehitykseen. (Patel et al. 2015a)

Li et al. (2016) käyttivät tutkimuksessaan teknisten indikaattorien lisäksi sentimenttianalyysiin perustuvia indikaattoreita, jotka heidän tapauksessaan perustuivat uutisiin. Sentimenttianalyysi vaatii datan laajaa esikäsittelyä ja he prosessoivatkin uutisia seuraavasti: ensin he poistivat välisanat niin, että jäljelle jäi vain sisältöä paremmin kuvaavat adjektiivit, verbit ja substantiivit. Näistä jäljelle jääneistä sanoista he valitsivat merkittävimmän 10%, jotka ajetaan kielentunnistusalgoritmin läpi. Tämä algoritmi yhdistää kunkin sanan kertoimeen, joka ilmoittaa, onko sana negatiivinen, neutraali vai positiivinen (-1, 0, 1). Nämä kertoimet vektorisoidaan, jonka jälkeen se kuvaa yhden uutisen sävyä. Näitä normalisoituja arvoja on helppo käyttää koneoppimisen algoritmien kanssa. He käyttivät malleissaan

sentimenttianalyysin lisäksi seuraavia teknisiä indikaattoreita, joita ei mainita muissa tutkimuksissa: Bias, psychological line (liite 5), 5:n, 10:n, 15:n, 20:n, 25:n ja 30 päivän hinnan muutos, (yhtälö 6). Indikaattoreita käytettiin kolmen eri koneoppimisalgoritmin kanssa ja tulokset ovat esillä taulukossa 4.

$$\frac{P(t)-P(t-n)}{P(t-n)} * 100\% \quad (6)$$

P(t) = Hinta päivänä t

n = monenko päivän muutos

Taulukko 4: Ennustemallien tuotot.

Algoritmi	Sharpen luku
Neuroverkko	0.4
SVM	1.7
ELM	1.8

Sharpen luku (Sharpe 1994)

$$\frac{R_p-R_f}{\sigma_p} \quad (7)$$

Rp = Portfolion tuotto

Rf = Riskitön korko

σp = Portfolion keskihajonta

Sharpen luku (yhtälö 7) kuvaa kuinka paljon ylimääräistä tuottoa portfolio saa vastineeksi riskistä, eli suurempi luku on parempi. Tutkimuksessa neuroverkolla saatu strategia sai Sharpen luvuksi 0.4, SVM 1.7 ja ELM 1.8, eli sen avulla mitattuna ELM oli algoritmeista parhaiten tuottava. Tutkimuksessa ei kuitenkaan otettu kantaa esimerkiksi kaupankäyntikuluihin, jotka

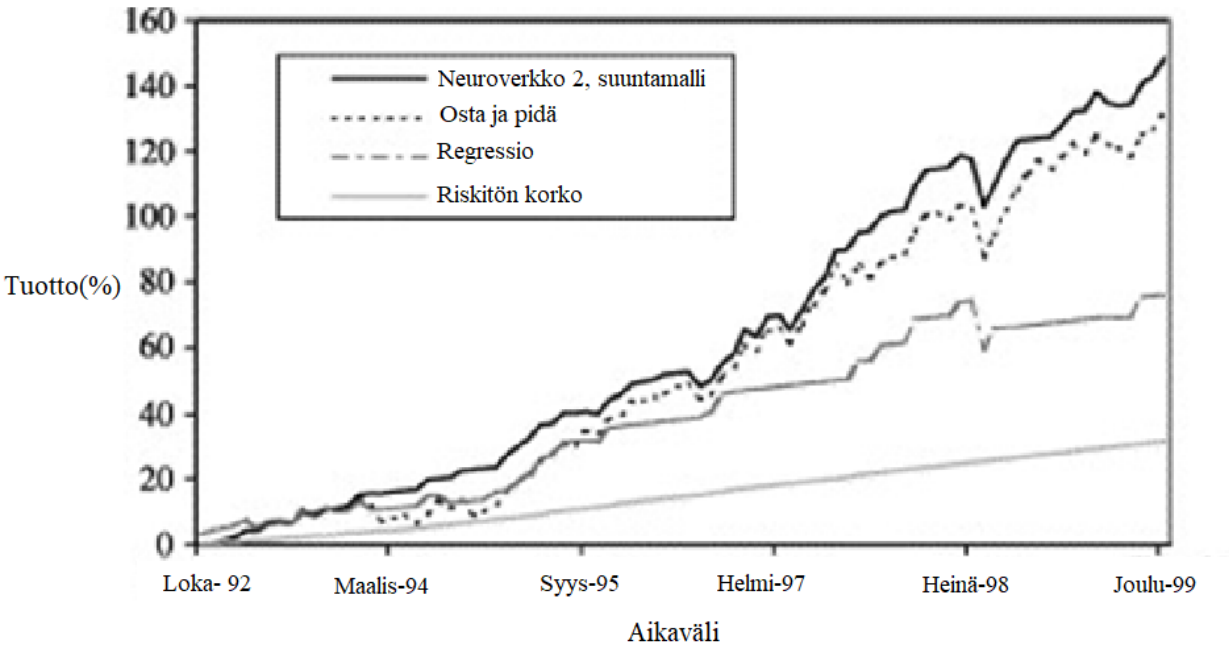
pienentävät saatuja tuottoja (Carhart, 1997), eikä muutenkaan huomioitu juurikaan sijoittamisen käytännön aspektia. Näihin olisi tutkimuksessa ollut erityisen tärkeää kiinnittää huomiota, sillä tutkimuksen perustana oli pyrkiä selvittämään algoritmin nopeutta ja tehokkuutta nopean kaupankäynnin (High Frequency Trading, HFT) tarkoitukseen. Siinä kauppoja tehdään jopa millisekuntitasolla, jolloin algoritmien tehokkuudella ja läpimenoajalla on suuri merkitys ja kaupankäntikustannukset ovat suuria. (Martin 2007)

Enke ja Thawornwong (2005) pyrkivät ennustamaan tunnettua Standard & Poor's 500 (S&P 500) -indeksiä. S&P 500 kuvaa viidensadan suuren yhdysvaltalaisen pörssiyrityksen kurssikehitystä ja toimii usein esimerkkinä osakemarkkinoiden keskimääräisestä tuotosta. S&P 500 on tuottanut vuodesta 1926 lähtien keskimäärin 9.8% vuodessa, vaikka se onkin mennyt alaspäin noin 30% näistä vuosista (SPindices). He käyttivät yhteensä 31 muuttujaa aikavälillä maaliskuu 1976 - Joulukuu 1999. Liitteessä 6 löytyy kuvaus heidän käyttämistänsä syötteistä. He kehittivät myös näihin syötteisiin ja koneoppimisalgoritmiin perustuvan kaupankäyntistrategian. He sijoittivat kaiken indeksiin, kun seuraavan kuukauden aikana estimoitu osaketuotto oli suurempi kuin nolla. Jos tuotto oli alle nolla, he sijoittivat riskittömään korkoon. Taulukossa 5 ja kuvassa 7 kuvataan heidän tuottojansa.

Taulukko 5: Simuloidun kaupankäynnin tulokset

		Kuukausittainen tuotto	Keskihajonta	Sharpe
Tasomallit	Neuroverkko 1	1.55	3.56	1.2
	Neuroverkko 2	1.58	3.61	1.2
	Neuroverkko 3	1.47	3.64	1.12
	Neuroverkko 4	1.62	3.55	1.25
Suuntamallit	Neuroverkko 1	1.51	2.99	1.32
	Neuroverkko 2	**1.72**	3.16	**1.43**
	Neuroverkko 5	1.26	2.67	1.2
Vertailukohteet	Regressio	0.89	**2.49**	0.89
	Osta ja pidä	1.54	3.68	1.16
	Riskintön korko	0.37	-	-

Tuotot ovat keskimääräisiä kuukausituottoja 86 kuukauden ajalta. Eri strategioiden tuottoja esitetään kuvassa 7. Tuloksista huomataan, että neuroverkkojen tuotot ja Sharpet olivat lähellä osta ja pidä -strategiaa, kun taas regressio hävisi molemmille merkittävästi. Sharpen luvulla mitattuna myös kaikki muut neuroverkot paitsi neuroverkko 3 voittivat osta ja pidä –strategian ja niiden mukaan myös suuntamallit tuottivat tasomalleja paremmin. Tutkimuksessa arvioitiin, että neuroverkot eivät menestyneet tuotoilla mitattuna merkittävästi paremmin kuin osta ja pidä –strategiaa, koska S&P 500 nousi aikavälillä huomattavasti. Siinä arvioitiinkin neuroverkkojen pärjäävän markkinoita paremmin erityisesti laskusuhdanteessa. (Enke & Thawornwong 2005).



Kuva 7: Tuotot aikavälillä (Enke & Thawornwong 2005)

Johtopäätöksinä tutkimuksessa kerrotaan, että muuttujien merkittävyys ja yhteydet vaihtelevat aikavälien kesken, joten vanhemmalla datalla on oltava pienempi painoarvo. Tutkimuksessa suositellaankin käyttämään koneoppimista jo datan relevanssin tutkimiseen, jolloin sitä voi karsia etukäteen. He myös huomauttavat, kuinka pienimmät tilastolliset virhearvot algoritmien testauksessa eivät taanneet hyvää suoriutumista kaupankäyntistrategiana. He eivät neuroverkkojen onnistumisesta huolimatta lähde tyrmäämään tehokkaiden markkinoiden hypoteesia, sillä osta ja pidä -strategia pärjäsi hyvin, eikä hävinnyt neuroverkoille merkittävästi. (Enke & Thawornwong 2005)

5. JOHTOPÄÄTÖKSET

Tutkimuksessa oli kaksi ydinkysymystä:

- Minkälaisia tuloksia koneoppimisella on saatu markkinoiden ennustamisessa?
- Mitkä ovat yleisimmin koneoppimisen kanssa käytetyt syötteet?

Tulokset markkinoiden ennustamisessa koneoppimisella olivat positiivisia ja useat erilaiset algoritmien ja syötteiden yhdistelmät saavuttivat yli 50% ennustetarkkuuden, eli ovat onnistuneet ennustamisessa. Kirjallisuuden mukaan parhaat tulokset saatiin support vector machine (SVM) pohjaisilla algoritmeilla mutta myös neuroverkolla, satunnaismetsällä, naiivilla bayesilaisella ja extreme learning machine -algoritmeilla on päästy lähes yhtä hyviin tuloksiin. Kullakin algoritmilla on omat puolensa ja ne soveltuvat eri tilanteisiin. Neuroverkon alustaminen on hidasta, mutta sillä ennustaminen on nopeampaa, kun taas SVM:n kanssa tilanne on päinvastainen. Kunkin algoritmien tuloksia pystytään parantamaan yhdistelemällä niitä fuusioalgoritmeiksi ja esikäsittelemällä dataa.

Syötteinä kirjallisuudessa käytetään usein hintaan perustuvia indikaattoreita. Tämä johtuu siitä, että historiallinen hintadata on datasta helpoiten saatavilla, jolloin näiden algoritmien testaaminen ja opettaminen on helpompaa. Syötteiden paremmuuden vertailu on aiheetonta, sillä niitä kaikkia voidaan käyttää yhdessä tai erikseen. Yksittäisten muuttujien vaikutusta lopputulokseen onkin hankala vertailla, sillä yhteydet ovat usein hämäriä. Taulukossa 6 esitetään tutkimuksissa käytetyt ja ehdotetut syötteet.

Taulukko 6: Indikaattorit kirjallisuudesta, selitteet liite 3 ja 5

Tekniset indikaattorit	**Fundamentaaliset tekijät**	**Muut tekijät**
SMA	Osinko	Valuuttakurssit
WMA	Osinkoprosentti	Muut pörssit
EMA		Raaka-aineet
Momentum		Jalometallit
Stochastic K%		Valtion lainojen korot
Stochastic D%		Yrityslainojen korot

RSI MACD Larry William's R% CCI A/D Oscillator BIAS PSY		Kuluttajahintaindeksi Tuottajahintaindeksi Teollisuuden tuotantoindeksi Valtion käteisvarannot Uutiset
Ehdotettu mutta ei kokeiltu: • Inflaatio • Lainsäädäntö • Muut fundamentaaliset tekijät • Volyymi • Tuotannon kasvunopeudet		

Kirjallisuudessa käytetyistä syötteistä huomataan, että erityisesti fundamentaalisia tekijöitä on käytetty todella vähän. Tämä saattaa johtua siitä, että ne ovat suositumpia perinteisessä osakeanalyysissä, jolloin tehokkaiden markkinoiden hypoteesin mukaan niitä hyödyntämällä ylituottoa ei ole saatavilla.

Koneoppimismallissa on syötteiden esikäsittelyssä hyvä huomioida myös niiden sisäiset ominaisuudet. Esimerkiksi suhteellisen vahvuuden indeksi (Relative Strenght Index, RSI) kertoo jo itsessään, onko osake "alimyyty" tai "ylimyyty". Osake on ylimyyty RSI:n mukaan, kun RSI saa arvon yli 70. Ylimyyty osake tarkoittaakin, että sen seuraava trendi olisi alaspäin. Näitä ominaisuuksia on mahdollista hyödyntää syötteiden esikäsittelyssä niin, että syötteet voidaan normalisoida niiden mukaisesti. Diskreettejä syötteitä kannattaa kuitenkin käyttää vain tilanteissa, joissa pyritään ennustamaan diskreettejä asioita (esimerkiksi ylös tai alas) ja tarkan arvon ennustamisessa kannattaa käyttää jatkuvia normalisoituja arvoja.

Vaikka markkinoiden ennustaminen selkeästi onnistuukin koneoppimisella ja tutkimuksissa koneoppiminen tuotti usein paremmin kuin verrokki-indeksi, ei tämä ole vielä riittävä tulos. Tutkimuksissa ei otettu huomioon markkinoiden tuottoon vaikuttavia CAPM:n tai three-factor -mallin tekijöitä, kuten riskiä, kokoa tai kirja-arvoa, jolloin koneoppimismallien suurempi tuotto saattaa selittyä vain niillä. Sekä akateemisissa piireissä, että käytännön sijoittamisessa

kiinnostavampi tekijä onkin CAPM:lla tai three-factor -mallilla mitattu alfa ja vaikka koneoppimismalleilla on saavutettu markkinoita suurempia tuottoja, ei alfaa silti välttämättä syntyisi. Koneoppimisella on kuitenkin todennäköisesti mahdollista saavuttaa alfaa myös three-factor mallilla mitattuna, sillä mallissa ei huomioida koneoppimisen kanssa käytettyjä yleisimpiä syötteitä, vaan se kiinnittää huomiota vain riskiin ja kahteen fundamentaaliseen tekijään. Osakemarkkinoita kuvaavat mallit ovatkin nykyään jo hieman vajavaisia, sillä nykyään käytettävissä olevan datan määrä on valtava verrattuna aikaisempaan.

Tulevissa tutkimuksissa olisikin hyvä ottaa huomion erityisesti rahoitusteorian näkökulma. Koneoppimismallien arvioimisessa esimerkiksi three-factor -mallin alfa olisi hyvä mittari oikean ylituoton selvittämiseksi ja se kertoo paljon enemmän mallin toimivuudesta, kun esimerkiksi ennustetarkkuus. Tilanne kuitenkin riippuu tutkimuksen tavoitteista ja koneoppimista tutkittaessa nämä mittarit eivät välttämättä ole yhtä kiinnostavia. Tutkimuksia erityisesti sentimenttitekijöiden lisäämisen vaikutuksista olisi myös hyvä saada erilaisille lähteille, kuten uutisille tai Twiiteille.

6. LÄHTEET

Julkaisut:

Bre, F., Gimenez, J.M. & Fachinotti, V.D., 2018. Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*, 158, pp.1429-1441.

Brock, W., Lakonishok, J. & LeBaron, B., 1992. Simple technical trading rules and the stochastic properties of stock returns. *The Journal of finance*, 47(5), pp.1731-1764.

Carhart, M.M., 1997. On persistence in mutual fund performance. *The Journal of finance*, 52(1), pp.57-82.

Engelberg, J.O.S.E.P.H. & Gao, P., 2011. In search of attention. *The Journal of Finance*, *66*(5), pp.1461-1499.

Enke, D. & Thawornwong, S., 2005. The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with applications*, *29*(4), pp.927-940.

Fama, E.F., 1995. Random walks in stock market prices. *Financial analysts journal*, 51(1), pp.75-80.

Fama, E.F. & French, K.R., 1992. The cross-section of expected stock returns. *the Journal of Finance*, 47(2), pp.427-465.

Kara, Y., Boyacioglu, M.A. & Baykan, Ö.K., 2011. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. *Expert systems with Applications*, *38*(5), pp.5311-5319.

Kotsiantis, S.B., Kanellopoulos, D. & Pintelas, P.E., 2006. Data preprocessing for supervised leaning. *International Journal of Computer Science*, 1(2), pp.111-117.

Lawrence, R., 1997. Using neural networks to forecast stock market prices. University of Manitoba, 333.

Le, Q.V., 2013, May. Building high-level features using large scale unsupervised learning. In 2013 IEEE *International conference on acoustics, speech and signal processing* (pp. 8595-8598). IEEE.

Li, X., Xie, H., Wang, R., Cai, Y., Cao, J., Wang, F., Min, H. & Deng, X., 2016. Empirical analysis: stock market prediction via extreme learning machine. *Neural Computing and Applications*, *27*(1), pp.67-78.

Liaw, A. & Wiener, M., 2002. Classification and regression by randomForest. *R news*, 2(3), pp.18-22.

Malkiel, B.G. & Fama, E.F., 1970. Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2), pp.383-417.

Merton, R.C., 1973. An intertemporal capital asset pricing model. *Econometrica*, 41(5), pp.867-887.

Michie, D., Spiegelhalter, D.J. & Taylor, C.C., 1994. Machine learning. Neural and Statistical Classification, 13.

Patel, J., Shah, S., Thakkar, P. & Kotecha, K., 2015a. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, *42*(1), pp.259-268.

Patel, J., Shah, S., Thakkar, P. & Kotecha, K., 2015b. Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications*, *42*(4), pp.2162-2172.

Rikkinen, M., 2019. Twiittien hyödyntäminen osakemarkkinoiden ennustamisessa.

Sharpe, W.F., 1994. The sharpe ratio. *Journal of portfolio management*, 21(1), pp.49-58

Shen, S., Jiang, H. & Zhang, T., 2012. Stock market forecasting using machine learning algorithms. Department of Electrical Engineering, Stanford University, Stanford, CA, pp.1-5.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T. and Lillicrap, T., 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv preprint arXiv:1712.01815.

Summers, L.H., 1986. Does the stock market rationally reflect fundamental values?. *The Journal of Finance*, 41(3), pp.591-601.

Sutton, R.S. & Barto, A.G., 1998. Introduction to reinforcement learning (Vol. 2, No. 4). Cambridge: MIT press.

Tay, F.E. & Cao, L., 2001. Application of support vector machines in financial time series forecasting. *omega*, *29*(4), pp.309-317.

Zhang, H., 2004. The optimality of naive Bayes. *AA*, 1(2), p.3.

Kirjat:

Bishop, C.M., 2006. Pattern recognition and machine learning. springer. s. 1-28, 225-227, 291-291, 325-326, 339-243

Duda, R.O., Hart, P.E. & Stork, D.G., 2012. Pattern classification. John Wiley & Sons.
Hansen, L.K. & Salamon, P., 1990. Neural network ensembles. IEEE Transactions on Pattern Analysis & Machine Intelligence, (10), pp.993-1001.

Harrington, P., 2012. Machine learning in action. Manning Publications Co..

Hastie, T., Tibshirani, R. & Friedman, J., 2009. The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.

Knüpfer, S. & Puttonen, V. 2018, Moderni rahoitus, 10th ed., Sanoma Pro Oy.

Verkkolähteet:

Bedell, Z., 2018, Support vector machine, Medium, [Viitattu 1.11.2019]. Saatavissa: https://medium.com/@zachary.bedell/support-vector-machines-explained-73f4ec363f13

Damidaran, A., 2019, Annual returns on stock, T.bonds and T.Bills: 1928 – current, Stern, [Viitattu 6.11.2019]. Saatavissa: http://pages.stern.nyu.edu/~adamodar/New_Home_Page/datafile/histretSP.html

Fate, 2015, Stackexchange. [Viitattu 1.11.2019]. Saatavissa: https://chess.stackexchange.com/questions/8331/is-the-number-of-possible-chess-games-infinite

Gupta, A., 2015, Random forest regression in python, GeeksForGeeks, [Viitattu 13.11.2019]. Saatavissa: https://www.geeksforgeeks.org/random-forest-regression-in-python/

Kasturi, S., 2019, Underfitting and overfitting in machine learning and how to deal with it, Towards Data Science, [Viitattu 10.12.2019]. Saatavissa: https://towardsdatascience.com/underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6fe4a8a49dbf

Kumar, N., 2017, Naive Bayes classifier, GeeksForGeeks, [Viitattu 3.11.2019]. Saatavissa: https://www.geeksforgeeks.org/naive-bayes-classifiers/

Martin, R., 2007, Wall Street's quest to process data at the speed of light, InformationWeek, [Viitattu 4.11.2019]. Saatavissa: https://www.informationweek.com/wall-streets-quest-to-process-data-at-the-speed-of-light/d/d-id/1054287?

Rubin, R. & Collins, M., 2015, How an excluse hedge fund turbocharged its retirement plan, Bloomberg, [Viitattu 13.11.2019]. Saatavissa: https://www.bloomberg.com/news/articles/2015-06-16/how-an-exclusive-hedge-fund-turbocharged-retirement-plan

SPindices, 2019, [Viitattu 27.11.2019]. Saatavissa: https://us.spindices.com/indices/equity/sp-500

7. LIITTEET

Liite 1. Tilastollisia indikaattoreita

Korrelaatio = $\frac{1}{n-1}\Sigma\left(\frac{x-\overline{x}}{s_x}\right)\left(\frac{y-\overline{y}}{s_y}\right)$

RMSE = Root Mean Squared Error = $\sqrt{\sum_{i=1}^{n}\frac{(\hat{y}_i - y_i)^2}{n}}$

MAPE = Mean Absolute Percentage Error = $\frac{100\%}{n}\Sigma\left|\frac{y-\widehat{y}}{y}\right|$

MAE = Mean Absolute Error = $\frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$

MSE = Mean Squared Error = $\frac{1}{n}\Sigma\left(y-\widehat{y}\right)^2$

Liite 2. Enke & Thawornwong (2005), algoritmien vertailun tulokset

		Korrelaatio	RMSE	SIGN
Tasomallit	Neuroverkko 1	0,023	1,1614	0,6628
	Neuroverkko 2	0,0528	1,1206	0,6860
	Neuroverkko 3	0,0714	1,1206	0,6860
	Regressio	0,030	1,4467	0,4767
Suuntamallit	Neuroverkko 1	0,2300	1,2200	0,6279
	Neuroverkko 2	0,3150	1,0997	0,6977
	Neuroverkko 4	0,3020	1,2575	0,6047

Liite 3. Patel et al (2015b). Indikaattorit ja selitteet

Name of indicators	Formulas
Simple $n(10\,here)$-day Moving Average	$\frac{C_t+C_{t-1}+\ldots+C_{t-9}}{n}$
Weighted $n(10\,here)$-day Moving Average	$\frac{(10)C_t+(9)C_{t-1}+\ldots+C_{t-9}}{n+(n-1)+\ldots+1}$
Momentum	$C_t - C_{t-9}$
Stochastic $K\%$	$\frac{C_t-LL_{t-(n-1)}}{HH_{t-(n-1)}-LL_{t-(n-1)}} \times 100$
Stochastic $D\%$	$\frac{\sum_{i=0}^{n-1} K_{t-i}}{10}\%$
Relative Strength Index (RSI)	$100 - \frac{100}{1+\left(\sum_{i=0}^{n-1} UP_{t-i}/n\right) / \left(\sum_{i=0}^{n-1} DW_{t-i}/n\right)}$
Moving Average Convergence Divergence (MACD)	$MACD\,(n)_{t-1} + \frac{2}{n+1} \times (DIFF_t - MACD\,(n)_{t-1})$
Larry William's R%	$\frac{H_n-C_t}{H_n-L_n} \times 100$
A/D (Accumulation/Distribution) Oscillator	$\frac{H_t-C_{t-1}}{H_t-L_t}$
CCI (Commodity Channel Index)	$\frac{M_t-SM_t}{0.015D_t}$

C_t is the closing price, L_t is the low price and H_t the high price at time t, $DIFF_t = EMA\,(12)_t - EMA\,(26)_t$, EMA is exponential moving average, $EMA\,(k)_t = EMA\,(k)_{t-1} + \alpha \times (C_t - EMA\,(k)_{t-1})$,$\alpha$ is a smoothing factor which is equal to $\frac{2}{k+1}$, k is the time period of k-day exponential moving average, LL_t and HH_t implies lowest low and highest high in the last t days, respectively. $M_t = \frac{H_t+L_t+C_t}{3}$, $SM_t = \frac{(\sum_{i\ 1}^{n} M_{t-i+1})}{n}$, $D_t = \frac{(\sum_{i=1}^{n} |M_{t-i+1}-SM_t|)}{n}$, UP_t means upward price change while DW_t is the downward price change at time t.

Liite 4. Indikaattorien muokkaus diskreeteiksi sisäisten ominaisuuksien mukaan.

Indikaattori	Ennustaa nousevaa arvo kun,
Simple moving average	Ct > SMA
Weighted moving average	Ct > WMA
Momentum	Momentum > 0
Stochastic K%	STCK%(t) > STCK% (t-1)
Stochastic D%	STCD%(t) > STCD% (t-1)
RSI	RSI < 30 tai RSI(t) > RSI(t-1) kun 30 <= RSI < 70
MACD	MACD(t) > MACD(t-1)
Larry William's R%	Williams R%(t) > Williams R% (t-1)
A/D Oscillator	A/D(t) > A/D(t-1)
CCI	CCI < -200 tai CCI(t) > CCI(t-1) kun -200 < CCI < 200

Ct = hinta pörssin sulkeutuessa

Muut selitteet: liite 3

Liite 5. muita käytettyjä syötteitä:

PSY = $$\frac{Up\ Movements\ in\ the\ last\ Periods_n}{Periods_n} \times 100$$

Bias = $$\frac{\mathrm{Count}(r_i | r_i \in [0, \sigma])}{1 + \mathrm{Count}(r_i | r_i \in [-\sigma, 0))}$$

[0, +σ] = tuottojen keskihajonta

r_i = tuotto kuukautena r, n = kuukausituottojen määrä

Liite 6. Enke ja Thawornwong (2005) syötteet

SP = S&P 500 hinta kuun lopussa

DIV = S&P 500 nimellinen osinko/osake kuukauden aikana

T1, T1H = Yhdysvaltain T-bill tuotto

R = S&P 500 nimellistuotto

ER =R - T1H, tuotto - riskitön tuotto

DY = osinkoprosentti

T3, T6, T12, T60, T120 = n kuukauden T-bill tuotto

CD1, CD3, CD6 = n kuukauden talletuskorko

AAA = AAA luokitellun yritysvelan korko

BAA = BAA luokitellun yritysvelan korko

PP = Producer Price Index, tuottajahintaindeksi

IP = Industrial Production Index, teollisuuden tuotantoindeksi

CP = Consumer Price, kuluttajahintaindeksi

M1 = Moneystock, FED:in käteisvarannot

TE1, TE2, TE3, TE4, TE5, TE6 = spreadit eri T-billien välillä

DE1, DE2, DE3, DE4, DE5, DE6, DE7 = Default spread eri luottoluokitusten välillä