Noora Etelä

PEHMOLASKENTAMENETELMÄT TUNKEILIJAN HAVAITSEMISESSA

JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2019

TIIVISTELMÄ

Etelä, Noora
Pehmolaskentamenetelmät tunkeilijan havaitsemisessa
Jyväskylä: Jyväskylän yliopisto, 2019, 46 s.
Tietojärjestelmätiede, kandidaatintutkielma
Ohjaajat: Luoma, Eetu; Palonen, Teija

Tässä kirjallisuuskatsauksessa tarkasteltiin kolmen yleisimmän pehmolaskentametodin, sumean logiikan, neuroverkkojen sekä evoluutiolaskennan, käyttöä tunkeilijan havaitsemisen prosesseissa. Tunkeilijan havaitsemisella tarkoitetaan järjestelmän luotettavuutta, eheyttä ja saatavuutta uhkaavien toimintojen havaitsemista. Tutkielman tarkoituksena oli jäsennellä aihealueen tutkimuskirjallisuuteen perustuen argumentteja kyseisten pehmolaskentametodien soveltamisen puolesta ja vastaan. Motiivina tutkielman suorittamiselle oli tietoverkkoturvallisuuden yleinen vastuu kehittyä kyberuhkien rinnalla ja torjua onnistuneesti tällaisia uhkia. Tutkimusasetteluna oli selvittää, miten älykkäät pehmolaskentametodit voivat parantaa tunkeilijan havaitsemisen prosessien toimintaa. Tutkielman tuloksien perusteella todettiin, että pehmolaskentametodien soveltaminen tunkeilijan havaitsemiseen on teoreettisesti perusteltua sekä eheä tutkimusalue, mutta pehmolaskentametodeja soveltavien tunkeilijan havaitsemisjärjestelmien toteuttaminen on puutteellisesti tutkittu sekä kehittyvä tutkimusaihe.

Asiasanat: pehmolaskenta, tunkeilijan havaitseminen, IDS, sumea logiikka, neuroverkot, evoluutiolaskenta

ABSTRACT

Etelä, Noora
Soft computing methods in intrusion detection
Jyväskylä: University of Jyväskylä, 2019, 46 pp.
Information System Science, Bachelor's Thesis
Supervisors: Luoma, Eetu; Palonen, Teija

The purpose of this literary review is to examine how the three most often used soft computing methods apply to the processes of intrusion detection. The intent of the study was to structure arguments for and against the application of fuzzy logic, neural networks and evolutionary computation to intrusion detection, based on available research literature of the subject area. The study was motivated with the overall responsibility of cyber security to develop alongside cyber threats and successfully combat such threats. The research problem was to find out how intelligent soft computing methods can improve the performance of intrusion detection processes. Based on the results of the study, it was found that the application of soft computing methods to detecting intrusions is justified as well as a theoretically sound research area, but the actual implementation of intrusion detection systems applying soft computing methods is poorly researched and still a developing research topic.

Keywords: soft computing, intrusion detection, IDS, fuzzy logic, neural networks, evolutionary computation

SISÄLLYS

1 JOHDANTO

Tietoyhteiskunnan kehittyminen on muuttanut jokaisen yhteiskuntasektorin toimintaa radikaalisti ja yhteiskunnan eri osa-alueiden yhdistäviksi tekijöiksi ovat nousseet kehittynyt tiedonsiirto sekä informaation kasvanut merkitys. Patchan ja Parkin (2007) mukaan internetin läsnäolo yhteiskunnan toiminnoissa on velvoittanut monimutkaisilta, sivistyneiltä tietojärjestelmiltä myös muun muassa globaalin liiketoiminnan vaatimia etä- ja langattoman käytön edellytyksiä sekä kehittyneitä salaus- ja todennustekniikoita. Uusiin ja kehittyviin vaatimuksiin vastauksena kehitetyt lisäominaisuudet lisäävät kuitenkin järjestelmien ja tietoverkkojen alttiutta tunkeutumisille ja hyökkäyksille (Patcha & Park, 2007). Tietoyhteiskunnan kehitys edellyttääkin tietoturvatoimenpiteitä ja tekniikoita suojaamaan laitteita, tietoverkkoja, ohjelmistoja ja dataa hyökkäyksiltä, luvattomalta pääsyltä sekä luvattomalta käsittelyltä (Buczak & Guven, 2016). Yhteiskunnan kasvavasta tietojärjestelmäriippuvuudesta sekä potentiaalisesta uhkaavasta toiminnasta johtuen hyökkäysten ja väärinkäytön kontrolloinnista on tullut välttämätön osatekijä niin yksityisiin kuin julkisiin turvallisuusinfrastruktuureihin (Scarfone & Mell, 2007).

Amerikkalaisen *National Institute of Standards and Technology* -viraston julkaisun (Scarfone & Mell, 2007) mukaan tietojärjestelmäympäristössä tunkeilijan havaitsemisella viitataan tietojärjestelmän tai tietoverkon tapahtumien seurannan sekä analysoinnin tapahtumasarjaan, jonka tehtävänä on löytää merkkejä mahdollisista tietoturvaloukkauksista. Tunkeilijan havaitsemisen prosessia automatisoidaan tunkeilijan havaitsemisjärjestelmällä (*eng. intrusion detection system, IDS*) (Scarfone & Mell, 2007). Tunkeutumiseksi määritellään mikä tahansa järjestelmän sisäinen tai ulkoinen toiminto tai joukko toimintoja, jotka vaarantavat tai heikentävät resurssin CIA-ominaisuuksia (luottamuksellisuus, eheys, saatavuus) (Abraham, Grosan & Chen, 2005). Tunkeilijan havaitsemisjärjestelmä pyrkii havaitsemaan, määrittelemään ja tunnistamaan tietojärjestelmän tai tietoverkon luvattoman käytön, minkä vuoksi kyseiset järjestelmät ovat palomuurien lisäksi yksi verkkoturvallisuuden

perustavanlaatuisista teknologioista (Buczak & Guven, 2016; Patcha & Park, 2007).

Eräs tunkeilijan havaitsemisen läpitunkevista ja paljon keskustelua herättävistä haasteista on kehittyvät hyökkäystyypit sekä niiden puutteellinen havaitseminen. Järjestelmällä ei ole uudenlaisista hyökkäystyypeistä aikaisempaa tietoliikennekaavaa, johon ajankohtaista liikennettä voisi verrata hyökkäysten havaitsemiseksi. Uusia hyökkäystyyppejä kutsutaan termillä nollapäivähaavoittuvuus (*eng. zero-day vulnerability*). Termi viittaa päivien lukumäärään, mikä järjestelmäkehittäjältä vie haavoittuvuuden havaitsemisesta sen paikkaamiseen. Yleinen tunkeilijan havaitsemisessa käytettyjen tiedonlouhinta- ja koneoppimismetodien heikkous nollapäivähaavoittuvuuksien havaitsemisessa on tarpeettomien hälytysten korkea määrä (Buczak & Guven, 2016), sillä kyseisten metodien on vaikea määritellä, missä kulkee vahingoittavan ja hyväntahtoisen poikkeavan käytöksen raja: milloin on syytä epäillä haitallista toimintaa ja milloin kyse on väärästä hälytyksestä?

Uusien ja kehittyvien hyökkäystyyppien aiheuttaman ongelman ratkaisemiseksi on pyritty luomaan älykkäitä järjestelmiä, mutta ponnisteluja ei ole vielä onnistuttu toteuttamaan pätevästi. Tekoälymetodeja on kritisoitu liian sopeutumattomiksi ja kategorisiksi soveltumaan tunkeilijan havaitsemisen erittäin vaihtelevaan ja labiiliin ympäristöön (Rudas & Fodor, 2008) ja lähestymistavaksi on ehdotettu niin puhtaasti laskennalliseen älykkyyteen (*eng. computational intelligence, CI*) perustuvaa tunkeilijan havaitsemisjärjestelmää kuin laskennallisen älykkyyden sekä tekoälytutkimuksen hybridistä lähestymistapaa.

Pehmolaskenta on laskennallinen metodologia ja kattotermi kokoelmalle menetelmiä, joiden lähtökohtana on yhteiskunnan luontaisen epätarkkuuden sietokyvyn hyödyntäminen laskennallisissa toimenpiteissä. Epätarkkuuden hyödyntäminen laskentaprosessissa mahdollistaa mukautuvaiset ja olosuhdevakaat laskentatulokset, jotka soveltuvat reaalimaailman suhteellisiin olosuhteisiin (Zadeh, 1994b). Pehmolaskennan tarkoituksena on siis mukautua reaalimaailman epätarkkuuteen jäljittelemällä ihmismielelle ominaista kykyä käyttää päätöksenteossa likimääräisiä ja suhteellisia premissejä täsmällisten sijaan (Zadeh, 1994a) ja pehmolaskentametodit noudattavat laskennallisen älykkyyden lähestymistapaa keinotekoiseen älykkyyteen, jossa järjestelmän sopeutumiskyky priorisoidaan (Fogel, 1995).

Tässä tutkielmassa tarkastellaan kriittisesti pehmolaskennan soveltuvuutta tunkeilijan havaitsemisen tarpeisiin saatavilla olevan tieteellisen kirjallisuuden pohjalta. Tutkielmassa analysoidaan kolmen yleisimmän pehmolaskentametodin, sumean logiikan, neuroverkkojen sekä evoluutiolaskennan soveltamista sekä sovellettavuutta tunkeilijan havaitsemisen prosesseihin. Tämä tutkimus ei ole tyhjentävä selvitys pehmolaskennan soveltuvuuteen tunkeilijan havaitsemisessa, sillä tutkielman rajauksen nimissä pehmolaskentametodien kirjosta tarkastellaan vain kolmea edellä mainittua eniten tutkittua metodia. Tutkielmassani esittelen

kirjallisuuskatsaukseni sekä aineistosta johdetut päätelmät koskien pehmolaskentametodien soveltuvuutta ja tehokkuutta tunkeilijan havaitsemisjärjestelmissä.

1.1 Motivointi

Tutkielmani premissinä pohdin, miten tilanteeseen sopeutuvat, epätarkkuutta hyödyntävät pehmolaskentametodit suoriutuisivat tunkeilijan havaitsemisesta ja erityisesti nollapäivähaavoittuvuuksien havaitsemisesta, jos monilla muilla laskentametodeilla on kyseisessä tehtävässä hankaluuksia. Jos pehmolaskentametodit suoriutuvat tunkeilijan havaitsemisesta paremmin, kuin yleisesti käytetyt metodit, voidaan yhteiskunnan yleistä tietoturvaa sekä kokonaisturvallisuutta parantaa jalostamalla tietokoneiden sekä tietoverkkojen tunkeilijan havaitsemisjärjestelmiä. Näin ollen on siis perusteltua tutkia, voidaanko olosuhteisiin nähden sopeutumiskykyisiä pehmolaskentametodeja hyödyntää tunkeilijan havaitsemisen jatkuvasti kehittyvässä ympäristössä ja miten edellä mainitut laskentametodit suoriutuvat tunkeilijan havaitsemisesta. Mielenkiintoista on myös selvittää pehmolaskentamenetelmien uniikit ongelmat tunkeilijan havaitsemisen kontekstissa. Tutkielman tarkoitus on esittää lukijalle kiinnostavia näkökulmia tunkeilijan havaitsemisen kehittämiseen sekä toimia kokoavana katsauksena tunkeilijan havaitsemisen sekä pehmolaskennan tutkimussuuntausten kohtaamiseen.

Tutkielman avulla pyritään selventämään, onko nykyisten tietoturvavaatimusten puitteissa perusteltua soveltaa pehmolaskentametodeja IDS-järjestelmissä. Tutkielman tavoitteena on vastata tutkimuskysymyksiin: *Miten pehmolaskentametodit parantavat tunkeilijan havaitsemista? Minkälaisia ongelmakohtia pehmolaskentametodien soveltaminen tunkeilijan havaitsemisen ympäristöön esittää?* Tutkielman rakennetta sekä johdonmukaisuutta ylläpidetään toissijaisilla tutkimuskysymyksillä: *mitä tarkoitetaan pehmolaskennalla* ja *mitä tarkoitetaan tunkeilijan havaitsemisella?* Tutkielman tuloksia on mahdollista soveltaa yleisen tietoturvan ja IDS-järjestelmien suorituksen parantamisessa sekä järjestelmän ylläpitäjän työkuorman optimoinnissa.

1.2 Toteutus

Tutkielma suoritetaan kirjallisuuskatsauksena. Tutkimusmenetelmä vaatii aihealueen aikaisempaan kirjallisuuteen perehtymistä ja tutkielman pohjustava tieteellinen kirjallisuus hankitaan digitaalisia verkkojulkaisuarkistoja, kuten Scopus-tietokantaa, Finna-hakupalvelua sekä Google Scholar -hakupalvelua hyödyntäen. Tutkimusaineistoa haetaan hakulausein, jotka on laadittu valitsemalla tutkimusaiheeseen liittyviä aihesanoja sekä tutkimalla valittujen

aihesanojen välisiä yhteyksiä yleisen suomalaisen asiasanaston ja Tieteen termipankin avulla. Aineistoa on etsitty muun muassa hakulauseilla ”soft computing” AND ”intrusion detection”, ”intrusion detection system” AND fuzzy OR ”neural networks” OR “evolutionary computing” ja “genetic algorithms” AND “intrusion detection”. Finna-hakupalvelussa hakutuloksista valittiin näytettäviksi vain vertaisarvioidut tieteelliset artikkelit ja ensimmäisellä hakulauseella hakutuloksina saatiin 708 vuosina 1995-2018 julkaistua tieteellistä artikkelia. Scopus-tietokannassa saman hakulauseen tuloksiksi saatiin 185 artikkelia, jotka järjestettiin viittausten määrän mukaan laskevasti. Tutkielman aineisto on valittu hakutuloksista tarkastelemalla tutkimusaiheeseen liittyvien artikkelien abstrakteja ja määrittelemällä artikkelien pätevyys suhteessa tutkimusaiheeseen, tarkastelemalla artikkelien viittausmääriä, kirjoittajia, kirjoittajien h-indeksejä ja tutkijaprofiileja Google Scholar -palvelussa sekä julkaisujen IF-arvoja.

Tutkielman tuloksena tunkeilijan havaitsemisen kehittämisen sekä pehmolaskentametodien soveltamisen on havaittu olevan monimutkainen kokonaisuus, johon ei tutkimusyhteisössä ole tämän tutkielman julkaisuun mennessä löydetty yleiseksi hyväksyttyä lähestymistapaa. Tunkeilijan havaitsemisen tutkimusyhteisössä eri pehmolaskentametodien sekä yhdistelmämetodologioiden soveltamista on tutkittu kattavasti, rohkaisevin tuloksin, mutta tutkimusehdotusten toteuttaminen sekä kaupallistaminen ohjelmistotuotteiksi on ollut puutteellista. Tämän uskotaan johtuvan tunkeilijan havaitsemisen epävakaasta ympäristöstä, jossa robustin suoriutumisen takaaminen on hankalaa.

Tutkielma etenee noudattaen seuraavanlaista rakennetta: Tutkielman toisessa luvussa vastataan ensimmäiseen toissijaiseen tutkimuskysymykseen perehtymällä pehmolaskentaan, sen läheisiin käsitteisiin sekä tutkielman keskeisiin pehmolaskentametodeihin. Seuraavassa luvussa tarkastellaan tunkeilijan havaitsemista sekä tunkeilijan havaitsemisjärjestelmiä, ja vastataan toiseen toissijaiseen tutkimuskysymykseen. Neljännessä luvussa esitellään tieteelliseen kirjallisuuteen perustuen pehmolaskentametodien soveltamista tunkeilijan havaitsemiseen. Viidennessä luvussa kootaan tutkielmasta yhteenveto ja tarkastellaan tutkielmaa tutkimuskysymysten näkökulmasta.

2 PEHMOLASKENTA

Monimutkaisen ja abstraktin luonteensa vuoksi pehmolaskentaan tutustuessa on syytä katsastaa myös käsitteen riippuvuussuhteet sekä läheiset konseptit. Tässä luvussa esitellään ensin pehmolaskennan käsite, jonka jälkeen tuodaan ilmi pehmolaskennan käsitteen suhde laskennalliseen älykkyyteen, tekoälytutkimukseen sekä älykkyyteen perustuen julkaistuun tieteelliseen kirjallisuuteen. Luvun lopussa tutustutaan yleisimpiin pehmolaskentametodeihin.

2.1 Pehmolaskenta laskennallisen älykkyyden kontekstissa

Professori Lotfi Zadeh, uranuurtaja sumean logiikan sekä pehmolaskennan teoretisoinnissa, on määritellyt artikkelissaan (Zadeh, 1994b) pehmolaskennan laskennallisten metodien kokoelmaksi, joka on syntynyt vastaamaan kvantitatiivisuutta, tarkkuutta ja toistettavuutta arvostavan tieteen puutteisiin. Perinteiset, täsmälliset matemaattiset mallit suosivat tarkkuutta sekä täydellisyyttä ja niitä noudattavalle kovalle tietojenkäsittelylle on tyypillistä, että laskennalliset prosessit suoritetaan manipuloimalla lukuja ja symboleja (Zadeh, 1999). Näin ollen reaalimaailman suhteellisia tutkimuskohteita ja muuttujia sekä niiden ominaisuuksia ja riippuvuuksia on abstrahoitava, jotta tarkat, symboliset laskentaprosessit voidaan suorittaa ja tutkimustulokseksi saadaan matemaattisesti looginen, täsmällinen ja vahva lopputulos. Kyseinen abstrahointi kuitenkin heikentää kovan tietojenkäsittelyn soveltuvuutta esittää ja ratkaista reaalimaailman monisyisiä ongelmia (Ibrahim, 2016), sillä kvantitatiivisen tutkimuksen perustavanlaatuinen tavoite on muuntaa reaalimaailman epätarkat havainnot perinteisen matemaattisen logiikan mukaan mitattavaan muotoon (Zadeh, 1999). Zadeh (1994b) mainitsee täsmällisyyden priorisoinnin johtavan myös intensiivisiin laskentaprosesseihin ja siten kelpaamattomiin laskentanopeuksiin ja korkeaan resurssien kulutukseen.

Kovan tietojenkäsittelyn vastakohtana pehmolaskenta on metodologinen kokoelma erillisiä laskentametodeja, jotka noudattavat pehmolaskennan perusperiaatetta: metodit pyrkivät löytämään joustavan, pätevän sekä edullisen ratkaisun epätarkkaan ja suhteelliseen ongelmaan käyttämällä hyväksi epätarkkuuden, epävarmuuden, suurpiirteisyyden ja epätäydellisyyden sietokykyään (Zaheh, 1994b; Mitra, Pal & Mitra, 2002). Epätarkkuuden hyödyntäminen laskentaprosessissa mahdollistaa laskentaresurssien säästämisen, laskentanopeuden eksponentiaalisen kasvun sekä mukautuvaiset ja olosuhdevakaat laskentatulokset, jotka soveltuvat reaalimaailman suhteellisiin olosuhteisiin (Zadeh, 1994b). Pehmolaskennan tarkoitus on siis mukautua reaalimaailman epätarkkuuteen jäljittelemällä ihmismielelle ominaista kykyä hyödyntää päätöksenteossa likimääräisiä ja suhteellisia premissejä täsmällisten sijaan (Zadeh, 1994a) ja samalla laskea tietojenkäsittelyyn liittyviä kustannuksia (Zadeh, 1994b). Metodologian tavoite on simuloida ihmisaivojen kykyä johtaa epätarkasta informaatiosta rationaalisia päätelmiä (Rudas & Fodor, 2008).

Pehmolaskennan alle sijoittuu useita eri metodeja, mutta koulukunnan yleisimmiksi metodeiksi lasketaan sumea logiikka *(eng. fuzzy logic, FL)*, neuroverkot *(eng. neural networks / artificial neural networks, NN/ANN)* (Zadeh, 1994b) sekä evoluutiolaskenta *(eng. evolutionary computation, EC)* (Ibrahim, 2016). Edellä mainittujen metodien yhdistävänä tekijänä jokainen metodeista mukautuu yllä mainittuun pehmolaskennan perusperiaatteeseen: hyödyntäen epätarkkuutta, metodit pyrkivät joustavaan, pätevään ja edulliseen ratkaisuun. Metodit ovat synergisiä ja yhteistoimivia kilpailevien sijaan (Mitra et al. 2002) ja imitoivat ihmismielen kykyä perustella suurpiirteisesti formalisoimalla ihmismielen perustavanlaatuisia kognitiivisia rakenteita ja toimintoja (Zadeh, 1994a). Tässä tutkielmassa tutkimuksen kohteeksi on valittu edellä mainitut yleisimmät pehmolaskentametodit ja jatkossa pehmolaskentametodeilla viitataan kyseisiin metodeihin. On kuitenkin syytä muistaa, etteivät tässä tutkielmassa käsitellyt metodit ole tyhjentävä näkökulma pehmolaskentametodeihin.

2.1.1 Pehmolaskennan suhde tekoälyyn

Garnham (2017) luonnehtii pehmolaskentaa menetelmälliseksi lähestymistavaksi kohti keinotekoista älykkyyttä ja koneälyä. Tunnetuin aksiooma keinotekoisen älykkyyden lähestymiseen on tekoälyn tutkimussuuntaus, jossa älykkyyttä ja älykästä käyttäytymistä pyritään ymmärtämään ja analysoimaan replikoimalla ja simuloimalla sitä keinotekoisesti. Tekoälytutkimuksen kaksi merkityksellisintä tavoitetta ovat inhimillisen älykkyyden ymmärtäminen sekä hyödyllisten työkalujen kehittäminen. Tutkimusalana tekoäly kytkeytyy vahvasti psykologiaan sekä kognitiotieteeseen ja edellä mainitut tutkimusalat saavat vaikutteita toisistaan. (Garnham, 2017, s. 1.)

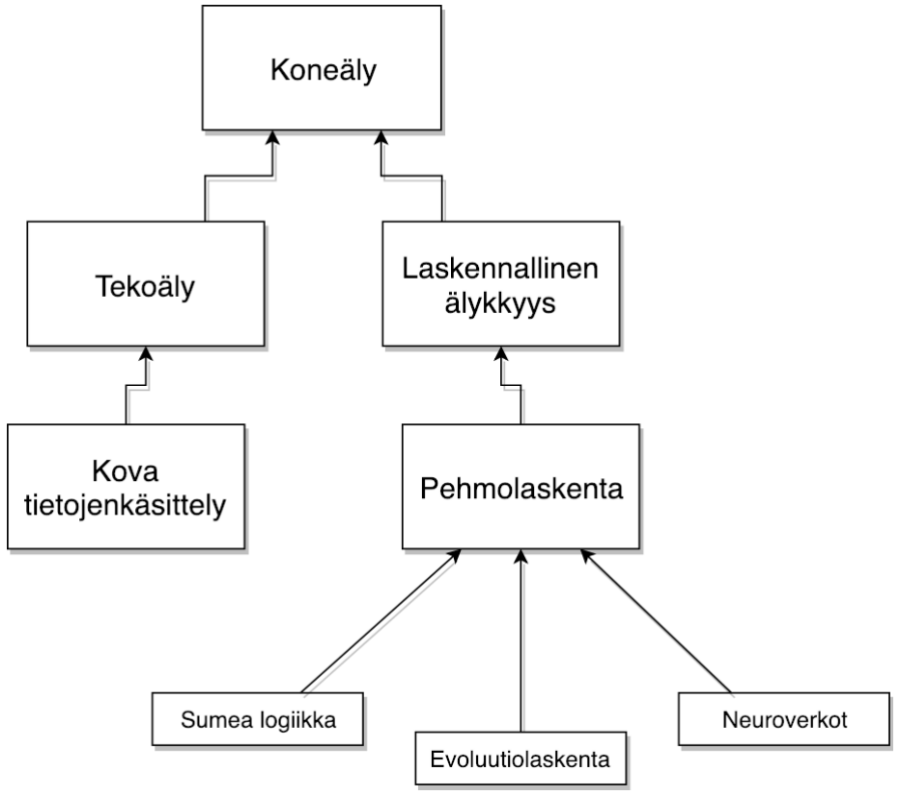
Garnham (2017, s. 3) täsmentää, ettei tekoälytutkimuksen tarkoitus ole jäljitellä ihmisen älykkään käytöksen perustavanlaatuisia mekanismeja tai kognitiivisia rakenteita. Tekoälytutkimuksessa on perinteisesti hyödynnetty kovan tietojenkäsittelyn metodeja ja tekniikoita, joiden perusominaisuuksia ovat tarkkuus, muodollisuus ja luokiteltavuus. Kyseiset metodit perustuvat symbolista logiikkaa noudattaviin fundamentaalisiin matemaattisiin teorioihin, kuten numeeriseen analyysiin, todennäköisyysteoriaan, differentiaaliyhtälöihin, funktionaaliseen analyysiin ja matemaattiseen ohjelmointiin (Rudas & Fodor, 2008). Tekoäly tieteenalana on dominoinut keinotekoisen älykkyyden tutkimusta 1950–luvun puolesta välistä lähtien (Garnham, 2017, s. 3), mutta siitä huolimatta harvat nykyhetken tietokoneista, ohjelmistoista tai järjestelmistä osoittavat määritelmän mukaista älykkyyttä (Garnham, 2017, s. 15). Rudas & Fodor (2008) väittävät artikkelissaan tämän johtuvan tekoälyn luontaisesta jäykkyydestä ja eksaktisuudesta: laskennallinen ympäristö, jota käytetään kyseisissä analyyttisissä lähestymistavoissa voi olla liian kategorinen sekä joustamaton mahdollistaakseen monimutkaisten ja kompleksisten reaalimaailman ongelmien ratkaisuun. On siis argumentoitavissa, että keinotekoisen älykkyyden luominen vaatii järjestelmältä epätarkkuuden sietokykyä.

Tarkkuuden ja täsmällisyyden tavoittelu tutkimustyössä on mahdollistanut yhteiskunnan loistavan teknisen kehityksen niin tietotekniikan saralla kuin muissa tekniikan moderneissa ilmentymissä, mutta teknisen menestyksen rinnalla inhimillisyyttä vaativan toiminnan ja päätöksenteon automatisoinnin kehityksen puutteellisuus on Zadehin (1999) mukaan silmiinpistävää. Hän päättelee artikkelissaan koneälyn kehityksen hidastumisen riippuvan tekoälyn tutkimussuuntauksen binäärisen lähestymistavan huonosta sopeutuvuudesta käsittelemään inhimillisen älykkyyden konseptin syvällisiä ja monisyisiä arvoja. Tekoälyn lähestymistavan sopeutumattomuus koneälyn konseptiin voidaan Zadehin (1962) mukaan johtaa systeemiteorian heikosta soveltumisesta yhteiskunnallisiin järjestelmiin, jotka koostuvat erillisistä taloudellisista, biologisista, sosiaalisista ja sosio-teknisistä järjestelmistä. Eroavaisuudet elollisten ja elottomien järjestelmien välillä kuvastavat perinteisen matematiikan fundamentaalista soveltumattomuutta luonnollisiin järjestelmiin, jotka sijoittuvat suuruusluokiltaan ja monimutkaisuudeltaan täysin eri ulottuvuuteen kuin ihmisten tuottamat keinotekoiset järjestelmät. Zadeh väittää, että luonnollisia järjestelmiä lähestyttäessä on lähestymistapaa muutettava käsittelemään biologisia, sumeita arvoja ja suhteita, jotka eivät noudata tuntemamme matematiikan sääntöjä (Zadeh, 1962). Näin ympyrä sulkeutuu ja palaamme pehmolaskennan ja kovan tietojenkäsittelyn koulukuntien väliin.

2.1.2 Laskennallisen älykkyyden suhde pehmolaskentaan

Ymmärtääksemme pehmolaskentaa, tulee meidän myös ymmärtää laskennallisen älykkyyden käsite sekä sen suhde tekoälyyn. Rudas ja Fodor (2008) viittaavat artikkelissaan Zadehin (1994c) konferenssijulkaisuun, jossa laskennallisen älykkyyden tutkimussuuntaus on erotettu tekoälytutkimuksesta. Luontaisesti päinvastaiset laskennan tyylisuuntaukset, pehmolaskenta sekä kova tietojenkäsittely, on kategorisesti erotettu toisistaan edellä mainittujen tutkimussuuntausten alle. Binäärisiä, formaaleja matematiikan sääntöjä noudattava kova tietojenkäsittely on ryhmitelty tukemaan tekoälytutkimuksen suuntausta ja vastaavasti epätarkkuutta hyödyntävä pehmolaskenta on sijoitettu laskennallisen älykkyyden tutkimussuuntauksen alle (Eberhart & Shi, 2007, s. 35). Määrittelevä ero tekoälyn ja laskennallisen älykkyyden välillä on laskennallisen älykkyyden tavoite soveltaa likiarvoja ja luonnollisuutta. (Rudas & Fodor, 2008.)

Laskennallinen älykkyys, johon tässä tutkielmassa viitataan myös lyhenteellä CI, on yksi tietojenkäsittelytieteen konsepteista, joita määriteltäessä alan tutkijat eivät ole päässeet yksimielisyyteen. Fogel (1995) viittaa artikkelissaan James C. Bezdekin (1994) määritelmään, jossa laskennallisesti älykkään järjestelmän vaatimuksiksi on määrätty ihmismielen tapainen, matalan tason datasta johdettu havainnointikyky, hahmontunnistuskomponentti, laskennallinen sopeutuvaisuus, virheensietokyky sekä inhimillinen suoritusaika ja virhemäärä. Bezdekin määritelmä nähdään yleisesti ensimmäisenä johdonmukaisena laskennallisen älykkyyden määritelmänä ja on sen vuoksi mainitsemisen arvoinen. Fogel (1995) kuitenkin väittää, että vahvin ratkaiseva ominaisuus järjestelmän laskennallisen älykkyyden määrittämisessä on järjestelmän itsenäinen laskennallinen sopeutuvaisuus. Bezdekin (1994) laskennallisen älykkyyden määritelmän mukaan algoritmi on laskennallisesti sopeutuva, ja näin ollen älykäs, kun se kykenee muuttamaan parametrejaan ja konfiguraatiotaan mahdollistamaan muutokset syötteissä, katkaisematta algoritmin prosessointikykyä. Bezdekin (1994) määritelmän mukaan laskennallisesti älykäs algoritmi siis simuloi aivojen nopeaa tilannekohtaista sopeutumiskykyä. Fogel (1995) tukee edellä mainittua määritelmää väitteellään, ettei järjestelmä, joka toimii vain rajoitetussa ympäristössä eikä kykene itsenäiseen sopeutuvaan käytökseen, osoita älykkyyttä. Kuviossa 1 visualisoidaan pehmolaskennan ja laskennallisen älykkyyden suhde koneälyyn.



KUVIO 1 Mallinnus laskennallisen älykkyyden suhteista (muokattu: Rudas & Fodor, 2008, s. 134)

Eberhartin ja Shin (2007, s. 30–31) määritelmä laskennallisesta älykkyydestä on laadultaan Bezdekin määritelmää yleistävämpi. Määritelmän mukaan laskennallinen älykkyys on tietojenkäsittelymetodologia, jonka tarkoitus on muodostaa järjestelmälle oppimiskyky sekä kyky käsitellä uusia tilanteita niin, että järjestelmän nähdään omaavan yksi tai useampi älykkyyden ominaispiirteistä. Älykkyyden ominaispiirteiksi lasketaan muun muassa kyky yleistää, havainnoida, assosioida ja abstrahoida. (Eberhart & Shi, 2007, s. 30-31.)

Laskennallisen älykkyyden tutkimussuuntaus tutkii siis sopeutuvia mekanismeja, jotka mahdollistavat älykkään käyttäytymisen moninaisissa ja epävakaissa ympäristöissä (Venayagamoorthy, 2009). Järjestelmän sopeutumiskyky nostetaan usein laskennallisen älykkyyden keskiöön ja sopeutuvaisuus nähdään itseorganisoitumisen rinnalla laskennallisen älykkyyden fundamentaalisena rakennuspalikkana (Eberhart & Shi, 2007, s. 20).

Sopeutuvaisuus määritellään tässä tutkielmassa Oxford English Dictionary -sanakirjan määritelmän (”Adaptation”, 2011) mukaan toimenpiteeksi, jolla laukaistaan muutos kohteiden olemuksessa tuomalla yksi itsenäinen kokonaisuus toiseen tai kaksi erillistä kokonaisuutta yhteen. Järjestelmän sopeutuessa uuteen ympäristöön ympäristön elementit ja ominaisuudet saapuvat järjestelmään ja laukaisevat näin sopeutuvan toiminnan järjestelmässä. Eberhart ja Shi (2007, s. 18) määrittelevät dynaamisen sopeutumisen järjestelmän kyvyksi sopeutua muuttuvassa ympäristössä

reaaliaikaisesti sen sijaan, että järjestelmä kytkettäisiin offline-tilaan ja konfiguroitaisiin sekä koulutettaisiin uudelleen. Dynaamisen sopeutumisen käsite puolestaan kytkeytyy Bezdekin (1994) määritelmään laskennallisesti älykkäästä algoritmista, mikä määritelmän mukaan pystyy sopeutumaan muutoksiin kesken suorituksen.

Eberhart ja Shi (2007, s. 26) toteavat luonnollisissa järjestelmissä esiintyvänä ilmiönä sopeutuvaisuuden olevan vahvasti sidoksissa itseorganisoinnin käsitteeseen. Itseorganisoitumisella, johon viitataan myös termillä spontaani järjestys, tarkoitetaan järjestelmän sisäisiä, itseaiheutettuja muutoksia, jotka mahdollistavat järjestelmän sopeutumisen ja evoluution (Eberhart & Shi, 2007, s. 29). Eberhart ja Shi (2007, s. 26–27) viittaavat teoksessaan Dysonin (1997) esimerkkiin vastasyntyneen lapsen aivojen hermoverkoston itseorganisoituvasta kehityksestä, jossa merkityksettömät kytkökset kuihtuvat pois antaen tilaa jatkuvalle sopeutumiselle. Näin älykkään toiminnan, sopeutumisen ja itseorganisoitumisen käsitteet liittyvät yhteen. Järjestelmän autonominen itseorganisoituminen mahdollistaa järjestelmän sopeutumisen muuttuvaan ympäristöön, mikä puolestaan on älykkään toiminnan perusta.

2.1.3 Koneäly

Kuten edellä käsitellyt käsitteet, myös koneäly on tietojenkäsittelytieteessä väitelty konsepti, ja määrittelevän tutkimuskirjallisuuden puutteesta sekä useiden samankaltaisten termien keskinäisestä käytöstä voidaan päätellä, ettei älykkään järjestelmän tai älykkään koneen käsite ole tietojenkäsittelytieteen alalla tieteellisesti vakiintunut. Useita erilaisia termejä, kuten älykäs järjestelmä (Krishnakumar, 2003), älykäs kone (Jain, Quteishat & Lim, 2007) tai koneäly, käytetään vaihtelevasti samanlaisen ajatuksen tai käsitteen kuvaamiseen. Useimmiten eri termien yhdistävä tekijä on pyrkimys muodostaa yleisesti hyväksytty yhteinen käsitys tekoäly- ja laskennallisen älykkyyden tutkimuksen ensisijaisesta tavoitteesta: itsenäisesti ja älykkäästi toimivasta keinotekoisesta koneesta.

Tässä tutkielmassa älykkäällä järjestelmällä viitataan Krishnakumarin (2003) määritelmän mukaan matemaattisesti luotettavaan ja laskennallisesti vaativaan järjestelmään, joka emuloi joitain luonnossa esiintyviä älykkyyden ilmentymiä, kuten oppimista, sopeutumista, sitkeyttä, ajallista tai tilallista kehittymistä, tiedon jalostamista tai päättelykykyä. Yksinkertaisesti, älykkääksi järjestelmäksi luokitellaan järjestelmä, joka osoittaa älykkyyttä. Määritelmän kompastuskivi kuitenkin on älykkyyden määritteleminen, mitä voidaan käsitellä pätevyyden, asiantuntemuksen, kyvykkyyden, koulutuksen, älykkyysosamäärän tai sosiaalisen kanssakäymisen näkökulmasta (Krishnakumar, 2003). Yleinen älykkään järjestelmän kehitykseen kytkeytyvä paradigma on, että älykäs järjestelmä saavutetaan soveltamalla tekoälytutkimusta sekä tekoälytutkimuksen hyväksymiä metodeja järjestelmän suunnittelussa (Rudas & Fodor, 2008). Perinteisesti tekoälytutkimuksessa on

hyödynnetty kovan tietojenkäsittelyn metodeja (Langin & Rahimi, 2010), mutta jälleen kerran, kuten älykkäiden järjestelmien suhteellisen hidas kehitys kovaa tietojenkäsittelyä noudattaen osoittaa, tekoälyn sekä laskennallisen älykkyyden koulukuntien risteytetty lähestymistapa älykkään järjestelmän kehittämiseen on perusteltu ja mahdollisesti käänteentekevä muutos. Ovaska, Kamiya ja Chen (2006) vertaavat artikkelissaan laskennallisen älykkyyden sekä tekoälyn suhdetta ihmisen aivojen vasempaan ja oikeaan puoliskoon. Artikkelissa teoretisoidaan kyseisellä analogialla syytä pehmolaskennan sekä kovan tietojenkäsittelyn integroinnille. Kyseinen metafora ei välttämättä ole tiukimman tieteellisen perustan mukainen, mutta antaa joka tapauksessa mielenkiintoisen näkökulman aiheeseen.

2.2 Pehmolaskentametodit

Kuten aiemmin mainittiin, pehmolaskentamenetelmät toimivat laskennallisen älykkyyden toteutuksen työkalupakkina. Pehmolaskennan tunnetuimmat ja tutkituimmat paradigmat, sumea logiikka, neuroverkot sekä evoluutiolaskenta, mallintavat jokainen vaihtelevasti biologisen älykkyyden piirteitä (Kulkarni, Förster & Venayagamoorthy, 2011). On olemassa muita laskentamenetelmiä, jotka noudattavat pehmolaskennan perusperiaatetta ja näin ollen lukeutuvat pehmolaskennan sateenvarjotermin alle, mutta edellä nimetyt paradigmat ovat laskennallisen älykkyyden vallitsevat ja yleisimmin tutkitut metodologiat ja tässä tutkielmassa tarkastellaan pehmolaskentaa sekä pehmolaskennan käyttöä kyseisten metodien kautta. Seuraavissa alaluvuissa perehdytään näihin metodeihin hieman paremmin.

2.2.1 Sumea logiikka

Tietojenkäsittelyn keskeinen osa-alue on symbolien ja lukuarvojen manipulointi laskentakaavojen suorittamiseksi (Zadeh, 1999). Kuten jo aikaisemmin käsiteltiin, maailmamme ei noudata mustavalkoisia absoluuttisia totuuksia tai binäärisiä arvoja (Eberhart & Shi, 2009, s. 9). Inhimillisyyden nimissä havaintomme, kommunikaatiomme ja kokemuksemme sisältävät aina osan epävarmuutta ja epätäsmällisyyttä. Eberhart ja Shi esittävät esimerkin inhimillisestä epävarmuudesta viittaamalla lauseeseen ”Ensi vuonna vierailen Havaijilla”, jota ei voida pitää absoluuttisena totuusarvona. (Eberhart & Shi, 2009, s. 9.)

Eberhartin ja Shin (2009, s. 9) mukaan epävarmuutta määritellessä käsite on jaettavissa kahteen alakäsitteeseen. Siinä missä tilastollinen epävarmuus perustuu todennäköisyyslakeihin ja on ratkaistavissa laskelmoinneilla ja havainnoilla, ei-tilastollinen epävarmuus puolestaan kumpuaa epätäsmällisyydestä ja moniselitteisyydestä, ja johon liittyy läheisesti kielellisesti epätäsmälliset ilmaisut, kuten *pian, melko paljon* ja *nopeasti*. Eberhart

ja Shi väittävät epätäsmällisyyden ja epävarmuuden, toisin sanoin sumeuden, olevan synnynnäinen osa ubiikkia järjestelmää, joka ei ole ratkaistavissa loogisilla havainnoinneilla ja mittauksilla. Tämän vuoksi luontaisen epätarkkuuden hyväksyminen ja hyödyntäminen mahdollistaa järjestelmän realistisemman analyysin, jolloin epätäsmällisiä ja epävarmoja muuttujia ei tarvitse abstrahoida laskentaprosessia varten. Sumea logiikka tarjoaa viitekehyksen ei-tilastollisen epätarkkuuden loogiseen käsittelyyn. (Eberhart & Shi, 2009, s. 9–11, 297.)

Sumean logiikan käsite on luotu edustamaan metodologiaa, jonka lähtökohtana on oletusarvoltaan epätarkan inhimillisen tiedon ja tietämyksen mallintaminen sekä käsittely laskennallisissa toimenpiteissä (Ibrahim, 2016). Sumeudella viitataan informaation ja datan luontaiseen ei-tilastolliseen epätarkkuuteen (Rudas & Fodor, 2008). Sumea logiikka perustuu kielelliseen moniarvologiikkaan ja sumeaan joukko-oppiin, kääntäen päälaelleen perinteisen kaksiarvoisen logiikan perustan (Niskanen, 2003, s. 2). Cantorin joukko-opin absoluuttisen joukkoon kuulumisen tai kuulumattomuuden sijasta sumeassa joukko-opissa joukkoon kuulumista mallinnetaan käyttämällä *jäsennyysastetta*, joka esitetään yleensä minä tahansa arvona lukujen 0 ja 1 välillä (Niskanen, 2003, s. 50). Kyseiset välimaastoarvot kuvaavat, kuinka paljon tai kuinka vähän alkio kuuluu tiettyyn joukkoon. Kielellinen moniarvologiikka puolestaan mahdollistaa käsiteltävän datan yleistämisen sekä abstrahoinnin, mikä on yksi aiemmin määritellyistä älykkyyden ominaispiirteistä. Esimerkkinä kielellisen arvon yleistävästä luonteesta Zadeh (1994a) viittaa kielelliseen muuttujaan *ikä*, jolle annetaan lukuarvojen sijaan kielelliset arvot *nuori, keski-ikäinen* ja *vanha*. Sumeaa logiikkaa hyödyntävät järjestelmät käyttävät kielellisiä muuttujia, sumeita joukkoja ja sumeita jäsennyysarvoja laskentaprosesseissa ja päätöksenteossa (Kulkarni ym., 2011).

Sumean logiikan tärkein ominaisuus on sen kyky hyödyntää epätäsmällisiä ja epävarmoja, sumeita muuttujia, jotka ovat reaalimaailmassa ubiikisti läsnä (Eberhart & Shi, 2009, s. 9). Sumea logiikka mahdollistaa siis laskentaprosessit käyttäen hyväksi luonnollisia, kielellisiä arvoja. Perusero symbolisen ja sumean laskennan välillä on symbolien ja lukuarvojen binäärisyys ja tarkkuus verrattuna kielellisen arvon kontekstiriippuvaisuuteen ja epätarkkuuteen. (Zadeh, 1999.)

Zadeh (1999) argumentoi kielellisten käsitteiden ja kieleen liittyvien sosiaalisten rakenteiden olevan avainasemassa inhimillisen, rationaalisen havainnointikyvyn, päätöksenteon ja suorittamisen kontekstissa. Kielellisiä arvoja hyödyntävä laskenta soveltuu ympäristöön ja tilanteeseen, jossa tutkimuskohde tai ongelmaan läheisesti kuuluva tärkeä informaatio on laadultaan liian epätarkkaa, jotta laskentaprosessit voitaisiin suorittaa symbolisesti (Zadeh, 1999.)

2.2.2 Neuroverkot

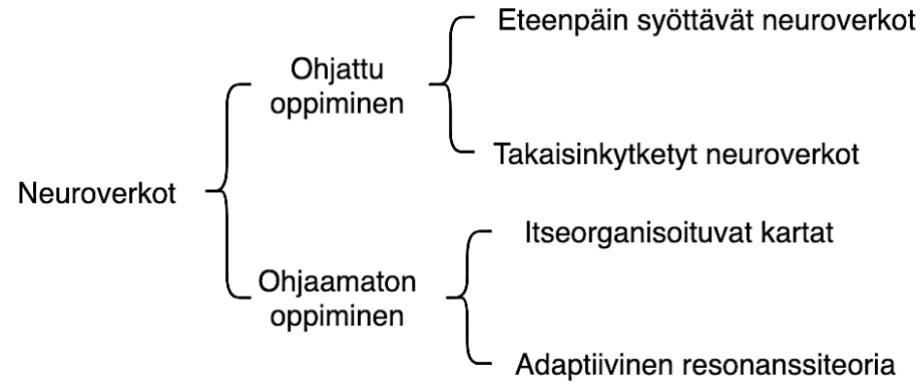
Evoluution suursaavutuksena ihmisaivot kykenevät oppimiseen, ulkoa opetteluun ja yleistämiseen, ja kaiken aivotoiminnan ytimessä on synapsien ja hermosolujen välinen signalointi (Kulkarni ym., 2011). Aivojen monipuolista adaptiivisuutta ja sen mahdollistamaa älykästä toimintaa on lähestytty neuroverkkojen tapauksessa mallintamalla ihmisaivojen mikroneuroanatomista rinnakkaista ja verkostoitunutta rakennetta. Neuroverkko on yksinkertaisista prosessointielementeistä, solmuista, koostuva keinotekoinen, linkittynyt, rinnakkainen laskennallinen rakenne (Eberhart & Shi, 2007, s. 2), joka simuloi aivojen hermosolujen ja synapsien välistä kommunikaatiota ja pyrkii näin emuloimaan biologisten hermoverkkojen kykyä yleistää, sopeutua sekä käsitellä epätarkkaa informaatiota (Mitra & Hayashi, 2000).

Suuruusluokkansa perusteella neuroverkot pohjautuvat biologiaan hyvin karkeasti. Biologiset hermojärjestelmät koostuvat miljardeista soluista ja lukemattomista synaptisista yhteyksistä (Kulkarni ym., 2011), siinä missä tyypillinen keinotekoinen neuroverkko koostuu kymmenistä tai muutamista sadoista solmuista (Eberhart & Shi, 2007, s. 6). Tyypillisesti neuroverkot koostuvat monista, useiden solmujen kerroksista, ja rinnakkaisten kerrosten solmut ovat linkittyneet toisiinsa painotettujen synaptisten linkkien avulla (Venayagamoorthy, 2009). Kerros-tyyppejä on yleisen neuroverkkojen topologian mukaan kolme: syötekerros, piilokerrokset sekä ulostulokerros (Ibrahim, 2016). Jokainen solmu suorittaa saamansa syötteen perusteella toimintoja ja tulostetta käytetään seuraavan tason solmun syötteenä (Kulkarni, 2011). Kokonaisuuden tulosteeksi saadaan solmujen suoritettujen toimintojen painotettu summa (Ibrahim, 2016).

Neuroverkot oppivat ja määrittelevät synaptisten linkkien painotukset harjoitusdatan syötteiden kaavojen perusteella. Oppimisprosessi, joka voi olla ohjattu tai ohjaamaton, tapahtuu muuttamalla synaptisten linkkien painoarvoja, jotta syötedatassa piilevät kaavat havaitaan ja suhteet syötteiden ja haluttujen tulosten välillä määritellään. Näin neuroverkon tulosteeksi saadaan haluttu, pätevä tulos (Ibrahim, 2016). Linkkien painoarvot asetetaan tarkkaan tasapainoon, jotta tulosteen virheet minimoidaan (Kulkarni ym., 2011).

Neuroverkoilla on kolme yleisintä toteutusmallia: eteenpäin syöttävät neuroverkot *(engl. Feedforward neural networks)*, takaisinkytketyt neuroverkot *(engl. Recurrent neural network)* sekä itseorganisoituvat kartat *(engl. Self-organizing maps, SOM)*. Itseorganisoituvat kartat ovat yleisimmistä toteutusmalleista ainoa ohjaamattomaan oppimiseen perustuva lähestymistapa: eteenpäin syöttävät ja takaisinkytketyt verkot perustuvat ohjattuun oppimiseen (Wu & Banzhaf, 2010) ja niiden oppimisprosessissa hyödynnetään usein vastavirta-algoritmia *(engl. Backpropagation algorithm)*. Eteenpäin syöttävässä verkossa datan etenemissuunta on yksisuuntainen, jokaisen solmun tulosteesta seuraavan solmun syötteeksi (Venayagamoorthy, 2009). Takaisinkytketty neuroverkko voi tilanteesta riippuen silmukoida datan etenemisen ja käyttää solmun tulostetta aikaisemman kerroksen solmun syötteenä (Kulkarni ym.,

2011). Ohjaamattomaan oppimiseen perustuva itseorganisoituva kartta on sen sijaan yksitasoinen, levymäinen neuroverkko (Wu & Banzhaf, 2010), jonka solmut virittyvät ohjaamattoman oppimisprosessin kautta havaitsemaan kaavoja syötteissä (Kohonen, 1990). Toinen yleisistä ohjaamatonta oppimista noudattava neuroverkon toteutustapa on adaptiivinen resonanssiteoria (Wu & Banzhaf, 2010). Adaptiiviseen resonanssiteoriaan ei perehdytä tässä tutkielmassa, sillä kyseistä neuroverkkometodologiaa on tutkittu tunkeilijan havaitsemisen kontekstissa hyvin vähän, mutta selkeyden ja läpinäkyvyyden nimissä lähestymistapa on esitelty. Kuviossa 2 visualisoidaan neuroverkkojen toteutustapojen jakautumista eri metodologioihin.



KUVIO 2 Neuroverkkokategoriat ja toteutusmallit (muokattu: Wu & Banzhaf, 2010, s. 5)

2.2.3 Evoluutiolaskenta

Käsitteiltään ja perustaltaan vahvasti biologiaan kytkeytyvä evoluutiolaskenta on kokoelma mekanismeja, jotka mallintavat evoluution funktioita kuten luonnonvalintaa ja geneettisyyttä (Eberhart & Shi, 2009, s. 3–9). Evoluutiolaskenta käsittää useita evoluutioalgoritmeja, kuten geneettiset algoritmit, geneettinen ohjelmointi ja kaaosteoria, jotka jokainen mallintavat luonnonvalintaa (Niskanen, 2003, s. 47; Kulkarni ym., 2011). Siinä missä neuroverkot voidaan nähdä yksilön biologista älykkyyttä simuloivana metodologiana, evoluutiolaskennan voidaan nähdä simuloivan kokonaisen lajin aikaan sitoutumatonta älykkyyttä (Eberhart & Shi, 2009, s. 7). Evoluutioalgoritmit pyrkivät valjastamaan luonnon sopeutumisprosessin, jonka avulla lajit yrittävät parantaa eloonjäämismahdollisuuksiaan (Kulkarni ym., 2011). Evoluutiolaskennan perusolettama on, että laji selviytyy lisääntymällä sekä siirtämällä informaatiota geeneissään seuraaville sukupolville niin, että vain parhaat yksilöt selviytyvät ja jatkavat lajin kehityksen kulkua. (Eberhart & Shi, 2009, s. 7.)

Genetiikan tutkimuksen perusväittämä on, että kromosomit koostuvat geeneistä, joista jokainen on eriteltävissä toisistaan sijaintinsa ja funktionsa

perusteella. Geenin funktio voi olla esimerkiksi henkilön silmien värin määrittäminen. Eberhart ja Shi (2009, s. 8) kuvailevat evoluutiolaskennan keinotekoisia kromosomeja biologisten kromosomien abstrahoituina kuvauksina, joiden paikkaan sidotut algoritmiset ominaisuudet simuloivat geenejä ja ominaisuuksista koostuva merkkijono simuloi kromosomia.

Kulkarni ym. (2011) selventävät evoluutiolaskennan kromosomien olevan siis algoritmeja, joista halutaan määrittää sopivin. Kelpoisuusfunktion avulla määritellään yksittäisen kromosomin soveltuvuus sekä toimintakyky. Nämä kaksi ominaisuutta pyritään maksimoimaan algoritmisukupolvien saatossa. Biologista lisääntymisprosessia simuloidaan sekoittamalla kahden tai useamman kromosomin ominaisuuksia mutaatioprosessissa keskenään jälkeläiskromosomeiksi. Jälkeläiskromosomit kokevat pieniä mutaatioita vanhemmista saatujen geenien perusteella, ja algoritmien monimuotoisuus kasvaa. Korkeimman kelpoisuusfunktion kromosomit valitaan seuraavaan sukupolveen, kunnes ratkaisuun soveltuva algoritmi löytyy. (Kulkarni ym., 2011.)

3 TUNKEILIJAN HAVAITSEMISJÄRJESTELMÄT

Tässä luvussa syvennytään johdantolukua perusteellisemmin tarkastelemaan tunkeilijan havaitsemista sekä tunkeilijan havaitsemisjärjestelmiä. Luvussa perehdytään tunkeilijan havaitsemisjärjestelmien ominaispiirteisiin sekä tutustutaan kirjallisuudessa ilmenneihin tunkeilijan havaitsemisen prosessien ongelmakohtiin. Aluksi tehdään lyhyt katsaus tunkeilijan havaitsemisen tarpeeseen sekä ympäristöön tietoturvasektorilla. Sen jälkeen tutustutaan kolmeen yleisimpään havaitsemismetodologiaan: tunnisteisiin perustuvaan tunkeilijan havaitsemiseen, poikkeuksiin perustuvaan tunkeilijan havaitsemiseen ja tilalliseen protokolla-analyysiin.

Tietoyhteiskunnallistumisen seurauksena tietoverkoista ja internetistä on muotoutunut fundamentaalinen tuotannontekijä muun muassa uusien liiketoimintasuuntausten ja hankkeiden alullepanossa ja etenemisessä. Tietoverkoista riippuva liiketoimintasektori vaatii kuitenkin kehittyneitä ja komplekseja tietojärjestelmiä sekä verkostoja, jotka kattavat globalisoituneen maailman nimissä etä- ja langattoman käytön ominaisuudet, salaus- ja autentikointitekniikat, hajautetut tietovarastot sekä verkkopalvelumahdollisuudet. Lisäksi, organisaatioiden yksityiset verkot ja intranetit ovat tulleet laajemmin työntekijöiden ja muiden sidosryhmien saataville, mahdollistaen aikaan ja paikkaan sitomattoman liiketoiminnan. (Patcha & Park, 2007.) Uudet teknologiat ja ominaisuudet edellyttävät yleisen tietoturvakehityksen tapahtuvan yhtäaikaisesti ja samaan suuntaan.

Pfleeger, Pfleeger ja Margulies (2015, s. 2–3) kiteyttävät tietoturvan arvokkaiden ja merkityksellisten objektien eli tietojärjestelmän voimavarojen suojelemiseksi. Voimavaroja on monenlaisia ja niihin kuuluvat muun muassa laitteisto, ohjelmisto, data, käyttäjät ja toiminnot, joista kuitenkin laitteisto, ohjelmisto sekä data ovat yleisimmin hyökkäyksen kohteena. Kyseiset elementit edustavat älyllisiä ja aineettomia pyrkimyksiä tai omaisuutta, kuten esimerkiksi laitteen kiintolevylle tallennettuja tutkimustyön tuloksia, projektidokumentteja tai valokuvia, jotka puolestaan luovat tietojärjestelmän objekteille arvon ja tekevät objekteista suojelun arvoisia. Tietoturva pyrkii siis suojelemaan tietojärjestelmän arvoa ja arvontekijöitä, ja kolme suojeltavaa

pääominaisuutta ovat järjestelmän sekä sen sisältämän datan luottamuksellisuus, eheys ja saatavuus *(engl. Confidentiality, integrity, availability)*, joita kutsutaan myös CIA-kolmioksi *(engl. C-I-A -triad)*. (Pfleeger ym., 2015, s. 2–3, 7.)

Kattava tietoturvakokonaisuus koostuu monista eri tietoturvamekanismeista, joilla jokaisella on oma funktionsa (García-Teodoro, Díaz-Verdejo, Maciá-Fernández & Vázquez, 2009). Järjestelmän rajapinnan mekanismit, kuten palomuuri, autentikointi- ja kulunvalvontamekanismit pyrkivät estämään luvattoman pääsyn järjestelmään (Pfleeger ym., 2015, s. 474), tavoitteenaan säilyttää järjestelmän luottamuksellisuus. Järjestelmän ulkopuolelta saapuvat tietomurrot eivät kuitenkaan ole ainoa tapa loukata järjestelmän tietoturvaa ja CIA-kolmiota. Suurimman osan tietoturvaloukkauksista on osoitettu johtuvan järjestelmän sisäisistä käyttäjistä tai heitä imitoivista tunkeilijoista (Durst ym., 1999). Pfleeger ym. (2016, s. 474) väittävät valtaosan järjestelmän sisältä saapuvien tietoturvaloukkauksien olevan todennäköisesti inhimillisiä virheitä, mutta muistuttavat muun muassa poliittisen vakoilun ja teollisuusvakoilun olevan aina mahdollista. Järjestelmän ulkopuolella sijaitsevat ennalta ehkäisevät turvallisuusmekanismit eivät havaitse järjestelmän sisäisiä rikkeitä, jonka vuoksi tietojärjestelmien ja tietoverkkojen on varauduttava myös reaaliaikaiseen tietoturvarikkeiden havaitsemiseen järjestelmän sisäisillä mekanismeilla. (Pfleeger ym., 2015, s. 474.) Verkkohyökkäysten havaitseminen on kasvanut tärkeäksi tietoturvaprioriteetiksi (Ahmed, Mahmood & Hu, 2016).

Tunkeilijan havaitsemisjärjestelmä, johon tässä tutkielmassa viitataan myös termillä IDS-järjestelmä, voidaan nähdä ensimmäisenä järjestelmän sisäisenä tietoturvan puolustusmekanismina. Se on ohjelmisto, tekninen tietoturvakontrolli ja tyypillisesti erillinen laite (Pfleeger ym., 2015, s. 30, 475), joka automatisoi tunkeilijan havaitsemisen prosesseja (Scarfone & Mell, 2007). Amerikkalaisen *National Institute of Standards and Technology* -viraston määritelmän (Scarfone & Mell, 2007) mukaan tunkeilijan havaitsemisella tarkoitetaan tietojärjestelmän tai tietoverkon tapahtumien valvonnan ja analysoinnin prosesseja, joiden tehtävä on havaita tietoliikenteestä merkkejä tietoturvarikkeistä tai mahdollisista tietoturvauhista. Tietoturvan rikkeiksi ja uhiksi on Scarfonen ja Mellin määritelmässä kategorisoitu organisaatiokohtaisen tietoturvapolitiikan, hyväksyttävien toimintatapojen tai tavanomaisten turvallisuuskäytäntöjen rikkominen. Rikkeet ja uhat ovat monisyisiä ja -selitteisiä: niiden taustalla voi olla muun muassa haittaohjelma, järjestelmän käyttöoikeuksien laiminlyönti tai puhdas inhimillinen virhe, kuten kirjoitusvirhe laitteen IP-osoitetta määritellessä. (Scarfone & Mell, 2007.)

Tunkeilijan havaitsemisjärjestelmän hyvin läheinen käsite on tunkeutumisen estojärjestelmä ja IDS-järjestelmiä käsitellessä onkin syytä tunnistaa erot havaitsemisjärjestelmien ja estojärjestelmien välillä. Tunkeutumisen estojärjestelmä *(engl. Intrusion prevention system, IPS)* toimii IDS-järjestelmän tavoin havainnoiden luvatonta toimintaa ja pyrkii sen lisäksi aktiivisesti estämään järjestelmään tunkeutumisen erilaisilla vastatoimilla.

Tässä tutkielmassa käsitellään kuitenkin tunkeilijan havaitsemisjärjestelmiä, sillä niin IDS- järjestelmien kuin IPS-järjestelmienkin ensisijainen objektiivi on havaita potentiaaliset uhat tietoliikenteestä. (Scarfone & Mell, 2007) Tutkielman tulokset pätevät myös IPS-järjestelmäympäristöön.

Patcha ja Park (2007) ovat verranneet IDS-järjestelmiä kyberavaruuden vastineeksi murtohälyttimille. Ne ovat dynaamisia valvontakokonaisuuksia, ja palomuurien rinnalla tietoturvan fundamentaalisia teknologioita. (Patcha & Park, 2007.) Kuten edellä kävi ilmi, IDS-järjestelmä on ohjelmistotyökalu, jota käytetään auktorisoimattoman toiminnan havaitsemisessa: se kokoaa ja analysoi tietokoneen tai tietoverkon tietoliikenteestä tai järjestelmän sisäisestä toiminnasta johdettua informaatiota havaitakseen mahdolliset turvallisuusrikkomukset ja sisältää reaktiofunktion, joka tyypillisesti reagoi tunkeilevaan toimintaan ilmoittamalla tunkeutumisepäilystä järjestelmänvalvojalle (Patcha & Park, 2007; Pfleeger ym., 2015, s. 475). Muita tunkeilijan havaitsemisjärjestelmän funktioita on esimerkiksi tunnistaa, jos tunkeutumisessa hyödynnetään järjestelmän haavoittuvuutta, kirjata tunkeutumisesta johdettua tietoa järjestelmävalvojalle, tunnistaa järjestelmäkohtaisten turvallisuussääntöjen rikkomuksia, epäilyttävän tiedostonsiirron, kuten suurten tietokantojen kopioinnin käyttäjän omalle laitteelle, tai tiedustelutoiminnan, kuten porttiskannauksen, mikä voi kieliä tulevasta hyökkäyksestä (Scarfone & Mell, 2007). Yhteenvetona, tunkeilijan havaitsemisen prosessi kattaa toiminnot, joiden tehtävä on havaita tietojärjestelmän tai -verkon CIA-ominaisuudet vaarantavat toimet.

Tunkeilijan havaitsemisjärjestelmät voidaan jakaa kategorioihin niiden sijainnin perusteella, jolloin järjestelmät jaetaan tavallisesti isäntäpohjaisiin *(engl. Host-based IDS, HIDS)*, verkkopohjaisiin *(engl. Network-based IDS, NIDS)* (Schepers, 1998; Pfleeger ym., 2015, s. 480) tai langattomiin järjestelmiin *(engl. Wireless IDS, WIDS)* (Scarfone & Mell, 2007), joskin myös vähemmän käytettyjä järjestelmätyyppejä, muun muassa hajautettuja tunkeilijan havaitsemisjärjestelmiä *(engl. Distributed IDS, DIDS)* tavataan (Axelsson, 1999; Maciá-Pérez, Mora-Gimeno, Marcos-Jorquera, Gil-Martínez-Abarca, Ramos-Morillo & Lorenzo-Fonseca, 2011). Isäntäpohjainen tunkeilijan havaitsemisjärjestelmä suojelee yksittäistä laitetta keräämällä ja analysoimalla kyseisen laitteen dataa (Pfleeger ym., 2015, s. 480). Isäntäpohjainen IDS-järjestelmä on asennettava jokaiseen suojelun arvoiseen laitteeseen erikseen, mikä puolestaan heikentää sovellettujen turvatoimien tehokkuutta ja johdonmukaisuutta verrattuna keskitettyyn järjestelmään (Schepers, 1998). Isäntäpohjaisia IDS-järjestelmiä käytetäänkin yleisimmin yksittäisten kriittisten asemien, kuten julkisten palvelimien tai arkaluontoista informaatiota sisältävien palvelimien suojelemisessa (Scarfone & Mell, 2007).

Pfleeger ym. (2015) täsmentävät verkkopohjaisen tunkeilijan havaitsemisjärjestelmän olevan useimmiten erillinen verkkolaite, joka valvoo koko tietoverkon liikennettä. IDS-järjestelmä vastaanottaa lisädataa palomuureilta, verkkoon kytkettyjen laitteiden käyttöjärjestelmiltä, verkon

antureilta sekä järjestelmänvalvojilta suojellakseen koko tietoverkkoa niin ulkopuolisilta kuin sisäisiltä hyökkäyksiltä. (Pfleeger ym., 2015, s. 480.)

Langaton tunkeilijan havaitsemisjärjestelmä valvoo langatonta tietoliikennettä analysoiden langattomia verkkoprotokollia (Scarfone & Mell, 2007). Langaton IDS-järjestelmä on kuitenkin sidottu TCP/IP-mallin terminologian mukaan linkkikerrokselle, eikä voi tarkkailla sovellus-, kuljetus- tai verkkokerroksella tapahtuvia toimintoja (Liao, Lin, Lin & Tung, 2012; Scarfone & Mell, 2007). Järjestelmän sijainnin lisäksi IDS-järjestelmät kategorisoidaan järjestelmäkohtaisten havaitsemismetodologioiden perustella. Seuraavassa alaluvussa tutustutaan kolmeen yleisimpään havaitsemismetodologiaan.

3.1 Yleiset havaitsemismetodologiat

Kuten edellä todettiin, tunkeilijan havaitsemisjärjestelmät ryhmitellään tyypillisesti niiden verkkosijainnin perusteella isäntäpohjaisiin, verkkopohjaisiin tai langattomiin tunkeilijan havaitsemisjärjestelmiin. Toinen vallitseva luokittelutapa on jakaa havaitsemisjärjestelmät niiden noudattamien havaitsemismetodologioiden perusteella lähteestä riippuen vaihtelevasti kahteen tai kolmeen yleisimpään tyyppikategoriaan. Nämä kategoriat ovat tunnisteisiin perustuvat tunkeilijan havaitsemisjärjestelmät *(engl. Signature-based intrusion detection systems)*, poikkeuksiin perustuvat järjestelmät *(engl. Anomaly-based intrusion detection systems)* ja tilalliseen protokolla-analyysiin perustuvat järjestelmät *(engl. Stateful protocol analysis)* (Liao, Lin, Lin & Tung, 2012; Patcha & Park, 2007; Scarfone & Mell, 2007). Tässä alaluvussa tutustutaan kyseisiin havaitsemismetodologioihin sekä niitä noudattavien IDS-järjestelmien eroihin, etuihin ja ongelmakohtiin.

3.1.1 Tunnisteisiin perustuvat järjestelmät

IDS-järjestelmien kontekstissa tunnisteella tarkoitetaan tunnettua hyökkäystyyppiä vastaavaa kaavaa, joka on verrannollistettu englanninkielisessä terminologiassa sormenjäljeksi. Sormenjälki-vertauskuva on kuvaava, sillä tunnisteisiin perustuva tunkeilijan havaitseminen viittaa prosessiin, jossa tunnisteita verrataan havaittuihin tapahtumiin mahdollisten vaaratilanteiden sekä rikkeiden tunnistamiseksi (Scarfone & Mell, 2007), mikä puolestaan vastaa yksinkertaisuudessaan hyvin paljon sormenjälkitutkimusta.

Tunnisteisiin perustuvat IDS-järjestelmät suorittavat yksinkertaisia kaavantunnistusprosesseja etsien tapahtumia tietoliikenteestä, jotka vastaavat tunnettujen hyökkäystyyppien tunnisteita (Pfleeger ym, 2015, s. 476). Kerätyt datansiirtopaketit analysoidaan käyttäen hyödyksi hahmontunnistusalgoritmeja. Jos datasta löydetään rikkomuksia, järjestelmä laukaisee järjestelmänvalvojalle hälytyksen. (Patcha & Park, 2007.)

Tunniste voi olla esimerkiksi tapahtumakaava palvelunestohyökkäyksestä, jossa hyökkääjä lähettää tunkeutumisen kohteena olevan järjestelmän palvelimelle useita TCP SYN yhteyden avaus -paketteja. Tunnisteisiin perustuva järjestelmä havaitsee hyvin tietynlaiset hyökkäykset, kuten muun muassa ping-hyökkäys tai echo-chargen-hyökkäys, joilla molemmilla on normaalista tietoliikenteestä poikkeavat pakettityypit (Pfleeger ym., 2015, s. 477; García-Teodoro ym., 2009).

Toisaalta, tunnisteisiin perustuvien järjestelmien kompastuskivi ovat juuri tunnisteet, sillä järjestelmä havaitsee ainoastaan tunnistetta vastaavan tietoliikenteen. Hyökkääjän on siis mahdollista muuntaa hyökkäystä niin, ettei se enää vastaa järjestelmään tallennettua tunnistetta. Muutosten ei tarvitse olla suuria: yksittäiset symbolimuutokset tai pakettien järjestyksen muuttaminen voi riittää. Tämän vuoksi tunnisteisiin perustuvat IDS-järjestelmät ovat valideja vain staattisten, tunnettujen hyökkäystyyppien havaitsemisessa (Pfleeger ym, 2015, s. 476–477).

Myös tietynlaiset hyökkäystyypit ovat tälle havaitsemismetodologialle haastavia. Näistä esimerkkinä voidaan pitää hyökkäyksiä, jotka koostuvat useista paketeista. Järjestelmä ei pysty havaitsemaan hyökkäystä, ennen kuin kaikki paketit tai suurin osa pakettifragmenteista ovat saapuneet perille. Tämän tekee käytännössä mahdottomaksi se, että nykyisessä tietoliikenteessä lähes kaikki data ja paketit ovat fragmentoituja, jolloin tunnisteisiin perustuvan järjestelmän pitäisi kerätä kaiken liikenteen kaikki fragmentit, verratakseen pakettikokonaisuutta hyökkäystunnisteiden tietokantaan. (Pfleeger ym, 2015, s. 477.) Tunnisteisiin perustuvat IDS-järjestelmät eivät myöskään tallenna lokia menneestä tietoliikenteestä, mikä estää järjestelmät havaitsemasta hyökkäyksiä, jotka koostuvat useista paketeista, ellei yksittäinen saapunut paketti sisällä selvää tunkeutumiseen viittaavaa tunnistetta (Scarfone & Mell, 2007).

Vastapainona ongelmakohdille tunnisteisiin perustuvien tunkeilijan havaitsemisjärjestelmien etuna tunnetut hyökkäystyypit havaitaan luotettavasti ja vähäisin virheellisin positiivisin FP-luvuin *(engl. false positive, FP)* viitaten vääriin hälytyksiin. Tämän tyyppisten järjestelmien fundamentaalinen haitta on kuitenkin niiden pätemättömyys uusien tai muokattujen hyökkäysten havaitsemisessa (Scarfone & Mell, 2007; Patcha & Park, 2007). Kuten yllä mainittiin, muun muassa symbolimuutoksilla, erilaisilla välttelymetodeilla *(engl. Evasion techniques)* kuten pakettien jakamisella, lisäämällä duplikaattilohkoja, paketin hyötykuorman mutaatiolla tai shellcode-mutaatiolla (Cheng, Lin, Lai & Lin, 2012) sekä hyökkäyksen eri versioilla on mahdollista kiertää tunniste sekä tunnisteisiin perustuvat havainnointimetodit (Scarfone & Mell, 2007; Pfleeger ym., 2015, s. 477). Ollakseen luotettava, tunnisteisiin perustuva järjestelmä tarvitsisi tyyppitunnisteen kaikista olemassa olevista ja mahdollisista hyökkäyksistä.

Patcha ja Park (2007) tuovat esille toisen vallitsevan ongelmakohdan tunnisteisiin perustuvissa IDS-järjestelmissä: tunnisteiden pätevään ylläpitoon, kokoamiseen ja käsittelyyn liittyvät ongelmat. Järjestelmän todennäköisesti tärkein objekti on tunnisteet sisältävä tietokanta, joka vaatii jatkuvaa

päivittämistä ollakseen pätevä, näin ollen myös runsas kuluerä. Myös tunnisteiden kokoamisen on havaittu olevan haastavaa, erityisesti hyökkäyksen tapahtuessa useamman erillisen paketin avulla. Tällöin tunniste on koottava useammasta paketista, mikä on todettu vaikeaksi. (Patcha & Park, 2007.)

Pfleeger ym. (2015) tiivistävät tunkeilijan havaitsemisen tunnisteiden avulla olevan hankalaa tapauksissa, joissa havaittava kaava on muuttuva tai pitkä. Koska tunnisteet rajoittuvat vain tiettyihin tunnettuihin hyökkäysmalleihin, toinen tunkeutumisen havaitsemismetodologia on olennainen. (Pfleeger ym, 2015, s. 477–479.)

3.1.2 Poikkeuksiin perustuvat järjestelmät

Poikkeuksien havaitsemisen prosessissa verrataan normaaliksi määriteltyä verkkotoimintaa järjestelmässä havaittuihin tapahtumiin merkittävien poikkeamien tunnistamiseksi (Scarfone & Mell, 2007). Ahmed, Mahmood ja Hu (2016) viittaavat artikkelissaan Hawkinsin (1980) usein viitattuun poikkeavuuden määritelmään:

> Poikkeama on havainto, joka poikkeaa niin paljon muista havainnoista, että herättää epäilyksiä siitä, että se on syntynyt eri mekanismilla.

Poikkeuksiin perustuvat järjestelmät etsivät siis epätavallista järjestelmätoimintaa, mikä viittaa harvinaisiin ja merkittäviin toimintoihin, jotka voivat esimerkiksi kieliä laitteen vaarantuneisuudesta ja datan siirrosta luvattomille osapuolille (Ahmed, Mahmood & Hu, 2016). Poikkeuksiin perustuvat IDS-järjestelmät toimivat luomalla ensin lähtötilanteen profiilin järjestelmäkäytöksestä, josta eteenpäin kaikki järjestelmän toiminta nähdään potentiaalisesti tunkeilevana käytöksenä (Patcha & Park, 2007). Profiilit voivat olla staattisia tai dynaamisia ja niitä voidaan editoida käyttöönoton jälkeen (Liao, Lin, Lin & Tung, 2012; Scarfone & Mell, 2007).

Scarfone ja Mell (2007) havainnollistavat poikkeuksiin perustuvan havaitsemismetodologian toimintaa. Poikkeuksiin perustuva järjestelmä sisältää käytösprofiileja, jotka mallintavat esimerkiksi käyttäjien, laitteiden, verkkoyhteyksien tai sovellusten normaalia käyttäytymistä. Profiilit voivat olla laajoja ja sisältää monia normaalikäytöstä mallintavia attribuutteja, kuten kyseisen laitteen prosessorin käytön tason, epäonnistuneiden kirjautumisten lukumäärän tai käyttäjän lähettämien sähköpostien määrän (Liao, Lin, Lin & Tung, 2012). Esimerkiksi verkkoyhteyden normaalikäytöstä mallintava profiili voi määritellä tyypillisen työpäivän keskimääräisen verkon kaistaleveyden kulutuksen. Poikkeuksiin perustuva IDS-järjestelmä vertaa tilastollisin metodein tietoverkon hetkellistä toimintaa verkkoyhteyden normaalin käytöksen profiiliin, ja havaitessaan merkittäviä poikkeamia järjestelmä laukaisee hälytyksen. (Scarfone & Mell, 2007.)

Sen ohella, että poikkeuksiin perustuva järjestelmä tarkastelee yksittäisten käyttäjien toimintaa havaitakseen epäilyttävän ja hälyttävän toiminnan, järjestelmä analysoi jatkuvasti myös koko järjestelmän likaisuuden tasoa ja

hälyttää järjestelmänvalvojalle, kun järjestelmän kokonaistoiminta ylittää ennalta määritellyn epäilyttävän kokonaistoiminnan tason (Pfleeger ym, 2015, s. 478).

Patcha ja Park (2007) tuovat tutkielmassaan esille poikkeuksiin perustuvien järjestelmien tarjoamat kolme huomattavaa etua verrattuna tunnisteisiin perustuviin järjestelmiin. Ensimmäisenä, poikkeuksiin perustuvat järjestelmät kykenevät tunnistamaan järjestelmän sisältä lähtöiset hyökkäykset. Jos järjestelmän auktorisoidun käyttäjän käyttäjätunnus varastetaan ja hyökkääjä pyrkii toimimaan vahingoittavalla tavalla, toimintojen poiketessa käyttäjäkohtaisesta normaalin toiminnan profiilista järjestelmä laukaisee hälytyksen. Toinen poikkeuksiin perustuvien järjestelmien etu on käytösmallinnuksien järjestelmä-, ohjelmisto- tai tietoverkkokohtainen personointi. Koska poikkeuksien havainnointi perustuu järjestelmäkohtaisiin räätälöityihin käytösprofiileihin, se vaikeuttaa uuden hyökkäyksen suunnitteluvaihetta, sillä hyökkäystä suunnitellessa ei voida tietää, mitä hyökkäyksen kohteen IDS-järjestelmän käytösmallinnuksissa on määritelty normaalikäytökseksi. Kolmantena, poikkeuksiin perustuvat järjestelmät kykenevät tunnistamaan uudet, ennennäkemättömät, nollapäivähaavoittuvuuksia hyödyntävät hyökkäystyypit. Luvattomasta toiminnasta syntyy hälytys, koska se eroaa järjestelmän tai käyttäjän normaalista toiminnasta. Poikkeuksiin perustuvat IDS-järjestelmät eivät siis tarvitse vastaavaa alituista tietokannan päivitystä, kuten tunnisteisiin perustuvat IDS-järjestelmät. (Patcha & Park, 2007.)

Valteistaan huolimatta poikkeuksiin perustuvissa järjestelmissä on myös useita ongelmakohtia (Scarfone & Mell, 2007; Patcha & Park, 2007; Ahmed, Mahmood & Hu, 2016; Liao, Lin, Lin & Tung, 2012). Näihin lukeutuvat muun muassa järjestelmien luontainen monimutkaisuus (Patcha & Park, 2007) sekä väärien hälytysten suuri määrä (Scarfone & Mell 2007; Patcha & Park, 2007; García-Teodoro ym., 2009), universaalisti sovellettavien tekniikoiden puute, datan kohina, sekä normaalin käytöksen kehittyminen (Ahmed, Mahmood & Hu, 2016).

Patcha ja Park (2007) tuovat poikkeuksiin perustuvien ilmeisimpänä ongelmana ilmi järjestelmän suuritöisen valmistelu- ja koulutusjakson, jonka aikana ensisijaiset käyttäjäprofiilit luodaan määrittelemällä järjestelmän ja yksittäisten käyttäjien normaalin toiminnan profiilit. Normaalia käyttäytymistä mallintavan profiilin luominen on kuitenkin haastavaa ja epäpätevä normaalin käytöksen mallintaminen johtaa helposti järjestelmän huonoon suoriutumiseen (Patcha & Park, 2007). Myös dynaamisten profiilien ylläpitäminen kuluttaa paljon aikaa ja resursseja (Liao, Lin, Lin & Tung, 2012; Patcha & Park, 2007; Scarfone & Mell, 2007).

Patcha ja Park (2007) selventävät poikkeuksiin perustuvien järjestelmien havaitsevan poikkeavaa toimintaa, eivätkä suoranaisesti hyökkäyksiä, minkä vuoksi kyseiset järjestelmät ovat hyvin alttiita aikaa vieville väärille hälytyksille. Tämä on poikkeuksiin perustuvien IDS-järjestelmien toinen olennainen heikkous. Väärillä hälytyksillä tarkoitetaan virheellisiä positiivisia *(engl. False*

positive, FP) hälytyksiä, jolloin järjestelmä raportoi tunkeilevaksi käytökseksi toimintoja, jotka ovat oikeutettuja ja perusteltuja verkkotoimintoja (García-Teodoro ym., 2009; Patcha & Park, 2007). FP-hälytykset kuluttavat järjestelmän resursseja, mikä voi johtaa virheellisiin negatiivisiin tuloksiin *(engl. False negative, FN)*. Virheellisten positiivisten tulosten sivutuotteena pahansuopa toiminta tai hyökkäys voi FP-hälytysten ja niiden kuluttamien resurssien takia jäädä huomaamatta (Patcha & Park, 2007). Tätä epäonnistunutta hyökkäyksen havaitsemista kutsutaan virheelliseksi negatiiviseksi eli FN-tulokseksi (García-Teodoro ym., 2009).

Poikkeuksiin perustuvan IDS-järjestelmän käyttöönotto koostuu tavallisesti kahdesta vaiheesta: koulutusvaiheesta ja testausvaiheesta (Patcha & Park 2007). Koulutusvaiheessa määritellään järjestelmän normaalin käytöksen mallinnusprofiilit (Scarfone & Mell, 2007) ja testausvaiheessa profiileja sovelletaan uuteen dataan (Patcha & Park, 2007). Koulutus- ja testausvaihe sisältävät kuitenkin myös heikkouden. Jos ilkivaltainen käyttäjä on päässyt järjestelmään käsiksi ennen poikkeuksiin perustuvan IDS-järjestelmän koulutusta, hänen voi olla mahdollista kouluttaa IDS-järjestelmä vaiheittain hyväksymään luvattoman käytöksen normaalina. (Patcha & Park, 2007; Scarfone & Mell, 2007.)

Scarfonen ja Mellin (2007) mukaan tahaton haitallisen käytöksen sisällyttäminen profiiliin on yleinen ongelma. Jo pelkästään pätevän käytösprofiilin rakentaminen voi olla tilanteesta riippuen hyvin haastavaa, esimerkiksi ympäristössä, jossa suuria ylläpitotoimintoja, kuten laajamittaista tiedostojen siirtoa, tapahtuu harvoin, eivätkä kyseiset toiminnot tapahdu IDS-järjestelmän käyttöönoton aikana. Tämä kytkeytyy takaisin väärien hälytysten ongelmaan, sillä poikkeuksiin perustuvat järjestelmät ovat varsinkin dynaamisissa ympäristöissä taipuvaisia hälyttämään vaarattomasta, suotuisasta järjestelmätoiminnasta, joka poikkeaa ennalta määritellystä käytösprofiilista. (Scarfone & Mell, 2007.)

Kumar ja Spafford (1994) esittävät tutkielmassaan tunkeilijan havaitsemisen keskinäisenä lähtökohtana tukeutumisen olevan poikkeavan käytöksen osajoukko. He tarkastelevat esimerkkinä tunkeilijaa, jolla ei ole ensisijaista tietoa auktorisoidun käyttäjän normaaleista toimintamalleista järjestelmässä. Tunkeutujan toiminta todetaan hyvin todennäköisesti poikkeavaksi käytökseksi. Ihannetapauksessa poikkeavien toimintojen joukko on sama kuin tunkeutumistoimintojen joukko ja tällaisessa tapauksessa kaiken poikkeavan toiminnan määritteleminen tunkeilevaksi toiminnaksi ei johda FP- tai FN-tuloksiin. Tunkeileva toiminta ei kuitenkaan aina ole yhtenevää poikkeavan toiminnan kanssa. (Kumar & Spafford, 1994.)

Kun mahdollisimman suuri määrä hyökkäyksiä pyritään havaitsemaan ja virheellisiä negatiivisia ilmoituksia minimoidaan, poikkeavan toiminnan kynnys on asetettava alhaiseksi. Tämä kuitenkin johtaa moniin virheellisiin positiivisiin tuloksiin ja vaikuttaa negatiivisesti järjestelmän automaattisten mekanismien tehokkuuteen sekä kasvattaa järjestelmänvalvojan taakkaa, sillä jokainen hälytys on tutkittava ja FP-tapaukset eroteltava. (Patcha & Park, 2007.)

3.1.3 Tilallinen protokolla-analyysi

Scarfone ja Mell (2007) täsmentävät tilallisen protokolla-analyysin prosessiksi, jossa verrataan ennalta määriteltyjä profiileja yleisesti hyväksytyistä vaarattoman protokollatoiminnan tilojen määritelmiä havaittuja tapahtumia vastaan poikkeamien tunnistamiseksi. Ero tilallisen protokolla-analyysin sekä poikkeuksiin perustuvien havaitsemismetodologian välillä on tilallisen protokolla-analyysin perustuminen protokollan kehittäjätahon määrittelemiin yleisiin profiileihin, joissa määritellään, miten tiettyjä protokollia tulisi ja ei tulisi toimia, kun taas poikkeuksiin perustuvat IDS-järjestelmät pohjautuvat järjestelmäkohtaisiin käytösprofiileihin. (Scarfone & Mell, 2007.)

Tilallisuuden termillä viitataan tilallista protokolla-analyysia noudattavan IDS-järjestelmän kykyyn tunnistaa ja jäljittää protokollan tilat, kuten yhdistää pyynnöt *(engl. Requests)* ja vastaukset *(engl. Replies)* (Liao, Lin, Lin & Tung, 2012; Scarfone & Mell, 2007). Pfleeger ym. (2015) esittää tilallisen protokolla-analyysin SYN flood -hyökkäys -esimerkillä. Kyseisellä hyökkäyksellä on yksinkertainen, keskeneräisen kolmitiekättelyn toimintakaava, mutta kaavan kolme ominaista piirrettä (saapuva SYN-paketti, SYN-ACK-kuittauspaketti ja saapumaton ACK-paketti) hajaantuvat eri ajan hetkille. Hyökkäyksen havaitseminen vaatii IDS-järjestelmältä hyökkäyksen ensimmäisen vaiheen tunnistamisen, myöhemmin toisen vaiheen löytämisen ja lopulta, kohtuullisen ajan odotuksen jälkeen, järjestelmä voi päätellä kyseessä olevan SYN flood -hyökkäys. (Pfleeger ym., 2015, s. 479.)

Tilallista protokolla-analyysiä hyödyntävä IDS-järjestelmä näin ollen ymmärtää ja tallentaa protokollan kyseisen hetken tilan ja vertaa ennalta määriteltyä protokollan toiminnan mallia kyseisen hetken havaittua toimintaan (Scarfone & Mell, 2007; Mudzingwa & Agrawal, 2012). Kuten yllä mainittiin, ennalta määritellyt toimintamallit perustuvat pääasiassa ohjelmistotoimittajien sekä standardointi organisaatioiden protokollastandardeihin. Protokollamalleissa on tyypillisesti hieman varaa joustolle ja ne ottavat huomioon kunkin protokollan toteutuksen vaihtelut. Tämä kuitenkin vaikeuttaa tarkan protokolla-analyysin suorittamista. (Scarfone & Mell, 2007.)

Scarfone ja Mell (2007) nimeävät tilallisen protokolla-analyysin ensisijaiseksi haittapuoleksi sen korkean resurssien kulutuksen analyysin monimutkaisuuden ja useiden samanaikaisten istuntojen ja yhteyksien tilan seurannan yhteydessä. Toinen huomattava ongelma on tilallisen protokolla-analyysin pätemättömyys hyökkäyksien, jotka eivät riko yleisesti hyväksyttävän protokollakäyttäytymisen ominaisuuksia, havaitsemisessa. Esimerkiksi palvelunestohyökkäys hyödyntää vaarattomiksi määriteltyjä protokollatoimintoja ylikuormittaakseen hyökkäyksen kohteena olevan palvelimen ja estääkseen palvelun saatavuuden. Mudzingwa ja Agrawal (2016) myötäilevät ja muistuttavat, että vaikka tilallisen protokolla-analyysin metodeilla on syvä ymmärrys eri protokollista, se voidaan helposti kiertää

hyökkäyksillä, jotka noudattavat protokollien hyväksyttävää toimintaa. Kolmas tilallisen protokolla-analyysin ongelmakohdista on protokollien moniselitteisyydestä johtuva standardoitujen protokollamallien sekä eri käyttöjärjestelmien ja sovellusten käyttämien protokollien toteutusten välillä ilmenevät mahdolliset ristiriidat, ja näistä ristiriidoista johtuvat väärät hälytykset tunkeilijan havaitsemisen prosessissa. (Scarfone & Mell, 2007)

Mudzingwa ja Agrawal (2016) huomauttavat, että tilallinen protokolla-analyysi nähdään enää harvoin erillisenä toteutettavana havaitsemismetodologiana. Se on nykyään lähes aina integroitu osa hybridi-IDS-järjestelmää, jossa yhdistellään useita eri havaitsemismetodologioita. Suurin osa tunkeilijan havaitsemisjärjestelmiä koskevasta tutkimuksesta keskittyy pääasiassa poikkeuksiin perustuviin järjestelmiin, tunnisteisiin perustuviin järjestelmiin sekä hybridijärjestelmiin. (Mudzingwa & Agrawal, 2016.)

3.2 Tunkeilijan havaitsemisjärjestelmien edut ja ongelmat

Riippumatta metodologiasta tai järjestelmän sijainnista, IDS-järjestelmien fundamentaaliset edut ovat tunkeutumisten havaitseminen sekä tietoturvarikkeeseen reagoinnin tukeminen. Primäärihyötyjen lisäksi järjestelmien on havaittu auttavan organisaation tietoturvasuunnitelman objektiivisessa laaduntarkkailussa, olemassa olevien tietoturvauhkien dokumentoinnissa sekä järjestelmän sisäisten käyttäjien aiheuttaman tahallisen ilkivallan reguloinnissa (Scarfone & Mell, 2007).

García-Teodoro ym. (2009) kuvailevat tunnisteisiin perustuvia ja poikkeuksiin perustuvia järjestelmiä käsitteellisyytensä sekä toimintansa kannalta hyvin samanlaisiksi, ja metodologioiden olennaisimmat erot liittyvät hyökkäyksen ja poikkeuksen käsitteisiin. Hyökkäys määritellään toiminnaksi, joka on riski järjestelmän turvalliselle toiminnalle, siinä missä poikkeama määritellään turvallisuuden kannalta epäilyttäväksi tapahtumaksi. Tämä ero onkin olennainen kunkin havaitsemismetodologian tärkeimpien etujen ja haittojen osoittamisessa. (García-Teodoro ym., 2009)

Perustavanlaatuisuudestaan huolimatta nykyiset kaupalliset tunkeilijan havaitsemisjärjestelmät ovat pääasiassa tunnisteisiin perustuvia järjestelmiä (García-Teodoro ym., 2009). Scarfone ja Mell (2007) nimeävät IDS-järjestelmien yleisimmäksi negatiiviseksi attribuutiksi niiden luottamattomuuden, sillä kyseiset järjestelmät eivät kykene täydelliseen ja virheettömään tunkeutumisten havaitsemiseen. Tunkeilijan havaitsemisjärjestelmät ovat kuitenkin jatkuvasti kehittyviä tuotteita, joiden tuotekehitykseen vaikuttaa yleinen tietoturvatutkimus. Yleinen tietotekniikan kehitys on tuonut IDS-järjestelmät jo enemmän saataville niin hintaluokaltaan kuin ylläpidoltaankin (Pfleeger ym, 2015, s. 488). García-Teodoro ym. (2009) panevat merkille poikkeuksiin perustuvien, verkkopohjaisten IDS-järjestelmien olevan tällä hetkellä tunkeilijan havaitsemisen alan tutkimus- ja kehitystoiminnan tärkein

keskittymisen kohde. Aihe ei kuitenkaan ole vielä kypsä ja keskeiset kysymykset on ratkaistava ennen kuin poikkeuksiin perustuvien järjestelmien laajamittainen käyttöönotto on mahdollista (García-Teodoro ym., 2009).

4 PEHMOLASKENTAMETODIT TUNKEILIJAN HAVAITSEMISESSA

Tässä luvussa pureudutaan yleisimpien pehmolaskentametodien käyttöön sekä soveltuvuuteen tunkeilijan havaitsemisjärjestelmissä saatavilla olevaan havainnollistavaan kirjallisuuteen perustuen. Tutkielman rajauksen mukaan luvussa käsitellään kolmea vallitsevinta pehmolaskentametodia, jotka ovat sumea logiikka, neuroverkot, evoluutiolaskenta. Myös edellä mainittujen metodien yhdistelmälähestymistapoja tarkastellaan tunkeilijan havaitsemisen ongelmien ratkaisun näkökulmasta. Metodeja sekä niiden käyttöä tutkitaan edellä mainitussa järjestyksessä. Jokaista metodia lähestyessä tarkastellaan sen käyttöä nykyisissä kaupallisissa IDS-järjestelmissä sekä sen sovellettuja, uusia käyttötarkoituksia IDS-tutkimuksessa. Lopuksi tarkastellaan pehmolaskentametodien soveltamisen ongelmakohtia tunkeilijan havaitsemisessa.

Kuten edellisessä luvussa mainittiin, valtaosa saatavilla olevista tunkeilijan havaitsemisjärjestelmistä ovat tunnisteisiin perustuvia järjestelmiä. Sommer ja Paxson (2010) toteavat alan tutkimusyhteisön tutkivan niin tunnisteisiin perustuvaa kuin poikkeuksiin perustuvaa havaitsemismetodologiaa laajasti, mutta todellisen metodologioiden käyttöönoton osalta on havaittavissa silmiinpistävä epätasapaino: toiminnallisissa asetuksissa näistä kahdesta IDS-järjestelmäpääluokasta löydämme lähes yksinomaan vain tunnisteisiin perustuvia järjestelmiä. Epätasapaino herättää hämmennystä, kun otetaan huomioon koneoppimismetodien menestys muilla tietojenkäsittelytieteen aloilla. (Sommer & Paxson, 2010.) Kysymys siitä, miksi näitä päteväksi todettuja menetelmiä on hankalaa soveltaa poikkeuksiin perustuvaan tunkeilijan havaitsemiseen, on todennäköisesti yksi syy aiheen herkeämättömään tutkimukseen ja kiinnostukseen. Myös nykyisen tietoliikenteen tunnusmerkkeihin kuuluvat rakenteellinen monimutkaisuus, automaatio, mutaatiokyky sekä yleinen haitallisuus (Pang, Yegneswaran, Barford, Paxson & Peterson, 2004) vaikuttavat poikkeuksien havaitsemisen tutkimuksen tarpeeseen, sillä kyseiset

ominaisuudet vaikeuttavat tunnisteisiin perustuvan tunkeilijan havaitsemisen suoriutumista järjestelmän suojaamisesta (Langin & Rahimi, 2010).

Lazarevic, Kumar ja Srivastava (2005) nimeävät tutkielmassaan mittavan poikkeuksien havaitsemisalgoritmien joukon olevan luokiteltavissa viiteen kategoriaan: tilastollisiin metodeihin, etäisyyksiin perustuviin metodeihin, sääntöihin perustuviin järjestelmiin, profilointimenetelmiin, sekä mallipohjaisiin lähestymistapoihin. García-Teodoro ym. (2009) ovat tiivistäen supistaneet yllä mainitut viisi kategoriaa kolmeen: tilastollisiin tekniikkoihin, tietoon perustuviin *(engl. Knowledge-based)* tekniikkoihin sekä koneoppimistekniikkoihin. He määrittelevät koneoppimistekniikoiden perustuvan eksplisiittisen normaalin käytösmallin luomiseen, jonka avulla analysoitujen havaintojen luokittelu vaarattomaksi tai epäilyttäväksi on mahdollista. Distinktiona tilastolliset tekniikat perustuvat satunnaisesti luotuun normaaliin käytösmalliin, kun taas tietoon perustuvat tekniikat kokoavat oletettavan normaalin käyttäytymisen mallinnuksen saatavilla olevista järjestelmätiedoista. (García-Teodoro ym., 2009.)

Tämän tutkielman tutkimuskohteena olevat pehmolaskentametodit sijaitsevat edellä mainitun algoritmiluokittelun mukaan koneoppimistekniikoiden joukossa. Koneoppimistekniikat, ja näin ollen kategorisen yhteyden kautta myös pehmolaskentametodit, ovat kykeneväisiä muuttamaan toteutusstrategiaansa uuden informaation valossa (García-Teodoro ym., 2009). Wu ja Banzhaf (2010) muistuttavat kyseisen, sopeutuvan ominaisuuden olevan IDS-järjestelmien merkittävä tekijä, sillä sekä tunkeutuva toiminta että käyttäjien, järjestelmien ja verkkojen oikeutettu toiminta kehittyvät ajan myötä. Jos tunkeilijan havaitsemisjärjestelmä ei kykene sopeutumaan käytösmuutoksiin, tunkeutumisen havaitsemisen tarkkuus laskee huomattavasti (Wu & Banzhaf, 2010).

Pehmolaskentametodeja on sovellettu kattavasti tunkeilijan havaitsemisen ongelmiin ja kyseisten metodien soveltuvuutta verkkopohjaiseen poikkeuksiin perustuvaan tunkeilijan havaitsemiseen on tutkittu hyvin aktiivisesti (Toosi & Kahani, 2007; Sommer & Paxson, 2010). Wu ja Banzhaf (2010) väittävät todennäköiseksi syyksi tutkimusaihetta ympäröivään kiinnostukseen pehmolaskennan tuottamat edut havaitsemisjärjestelmien suorituskykyyn. IDS-järjestelmien suorituskyvyn arviointiin liittyy vahvasti järjestelmän kyky tehdä oikeita ennusteita. Yleisimmät arviointikriteerit järjestelmän suorituskyvyn määrittelyyn ovat havaitsemisnopeus *(engl. Detection rate, DR)* yhdessä väärien hälytysten määrän *(engl. False alarm rate, FAR)* kanssa. Pätevän IDS-järjestelmän suoritustavoite on korkea DR ja alhainen FAR. (Wu & Banzhaf, 2010.)

Kaupalliset IDS-järjestelmät soveltavat päteviksi todennettuja tekniikoita sekä metodeja ja harkitsevat harvoin uusimpien innovaatioiden soveltamista järjestelmiin. Siitä huolimatta, poikkeuksiin perustuvia järjestelmiä sekä hybridijärjestelmiä tutkitaan kattavasti. Hybridijärjestelmien kahden tai useamman havaintometodologian yhdistelmät pyrkivät parantamaan järjestelmien suorituskykyä välttäen kunkin metodologian tavanomaiset sudenkuopat, kuten korkeat FP-luvut. (García-Teodoro ym., 2009.)

4.1 Sumea logiikka tunkeilijan havaitsemisessa

Bridges ja Vaughn (2000) perustelevat sumean logiikan tekniikoiden käyttöä poikkeuksien havaitsemisessa pääasiassa siksi, että tarkasteltavia ominaisuuksia voidaan pitää sumeina muuttujina. Tunkeilijan havaitsemisessa kvantitatiivisten mittausten perusteella määritellyt käyttäytymisen normaalinarvojen ja epänormaaliarvojen joukot ovat äkillisiä ja jyrkkiä: kaikki normaali toiminta on yhtä normaalia ja kaikki epänormaali toiminta on samassa määrin epänormaalia. Näiden kvantitatiivisten piirteiden sumentaminen auttaa tasoittamaan normaalin ja poikkeavan äkillisiä eroja ja tarjoaa mitan tietyn toimenpiteen normaaliudesta tai poikkeavuudesta. (Bridges & Vaughn, 2000.) Samankaltaista perustelua heijastuu Wun ja Banzhafin (2010) artikkelissa, jossa tutkijat toteavat tunkeilijan havaitsemismallien kokoamisen suoraan kvantitatiivisista arvoista aiheuttavan virheitä järjestelmän toiminnassa: esimerkiksi vain hieman normaalista poikkeava vaaraton toiminta voi laukaista väärän hälytyksen.

Wu ja Banzhaf (2010) selventävät sumean logiikan käsittelevän numeerista dataa kielellisessä muodossa, mikä mahdollistaa järjestelmän suuren numeerisen syöteavaruuden supistamisen pienempään kielellisten arvojen hakuavaruuteen. Kielellisten arvojen sisältämän epätarkkuuden avulla on mahdollista luoda joustavampia tunkeilijan havaitsemisjärjestelmiä, mikä lisää kyseisten järjestelmämallien mukautumiskykyä ja luotettavuutta. Kielellisten muuttujien käyttö mahdollistaa myös normaalien ja epänormaalien käytösmallien helppokäyttöisyyden sekä ymmärrettävyyden, mikä parantaa IDS-järjestelmän luotettavuutta. (Wu & Banzhaf, 2010.)

Sumeaa logiikkaa on sovellettu tunkeilijan havaitsemisen prosesseihin sumeiden sääntöjen ja sumeiden päättelyalgoritmien avulla (Langin & Rahimi, 2010). Wu ja Banzhaf (2010) mainitsevat kyseisten sumeiden sääntöjen olevan käytetty tunnisteisiin perustuvassa tunkeilijan havaitsemisessa usein osana asiantuntijajärjestelmiä, joissa sumeilla IF-THEN säännöillä on korvattu perinteinen järjestelmän sääntöpohja. Laskennallisen älykkyyden kiivaan kehityksen takia lähestymistapoja, jotka osoittavat oppimisen ja adaptiivisuuden ominaisuuksia, kuten neuroverkot sekä evoluutiolaskenta, on sovellettu sumeiden sääntöjen generoinnin automatisointiin. Samoja tekniikoita on sovellettu myös poikkeuksiin perustuvan tunkeilijan havaitsemisen tutkimukseen, jossa eräs kiinnostuksen kohteista on ollut sumean normaalikäytöksen mallinnuksen kokoaminen. (Wu & Banzhaf, 2010.)

García-Teodoro ym. (2009) väittävät, että vaikka sumea logiikka on osoittautunut tehokkaaksi metodologiaksi etenkin porttiskannauksia ja probeja vastaan, sen pääasiallinen haitta on suuri resurssien kulutus. Toisaalta on myös huomattava, että sumea logiikka on joissakin tapauksissa kiistanalainen metodivaihtoehto, koska huomattava osa tilastotieteilijöistä tunnustaa todennäköisyyslaskennan ainoana todellisena matemaattisena kuvauksena epävarmuudesta. (García-Teodoro ym., 2009.)

Sumean logiikan soveltamisessa IDS-järjestelmiin on Wun ja Banzhafin (2010) mukaan kaksi aktiivista tutkimussuuntausta: sumean logiikan käyttö järjestelmän ymmärrettävyyden parantamisessa sekä oppivia ja mukautuvia ominaisuuksia omaavien algoritmien tutkimus, jonka avulla pyritään pätevään, automaattiseen sumeiden sääntöjen generointiin. Wu ja Banzhaf (2010) väittävät menetelmien suosion kertovan niiden soveltuvuudesta kyseisiin ongelmiin.

4.2 Neuroverkot tunkeilijan havaitsemisessa

García-Teodoro ym. (2009) mainitsevat neuroverkkojen joustavuuden ja sopeutumiskyvyn olevan merkityksellisin vaikutustekijä niiden käyttöönottoon tunkeilijan havaitsemisen tutkimuksessa. Erilaisia neuroverkkotyyppejä, kuten eteenpäin syöttäviä ja takaisinkytkettyjä neuroverkkoja sekä itseorganisoituvia karttoja, on kattavasti sovellettu tunkeilijan havaitsemisen piiriin viimeisten kahdenkymmenen vuoden aikana (Wu & Banzhaf, 2010).

Wun ja Banzhafin (2010) mukaan eteenpäin syöttäviä neuroverkkoja on esiintynyt paljon normaalien ja poikkeavien toimintojen kaavojen mallinnuksessa niin käyttäjäprofiilien kuin koko järjestelmän kattavien profiilien luonnissa. Takaisinkytkettyjä neuroverkkoja on sovellettu tapahtumien ennustamiseen, jossa neuroverkko ennustaa tulevan tapahtuman aikaisempien tapahtumien perusteella. Jos odotettavissa olevan tapahtuman ja todellisen tapahtuman välillä on riittävä poikkeama, takaisinkytkettyä neuroverkkoa hyödyntävä IDS-järjestelmä hälyttää potentiaalisesta tunkeilijasta. (Wu & Banzhaf, 2010.)

Shenfield, Day ja Ayesh (2018) ovat suunnitelleet tutkimuksessaan eteenpäin syöttävään neuroverkkoon perustuvan luokittelumenetelmän haitallisten, mutta hyvin vaikeasti havaittavien shellcode-kaavojen tunnistamiseksi. Menetelmää sovellettiin offline-ympäristössä tutkimuskohtaiseen verkkoliikenteen datajoukkoon, joka sisälsi sekä vaarattomia että haitallisia tiedostoja. Tutkielmassa esitetyt tulokset viittaavat kyseisen luokittelumenetelmän olevan suorituskyvyltään erittäin pätevä shellcode-kaavojen havaitsemisessa, luokittelijan saavuttaessa alle kahden prosentin FP-luvun 400 000:n otoksen testijoukolla. (Shenfield ym., 2018.)

Shenfield ym. (2018) tutkielmassa on kuitenkin kaksi epäkohtaa. Ensimmäisenä, luokittelumenetelmän testauksessa käytetty datajoukko ei ole yleisesti hyväksytty, laajasti testattu datajoukko, mikä tekee datajoukosta epäluotettavan. Shenfield ym. (2018) täsmentävät datajoukon koostuvan yleisistä vaarattomista tiedostoista, kuten kuvatiedostoista, dynaamisista linkkikirjastotiedostoista ja musiikkitiedostoista, sekä haavoittuvuusvarasto *exploitdb*:stä peräisin olevista vaarallisista shellcode-tiedostoista. Shenfield ym. suunnittelema luokittelumenetelmä on siis testattava kontrolloidussa, yleisesti hyväksytyssä ympäristössä. Toisena, luokittelumenetelmä on testattu täysin

offline-ympäristössä, joka edelleen heikentää menetelmän testauksen luotettavuutta. On perusteltua suorittaa neuroverkon kouluttaminen offline-tilassa, sillä koulutusvaihe vie aikaa sekä resursseja (Kang & Kang, 2016), mutta tutkielman validiointi ja ehdotetun menetelmän testaus on syytä suorittaa online-ympäristössä.

Kang ja Kang (2016) ovat soveltaneet eteenpäin syöttävää neuroverkkoa autojen moottorinohjausyksiköiden välisten verkkoyhteyksien turvallisuusongelmiin. Ehdotetussa lähestymistavassa sovellettiin usean piilokerroksen neuroverkkoa sekä ohjaamatonta esikoulutusmenetelmää neuroverkon syötteiden alustamiseksi. Offline-tilassa suoritetulla ohjaamattomalla esikoulutusmenetelmällä pyrittiin ratkaisemaan syviä neuroverkkoja vaivaava katoavan gradientin ongelma *(engl. Vanishing gradient problem)*, jossa myöhempien piilokerrosten neuronien painokertoimet muuttuvat aikaisempia kerroksia nopeammin, tehden monikerroksisen lähestymistavan epäpäteväksi. Ehdotettua IDS-järjestelmää testattiin ja kokeellisilla tuloksilla osoitettiin esitetyn tekniikan korkea, noin 98 %:n DR. (Kang & Kang, 2016.)

Historiallisesti, itseorganisoituvia karttoja on käytetty kaikista neuroverkkotyypeistä laajimmin tunkeilijan havaitsemisen prosesseissa (Zanero & Savaresi, 2004; Wu & Banzhaf, 2010). Maciá-Pérez ym. (2011) mukaan tämä johtuu SOM-karttojen ohjaamattomaan oppimiseen, mikä tekee kyseisen teknologian toteuttamisesta yleisluonteisesti muita neuroverkkotyyppejä helpompaa ja nopeampaa. Omassa tutkimuksessaan Maciá-Pérez ym. toteuttivat verkkopohjaisen, poikkeuksiin perustuvan hajautetun IDS-järjestelmän, jonka analyysimoduulin havaitsemismoottori perustui itseorganisoituvaan karttaan. Tutkimuksen tulokset sekä toimivan prototyypin suoriutuminen osoittivat laitteen vakaan käyttäytymisen epäedullisissa tietoliikenneolosuhteissa. (Maciá-Pérez ym., 2011.)

4.3 Evoluutiolaskenta tunkeilijan havaitsemisessa

Evoluutiolaskentaa on sovellettu useisiin tunkeilijan havaitsemisen prosesseihin, kuten optimointiin, luokittelijan generointiin ja automaattiseen järjestelmän mallirakenteen suunnitteluun (Wu & Banzhaf, 2010), mutta eniten evoluutiolaskennan metodeja käytetään todennäköisesti sääntöjen generoinnissa (Abadeh, Mohamadi & Habibi, 2011). Li (2004) on lähestynyt tunnisteisiin perustuvan tunkeilijan havaitsemisen tutkimuksessaan geneettisiä algoritmeja eri suunnasta. Tutkimuksessaan Li käyttää geneettisiä algoritmeja hyökkäystunnisteiden muodostamiseen, etsimällä analysointityökalun avulla DARPA-datajoukosta hyökkäyskaavoja ja kytkemällä analyysin tulosteet geneettisen algoritmin syötteiksi. Geneettinen algoritmi tuottaa joukon päteviä tunnisteita IDS-järjestelmän tunnistetietokantaan. (Li, 2004.)

Hansen, Lowry, Meservy ja McDonald (2007) kehittivät ja testasivat tutkimuksessaan välittömästi käyttöönotettavia, geneettistä ohjelmointia hyödyntäviä tunnisteisiin perustuvia IDS-järjestelmiä. He kehittivät saatavilla olevia järjestelmiä soveltamaan konfiguraatioissaan geneettisen ohjelmoinnin algoritmia ja tutkivat muunneltujen järjestelmien suoriutumista KDD Cup 99 -datajoukolla. Tutkimuksen tulokset ovat lupaavia: järjestelmien havaitsemisnopeudet vaihtelivat 67 %:sta 100 %:iin tasaisin alhaisin FAR arvoin. (Hansen ym., 2007.)

Samoin Lu ja Traoré (2004) tutkivat geneettisen ohjelmoinnin soveltamista tunkeilijan havaitsemisen prosesseihin. Geneettisen ohjelmoinnin avulla luotiin lähestymistapa sääntöihin perustuvaan poikkeuksien havaitsemiseen, jossa järjestelmän sääntöjoukko tuotettiin geneettisen ohjelmoinnin avulla. Ehdotettua lähestymistavan suorituskykyä testattiin DARPA-tietokannan avulla, ja kokeiden tulokset osoittavat lähes 100 %:n DR:n sekä 1,4 % - 1,8 % FAR-luvun. Ehdotus koettiin potentiaaliseksi poikkeuksien havaitsemismetodologiaksi. (Lu & Traoré, 2004.)

Evoluutiolaskentaa käytetään paljon myös yhdistelmälähestymistavoissa. Yhdistelmämenetelmät yhdistävät useita havaitsemismetodeja pyrkien kollektiivisesti parempaan suorituskykyyn (Buczak & Guven, 2016). Zainal, Maarof ja Shamsuddin (2009) jalostivat tutkimuksessaan lineaarista geneettistä ohjelmointia, sumeaa logiikkaa soveltavaa eteenpäin syöttävää neuroverkkoa eli ANFIS-menetelmää *(engl. Adaptive neural fuzzy inference system)* sekä satunnaisen metsän *(engl. Random forest)* algoritmia. Lähestymistapojen hybridisaation tarkoituksena on parantaa tunkeilijan havaitsemisjärjestelmän luokittelukykyä. Tutkimus osoitti eri oppimisparadigmojen kokoelman olevan kykeneväinen parantamaan havaitsemisen tarkkuutta. Ehdotettu yhdistelmäluokittelija toimi 99 %:n havaitsemistarkkuudella, minimaalisin väärin positiivisin arvoin. (Zainal ym., 2009.) Tutkimuksessa ei kuitenkaan ole eksplisiittisesti mainittu yhdistelmäluokittelijan DR-arvoa tai FAR-lukua, mikä vaikeuttaa tulosten arviointia.

4.4 Yhdistelmämetodologiat

Tunkeilijan havaitsemista voidaan lähestyä myös yhdistelemällä samaan järjestelmään eri metodeja hyödyntäviä itsenäisiä moduuleja ja tämän modulaarisen rakenteen odotetaan helpottavan tulevaisuuden järjestelmien laajentumista (Bridges & Vaughn, 2000). Toosi & Kahani (2007) ovat lähestyneet tunkeilijan havaitsemista tarjoamalla viitekehyksen yhdistelmäluokittelijan rakentamiseen sekä soveltamiseen. Tutkimuksessa toteutettiin ANFIS-menetelmää, sumeaa päättelymoottoria sekä geneettistä algoritmia yhdistelevä tunkeilijan havaitsemisjärjestelmä. Ehdotetun järjestelmän ensimmäisellä kerroksella ANFIS-verkko prosessoi järjestelmän syötteet, poimien syötedatasta havaitut hyökkäykset. ANFIS-verkon tulosteet syötetään

järjestelmäarkkitehtuurin toisella kerroksella sumeaan päättelymoottoriin, joka tekee lopullisen päätöksen havainnosta. Geneettistä algoritmia käytetään ehdotetussa järjestelmässä päättelymoottorin aktiiviseen optimointiin. Menetelmää koulutettiin ja testattiin KDD Cup 99 -datajoukon osajoukolla, ja tutkimuksen tulokset olivat rohkaisevia. Yhdistelmäluokittelijaa saavutti testituloksissa 95,3 %:n havaitsemisnopeuden 1,9 %:n FAR-arvolla. (Toosi & Kahani, 2007.)

Bridges ja Vaughn (2000) ovat kehittäneet tutkimuksessaan prototyypin hybridi-IDS-järjestelmästä, joka lähestyy poikkeuksien havaitsemista soveltaen sumeaa logiikkaa ja tunnisteiden havaitsemista käyttäen perinteisiä, sääntöihin perustuvia asiantuntijajärjestelmien tekniikoita. Sääntöihin perustuvat asiantuntijajärjestelmät ovat perinteisesti johtaneet sääntönsä työlään ja virhealttiin konsultointiprosessin kautta. Bridges ja Vaughn pyrkivät lievittämään kyseisten asiantuntijajärjestelmien tietämyksen pullonkaulaongelman vaikutusta tunnisteisiin perustuvan tunkeilijan havaitsemisen suorituskykyyn soveltamalla tunnisteisiin perustuvan havaitsemisen moduuliin sumeita sääntöjä. Geneettisiä algoritmeja käytettiin sumeiden joukkojen jäsennyysfunktioiden optimoimiseksi. (Bridges & Vaughn, 2000.) Bridgesin ja Vaughnin suunnittelema hybridijärjestelmäarkkitehtuuri on laaja-alainen, kunnianhimoinen, mutta teoreettiselle tasolle jäänyt toteuttamaton lähestymistapa tunkeilijan havaitsemiseen, eikä sen suoriutumista tunkeilijan havaitsemisen prosesseista ole validoitu testauksella.

Abadeh, Habibi ja Lucas (2007) tutkivat sumeaan genetiikkaan pohjautuvaa oppimismetodia tunkeilijan havaitsemisessa. Sumealla genetiikalla viitataan tässä tapauksessa geneettisten algoritmien hyödyntämiseen sumean sääntöpohjaisen järjestelmän kehitysprosessissa. Tutkielmassa ehdotettu järjestelmä on sumea sääntöpohjainen luokittelujärjestelmä, joka johtaa sääntötietokantansa geneettisen algoritmin mutaatioprosessista. Tällä tavoin pyritään maksimoimaan sääntöpohjan suoriutuminen poikkeuksien havaitsemisessa. Geneettisen algoritmin suoriutumista sääntöpohjan generoinnissa tutkittiin käyttäen kahta erillistä kelpoisuusfunktiota, SRPP:tä ja NCP:tä. Kelpoisuusfunktioiden tuottamien sääntöpohjien suoriutumista poikkeuksiin perustuvasta tunkeilijan havaitsemisesta vertailtiin ja vertailun tuloksena SRPP-metodin tuottaman sääntöpohjan kokonaissuorituskyky on NCP-metodia korkeampi. (Abadeh, Habibi & Lucas, 2007.)

4.5 Problematiikka

Sommer ja Paxson (2010) ovat artikkelissaan perehtyneet kattavasti koneoppimismetodien ja tunkeilijan havaitsemisen problematiikkaan. Perusväittämänään tutkijat luonnehtivat tietojenkäsittelytieteen muihin soveltamisaloihin nähden tunkeilijan havaitsemisen aihealueen hyvin erilaiseksi ympäristöksi, jossa koneoppimismetodien tehokas soveltaminen ei ole läheskään yhtä suoraviivaista. Tutkijat huomauttavat, että

fundamentaalinen ongelma ilmenee jo poikkeuksiin perustuvan tunkeilijan havaitsemisen teoretisoinnissa. Poikkeuksiin perustuvassa tunkeilijan havaitsemisessa ensisijainen tavoite on tunnistaa tietoliikenteestä normaalista poikkeavaa toimintaa ja tällä tavoin havaita tietoturvarikkeet. Sommer ja Paxson (2010) muistuttavat koneoppimistyökalujen vahvuuden piilevän yhtäläisyyksien tunnistamisessa joukkoon kuulumattomien aktiviteettien tunnistamisen sijaan. Väite on todenmukainen, sillä koneoppimisen fundamenttina luokittelujärjestelmät on koulutettava jokaisen luokittelukategorian mukaan ja kunkin luokan koulutusjoukossa olevien edustajien määrän tulee olla suuri. Poikkeuksien havaitsemisessa pyritään kuitenkin löytämään tietoliikenteestä uusia hyökkäysmalleja, jotka eivät välttämättä noudata tunnettua toimintakaavaa. Näin ollen, koneoppimisalgoritmia voidaan kouluttaa ainoastaan normaalin toiminnan sekä tunnettujen hyökkäysmallien perusteella, mikä ei puolestaan kouluta algoritmia poikkeuksien havaitsemiseen. (Sommer & Paxson, 2010.) Vaikka Sommer ja Paxson käsittelevät artikkelissaan koneoppimismetodeja, joihin lukeutuu myös pehmolaskennan ulkopuolelle sijoittuvia kovan tietojenkäsittelytieteen metodeja, heidän esille tuomansa huomautukset pätevät myös pehmolaskennan soveltamiseen tunkeilijan havaitsemisessa.

5 YHTEENVETO JA POHDINTA

Tässä tutkielmassa tarkasteltiin pehmolaskentametodien käyttöä tunkeilijan havaitsemisjärjestelmissä. Tutkielman tavoitteena oli selvittää, miten yleisimmät pehmolaskentametodit, sumea logiikka, neuroverkot sekä evoluutiolaskenta, sopeutuvat alati muuttuviin tunkeilijan havaitsemisen olosuhteisiin ja miten kyseiset metodit edistävät tunkeilijan havaitsemista. Pehmolaskennan kattava soveltaminen IDS-järjestelmissä voi olla seuraava kriittinen kehitysaskel tietoturvakontekstissa.

Tutkielma taustoitettiin kirjallisuuskatsauksen ensimmäisessä luvussa, jossa valaistiin myös tutkielman tarkoitusta. Nykyisissä kaupallisissa IDS-järjestelmissä on puutteita (Patcha & Park, 2007), ja tietoturvakokonaisuuden puutteellisuus avaa oven erittäin hintaville tietoturvarikkeille. Tutkimuksen avulla pyrittiin kokoamaan yhtenäinen kuvaus pehmolaskentametodien myötävaikutuksista nykyisten IDS-järjestelmien toiminnan parantamisessa. Toisessa luvussa määriteltiin pehmolaskennan käsitteistö suhteessa laskennalliseen älykkyyteen, tekoälyyn ja koneälyyn, sekä tutustuttiin kolmeen yleisimpään pehmolaskentametodiin: sumeaan logiikkaan, neuroverkkoihin sekä evoluutiolaskentaan. Kolmannessa luvussa tutustuttiin tunkeilijan havaitsemisen prosessiin, tunkeilijan havaitsemisjärjestelmiin sekä niiden ongelmakohtiin. Neljännessä luvussa tutkittiin tieteelliseen kirjallisuuteen perustuen pehmolaskentametodien soveltuvuutta tunkeilijan havaitsemiseen.

Seuraavaksi tarkastellaan edellisissä luvuissa käsiteltyjä aiheita tutkimusongelman näkökulmasta ja pohditaan vastausta tutkimuskysymyksiin: Miten pehmolaskentametodit parantavat tunkeilijan havaitsemista? Minkälaisia ongelmakohtia pehmolaskentametodien soveltaminen tunkeilijan havaitsemisen ympäristössä esittää? Pehmolaskentametodien, tässä tapauksessa sumean logiikan, neuroverkkojen ja evoluutiolaskennan, hyödyntämisen tunkeilijan havaitsemisessa on tutkimuskirjallisuudessa osoitettu tehostavan tunkeilijan havaitsemisen suoritusta. Kirjallisuudessa on paneuduttu etenkin poikkeuksiin perustuvan tunkeilijan havaitsemisen tutkimukseen ja kokeellisilla tutkimuksilla on todistettu edellä mainittujen pehmolaskentametodien korostavan poikkeuksiin perustuvan

havaitsemisprosessin suorituskykyä. Suorituskyvyn optimoituminen on selitettävissä pehmolaskentametodien sopeutuvaisuudella, virheensietokyvyllä sekä suurilla laskennallisilla nopeuksilla (Wu & Banzhaf, 2010).

Älykästä käytöstä simuloivien pehmolaskentamenetelmien soveltuvuuden tutkimisen tunkeilijan havaitsemiseen voidaan myös nähdä kytkeytyvän yleiseen koneälytutkimukseen. Näillä älykkäillä metodeilla voi olla potentiaalia tehostaa nykyisiä yleisiä laskennallisia prosesseja ja tällä voi olla kauaskantoisia vaikutuksia tietojenkäsittelytieteen kehitykseen sekä epäsuorasti yleiseen yhteiskuntakehitykseen. Täten laskennallista älykkyyttä noudattavien pehmolaskentametodien soveltaminen tunkeilijan havaitsemiseen on ymmärrettävä tutkimusyhteisön kiinnostuksen kohde.

Tässä tutkielmassa tarkastellut kokeelliset tutkimukset osoittavat, että pehmolaskentametodeja soveltamalla voidaan saavuttaa lupaavia tuloksia havaitsemistehokkuudessa sekä vahvistaa uusia soveltamisympäristöjä relevanteiksi tutkimusaiheiksi. Esimerkiksi Shenfieldin, Dayn ja Ayeshin (2018) tutkimus neuroverkkoluokittelijan soveltamisesta shellcode-kaavojen havaitsemiseen osoittaa yllättävää menestystä, ottaen huomioon shellcode-kaavojen tunnetun transformatiivisuuden. Kangin ja Kangin (2016) tutkimus neuroverkkojen soveltamisesta ajoneuvojen moottorinohjausyksikköihin niiden tietoturvan parantamiseksi on ensimmäinen laatuaan ja erittäin tärkeä tutkimusaihe. Näiden esimerkkien tarkoituksena on osoittaa, että pehmolaskentamenetelmien soveltaminen tunkeutumisen havaitsemiseen avaa uusia ovia tietojenkäsittelytieteen yleiselle kehitykselle.

Tässä kirjallisuuskatsauksessa ilmeni, että pehmolaskentametodien soveltamisessa tunkeilijan havaitsemiseen on kiistattomia hyötyjä, mutta myös haittoja. Tekniikoita arvioidaan niiden havaitsemisprosessin tehokkuuden sekä kustannustehokkuuden näkökulmasta (García-Teodoro ym., 2009). Pehmolaskentametodien on yleisesti osoitettu kasvattavan havaitsemisprosessin tehokkuutta, mutta niiden soveltaminen ei ole yleisesti onnistunut parantamaan poikkeuksien havaitsemisen prosessin heikkoa kustannustehokkuutta. Poikkeuksiin perustuvien havaitsemisjärjestelmien kehno kustannustehokkuus sekä kaupallisten poikkeuksien havaitsemisjärjestelmien silmiinpistävä läsnäolon puute validioivat Sommerin ja Paxsonin (2010) väitteen, että poikkeuksiin perustuvan tunkeilijan havaitsemisen ympäristö on fundamentaalisesti erilainen kuin monet muut tietojenkäsittelytieteen sovellusalueista, tehden koneoppimismetodien soveltamisen kyseiseen aihealueeseen hankalaksi huolimatta koneoppimisen ansiokkaasta soveltamisesta muissa tutkimusympäristöissä. Tämä hankaluus selittää kuitenkin myös tutkimusaiheen houkuttelevuutta.

On myös syytä huomioida Sommerin ja Paxsonin (2010) kriittinen kommentti koskien joitain tässä kirjallisuuskatsauksessa käsiteltyjä kokeellisia tutkimuksia. Sommer ja Paxson (2010) huomauttavat tunnettujen ja usein käytettyjen testidatajoukkojen, nimenomaan DARPA ja KDD Cup 99 -tietokantojen sisältävän ikääntynyttä dataa, mikä ei enää luotettavasti kuvaa nykyhetken tietoliikennettä. Tästä johtuen kokeelliset tutkimukset, jotka

pohjustavat kokeensa kyseisiin datajoukkoihin, eivät välttämättä toimi kokeiden tulosten mukaisesti sovellettuna aktiiviseen, nykyhetken tietoliikenteeseen. Tämä on tärkeä tekijä määriteltäessä tässä tutkielmassa käsiteltyjen tutkimusten luotettavuutta.

Pehmolaskenta ja pehmolaskentametodit nähdään potentiaalisena tutkimussuuntauksena tunkeilijan havaitsemisen alalla. Metodien soveltaminen IDS-tutkimukseen on perusteltavissa niiden tarjoamalla suoritustehokkuudella, mutta erityistä huomiota on kiinnitettävä yleiseen järjestelmäsuunnitteluun, jotta tunkeilijan havaitsemisjärjestelmien yleinen heikon kustannustehokkuuden ongelma vältetään. Todetusta epätasapainosta eri tunkeilijan havaitsemismetodologioiden toteuttamisessa kaupallisiksi IDS-järjestelmiksi voidaan johtaa mielenkiintoinen jatkotutkimusaihe: poikkeuksiin perustuvaa tunkeilijan havaitsemista ympäröivä kiinnostus voi olla perusteltua suunnata seuraavaksi poikkeuksiin perustuvien IDS-järjestelmien pätevän toteuttamisen tutkimukseen.

LÄHTEET

Abadeh, M. S., Mohamadi, H. & Habibi, J. (2011). Design and analysis of genetic fuzzy systems for intrusion detection in computer networks. *Expert Systems with Applications, 38,* 7067–7075.

Abadeh, M. S., Habibi, J. & Lucas, C. (2007). Intrusion detection using a fuzzy genetics-based learning algorithm. *Journal of Network and Computer Applications, 30*, 414–428.

Abraham, A., Grosan, C. & Chen, Y. (2005). Cyber Security And The Evolution Of Intrusion Detection Systems. *I-manager's Journal on Future Engineering and Technology, 1(1),* 74-82. doi: 10.26634/jfet.1.1.968

Adaptation. (2011). *Oxford English online dictionary* (3. painos). Haettu 29.11.2018 osoitteesta http://www.oed.com.ezproxy.jyu.fi/view/Entry/2115?redirectedFrom=adaptation#eid

Ahmed, M., Mahmood, A. N. & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications 60,* 19-31.

Axelsson, S. (1999). *Research in Intrusion-Detection Systems: A Survey* (Technical report No. 98-17). Göteborg, Sweden: Department of Computer Engineering, Chalmers University of Technology.

Bezdek, J. (1994). What is computational intelligence? Teoksessa J. Zurada, R. Marks & C. Robinson (toim.), *Computational Intelligence: Imitating Life* (1-11). Piscataway, NJ: IEEE Press.

Bridges, S. M. & Vaughn, R. B. (2000). Fuzzy data minig and genetic algorithms applied to intrusion detection. Teoksessa *Proceedings of the National Information Systems Security Conference (13-31).*

Buczak, A. & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys & Tutorials, 18(2),* 1153-1176.

Cheng, T.-H., Lin, Y.-D., Lai, Y.-C. & Lin, P.-C. (2012). Evasion Techniques: Sneaking through Your Intrusion Detection/Prevention Systems. *IEEE Communications Surveys and Tutorials, 14(4),* 1011-1020.

Durst, R., Champion, T., Witten, B., Miller, E. & Spagnuolo, L. (1999). Testing and evaluating computer intrusion detection systems. *Communications of the ACM, 42(7),* 53-61.

Dyson, G. B. (1997). *Darwin among the Machines: The Evolution of Global Intelligence.* Reading, Massachusetts: Perseus Books.

Eberhart, R. C. & Shi, Y. (2007). *Computational Intelligence : Concepts to Implementations* [e-kirja]. Haettu 15.11.2018 osoitteesta https://ebookcentral.proquest.com

Fogel, D. (1995). Review of Computational Intelligence: Imitating Life. *IEEE Transactions on Neural Networks, 6(6),* 1562-1265.

García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G. & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers and Security 28*, 18-28.

Garnham, A. (2017). *Artificial Intelligence: An Introduction* [e-kirja]. Haettu 14.11.2018 osoitteesta https://ebookcentral.proquest.com

Hansen, J. V., Lowry, P. B., Meservy, R. D. & McDonald, D. M. (2007). Genetic programming for prevention of cyberterrorism through dynamic and evolving intrusion detection. *Decision Support Systems, 43,* 1362–1374.

Hawkins, D. M. (1980). *Identification of Outliers* (1. painos). Hollanti: Springer

Ibrahim, D. (2016). An overview of soft computing. *Procedia Computer Science, 102,* 34-38. doi:10.1016/j.procs.2016.09.366

Jain, L. C., Quteishat, A. & Lim, C. P. (2007) Intelligent Machines: An Introduction. Teoksessa J. S. Chahl, L. C. Jain, A. Mizutani & M. Sato-Ilic (toim.), *Innovations in Intelligent Machines – 1 (1-9)*. Berliini: Springer.

Kang, M.-J. & Kang, J.-W. (2016). Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security. *PLoS ONE, 11(6).* doi: 10.1371/journal.pone.0155781

Kohonen, T. (1990). The Self-Organizing Map. *Proceedings of the IEEE, 78(9),* 1464-1480.

Krishnakumar, K. (2003). *Intelligent Systems for Aerospace Engineering – An Overview* (NASA Technical Report, dokumentti 20030105746). Haettu 14.11.2018 osoitteesta https://ti.arc.nasa.gov/m/pub-archive/364h/0364%20(Krishna).pdf

Kulkarni, R. V., Förster, A. & Venayagamoorthy, G. K. (2011). Computational Intelligence in Wireless Sensor Networks: A Survey. *IEEE Communications Surveys & Tutorials, 13(1),* 68-96.

Kumar, S. & Spafford, E. H. (1994). *An Application of Pattern Matching in Intrusion Detection* (Report No. 94-013). West Lafayette, IN: Purdue University

Langin, C. & Rahimi, S. (2010). Soft computing in intrusion detection: the state of the art. *Journal of Ambient Intelligence and Humanized Computing, 1(2),* 133-145.

Lazarevic, A., Kumar, V. & Srivastava, J. (2005) Intrusion detection: a survey. Teoksessa Lazarevic, A., Kumar, V. & Srivastava, J (toim.), *Managing cyber threats: issues, approaches, and challenges (luku 2)*. Springer Verlag.

Liao, H.-J., Lin, C.-H. R., Lin, Y.-C. & Tung, K.-Y. (2012). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications 36,* 16-24.

Lu, W., & Traoré, I. (2004). Detecting New Forms of Network Intrusion Using Genetic Programming. *Computational Intelligence, 20,* 475-494.

Maciá-Pérez, F., Mora-Gimeno, F. J., Marcos-Jorquera, D., Gil-Martínez-Abarca, J. A., Ramos-Morillo, H. & Lorenzo-Fonseca, I. (2011). Network intrusion detection system embedded on a smart sensor. *IEEE Transactions on Industrial Electronics, 58(3),* 722-732.

Mitra, S. & Hayashi, Y. (2000). Neuro–Fuzzy Rule Generation: Survey in Soft Computing Framework. *IEEE Transactions on Neural Networks, 11(3),* 748-768.

Mitra, S., Pal, S. K. & Mitra, P. (2002). Data Mining in Soft Computing Framework: A Survey. *IEEE Transactions on Neural Networks, 13(1),* 3-14.

Mudzingwa, D. & Agrawal, R. (2012). A study of Methodologies used in Intrusion Detection and Prevention Systems (IDPS). Teoksessa *2012 Proceedings of IEEE Southeastcon* (1-6). Orlando, Florida, USA, March 15-18, 2012.

Niskanen, V. A. (2003). *Sumea logiikka: Kirkasta älyä ja mallinnusta*. Helsinki: WSOY.

Ovaska, S. J., Kamiya, A. & Chen, Y. (2006). Fusion of Soft Computing and Hard Computing: Computational Structures and Characteristic Features. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, 36(3),* 439-448.

Pang, R., Yegneswaran, V., Barford, P., Paxson, V. & Peterson, L. (2004). Characteristics of internet background radiation. Teoksessa *Proceedings of ACM IMC (27-40),* Taormina, Sicily, Italy, October 25–27, 2004.

Patcha, A. & Park, J-M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks, 51,* 3448-3470.

Pfleeger, C. P., Pfleeger, S. L. & Margulies, J. (2015). *Security in Computing* (5. painos). Upper Saddle River, New Jersey: Prentice Hall.

Rudas, I. J. & Fodor, F. (2008). Intelligent Systems. *International Journal of Computers, Communications and Control, 3(SPL.ISS),* 132-138.

Scarfone, K. & Mell, P. (2007). NIST: Guide to Intrusion Detection and Prevention Systems (IDPS). Special Publication (NIST SP) - 800-94.

Schepers, F. (1998). Network- versus host-based intrusion detection. *Information Security Technical Report, 3(4),* 32-42.

Shenfield, A., Day, D. & Ayesh, A. (2018). Intelligent intrusion detection system using artificial neural networks. *ICT Express, 4,* 95-99.

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

Toosi, A. N. & Kahani, M. (2007). A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers. *Computer Communications, 30,* 2201-2212.

Venayagamoorthy, G. K. (2009) A Successful Interdisciplinary Course on Computational Intelligence. *IEEE Computational Intelligence Magazine, 4(1),* 14-23. doi:10.1109/MCI.2008.930983

Wu, S. X. & Banzhaf, W. (2010). The use of computational intelligence in intrusion detection systems: A review. *Applied Soft Computing, 10,* 1-35.

Zadeh, L. A. (1962). From Circuit Theory to System Theory. *Proceedings of the IRE, 50(5),* 856-865. doi:10.1109/JRPROC.1962.288302

Zadeh, L. (1994a). Fuzzy Logic, Neural Networks, and Soft Computing. *Communications of the ACM, 37(3),* 77-84. doi: 10.1145/175247.175255

Zadeh, L. (1994b). Soft Computing and Fuzzy Logic. *IEEE Software, 11(6)*, 48-56. doi: 10.1109/52.329401

Zadeh, L. A. (1994c). Fuzzy Logic and Soft Computing: Issues, Contentions and Perspectives. Teoksessa *Proceedings of the Third International Conference on Fuzzy Logic, Neural Nets and Soft Computing, IIZUKA'94,* (1-2). Iizuka, Fukuoka, Japan, August 1-7, 1994.

Zadeh, L. A. (1999). From computing with numbers to computing with words - from manipulation of measurements to manipulation of perceptions. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 46(1),* 105-119.

Zainal, A., Maarof, M. A. & Shamsuddin, S. M. (2009). *Journal of Information Assurance and Security, 4,* 217-225.