Henrik Seppänen

KONEOPPIMISEN HYÖDYNTÄMINEN KYBERHYÖKKÄYSTEN HAVAITSEMISESSA JA TORJUNNASSA

JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2020

TIIVISTELMÄ

Seppänen, Henrik
Koneoppimisen hyödyntäminen kyberhyökkäysten havaitsemisessa ja torjunnassa
Jyväskylä: Jyväskylän yliopisto, 2020, 29s.
Tietojärjestelmätiede, kandidaattitutkielma

Tekoäly on noussut yhdeksi nykypäivän puhutuimmista uusista teknologioista. Tekoälyyn kuuluva koneoppiminen on ollut jo kauan tutkimuksen kohteena. Lähivuosina tekoälyn noustessa pinnalle, on myös koneoppimisteknologian hyötyjä ruvettu tutkimaan entistä enemmän eri aloilla. Tämän kanditutkielman tarkoituksena on ollut perehtyä koneoppimiseen, kyberturvallisuuteen sekä koneoppimisteknologian käyttöön kyberturvallisuuden kontekstissa. Tutkielmassa esitellään tapoja, joilla erityisesti yritysten kyberturvallisuutta voidaan parantaa koneoppimisteknologiaa käyttämällä. Tutkielmassa käydään läpi koneoppimismenetelmät, niiden yleisimmät haasteet sekä kyberturvallisuuden ja kyberhyökkäyksen määritelmät. Suurimmat hyödyt koneoppimisen käytöstä kyberturvallisuuden kontekstissa ovat laajojen datamäärien analysointi sekä kyberturvallisuudessa työskentelevien työntekijöiden työmäärän vähentäminen. Haasteiksi koneoppimisen käytöstä kyberturvallisuudessa löydettiin varsinkin laadukkaiden data-aineistojen saatavuus koneoppimismallien kouluttamiseen ja oikean koneoppimismenetelmän valinta haluttuun käyttötarkoitukseen. Myös kyberhyökkäystapojen jatkuva muutos sekä hyökkäysten jatkuva kehittyminen aiheuttavat haasteita koneoppimisen käytölle kyberhyökkäysten torjunnassa.

Asiasanat: Koneoppiminen, kyberturvallisuus, kyberhyökkäys

ABSTRACT

Seppänen, Henrik
Use of Machine Learning in detecting and preventing cyber-attacks.
Jyväskylä: University of Jyväskylä, 2020, 29pp.
Information Systems, Bachelor's Thesis
Supervisor: Riekkinen, Janne

Artificial intelligence has become one of the most talked about new technologies today in the field of Information Technology. Machine learning, which is part of artificial intelligence, has been for a long time a subject of research. In the last few years, when artificial intelligence has surfaced, the benefits of machine learning technology have also been explored in various fields. The purpose of this bachelor's thesis has been to get acquainted with machine learning, cyber security and the use of machine learning technology in the context of cyber security. The dissertation presents ways in which cyber security in particular in companies, can be improved by using machine learning technology. The dissertation reviews machine learning methods, their most common challenges, and the definitions of cybersecurity and cyber attack. The biggest benefits of using machine learning in the context of cybersecurity are the analysis of large amounts of data and the reduction of the workload of employees working in cybersecurity. The availability of high-quality data for training machine learning models and choosing the right machine learning method for the desired purpose were found to be challenges in the use of machine learning in cybersecurity. The constant change in cyber-attack methods and the constant development of attacks also pose challenges to the use of machine learning in the fight against cyber-attacks.

Keywords: Machine Learning, Cyber security, Cyber attack

KUVIOT

SISÄLLYS

1 JOHDANTO

Tekoäly, jonka yksi osa-alue on koneoppiminen, on ollut lähiaikoina erittäin paljon esillä niin julkisessa kuin akateemisessa keskustelussa. Koneoppiminen on tutkimusala, jossa tietokoneohjelmistot käyttävät algoritmeja rakentaakseen matemaattisen mallin, jonka avulla ne voivat suorittaa tehtäviä ilman täsmällisiä ohjeita, turvautuen sen sijaan päättelyyn ja datasta oppimiseen. Koneoppimisessa koneoppimisalgoritmit rakentavat matemaattisen mallin, joka perustuu mallidataan tai toisin sanoen ”harjoitusdataan”, tehdäkseen ennustuksia ja päätöksiä datasta olematta selkeästi ohjelmoituja tähän tehtävään. (Samuel, 1959.)

Myös kyberturvallisuus on lähiaikoina noussut julkisen keskustelun kohteeksi, kun kyberhyökkäyksiä on kohdistettu valtiollisiin toimijoihin ja toimielimiin. Kyberhyökkäysten määrän kasvu, tekotapojen laajuus sekä hyökkäysten jatkuva kehittyminen ovat suurena haasteena kybertuvallisuuden kentässä. Esimerkiksi vuonna 2017 tehdyssä kyberturvallisuusalan työntekijätutkimuksessa ennustettiin vuonna 2022 Euroopassa olevan 350 000- ja kansainvälisesti 1,8 miljoonan työntekijän vajaus kyberturvallisuusalalla (”2017 Global Information Security Workforce Study”, 2017). Koneoppimisteknologioiden käyttöä pidetäänkin yhtenä auttavista tekijöistä kyberturvallisuuden mahdollistajana tulevaisuudessa. Koneoppimisteknologiat pystyvät käsittelemään suuria määriä tietoa sekä oppimaan ja muokkaamaan toimintaansa saadun tiedon pohjalta ja näin ne voivat toimia osana helpottamaan kyberturvallisuuden työtaakkaa.

Tämä tutkielma käsittelee koneoppimisen hyödyntämistä kyberhyökkäysten havaitsemisessa sekä torjunnassa. Tutkielmassa tutkin, mitä erilaisia tapoja koneoppimismenetelmien käytölle on kyberturvallisuusjärjestelmissä sekä miten ne auttavat parantamaan yritysten kyberturvallisuutta.

Tutkielmalla oli seuraavat kolme tutkimuskysymystä:

- Miten koneoppimista voidaan hyödyntää kyberhyökkäysten havaitsemisessa ja torjunnassa?
- Millaisia hyötyjä ja haittoja sisältyy koneoppimista käyttäviin tietoturvajärjestelmiin?

- Millaisia koneoppimisteknologioita tietoturvajärjestelmissä voidaan käyttää?

Tutkielma suoritettiin kirjallisuuskatsauksena. Suurin osa lähteistä haettiin Google Scholar, IEEE Xplore ja JYKDOK-tietokannoista. Tutkielman rakenne on jaettu kolmeen eri sisältölukuun. Johdannon jälkeinen luku käsittelee koneoppimista sekä koneoppimisen keskeisiä haasteita. Kolmas luku käsittelee kyberturvallisuutta, kyberhyökkäysten luokittelutapoja sekä kyber, tieto- ja ICT-turvallisuuden eroja. Neljäs sisältöluku keskittyy koneoppimisen käyttöön kyberturvallisuuden kontekstissa ja esittelee erilaisia kyberturvallisuusjärjestelmiä sekä koneoppimisen hyödyntämistä näissä järjestelmissä.

2 KONEOPPIMINEN

Tässä luvussa käydään läpi koneoppimisen taustoja ja historiaa, koneoppimisen käyttöön liittyviä keskeisiä lähestymistapoja ja haasteita. Luku 2.1 sisältää koneoppimisen määritelmän ja lyhyesti koneoppimisen historiaa. Luku 2.2 käy läpi koneoppimismenetelmät ja miten menetelmien valinta eroaa käyttötarkoituksien mukaan. Luku 2.3. käsittelee koneoppimisen keskeisiä haasteita.

2.1 Koneoppimisen määritelmä

Jotta tietokoneella voidaan ratkaista jokin ongelma, on siihen oltava algoritmi. Algoritmi kertoo tietokoneelle sarjan yksityiskohtaisia käskyjä, jotka suorittamalla saadaan haluttu lopputulos. Joihinkin ongelmiin ei kuitenkaan pystytä rakentamaan algoritmia. Esimerkiksi roskapostiviestien tunnistaminen oikeista viesteistä on vaikeaa, koska roskapostin määritelmä on erilainen jokaisella käyttäjällä. Tilanteissa, joissa ongelmaa ei pystytä ratkaisemaan suoraan rakentamalla sille algoritmia, voidaan sen sijaan hyödyntää esimerkkidataa ja koneoppimista. Koneoppimisessa tietokoneelle rakennetaan algoritmi, jota harjoitetaan esimerkkidatan pohjalta. Harjoitettua algoritmia kutsutaan malliksi. Malli pystyy tekemään hyödyllisiä ennustuksia ja havaintoja jostakin data-aineistosta sekä muokkaamaan algoritmiaan esimerkkidatan tai kokemuksensa pohjalta. Näin malli pystyy tehostamaan toimintaansa ja näin antaa tietokoneelle mahdollisuuden oppia itsenäisesti. (Alpaydin, 2010.) Mitchell (1997) on kirjoittanut seuraavan, nykyään paljon käytetyn määritelmän koneoppimiselle:

> Computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Tietokoneohjelman voidaan siis sanoa olevan oppiva, jos sen suorituskyky P paranee joissain tehtäväluokissa T kokemuksen E avulla.

Yksi ensimmäisistä menestyksekkäistä koneoppimismenetelmistä oli Arthur Samuelin tammea pelaava tietokoneohjelmisto. Samuelin rakentamaa itse-

näisesti oppivaa ohjelmaa pidetään yhtenä ensimmäisistä onnistuneista koneoppimisohjelmistoista, ja samalla hän esitteli ja vakiinnutti termin koneoppiminen tietojenkäsittelytieteiden alalla (Michalski, Carbonell & Mitchell, 2013, 14). Jo 90-luvun lopulla koneoppimismenetelmät pystyivät tunnistamaan puhuttuja sanoja, ennustamaan keuhkokuumepotilaiden toipumisastetta, havaitsemaan luottokorttien väärinkäyttöä, ajamaan autonomisia ajoneuvoja julkisilla teillä sekä pelaamaan erilaisia lautapelejä maailman parhaiden pelaajien tasolla (Mitchell 1997, 2). Nykyään koneoppimismenetelmiä käytetäänkin hyvin monessa eri kontekstissa. Kyberturvallisuuteen liittyviä käyttötapoja ovat esimerkiksi haitallisten ohjelmien havaitseminen ja erityisesti verkkotunkeutumisten havaitseminen. Näihin esimerkkeihin palataan tarkemmin luvussa 4.

2.2 Koneoppimismenetelmät

Koneoppimismenetelmä kertoo tavan, jota koneoppimisalgoritmin harjoittamiseen on käytetty. Menetelmän valintaan vaikuttaa ratkaisevasti saatavilla oleva esimerkkiaineisto sekä mallin käyttötarkoitus. Koneoppimismenetelmiä on esitelty kirjallisuudessa monia, ja menetelmät voidaan luokitella neljään eri kategoriaan. Yleisimmät ja eniten käytetyt menetelmät ovat ohjatut (supervised) sekä ohjaamattomat (unsupervised) menetelmät (Zhu & Goldberg, 2009). Puoliohjatut (semi-supervised) sekä vahvistusoppimisen (reinforcement learning) menetelmät sijoittuvat edellä mainittujen välille.

Ohjatussa oppimisessa käytetään nimiöitä (label) luokittelemaan harjoitusdatan harjoitusesimerkkien oikeat vastaukset joidenkin harjoitusdatan piirteiden (features) pohjalta. Ohjatun oppimismenetelmän käyttö vaatii sen, että jokainen harjoitusdatan esimerkin vastaus voidaan luokitella tietyksi nimiöksi. Oikeiden ja väärien tulosten määritteleminen harjoitusdatasta vaatii ihmisen osallistumista, jota sanotaan algoritmin harjoittamisen ”ohjaamiseksi”. Tästä menetelmä on saanut nimensä ohjattu oppiminen. Algoritmi opetetaan siis ohjatusti tuottamaan malli, jota se voi hyödyntää myös uuden, samankaltaisen datan analysoinnissa. (Portugal, Alencar & Cowan, 2017.) Esimerkki ohjatusta oppimisesta on erilaisten eläinlajien tunnistaminen kuvista. Tällöin algoritmin käyttää aiemmin oppimaansa tietoa eläinten piirteistä eläinlajin luokittelun pohjana.

Ohjaamaton oppiminen on nimensä mukaisesti ohjatun oppimisen vastakohta. Ohjaamattomassa oppimisessa dataa ei voida luokitella nimiöillä, eikä varsinaista harjoitusdataa tai oikeita vastauksia ole saatavilla. Harjoitettu malli luokittelee data-aineiston luokkiin alkioiden mukaan, jotka muistuttavat toisiaan enemmän kuin muiden luokkien alkiot. Ohjaamatonta oppimista käytetään yleisesti etsittäessä suurista datamääristä piilossa olevia rakenteita tai eroavuuksia, joita hyödynnetään ennustuksien ja päätelmien teossa. (Shalev-Shwartz & Ben-David, 2014, 4.) Esimerkiksi suurista määristä käyttäjätietoa voidaan etsiä yhteneväisyyksiä ja käyttäjät pystytään näinluokittelemaan esi-

merkiksi erilaisiin persoonallisuuskategorioihin, joiden avulla voidaan tarkentaa mainoksia tai etsiä poliittisia kantoja (Portugal ym. 2017).

Puoliohjatuissa oppimismenetelmissä koneoppimisalgoritmilla on yleensä käytössään vain hiukan nimiöityä harjoitusdataa, ja loput datasta on nimiöimätöntä. Koska datan nimiöimiseen vaaditaan ihmisen työpanos, voi koko harjoitusdatan nimiöiminen olla mahdotonta käsiteltävän datamäärän laajuuden vuoksi. Tällaisissa tilanteissa voi puoliohjatulla oppimismenetelmällä olla suuri käytännöllinen arvo. Tämän lähestymistavan on huomattu parantavan ohjaamattoman oppimisen oppimistarkkuutta huomattavasti, koska algoritmilla on parempi käsitys toivotusta toiminnasta. (Zhu, 2005.)

Vahvistusoppimista voidaan käyttää, kun harjoitusdatan nimiöitä ei ole saatavilla, mutta algoritmi pystytään harjoittamaan annetun palautteen pohjalta. Kun algoritmi tekee oikean päätöksen, se saa positiivisen palautteen ja väärän päätöksen kohdalla negatiivisen palautteen. Algoritmi pyrkii tekemään päätöksiä, jotka johtavat positiiviseen palautteeseen. Tämä taas johtaa toivottuun lopputulokseen. (Barto & Dietterich, 2004.)

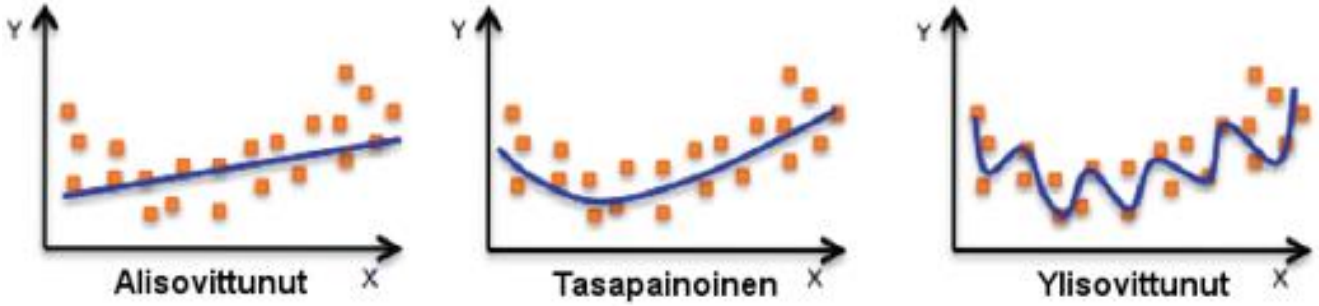
2.3 Keskeisiä haasteita koneoppimisessa

Kyberturvallisuusalan ollessa jatkuvasti muuttuvien hyökkäysten kohteena, on koneoppimisteknologian käyttö haastavampaa kyberturvallisuuden kontekstissa, kuin muissa yleisesti käytetyissä applikaatioissa. Buczak ja Guvenin (2016) mukaan kyberturvallisuusapplikaatioissa käytettävien koneoppimisjärjestelmien käyttämää harjoitusdataa joudutaan muokkaamaan jatkuvasti sekä järjestelmä joudutaan ”kouluttamaan” uudestaan päivittäin, aina kun järjestelmän käyttäjä katsoo sen välttämättömäksi tai sen käyttämä malli saadaan selville. Tämä muodostaa ongelman kouluttamisen kestossa, jonka on tällöin oltava vähemmän kuin yksi päivä. Myös laadukkaan harjoitusdatan kerääminen on vaikeaa tietoliikenneverkoista, koska datan määrä on päivittäin niin suuri, että sen varastointi on vaikeaa ja sen analysointi on hidasta (Buczak & Guven, 2016.) Useat koneoppimisteknologian haasteet liittyvätkin laadukkaan datan keräämiseen ja sen riittävään määrään. Seuraavaksi käsitellyt ongelmat korostuvatkin erityisesti pieniä aineistoja käytettäessä (Pasupa & Sunhem, 2016; Joo Er, Kashyap & Wang, 2016).

2.3.1 Yli- ja alisovittaminen

Koneoppimismallin tulisi aina suoriutua mahdollisimman hyvin sille tuntemattoman aineiston käsittelystä. Jos malli suoriutuu hyvin harjoitusdatan käsittelystä, mutta sen suorituskyky laskee siirryttäessä tuntemattomaan aineistoon, on malli ylisovittunut (overfitted) (Myung, 2000). Ylisovittunut malli on harjoitettu vastaamaan liian hyvin harjoitusdatassa ilmeneviin piirteisiin, jolloin malli

on käytännössä oppinut harjoitusdatan ulkoa. Malli on myös yleensä kehittynyt tarpeettoman monimutkaiseksi. Tällöin mallissa otetaan huomioon harjoitusdatassa esiintyviä yksityiskohtia, jotka eivät ole oppimisen kannalta oikeita tai dataa luokitellaan väärin perustein. Malli ei näin opi yleistettävää tapaa, jota se voisi käyttää tuntemattoman aineiston käsittelemiseen, jolloin mallin yleistettävyys on huonoa. (Dietterich, 1995.) Alisovitetussa (underfitted) mallissa halut-



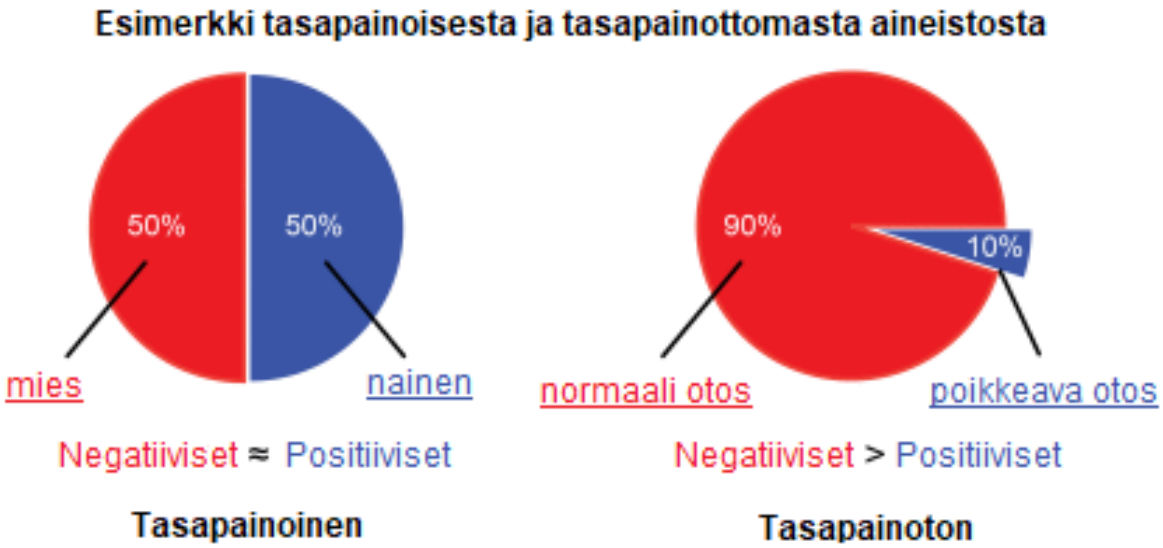
tuja lopputuloksia ei tunnisteta tarpeeksi tarkasti, koska malli on liian yksinkertainen annetun aineiston käsittelyyn. Tällöin sekä harjoitusdatan, että tuntemattoman aineiston käsittelyssä esiintyy liiallinen määrä virheitä. Malli yleistää harjoitusdataa liikaa eikä osaa tunnistaa haluttuja rakenteita tai tietoja datasta (van der Aalst ym. 2010).

Kuvio 1 Eri tavoilla sovittuneet mallit samaa aineistoa käytettäessä (suom. Amazon, 2020)

2.3.2 Harjoitusaineiston tasapainottomuus

Kun koneoppimismallin harjoitusaineistossa on jotakin näytettä toisia näytteitä enemmän, on aineisto tasapainoton. Tällaisia aineistoja esiintyy esimerkiksi lääketieteen applikaatioissa tai luottokorttipetoksia tutkittaessa, joissa suurin osa otetuista näytteistä tai havainnoista on negatiivisia.

Davis ja Goadrich (2006) ovat tutkineet koneoppimismallin suorituskykyä, kun harjoitusdatassa esiintyvät näytteet eivät ole tasapainossa. Koska positiivisten havaintojen määrä on hyvin pieni aineiston kokoon nähden, voi koneoppimismalli luokitella kaikki havainnot negatiiviseksi ja näin saavuttaa virheellisen korkean suorituskyvyn, vaikka malli ei osaisi käsitellä aineistoa millään tapaa. Datassa voi siis esiintyä suuri määrä negatiivisia lopputuloksia, mutta havaitsematta jäänyt positiivinen lopputulos tekee mallista käyttökelvottoman.



Kuvio 2 Havainnollistus tasapainoisen ja tasapainottoman aineiston erosta (suom. Tripathi, 2019)

2.3.3 Ulottuvuuden kirous

Yksi keskeisimmistä ohjaamattomaan koneoppimiseen liittyvistä ongelmista on ulottuvuuden kirous (curse of dimensionality) (Zanero & Serazzi, 2008). Ongelma on ollut esillä jo koneoppimisteknologian alkuajoista lähtien ja ongelma on havaittu jo vuonna 1957 (Bellman, 1957). Ulottuvuuden kirouksen mukaan piirteiden määrän kasvaessa, mallin tarvitsema datamäärä kasvaa suhteessa eksponentiaalisesti piirteiden lukumäärän kanssa (Har-Peled, Indyk & Motwani, 2012). Zanero ja Serazzi (2008) muistuttavat, että vaikka piirteiden lukumäärää kasvattamalla voidaan dataa kuvata tarkemmin ja yksityiskohtaisemmin, mallin toiminta hidastuu ja ennustustarkkuus heikkenee myös samassa suhteessa.

2.3.4 Piirteiden- ja mallin valinta

Kuten luvussa 2.2 kerrottiin, aineistosta tehdyt havainnot voidaan luokitella toisistaan piirteiden avulla. Monissa koneoppimisongelmissa piirteiden määrä mallissa on hyvin suuri, jolloin ulottuvuuden kirous haittaa mallin toimintaa. Piirteiden valinta on prosessi, jossa mallin käyttöön valitaan vain tarpeelliset piirteet, jolloin mallin suorituskyky saadaan maksimoitua. Oikein tehty piirteiden valinta voikin olla ratkaiseva tekijä koneoppimismallin onnistuneelle toiminnalle. (Kroon & Whiteson, 2009.)

Oikean koneoppimismallin valinta on tärkeää, jotta aineiston käsittely ja havaintojen luokittelu on mahdollisimman tehokasta ja oikeaa. Paras lopputulos mallin valintaan saadaan, kun käytettävissä on ”riittävä” määrä dataa. Vaa-

dittava määrä voi kuitenkin olla lähes ääretön riippuen mallin kompleksisuudesta. Ihanteellisessa tapauksessa data jaetaan harjoitus-, varmistus- ja testidataan. Malleja, joiden uskotaan olevan ongelmaan sopivia, testataan harjoitusdatalla, toiminta varmistetaan varmistusdatalla ja lopulta mallin yleistettävyys testataan testidatalla. Näin eri malleja saadaan vertailtua keskenään ja voidaan valita tehtävästä parhaiten suoriutuva malli. Tämä on kuitenkin yleisesti epäkäytännöllinen tapa mallin valintaan, koska tarvittavaa määrää dataa on harvoin saatavilla. (Hastie, Tibshirani & Friedman, 2011.)

Todennäköisyyteen pohjautuvat valintamenetelmät arvioivat malleja vain perustuen niiden suoriutumiseen harjoitusdatasta sekä mallin kompleksisuudesta. Tällöin mallin valintaan riittävä määrä dataa on pienempi, koska mallia ei tarvitse arvioida testidatalla. Koska koneoppimismallien halutaan myös olevan hyvin yleistettävissä, valinnassa suositaan malleja, jotka ovat vähemmän kompleksisia. Todennäköisyyteen perustuvia menetelmiä voidaan kuitenkin käyttää vain, jos mitattava malli on lineaarisesti- tai logistisesti regressiivinen (Bishop, 2006, 32-33.) Uudelleenotantamenetelmissä mallin suorituskyky voidaan mitata pelkän testidatan perusteella. Menetelmässä testidata jaetaan aliluokkiin, joiden avulla malli voidaan sekä harjoittaa, että testata. Prosessi voidaan toistaa useita kertoja ja malli arvioidaan jokaisen testikierroksen tulosten keskiarvon perusteella. Uudelleenotantamenetelmät ovat yleisesti käytetyimpiä mallien valinnassa, niiden helpon käytön ja oppimisen vuoksi. (Good, 2005, xiii).

3 KYBERTURVALLISUUS

Tässä luvussa käsitellään kyberturvallisuuden keskeisiä käsitteitä ja niiden eroja, kyberhyökkäyksiä ja niiden luokittelutapoja sekä erilaisia kyberuhkien muotoja. Luku 3.1 käsittelee kyberturvallisuudessa käytettyjä keskeisiä käsitteitä ja niiden eroja. Luvussa 3.2. käydään läpi erilaisia kyberhyökkäyksiä ja niiden luokittelutapoja.

3.1 Kyber-, tieto- ja ICT-turvallisuuden erot

Samalla kun kyberturvallisuustermistön käyttö on lisääntynyt julkisessa keskustelussa, on sen merkitys myös sekoittunut muun muassa termien, kuten ICT- ja tietoturvallisuus kanssa (Schatz, Bashroush & Wall, 2017). Vaikka ICT-, tieto- ja kyberturvallisuustermejä käytetäänkin useasti samoissa asiayhteyksissä käsittävät ne kuitenkin pohjimmiltaan eri asioita. Schatz, ym. (2017) muistuttavatkin, että vaikka termien sekoittuminen ei ole erityisen haitallista informaalissa keskustelussa, voi se luoda huomattavia ongelmia formaaleissa asiayhteyksissä käytettäessä ja näin vaikuttaa jopa kansainvälisellä tasolla tehtyihin päätöksiin. Tämän takia termien erot ja yhtäläisyydet onkin hyvä määritellä.

Turvallisuudesta puhuttaessa tarkoitetaan aina jonkin omaisuuden suojaamista sen erilaisten luontaisten haavoittuvuuksien aiheuttamilta uhilta (Gerber & von Solms, 2006). Farnin, Linin & Fungin (2004) mukaan haavoittuvuuksista aiheutuvia riskejä voidaan vähentää luomalla turvallisuusprosesseja, jotka määrittävät eri turvallisuusrajoitteiden (ts. vastatoimien) valinnan sekä käyttöönoton.

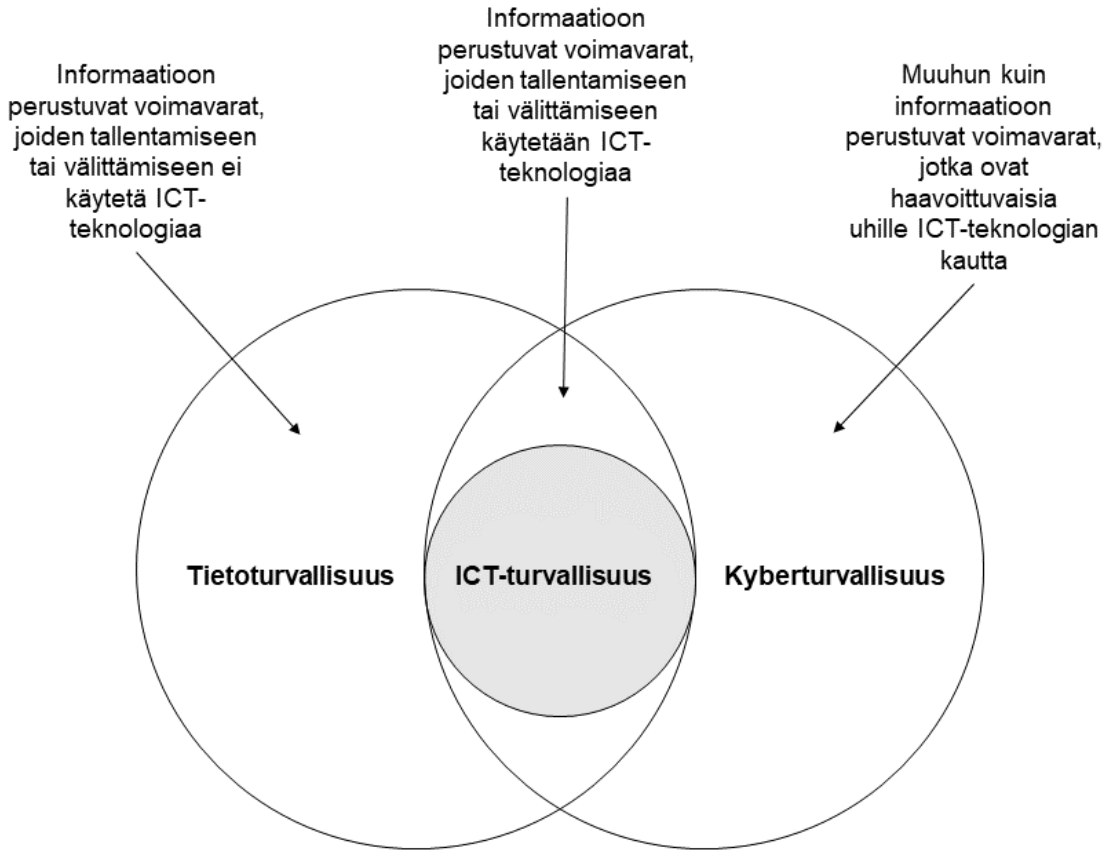
Tietoturvallisuus on nimensä mukaisesti tiedon sekä sen kriittisten elementtien suojaamista, ja se käsittää myös järjestelmät ja laitteet, jotka käyttävät, tallentavat tai siirtävät tietoa (Whitman ja Mattord, 2009, 8). Tietoturvallisuuden perustana on niin sanottu LES-kolmikko (eng. CIA-triad), joka sisältää seuraavat tietoturvallisuuden perusperiaatteet:

- Luottamuksellisuus (Confidentiality); Tieto on suojattu luvattomalta käytöltä.
- Eheys (Integrity); Tiedon sisältö ei saa sen elinaikanaan muuttua tahattomasti tai ulkopuolisen tahon toimesta.
- Saatavuus (Availability); Tieto täytyy turvata niin, että siitä ei tule saavuttamatonta väliaikaisesti tai lopullisesti ulkopuolisen tahon toimesta.

Tiedon, jonka halutaan olevan tietoturvallista, on siis täytettävä nämä vaatimukset. (Kotzanikolaou & Douligeris, 2007, 1; von Solms & van Niekerk, 2013.)

Kyberturvallisuus keskittyy taas kybertoimintaympäristön turvaamiseen, jonka Singer ja Friedman (2013) ovat määritelleet toiminnan alueeksi, joka koostuu tietokoneista (sekä niiden käyttäjistä) ja tietoverkoista, joissa informaatiota jaetaan, säilytetään ja joiden kautta on mahdollista kommunikoida (Singer & Friedman, 2013, 13). Koska kybertoimintaympäristö koostuu niin tietotekniikkaa käyttävistä laitteista, järjestelmistä, tietoverkoista sekä ihmisistä, on kyberturvallisuuden käsite myös yhteiskunnallisesti merkittävä. Von Solms ja van Niekerk (2013) määrittelevätkin tieto- ja kyberturvallisuuden eron siten, että tietoturvallisuuden ollessa organisaatiolle tai henkilölle tärkeän tiedon turvaamista, jossa ihminen on vain yksi osa tietoturvaprosessia, on kybertuvallisuus taas ihmisten ja yhteiskunnallisten palveluiden ja järjestelmien suojaamista uhilta, joille ne ovat altistuneet ICT-teknologian käytön myötä.

Jotta aiemmin määritellyt tietoturvallisuuden ja kyberturvallisuuden periaatteet voidaan saavuttaa, on myös laitteet ja verkot, joissa tietoa säilytetään ja jaetaan, suojattava hyökkäyksiltä. Käsite tieto- ja viestintäteknologian (lyh. ICT) turvallisuus onkin näiden fyysisten laitteiden ja verkkojen suojaamista. Tämän takia niin tieto-, kuin kyberturvallisuuden yhtenä osana voidaan pitää ICT-turvallisuutta (ks. kuvio 3). (von Solms & van Niekerk, 2013.)



Kuvio 3 Kyber-, tieto- ja ICT-turvallisuuden välinen suhde (suom. von Solms & van Niekerk, 2013, 101)

3.2 Kyberhyökkäykset

Kyberhyökkäysten tavoitteena on Hathawayn ym. (2012) mukaan tietokoneverkon toiminnan heikentäminen tai häiritseminen. Jotta hyökkäys voidaan laskea kyberhyökkäykseksi, on hyökkäyksellä oltava myös poliittinen tai kansalliseen turvallisuuteen vaikuttava päämäärä, Tämä määritelmä erittelee kyberhyökkäykset muista kyberuhista kuten kyberrikollisuudesta ja kybervakoilusta, koska tekijällä ei tällöin välttämättä ole poliittista, tai kansalliseen turvallisuuteen vaikuttavaa päämäärää eikä verkon toimintaa näissä tapauksissa heikennetä tai häiritä. Koska määritelmän mukaan kyberhyökkäykseksi voidaan laskea mikä vain verkon toimintaa haittaava toiminta, voidaan tällöin myös verkon fyysisiin osiin kohdistuva hyökkäys, kuten mannertenvälisen tietoliikenneyhteyden fyysinen katkaiseminen, laskea kyberhyökkäykseksi. (Hathaway ym., 2012.) Janssonin ja Sihvosen (2018) mukaan kyberhyökkäykset ja niiden räjähtävä kasvu ovat osittain Internetin ja sen infrastruktuurin avoimuuden ja demokraattisuuden mahdollistamia.

Tietokoneverkot ovat verkkoja, jotka koostuvat tietokoneista ja laitteista, jotka ovat yhdistettynä toisiinsa joko internetin, tai esimerkiksi organisaation sisäiseen käyttöön rajatun verkon välityksellä (Hathaway ym., 2012). Tietokoneverkkoon kohdistuva kyberhyökkäys vaatii aina jonkin haavoittuvuuden, jota käyttämällä verkkoon voidaan hyökätä (Lin, 2010). Haavoittuvuuksia kutsutaan myös hyökkäysvektoreiksi. Suosittuja hyökkäysvektoreita ovat Fraleyn ja Cannadyn (2017) mukaan muun muassa sähköpostin kautta lähetetyt tietojenkalasteluviestit, erilaiset haittaohjelmat ja hajautetut palvelunestohyökkäykset.

Koska tietokoneverkkoja vastaan voidaan hyökätä monella eri tavalla, ovat Hathaway ym. (2012) yleistäneet hyökkäysten erottelun syntaktisiin ja semanttisiin hyökkäyksiin. Syntaktisessa hyökkäyksessä verkkoon pyritään aiheuttamaan toimintahäiriöitä saastuttamalla verkossa olevan tietokoneen tai laitteen käyttöjärjestelmä jollakin haittaohjelmalla, esimerkiksi viruksella tai troijanhevosella. Syntaktisessa hyökkäyksessä verkon toimintahäiriöt ovat yleensä helposti havaittavissa. Semanttisessa hyökkäyksessä järjestelmän ja verkon toiminta kuitenkin vaikuttaa normaalilta, mutta sen käsittelemää ja tuottamaa informaatiota on jollakin tavalla muokattu. (Hathaway ym. 2012.)

4 KONEOPPIMINEN KYBERTURVALLISUUDESSA

Tietoverkkoihin kohdistuvien hyökkäysten sekä haittaohjelmien määrä on jokavuotisessa kasvussa, ja hyökkäykset kohdistuvat yhä useammin yrityksien tietoverkkoja ja laitteita vastaan (Malwarebytes, 2020; Symantec Corporation, 2019). Hyökkäysten tunnistamisesta ja niiltä suojautumisesta on tullut yhä vaikeampaa hyökkääjien kehittäessä yhä uusia tapoja harhauttaakseen kyberturvallisuusjärjestelmiä sekä virustorjuntaohjelmistoja. Tämä muutos yrityksiin kohdistuvien hyökkäysten kasvussa sekä käytössä olevien suojauskeinojen riittämättömyys vaativat uusien teknologioiden käyttöönottoa kyberturvallisuusjärjestelmissä. Koneoppimisteknologiat ovat yksi tulevaisuuden ehdotetuista teknologioista kyberturvallisuuden mahdollistajana ja koneoppimisteknologioita käytetäänkin jo nykyään monilla eri tavoilla kyberturvallisuuden kentässä. Seuraavissa luvuissa annetaan esimerkkejä erilaisista kyberturvallisuusjärjestelmistä sekä tavoista käyttää koneoppimista näiden järjestelmien apuna. Koneoppiminen ei kuitenkaan ole ratkaisu kaikkiin kyberturvallisuuden ongelmiin, ja koneoppimisen käytöllä on omat haasteensa myös kyberturvallisuuden kontekstissa. Näitä haasteita käydään läpi tarkemmin luvussa 4.4

4.1 Tietoverkkotunkeutumisten havaitseminen

Organisaatioiden kyberturvallisuusjärjestelmän keskeisenä osana toimivat tunkeutumisen havaitsemisjärjestelmät (Intrusion Detection System, lyh. IDS). Buczakin ja Guvenin (2016) mukaan kyberturvallisuusjärjestelmässä toimivien tietokone- ja tietoverkkosuojausjärjestelmien tulee olla suojattuna IDS:llä, palomuurin ja antivirusohjelmiston ohella. IDS:t valvovat niin organisaation ulkoa, kuin sisältäkin tulevia hyökkäyksiä ja auttavat havaitsemaan, määrittelemään ja tunnistamaan tietojärjestelmiin kohdistuvaa luvatonta käyttöä, niihin tehtäviä muutoksia sekä tuhoamisyrityksiä (Mukkamala, Sung & Abraham, 2005). IDS:t voidaan yleisesti jakaa kolmeen eri kategoriaan niiden käyttämien toimintata-

pojen perusteella: väärinkäyttöön perustuvat- (misuse-based), anomalioihin perustuvat- (anomaly-based) ja hybridijärjestelmät (Buczak & Guven, 2016).

Väärinkäyttöön perustuva IDS tunnistaa haitallista toimintaa etukäteen määriteltyjen sääntöjen ja tunnusmerkkien perusteella. Menetelmää käytetään silloin, kun järjestelmään kohdistuvien hyökkäysten toimintatapa on tiedossa. Väärinkäyttöön perustuvat järjestelmät suoriutuvat hyvin tunnettujen hyökkäysten tunnistamisesta eivätkä ne tuota suurta määrää turhia hälytyksiä. (Buczak & Guven, 2016.) Menetelmä ei kuitenkaan osaa tunnistaa hyökkäystä, joka ei täysin vastaa tunnettua toimintaa. Väärinkäyttöön perustuvien järjestelmien hyöty onkin vähentynyt hyökkäysten kehittyessä, koska uusien hyökkäysten toimintatavat eivät ole tiedossa. Myös sääntöjen ja tunnusmerkistöjen sisältävää tietokantaa joudutaan päivittämään jatkuvasti uusia hyökkäyksiä havaitessa. (Khraisat, Gondal, Vamplew & Kamruzzaman, 2019.)

Anomalioihin perustuva IDS hyödyntää tietoverkon haitallisen toiminnan tunnistamiseen mallia, joka on tuotettu verkon normaalista toiminnasta. Järjestelmä vertaa tietoverkossa tapahtuvaa toimintaa normaalin toiminnan malliin ja luokittelee kaiken mallista poikkeavan toiminnan hyökkäyksiksi. (Buczak & Guven, 2016.) Khraisat ym. (2019) pitävät anomalioihin perustuvia järjestelmiä nykypäivänä hyödyllisempinä kuin väärinkäyttöön perustuvia järjestelmiä, koska analysoimalla kaikkea verkossa tapahtuvaa liikennettä, eikä vain verkkoon kohdistuvia hyökkäyksiä, voidaan myös verkon sisäisessä liikenteessä tapahtuva haitallinen toiminta havaitsemaan. Buczak ja Guven (2016) ovat myös havainneet menetelmän hyväksi puoleksi mahdollisuuden uniikeille malleille eri järjestelmissä, jolloin hyökkääjä ei voi olla varma minkä toiminnan järjestelmä havaitsee tai sallii. Heidän mukaansa menetelmän suurin heikkous on kuitenkin suuri määrä vääriä hälytyksiä, koska kaikki verkossa tapahtuva uusi toiminta, myös sallittu, luokitellaan hyökkäykseksi.

Hybridimenetelmään perustuva IDS on yhdistelmä anomalioihin sekä väärinkäyttöön perustuvia menetelmiä. Buczakin ja Guvenin (2016) mukaan useimmat käytössä olevista IDS:istä ovat hybridijärjestelmiä. Hybridijärjestelmässä anomaliamenetelmään perustuva osa järjestelmää erittelee verkkoliikenteestä poikkeuksellisen liikenteen, josta väärinkäyttömenetelmä luokittelee tunnistetut hyökkäykset tai päivittää käyttämiään sääntöjä ja tunnusmerkistöä, jos hyökkäys ei ollut vielä tunnettu.

Monet nykyisistä koneoppimisalgoritmeista soveltuvat käytettäväksi tunkeutumisten havaitsemisjärjestelmissä. Koneoppimisalgoritmien käytön tavoitteena on parantaa järjestelmien havaintotarkkuutta sekä vähentää tietotaidon tarvetta hyökkäysten havaitsemisessa (Khraisat ym., 2019). Oikean koneoppimisalgoritmin valinta on kuitenkin hyvin riippuvaista saatavilla olevasta datasta ja järjestelmän käyttökohteesta. Esimerkiksi Farid, Harbi ja Rahman (2010) rakensivat tutkimuksessaan hybridimenetelmään perustuvan IDS:n yhdistelemällä Naive Bayes- ja päätöspuualgoritmeja. Järjestelmää testattiin verkkotunkeutumisen havaitsemisjärjestelmille suunnitellulla aineistolla, josta järjestelmä pystyi havaitsemaan hyökkäykset jopa 99,63% tarkkuudella. Oikeiden koneop-

pimisalgoritmien avulla voidaan hybridimenetelmää käyttävä järjestelmä saada siis erittäin hyväksi apuvälineeksi hyökkäysten havaitsemisessa.

4.2 SIEM-järjestelmät

SIEM-järjestelmiä (Security information and event management systems) käytetään erityisesti suurikokoisissa tietoverkoissa analysoimaan verkon sisäisten järjestelmien tuottamaa dataa sekä lokitietoja. Järjestelmä voi analysoimansa datan pohjalta varoittaa henkilöstöä verkkoon kohdistuvasta hyökkäyksestä sekä tehdä oma-aloitteisia toimia estääkseen hyökkäyksen. SIEM-järjestelmiä käytetään yleisesti yhteistyössä myös IDS:ien kanssa. IDS:t tarkkailevat verkkoliikennettä ja erittelevät liikenteestä epäilyttävän toiminnan. Tämä tieto viedään SIEM-järjestelmään, joka analysoi saadun datan ja varoittaa hyökkäyksestä sekä tekee tarpeellisia toimia hyökkäyksen estämiseksi. SIEM-järjestelmien etuja ovatkin eri järjestelmien tuottaman datan analysoinnin keskittäminen yhteen järjestelmään sekä järjestelmän mahdollisuus toimia myös itsenäisesti hyökkäystä vastaan.

SCADA-järjestelmät ovat teollisia hallintajärjestelmiä, jotka keräävät dataa ja monitoroivat fyysisessä infrastruktuurissa käytettyä automaatiota (Sullivan, Luiijf & Colbert, 2016). SCADA-järjestelmiä käytetään esimerkiksi öljy- ja vesilaitoksia, ydinvoimaloita ja ilmanvaihtoa valvovissa järjestelmissä. Tämä muutos on mahdollistanut ennen eristettyinä olleiden järjestelmien ja verkkojen etävalvonnan sekä IoT-laitteiden käyttöönoton, joka on parantanut järjestelmien käyttövarmuutta ja toiminnallisuutta. Haittana järjestelmien kytkemisellä ulkoiseen verkkoon ovat kuitenkin kyberhyökkäykset. SCADA-järjestelmien valvoessa myös yhteiskunnalle kriittisen infrastruktuurin, kuten veden- ja sähkönjakeluverkoston toimintaa, on järjestelmät turvattava kyberhyökkäysten havaitsemis- ja estojärjestelmällä, kuten SIEM-järjestelmällä. (Hindy, Brosset, Bayn, Seeam & Bellekens, 2019.)

Hanan ym. (2019) ovat tutkimuksessaan parantaneet SCADA-järjestelmän turvallisuutta koneoppimista käyttävän SIEM-järjestelmän avulla. Tutkimuksessa vertailtiin kuutta eri koneoppimisalgoritmia verkossa esiintyvien anomalioiden havaitsemiseen ja luokitteluun. Näihin kuuluivat järjestelmän omien laitteiden vikatilanteet, sabotaasi ja kyberhyökkäykset. Lisäksi järjestelmä antoi käyttäjälleen havaitsemansa hyökkäyksen todennäköisyys- ja varmuustason auttaakseen käyttäjää valitsemaan oikeat toimenpiteet hyökkäyksen estämiseksi. Järjestelmä saavutti parhaan tuloksen k-NN-koneoppimisalgoritmia käyttämällä, jolloin se pystyi havaitsemaan hyökkäykset 94% tarkkuudella sekä luokittelemaan hälytyksen syyn 95,64% tarkkuudella. Tutkijoiden mukaan havaintotarkkuutta voitaisiin parantaa kasvattamalla algoritmien harjoitusaineiston kokoa, sekä rakentamalla hybridijärjestelmä, joka voisi luokitella eri anomaliat aliluokkiin paremman oppimisen saavuttamiseksi. (Hanan ym., 2019.)

4.3 Haittaohjelmien havaitseminen ja luokittelu

Haittaohjelma (malware) on ohjelma tai tiedosto, joka on suunniteltu vahingoittamaan sen kohteena olevaa laitetta tai tietojärjestelmää. Haittaohjelmien estämiseen yleisesti käytettyjä virustorjuntaohjelmistoja on lähivuosina tutkittu ja muun muassa Spafford (2014) on kirjoittanut artikkelissaan, että virustorjuntaohjelmistot eivät varsinkaan yrityksissä ole enää riittävä keino haittaohjelmilta suojautumiseen. Haittaohjelmien havaitseminen ja niiltä suojautuminen on erityisen tärkeää juuri yrityksissä, koska jo yhden haittaohjelman vaikutukset voivat aiheuttaa yritykselle miljoonien tappiot (Anderson ym., 2013).

Koneoppimisen hyödyntämistä on tutkittu haittaohjelmien havaitsemisessa sekä luokittelussa. Raff ym. (2017) rakensivat neuroverkkoja käyttävän koneoppimisjärjestelmän, joka oli ensimmäinen maailmassa haittaohjelmien havaitsemiseen raakatavujen analysointia hyödyntäen (Raff ym., 2017). Järjestelmä pystyi havaitsemaan haittaohjelmat aiempia havaitsemismenetelmiä paremmin kolmessa tapauksesta neljästä sekä välttämään ylisovittuneisuuden, jota havaittiin aiemmissa koneoppimista käyttävissä havaitsemismenetelmissä. Le, Boydell, Mac Namee ja Scanlon (2018) taas käyttivät koneoppimista haittaohjelmien luokittelujärjestelmässä, joka ei vaadi piirteiden valintaa toimiakseen. Tämä mahdollistaa koneoppimista hyödyntävän haittaohjelmien luokittelujärjestelmän hyödyntämisen myös henkilöille, joilla ei ole aiempaa osaamista koneoppimisteknologioiden käytöstä. Koneoppimista hyödyntävä järjestelmä pystyi tunnistamaan sekä luokittelemaan haittaohjelmat jopa 0,02 sekunnissa niiden havaitsemisesta ja järjestelmä saavutti 98,8% luokittelutarkkuuden testidatalla.

4.4 Koneoppimisen haasteet kyberturvallisuuden kontekstissa

Vaikka esitellyissä esimerkeissä koneoppiminen pystyi tunnistamaan haittaohjelmat ja tietoverkkoon kohdistuvat hyökkäykset hyvin suurella todennäköisyydellä, on koneoppimisen käyttäminen erityisen vaikeaa kyberturvallisuuden kontekstissa. Tietoverkkoihin ja järjestelmiin kohdistuvien hyökkäysten määrä ollessa jatkuvassa kasvussa sekä hyökkäysten toimintatapojen muuttuessa jatkuvasti, on koneoppimismenetelmien algoritmien valinta ja niiden harjoittaminen työlästä. Erilaiset koneoppimisalgoritmit eivät suoriudu hyökkäyksistä yhtä hyvällä lopputuloksella, joten yhtä koneoppimisjärjestelmää voidaan käyttää vain yhdenlaisen hyökkäyksen tunnistamiseen. Koneoppimisjärjestelmää käyttävän järjestelmän kehittäjän onkin tiedettävä tarkalleen mihin tarkoitukseen järjestelmää tullaan käyttämään sekä millaisia hyökkäyksiä järjestelmää kohtaan tullaan toteuttamaan. Ulottuvuuden kirous on myös ongelmana koneoppimisteknologioiden käytössä kyberturvallisuudessa, koska hyökkäysten toimintatavat ovat hyvin erilaisia, joka vaatii piirteiden lukumäärän kasvattamista korkeaksi (Kabiri, 2012). Myös järjestelmän tekemät virheet ovat yleisesti haital-

lisempia kuin perinteisissä koneoppimista käyttävissä järjestelmissä. Kyberhyökkäyksen havainnoimatta jättäminen tai sen väärä luokittelu voi johtaa erittäin suuriin tappioihin yrityksen liikevaihdossa (Sommer & Paxson, 2010).

Kuten luvussa 2.3 kerrottiin, koneoppimisen yleisimmät haasteet korostuvat pieniä aineistoja käytettäessä. Myös kyberhyökkäyksiin liittyvien tehtävien ollessa yleisesti ohjattua oppimista käyttäviä tehtäviä, on oikeiden nimiöiden käyttäminen erityisen tärkeää koneoppimismallia koulutettaessa. Koska yritykset eivät halua jakaa arkaluontoista tietoa heidän käyttämistään tietoverkoista ja järjestelmistä, on tutkimusta koneoppimisen käytöstä kyberturvallisuudessa hidastanut tutkijoiden pääsy kyberturvallisuuteen liittyviin laadukkaisiin data-aineistoihin. Lähiaikoina tutkijoiden käyttöön on kuitenkin julkaistu data-aineistoja, kuten Palo Alto Networksin julkaisema aineisto haittaohjelmiin ja tietoverkkoihin liittyen. Näiden aineistojen avulla akateemisen tutkimuksen ja koneoppimisen käytön kyberturvallisuudessa toivotaan kehittyvän, kun tutkijat pääsevät kehittämään teknologiaa oikeasta elämästä tuotetun datan pohjalta. (Amit ym. 2019.)

5 YHTEENVETO

Tutkielmalla oli kolme tutkimuskysymystä: *"Miten koneoppimista voidaan hyödyntää kyberhyökkäysten havaitsemisessa ja torjunnassa?", "Millaisia hyötyjä ja haittoja sisältyy koneoppimista käyttäviin tietoturvajärjestelmiin?"* ja *"Millaisia koneoppimisteknologioita tietoturvajärjestelmissä voidaan käyttää?"*. Tutkielman ensimmäinen sisältöluku kävi läpi keskeiset koneoppimiseen liittyvät käsitteet, menetelmät sekä haasteet. Toinen sisältöluku käsitteli kyberturvallisuuteen ja kyberhyökkäyksiin liittyvät käsitteet, haasteet sekä kyber-, tieto- ja ICT-turvallisuuden erot. Kolmas sisältöluku käsitteli erilaisia kyberturvallisuusjärjestelmiä sekä koneoppimisen käyttämistä kyberturvallisuuden kontekstissa.
Tutkimuskysymyksiin pystyttiin kirjallisuuskatsauksen perusteella vastaamaan ja tuloksista selvisi, että koneoppimista voidaan käyttää nykyisin kyberturvallisuuden kontekstissa, mutta se vaatii oikeiden koneoppimismenetelmien sekä algoritmien tarkan valinnan. Erilaisia koneoppimisalgoritmeja on valtava määrä, joista vain jotkut toimivat erityisen hyvin tiettyä hyökkäystä vastaan toimittaessa. Myös pääsy laadukkaisiin data-aineistoihin on hidastanut koneoppimisen kehittymistä kyberturvallisuuden kontekstissa. Lähiaikoina kuitenkin varsinkin akateemiseen tutkimukseen julkaistujen data-aineistojen toivotaan mahdollistavan alan kehittyminen tutkijoiden päästessä kehittämään teknologiaa oikeasta elämästä tuotetun datan pohjalta.

Kyberturvallisuutta voidaan pitää erityisen tärkeänä aiheena nykypäivänä, kun lähes kaikki järjestelmät, myös kriittistä infrastruktuuria valvovat, ovat yhteydessä ulkopuoliseen verkkoon. Tämä tekee järjestelmistä haavoittuvaisia kyberhyökkäyksille, joiden määrä varsinkin yrityksiä kohtaan kasvaa jatkuvasti. Kyberturvallisuuden työntarpeen kasvaessa ei kyberalan ammattilaisia ole tarpeeksi ylläpitämään kyberturvallisuutta nykyisiä menetelmiä käyttämällä. Koneoppimisteknologiat voivat tulevaisuudessa vähentää kyberturvallisuuteen liittyvää työmäärää ja näin myös parantaa kyberturvallisuuden mahdollisuutta kaikilla osa-alueilla.

Erilaisia esitettyjä jatkotutkimusaiheita aiheelle ovat esimerkiksi nopeampien koulutustapojen löytäminen koneoppimisalgoritmeille, jolloin koneoppimista käyttävät järjestelmät voisivat reagoida muuttuviin hyökkäystapoihin

paremmin. (Buczak & Guven, 2016). Toinen ehdotettu esimerkki on kahden eri koneoppimisalgoritmin sulauttaminen yhteen parantaakseen järjestelmän havaintotarkkuutta entisestään. (Kotsiantis, 2007).

LÄHTEET

Alpaydin, E. (2010). *Introduction to Machine Learning* (2. uud. painos). London, England: The MIT Press.

Amazon.com, Inc. (2020). Model Fit: Underfitting vs. Overfitting. Haettu osoitteesta https://docs.aws.amazon.com/machine-learning/latest/dg/model-fit-underfitting-vs-overfitting.html.

Amit, I., Matherly, J., Hewlett, W., Xu, Z., Meshi, Y. & Weinberger, Y. (2019). *Machine Learning in Cyber-Security - Problems, Challenges and Data Sets.* The AAAI-19 Workshop on Engineering Dependable and Secure Machine Learning Systems. ArXiv.

Anderson, R., Barton, C., Böhme, R., Clayton, R., van Eeten, M. J. G., Levi, M., Moore, T. & Savage, S. (2013). *Measuring the Cost of Cybercrime.* doi: 10.1007/978-3-642-39498-0_12.

Barreno, M., Nelson, B., Sears, R., Joseph, A. D. & Tygar, J. D. (2006). *Can machine learning be secure?* Proceedings of the 2006 ACM Symposium on Information, computer and communications security, 16-25.

Barto, A. & Dietterich, T. (2004). *Reinforcement learning and its relationship to supervised learning.* Handbook of Learning and Approximate Dynamic Programming, 47–64.

Bellman, R. (1957). *Dynamic Programming.* Princeton, New Jersey: Princeton University Press.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Springer-Verlag New York.

Buczak, A. L. & Guven, E. (2016). *A survey of data mining and machine learning methods for cyber security intrusion detection.* COMST, 18(2), 1153-1176. doi: 10.1109/COMST.2015.2494502.

Davis, J. & Goadrich, M. (2006). *The Relationship Between Precision-Recall and ROC Curves.* Proceedings of the 23rd International Conference on Machine Learning, ACM.

Dietterich T. (1995). *Overfitting and undercomputing in machine learning.* ACM Computing Surveys, 27(3), 326-327.

Farid, D. Md., Harbi, N. & Rahman, M. Z. (2010). *Combining Naive Bayes and Decision Tree for Adaptive Intrusion Detection.* International journal of Network Security & Its Applications, 2(2), 12–25.

Farn, K.-J., Lin, S.-K. & Fung A. R.-W. (2004). *A study on information security management system evaluation: assets, threat and vulnerability.* Computer Standards & Interfaces, 26(6), 501-513. http://dx.doi.org/10.1016/j.csi.2004.03.012.

Fraley, J. B. & Cannady, J. (2017). *The promise of machine learning in cybersecurity.* Conference Proceedings – IEEE SoutheastCon.

Gerber, M. & Von Solms, R. (2005). *Management of risk in the information age.* Computers & Security, 24(1), 16-30. http://dx.doi.org/10.1016/j.cose.2004.11.002.

Good, P. I. (2005). *Resampling Methods: A Practical Guide to Data Analysis* (3. painos). Birkhäuser.

Har-Peled, S., Indyk, P. & Motwani, R. (2012). *Approximate Nearest Neighbor: Towards Removing the Curse of Dimensionality.* Theory of Computing, 8(14), 321-350.

Hastie, T., Tibshirani, R. & Friedman, J. (2011). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, New York.

Hathaway, O.A., Crootof, R., Levitz, P., Nix, H., Nowlan, A., Perdue, W. & Spiegel, J. (2012). *The Law of Cyber-Attack.* California Law Review, 100(4), 817-885).

Hindy, H., Brosset, D., Bayne, E., Seeam, A. & Bellekens, X. (2019). Improving SIEM for Critical SCADA Water Infrastructures Using Machine Learning. Teoksessa Katsikas S. ym. (toim.) *Computer Security. SECPRE 2018, CyberICPS 2018. Lecture Notes in Computer Science*, 11387, 3-19. Springer, Cham.

ISC2 2017 Global Information Security Workforce Study. (2017). Haettu osoitteesta https://www.isc2.org/-/media/Files/Research/GISWS-Report-Europe.ashx?la=en&hash=6BCA521488491848DBCF91E8F350DBE3E0A65367.

Jansson, S. & Sihvonen, T. (2018). *Kyberturvallisuus valtiollisena toimintaympäristönä ja siihen kohdistuvat uhkat*. Media & viestintä, 41(1), 1–28.

Joo Er, M., Kashyap, A. & Wang, N. (2016). *Deep Semi-supervised Learning Using Multi-Layered Extreme Learning Machines.* The 6th Annual IEEE

International Conference on Cyber Technology in Automation, Control and Intelligent Systems, 457-462.

Kabiri, P. (2012). *Privacy, Intrusion Detection, and Response: Technologies for Protecting Networks*. IGI Global.

Khraisat, A., Gondal, I., Vamplew, P. & Kamruzzaman, J. (2019). *Survey of intrusion detection systems: techniques, datasets and challenges.* Cybersecurity. doi: 2. 10.1186/s42400-019-0038-7.

Kotsiantis, S.B. (2007). Supervised Machine Learning: A Review of Classification Techniques. Teoksessa Maglogiannis, I., ym. (toim.) *Emerging Artificial Intelligence Applications in Computer Engineering*. IOS Press.

Kotzanikolaou, P. & Douligeris. C. (2007). *Network Security: Current Status and Future Directions.* The Institute of Electrical and Electronics Engineers, Inc., 1-12.

Kroon, M. & Whiteson, S. (2009). *Automatic Feature Selection for Model-Based Reinforcement Learning in Factored MDPs.* International Conference on Machine Learning and Applications, Miami Beach, FL, 324-330.

Le, Q., Boydell, O., Mac Namee, B. & Scanlon, M. (2018). *Deep learning at the shallow end: Malware classification for non-domain experts*. Digital Investigation, 26, 118-126.

Lin, H. S. (2010). *Offensive Cyber Operations and the Use of Force.* Journal of National Security Law & Policy, 4(63), 63-86.

Livadas, C., Walsh, R., Lapsley, D., & Strayer, W. T. (2006). *Using machine learning techniques to identify botnet traffic.* Proceedings. 2006 31st IEEE Conference on Local Computer Networks, 967-974. doi: 10.1109/LCN.2006.322210.

Malwarebytes. (2020). 2020 State of Malware Report. Haettu osoitteesta https://resources.malwarebytes.com/files/2020/02/2020_State-of-Malware-Report.pdf.

Michalski, R., Carbonell, J. & Mitchell, T. (2013). *Machine Learning: An Artificial Intelligence Approach.* Springer Publishing Company.

Mitchell, T. M. (1997). *Machine Learning*. WCB/McGraw-Hill.

Mukkamala, A., Sung, A. & Abraham, A. (2005). Cyber security challenges: Designing efficient intrusion detection systems and antivirus tools. Teoksessa Vemuri, V. R. (toim.), *Enhancing Computer Security with Smart Technology* (s. 125-163). Auerbach Publications.

Myung, J. I. (2000). *The importance of complexity in model selection.* Journal of Mathematical Psychology, 44(1), 190–204.

Pasupa, K. & Sunhem, W. (2016). *A Comparison between Shallow and Deep Architecture Classifiers on Small Dataset*. 2016 8th International Conference on Information Technology and Electrical Engineering, Yogyakarta, Indonesia.

Portugal, I., Alencar, P. & Cowan, D. (2017). *The Use of Machine Learning Algorithms in Recommender Systems: A Systematic Review.* Expert Systems with Applications, 97, 205–227.

Raff, E., Barker, J., Sylvester, J., Brandon, R., Catanzaro, B. & Nicholas, C. (2017). *Malware detection by Eating a Whole EXE*. ArXiv.

Samuel, A. (1959). *Some Studies in Machine Learning Using the Game of Checkers.* IBM Journal of Research and Development.

Schatz, D., Bashroush, R. & Wall, J. (2017) *Towards a More Representative Definition of Cyber Security*. Journal of Digital Forensics, Security and Law, 12(2), 53-74. https://doi.org/10.15394/jdfsl.2017.1476.

Shalev-Shwartz, S. & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press. ISBN 978-1-107-05713-5.

Singer, P. W. & Friedman, A. (2013). *Cybersecurity: What Everyone Needs to Know.* Oxford University Press.

Sommer, S. & Paxson, V. (2010). *Outside the Closed World: On Using Machine Learning For Network Intrusion Detection.* 2010 IEEE Symposium on Security and Privacy, 305-316.

Spafford, E. C. (2014). *Is Anti-virus Really Dead?*. Computers & Security, 44, iv.

Sullivan, D., Luiijf, E. & Colbert, E. J. M. (2016). Components of Industrial Control Systems. Teoksessa Kott, A. & Colbert, E. J. M. (toim.) *Cyber-security of SCADA and Other Industrial Control Systems*, 15-28. Springer.

Symantec Corporation. (2019). ISTR Internet Security Threat Report, 24. Haettu osoitteesta https://docs.broadcom.com/doc/istr-24-2019-en.

Tripathi, H. (24.9.2019). What Is Balanced And Imbalanced Dataset? Haettu osoitteesta https://medium.com/analytics-vidhya/what-is-balance-and-imbalance-dataset-89e8d7f46bc5.

Van der Aalst, W. M. P., Rubin, V., Verbeek, H. M. W., Van Dongen, B. F., Kindler, E. & Günther, C. W. (2010*). Process mining: a two-step approach to*

balance between underfitting and overfitting. Softw Syst Model, 9, 87-111. https://doi.org/10.1007/s10270-008-0106-z.

Von Solms, R. (1998). *Information security management (3): the Code of Practice for Information Security Management (BS 7799).* Information Management & Computer Security, 6(5), 224–225.

Von Solms, R. & Van Niekerk, J. (2013). From information security to cyber security. Computers & Security, 38, 97-102. https://doi.org/10.1016/j.cose.2013.04.004.

Whitman, M. E. & Mattord, H. J. (2009). *Principles of information security* (3. uud. painos). Thompson Course Technology.

Zanero, S. & Serazzi, G. (2008). *Unsupervised learning algorithms for intrusion detection.* Network Operations and Management Symposium 2008 NOMS 2008 IEEE, 1043–1048.

Zhu, X. (2005). *Semi-Supervised Learning Literature Survey* (Väitöskirjan osa, University of Wisconsin-Madison). Haettu osoitteesta http://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf.

Zhu, X., & Goldberg, A. B. (2009). *Introduction to Semi-Supervised Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning, 3(1), 1–130.