Bitcoinin arvon ennustaminen rakenteellisen ja rakenteettoman tiedon perusteella käyttäen syväoppimista

Petra Torvinen

Pro gradu –tutkielma

Tietojenkäsittelytieteen laitos
Tietojenkäsittelytiede
Kesäkuu 2019

ITÄ-SUOMEN YLIOPISTO, Luonnontieteiden ja metsätieteiden tiedekunta, Joensuu
Tietojenkäsittelytieteen laitos
Tietojenkäsittelytiede

Torvinen, Petra Liisa Maria: Bitcoinin arvon ennustaminen rakenteellisen ja rakenteettoman tiedon perusteella käyttäen syväoppimista
Pro gradu –tutkielma, 54 s.
Pro gradu –tutkielman ohjaajat: Markku Tukiainen ja Erkki Pesonen
Kesäkuu 2019

Tutkielma käsittelee kryptovaluutta bitcoinin arvon ennustamista rakenteellisen ja rakenteettoman tiedon perusteella, hyödyntäen syväoppimista. Osana tutkielmaa on toteutettu kokeellinen osa, jossa bitcoinin arvon suuntaa ennustettiin seuraavalle vuorokaudelle käyttäen rakenteellisia ja rakenteettomia tietoja. Kokeellista osaa varten rakenteellisia ja rakenteettomia tietoja hankittiin erilaisista ilmaisista palveluista. Rakenteellisina tietoina käytettiin bitcoinin historiatietoja ja bitcoiniin liittyvien twiittien vuorokausikohtaisia volyymitietoja. Rakenteettomana tietona käytettiin bitcoiniin liittyviä suosituimpia twiittejä, joista tehtiin vuorokausikohtaisia sentimenttianalyyseja. Kokeellisen osan tarkoituksena oli ennen kaikkea kokeilla, kuinka hyvin bitcoinin arvoa voidaan ennustaa seuraavalle vuorokaudelle. Kokeellisessa osassa tutkittiin myös, millaisilla koulutusdatan yhdistelmillä saavutetaan parhaat ennustetarkkuudet. Kokeellisessa osassa valittiin käytettäväksi takaisinkytkeytyvä LSTM -neuroverkko, josta käytettiin viittä erilaista sovellettua mallia. Näihin LSTM -neuroverkon sovellettuihin malleihin käytettiin pohjana aiemmissa artikkeleissa kuvattuja ja käytettyjä malleja. Kokeellisessa osassa neuroverkoissa valittiin käytettäväksi vain muutamia erilaisia parametriyhdistelmiä, koska käytössä oleva aika oli rajallinen. Kokeellisessa osassa paras saavutettu ennustetarkkuus oli 0.791, kun ennustettiin bitcoinin arvon suuntaa seuraavalle vuorokaudelle käyttäen historiatietoja, twiittien volyymitietoja sekä suosituimpien twiittien sentimenttianalyyseja. Kokeellisen osan paras ennustetarkkuus saavutettiin käyttäen takaisinkytkeytyvän LSTM -neuroverkon sovellusta, joka sisälsi useita piilotettuja kerroksia. Tutkielman teksti koostuu kokeellisen osan toteutuksen ja tuloksien lisäksi kirjallisuuskatsauksesta, joka painottuu tutkielman alkupuolelle. Kirjallisuuskatsauksessa käsitellään myös aiempia bitcoinin arvon ennustamiseen liittyviä tutkimuksia.

Avainsanat: bitcoin, rakenteellinen tieto, rakenteeton tieto, neuroverkot, LSTM -neuroverkko

ACM-luokat (ACM Computing Classification System, 1998 version): I.2.m, I.2.7

UNIVERSITY OF EASTERN FINLAND, Faculty of Science and Forestry, Joensuu
School of Computing
Computer Science

Torvinen, Petra Liisa Maria: Deep learning based bitcoin price prediction using structured and unstructured data
Master's Thesis, 54 p.
Supervisors of the Master's Thesis: Markku Tukiainen and Erkki Pesonen
June 2019

This thesis deals with prediction of the price of bitcoin using structured and unstructured data, utilizing deep learning. The thesis focuses on the experimental part where the direction of bitcoin price was predicted for the next day using structured and unstructured data. In the experimental part, structured and unstructured data was obtained from variety of free services. Bitcoin history data and daily volume data for tweets were used as structural data. Sentiment analyses of the most popular tweets associated with bitcoin were used as unstructured data. The purpose of the experimental part was to test how well the value of bitcoin can be predicted for the next day. The experimental part also examined which combinations of training data will achieve the best prediction accuracy. LSTM neural network was selected for use with five different models in the experimental part. The models of these LSTM neural networks were based on the models described and used in previous articles. Only a few different combinations of parameters were selected for use in the LSTM neural networks, because available time was limited. In the experimental part, the best prediction accuracy was 0.791 when using history data, tweets volume data and sentiment analyses of the most popular tweets. The best accuracy of the experimental part was achieved using a recurrent LSTM neural network that contained multiple hidden layers. In addition to the experimental part, the text of the thesis consists of a literature review. The literature review also discusses previous studies related to bitcoin value prediction.

Keywords: bitcoin, structured data, unstructured data, neural networks, LSTM neural network

CR Categories (ACM Computing Classification System, 1998 version): I.2.m, I.2.7

Esipuhe

Tämä pro gradu -tutkielma on tehty Itä-Suomen yliopiston Tietojenkäsittelytieteen laitokselle keväällä 2019. Haluan kiittää erityisesti ohjaajiani Markku Tukiaista ja Erkki Pesosta asiantuntevasta ja kannustavasta ohjauksesta. Kiitos myös tutkielmani tarkastajalle Ville Hautamäelle. Ystävääni Aaroa haluan kiittää avusta tutkielmani kokeellisen osan käyttötapauksen aiheen ideoinnissa. Lisäksi haluan kiittää avopuolisoa, perhettä ja ystäviä tuesta tämän projektin aikana.

Joensuussa 9.6.2019

Petra Torvinen

Lyhenneluettelo

API	Joukko toimintoja ja funktioita, joiden avulla voidaan luoda sovelluksia.
CNN	Konvoluutio neuroverkko.
csv	Tiedostomuoto, jolla tallennetaan tietoa tekstitiedostoon. Tiedot ovat taulukkomuotoisia.
GRU	Gated recurrent units, on erikoistapaus takaisinkytkeytyvästä neuroverkosta.
JSON	JavaScript Object Notation, on tiedonvälityksessä käytettävä tiedostomuoto.
LSTM	Long short-term memory on erikoistapaus takaisinkytkeytyvästä neuroverkosta. LSTM -verkossa yksittäisellä neuronilla on muistisolu, joka mahdollistaa erilaisten yhteyksien löytämisen datasta.
MSE	Keskimääräinen neliövirhe. Neuroverkkojen kouluttamisessa käytettävä virheen laskentakaava.
ReLu	Aktivointifunktio, jota käytetään neuroverkkojen kouluttamisessa.
USD	Yhdysvaltain dollari.
UTC	Coordinated Universal Time on aikastandardi, jota maailmassa käytetään.

Sisällysluettelo

1 Johdanto

Kryptovaluutta bitcoin on ollut esillä viime vuosina erityisesti suurien arvon vaihteluidensa ja erilaisten siihen liittyvien huijauksien vuoksi. Bitcoinien käyttö on yleistynyt niin sijoitusmarkkinoilla kuin myös käyttörahana. Nykyään bitcoineja voi käyttää esimerkiksi joissain verkkokaupoissa ja matkanvaraussivustoilla. Bitcoinin arvon ennustamista on tutkittu viime vuosina jonkin verran, mutta tutkimuksien tulokset ovat olleet erittäin vaihtelevia. Bitcoinin arvon ennustaminen on mielenkiintoinen aihe, koska bitcoinin hinnanmuodostumista ei voida selittää vastaavalla tavalla kuin tavallisten valuuttojen kohdalla, eikä siihen muutenkaan löydy yksiselitteistä selitystä. [1] [2]

Tämä tutkielma on rakentunut bitcoinin arvon ennustamisen ympärille ja tämän tutkielman kokeellisessa osassa tavoitteena oli ennustaa bitcoinin arvon suuntaa mahdollisimman tarkasti. Käytettäväksi menetelmäksi valikoitui syväoppiminen, sillä sen avulla on saatu viime vuosina hyviä tuloksia ennustetehtävissä ja se on menetelmänä muutenkin itseäni kiinnostava. Lisäksi olin kiinnostunut erityisesti rakenteellisen ja rakenteettoman tiedon yhdistämisestä koneoppimisessa, joten tässä työssä yhdistyvät monet minua kiinnostaneet osa-alueet.

Osana tutkielmaa on toteutettu kokeellinen osa, jossa bitcoinin arvon suuntaa ennustettiin seuraavalle vuorokaudelle käyttäen bitcoinin historiatietoja, bitcoiniin liittyvien suosituimpien twiittien vuorokausikohtaisia volyymitietoja sekä bitcoiniin liittyvien twiittien vuorokausikohtaisia sentimenttianalyyseja. Tutkielma painottuu kokeelliseen osaan, mutta tutkielman alkupuolelle on sisällytetty kirjallisuuskatsaus aiheeseen liittyen. Kirjallisuuskatsauksessa käsitellään myös aiempia bitcoinin arvon ennustamiseen liittyviä tutkimuksia.

Kokeellisen osan tarkoituksena oli ennen kaikkea kokeilla, kuinka hyvin bitcoinin arvoa voidaan ennustaa seuraavalle vuorokaudelle käyttäen syväoppimista. Kokeellisessa osassa tutkittiin myös, millaisilla koulutusdatan yhdistelmillä saavutetaan parhaat ennustetarkkuudet. Kokeellisessa osassa valittiin käytettäväksi takaisinkytkeytyvä LSTM -neuroverkko, josta käytettiin viittä erilaista sovellettua mallia.

Kokeellisessa osassa neuroverkoissa valittiin käytettäväksi vain muutamia erilaisia parametriyhdistelmiä, koska käytössä oleva aika oli rajallinen.

Tämä tutkielma koostuu seitsemästä luvusta. Luvussa 2 käsitellään tutkimuksen taustaa, sisältäen muun muassa aiheeseen liittyvien käsitteiden määrittelyä ja katsauksen aiempiin bitcoinin arvon ennustamiseen liittyviin tutkimuksiin. Tämän jälkeen, luvussa 3 käsitellään rakenteellisen ja rakenteettoman tiedon yhdistämistä, ja tähän liittyen rakenteettoman tiedon prosessointia. Lisäksi luvussa 3 kerrotaan rakenteellisen ja rakenteettoman tiedon yhdistämisestä aiemmissa tutkimuksissa. Luvussa 4 kerrotaan neuroverkoista, niiden rakenteesta, neuroverkkotyypeistä sekä neuroverkkojen kouluttamisesta ja mallien testaamisesta. Luku 5 käsittelee kokeellisen osan toteutusta eli rakenteellisten ja rakenteettomien tietojen hankkimista, rakenteettomien tietojen prosessointia, tietojen yhdistämistä sekä käytettäviä neuroverkkoja sekä niiden kouluttamista. Luvussa 6 kerrotaan kokeellisen osan tuloksista. Tutkielman lopussa, luku 7 sisältää johtopäätökset ja yhteenvedon.

2 Tutkimuksen taustaa

Tämän tutkielman kokeellisen osan käyttötapauksena on kryptovaluutta *bitcoinin* arvon suunnan ennustaminen seuraavalle vuorokaudelle *rakenteellisen* sekä *rakenteettoman tiedon* avulla *syväoppimista* hyödyntäen. Kokeellisessa osassa keskitytään erityisesti siihen, että mitä rakenteellisia ja rakenteettomia tietoja on järkevää yhdistää, jotta ennusteista saadaan mahdollisimman hyviä. Kokeellisessa osassa rakenteellisena tietona käytetään bitcoinin kurssin historiadataa sekä twiittien *volyymitietoja*, ja rakenteettomana tietona käytetään bitcoiniin liittyviä twiittejä. Twiitit käsitellään ja niistä tehdään vuorokausikohtaiset *sentimenttianalyysit*, jotka sitten yhdistetään rakenteelliseen tietoon eli rakenteellista tietoa ikään kuin rikastetaan rakenteettomalla tiedolla. Nämä tiedot yhdessä syötetään *neuroverkkoon*, ja näin pyritään saavuttamaan mahdollisimman hyviä tuloksia bitcoinin arvon suunnan ennustamisessa seuraavalle vuorokaudelle. Kokeellisessa osassa ennustejärjestelmä luodaan itse, hyödyntäen muun muassa valmiita kirjastoja ja käyttämällä valmiita *neuroverkkomalleja* pienillä muokkauksilla. Kokeellisessa osassa ei vertailla erityyppisiä neuroverkkoja, eikä myöskään erityisesti neuroverkkojen kouluttamiseen käytettäviä algoritmeja, vaan hyödynnetään tältä osin aiemmissa tutkimuksissa hyväksi havaittuja valintoja.

Tämä tutkielma on rakentunut kokeellisen osan ympärille, joten kirjallisuuskatsauksessa käsitellään kokeellisessa osassa käytettäviä menetelmiä, rakenteita ja muita asioita. Kirjallisuuskatsausta ja tutkielman kokeellista osaa on tehty osittain rinnakkain, joten kirjallisuuskatsaus on täydentynyt kokeellisen osan aikana. Kirjallisuuskatsauksessa ja koko työssä ensimmäisinä lähteinä ovat toimineet aiemmat bitcoinin arvon ennustamiseen liittyvät tutkimukset ja projektit. Niiden pohjalta on määritelty tämän tutkielman kokeellisen osan sisältö ja tutkimuskysymykset. Kirjallisuuskatsauksen muut lähteet on etsitty erinäköisistä tietokannoista ja internetistä sitä myötä, kun jotain tietoa on tarvittu.

Kohdassa 2.1 määritellään rakenteellinen ja rakenteeton tieto, jotka ovat tämän tutkielman kannalta olennaisia käsitteitä. Kohdassa 2.2 puolestaan kerrotaan *tiedon laadusta*, joka on myös olennainen käsite tällaisessa tutkielmassa ja sen kokeellisessa osassa. Kohdassa 2.4 kerrotaan bitcoinista, sen kurssimuutoksista ja niihin johtaneista

mahdollisista syistä vuoden 2016 alusta nykyhetkeen. Muutamista bitcoinin arvon ennustamista käsittelevistä aiemmista tutkimuksista ja niiden tuloksista kerrotaan kohdassa 2.5.

2.1 Rakenteellinen ja rakenteeton tieto

Rakenteellinen tieto sijaitsee tyypillisesti relaatiotietokannoissa, jonka kenttiin tallennetaan pituudeltaan ja ominaisuuksiltaan rajattua tietoa. Kentät voivat olla esimerkiksi numeerisia, merkkijonoja, päivämääriä tai merkkejä. Rakenteellinen tieto on ihmisten tai koneiden tuottamaa sekä säilömää, ja siihen on helppo kohdistaa erilaisia hakuja ja kyselyjä. Rakenteellisesta tiedosta tyypillinen esimerkki on csv-tiedosto, jossa sarakkeiden muodolle ja sisällölle on asetettu tarkat rajoitteet. [3]

Rakenteeton tieto puolestaan on kaikkea tietoa, missä ei ole ennalta määriteltyä tarkkaa tietomallia. Tyypillisimpiä esimerkkejä rakenteettomasta tiedosta ovat kirjoitetut tekstit ja sosiaalisen median päivitykset, kuvat, videot sekä äänitteet. Myös rakenteeton tieto on ihmisten tai koneiden tuottamaa, mutta siihen ei ole läheskään yhtä helppoa kohdistaa hakuja kuin rakenteelliseen tietoon. [3] Tutkielman kokeellisessa osassa rakenteellisena tietona käytetään bitcoinin historiatietoja sekä bitcoiniin liittyvien twiittien volyymitietoja, ja rakenteettomana tietona käytetään bitcoiniin liittyviä twiittejä.

2.2 Tiedon laatu

Tiedon laadun määritelmä riippuu kontekstista ja tietojen käyttäjän vaatimuksista. Tiedon laatua koskevat odotukset eivät välttämättä ole aina selkeitä, ja siksi niitä joudutaan usein määrittelemään tapauskohtaisesti. Tiedon laatu voidaan kuitenkin määritellä muutamilla yleisesti käytetyillä laatukriteereillä, joita ovat muun muassa aineiston kattavuus, tarkkuus, eheys, ajantasaisuus, oikeellisuus ja saavutettavuus. Olennaisinta tiedon laadun osalta on kuitenkin se, että se soveltuu käyttötapaukseen ja täyttää käyttäjän tarpeet. [4]

Tiedon laatu on tärkeää ottaa huomioon myös tämän tutkielman kokeellisessa osassa, sillä tietoja ladataan useista erilaisista ilmaisista rajapinnoista ja lähteistä.

Käytettävistä ilmaisista rajapinnoista ja lähteistä johtuen tiedon laadun varmistaminen on kuitenkin haasteellista. Siitä huolimatta monia laatukriteerejä voidaan tarkastella myös tämän tutkielman kokeellisessa osassa käytettävien tietojen osalta. Esimerkiksi aineiston kattavuutta voidaan tarkastella, eli voidaan katsoa, että onko valitulta ajanjaksolta jokaiselta vuorokaudelta saatavilla kaikki tai ainakin suurin osa tiedoista. Aineiston tarkkuuden ja oikeellisuuden tarkasteleminen on puolestaan haasteellisempaa, sillä kyseessä on ilmaiset tietolähteet, ja siten näiden laatukriteerien toteutumisen määritteleminen on haastavaa.

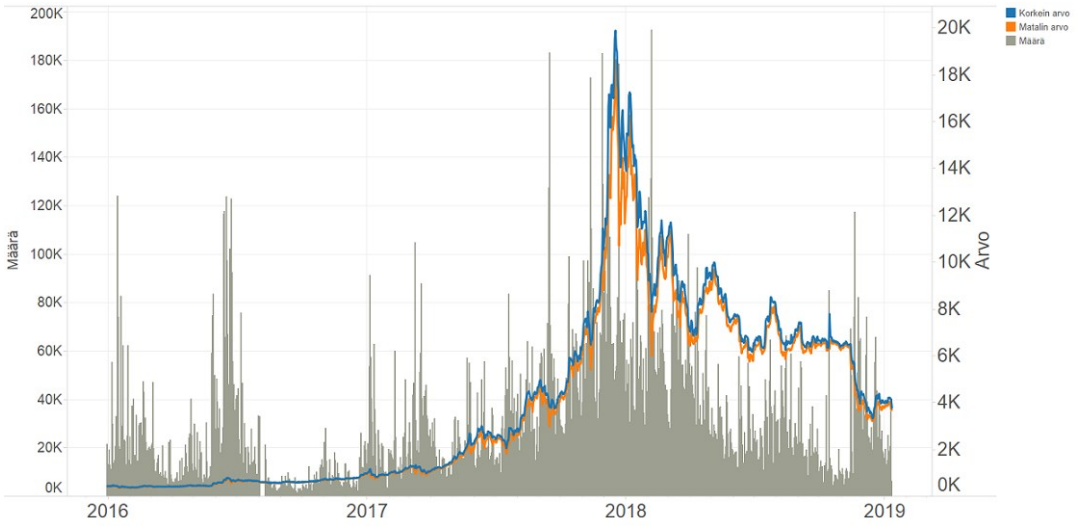
2.3 Bitcoin ja kurssimuutokset

Bitcoin on vuonna 2008 luotu kryptovaluutta, joka pohjautuu *hajautettuun kirjanpitotietokantaan*. Hajautetun kirjanpitotietokannan eli bitcoin -verkon keskeinen tietorakenne on *lohkoketju*, johon tallennetaan kaikki bitcoin -verkossa tehdyt *siirrot*. Siirrolla tarkoitetaan kahden käyttäjän välistä rahanvaihtoa. Lohkoketjussa yksittäiset lohkot on liitetty yhteen ketjuksi siten, että edellisen lohkon otsikon tiiviste on sisällytetty seuraavan lohkon otsikkoon. Siirtohistorian muokkaamisesta on tehty kannattamatonta, kun lohkojen luomisesta on tehty riittävän kallista. Uusia lohkoja luotaessa lasketaan lohkon otsikolle sellainen tiiviste, joka saavuttaa riittävän vaikeustason. Uusi lohko on hyväksyttävä, jos se täyttää asetetun vaikeustason ja kaikki lohkon siirrot ovat sääntöjen mukaisia. Sääntöjen mukaisissa siirroissa sisääntulojen summa on yhtä suuri tai suurempi kuin ulostulojen summa. Siirrossa sisääntulo muodostuu siis viittauksesta edellisen siirron ulostuloon sekä komentojonoista, ja ulostulo puolestaan komentojonosta sekä bitcoinien määrästä. Komentojonot ovat ihmisten päätettävissä, mutta niille on asetettu muutamia rajoitteita. Jokaisen lohkon ensimmäinen siirto on niin kutsuttu *luojasiirto*, joka poikkeaa muista siirroista siten, että siltä ei vaadita sisääntuloja. Luojasiirtoa on rajoitettu siten, että sen ulostulojen summa saa olla maksimissaan siirtopalkkiot yhdistettynä uusien bitcoinien sallittuun määrään. Luojasiirroilla saadaan luotua uusia bitcoineja ja lisäksi saadaan ihmiset osallistumaan uusien lohkojen muodostamiseen. Tällä tavalla lohkoista muodostuu lohkoketju, jossa on sisällä bitcoinien täydellinen siirtohistoria. On myös mahdollista, että lohkoketjussa on haaraumia. Yksimielisyyden saavuttamiseksi onkin määritelty, että voimassa oleva lohkoketju on se, minkä lohkojen vaikeustason summa on suurin. [5, pp. 33-57] [6]

Bitcoin mahdollistaa siis kahden tai useamman osapuolen rahanvaihtamisen keskenään verkon välityksellä, ilman että se kulkee välittäjien, kuten pankkien tai maksuprosessorien kautta. Kustannukset vähenevät, kun käyttäjät itse validoivat tapahtumat vertaisverkossa. Bitcoineja säilytetään joko henkilökohtaisella tietokoneella tai kolmannen osapuolen palveluntarjoajalla, eli valuuttaa ei fyysisesti ole olemassa. Helppokäyttöisyys ja anonyymisyys olivat bitcoinin alkuperäisen suunnitelman avainasioita, sillä käyttäjiä ei vaadita rekisteröimään henkilöllisyyttään tiliä luotaessa, joten käyttäjät voivat suojella identiteettiään käyttämällä esimerkiksi salanimiä. Kiinnostus bitcoinia kohtaan on viime vuosina kasvanut suuresti, mutta sen myötä myös erinäköiset huijaukset ja rikollinen toiminta ovat yleistyneet. [5, pp. 1-5]

Bitcoinin hinnanmuodostumista ei voida selittää vastaavalla tavalla kuin tavallisten valuuttojen kohdalla, sillä monet tavallisten valuuttojen kysynnän ja tarjonnan piirteet puuttuvat. Bitcoinia ei ole myöntänyt mikään keskuspankki tai hallitus, joten ei ole makrotaloudellisia perusteita bitcoinin hinnanmuodostumisen määrittämiseen. Bitcoinin hintakehitystä vaikuttaisi hallitsevan sen kysyntä ja tarjonta, sekä esimerkiksi huijaukset, huhut ja ylipäätään maailman tapahtumat. [1] [2]

Bitcoinin kurssikehitys ja vaihdettujen bitcoinien arvo vuoden 2016 alusta nykyhetkeen on visualisoituna kuvassa 1. Visualisointi on tehty Tableau -työkalulla. Yksikönä visualisoinnissa on *USD* eli Yhdysvaltain dollari, sillä sitä käytetään tyypillisimmin kaikissa tutkimuksissa ja uutisissa. Visualisointiin päiväkohtainen historiadata on ladattu Bitfinex:in avoimesta rajapinnasta [7]. Visualisoinnissa harmaa alue kuvaa *kaupankäynnin volyymia* eli vaihdettujen bitcoinien arvoa päivittäin, ja vasemmanpuoleiselta akselilta voi lukea tämän pylväsdiagrammin arvot. Sininen viivadiagrammi kuvaa bitcoinin korkeinta päivittäistä arvoa ja oranssi matalinta päivittäistä arvo, näiden arvot voi lukea oikeanpuoleiselta akselilta.



Kuva 1. Bitcoinin kurssikehitys ja vaihdettujen bitcoinien arvo vuoden 2016 alusta nykyhetkeen. Datan lähde Bitfinex:in avoin rajapinta.

Historiadatan perusteella bitcoinin arvo on 2016 vuoden alusta 2017 vuoden toukokuun alkuun asti keskimäärin noussut tasaisesti, mutta maltillisesti koko ajan. Vuoden 2016 alussa bitcoinin arvo oli noin 430 USD ja vuonna 2017 toukokuun alussa arvo oli noin 1 500 USD. Toukokuusta 2017 alkaen bitcoinin arvo alkoi kasvamaan huomattavasti kovempaa vauhtia kuin aiemmin ja lokakuun lopussa sen arvo oli jo noin 6 200 USD. Seuraavat pari kuukautta bitcoinin arvo nousi suorastaan räjähdysmäisesti, ja 17. joulukuuta 2017 bitcoin saavutti tähänastisen suurimman arvonsa eli 19 891 USD. Tämän jälkeen bitcoinin arvossa tapahtuikin suunnan muutos ja siitä lähtien arvo on pääsääntöisesti laskenut, vaikka välillä on ollut melko suuriakin heittoja kumpaankin suuntaan. Marraskuun loppupuolella vuonna 2018 arvo romahti entisestään, ja sen jälkeen bitcoinin arvo on pyörinyt noin 3 800 USD:n molemmin puolin.

Bitcoinin kaupankäynnin volyymi eli päivittäin vaihdettujen bitcoinien arvo on vaihdellut huomattavasti suuremmin kuin bitcoinin arvo, ja todella suuria yksittäisiä piikkejä on useita vuoden 2016 alusta alkaen. Vuoden 2016 alusta saman vuoden elokuun alkupuolelle bitcoinin kaupankäynnin volyymi oli jopa suhteellisen suurta, vaikka bitcoinin arvo kasvoi silloin maltillisesti. Esimerkiksi kesäkuun puolivälissä vuonna 2016 on useita päiviä, jolloin kaupankäynnin volyymi on ollut yli 120 000 USD. Tämän jälkeen kaupankäynnin volyymi pieneni, ja alkoi kasvaa huomattavasti arvon nousemisen myötä 2017 vuoden puolivälistä alkaen. Kaupankäynnin volyymin suurin päivittäinen arvo 192 768 USD saavutettiin 6. helmikuuta 2018, kun samaan aikaa

bitcoinin arvo romahti merkittävästi. Kaupankäynnin volyymi oli keskimäärin suurimmillaan, kun myös bitcoinin arvo oli suurimmillaan ja myös kaupankäynnin volyymin pääsääntöinen suunta lähti laskuun bitcoinin arvon laskiessa. Marraskuun puolivälissä vuonna 2018 kaupankäynnin volyymi kasvoi uudestaan, samaan aikaan kun arvo romahti entisestään ja sen jälkeen se on pyörinyt noin 30 000 USD:n molemmin puolin.

Edellisen vuoden ajalta uutisista löytyy paljon erilaisia arvioita bitcoinin markkinoiden muutoksiin liittyen. Esimerkiksi vuoden 2018 tammikuulta löytyy useita uutisia, joissa kerrotaan mahdollisia syitä saavutetun arvohuipun jälkeiselle romahdukselle. Näitä mahdollisia syitä ovat Kiinan ja Etelä-Korean toimien myötä kaikonneet sijoittajat, nopeasta arvon laskusta aiheutunut vilkas bitcoinien myyminen sekä voimakkaasti kasvanut kiinnostus kryptovaluuttamarkkinoita kohtaan. [8] [9] Edelliseen merkittävään muutokseen eli vuoden 2018 marraskuun puolivälin bitcoinin arvon romahdukseen liittyen löytyy arvioita useista mahdollisista syistä, joista esimerkkinä 2017 vuoden joulukuussa mahdollisesti tapahtunut keinotekoinen bitcoinin arvon hetkellinen nostaminen. Mahdollisen keinotekoisen arvon hetkellisen nostamisen uskotaan jopa vaikuttaneen koko 2018 vuoden kurssiin. [10] [11] Useita erilaisia teorioita on siis liikkeellä ja on myös mahdollista, että jotkin tapahtumat ovat vain sattuneet osumaan samanlaisiin ajankohtiin kuin bitcoinin arvon suurimmat muutokset.

2.4 Aikasarjaennustaminen

Tämän tutkielman kokeellisessa osassa käytetään aikasarjatietoja. Aikasarjojen ennustamiseen on käytettävissä useita erilaisia menetelmiä, jotka voidaan ryhmitellä karkeasti *tilastollisiin menetelmiin* ja koneoppimiseen pohjautuviin menetelmiin. [12, pp. 51-55] [13]Tämän tutkielman kokeellisessa osassa on valittu käytettäväksi koneoppimiseen pohjautuvista menetelmistä syväoppiminen, koska haluttiin tutkia juuri sen menetelmän käyttämistä tutkielman kokeellisen osan käyttötapaukseen.

Tilastolliset menetelmät pohjautuvat matematiikkaan ja tilastotieteeseen. Tilastolliset menetelmät ovat objektiivisia sekä johdonmukaisia ja pystyvät käsittelemään suuria datamääriä. Tilastollisten menetelmien avulla voidaan ottaa huomioon monia muuttujia sekä monimutkaisia suhteita muuttujien välillä. Tilastollisten menetelmien käyttäminen edellyttää erityisesti teknistä ymmärrystä käytettävästä mallista.

Aikasarjatietoihin käytettävät tilastolliset mallit perustuvat historiatietojen analysointiin. Eräs esimerkki tilastollisista menetelmistä ovat regressiomallit, jotka perustuvat muuttujien yhteyksien estimointiin. [12, pp. 51-55,78-87]

Koneoppimiseen pohjautuvia menetelmiä puolestaan ovat erilaiset neuroverkkojen sovellukset, algoritmiset lähestymistavat sekä tukivektorikoneet, joita voidaan kaikkia soveltaa myös aikasarjaennustamiseen. Aikasarjaennustamiseen käytetään esimerkiksi takaisinkytkeytyviä neuroverkkoja, joita hyödynnetään myös tämän tutkielman kokeellisessa osassa. Neuroverkkojen etuna on, että ne pystyvät oppimaan monimutkaisia yhteyksiä suuresta datamäärästä. [14] Neuroverkoista ja niiden sovelluksia kerrotaan tarkemmin luvussa 4. Joissain tapauksissa voidaan myös yhdistää erilaisia menetelmiä. Esimerkiksi Pant & All. [15] tekemässä tutkimuksessa käytettiin sekä algoritmeja että neuroverkkoja aikasarjaennustamiseen, kun ennustettiin bitcoinin arvoa.

2.5 Aiemmat tutkimukset

Bitcoinin kurssin ennustamista on tutkittu viime vuosina melko paljon käyttäen monia erilaisia menetelmiä sekä erilaisia kombinaatiota hyödynnettävästä datasta. Tässä kohdassa esitellään neljä erilaista bitcoinin kurssin ennustamiseen liittyvää tutkimusta, joissa on hyödynnetty erilaisia menetelmiä ja tulkittu tuloksia erilaisista näkökulmista. Tutkimuksissa on käytetty tilastollisia sekä koneoppimiseen pohjautuvia menetelmiä ja lisäksi joidenkin tutkimuksien toteutuksissa on ikään kuin yhdistetty näitä molempia menetelmiä.

Erilaisten koneoppimismallien vertailu bitcoinin arvon ennustamisessa

Phaladisailoed ja Numnonda [16] vertailivat tutkimuksessaan muutamia erilaisia koneoppimismalleja bitcoinin arvon ennustamisessa. He käyttivät tutkimuksessaan dataa vaihtokursseista minuutin tarkkuudella aikaväliltä 1.1.2012 - 8.1.2018. Tutkimuksessa data muutettiin päivätasolle ja siitä valittiin käytettäviksi ominaisuuksiksi päivän viimeisin kauppa, ensimmäinen kauppa, suurin kauppa, pienin kauppa, bitcoinin keskihinta, koko kaupankäynnin määrä bitcoineina, koko kaupankäynnin määrä Yhdysvaltain dollareina ja tietojen tallennusaika. Kaikki nämä ominaisuudet osoittautuivat korreloituviksi ennustettavan asian eli bitcoinin arvon kanssa. Tutkimuksessa data jaettiin

kahteen osaan, jossa opetusosa oli 70% ja testiosa 30%. Tässä tutkimuksessa valittiin käytettäväksi Scikit-learn -kirjaston regressiomallit *Theil-Sen* -regressio ja *Huber* -regressio sekä lisäksi syväoppimiseen pohjautuvista regressiomenetelmistä *LSTM*- ja *GRU* -mallit. Theil-Sen -regressio on menetelmä, joka käyttää kaikkien datapisteiden kautta piirrettyä mediaania [16]. Tästä syystä kaksiulotteiselle datalle poikkeamat voivat kasvaa jopa 29%:iin. Huber -regressio puolestaan käyttää lineaarista häviötä (linear loss) erottaakseen poikkeamat datasta [16]. LSTM- ja GRU -mallit ovat erikoistapauksia *takaisinkytkeytyvästä neuroverkosta*. Eri neuroverkkotyypeistä kerrotaan tarkemmin luvussa 4. Tutkimuksessa eri koneoppimismallien tarkkuutta arvioitiin käyttäen *keskimääräistä neliövirhettä* (MSE) ja *R-neliötä* (R-Square). Tutkimus osoitti, että syväoppimiseen pohjautuvat regressiomallit eli GRU- ja LSTM -malli antavat parempia tuloksia kuin Theil-Sen -regressio ja Huber -regressio. Parhaimmat tulokset tutkimuksessa saatiin GRU -mallista, jonka neliövirhe oli 0.00002 ja R-neliö 0.992 eli 99.2%. Tutkimuksessa vertailtiin myös eri koneoppimismallien laskenta-aikaa, ja ylivoimaisesti nopein oli Huber -regressio, jonka laskenta-aika oli vain 0.0002 sekuntia. LSTM-mallin laskenta-aika oli noin 111 sekuntia ja GRU -mallin noin 85 sekuntia. Tutkimuksessa kuitenkin todetaan, että parempia tuloksia voitaisiin saada yhdistämällä esimerkiksi sosiaalisen median dataa historiadataan. [16]

Bitcoinin arvon ennustaminen takaisinkytkeytyvien neuroverkkojen avulla

Pant & All. [15] tekemässä tutkimuksessa normalisoidut bitcoinin historiatiedot syötettiin yhdessä twiiteistä saatujen päiväkohtaisten sentimenttiosuuksien kanssa takaisinkytkeytyvään neuroverkkoon, joka sitten ennusti numeerista bitcoinin arvoa. Twiittien luokitteluun oli tässä tutkimuksessa käytetty kahta menetelmää, jotka olivat *Word2Vector* ja *Bag-of-Words*. Word2Vector tekee jokaisesta twiitistä 300-ulotteisen vektoriesityksen, jota käytetään sitten ominaisuutena. Bag-of-words puolestaan muodostaa taajuusjakauman twiitin kaikista sanoista, ja näitä taajuuspisteitä käytetään sitten myös ominaisuutena myöhemmin. Näistä molemmista menetelmistä saadut ominaisuudet yhdistettynä käsin luokiteltuihin twiitteihin syötettiin sitten koulutusdatana viidelle erilaiselle algoritmille: *Naive Bayes, Bernoulli Naive Bayes, Multinomial Naive Bayes, Linear Support Vector Classifier* ja *Random Forest*. Tämän jälkeen luotiin niin sanottu äänestysluokittelija, joka otti jokaisen algoritmin tuotoksen ja luokitteli siten uudet twiitit siihen luokkaan, jolle ääni oli suurin. Aikasarjojen

ennustamiseen käytettiin siis takaisinkytkeytyvää neuroverkkoa ja sen sovelluksia LSTM- ja GRU -neuroverkkoja. Tämän tutkimuksen suurin tuotos oli äänestysluokittelija, joka pystyi 81.39% tarkkuudella luokittelemaan bitcoiniin liittyvät twiitit positiivisiin ja negatiivisiin. Takaisinkytkeytyvän neuroverkon seuraavan päivän bitcoinin arvon *ennustetarkkuudeksi* saatiin 77.62%. Tutkimuksessa löydettiin myös maltillinen korrelaatio bitcoiniin liittyvien negatiivisten twiittien ja bitcoinin arvon laskun välillä, joka oli 0.41. [15]

Kryptovaluuttojen arvojen ennustaminen uutisartikkeleiden ja sosiaalisen median päivityksien perusteella algoritmeja käyttäen

Lamon & All. [17] tekemässä tutkimuksessa kolmen kryptovaluutan arvon ennustamiseen käytettiin uutisia ja sosiaalisen median sentimenttianalyysia. Heidän tutkimuksensa tarkoitus oli selvittää, että voiko uutisartikkeleiden otsikkojen sekä sosiaalisen median päivityksien perusteella ennustaa bitcoinin, *litecoinin* ja *ethereumin* arvoa. Lisäksi he tutkivat, että ovatko uutisartikkeleiden otsikot parempi indikaattori kuin sosiaalisen median päivitykset ennustettaessa kryptovaluuttojen arvoa. Tutkimuksessa käytettiin päiväkohtaista historiadataa, uutisartikkeleiden otsikoita ja twiittejä aikaväliltä 1.1.2017 – 30.11.2017. Tutkimuksessa päivittäiset hintatiedot yhdistettiin kuudella arvolla jokaisen uutisartikkelin otsikkoon ja kahdella arvolla jokaiseen twiittiin. Uutisartikkeleiden otsikoihin yhdistettiin jokaisen kryptovaluutan arvon suunta kahden seuraavan päivän osalta ja jokaiseen twiittiin yhdistettiin kyseiseen twiittiin liittyvän kryptovaluutan arvon suunta kahden seuraavan päivän osalta. Kummassakin tapauksessa arvot olivat joko 0 eli kryptovaluutan arvon lasku tai 1 eli kryptovaluutan arvon nousu. Tämän jälkeen data jaettiin kolmeen osaa, jossa opetusosa oli 60%, testiosa 20% ja tuloksien validointiosa 20%. Tutkimuksen mallissa käytettiin neljää erilaista luokittelijaa ominaisuuspainojen oppimiseen, jotka olivat *logistinen regressio, tukivektorikone,* Bernoulli Naive Bayes ja Multinomial Naive Bayes. Jokaiselle kryptovaluutalle testattiin näistä luokittelijoista parhaiten sopiva ja tuloksia tarkasteltiin seuraavan päivän ennusteiden osalta. Bitcoinin tapauksessa paras luokittelija oli logistinen regressio, joka pystyi ennustamaan arvon nousua 43.9% tarkkuudella ja arvon laskua 61.9% tarkkuudella. Ethereumin tapauksessa paras luokittelija oli Bernoulli Naive Bayes, joka pystyi ennustamaan arvon nousua jopa 75.8% tarkkuudella, mutta arvon laskua vain 16.1% tarkkuudella. Litecoinin kohdalla paras luokittelija oli

logistinen regressio, mutta kokonaisuutena arvon ennustaminen litecoinin tapauksessa oli näistä kolmesta kolikosta heikoin. Litecoinin tapauksessa luokittelija arvon nousua saatiin ennustettua 0% tarkkuudella ja arvon laskua 100% tarkkuudella. Tutkijat arvioivat, että litecoinin tapauksessa ennustaminen epäonnistui mahdollisesti johtuen hyvin erilaisista litecoiniin liittyvistä twiiteistä opetus- ja testidatan välillä, sillä litecoinin suosio kasvoi huomattavasti tutkimuksessa käytetyn aikavälin aikana. Tutkijat myös totesivat, että tutkimuksessa käytettynä ajanjaksona kaikkien kolmen kryptovaluutan arvo keskimäärin nousi ja erityisesti testauksessa käytetyn aikavälin osalta arvon nousu oli keskimäärin huomattavasti suurempaa kuin opettamiseen käytetyn aikavälin osalta. Tutkijoiden mukaan nämä asiat ovat voineet vaikuttaa myös osittain tutkimuksen tuloksiin. [17]

Bitcoinin arvon ennustaminen historiatietojen ja twiittien perusteella algoritmeja käyttäen

Colianni & All. [18] tekemässä tutkimuksessa ennustettiin bitcoinin arvon suuntaa seuraavan tunnin sekä seuraavan päivän osalta hyödyntäen algoritmista lähestymistapaa. Tutkimuksessa hyödynnettiin avoimenlähdekoodin *Tweepy* -kirjastoa, jonka avulla ladattiin twiittejä reaaliajassa Twitterin rajapinnasta bitcoin-hakusanalla. Twiittien sisällön lisäksi tekstitiedostoon tallennettiin twiitin julkaisija, twiitin yksilöllinen tunniste ja aikaleima. Näiden lisäksi bitcoinin tuntikohtaiset arvot haettiin cryptonator.com -rajapinnasta, ja tallennettiin erilliseen tekstitiedostoon. Twiittejä sekä historiatietoja kerättiin kaksikymmentäyksi päivää ja tällä aikavälillä twiittejä kertyi yli miljoona. Algoritmin tavoitteena oli ennustaa bitcoinin arvon mahdollista nousua tai laskua valitussa aikaikkunassa, eli joko seuraavan tunnin tai seuraavan päivän osalta. Twiittien sisällön luokitteluun käytettiin kolmea erilaista algoritmia, jotka olivat logistinen regressio, Naive Bayes ja tukivektorikone Scikit.learn -kirjastosta. Ensimmäisessä kokeilussa parhaat tulokset saatiin hyödyntäen Bernoulli Naive Bayesia, jonka päiväkohtainen ennustetarkkuus oli 95.00% ja tuntikohtainen ennustetarkkuus oli 76.23%. Jälkimmäisessä kokeilussa tutkijat käyttivät textprocessing.com -rajapintaa laskemaan jokaiselle twiitille negatiivisuus-, neutraalisuus- ja positiivisuusarvot. Nämä arvot syötettiin ominaisuusvektoreihin ja niistä saadut palautusarvot annettiin puolestaan syötteeksi luokittelijoille. Tällaista ominaisuusvektoria hyödyntäen parhaat tulokset saavutettiin käyttäen logistista regressiota, jonka päiväkohtainen

ennustetarkkuus oli 86.00% ja tuntikohtainen ennustetarkkuus oli 98.58%. Parhaalle päiväkohtaisen ennustetarkkuuden saavuttaneelle luokittelijalle eli Bernoulli Naive Bayesille *F-arvo* oli 0.96, *sensitiivisyys* oli 0.92, ja yleinen ennustetarkkuus oli siis 0.95. Parhaalle tuntikohtaisen ennustetarkkuuden saavuttaneelle luokittelijalle eli logistiselle regressiolle puolestaan F-arvo oli 0.99, sensitiivisyys oli 0.98, ja yleinen ennustetarkkuus oli siis 0.986. [18]

3 Rakenteellisen ja rakenteettoman tiedon yhdistäminen

Rakenteellisen ja rakenteettoman tiedon yhdistämistä varten rakenteetonta tietoa eli twiittejä on ensin prosessoitava, koska twiitit on kirjoitettu *luonnollisella kielellä*. Luonnollisella kielellä tarkoitetaan ihmisten jokapäiväisessä viestinnässä käyttämää kieltä, eli kaikkea kirjoitettua ja puhuttua kieltä. *Luonnollisen kielen prosessoinnilla* puolestaan tarkoitetaan luonnollisen kielen koneellista käsittelyä, josta esimerkkejä ovat sentimenttianalyysi, luokittelu ja kielen kääntäminen. Luonnollisen kielen prosessointi on yleistynyt valtavasti viime vuosina ja ihmisten jokapäiväisessä elämässä se esiintyy muun muassa puhelimien ennakoivana tekstinsyöttönä ja puheentunnistussovelluksina. Luonnollisen kielen prosessointiin haastetta tuovat ainakin lukuisat erilaiset käytössä olevat kielet, murteet ja niiden muuttuminen aikojen kuluessa, mahdolliset kirjoitusvirheet sekä äänen paino puhuessa. [19, p. ix]

Luonnollisen kielen prosessoinnista ja siihen käytössä olevista vaihtoehdoista kerrotaan tarkemmin kohdassa 3.1. Rakenteellisen ja rakenteettoman tiedon yhdistäminen voidaan hoitaa ainakin muutamalla eri tavalla, ja erityisesti bitcoiniin liittyvien aiempien tutkimuksien ratkaisuista kerrotaan kohdassa 3.2.

3.1 Luonnollisen kielen prosessointi

Luonnollisen kielen prosessointiin on olemassa useita erilaisia tekniikoita. Käytettävän tekniikan valintaan vaikuttavat ainakin prosessoitavan luonnollisen kielen muoto sekä analysoitavat ominaisuudet. Alakohdassa 3.1.1 kerrotaan ominaisuuksista, joita luonnollisen kielen prosessoinnilla voidaan analysoida. Ominaisuuksista ja niiden analysoinnista kerrotaan vain pääpiirteittäin. Erityisesti tämän tutkielman kokeellisessa osassa käytettävästä sentimenttianalyysista kerrotaan alakohdassa 3.1.2. Alakohdassa 3.1.3 puolestaan kerrotaan yleiskatsaus käytössä olevista tekniikoista, mutta erityisesti keskitytään tämän tutkielman kokeellisessa osassa käytettäviin menetelmiin.

3.1.1 Luonnollisesta kielestä prosessoitavat ominaisuudet

Luonnollisesta kielestä voidaan prosessoida erilaisia ominaisuuksia. Sanoja voidaan kategorisoida eri sanaluokkiin, eli esimerkiksi substantiiveihin, verbeihin, adjektiiveihin ja adverbeihin. Sanojen kategorisoinnista on hyötyä monissa luonnollista kieltä käsittelevissä ja analysoivissa tehtävissä. [19, p. 179] Luonnollista kieltä voidaan myös luokitella. Luonnollisen kielen luokittelulla tarkoitetaan esimerkiksi roskapostien seulomista ja uutisartikkeleiden luokittelua aiheittain. [19, pp. 221-223] Myös lauseiden rakennetta ja tarkoitusta voidaan analysoida [19, pp. 291, 361]. Luonnollisesta kielestä voidaan analysoida sentimenttejä, eli tunnistaa ja luokitella esitettyjä mielipiteitä. Sentimenttianalyysista kerrotaan tarkemmin alakohdassa 3.1.2. Puhutusta kielestä voidaan lisäksi analysoida ainakin murteita ja äänen painoa [19, p. ix].

3.1.2 Sentimenttianalyysi

Sentimenttianalyysilla pyritään tunnistamaan tekstistä tunteet ja mielipiteet. Monet yritykset hyödyntävät sentimenttianalyyseja, kun haluavat saada selville esimerkiksi yleisen mielipiteen tuotteestaan. Erityisesti sosiaalisen median päivityksistä sentimenttianalyyseja tehdään paljon, sillä data on helposti saatavilla. Tyypillisimmin sentimentit jaetaan kolmeen kategoriaan, jotka ovat positiiviset, neutraalit ja negatiiviset. Joskus käytetään myös jaottelua viiteen kategoriaan, jolloin edellisten kategorioiden lisäksi käytetään kategorioita erittäin positiiviset ja erittäin negatiiviset. [20]

Sentimenttianalyysin yhteydessä puhutaan myös *polariteetista*, jolla tarkoitetaan tunteen suuntautumista. Osa käytössä olevista kirjastoista analysoi polariteetin syötetystä luonnollisen kielen lauseesta, ja siten polariteetin arvon perusteella sentimentit voidaan luokitella. Toinen sentimenttianalyysin yhteydessä käytettävä käsite on *subjektiivisuus.* Subjektiivisuudella tarkoitetaan sitä, miten tunnepohjainen analysoitava teksti on. Subjektiivisuuden arvo on pieni, mikäli analysoitava teksti on objektiivinen eli faktoihin pohjautuva. Esimerkiksi tämän tutkielman kokeellisessa osassa twiittien sentimenttien analysointiin käytettävä *TextBlob* -kirjasto palauttaa polariteetin ja subjektiivisuuden arvot. [21]

3.1.3 Käytössä olevat menetelmät

Nykyään on olemassa useita erilaisia valmiita ja ilmaisia kirjastoja luonnollisen kielen prosessointiin. Alakohdassa 3.2.1 mainittu TextBlob -kirjasto on tällainen kirjasto, jonka avulla voi tehdä muun muassa sentimenttianalyyseja, sanojen kategorisointia, osittaista puheentunnistusta ja kääntämistä [21]. Toinen esimerkki käytettävissä olevista kirjastoista on *NLTK* -kirjasto, joka on alusta ihmisten tuottaman englanninkielisen luonnollisen kielen kanssa työskentelyyn. NLTK -kirjastosta löytyy lukuisia ominaisuuksia, joita hyödyntämällä voi rakentaa luonnollista kieltä prosessoivia tai hyödyntäviä sovelluksia. [22] TextBlob ja NLTK -kirjastot esitellään tarkemmin kohdassa 5.3.

Luonnollisen kielen prosessointiin voidaan käyttää myös *neuroverkkoja*. *Konvoluutio neuroverkkoja* voidaan käyttää muun muassa sentimenttianalyysien tekemiseen, kääntämiseen ja sarkasmin havaitsemiseen. Takaisinkytkeytyviä neuroverkkoja voidaan puolestaan käyttää esimerkiksi sanojen kategorisointiin sekä luonnollisen kielen generointiin. [23] Neuroverkoista kerrotaan tarkemmin luvussa 4.

3.2 Tietojen yhdistäminen aiemmissa tutkimuksissa

Bitcoiniin liittyvissä aiemmissa tutkimuksissa rakenteellisen ja rakenteettoman tiedon yhdistäminen oli hoidettu ainakin kahdella eri tavalla. Aiemmissa tutkimuksissa yleisempi tapa yhdistää rakenteellinen ja rakenteeton tieto oli ollut ensin prosessoida tai luokitella luonnollinen kieli jollain koneellisella menetelmällä, ja sen jälkeen yhdistää tulokset rakenteelliseen tietoon ikään kuin lisäämällä sinne sarake tai kenttä. Bitcoiniin liittyvissä aiemmissa tutkimuksissa tämä oli käytännössä tarkoittanut sitä, että twiiteistä oli tehty esimerkiksi sentimenttianalyysi, jonka tulokset oli sitten yhdistetty rakenteelliseen tietoon uudeksi sarakkeeksi. Tässä tapauksessa sentimenttianalyysi oli tehty twiiteistä vastaavalta ajanjaksolta, joka oli käytössä rakenteellisen tiedon osalta eli esimerkiksi ajanjaksona tunti tai vuorokausi. Yksi esimerkki tästä tavasta yhdistää tiedot on jo alakohdassa 2.3 esitelty Pant & All. tekemä tutkimus, jossa twiiteistä saadut vuorokausikohtaiset sentimenttiosuudet syötettiin normalisoitujen bitcoinin historiatietojen kanssa takaisinkytkeytyvään neuroverkkoon, joka sitten ennusti numeerista bitcoinin arvoa [15]. Toinen esimerkki tästä tavasta yhdistää tiedot on myös

alakohdassa 2.3 esitelty Colianni & All. tekemä tutkimus, jossa ensin luokiteltiin twiittejä useilla erilaisilla luokittelijoilla ja sitten yhdistettiin twiiteistä saadut tiedot bitcoinin tuntikohtaisiin historiatietoihin [18]. Tämän tutkielman kokeellisessa osassa tullaan käyttämään tätä yleisempää tietojen yhdistämiseen käytettävää tapaa, eli twiiteistä tehdään ensin vuorokausikohtaiset sentimenttianalyysit ja niiden tulokset yhdistetään sitten rakenteelliseen tietoon.

Toinen bitcoiniin liittyvissä aiemmissa tutkimuksissa esiintynyt, mutta vähän harvinaisempi tapa oli yhdistää jokaiseen yksittäiseen twiittiin ja/tai uutisotsikkoon historiatietoja muutaman edellisen päivän osalta, ja siten ennustaa bitcoinin arvoa tällaisien tietokokonaisuuksien perusteella. Yksi esimerkki tästä tavasta yhdistää tiedot on jo alakohdassa 2.3 esitelty Lamon & All. tekemä tutkimus, jossa päivittäiset hintatiedot yhdistettiin kuudella arvolla jokaisen uutisartikkelin otsikkoon ja kahdella arvolla jokaiseen twiittiin. Uutisartikkeleiden otsikoihin yhdistettiin jokaisen tutkimuksessa käytetyn kryptovaluutan arvon suunta kahden seuraavan päivän osalta eli yhteensä kuusi arvoa, ja jokaiseen twiittiin yhdistettiin kyseiseen twiittiin liittyvän kryptovaluutan arvon suunta kahden seuraavan päivän osalta. Molemmissa tapauksissa arvot olivat joko 0 eli kryptovaluutan arvon lasku tai 1 eli kryptovaluutan arvon nousu [17]. Tällä menetelmällä voidaan kuitenkin ennustaa bitcoinin arvolle pelkästään suuntaa, kuten tässäkin tutkimuksessa oli tehty. Eräs vaihtoehto voisi kuitenkin olla yhdistää tarkat bitcoinin arvot uutisotsikoihin ja twiitteihin, ja sen avulla saada myös arvon suuruutta ennustettua. Tutkielman kokeelliseen osaan käytössä oleva rajallinen aika ei kuitenkaan mahdollista kaikkien sivupolkujen kokeilemista ja tutkimista, joten tätä vaihtoehtoa ei tulla tutkimaan.

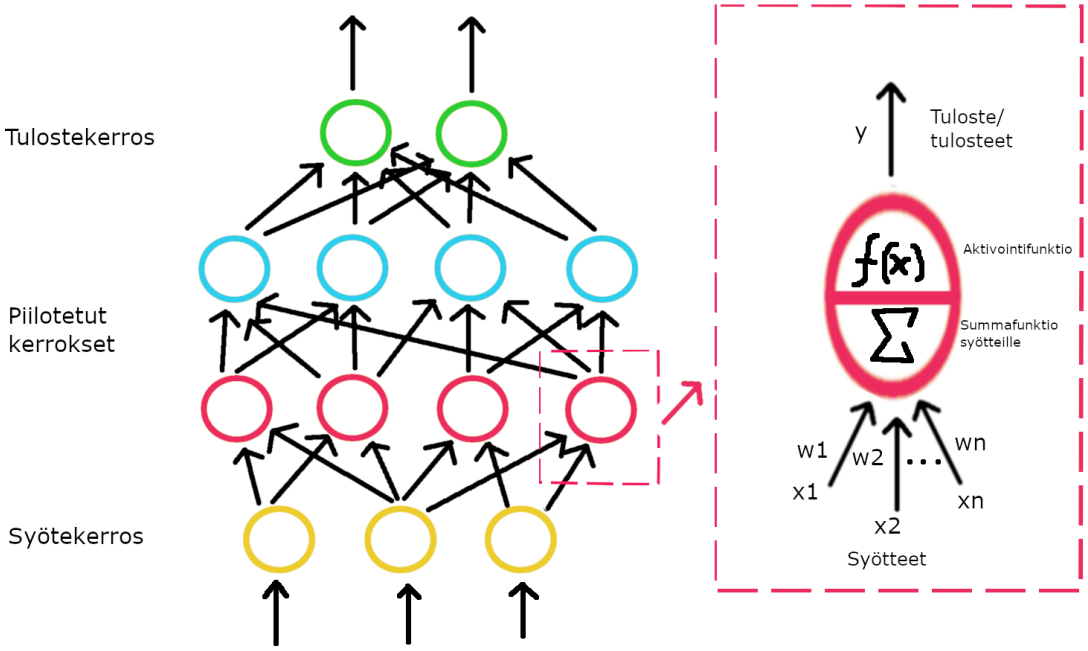
4 Neuroverkot

Neuroverkko on koneellinen työkalu datan käsittelyyn, ja sitä koulutetaan haluttuun tehtävään, kuten ennusteiden ja johtopäätösten tekemiseen. Hyötyinä neuroverkoissa on datan tehokas käsittely, ja sellaisten yhteyksien löytäminen, mikä muilla menetelmillä on huomattavasti hitaampaa tai jopa mahdotonta. [24, pp. 253-255] Neuroverkkoja hyödynnetään nykyään todella monilla aloilla, joista muutamia esimerkkejä ovat kuvantunnistus (esimerkiksi Googlen käänteinen kuvanhakutoiminto [25]), puheentunnistus (esimerkiksi kuluttajien jokapäiväisessä käytössä oleva Applen Siri [26]) sekä tässäkin tutkielmassa käsiteltävä bitcoinin arvon ennustaminen.

Kohdassa 4.1 esitellään *neuronien* ja neuroverkon rakenne, sekä kerrotaan neuroverkkojen erilaisista säädeltävissä olevista parametreista. Kohdassa 4.2 esitellään eri tyyppisiä neuroverkkoja, joista syvällisimmin kerrotaan LSTM -neuroverkosta, jota käytetään myös tutkielmani kokeellisessa osassa. Kohdassa 4.3 kerrotaan neuroverkkojen kouluttamisesta ja saatujen mallien testaamisesta.

4.1 Neuroverkon rakenne

Neuroverkko on matemaattinen malli, joka jäljittelee ihmisen aivojen toimintatapaa ja rakennetta. Neuroverkko koostuu useista keinotekoisista neuroneista, joiden tarkoitus on vastata aivojen neuroneja. Kuten aivojen neuronit, myös keinotekoisen neuroverkon neuronit on kytketty yhteen verkoksi suunnatuilla ja painotetuilla yhteyksillä. Neuroverkko koostuu useista kerroksista, joissa yksittäiset neuronit sijaitsevat. Neuroverkossa on vähintäänkin *syötekerros* sekä *tulostekerros*, mutta niiden lisäksi verkko voi sisältää 1-n (n=positiivinen kokonaisluku) *piilotettua kerrosta* datan käsittelyyn. [24, pp. 254-255, 261] Kuvan 2 vasemmalla puolella on esitetty pelkistetyn neuroverkon rakenne. Kuva 2 pohjautuu Géron (2017) kirjassaan esittelemään teoriaan neuroverkoista [24, pp. 257-262]. Erilaisista neuroverkkotyypeistä ja niiden käyttötarkoituksista kerrotaan lisää kohdassa 4.2.



Kuva 2. Neuroverkon ja neuronin rakenne.

Yksittäisissä neuroneissa tapahtuu neuroverkon datan käsittely. Keinotekoisen neuronin rakenne on esitetty yksinkertaistettuna kuvassa 2 oikealla puolella. Neuroniin tulee siis 1-n (n=positiivinen kokonaisluku) painotettua syötettä, joista neuronin sisällä lasketaan ensin painotettu summa. Tämä painotettu summa syötetään neuronin sisällä olevaan *aktivointifunktioon*, joka tuloksen perusteella määrittelee, että lähetetäänkö tulostetta eteenpäin. [24, pp. 257-259]

Neuroverkoissa on monia erilaisia parametreja, joita säätelemällä pyritään saavuttamaan parhaita mahdollisia tuloksia valittuun käyttötapaukseen. Näitä parametreista yleisimmin säädeltyjä ovat neuroverkon piilotettujen kerrosten lukumäärä, kerroksen neuronien lukumäärä ja aktivointifunktio. Myös muita säädeltäviä parametreja on lukuisia, mutta kaikkien erilaisten kombinaatioiden kokeileminen on työlästä ja aikaa vievää. Yleisesti ottaen suositeltavaa on aloittaa yhdellä tai kahdella piilotetulla kerroksella, ja sen jälkeen lisätä kerroksia tarpeen mukaan. Joissain tapauksissa, kuten puheentunnistusteknologiassa kuitenkaan muutama piilotettu kerros ei riitä, sillä hyvien tuloksien saavuttamiseen voidaan tarvita kymmeniä tai jopa satoja piilotettuja kerroksia. Syöte- ja tulostekerrosten neuronien lukumäärä määräytyy käyttötapauksen, eli syötettävien ja tulostettavien muuttujien lukumäärän mukaan. Piilotettujen kerrosten optimaalinen neuronien lukumäärä ei olekaan yhtä yksiselitteinen. Kaksi yleisintä tapaa rakentaa piilotetut kerrokset ovat joko laittaa jokaiselle piilotetulle kerrokselle

yhtä monta neuronia tai vaihtoehtoisesti vähentää neuronien lukumäärää edelliseen piilotettuun kerrokseen verrattuna. Optimaalinen kerrosten ja neuronien lukumäärä löytyy kokeilemalla, mutta yleisistä käytännöistä aloittaminen säästää huomattavasti aikaa ja resursseja. [24, pp. 270-272] Neuronien aktivointifunktioita on lukuisia erilaisia, mutta muutamia yleisesti käytettyjä aktivointifunktioita ovat *Sigmoid,* ja *ReLU* [27, pp. 7-9] [24, p. 272]. Sigmoid -funktio määritellään kaavalla:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Kaavassa e tarkoittaa *Neperin lukua*, joka on luonnollisen logaritmifunktion kantaluku. Kun x:n arvo pienenee, niin y:n arvo lähestyy nollaa ja kun x:n arvo suurenee niin y:n arvo lähestyy lukua yksi. Sigmoid -funktiosta saatavat arvot siis vaihtelevat välillä nolla ja yksi. [27, p. 8]

ReLU -funktio puolestaan määritellään seuraavasti:

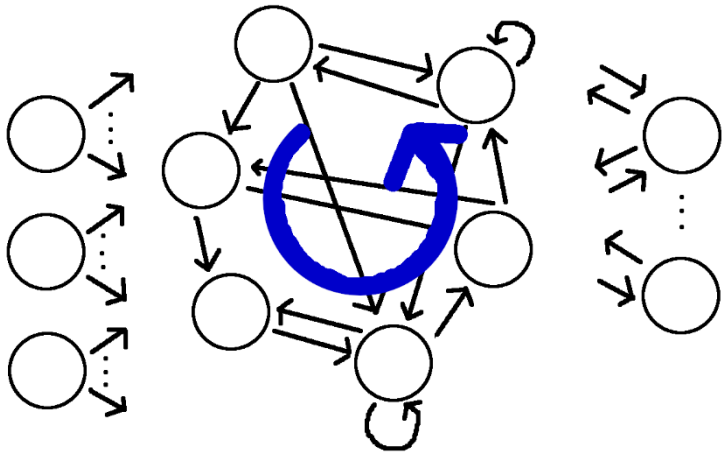
$$R(x) = \max(0, x)$$

Tämä tarkoittaa sitä, että ReLU -funktiossa y:n arvo on aina vähintään nolla. Kun x:n arvot ovat yli nollan, niin ReLU -funktio käyttäytyy vastaavalla tavalla kuin identiteettifunktio. Tällöin y:n arvo on sama kuin x:n arvo. [28]

4.2 Neuroverkkotyypit

Neuroverkkoja on erityyppisiä, joista ennustetehtäviin yleisimmin käytettyjä ovat *monikerroksiset verkot*, konvoluutioneuroverkot ja takaisinkytkeytyvät neuroverkot erillisine sovelluksineen. Monikerroksisissa verkoissa käytetään useampaa kuin yhtä piilotettua kerrosta ja ne tunnetaan myös eteenpäin syötettävinä verkkoina. Konvoluutioneuroverkot sisältävät konvoluutiokerroksia, jotka suorittavat konvoluutiot halutun suodattimen ja syöttödatan välillä. Konvoluutioneuroverkot ovat tehokkaita oppimaan hierarkkisia ominaisuuksia datasta ja ne ovat yleisesti käytettyjä erityisesti kuva-analytiikassa, puheentunnistuksessa sekä luonnollisen kielen käsittelyssä. [29] [24, pp. 353-357]

Takaisinkytkeytyvissä neuroverkoissa piilotettujen kerrosten neuroneilla on yhteyksiä takaisin itseensä. Lisäksi joillain takaisinkytkeytyvien neuroverkkojen sovelluksilla yksittäisen neuronin sisällä on *muistisolu*, joka mahdollistaa sen, että neuronilla on muistia. LSTM- verkko on esimerkki tällaisesta takaisinkytkeytyvästä neuroverkosta, jonka yksittäisissä neuroneissa on muistisolut. Takaisinkytkeytyvät neuroverkot ovat erittäin hyödyllisiä esimerkiksi tekstinkäsittelyssä, koska sanoja voidaan yhdistää kontekstiin ja viereisiin sanoihin. Yleisesti ottaen neuroverkoissa aika on muuttujana haasteellinen, koska se liikkuu vakionopeudella eteenpäin. Takaisinkytkeytyvien neuroverkkojen rakenteen ja yksittäisissä neuroneissa sijaitsevien muistisolujen ansiosta aikasarjaongelmia saadaan kuitenkin tehokkaasti ratkaistua, ja siksi toinen yleinen käyttötapaus takaisinkytkeytyville neuroverkoille onkin erilaisten aikasarjatietojen ennustaminen. Takaisinkytkeytyvän neuroverkon rakenne on esitetty pelkistetysti kuvassa 3, joka on osa yhdestä Jaokar & All. (2015) julkaiseman artikkelin kuvasta. Takaisinkytkeytyvässä neuroverkossa on samaan tapaan syötekerros, piilotetut kerrokset sekä tulostekerros kuin kaikissa neuroverkkotyypeissä. Näiden lisäksi takaisinkytkeytyvissä neuroverkoissa piilotettujen kerroksien neuroneilla on silmukoita sekä yhteyksiä takaisin itseensä. Lisäksi myös tulostekerroksen neuroneilla voi olla yhteyksiä takaisin piilotettuihin kerroksiin, ja siten voidaan saavuttaa vielä parempia tuloksia. [30]



Kuva 3. Takaisinkytkeytyvän neuroverkon rakenne. Kuva on osa yhdestä Jaokar & All. julkaiseman artikkelin kuvasta [30].

Takaisinkytkeytyvistä neuroverkoista on olemassa useita erikoistapauksia, joista yksi tällainen on LSTM -verkko (Long-Short Term Memory Network), jossa muistisolu on sisällytetty kuhunkin piilokerroksen neuroniin. LSTM -verkon vahvuutena on, että se pystyy oppimaan riippuvuuksia pitkältäkin aikaväliltä. [24, pp. 400-402]

4.3 Neuroverkon kouluttaminen ja mallin testaaminen

Neuroverkon kouluttamiseen käytettävät yleisimmät menetelmät voidaan jakaa kolmeen pääkategoriaan: *ohjattuun oppimiseen, ohjaamattomaan oppimiseen* sekä *vahvistusoppimiseen*. Ohjatussa oppimisessa lähtödata sisältää halutut lopputulokset, kun ohjaamattomassa oppimisessa puolestaan lähtödatasta on poistettu lopputulokset. Tyypillisesti ohjattua oppimista käytetään asioiden luokittelemiseen ja numeeristen arvojen ennustamiseen. Ohjaamatonta oppimista käytetään esimerkiksi poikkeamien havaitsemiseen sekä assosiaatiosääntöjen oppimiseen. Vahvistusoppiminen on hyvin erilainen menetelmä, sillä siinä oppiminen tapahtuu itsenäisesti, kun malli ja ympäristö vuorovaikuttavat jatkuvasti keskenään. Vahvistusoppimista käytetään tyypillisimmin robotiikassa. [24, pp. 8-14] [31, pp. 7-9] Tässä tutkielmassa käsitellään näistä menetelmistä tarkemmin pelkästään ohjattua oppimista, sillä sitä hyödynnetään neuroverkon kouluttamisessa tutkielman kokeellisessa osassa.

Neuroverkko koulutetaan haluttuun tehtävään syöttämällä syötekerrokselle ennestään tiedossa olevat attribuutit, eli *koulutusdata*. Ohjattua oppimista hyödynnettäessä, ennen neuroverkon kouluttamista lähtödata jaetaan joko kahteen tai kolmeen osaan siten, että jokaisesta attribuutista on yhtä suuri prosentuaalinen osuus jokaisessa osassa. Kahteen osaan jaettaessa lähtödatasta 70-80% käytetään neuroverkon kouluttamiseen ja 20-30% verkon testaamiseen. Mikäli lähtödata jaetaan kolmeen osaan, silloin lähtödatasta 50-60% käytetään neuroverkon kouluttamiseen, noin 30% verkon testaamiseen ja loput 10-20% tuloksien validointiin. [27, pp. 17-19]

Neuroverkkojen kouluttamiseen käytetään erilaisia *optimointialgoritmeja* riippuen neuroverkon topologiasta ja käyttötarkoituksesta. Tämän tutkielman kokeellisessa osassa optimointialgoritmina käytetään Adam -algoritmia. Adam -algoritmi on mukautuva oppimismenetelmä, toisin sanoen se laskee eri parametreille yksilölliset asteet oppimista varten [32]. Adam -algoritmi valikoitui käytettäväksi optimointialgoritmina tämän tutkielman kokeellisessa osassa, koska se vaikutti yleisesti käytetyltä ja etukäteen mahdollisesti toimivalta algoritmilta. Optimointialgoritmin valintaan liittyen ei tehty varsinaisia omia taustatutkimuksia. Rajallisen käytössä olevan ajan vuoksi tässä tutkielmassa ei tarkemmin perehdytä optimointialgoritmeihin, eikä kokeellisessa osassa kokeilla erilaisia optimointialgoritmeja.

Ohjatussa oppimisessa neuroverkon kouluttamista eli yksittäisten neuronien painon säätelemistä jatketaan, kunnes laskennalliselle virheelle asetettu maksimi alittuu [27, pp. 19-20]. Tämän tutkielman kokeellisessa osassa virheen laskentakaavana käytetään keskimääräistä neliövirhettä eli MSE (Mean Square Error), joka määritellään kaavalla [33]:

$$\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2$$

Kaavaa voidaan ajatella koordinaatistossa, jolloin n on ikään kuin pisteiden lukumäärä eli neuroverkkojen tapauksessa se on koulutukseen käytettävien rivien tai osien lukumäärä. y_i puolestaan tarkoittaa pisteen i y-koordinaattia eli tässä tapauksessa todellista bitcoinin arvoa, jota pyritään ennustamaan. $f(x_i)$ tarkoittaa funktion arvoa pisteen i x-koordinaatilla eli tässä tapauksessa ennustettua bitcoinin arvoa. Keskimääräinen neliövirhe lasketaan siis summaamalla kaikkien ennusteiden ja todellisten arvojen erotuksen neliö, ja lopuksi jakamalla tämä tapausten lukumäärällä. Neuroverkkojen tapauksessa tavoitteena on minimoida tämä keskimääräisen neliövirheen tuottama arvo, koska silloin kaavan tuottama kuvaaja kulkee mahdollisimman läheltä kaikkia pisteitä. [33] Tämän tutkielman kokeellisessa osassa myöskään erilaisia virheen laskentakaavoja ei kokeilla rajallisen käytettävissä olevan ajan vuoksi.

Varsinaisen kouluttamisen jälkeen suoritetaan tuotetun mallin testaus sellaisella datalla, jota neuroverkko ei ole aiemmin käsitellyt. Neuroverkon ennustamia tai antamia tuloksia verrataan todellisiin tuloksiin, joita neuroverkko ei ole koskaan nähnytkään. Tyypillisesti neuroverkon mallin arviointiin käytetään *sekaannusmatriisia* (confusion matrix), jonka rivit muodostuvat ennustetuista luokista ja sarakkeet todellisista luokista. [24, pp. 84-87] [34, pp. 123-124] Taulukossa 1 esitetään binäärisen luokittelijan sekaannusmatriisi [34, pp. 123-124]:

Taulukko 1. Sekaannusmatriisi [34, pp. 123-124]**.**

		Todelliset luokat	
		Kuuluu luokkaan	Ei kuulu luokkaan
Ennustetut luokat	Kuuluu luokkaan	TP	FP
	Ei kuulu luokkaan	FN	TN

Taulukossa 1 esimerkiksi *TP* (true positive) tarkoittaa niitä arvoja, jotka malli on luokitellut kuulumaan tiettyyn luokkaa, ja jotka myös todellisuudessa kuuluvat siihen luokkaan. *FP* (false positive), *FN* (false negative) ja *TN* (ture negative) määritellään vastaavalla tavalla. Sekaannusmatriisista saadaan laskettua lukuisia erilaisia koulutetusta mallista kertovia arvoja, joista yksi yleisimmin käytetyistä on tarkkuus (accuracy). Tarkkuus kertoo, kuinka suurella todennäköisyydellä malli antaa oikean tuloksen. [34, pp. 123-124] Tarkkuus saadaan laskettua sekaannusmatriisista kaavalla [34, p. 124]:

$$\frac{TP + TN}{TP + FP + FN + TN}$$

Tämän tutkielman kokeellisessa osassa neuroverkkojen ennustetarkkuus arvioidaan käyttäen edellistä kaavaa.

5 Tutkimuksen toteutus

Kokeellisessa osassa tutkitaan, millä rakenteellisten ja rakenteettomien tietojen yhdistelmillä saavutetaan parhaat tulokset bitcoinin arvon ennustamisessa seuraavalle vuorokaudelle. Kokeellisessa osassa tullaan käyttämään viittä erilaista LSTM -neuroverkon mallia ja muutamia erilaisia yhdistelmiä eri parametreista. Rakenteellista sekä rakenteetonta tietoa hankitaan aikaväliltä 1.6.2018 - 10.2.2019, eli reilun kahdeksan kuukauden ajalta. Aikaväli valikoitui käytettäväksi tutkimuksessa, sillä sen pitäisi olla neuroverkon opettamista varten riittävä ja tällä aikavälillä on tapahtunut bitcoinin arvon heilahduksia molempiin suuntiin.

Kohdassa 5.1 kerrotaan rakenteellisen tiedon eli bitcoinin historiatietojen sekä bitcoiniin liittyvien twiittien vuorokausikohtaisten volyymitietojen hankkimisesta. Kohdassa 5.2 puolestaan kerrotaan rakenteettoman tiedon eli twiittien hankkimisesta ja siihen liittyen muun muassa hankittavien twiittien rajauksesta sekä rajausta varten tehdystä taustatutkimuksesta. Kohta 5.3 käsittelee twiittien prosessointia ennen sentimenttianalyysiä, varsinaisen sentimenttianalyysin tekemistä sekä sentimenttianalyysin tuloksia. Rakenteellisten ja rakenteettomien tietojen yhdistämistä käsitellään kohdassa 5.4. Kohdassa 5.5 esitellään tietojen syöttämistä neuroverkkoon, käytettäviä LSTM -malleja sekä saatujen tuloksien analysointitapaa.

5.1 Rakenteellisen tiedon hankkiminen

Tämän tutkielman kokeellista osaa varten hankitaan historiatietoja, jotka ovat bitcoinin historiatietoja sekä bitcoiniin liittyvien twiittien vuorokausikohtaisia volyymitietoja. Alakohdassa 5.1.1 kerrotaan bitcoinin historiatietojen hankkimisesta ja alakohdassa 5.1.2 kerrotaan bitcoiniin liittyvien twiittien vuorokausikohtaisen volyymin hankkimisesta.

5.1.1 Bitcoinin historiatietojen hankkiminen

Bitcoinin historiatietoja ladataan *Bitfinex:*in avoimen lähdekoodin *REST*-rajapinnasta, josta voi ladata kryptovaluuttojen historiatietoja ilman palveluun kirjautumista tai erillistä latausavainta. Palvelusta ladattaessa määritellään kryptovaluutta, yksikkö,

ajanjakso ja hakujen määrä. [7] Tämän tutkielman kokeellisen osan tapauksessa haetaan siis bitcoinin arvoa Yhdysvaltain dollarilla, ajanjaksoksi on asetettu yksi päivä ja hakujen määrä on 2190 eli vuoden 2013 alusta alkaen. Tämän tutkielman kokeellista osaa varten aikavälin ei olisi tarvinnut olla näin pitkä, mutta bitcoinin historiatietoja haluttiin ladata pidemmältä aikaväliltä, jotta bitcoinin hintakehityksestä saadaan parempi kokonaiskuva. Ajanjaksoksi on asetettu yksi päivä, koska tutkimuksessa pyritään ennustamaan bitcoinin arvon suuntaa seuraavalle vuorokaudelle.

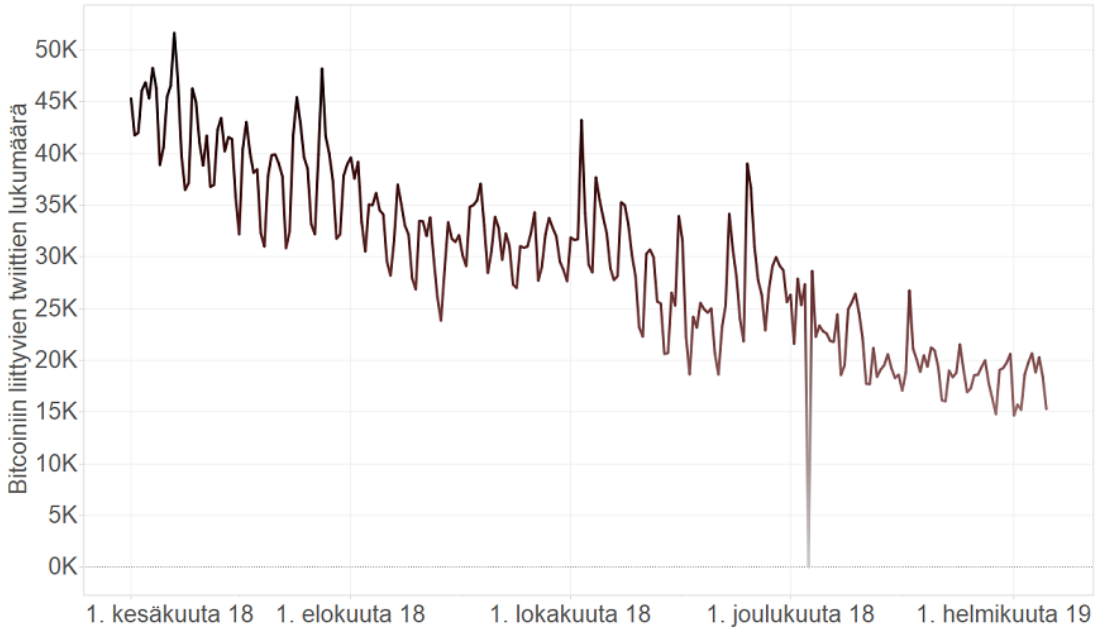
Bitfinex:in avoimesta rajapinnasta saadaan historiadata *JSON*-formaatissa, josta se parsitaan käyttökelpoiseen muotoon. Tässä tapauksessa jokaisen vuorokauden osalta saadaan bitcoinin korkein ja matalin arvo, arvo pörssin auetessa ja pörssin sulkeutuessa sekä kaupankäynnin volyymi. [7] Näistä tutkimuksessa neuroverkkoon tullaan syöttämään bitcoinin korkein arvo sekä kaupankäynnin volyymi. Bitcoiniin liittyviä historiatietoja on esitetty graafisessa muodossa kohdassa 2.2.

5.1.2 Bitcoiniin liittyvien twiittien vuorokausikohtaisten volyymitietojen hankkiminen

Bitcoiniin liittyvien twiittien vuorokausikohtaisten volyymitietojen hankkiminen osoittautui melko haasteelliseksi tehtäväksi, kun tiedot haluttiin saada ilmaiseksi. Eräs vaihtoehto olisi ollut ladata kaikki mahdolliset twiitit jokaiselta päivältä, mutta niiden määrä olisi ollut liian suuri ja siihen olisi kulunut paljon aikaa. Saatavilla on kuitenkin useita erilaisia palveluja, joista pystyy näkemään graafisessa muodossa valitsemansa aihealueen tai hakusanan perusteella twiittien vuorokausikohtaiset volyymit. Tämän tutkielman kokeellista osaa varten bitcoiniin liittyvien twiittien vuorokausikohtaiset volyymitiedot otettiin *BitInfoCharts*-sivustolta (https://bitinfocharts.com/) [35]. Bitcoiniin liittyvien twiittien vuorokausikohtaisia volyymitietoja hankittiin siis aikaväliltä 1.6.2018 - 10.2.2019, sillä aikaväli valittiin aiemmin käytettäväksi tämän tutkielman kokeellisessa osassa.

Kuvassa 4 on esitettynä Tableaulla tehty visualisointi bitcoiniin liittyvien twiittien vuorokausikohtaisesta volyymista. Visualisoinnissa vaaka-akselilla on aika ja pystyakselilla on bitcoiniin liittyvien twiittien lukumäärä. Aikaväliltä 1.6.2018 – 10.2.2019 puuttui yksi arvo, ja siksi visualisoinnissa yhden vuorokauden osalta bitcoiniin liittyvien twiittien vuorokausikohtainen arvo on nolla. Kokonaisuudessaan tällä aikavälillä

bitcoiniin liittyvien twiittien vuorokausikohtainen lukumäärä oli keskimäärin laskeva. Vaihtelu vuorokausien välillä bitcoiniin liittyvien twiittien lukumäärässä oli suhteellisen suurta, sillä tällä aikavälillä vaihtelu oli 14 658 – 51 654 twiitin välillä. Myös peräkkäisien vuorokausien osalta bitcoiniin liittyvien twiittien lukumäärän vaihtelut olivat jopa lähes 20 000 twiittiä. Keskimäärin tällä aikavälillä bitcoiniin liittyvien twiittien vuorokausikohtainen lukumäärä oli 29 793. Eräs mielenkiintoinen huomio on, että twiittien lukumäärä on keskimäärin melko lineaarisesti laskenut tällä tarkastelujaksolla, kun samaan aikaa myös bitcoinin arvo on ollut keskimäärin laskusuunnassa, vaikka bitcoinin arvo onkin vaihdellut molempiin suuntiin.



Kuva 4. Bitcoiniin liittyvien twiittien vuorokausikohtainen volyymi.

5.2 Rakenteettoman tiedon hankkiminen

Rakenteetonta tietoa eli twiittejä ladataan hyödyntäen *GetOldTweets3* -kirjastoa, joka pohjautuu Jefferson Henriquenin GetOldTweets -python kirjastoon [36]. Twitterin omasta rajapinnasta twiittien lataaminen ilmaiseksi osoittautui erittäin rajoitetuksi sekä määrällisesti että ajallisesti, mutta GetOldTweets3 -kirjastoa hyödyntämällä pystytään ikään kuin ohittamaan nämä rajoitukset. GetOldTweets3 -kirjaston toimista perustuu Twitterin oman selaimessa toimivan hakutoiminnon käyttämiseen, sillä twiittejä latautuu automaattisesti lisää, kun hakutoiminnon tulossivulla liikutaan alaspäin.

Joten tämän kirjaston avulla vanhojakin twiittejä voidaan hakea käytännössä rajattomasti, ja kaikki twiitit saadaan ladattua helposti käsiteltävässä JSON-formaatissa.

GetOldTweets3 -kirjastolla voi hakea twiittejä käyttäjänimien mukaan ja/tai hakusanan mukaan. Lisäksi hauille voi asettaa rajoituksia liittyen alku- ja loppupäivämäärään, sijaintiin, etäisyyteen sijainnin mukaan, twiittien suosioon, twiittien kirjoituskieleen sekä ladattavien twiittien lukumäärään. Twiitit siis palautetaan JSON-formaatissa ja jokaisesta twiitistä on saatavilla tiedot twiitin yksilöivästä numerosta, julkaisijan käyttäjätunnuksesta, julkaisun kohteesta, twiitin tekstistä, twiitin tekstin kirjoituskielestä, päivämäärästä *UTC*-muodossa, uudelleentwiittauksista, suosikeista, maininnoista, hashtageista sekä julkaisun sijainnista. [36]

Kokeellista osaa varten ladattavien twiittien osalta tehtiin taustatutkimusta, jonka perusteella päätettiin rajaus ladattaville twiiteille. Tästä taustatutkimuksesta kerrotaan alakohdassa 5.2.1 ja twiittien lataukseen käytettävästä rajauksesta kerrotaan alakohdassa 5.2.2.

5.2.1 Taustatutkimus twiitteihin liittyen

Kuten alakohdassa 5.1.2 esitettiin, bitcoiniin liittyviä twiittejä on twiitattu todella suuri määrä vuorokausittain ja siksi on tehty pientä taustatutkimusta siitä, että minkälaisilla hakuehdoilla niitä kannattaisi mahdollisesti ladata ja minkälaisia määriä vuorokautta kohti. Tässä taustatutkimuksessa hyödynnettiin GetOldTweets3 -kirjastoa twiittien lataamiseen ja TextBlob -kirjastoa sentimenttianalyysin tekemiseen [36] [21]. TextBlob -kirjastosta ja sen käyttämisestä kerrotaan tarkemmin kohdassa 5.3. Taustatutkimuksen ensimmäisessä osassa otettiin tarkasteluun 9.2.2019 twiitatut bitcoiniin liittyvät twiitit, jotka oli kirjoitettu englanniksi. Taulukossa 2 on esitetty taustatutkimuksen ensimmäisen osan tulokset.

Taulukko 2. Bitcoiniin liittyvien twiittien analyysi päivältä 9.2.2019

	Neutraalit	Positiiviset	Negatiiviset
kaikki twiitit	43.42%	43.10%	13.48%
500 ensimmäistä	42.60%	46.60%	10.80%
1000 ensimmäistä	40.30%	47.70%	12.00%
suosituimmat (234)	35.47%	51.71%	12.82%

Taulukossa 2 esitetään neljällä erilaisella hakuehdolla toteutetun sentimenttianalyysin tulokset, kun sentimenttianalyysin tulokset on jaettu karkeasti kolmeen luokkaan eli neutraaleihin, positiivisiin ja negatiivisiin. Taustatutkimuksen ensimmäisessä osassa otettiin tarkasteluun kaikki 9.2.2019 twiitatut bitcoiniin liittyvät twiitit (18 400), 500 ensimmäistä twiittiä, 1000 ensimmäistä twiittiä sekä suosituimmat twiitit, joita oli tämän vuorokauden tapauksessa 234. Suosituimpien twiittien lukumäärä vaihtelee vuorokausittain, ja tämä pohjautuu Twitterin tapaan luokitella twiitit [37]. Vuorokauden kaikkien twiittien, 500 ensimmäisen twiitin ja 1000 ensimmäisen twiitin välillä ei ollut kovinkaan suuria eroja sentimenttianalyysissä, sillä vaihteluväli oli neutraalien twiittien osalta 40.30% - 43.42%, positiivisten twiittien osalta 43.10% - 47.70% sekä negatiivisten twiittien osalta 10.80% - 13.48%. Prosentuaaliset erot näissä olivat siis todella pieniä. Vuorokauden suosituimpien twiittien osalta sentimenttianalyysi erosi puolestaan huomattavasti kolmen muun taustatutkimukseen valitun twiittijoukon sentimenttianalyysistä, sillä siinä neutraalien twiittien osuus oli vain 35.47%, positiivisten jopa 51.71% ja negatiivisten oli 12.82%.

Taustatutkimuksen ensimmäisen osan perusteella päädyttiin tutkimaan, että minkälaisia eroja vuorokauden 500 ensimmäisen twiitin ja vuorokauden suosituimpien twiittien välillä on, kun tarkasteluun otetaan kolme erillistä vuorokautta. Tässä taustatutkimuksen toisessa osassa tarkasteluun valikoituivat vuorokaudet 9.12.2018, 9.1.2019 ja 9.2.2019, sillä haluttiin nähdä miten sentimenttianalyysin tulokset eroavat, kun vuorokaudet eivät ole peräkkäisiä. Taulukossa 3 on esitetty tämän taustatutkimuksen toisen osan tulokset.

Taulukko 3. Sentimenttianalyysit kolmen vuorokauden osalta käyttäen 500 ensimmäistä ja suosituimpia twiittejä.

	käytetyt twiitit	Neutraalit	Positiiviset	Negatiiviset
9.12.2018	500	45.00%	42.20%	12.80%
	suosituimmat (306)	31.37%	49.02%	19.61%
9.1.2019	500	42.60%	42.60%	14.80%
	suosituimmat (380)	32.89%	51.84%	15.26%
9.2.2019	500	42.60%	46.60%	10.80%
	suosituimmat (234)	35.47%	51.71%	12.82%

Taustatutkimuksen toisen osan tuloksista on havaittavissa, että käytettyjen kolmen erillisen vuorokauden 500 ensimmäistä bitcoiniin liittyvää twiittiä ovat sentimenttianalyysiltään hyvin samankaltaiset. 500 ensimmäisen twiitin kohdalla vaihteluväli oli neutraalien twiittien osalta 42.60% - 45.00%, positiivisten twiittien osalta 42.20% – 46.60% ja negatiivisten twiittien osalta 10.80% - 14.80%. Kolmen erillisen vuorokauden suosituimpien twiittien kohdalla vaihteluväli oli puolestaan neutraalien twiittien osalta 31.37% – 35.47%, positiivisten twiittien osalta 49.02% - 51.71% ja negatiivisten twiittien osalta 12.82% - 19.61%. Myöskään suosituimpien twiittien tapauksessa sentimenttianalyysit eivät poikenneet ainakaan tähän taustatutkimukseen valittujen vuorokausien kohdalla erityisen suuresti, mutta sentimenttianalyysin kategorioiden väliset vaihteluvälit olivat kuitenkin vähän suurempia kuin 500 ensimmäisen twiitin tapauksessa. Jos tuloksia tarkastellaan jokaisen valitun vuorokauden osalta erikseen, niin voidaan havaita, että 500 ensimmäisen twiitin ja suosituimpien twiittien sentimenttianalyyseissä on huomattavia eroavaisuuksia.

5.2.2 Ladattavien twiittien rajaus

Alakohdassa 5.2.1 esitetyn taustatutkimuksen pohjalta tehtiin rajaus ladattaville twiiteille. Tutkielman kokeellista osaa varten päädyttiin lataamaan jokaiselta vuorokaudelta kaikki suosituimmat bitcoiniin liittyvät twiitit, sillä taustatutkimuksen perusteella niiden sentimenttianalyysien vaihteluvälit olivat vähän suurempia kuin jokaisen vuorokauden 500 ensimmäisen bitcoiniin liittyvän twiitin sentimenttianalyysien vaihteluvälit. Lisäksi voisi päätellä, että suosituimmat twiitit antaisivat mahdollisesti parempaa suuntaa yleisestä mielipiteestä bitcoinia kohtaan. Suosituimpiin twiitteihin päädyttiin, koska jokin rajaus oli tehtävä ja taustatutkimuksen perusteella saadut sentimenttianalyysien vaihteluvälit valittiin ratkaisevaksi tekijäksi. On hyvinkin mahdollista, että tämä rajaus ei ole mitenkään optimaalinen ja parhaisiin tutkimustuloksiin johtava, mutta käytettävissä olevan ajan ja työmäärän vuoksi jonkinlainen rajaus oli tehtävä. Esimerkiksi kaikkien twiittien lataaminen jokaiselta vuorokaudelta olisi erittäin raskas ja aikaa vievä prosessi, sillä twiittien vuorokausikohtaiset määrät ovat niin suuria. Twiittien lataamiseen käytettävässä GetOldTweets -kirjastossa ei ole toimintoa, jolla pystyisi lataamaan twiittejä satunnaisesti valitun vuorokauden ajalta, ja siten myöskään twiittien satunnainen lataaminen ei ollut mahdollista.

Tutkielman kokeellista osaa varten twiittejä ladataan siis hakusanan mukaan, jonka parametrina käytetään bitcoin-sanaa. Tämän tutkielman kokeellisessa osassa ollaan siis kiinnostuneita erityisesti yleisestä asenteesta bitcoinia kohtaan ja sen vaikutuksesta bitcoinin arvoon, ja siksi twiittien julkaisijaa ei rajata twiittejä ladattaessa. Ladattavien twiittien tekstin kirjoituskieleksi rajoitetaan pelkästään englanti, koska englanninkielisten twiittien myöhempi prosessointi on kaikista helpointa ja yksinkertaisinta. Näiden rajauksien lisäksi jokaiselta vuorokaudelta ladataan vain suosituimmat twiitit, kuten taustatutkimuksen tuloksien perusteella päätettiin.

Twiittien lataamiseen käytetään python-ohjelmaa, joka hakee käytännössä jokaisen vuorokauden hakuehdot täyttävät twiitit erikseen, ja tallentaa ne ennen seuraavaa hakua csv-tiedostoon. Siten jokaisen vuorokauden kaikki hakuehdot täyttävät twiitin saadaan suoraan omaan tiedostoonsa, eikä niitä tarvitse lajitella päivämäärän mukaan enää myöhemmin. Twiittejä ladataan siis aikaväliltä 1.6.2018 - 10.2.2019, sillä aikaväli valittiin aiemmin käytettäväksi tämän tutkielman kokeellisessa osassa.

5.3 Twiittien prosessointi

Tämän tutkielman kokeellisessa osassa twiittien vuorokausikohtaisen sentimenttianalyysin tekemiseen käytetään valmista TextBlob -kirjastoa, joka on python-kirjasto englanninkielisen tekstitiedon prosessointiin. TextBlob -kirjasto tarjoaa sovellusliittymän, jonka avulla luonnollisen kielen prosessointi onnistuu nopeasti ja helposti. Tämän kirjaston avulla voi tehdä muun muassa sentimenttianalyysejä, puheentunnistussovelluksia, luokitteluja sekä käännöksiä. [21] TextBlob -kirjasto hyödyntää sekä NLTK- että pattern.en -kirjastojen toiminnallisuuksia, ja tarjoaa näiden avulla valmiiksi luotuja toiminnallisuuksia luonnollisen kielen prosessointiin. NLTK -kirjasto on johtava alusta ihmisten tuottaman englanninkielisen luonnollisen kielen kanssa työskentelyyn, ja myös se on toteutettu python-kielellä. NLTK -kirjasto tarjoaa toiminnallisuuksia muun muassa tekstin luokitteluun, merkitsemiseen, jäsentämiseen ja semanttiseen päättelyyn. [22] *Pattern.en* -kirjasto puolestaan tarjoaa työkalun muun muassa substantiivien, adjektiivien, verbien jne. tunnistamiseen englanninkielisistä lauseista. Lisäksi myös pattern.en -kirjasto tarjoaa mahdollisuuden sentimenttianalyysiin englanninkielisien lauseiden osalta. [38]

TextBlob -kirjasto valikoitui käytettäväksi tämän tutkielman kokeellisessa osassa, koska se vaikutti helppokäyttöiseltä, nopeasti toimivalta kokonaisuudelta ja ennen kaikkea se oli valmis työkalu. Sentimenttianalyysien tekemiseen on lukuisia erilaisia vaihtoehtoja, mutta tämän tutkielman kokeellisessa osassa aikaa on rajallisesti, eikä sentimenttianalyysien tekemiseen haluttu kuluttaa paljon aikaa. Tulevaisuudessa olisi kuitenkin mielenkiintoista testata erilaisia kirjastoja ja ratkaisuja sentimenttianalyysien tekemiseen, ja siten vertailla niistä saatuja tuloksia. Tämä onkin eräs potentiaalinen jatkotutkimusmahdollisuus.

Ennen varsinaisten sentimenttianalyysien tekemistä twiitit siivotaan, sillä TextBlob -kirjaston sentimenttianalyysistä saadaan parempia ja tarkempia tuloksia siivotuilla teksteillä. *Siivotuilla twiiteillä* tarkoitetaan tässä yhteydessä twiittejä, joista on poistettu ohjelmallisesti erikoismerkit ja ylimääräiset välilyönnit. [21] Kuvassa 5 on esimerkki twiitin siivoamiseen käytettävästä cleantweet -funktiosta python-ohjelmassa.

```
import re
def cleanTweet(tweet):
          cleaned  =  '  '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z
          \t]) | (\w +:\ / \ / \S +)", " ", tweet).split())
return cleaned
```

Kuva 5. Twiittien siivoamiseen käytettävä cleantweet -funktio.

Kuvassa 5 on cleantweet -funktio, jonka avulla twiitit siivotaan yksitellen. cleantweet -funktiolle annetaan syötteenä twiitti alkuperäisessä muodossaan ja lopuksi se palautetaan siivottuna. Varsinainen twiitin siivoaminen hoidetaan *re.sub* -funktiolla, jolle määritellään tekstistä siivottavat merkit [39]. Tässä tapauksessa erikoismerkit korvataan välilyönneillä, ja lopuksi turhat välilyönnit siivotaan vielä hyödyntäen *split* -funktiota [39]. Esimerkki siivoamattomasta twiitistä, joka oli julkaistu 6.6.2018:

“1 year ago today I turned my 6 year passion for #bitcoin and #blockchain into a Youtube channel. Those 250+ videos have now been watched 1.5 million times! Thanks everyone who's helped shape my journey thus far :D“

Ja sama twiitti siivottuna cleantweet -funktiolla:

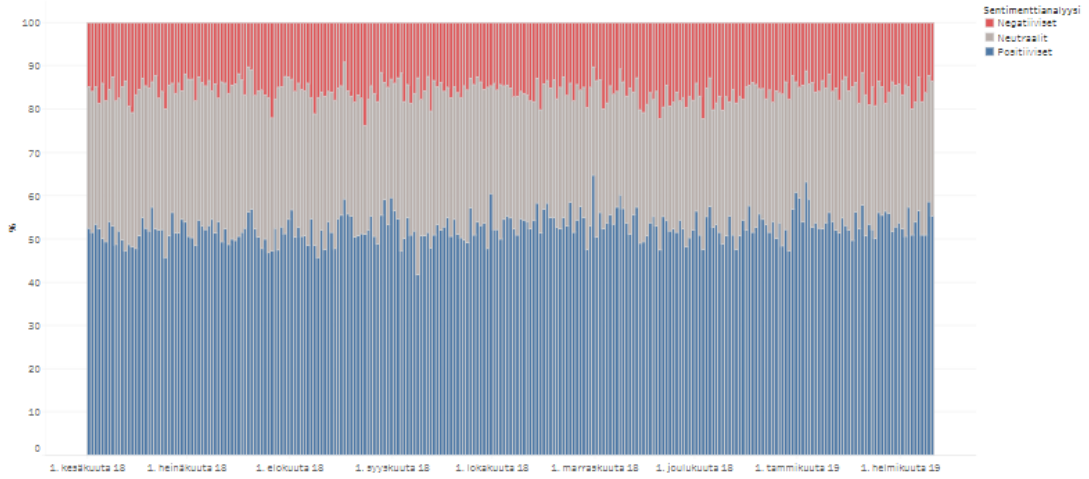
“1 year ago today I turned my 6 year passion for bitcoin and blockchain into a Youtube channel Those 250 videos have now been watched 1.5 million times Thanks everyone who s helped shape my journey thus far”

Twiittien siivoamisen jälkeen jokaisen twiitin sentimentti analysoidaan erikseen TextBlob -kirjaston tarjoamalla toiminnallisuudella. TextBlob -kirjaston *sentiment* -funktio antaa analysoidusta tekstistä tulosteena kaksi erillistä arvoa, jotka ovat polariteetti ja subjektiivisuus. Polariteetin arvo on desimaaliluku väliltä [-1.0, 1.0], jossa -1.0 on erittäin negatiivinen ja 1.0 on erittäin positiivinen. Subjektiivisuus puolestaan on desimaaliluku väliltä [0.0, 1.0], jossa 0.0 on hyvin objektiivinen ja 1.0 on hyvin subjektiivinen. [21] Luku nolla on matemaattisena käsitteenä haasteellinen, mutta joka tapauksessa TextBlob -kirjasto palauttaa myös arvoja 0 sekä polariteetin että subjektiivisuuden osalta.

Tämän tutkielman kokeellisessa osassa hyödynnetään pelkästään polariteetin arvoa, jonka perusteella kaikki twiitit luokitellaan kolmeen ryhmään. Nämä kolme ryhmää ovat positiiviset, neutraalit ja negatiiviset twiitit. Twiitti luokitellaan siis positiiviseksi, mikäli sen polariteetti on yli nollan, neutraaliksi jos sen polariteetti on tasan nolla ja

negatiiviseksi jos sen polariteetti on alle nollan. Kun kaikki twiitit on luokiteltu sentimenttinsä perusteella kuulumaan johonkin ryhmään, niin seuraavaksi lasketaan kaikkien saman vuorokauden twiittien sentimenttien prosentuaaliset jakaumat. Nämä prosentuaaliset jakaumat tallennetaan tiedostoon, johon kerätään kaikkien vuorokausien prosentuaaliset jakaumat. Tiedostoon tallennetaan siis neljä saraketta, jotka ovat päivämäärä ja positiivisten, neutraalien sekä negatiivisten twiittien prosentuaalinen osuus. Sentimenttianalyysin myötä tutkielman kokeelliseen osaan käytettävä rakenteeton tieto on saatu rakenteelliseen eli käytettävään muotoon.

Twiittien vuorokausikohtaisten sentimenttianalyysien tulokset on esitetty kuvassa 6. Pylväsdiagrammissa siniset pylväät kuvaavat positiivisten, harmaat neutraalien ja punaiset negatiivisten twiittien vuorokausikohtaista prosentuaalista osuutta. Positiivisten twiittien vuorokausikohtainen vaihteluväli oli 41,82% - 64,78%, neutraalien twiittien vaihteluväli oli 9,09% - 23,68% ja negatiivisten twiittien vaihteluväli oli 24,77% - 45,45%. Eräs mielenkiintoinen havainto sentimenttianalyysien tuloksista on, että pylväsdiagrammista ei pysty havaitsemaan mitään sentimenttijakauman yleistä suuntaa, vaan jakauma on vaihdellut hyvin samankaltaisesti koko aikavälillä eli 1.6.2018-10.2.2019. Twiittejä ei jostain syystä saatu ladattua 28.10.2018. Varsinainen syy tähän ei koskaan selvinnyt, mutta oma arvioni on, että twiittien lataamiseen käytettävässä GetOldTweets3 -kirjastossa on kyseisen päivän osalta jokin ongelma. Twiittejä ei kyseiseltä vuorokaudelta saatu ladattua ollenkaan missään muodossa. Tämän 28.10.2018 vuorokauden osalta tietoja ei ole ollenkaan mukana datassa, vaan kyseinen rivi on poistettu datasta. Tutkielman kokeellisen osan toteutuksen jälkeen on mielenkiintoista nähdä, mikä on sentimenttianalyysin vaikutus bitcoinin arvon ennustamisessa.



Kuva 6. Twiittien vuorokausikohtaisten sentimenttianalyysien tulokset.

5.4 Tietojen yhdistäminen

Tiedot ovat kolmessa erillisessä tiedostossa, joista yksi sisältää bitcoinin historiatiedot, toinen bitcoiniin liittyvien twiittien vuorokausikohtaiset volyymitiedot ja kolmas jokaisen vuorokauden suosituimpien twiittien sentimenttianalyysin tulokset. Kaikkia näitä kolmea tiedostoa yhdistävä tekijä on päivämäärä, ja siten tiedot saadaan jatkossa yhdistettyä helposti toisiinsa päivämäärän perusteella. Tämän tutkielman kokeellisessa osassa on tarkoitus tutkia, että mitä tietoja yhdistämällä saavutetaan parhaat tulokset bitcoinin arvon ennustamisessa. Tutkielman kokeellisessa osassa neuroverkkoja opetetaan kolmella erilaisella tietojen kombinaatiolla, jotka ovat bitcoinin historiatiedot yhdistettynä twiittien sentimenttianalyysiin, historiatiedot yhdistettynä twiittien vuorokausikohtaisiin volyymitietoihin ja historiatiedot yhdistettynä sekä sentimenttianalyysiin että volyymitietoihin. Käytännössä tiedot kootaan yhteen tiedostoon, ja käytettävät sarakkeet määritellään neuroverkkoon syötettäessä.

5.5 Tietojen syöttäminen neuroverkkoon ja sen opettaminen

Alakohdassa 5.5.1 kerrotaan *TensorFlow*:sta ja *Keras*:sta, joita käytetään kokeellisen osan toteutuksessa. Tietoja täytyy käsitellä ennen neuroverkkoon syöttämistä, ja tästä kerrotaan alakohdassa 5.5.2. Neuroverkkojen kouluttamiseen on valittu käytettäväksi muutamia erilaisia parametreja muutamien kokeilujen perusteella, ja näitä valittuja parametreja käsitellään alakohdassa 5.5.3. Alakohdassa 5.5.4 esitellään

koodiesimerkkien ja tekstin muodossa käytettävät LSTM-mallit. Alakohdassa 5.5.5 käydään läpi neuroverkon opettaminen ja tuloksien analysointitapa.

5.5.1 TensorFlow ja Keras

Tämän tutkielman kokeellisessa osassa neuroverkkojen toteutukseen käytetään TensorFlow:ta ja Keras:ia. TensorFlow on erilaisiin koneoppimisen sovelluksiin tarkoitettu avoimen lähdekoodin alusta, joka sisältää kattavasti erilaisia työkaluja ja kirjastoja. Esimerkiksi neuroverkkoja voi rakentaa ja kouluttaa hyödyntäen TensorFlow:ta ja muita korkean tason API:a, kuten Keras:ia. TensorFlow ei itsessään ole ohjelmointikieli tai -ympäristö riippuvainen, vaan sitä voi käyttää useimpien ohjelmointikielien kanssa sekä pilviympäristössä, selaimessa tai omalla laitteella. [40] Keras puolestaan on korkean tason API neuroverkkojen toteuttamiseen. Keras on kirjoitettu pythonilla ja sitä pystyy ajamaan TensorFlow:n lisäksi *CNTK*:n ja *Theano*:n päällä. Keras:in vahvuuksia ovat sen helppokäyttöisyys, monipuolisuus ja laajennettavuus. Keras sopii käytettäväksi tämän tutkielman kokeellisessa osassa hyvin, sillä se tukee myös takaisinkytkeytyviä neuroverkkoja. [41]

5.5.2 Tietojen käsittely

Tässä kokeellisessa osassa käytetään 70% datasta neuroverkkojen kouluttamiseen ja loput 30% neuroverkkojen testaamiseen. Data ladataan ensin csv-tiedostosta, sekä jaetaan valmiiksi kouluttamiseen ja testaamiseen tarkoitettuihin kokonaisuuksiin. Kuvassa 7 on koodiesimerkki datan lataamisesta sekä jakamisesta kouluttamiseen ja testaamisen tarkoitettuihin kokonaisuuksiin.

```
import pandas as pd
df = pd.read_csv("bitcoin_tiedot.csv", sep=';')
dataset_train = df.iloc[:179, [1,3,5,9]].values
dataset_test = df.iloc[179:, [1,3,5,9]].values
```

Kuva 7. Datan lataaminen csv-tiedostosta, sekä jakaminen kouluttamiseen ja testaamiseen tarkoitettuihin kokonaisuuksiin.

Kuvan 7 koodiesimerkissä koko csv-tiedosto ladataan ensin df -muuttujaan hyödyntäen *pandas* -kirjastoa [42]. Tämän jälkeen data jaetaan kouluttamiseen ja testaamiseen tarkoitettuihin kokonaisuuksiin siten, että 179 ensimmäistä riviä eli 70% datasta ladataan dataset_train -muuttujaan ja loput eli 30% datasta ladataan dataset_test -

muuttujaan. Tässä vaiheessa myös määritellään, mitkä csv-tiedoston sarakkeista otetaan mukaan neuroverkon kouluttamiseen.

Koska käytetään aikasarjatietoja, on data jaettava vielä osiin ennen kuin se voidaan syöttää neuroverkkoon. Data jaetaan siis *osakokonaisuuksiin*, jotka yksitellen syötetään neuroverkkoon. Osakokonaisuuden koko tarkoittaa sitä, että monenko vuorokauden data syötetään kerrallaan neuroverkkoon sekä koulutettaessa että ennustettaessa bitcoinin arvoa seuraavalle vuorokaudelle. Kuvassa 8 on esimerkki split_seq -funktiosta, joka jakaa datan myöhemmin käytettäviin osakokonaisuuksiin.

```
from numpy import array

def split_seq(data, step, features):
          x, y = list(), list()
          for i in range (len(data)):
                    seq_end = i + step
                    if seq_end > len(data):
                              break
                    step_x, step_y = data[i:seq_end, 0:features-1],
                    data[seq_end-1, features-1]
                    x.append(step_x)
                    y.append(step_y)
          return array(x), array(y)
```

Kuva 8. Koodiesimerkki split_seq -funktiosta, joka jakaa datan osakokonaisuuksiin.

Kuvassa 8 on split_seq -funktio, joka ottaa syötteenään jaettavan datan, osakokonaisuuksien koon sekä datan sisältämän muuttujien määrän. Osakokonaisuuksien koko on funktiossa step -attribuuttina ja datan sisältämä muuttujien määrä on funktiossa features -attribuuttina. Tämä funktio erottelee myös ennustettavan muuttujan valmiiksi erilleen muusta datasta, eli x-taulukkoon tallennetaan kaikki kouluttamiseen käytettävät muuttujat ja y-taulukkoon tallennetaan ennustettavan muuttujan arvot, eli tässä tapauksessa aina viimeisimmän sarakkeen lukuarvot. Funktio jakaa siis datan halutun kokoisiin osakokonaisuuksiin ja palauttaa ne lopuksi taulukkoina x eli kouluttamiseen tai ennustamisen käytettävät arvot sekä y eli ennustettavat arvot. Tässä kokeellisessa osassa on pääosin käytetty osakokonaisuuksien kokona kolmea, neljää ja viittä vuorokautta. Käytettävien muuttujien määrä vaihtelee sen perusteella, mitä tietoja käytetään neuroverkon kouluttamiseen. Kuvassa 9 on esimerkki split_seq -funktion palauttamista taulukoista, kun step -attribuutiksi eli osakokonaisuuksien kooksi oli määritelty kolme ja neuroverkon kouluttamiseen käytettäviä muuttujia oli myös kolme. Kuvassa 9 on esitettynä pelkästään alkuosat x- ja y-taulukoista.

```
x = [[[7599.8, 19041.1, 45316], [7698.8, 17363.6, 41760],
[7779.0, 13796.1, 41999]], 36 [[7698.8, 17363.6, 41760],
[7779.0, 13796.1, 41999], [7764.0, 18132.7, 46104]], …]
y = [7764.0, 7675.0, …]
```

Kuva 9. Esimerkki split_seq -funktion palauttamien x- ja y-taulukoiden alkuosista.

Neuroverkoille syötetään sekä koulutus että ennustusvaiheessa samankokoisia osakokonaisuuksia. Kuvan 9 x-taulukossa on edellisen kolmen vuorokauden muuttujien tiedot, ja y-taulukossa on seuraavalle vuorokaudelle ennustettava lukuarvo, eli tässä tapauksessa bitcoinin seuraavan vuorokauden korkein arvo.

5.5.3 Käytettävät parametrit

Neuroverkoissa on lukuisia erilaisia parametreja mitä voisi muuttaa ja kokeilla parhaiden tuloksien saavuttamiseksi, mutta rajallisen käytössä olevan ajan vuoksi moniin parametreihin oli valittava muutamien kokeilujen perusteella sopivimmilta vaikuttavat. Lisäksi monien parametrien kohdalla käytetään oletusarvoja, eli kaikkia parametreja ei edes varsinaisesti määritellä. ReLU -funktio valikoitui käytettäväksi aktivointifunktiona kaikissa koulutettavissa malleissa. Aktivointifunktioina testattiin aluksi myös sigmoid -funktiota, mutta kaikki kokeillut mallit ennustivat silloin pelkästään, että bitcoinin arvo laskee seuraavalle päivälle. Lisäksi kokeiltiin *tanh* -funktiota aktivointifunktiona, mutta ainakin pienien testien perusteella ReLU -funktio antoi hieman parempia tuloksia. Optimointialgoritmina puolestaan kaikissa koulutettavissa malleissa käytetään adam -algoritmia. Virheen laskentakaavana käytetään kaikissa koulutettavissa malleissa keskimääräistä neliövirhettä. Aktivointifunktioista, optimointialgoritmeista ja virheen laskentakaavoista on kerrottu tarkemmin kohdassa 4.3. Näiden lisäksi *epochs* -parametrin käytettäviksi arvoiksi on valittu muutamien kokeilujen perusteella 500, 1000 ja 5000. Epochs -parametrin osalta kokeiltiin useita erilaisia arvoja väliltä 5 – 5000, ja näiden kokeilujen tulosten perusteella valittiin käytettävät arvot. Epochs -parametrin osalta valittiin käytettäväksi vain kolme erilaista arvoa, koska tutkimuksen kokeelliseen osaan käytettävissä oleva aika oli rajallinen. Epochs -parametri määrittelee neuroverkon kouluttamisessa käytettävien syklien lukumäärän ja syklillä tarkoitetaan tässä tapauksessa sitä, että moneenko kertaan koulutusdata ajetaan läpi neuroverkosta [41].

5.5.4 Käytettävät LSTM-neuroverkkojen mallit

Tämän tutkielman kokeellisessa osassa valittiin käytettäväksi viisi erilaista LSTM -neuroverkon mallia. Nämä kaikki viisi mallia pohjautuvat Jason Brownleen kirjoituksessaan esittelemiin malleihin, mutta mallien rakennetta ja muutamia käytettäviä parametreja on muutettu. Mallit ovat *Vanilla LSTM, Stacked LSTM, Bidirectional LSTM, CNN LSTM* sekä *ConvLSTM*. [43] Näistä kaikista viidestä mallista käytetään melko yksinkertaistettuja versioita, eikä esimerkiksi piilotettuja kerroksia ollut kuin maksimissaan muutamia. Nämä viisi mallia yhdistettyinä kaikkiin valittuihin tietojen kombinaatioihin ja käytettäviin attribuutteihin tarkoittaa sitä, että yhteensä 117 erilaista kokonaisuutta koulutetaan ja testataan tämän tutkielman kokeellisessa osassa.

Vanilla LSTM -mallilla tarkoitetaan pelkistettyä mallia, jossa on vain yksi piilotettu kerros ja lisäksi tulostekerros, jota käytetään varsinaisten ennusteiden tekemiseen. Tämä kokeellisessa osassa käyttämäni malli on hyvin samankaltainen Brownleen kirjoituksessaan esittelemän Vanilla LSTM -mallin kanssa, mutta pienellä hienosäädöllä. [43] Kokeellisessa osassa käyttämästäni Vanilla LSTM -mallista on koodiesimerkki kuvassa 10.

```
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dense

model = Sequential()
model.add(LSTM(80, activation = 'relu', input_shape = (step, fea-
tures)))
model.add(Dense(1))
```

Kuva 10. Kokeellisessa osassa käytetyn Vanilla LSTM -mallin koodiesimerkki.

Kuvan 10 koodiesimerkissä määritellään ensin käytettävät kirjastot. Sen jälkeen model -muuttujaan määritellään keras.models -kirjaston *Sequential* -malli, joka tarkoittaa lineaarista pinoa, johon voidaan koota useita erilaisia kerroksia. Näitä mahdollisia kerroksia ovat syötekerros, piilotetut kerrokset sekä tulostekerros. [41] Seuraavaksi tähän lineaariseen pinoon lisätään keras.layers -kirjaston LSTM -kerros, jossa on 80 neuronia, aktivointifunktiona on ReLU -funktio, ja syötteen koko on vastaava, kuin mikä aiemmin esitetyllä split_seq -funktiolla muodostetaan. Syötteen koko tarkoittaa siis taulukon kokoa, jossa rivien lukumäärä on valittu askeleen (step) eli osakokonaisuuden koko ja sarakkeiden lukumäärä vastaa ennustukseen käytettävien muuttujien

lukumäärää. Neuronien lukumääräksi valittiin 80, koska se osoittautui kokeiluissa parhaimmaksi lukumääräksi. Kaikissa koulutettavissa neuroverkoissa käytetään samaa neuronien lukumäärää, koska käytössä oleva aika on rajallinen, eikä kaikkia erilaisia käytettävien parametrien yhdistelmiä voida kokeilla tämän tutkielman kokeellisen osan aikana. Valittu neuronien lukumäärä ei siis välttämättä ole optimaalisin. Lopuksi vielä lisätään tulostekerros, jossa käytetään keras.models -kirjaston *Dense* -kerrosta [41]. Tälle tulostekerrokselle määritellään pelkästään ulostulojen eli ennustettavien ominaisuuksien lukumäärä, joka on tässä tapauksessa yksi.

Stacked LSTM -mallilla tarkoitetaan LSTM -neuroverkon sovellusta, joka sisältää useita piilotettuja kerroksia. Kun piilotettuja kerroksia on useampia, niin silloin LSTM -kerrosta määriteltäessä return_sequenses -parametrin arvoksi asetetaan True, jolloin yksittäisen neuronin ulostulo palautetaan uudelleenkäytettäväksi. [41] [43] Tämän tutkielman kokeellisessa osassa tässä mallissa käytetään kolmea piilotettua kerrosta, mutta muuten malli on samankaltainen Brownleen kirjoituksessaan esittelemän mallin kanssa [43]. Kuvassa 11 on esitettynä koodiesimerkki kokeellisessa osassa käytetystä Stacked LSTM -mallista.

```
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dense

model = Sequential()
model.add(LSTM(80, activation = 'relu', return_sequences = True, in-
put_shape = (step, features)))
model.add(LSTM(80, activation = 'relu', return_sequences = True))
model.add(LSTM(80, activation = 'relu'))
model.add(Dense(1))
```

Kuva 11. Kokeellisessa osassa käytetyn Stacked LSTM -mallin koodiesimerkki.

Kuvassa 11 oleva Stacked LSTM -malli määritellään siis hyvin samalla tavalla, kuin Vanilla LSTM -malli määriteltiin kuvassa 10, mutta nyt LSTM -kerroksia on kolme. Periaatteessa varsinaisia piilotettuja kerroksia on nyt kaksi, jos ensimmäinen LSTM -kerros lasketaan syötekerrokseksi. Kahdella ensimmäisellä LSTM -kerroksella return_sequences -parametrin arvoksi määritellään siis True, ja kaikkien LSTM -kerroksien neuronien lukumäärä on 80.

Bidirectional LSTM -mallilla tarkoitetaan sellaista mallia, joka oppii syötteet kahteen suuntaan eli sekä eteen- että taaksepäin ja yhdistää molemmista saamansa tulkinnat

[43]. Tällainen *Bidirectional -kerros* saadaan käyttöön suoraan keras.models -kirjastosta [41]. Tämän tutkielman kokeellisessa osassa käytetään tämän mallin osalta kahta piilotettua kerrosta, joista ensimmäinen on Bidirectional -kerros. Myös tämä malli on samankaltainen Brownleen kirjoituksessaan esittelemän mallin kanssa, mutta yksi piilotettu kerros on lisätty ja käytettäviä parametreja on muutettu [43]. Kuvassa 12 on koodiesimerkki kokeellisessa osassa käytetystä Bidirectional LSTM -mallista.

```
from keras.layers import Bidirectional

model = Sequential()
model.add(Bidirectional(LSTM(80, activation = 'relu', input_shape =
(step, features), merge_mode = 'ave'))
model.add(Dense(80, activation = 'relu'))
model.add(Dense(1))
```

Kuva 12. Kokeellisessa osassa käytetyn Bidirectional LSTM -mallin koodiesimerkki.

Kuvassa 12 Bidirectional -kerrokselle syötetään parametreiksi vastaavanlainen LSTM -kerros, jota on aiemminkin käytetty muissa malleissa. Lisäksi kerroksen merge_mode -parametriksi eli eteen- ja taaksepäin menevien syötteiden yhdistämiseen käytettäväksi funktioksi määritellään 'ave', joka tarkoittaa syötteiden keskiarvoa [41]. Muitakin merge_mode -parametreja kokeiltiin, mutta jälleen kerran pienen taustatutkimuksen perusteella syötteiden keskiarvo valikoitui käytettäväksi. Bidirectional -kerroksen jälkeen on vielä toinen piilotettukerros, tällä kertaa Dense -kerros 80 neuronilla, sillä kokeilujen perusteella tämä vaikutti toimivan tässä paremmin kuin pelkkä Bidirectional -kerros.

CNN LSTM -malli eroaa kolmesta edellisestä mallista siten, että siinä LSTM -malli on ikään kuin yhdistettynä CNN -malliin, eli konvoluutioneuroverkon malliin. Tässä mallissa syötekerroksen jälkeen käytetään CNN -kerrosta, joka tulkitsee aikasarjatietojen osakokonaisuuksia ja antaa näiden osakokonaisuuksien tulkinnan tuloksen syötteeksi LSTM -kerrokselle. Tämän mallin tapauksessa neuroverkolle syötettävät osakokonaisuudet on jaettava hieman eri tavalla kuin aiempien mallien osalta, sillä osakokonaisuudet itsessään on jaettava vielä osiin. [43] [44] Kuvassa 13 on esimerkki taulukkojen x ja y alkuosista, kun osakokonaisuuksien kooksi on määritelty kuusi ja jokainen osakokonaisuus on jaettu vielä kolmeen osaan.

```
x = [[[[7599.8, 19041.1, 45316], [7698.8, 17363.6, 41760]],
[[7779.0, 13796.1, 41999], [7764.0, 18132.7, 46104]],
```

```
[[7675.0, 16741.6, 46860], [7700.0, 18269.8, 45323]]], …]

y = [7756.3, 7697.0, …]
```

Kuva 13. Esimerkki CNN LSTM -malliin syötettävien x- ja y-taulukkojen alkuosista.

Taulukko y on rakenteeltaan vastaavanlainen kuin kolmen aiemman mallin tapauksessa. Kuvasta 13 nähdään, että taulukossa x puolestaan on muutoksia verrattuna aiemmissa malleissa käytettyyn taulukkoon, sillä osakokonaisuudet on jaettu vielä osiin. Kuvassa 13 on esitetty taulukossa x yksi kuuden vuorokauden mittainen osakokonaisuus, joka on jaettu vielä kolmeen osaan eli toisin sanoen nämä osat sisältävät aina kahden vuorokauden muuttujien tiedot. Tämä tutkielman kokeellisessa osassa käyttämäni malli on hyvin samankaltainen kuin Brownleen kirjoituksessaan esittelemä CNN LSTM -malli, mutta muutamia parametreja on muutettu ja yksi piilotettu kerros on lisätty [43]. Kuvassa 14 on esitettynä kokeellisessa osassa käytetyn CNN LSTM -mallin koodiesimerkki.

```
from keras.layers import TimeDistributed
from keras.layers.convolutional import Conv1D
from keras.layers.convolutional import MaxPooling1D
from keras.layers import Flatten

model = Sequential()

#CNN malli
model.add(TimeDistributed(Conv1D(filters = 95, kernel_size = 1, acti-
vation = 'relu', input_shape = (None, step, features)))
model.add(TimeDistributed(MaxPooling1D(pool_size = 2)))
model.add(TimeDistributed(Flatten())

#LSTM malli
model.add(LSTM(80, activation = 'relu'))
model.add(Dense(80, activation = 'relu'))
model.add(Dense(1))
```

Kuva 14. Kokeellisessa osassa käytetyn CNN LSTM -mallin koodiesimerkki.

Kuvassa 14 ensin määritellään CNN -malli eli konvoluutiomalli, jonka kaikki kolme kerrosta asetetaan keras.layers -kirjaston *TimeDistributed* -kerroksen sisään. TimeDirstibuted -kerroksen avulla mallia sovelletaan jokaista yksittäistä osajoukon osaa kohti. Ensimmäinen konvoluutiomallin kerros muodostuu keras.layers.convolutional -kirjaston *Conv1D* -kerroksesta, joka on yksiulotteisen datan käsittelyyn tarkoitettu konvoluutiokerros. Tälle kerrokselle määritellään muutamia parametreja, joista esimerkiksi filters -parametriksi eli ulostulojen ulottuvuudeksi asetetaan 95. Konvoluutiomallin toinen kerros muodostuu keras.layers.convolutional -kirjaston *MaxPooling* -

kerroksesta, joka ikään kuin suodattaa tärkeimmät ominaisuudet konvoluutiomallin viimeseille eli keras.layers.convolutional -kirjaston *Flatten* -kerrokselle. Flatten -kerros hoitaa lähinnä tietojen siirtämisen konvoluutiomallista LSTM -mallille. [41] [43] LSTM -mallina tässä tapauksessa käytetään melko pelkistettyä, kahden kerroksen mallia.

Viides käytettävä malli eli Conv LSTM -malli on hyvin samankaltainen kuin CNN LSTM -malli. Eroavaisuutena näiden mallien välillä on se, että Conv LSTM -mallissa jokaiseen LSTM -yksikköön on rakennettu suoraan vastaavanlainen sisääntulo kuin CNN -mallissa. Koska Conv LSTM -malli on alun perin kehitetty kaksiulotteisten paikkatietojen lukemiseen, on siihen syötettävä data muutettava hieman erilaiseen muotoon. [43] Osakokonaisuudet tämän mallin osalta on jaettu vastaavalla tavalla kuin CNN LSTM -mallissa. Myös tämän mallin osalta on käytetty hyvin samankaltaista mallia, jonka Brownlee kirjoituksessaan esittelee, mutta muutamia parametreja on hienosäädetty ja yksi piilotettu kerros on lisätty ennen tulostekerrosta [43]. Kuvassa 15 on esitettynä kokeellisessa osassa käytetyn Conv LSTM -mallin koodiesimerkki.

```
from keras.layers import ConvLSTM2D

model = Sequential()
model.add(ConvLSTM2D(filters = 95, kernel_size = (1,2), activation =
'relu', input_shape(seq, 1, step, features)))
model.add((Flatten())
model.add(Dense(80, activation = 'relu'))
model.add(Dense(1))
```

Kuva 15. Kokeellisessa osassa käytetyn Conv LSTM -mallin koodiesimerkki.

Conv LSTM -malli on määritelty kuvassa 15. Conv LSTM -mallissa ensimmäisenä kerroksena käytetään keras.layers -kirjaston *ConvLSTM2D* -kerrosta, joka on siis samankaltainen kuin LSTM -kerros, mutta kerroksen toiminta on konvoluutiomaista. Tässä kernel_size -parametri on nyt oltava kaksiulotteinen ja lisäksi imput_shape -parametri sisältää neljä erillistä lukuarvoa johtuen kerroksen rakenteesta. Tässä mallissa käytetään myös Flatten -kerrosta, joka siis yhdistää edellisen kerroksen antamat tulosteet yksiulotteiseksi vektoriksi seuraavaa piilotettua kerrosta varten. [41]

5.5.5 Neuroverkon opettaminen ja tuloksien analysointitapa

Mallien määrittelyjen jälkeen neuroverkkojen kouluttaminen käynnistetään kuvan 16 koodiesimerkin komentoja käyttäen.

```
model.compile(optimizer = 'adam', loss = 'mse')
model.fit(x, y, batch_size = 32, epochs = 500)
```

Kuva 16. Koodiesimerkki neuroverkkojen kouluttamisen käynnistämisestä.

Kuvassa 16 ensin compile -komennolla konfiguroidaan malli neuroverkon kouluttamista varten. Tässä vaiheessa määritellään myös neuroverkon kouluttamisessa käytettävä optimointialgoritmi ja virheen laskentakaava. Tämän jälkeen neuroverkkoon syötetään x- ja y-taulukot sekä määritellään batch_zise - ja epochs -parametrit fit -komennon avulla. [41]

Vaikka koulutetut neuroverkot asetettiin ennustamaan bitcoinin seuraavan vuorokauden arvoa tarkkana lukuarvona, niin tuloksia tullaan tarkastelemaan tämän tutkielman kokeellisessa osassa pelkästään suunnan ennustamisen osalta. Siten neuroverkkojen ennustukset luokitellaan kahteen kategoriaan, jotka ovat arvon nouseminen seuraavalle vuorokaudelle ja arvon laskeminen tai samana pysyminen seuraavalle vuorokaudelle. Testidatalla ennustettuja arvoja verrataan todellisiin arvoihin, joita neuroverkko ei ole aiemmin nähnyt ja näistä muodostetaan sekaannusmatriisi, jonka käyttö esiteltiin kohdassa 4.3. Sekaannusmatriisista puolestaan lasketaan jokaisen koulutetun neuroverkon ennusteen tarkkuus, ja näitä tuloksia verrataan toisiinsa. Esimerkki erään neuroverkon tuottamien ennusteiden sekaannusmatriisista:

Taulukko 4. Esimerkki sekaannusmatriisista.

		Todelliset luokat	
		Arvo nousee	Arvo laskee tai pysyy samana
Ennustetut luokat	Arvo nousee	19	26
	Arvo laskee tai pysyy samana	10	18

Tämän neuroverkon tapauksessa bitcoinin arvon nousu ennustettiin 19 vuorokauden osalta oikein ja bitcoinin arvon lasku tai samana pysyminen ennustettiin 18 vuorokauden osalta oikein. Yhteensä 36 vuorokauden osalta ennustettiin bitcoinin arvon suunta väärin. Tämä tarkoittaa sitä, että tämän esimerkin neuroverkon tarkkuus oli:

$$\frac{TP + TN}{TP + FP + FN + TN} = \frac{19 + 18}{19 + 26 + 10 + 18} \approx 0.50$$

6 Tutkimuksen tulokset

Neuroverkot laitettiin ennustamaan tarkkaa lukuarvoa, mutta tulokset esitetään ja analysoidaan bitcoinin arvon suunnan ennustamisen osalta. Neuroverkon ennustamaa lukuarvoa verrataan todelliseen lukuarvoon, ja sen perusteella analysoidaan, kuinka hyvin neuroverkko pystyy ennustamaan bitcoinin korkeimman arvon suuntaa seuraavalle vuorokaudelle. Yhteensä erilaisia neuroverkkoja koulutettiin 117, käyttäen viittä erilaista mallia, eri yhdistelmiä koulutusdatasta sekä erilaisia parametreja. Näistä valituista malleista, koulutusdatan yhdistelmistä sekä parametreista kerrottiin tarkemmin kohdassa 5.

Kokeellisen osan ensimmäisessä vaiheessa tutkittiin bitcoinin arvon ennustamista seuraavalle vuorokaudelle käyttäen historiatietoja ja bitcoiniin liittyvien twiittien volyymitietoja. Kokeellisen osan ensimmäisen vaiheen tuloksista kerrotaan kohdassa 6.1. Kokeellisen osan toisessa vaiheessa tutkittiin bitcoinin arvon ennustamista seuraavalle vuorokaudelle käyttäen historiatietoja ja bitcoiniin liittyvien suosituimpien twiittien vuorokausikohtaisia sentimenttianalyyseja. Toisen vaiheen tuloksista kerrotaan kohdassa 6.2. Kolmannessa eli viimeisessä vaiheessa tutkittiin bitcoinin arvon ennustamista seuraavalle vuorokaudelle käyttäen historiatietoja, twiittien volyymitietoja sekä sentimenttianalyyseja. Kokeellisen osan viimeisen vaiheen tuloksista kerrotaan kohdassa 6.3. Kohdassa 6.4 vertaillaan ja analysoidaan kokeellisen osan eri vaiheiden tuloksia keskenään.

6.1 Suunnan ennustamisen tulokset käyttäen historiatietoja ja twiittien volyymitietoja

Tuloksien analysointitapa esitettiin alakohdassa 5.5.5. Jokaisesta koulutetusta neuroverkosta on laskettu tarkkuus, ja näitä tarkkuuden arvoja verrataan keskenään. Tulokset esitetään kahdessa erillisessä taulukossa, sillä osassa käytetyistä malleista koulutusdata syötetään eri tavalla jaoteltuna neuroverkkoon. Taulukossa 5 on esitettynä suunnan ennustamisen tulokset Vanilla LSTM-, Stacked LSTM- sekä Bidirectional LSTM -malleja käyttäen, kun käytettiin historiatietoja ja twiittien volyymitietoja.

Taulukko 5. Suunnan ennustamisen tarkkuudet käyttäen historiatietoja ja twiittien volyymitietoja. Käytettyinä malleina Vanilla LSTM, Stacked LSTM sekä Bidirectional LSTM.

	Askel = 3			Askel = 4			Askel = 5		
	Epochs =500	Epochs =1000	Epochs =5000	Epochs =500	Epochs =1000	Epochs =5000	Epochs =500	Epochs =1000	Epochs =5000
Vanilla	0.547	0.581	0.567	0.536	0.504	0.520	0.547	0.615	0.702
Stacked	0.456	0.410	0.567	0.568	0.427	0.602	0.662	0.567	0.615
Bidirectional	0.449	0.522	0.432	0.565	0.531	0.547	0.506	0.610	0.619

Taulukossa 5 on esitettynä Vanilla LSTM-, Stacked LSTM- sekä Bidirectional LSTM -mallien sekä kaikkien niissä käytettyjen parametriyhdistelmien tulokset. Näiden jokaisen mallin kohdalla koulutettiin yhdeksän erilaista neuroverkkoa, joissa askel -muuttujan ja epochs -parametrin arvoja muutettiin. Taulukossa askel -muuttuja tarkoittaa sitä, että monenko vuorokauden kokoisissa osakokonaisuuksissa tietoja syötettiin neuroverkkoon sekä koulutusvaiheessa että testausvaiheessa. Käytetyt lukuarvot olivat kolme, neljä ja viisi vuorokautta. Epochs -parametri puolestaan tarkoittaa sitä, että moneenko kertaan koulutusdata ajetaan läpi neuroverkosta. Epochs -parametrissa käytetyt lukuarvot olivat 500, 1000 ja 5000. Jokaista mallia ja kaikkia parametriyhdistelmiä käyttäen koulutettiin aina neuroverkko kolmeen kertaan ja tulokseksi poimittiin sitten paras näistä kolmesta.

Taulukosta 5 nähdään, että Vanilla LSTM -mallin osalta parhaaksi tulokseksi kokeellisen osan ensimmäisessä vaiheessa saatiin tarkkuus 0.702, kun askel oli viisi ja epochs oli 5000. Stacked LSTM -mallin osalta paras tarkkuus oli 0.662, kun askel oli viisi ja epochs oli 500. Bidirectional LSTM -mallin osalta paras tarkkuus oli puolestaan 0.619, kun askel oli viisi ja epochs oli 5000. Taulukosta 5 voidaan siis havaita, että näiden kolmen mallin osalta kokeellisen osan ensimmäisessä vaiheessa parhaimmat tulokset saavutettiin, kun askeleena oli viisi.

Seuraavaksi taulukossa 6 puolestaan on esitettynä suunnan ennustamisen tulokset CNN LSTM- ja Conv LSTM -malleja käyttäen. Näiden kahden mallin osalta epochs -parametrin arvoina käytettiin samoja arvoja kuin aiemminkin eli 500, 1000 ja 5000. Näiden mallien tapauksessa neuroverkkoon syötettävät tiedot on jaoteltu kuitenkin eri

tavalla, kun neuroverkkoon syötettävien osakokonaisuuksien sisältö on jaettu vielä pienempiin osiin. Taulukossa 6 askel -muuttujan lisäksi on myös jaottelu -muuttuja, joka kuvaa osakokonaisuuksien jaotellun sisällön osien lukumäärän. Näiden muuttujien osalta valittiin käytettäväksi kahta erilaista yhdistelmää, joissa ensimmäisessä askel oli kuusi ja jaottelu oli kolme, ja jälkimmäisessä yhdistelmässä askel oli neljä ja jaottelu oli kaksi. Sekä CNN LSTM – että Conv LSTM -mallien osalta koulutettiin siis kuusi erilaista neuroverkkoa.

Taulukko 6. Suunnan ennustamisen tarkkuudet käyttäen historiatietoja ja twiittien volyymitietoja. Käytettyinä malleina CNN LSTM ja Conv LSTM.

	Askel = 6, jaottelu 3			Askel = 4, jaottelu 2		
	Epochs =500	Epochs =1000	Epochs =5000	Epochs =500	Epochs =1000	Epochs =5000
CNN	0.607	0.568	0.690	0.547	0.580	0.534
Conv	0.521	0.605	0.633	0.493	0.520	0.575

Taulukosta 6 nähdään, että sekä CNN LSTM – että Conv LSTM -mallin osalta parhaat tarkkuuden arvot saavutettiin, kun askel oli kuusi, jaottelu oli kolme ja epochs -parametrin arvo oli 5000. Tässä kokeellisen osan ensimmäisessä vaiheessa CNN LSTM -mallin paras tarkkuus oli siis 0.690 ja Conv LSTM -mallin paras saavutettu tarkkuus oli 0.633.

Kokonaisuudessaan voidaan todeta, että tämän kokeellisen osan ensimmäisen vaiheen perusteella historiatietoja ja bitcoiniin liittyvien twiittien volyymitietoja käyttämällä paras tarkkuus bitcoin arvon suunnan ennustamiseksi seuraavalle vuorokaudelle oli 0.702. Tämä tarkkuus saavutettiin Vanilla LSTM -mallilla, kun askel oli viisi ja epochs -parametri oli 5000.

6.2 Suunnan ennustamisen tulokset käyttäen historiadataa ja twiittien sentimenttianalyyseja

Kokeellisen osan toisessa vaiheessa tutkittiin bitcoinin arvon suunnan ennustamista käyttäen historiatietoja ja bitcoiniin liittyvien suosituimpien twiittien

vuorokausikohtaisia sentimenttianalyyseja. Kokeellisen osan toisessa vaiheessa käytettiin samoja malleja ja parametreja kuin kokeellisen osan ensimmäisessä vaiheessa, ja siten kokeellisen osan toisen vaiheen tulokset on esitetty vastaavanlaisissa taulukoissa kuin kokeellisen osan ensimmäisen vaiheen tulokset. Taulukossa 7 on esitettynä suunnan ennustamisen tulokset Vanilla LSTM-, Stacked LSTM- sekä Bidirectional LSTM -malleja käyttäen, kun käytettiin historiatietoja ja suosituimpien twiittien vuorokausikohtaisia sentimenttianalyyseja.

Taulukko 7. Suunnan ennustamisen tarkkuudet käyttäen historiatietoja ja suosituimpien twiittien vuorokausikohtaisia sentimenttianalyyseja. Käytettyinä malleina Vanilla LSTM, Stacked LSTM sekä Bidirectional LSTM.

	Askel = 3			Askel = 4			Askel = 5		
	Epochs =500	Epochs =1000	Epochs =5000	Epochs =500	Epochs =1000	Epochs =5000	Epochs =500	Epochs =1000	Epochs =5000
Vanilla	0.527	0.418	0.527	0.493	0.534	0.712	0.611	0.569	0.625
Stacked	0.486	0.527	0.486	0.547	0.479	0.534	0.569	0.486	0.638
Bidirectional	0.50	0.472	0.459	0.589	0.589	0.753	0.680	0.597	0.611

Taulukosta 7 nähdään, että Vanilla LSTM -mallin osalta paras ennustetarkkuus eli 0.712 saavutettiin, kun askel oli neljä ja epochs -parametrin arvo oli 5000. Stacked LSTM -mallin osalta puolestaan paras ennustetarkkuus oli 0.638, kun askel oli viisi ja epochs -parametrin arvo oli 5000. Bidirectional LSTM -mallin osalta paras ennustetarkkuus oli 0.753, kun askel oli neljä ja epochs -parametrin arvo oli 5000. Taulukosta 7 voidaan siis havaita, että näiden kolmen mallin osalta kokeellisen osan toisessa vaiheessa parhaimmat tulokset saavutettiin, kun epochs -parametrin arvo oli 5000, ja askel muuttuja oli neljä tai viisi.

Seuraavaksi taulukossa 8 on esitettynä suunnan ennustamisen tulokset CNN LSTM- ja Conv LSTM -malleja käyttäen kokeellisen osan toisessa vaiheessa.

Taulukko 8. Suunnan ennustamisen tarkkuudet käyttäen historiatietoja ja suosituimpien twiittien vuorokausikohtaisia sentimenttianalyyseja. Käytettyinä malleina CNN LSTM ja Conv LSTM.

	Askel = 6, jaottelu 3			Askel = 4, jaottelu 2		
	Epochs =500	Epochs =1000	Epochs =5000	Epochs =500	Epochs =1000	Epochs =5000
CNN	0.563	0.619	0.704	0.438	0.616	0.506
Conv	0.549	0.633	0.619	0.452	0.438	0.437

Taulukosta 8 nähdään, että CNN LSTM -mallin osalta paras ennustetarkkuus oli 0.704, kun askeleena oli kuusi, jaotteluna oli kolme, ja epochs -parametrin arvo oli 5000. Conv LSTM -mallin osalta paras ennustetarkkuus oli puolestaan 0.633, kun askeleena oli kuusi, jaotteluna oli kolme ja epochs -parametrin arvona oli 1000.

Kokonaisuudessaan voidaan todeta, että tämän kokeellisen osan toisen vaiheen perusteella historiatietoja ja bitcoiniin liittyvien suosituimpien twiittien vuorokausikohtaisia sentimenttianalyyseja käyttämällä paras tarkkuus bitcoin arvon suunnan ennustamiseksi seuraavalle vuorokaudelle oli 0.753. Tämä tarkkuus saavutettiin Bidirectional LSTM -mallilla, kun askel oli neljä, ja epochs -parametrin arvo oli 5000.

6.3 Suunnan ennustamisen tulokset käyttäen historiatietoja, twiittien volyymitietoja sekä sentimenttianalyyseja

Kokeellisen osan kolmannessa eli viimeisessä vaiheessa tutkittiin bitcoinin arvon suunnan ennustamista käyttäen historiatietoja, twiittien vuorokausikohtaisia volyymitietoja sekä bitcoiniin liittyvien suosituimpien twiittien vuorokausikohtaisia sentimenttianalyyseja. Kokeellisen osan kolmannessa vaiheessa käytettiin samoja malleja ja parametreja kuin kokeellisen osan edellisissä vaiheissa, ja siten kokeellisen osan kolmannen vaiheen tulokset on esitetty myös vastaavanlaisissa taulukoissa kuin kokeellisen osan aiempien vaiheiden tulokset. Taulukossa 9 on esitettynä suunnan ennustamisen tulokset Vanilla LSTM-, Stacked LSTM- sekä Bidirectional LSTM -malleja käyttäen, kun käytettiin historiatietoja, twiittien volyymitietoja sekä suosituimpien twiittien vuorokausikohtaisia sentimenttianalyyseja.

Taulukko 9. Suunnan ennustamisen tarkkuudet käyttäen historiatietoja, twiittien volyymitietoja sekä suosituimpien twiittien vuorokausikohtaisia sentimenttianalyyseja. Käytettyinä malleina Vanilla LSTM, Stacked LSTM sekä Bidirectional LSTM.

	Askel = 3			Askel = 4			Askel = 5		
	Epochs =500	Epochs =1000	Epochs =5000	Epochs =500	Epochs =1000	Epochs =5000	Epochs =500	Epochs =1000	Epochs =5000
Vanilla	0.58	0.513	0.364	0.547	0.616	0.616	0.680	0.527	0.666
Stacked	0.472	0.472	0.418	0.547	0.520	0.643	0.597	0.486	0.791
Bidirec-tional	0.486	0.391	0.567	0.561	0.602	0.630	0.55	0.583	0.569

Taulukosta 9 nähdään, että Vanilla LSTM -mallin osalta paras ennustetarkkuus oli 0.680, kun askeleena käytettiin neljää ja epochs -parametrin arvona oli 500. Tässä kokeellisen osan kolmannessa vaiheessa Stacked LSTM -mallilla saavutettiin huomattavasti parempi ennustetarkkuus, kuin aiemmin koulutetuilla neuroverkoilla. Tämä ennustetarkkuus oli 0.791, kun askel oli viisi ja epochs -parametrina oli 5000. Bidirectional LSTM -mallilla puolestaan paras ennustetarkkuus oli 0.630, kun askel oli neljä ja epochs -parametri oli 5000.

Seuraavassa taulukossa 10 on esitettynä suunnan ennustamisen tulokset CNN LSTM- ja Conv LSTM -malleja käyttäen kokeellisen osan kolmannessa vaiheessa.

Taulukko 10. Suunnan ennustamisen tarkkuudet käyttäen historiatietoja, bitcoiniin liittyvien twiittien volyymitietoja sekä suosituimpien twiittien vuorokausikohtaisia sentimenttianalyyseja. Käytettyinä malleina CNN LSTM ja Conv LSTM.

	Askel = 6, jaottelu 3			Askel = 4, jaottelu 2		
	Epochs =500	Epochs =1000	Epochs =5000	Epochs =500	Epochs =1000	Epochs =5000
CNN	0.619	0.619	0.647	0.452	0.520	0.356
Conv	0.492	0.577	0.662	0.506	0.507	0.479

Taulukosta 10 nähdään, että CNN LSTM -mallin osalta paras ennustetarkkuus oli 0.647, kun askeleena oli käytetty kuutta vuorokautta, jaotteluna oli kolme ja epochs -parametrin arvona oli 5000. Tässä kokeellisen osan kolmannessa vaiheessa Conv

LSTM -mallilla saavutettiin vastaavilla parametreilla hieman parempi ennustetarkkuus, joka oli 0.662.

Kokonaisuudessaan voidaan todeta, että kolmannen vaiheen perusteella historiatietoja, twiittien volyymitietoja sekä suosituimpien bitcoiniin liittyvien twiittien sentimenttianalyyseja käyttämällä paras tarkkuus bitcoin arvon suunnan ennustamiseksi seuraavalle vuorokaudelle oli 0.791. Tämä tarkkuus saavutettiin Stacked LSTM -mallilla, kun askeleena oli viisi, ja epochs -parametrina oli 5000.

6.4 Yhteenveto tuloksista

Kohtien 6.1 – 6.3 taulukoiden perusteella voidaan päätellä, että eri neuroverkoilla aikaansaadut ennustetarkkuudet vaihtelevat aika paljon. Tämän tutkielman kokeellisessa osassa oltiin kuitenkin erityisesti kiinnostuneita siitä, että miten hyvin bitcoinin arvon suuntaa pystytään ennustamaan seuraavalle vuorokaudelle. Tämän vuoksi ollaan erityisen kiinnostuneita parhaista saavutetuista ennustetarkkuuksista. Kokeellisen osan ensimmäisessä vaiheessa käytettiin siis historiatietoja ja bitcoiniin liittyvien twiittien volyymitietoja, ja silloin parhaaksi ennustetarkkuudeksi saatiin 0.702. Kokeellisen osan toisessa vaiheessa käytettiin puolestaan historiatietoja ja suosituimpien bitcoiniin liittyvien twiittien vuorokausikohtaisia sentimenttianalyyseja, ja silloin parhaaksi ennustetarkkuudeksi saatiin 0.753. Kokeellisen osan kolmannessa vaiheessa saavutettiin koko tutkimuksen paras ennustetarkkuus, joka oli 0.791. Tällöin käytettiin siis kaikkia tietoja, eli historiatietoja, twiittien volyymitietoja sekä sentimenttianalyyseja. Mallina silloin oli Stacked LSTM, askeleena oli viisi, ja epochs -parametrina oli 5000.

Kokeellisen osan perusteella vaikuttaa siltä, että ennustetarkkuudet paranivat, kun epochs -parametrin arvot kasvoivat. Poikkeuksiakin toki löytyi, mutta pääosin epochs -parametrin arvolla 5000 saavutettiin parempia tuloksia kuin arvoilla 500 ja 1000. Vanilla LSTM-, Stacked LSTM- ja Bidirectional LSTM -mallien osalta ennustetarkkuudet vaikuttivat parantuvan, kun askeleen määrää kasvatettiin. Myös CNN LSTM – ja Conv LSTM -mallien osalta askeleena kuusi ja jaotteluna kolme vaikutti toimivan paremmin kuin askeleena neljä ja jaotteluna kaksi.

7 Johtopäätökset ja yhteenveto

Tässä tutkielmassa keskityttiin erityisesti kokeelliseen osaan, jossa tutkittiin kryptovaluutta bitcoinin arvon suunnan ennustamista seuraavalle vuorokaudelle käyttäen rakenteellisia ja rakenteettomia tietoja. Tutkielma sisälsi myös kirjallisuuskatsauksen aiheeseen liittyen, jonka yhteydessä kerrottiin aiemmista tutkimuksista liittyen bitcoinin arvon ennustamiseen. Kokeellisen osan tärkeimpänä tarkoituksena oli ennustaa bitcoinin arvon suuntaa seuraavalle vuorokaudelle mahdollisimman hyvin. Kokeellisessa osassa tutkittiin myös, millaisilla koulutusdatan yhdistelmillä saavutetaan parhaat ennustetarkkuudet.

Kokeellisessa osassa rakenteellisina tietoina käytettiin bitcoinin historiatietoja ja bitcoiniin liittyvien twiittien vuorokausikohtaisia volyymitietoja. Rakenteettomina tietoina käytettiin bitcoiniin liittyviä suosituimpia twiittejä, ja näistä twiiteistä tehtiin vuorokausikohtaisia sentimenttianalyyseja. Sentimenttianalyysien tulokset yhdistettiin rakenteelliseen tietoon ja valitut koulutusdatan yhdistelmät syötettiin neuroverkkoihin. Kokeellisessa osassa käytetyt koulutusdatan yhdistelmät olivat historiatiedot yhdistettynä bitcoiniin liittyvien twiittien vuorokausikohtaisiin volyymitietoihin, historiatiedot yhdistettyinä bitcoiniin liittyvien suosituimpien twiittien vuorokausikohtaisiin sentimenttianalyyseihin sekä historiatiedot yhdistettynä twiittien volyymitietoihin ja sentimenttianalyyseihin. Kokeellisessa osassa valittiin käytettäväksi takaisinkytkeytyvä LSTM -neuroverkko, josta käytettiin viittä erilaista sovellettua mallia. Kokeellisessa osassa neuroverkoissa valittiin käytettäväksi vain muutamia erilaisia parametriyhdistelmiä, koska käytössä oleva aika oli rajallinen.

Tutkielman kokeellisessa osassa parhaaksi ennustetarkkuudeksi saatiin 0.791, kun neuroverkon koulutukseen käytettiin kaikkia tietoja, eli historiatietoja, twiittien volyymitietoja sekä sentimenttianalyyseja. Tämä tulos saavutettiin LSTM -mallin sovelluksella, jossa käytettiin kolmea piilotettua LSTM -kerrosta ja jokaisella piilotetulla kerroksella oli 80 neuronia. Kokeellisessa osassa yhdistämällä suosituimpien twiittien sentimenttianalyysit historiatietoihin saavutettiin parempia tuloksia kuin yhdistämällä twiittien volyymitiedot historiatietoihin. Twiittien sentimenttianalyyseja ja

historiatietoja käyttämällä paras saavutettu ennustetarkkuus oli 0.753, kun twiittien volyymitietoja ja historiatietoja käyttämällä paras saavutettu ennustetarkkuus oli 0.702.

Tutkimuksen kokeellisessa osassa saavutettu paras ennustetarkkuus eli 0.791 oli melko hyvä verrattaessa aiempiin tutkimuksiin. Tämän tutkielman kokeellisessa osassa ennustettiin kuitenkin pelkästään bitcoinin arvon suuntaa seuraavalle vuorokaudelle, kun monissa aiemmissa tutkimuksissa on ennustettu tarkkaa lukuarvoa ja tulokset on esitetty sen pohjalta. Siten tuloksien vertaaminen aiempiin tutkimuksiin on haasteellista.

Kokeellisen osan toteutuksessa monet asiat oli tarkasti rajattu, koska käytössä oli aikaa rajallisesti. Siten lähes kaikkiin näihin rajauksiin liittyen voisi tehdä jatkotutkimusta. Rakenteettomana tietona käytettyjä twiittejä ladattiin hyvin rajallisesti, sillä käytettiin pelkästään jokaisen vuorokauden suosituimpia twiittejä. Twiittien rajaukseen ja sen vaikutuksesta ennustetarkkuuteen voisi tehdä paljon jatkotutkimusta, sillä olisi mielenkiintoista nähdä sen vaikutus ennustetarkkuuteen. On hyvin mahdollista, että käytetty rajaus ei tuottanut parasta mahdollista ennustetarkkuutta. Myös twiittien prosessointitavan ja twiiteistä prosessoitavien ominaisuuksien vaikutukseen liittyen voisi tehdä jatkotutkimusta. Kokeellisessa osassa käytettiin valmista TextBlob -kirjastoa sentimenttianalyysien tekemiseen, eikä muita vaihtoehtoja tarkemmin kokeiltu.

Eräs erittäin potentiaalinen jatkotutkimusmahdollisuus olisi rakenteellisen ja rakenteettoman tiedon yhdistämistapa. Tässä kokeellisessa osassa yhdistettiin tiedot prosessoimalla rakenteetonta tietoa siten, että siitä saatiin rakenteellista tietoa käytettäväksi. Eräs vaihtoehto olisi ollut yhdistää tarkat bitcoinin arvot yksittäisiin twiitteihin ja siten syöttää tällaisia tietokokonaisuuksia neuroverkoille. Rakenteettomana tietona voitaisiin käyttää lisäksi jotain muutakin dataa, esimerkiksi uutisdataa, jos haluttaisiin tutkia eri asioiden vaikutusta bitcoinin arvon liikkeisiin.

Kokeellisessa osassa valittiin tutkittavaksi pelkästään bitcoinin arvon suunnan ennustamista seuraavalle vuorokaudelle, koska kaikkien kokeellisessa osassa käytettyjen tietojen hankkiminen vuorokausittain oli helpointa ja yksinkertaisinta. Ennusteen ajanjaksoksi olisi voitu valita esimerkiksi jokin lyhyempikin ajanjakso, kuten tunti. Tällainen ennuste voisi olla käyttökelpoinen, varsinkin jos ajatellaan sijoittajia. Ennusteen

ajanjakso onkin myös eräs potentiaalinen jatkotutkimusmahdollisuus. On kuitenkin haasteellista arvioida, että millaisia tuloksia esimerkiksi tuntikohtaisesta ennusteesta saataisiin.

Käytettävien neuroverkkojen osalta jatkotutkimusta voisi tehdä käytännössä loputtomasti. Erilaisia neuroverkkotyyppejä voisi kokeilla, erilaisia malleja voisi rakentaa ja kaikkia neuroverkkoihin syötettäviä parametreja voisi muuttaa. Neuroverkkoihin syötettävistä parametreista erityisesti erilaisia aktivointifunktioita, optimointialgoritmeja, virheen laskentakaavoja, käytettävien opetussyklien lukumääriä ja yksittäisten kerroksien neuronien lukumääriä voisi kokeilla. Myös opetusdatan jakamista erikokoisiin osiin voisi tutkia. Tämän tutkielman kokeellisessa osassa valittiin käytettäväksi kolmea, neljää ja viittä vuorokautta opetusdatan osien kokona. Tuloksien perusteella parhaita tuloksia saatiin, kun neuroverkkoihin syötettiin kerrallaan viiden vuorokauden mittaisia datakokonaisuuksia. Siksi olisi mielenkiintoista tutkia, miten esimerkiksi kymmenen vuorokauden mittaiset datakokonaisuudet toimisivat bitcoinin arvon suunnan ennustamisessa.

Kokonaisuudessaan voidaan todeta, että tämän tutkielman kokeellinen osa oli melko tarkasti rajattu, ja siten käytännössä kaikkiin rajauksiin liittyen voisi tehdä jatkotutkimusta. Kokeellisen osan tuloksiin ollaan melko tyytyväisiä, sillä bitcoinin arvon suunnan ennustetarkkuus 0.791 on kryptovaluutan tapauksessa kelvollinen ja käytännössä hyödyllinen.

Viitteet

[1] L. Kristoufek, ”BitCoin meets Google Trends and Wikipedia: Quantifying the relationship between phenomena of the Internet era,” 2013.

[2] P. Ciaian, M. Rajcaniova ja d. Kancs, ”The economics of BitCoin price formation,” *Applied Economics,* 13 Marraskuu 2015.

[3] C. Taylor, ”Structured vs. Unstructured Dara,” Datamation, 28 Maaliskuu 2018. [Online]. Available: https://www.datamation.com/big-data/structured-vs-unstructured-data.html. [Haettu 9 Tammikuu 2019].

[4] M. Knight, ”What is Data Quality?,” Datawersity, 20 Marraskuu 2017. [Online]. Available: https://www.dataversity.net/what-is-data-quality/. [Haettu 23 Huhtikuu 2019].

[5] K. Ghassan ja A. Elli, Bitcoin and Blockchain Security, Norwood, MA: Artech House, 2016.

[6] Prasos Oy, ”Bittiraha.fi,” Prasos Oy, [Online]. Available: https://bittiraha.fi/content/bitcoinin-tekninen-kuvaus. [Haettu 23 Tammikuu 2019].

[7] Bitfinex, ”Bitfinex REST General,” [Online]. Available: https://docs.bitfinex.com/v2/docs/rest-general. [Haettu 12 Tammikuu 2019].

[8] J. Steiner, ”Web 3.0’s Crypto Winter Mission: Keep Our Heads Above the Hype,” coindesk, 18 Tammikuu 2018. [Online]. Available: https://www.coindesk.com/web-3-0s-crypto-winter-mission-keep-our-heads-above-the-hype. [Haettu 12 Tammikuu 2019].

[9] NewsBTC, ”January 2018: Cryptocurrency Bloodbath,” NewsBTC, 1 Helmikuu 2018. [Online]. Available: https://www.newsbtc.com/2018/02/01/january-2018-cryptocurrency-bloodbath/. [Haettu 12 Tammikuu 2019].

[10] NewsBTC, ”What Could Have Caused The Year’s Biggest Crypto Crash?,” NewsBTC, 21 Marraskuu 2018. [Online]. Available: https://www.newsbtc.com/2018/11/21/why-have-crypto-markets-fallen-so-hard-this-week/. [Haettu 12 Tammikuu 2019].

[11] Prasos Oy, ”Bittiraha.fi,” Prasos Oy, Marraskuu 2018. [Online]. Available: https://bittiraha.fi/content/viikkokatsaus-472018-kryptomarkkinat-romahtivat. [Haettu 12 Tammikuu 2019].

[12] N. Sanders, Forecasting Fundamentals, New York: Business Expert Press, 2017.

[13] ”Dense, Keras Documentation,” [Online]. Available: https://keras.io/layers/core/#dense. [Haettu 17 Huhtikuu 2019].

[14] L. Dey, H. Meisheri ja I. Verma, ”Predictive Analytics with Structured and Unstructured data - A Deep Learning based Approach,” tekijä: *IEEE Intelligent Informatics Bulletin*, New Delhi, India, 2017.

[15] D. R. Pant, P. Neupane, A. Poudel, A. K. Pokhler ja B. K. Lama, ”Recurrent Neural Network Based Bitcoin Price Prediction by Twitter Sentiment Analysis,” tekijä: *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*, Kathmandu, Nepal, 2018.

[16] T. Phaladisailoed ja T. Numnonda, ”Machine Learning Models Comparison for Bitcoin Price Prediction,” tekijä: *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, Kuta, Indonesia, 2018.

[17] C. Lamon, E. Nielsen ja E. Redondo, ”Cryptocurrency Price Prediction Using News and Social Media Sentiment,” http://cs229.stanford.edu/proj2017/final-reports/5237280.pdf, 2017.

[18] S. Colianni, S. Rosales ja M. Signorotti, ”Algorithmic Trading of Cryptocurrency Based on Twitter Sentiment Analysis,” http://cs229.stanford.edu/proj2015/029_report.pdf.

[19] S. Bird, E. Klein ja E. Loper, Natural Language Processing with Python, O'Reilly Media, 2009.

[20] S. Gupta, ”Sentiment Analysis: Concept, Analysis and Applications,” Towards Data Science, 7 Tammikuu 2018. [Online]. Available: https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17. [Haettu 24 Huhtikuu 2019].

[21] S. Loria, ”TextBlob: Simplified Text Processing,” 21 Marraskuu 2018. [Online]. Available: https://textblob.readthedocs.io/en/dev/. [Haettu 17 Helmikuu 2019].

[22] N. Project, ”Natural Language Toolkit,” NLTK Project, 2019. [Online]. Available: http://www.nltk.org/. [Haettu 19 Maaliskuu 2019].

[23] Medium, ”Deep Learning for NLP: An Overview of Recent Trends,” Medium, 24 Elokuu 2018. [Online]. Available: https://medium.com/dair-ai/deep-learning-for-nlp-an-overview-of-recent-trends-d0d8f40a776d. [Haettu 24 Huhtikuu 2019].

[24] A. Géron, Hands-on Machine Learning with Scikit-Learn and TensorFlow, O'Reilly Media , 2017.

[25] Google, ”Käänteinen kuvanhaku,” 2018. [Online]. Available: https://support.google.com/websearch/answer/1325808?hl=fi. [Haettu 29 Joulukuu 2018].

[26] A. Inc, ”Apple Siri,” 2018. [Online]. Available: https://support.apple.com/fi-fi/HT204389. [Haettu 29 Joulukuu 2018].

[27] A. Kattan, Z. W. Geem ja R. Abdullah, ”Artificial Neural Network Training and Software Implementation Techniques,” Hauppauge, N.Y., Nova Science Publishers, Inc., 2011.

[28] S. Sharma, ”Activation Functions in Neural Networks,” Towards Data Science, 6 Syyskuu 2017. [Online]. Available: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6. [Haettu 20 Huhtikuu 2019].

[29] D. Lipika, M. Hardik ja V. Ishan, ”Predictive Analytics with Structured and Unstructured data - A Deep Learning based Approach,” New Delhi, India, 2017.

[30] A. Jaokar, P. Katsande ja V. Mehendiratta, ”Recurrent neural networks, Time series data and IoT – Part One,” Posted by Ajit Jaokar, 2015.

[31] S. J. Kwon, Artificial Neural Networks, New York: Nova Science Publishers, 2011.

[32] V. Bushaev, ”Adam—latest trends in deep learning optimization.,” Towards Data Science, 22 Lokakuu 2018. [Online]. Available: https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c. [Haettu 20 Huhtikuu 2019].

[33] M. Binieli, ”Machine learning: an introduction to mean squared error and regression lines,” freeCodeCamp, 15 Lokakuu 2018. [Online]. Available: https://medium.freecodecamp.org/machine-learning-mean-squared-error-regression-line-c7dde9a26b93. [Haettu 20 Huhtikuu 2019].

[34] B. Manjubala ja K. G. Neeraj, Artificial Neural Network Applications for Software Reliability Prediction, John Wiley & Sons, Incorporated, 2017.

[35] ”BitInfoCharts,” 2019. [Online]. Available: https://bitinfocharts.com/comparison/bitcoin-tweets.html. [Haettu 14 Helmikuu 2019].

[36] P. S. Foundation, ”GetOldTweets3 0.0.9,” Joulukuu 2018. [Online]. Available: https://pypi.org/project/GetOldTweets3/. [Haettu 10 Helmikuu 2019].

[37] I. Twitter, ”Twitter Help Center,” Twitter, Inc., 2019. [Online]. Available: https://help.twitter.com/en/using-twitter/top-search-results-faqs. [Haettu 18 Huhtikuu 2019].

[38] C. R. Center, ”pattern.en,” CLiPS Research Center, 22 Kesäkuu 2018. [Online]. Available: https://www.clips.uantwerpen.be/pages/pattern-en. [Haettu 19 Maaliskuu 2019].

[39] P. S. Foundation, ”Python Documentation,” Python Software Foundation, 18 Maaliskuu 2019. [Online]. Available: https://docs.python.org/2/library/. [Haettu 18 Huhtikuu 2019].

[40] ”TensorFlow,” Google, [Online]. Available: https://www.tensorflow.org/. [Haettu 17 Huhtikuu 2019].

[41] ”Keras Documentation,” [Online]. Available: https://keras.io/. [Haettu 17 Huhtikuu 2019].

[42] T. p. d. team, ”Pandas Documentation,” The pandas development team, 12 Maaliskuu 2019. [Online]. Available: https://pandas.pydata.org/pandas-docs/stable/. [Haettu 18 Huhtikuu 2019].

[43] J. Brownlee, ”How to Develop LSTM Models for Time Series Forecasting,” Machine Learning Mastery, 14 Marraskuu 2018. [Online]. Available: https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/. [Haettu 17 Huhtikuu 2019].

[44] J. Brownlee, ”Stacked Long Short-Term Memory Networks,” Machine Learning Mastery, 18 Elokuu 2017. [Online]. Available: https://machinelearningmastery.com/stacked-long-short-term-memory-networks/. [Haettu 18 Huhtikuu 2019].

[45] ”Getting started with the Keras Sequential model,” [Online]. Available: https://keras.io/getting-started/sequential-model-guide/. [Haettu 17 Huhtikuu 2019].