Juuso Kapulainen

Vertailumalli tietojoukoille tunkeutumisen havaitsemisjärjestelmissä

Tietotekniikan pro gradu -tutkielma

9. tammikuuta 2023

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Juuso Kapulainen

Yhteystiedot: juuso.kapulainen@gmail.com

Ohjaajat: Timo Hämäläinen

Työn nimi: Vertailumalli tietojoukoille tunkeutumisen havaitsemisjärjestelmissä

Title in English: Comparison model for datasets in intrusion detection systems

Työ: Pro gradu -tutkielma

Opintosuunta: Ohjelmisto- ja tietoliikennetekniikka

Sivumäärä: 58+12

Tiivistelmä: Tietoverkkojen ja verkossa olevien järjestelmien jatkuva kasvu on nostanut tietoverkkoturvallisuuden merkityksen ennennäkemättömän tärkeään asemaan. Anomaliapohjaiset tunkeutumisen havaitsemisjärjestelmät pyrkivät havainnoimaan verkkoliikenteen anomalioita, eli epänormaalia ja näin puolustamaan järjestelmiä haitalliselta liikenteeltä. Näiden järjestelmien koulutukseen ja arviointiin tarvitaan tietojoukkoja, jotka koostuvat tietoliikenne informaatiosta. Jotta järjestelmistä voidaan tehdä mahdollisimman luotettavat ja tehokkaat, tulee niiden käyttöön valita parhaat mahdolliset tietojoukot. Tässä tutkimuksessa luodaan konstruktiivisen tutkimusmetodin avulla vertailumalli, joiden avulla tietojoukkoja voidaan vertailla keskenään. Mallin toimivuus todistetaan soveltamalla sitä joukkoon tunnetuimpia tietojoukkoja. Vertailumallilla saatiin selkeästi eroteltua tietojoukkojen laatu ja niiden keskinäiset erot eri laadun kriteereillä. Vertailusta kävi ilmi, että etenkin uudet tietojoukot ovat suurimmaksi osin laadukkaampia kuin vanhat ja CSE-CIC-IDS2018 tietojoukko menestyi testijoukosta parhaiten.

Avainsanat: Tunkeutumisen havaitseminen, IDS, tietojoukot, kyberturvallisuus, tietoturva, tietoverkkoturvallisuus

Abstract: The continuous growth of information networks and online systems has raised the importance of information network security to an unprecedentedly important position. Anomaly-based intrusion detection systems aim to detect network traffic anomalies, i.e.,

abnormal traffic, and thus defend systems against harmful traffic. For the training and evaluation of these systems, datasets consisting of telecommunication information are needed. In order to make the systems as reliable and efficient as possible, the best possible data sets must be selected for their use. In this study, a comparison model is created with the help of a constructive research method, with the help of which data sets can be compared with each other. The functionality of the model is proven by applying it to the most well-known data sets. With the comparison model, it was possible to clearly distinguish the quality of the data sets and their mutual differences with different quality criteria. The comparison showed that especially the new data sets are mostly of higher quality than the old ones and the CSE-CIC-IDS2018 data set performed best among the test set.

Keywords: Intrusion detection, IDS, data sets, cyber security, information security, information network security

Sisältö

1 Johdanto

Nykypäivän tietoverkkojen ja verkossa toimivien sovellusten alati kasvava käyttö yhteiskunnan kaikilla osa-alueilla on korostanut tietoverkkoturvallisuuden merkityksen ennennäkemättömän tärkeään asemaan. Tietoverkkojen kasvu tapahtuu alati kiihtyvällä vauhdilla ja yhä useampi toimija on jollain tasolla yhteydessä globaaliin tietoverkkoon. Ciscon vuosittaisen raportin (2020) mukaan 66 %:lla maailman väestöstä on Internet-yhteys vuoteen 2023 mennessä. Mahdollisten hyökkäyskohteiden jatkuvasti lisääntyessä, myös verkossa tapahtuvien hyökkäysten ja tunkeutumisyrityksien määrä kasvaa, jonka takia yksityishenkilöiden ja yritysten verkkotietoturvaan liittyvät riskit ovat luonteeltaan jatkuvasti kasvussa (Hnamte ja Hussain 2021). Verkon altistuminen haitalliselle toimijalle voi johtaa mahdolliseen tärkeän tiedon menettämiseen, tietomurtoihin ja käyttäjien luottamuksen menettämiseen (Verma, Bhandari, ja Singh 2020). Entuudestaan tunnettujen verkkohyökkäysten lisääntymisen lisäksi tietoturvauhat, kuten nollapäivähyökkäykset, jotka on suunniteltu kohdistumaan Internetin käyttäjiin, ovat lisääntyneet (Khraisat ym. 2019). Koska kaikki yhteiskunnan osa-alueet, yksityishenkilöt mukaan lukien, ovat enenevissä määrin riippuvaisia tietoverkkojen kautta yhdistetyistä laitteista, on kysyntä tietoverkkoihin liittyviin uhkiin reagoimiselle ja sen tutkimukselle kasvanut.

Tietoverkkoon liittyviin tietoturvauhkiin on jo pitkään pyritty vastaamaan erilaisilla tunkeutumisen havaitsemisjärjestelmillä (*engl. intrusion detection system, IDS*), joiden tehtävänä on tarkkailla annetun tietoverkon läpi virtaavaa verkkoliikennettä epäilyttävän tai epätavallisen, mahdollisesti haitallisen, liikenteen havaitsemiseksi. Tunkeutumisen havaitseminen on prosessi, jossa seurataan tietokonejärjestelmässä tai verkossa tapahtuvia tapahtumia ja analysoidaan niitä tunkeutumisen merkkien varalta (Liao ym. 2013). Tunkeutumisen havaitsemisjärjestelmä automatisoi tämän prosessin. Arqane ym. (2021) mukaan nykypäivänä tietoturva-asiantuntijat korostavat tunkeutumisen havaitsemisjärjestelmän merkitystä järjestelmien puolustuskyvyn vahvistamisessa varoittamalla epäilyttävistä toiminnoista ja haitallisista hyökkäyksistä. Verkkoon strategisesti asetetut tunkeutumisen havainnointijärjestelmät pyrkivät suojaamaan tietoverkkoa sekä ulkoisilta tunkeilijoilta että sisäverkossa tapahtuvilta mahdollisesti haitallisilta verkkotapahtumilta. Historiassa perinteisempi metodi

hyökkäysten ja tunkeutumisten havaitsemiseen ovat olleet palomuurit, mutta ne ovat luonteeltaan staattisia, eli ne eivät pysty mukautumaan tai sopeutumaan muuttuviin verkkohyökkäyksiin Palomuureissa on yksinkertaiset säännöt protokollien sallimiseksi tai kieltämiseksi, kun taas tunkeutumisen havainnointijärjestelmää on mahdollista käyttää monimutkaisempien hyökkäysten käsittelyyn ja se on myös luonteeltaan dynaaminen (Singh ja Singh 2014).

Yleisesti tunkeutumisen havaitsemisjärjestelmät voidaan laajasti kategorisoida kahteen eri kategoriaan riippuen siitä, mihin järjestelmä on asennettu ja mitä sillä valvotaan. Verkkopohjainen tunkeutumisen havainnointijärjestelmä (*engl. network based intrusion detection system, NIDS*) tarkkailee ja analysoi verkkoliikennettä, kun taas isäntäpohjainen tunkeutumisen havaitsemisjärjestelmä (*engl. host based intrusion detection system, HIDS*) tarkkailee ja analysoi yksittäiseen laitteeseen (*host*) liittyviä tietoja (Sulaiman ym. 2021). Perinteisesti verkkopohjaiset tunkeutumisen havainnointijärjestelmät luokitellaan laajasti niiden käyttämän havaitsemistyylin perusteella: tunnistepohjaiset järjestelmät (*engl. signature based IDS*) valvovat toimintaa ja havaitsevat tunkeutumisyrityksiä tarkasti tunnetuista haitallisen käyttäytymisen malleista johdettujen tunnisteiden perusteella, kun taas anomaliapohjaisilla havaitsemisjärjestelmillä (*engl. anomaly based IDS*) on käsitys normaalista toiminnasta ja se havaitsee poikkeamat kyseisestä normaalin liikenteen profiilista (Sommer ja Paxson 2010). Anomaliapohjaiset järjestelmät pyrkivät havainnoimaan anomalioita, eli normaalista poikkeavaa liikennettä ottamatta kantaan siihen, mikä epänormaalin liikenteen aiheuttaja tai syy on. Koneoppimisalgoritmeja käytetään laajasti kyberturvallisuudessa tunkeutumisen anomaliapohjaiseen havaitsemiseen, ja niiden on todistettu tarjoavan korkeat tunkeutumisen havaitsemisen tasot (Lakshminarayana, Philips ja Tabrizi 2019).

Ottaen huomioon anomaliapohjaisen verkkotunkeutumisen havaitsemisjärjestelmien lupaavat ominaisuudet, kuten paremman tarkkuuden ja nopeamman tunnistusnopeuden saavuttaminen (Wang ym. 2018), tämä lähestymistapa on tällä hetkellä tunkeutumisen havaitsemisen alan tutkimuksen ja kehityksen pääpaino (Nassif ym. 2021). Uusien ja kehittyviin tarpeisiin vastaavien tunkeutumisen havaitsemisjärjestelmien kehittämiseksi ja rakentamiseksi on tärkeää saada perusteellinen tieto olemassa olevien järjestelmien vahvuuksista ja puutteista (Hnamte ja Hussain 2021). Tehokkaan anomaliapohjaisen tunkeutumisen havaitsemisjärjestelmän kehittämisessä tarvitaan paljon tietoliikennedataa järjestelmän koulutus- ja

testausprosesseihin. Tietoliikennedataa sisältävät tietojoukot mahdollistavat järjestelmän oppimaan normaaleja liikenne- sekä hyökkäysmalleja, joiden avulla järjestelmä pystyy luokittelemaan sille annetut syöttötiedot, eli tietoliikenteen, oikeisiin kategorioihin. Järjestelmän koulutukseen ja testaukseen käytettävä tietojoukko rakennetaan normaalista verkkoliikenteestä ja poikkeavasta verkkoliikenteestä, joka auttaa luokittelijaa tunnistamaan tunkeutumisyritykset syöttötietojen perusteella. (Thakkar ja Lohiya 2020.) Tämän informaation sisällöllinen laatu on luokittelun oppimisen kannalta erittäin kriittistä ja vaikuttaa ensisijaisesti tunkeutumisen havaitsemisjärjestelmämallin saavuttamiin tuloksiin (Al-Daweri ym. 2020). IDS järjestelmien kouluttamisessa ja evaluoimisessa käytettävät tietojoukot ovat kaikkien koneoppimistekniikoiden ja niiden päälle rakentuvien AIDS järjestelmien ytimessä. Thakkar ja Lohiya (2020) kuvailevat tietojoukkojen sisällön koostuvan normaalista verkkoliikenteestä ja poikkeavasta verkkoliikenteestä, joka auttaa tunkeutumisen havaitsemisjärjestelmän tunnistamaan verkkoliikennedatan mallin riittävällä määrällä esimerkkejä. Kerätyt tiedot on jaettu harjoitustietojoukoksi ja testaustietojoukoksi informaationluokitteluominaisuuden kouluttamista ja testausta varten.

On olemassa useita tietoliikennedataa sisältäviä tietojoukkoja, joita tutkijat ovat käyttäneet arvioidakseen tunkeutumisen havainnointi- ja ehkäisymenetelmiensä tehokkuutta, mutta itse varsinaisten tietojoukkojen arviointiin ja arviointiin ei ole keskittynyt riittävästi tutkimusta (Gharib ym. 2016). Nassifin ym. (2021) mukaan aiempi tutkimus on keskittynyt pääasiassa erilaisten koneoppimismenetelmien testaamiseen ja implementointiin hyödyntäen usein vain yhtä tai kahta tietojoukkoa kerrallaan. Näissä tutkimuksissa on esitelty, usein hyvin lyhyesti, eri tietojoukkoja ja niiden ominaisuuksia sekä soveltuvuutta käytettävään menetelmään. Tietojoukkojen sisällölliset ominaisuudet saattavat erota toisistaan merkittävästikin, ja tietojoukon sisällöllä on suora vaikutus tunkeutumisen havaitsemisjärjestelmän toimintamalleihin ja performanssiin, joten tietojoukkojen ominaisuuksien ja niiden vaikutusten tarkka tunteminen ja arviointi on tärkeää tunkeutumisen havaitsemisjärjestelmien kehittämisen kannalta. Jotta eri tunkeutumisen havaitsemistietojoukkoja voitaisiin vertailla rinnakkain ja auttaa tutkijoita löytämään sopivia tietojoukkoja omiin spesifeihin arviointiskenaarioihinsa, on tarpeen määritellä tietojoukkojen yhteiset ominaisuudet arvioinnin perustaksi (Ring ym. 2019).

Vaikka IDS-tietojoukkojen luomisesta on tehty merkittävästi tutkimuksia, IDS-tietojoukkojen evaluoinnista ja arvioinnista on tehty vain vähän tutkimusta (Sharafaldin ym. 2017). Tämän tutkimuksen tarkoituksena onkin löytää tietojoukkoista ne ominaisuudet, joiden avulla voidaan suorittaa tietojoukkojen vertailua tunkeutumisen havaitsemisjärjestelmille relevantilla tavalla ja tuottaa järkevällä tavalla suunniteltu arviointi yleisimmin käytetyille tietojoukoille. Monet tutkijat kamppailevat löytääkseen kattavia ja päteviä tietojoukkoja ehdotettujen tekniikoiden testaamiseksi ja arvioimiseksi, ja sopivan aineiston saaminen on sinänsä merkittävä haaste (Ferrag ym. 2020). Lisäksi Nehinbe (2011) nostaa esille muita kriittisiä kysymyksiä, kuten tietojoukkojen epäsäännöllisyydet puutteet tietojoukkojen luomismenettelyissä ja sen, että tästä huolimatta tutkijat käyttävät jatkuvasti olemassa olevia aineistoja pääasiassa siksi, että kunkin tietojoukon rajoituksia ei ymmärretä riittävästi.

Yleensä eri tietojoukot korostavat erilaisia tietojoukon ominaisuuksia. Tiettyjen ominaisuuksien tärkeys riippuu niiden arviointiskenaariosta, ja niiden yleistäminen vertailtavaan muotoon on vaikeaa. Tutkijoita onkin kannustettu tutkijoiden löytämään tarpeisiinsa sopivia tietokokonaisuuksia arvioimalla omaa tutkimusskenaariotaan. (Ring ym. 2019 .) Tällä tutkimuksella pyritään tuottamaan arvokasta tietoa siitä, miten ja millä perusteilla tietojoukkoja voidaan valita tunkeutumisen havaitsemisjärjestelmien käyttöön ja tutkimukseen tulevaisuudessa. Pyrkimyksenä on tuottaa malli, jonka avulla on nykyistä helpompi vertailla tietojoukkoja keskenään ja tämän avulla valita tietojoukko. Tässä tutkimuksessa pyritään tunnistamaan ne arviointikriteerit ja tietojoukkojen ominaisuudet, joilla tietojoukkoja pystytään vertailemaan keskenään. Tunnistetuista ominaisuuksista pyritään muodostamaan tehokkaan vertailun mahdollistava mittaristo, jota sitten sovelletaan tiettyjen tietojoukkojen vertailuun.

2 Tutkimus

Tutkimuksen tarkoituksena on löytää tietojoukoista ne ominaisuudet, joiden avulla voidaan suorittaa tietojoukkojen vertailua tunkeutumisen havaitsemisjärjestelmille relevantilla tavalla, ja näiden perusteella muodostaa käyttökelpoinen vertailumalli. Ominaisuuksilla on oltava jonkinlainen suhde tai vaikutus sen avulla saavutettavan tunkeutumisen havaitsemisjärjestelmän toimivuuteen. Kaikki tietojoukkojen väliset eroavaisuudet ja ominaisuudet eivät välttämättä suoraan vaikuta niitä käyttävään järjestelmään, joten tutkielmassa tulee tunnistaa ja käsitellä vain tässä kontekstissa relevantteja ominaisuuksia. Ominaisuuksia ja arviointikriteerejä suunniteltaessa ja tutkittaessa on tärkeää tiedostaa siihen liittyvät mahdolliset ongelmat aikaisemman tutkimuksen perusteella, jotta voidaan välttyä tunnetuilta kompastuskiviltä. Tiedostettujen ongelmien perusteella pystytään tuottamaan toteutuksia, jotka reagoivat aiemmin tunnistettuihin ongelmiin.

Kun tietojoukoista on tunnistettu vertailukelpoiset ominaisuudet, on tarkoitus johtaa näistä ominaisuuksista jonkinlainen ”vertailumittari” ja soveltaa sitä olemassa oleviin tietojoukkoihin. Löydettyjen arviointitapojen soveltaminen yleisesti saatavilla oleviin tietojoukkoihin antaa mahdollisuuden testata suunnitellun arviointitavan toimivuutta, antaen samalla tietoa eri tietojoukkojen eroavaisuuksista mahdollisesti uudella tarkkuudella.

- Mitkä ovat ne tietojoukkojen ominaisuudet, jotka muodostavat perustan käytettävissä olevalle IDS-tietojoukkojen vertailulle ja arvioinnille?
- Mitä mahdollisia ongelmia liittyy tiettyyn tapaan arvioida ja verrata tietojoukkojen ominaisuuksia?
- Miten tietojoukkojen vertailu ja arviointi voidaan toteuttaa löydettyjen ominaisuuksien pohjalta?

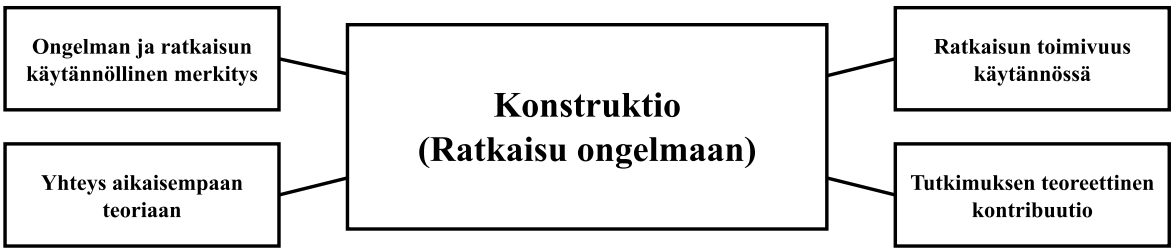
2.1 Tutkimusmetodi

Tutkimusmetodi, jota tässä tutkielmassa tullaan soveltamaan, on konstruktiivinen tutkimusote. Konstruktiivisen tutkimusotteen toteuttamisessa tullaan hyödyntämään ja nojaamaan laajasti Lukan (2001) kuvailemaan prosessiin tutkimusmetodista.

Konstruktiivisessa tutkimuksessa rakennetaan jonkin reaalimaailman ongelman ratkaiseva tuotos, kuten esimerkiksi uusi teoria, malli algoritmi tai ohjelma, joka tämän tutkimusmenetelmän puitteissa on määritelty konstruktioksi (Lukka 2001). Tutkimusmenetelmää hyödyntävän tutkimusprojektin ideana on, että jokin todellinen ongelma ratkaistaan uudella toteutetulla konstruktiolla, jolla on sekä suuri käytännön että teoreettinen panos (Lukka 2003). Konstruktiivista tutkimusta voidaan luonnehtia soveltaviksi tutkimuksiksi, jotka usein johtavat uuteen tietoon normatiivisten sovellusten muodossa (Oyegoke 2011). Esitellyn ongelman ratkaisun suunnittelu ja kehitystyö aiemman tiedon pohjalta ja lopulta tuotetun uuden ratkaisun sitominen teoreettiseen kontribuutioon on konstruktiivisen tutkimuksen ydinajatus.

Konstruktiivisen tutkimuksen lopputuloksena olevalle konstruktiolle ei ole olemassa yhtenäistä mallia, joka kuvaisi kelvollisen konstruktion ominaisuuksia. Lukan (2001) mukaan konstruktio on abstrakti käsite, jolla on loputon määrä mahdollisia toteutumia. Konstruktiolle tyypillisiä ominaisuuksia ovat lähinnä sen uutuusarvo, eli konstruktio on tutkimusprosessin aikana kehitetty uusi asia. Konstruktio ominaisuudeksi voidaan määritellä myös suoraan tutkimusmenetelmästä johdettu ominaisuus, joka on pyrkimys esitellyn ongelman ratkaisuun. Tieteellisen tutkimuksen näkökulmasta konstruktio rakentuu tutkimusmenetelmän osien summana, kuten on nähtävissä Kuva 1.

Konstruktiivinen tutkimusote pyrkii aina kohti ratkaisua, joka voidaan mieltää ”hyväksi”. Tutkimusmenetelmän avulla syntyneellä ratkaisulla, eli konstruktiolla, on saavutettu joitain etuja tutkimusta edeltävään tilanteeseen nähden. Konstruktiivinen tutkimusote edellyttää toteutettavan tutkimuksen tarkkaa kytkemistä aiempaan tutkimukseen ja olemassa olevaan teoriaan ollakseen riittävästi tieteellisten vaatimusten mukainen (Virtanen 2006). Konstruktio on kuitenkin vain pyrkimys ongelman ratkaisuun. Myös konstruktio, joka ei ratkaise esiteltyä ongelmaa, on tutkimusotteen kannalta onnistuneen konstruktiivisen tutkimusprosessin mukainen, kunhan sen voidaan nähdä tuottavan teoreettista kontribuutiota (Lukka 2001).



Kuva 1. Konstruktion rakentuminen (Lukka 2001)

Tässä tutkimuksessa pyritään konstruktiivisella tutkimusotteella muodostamaan malli, jonka avulla on mahdollista suorittaa vertailua eri tietojoukkojen välillä niiden ominaisuuksien perusteella. Konstruktio, jonka tämä tutkimus pyrkii tuottamaan, on siis tietojoukkojen vertailumalli, jota sovelletaan olemassa olevaan ongelmaan. Tutkimus luo siis reaalimaailman ongelmaan ratkaisuksi konstruktion ja sen toimivuus tullaan todentamaan käytännön sovellutuksena.

Konstruktiivinen tutkimusmenetelmän mukaisesti toteutettu tutkimusprosessi etenee tyypillisesti tiettyjen, ennalta tarkasti määriteltyjen, vaiheiden kautta. Lukka (2003) on jakanut tutkimusprosessin seitsemään eri vaiheeseen:

1. Etsi käytännössä relevantti ongelma, jolla on myös potentiaalia teoreettiselle kontribuutiolle.
2. Selvitä mahdollisuudet pitkäaikaiseen tutkimusyhteistyöhön kohdeorganisaation kanssa.
3. Hanki syvällinen ymmärrys aiheesta sekä käytännössä että teoreettisesti.
4. Innovoida ratkaisuidea ja kehittää ongelmanratkaisukonstruktio, jolla on myös potentiaalia teoreettiselle kontribuutiolle.
5. Toteuta ratkaisu ja testaa, miten se toimii.
6. Pohdi ratkaisun sovellettavuutta.
7. Tunnista ja analysoi teoreettista panosta.

Tutkimusprosessista on esitelty ja johdettu myös kuusivaiheisia malleja, jotka muuten rakentuvat samojen ylläesiteltyjen vaiheiden ympärille, mutta niistä puuttuu yllä esitellyistä vaiheista vaihe kaksi (Oyegoke 2011;Virtanen 2006). Vaihe kaksi keskittyykin ainoastaan tutkijan ja kohdeorganisaation väliseen suhteeseen ja näin ollen edellyttää

kohdeorganisaation läsnäoloa tutkimuksessa. Myös tässä tutkimuksessa on jätetty vaihe kaksi tutkimusprosessin ulkopuolelle. Vaihe ei olisi tuottanut tälle tutkimukselle relevanttia lisäarvoa, sillä organisatorisia tai tutkimusyhteistyötä koskevia ongelmia tai rakenteita ei tämän tutkielman piirissä esiinny.

Tämän tutkimuksen kirjallinen rakenne noudattaa sovellettua tutkimusmenetelmää ja sen vaiheita. Kappaleessa kaksi esitellään tutkimusongelma ja -kysymykset, sekä tutkimusmetodi, jolla pyritään löytämään ratkaisu esiteltyyn tutkimusongelmaan. Kappaleet kolme, neljä ja viisi käsittelevät ja avaavat aihealueeseen liittyvää teoriaa, jonka avulla hankitaan syvällinen ymmärrys tutkimuksen aihealueesta. Kappale kuusi käsittelee konstruktion kehittämistä ensin innovaatioprosessia ja sen jälkeen kehitysprossia kuvaillen. Kappaleessa seitsemän testataan kehitettyä konstruktiota soveltamalla sitä käytännössä eri tietojoukkojen rinnakkaiseen vertailuun reaalimaailman skenaariossa. Kappale kahdeksan keskittyy pohtimaan kehitetyn konstruktion teoreettista kontribuutiota ja millaista uutuusarvoa se tuottaa aihealueen tutkimukselle, eli millaisia uusia näkökulmia ja teorioita pystyttiin tuottamaan tietojoukkojen vertailusta ja millaisessa suhteessa ne ovat aihealueen aikaisemman tutkimuksen kanssa. Lopuksi kappale yhdeksän tekee yhteenvedon koko toteutetusta tutkimuksesta ja sen tuloksista analysoiden sekä tutkimusprosessia että tuotettuja ratkaisuja kriittisesti.

3 Tunkeutumisen havaitsemisjärjestelmät

Shireyn (2007) määritelmän mukaan tunkeutuminen on tietoturvatapahtuma tai useiden tietoturvatapahtumien yhdistelmä, joka muodostaa tietoturvatapahtuman, jossa tunkeilija pääsee tai yrittää päästä järjestelmään tai järjestelmäresurssiin ilman valtuutusta. Scarfone ja Mell (2007) puolestaan määrittelevät tunkeutumisen yrityksenä vaarantaa tietyn tietokonejärjestelmän tai verkon luottamuksellisuutta (*engl. confidentiality*), eheyttä (*engl. integrity*) tai saatavuutta (*engl. availability*) tai muutoin ohittaa sen turvamekanismit. Tunkeutuminen voi siis olla mikä tahansa potentiaalisesti haitalliseksi tai luvattomaksi mielletty entiteetti tai niiden yhdistelmä, joka pyrkii vaikuttamaan tietojärjestelmään tietojärjestelmälle epäedullisella tavalla. Tämän määritelmän piiriin kuuluu laaja skaala erilaista toimintaa, mukaan lukien yritykset horjuttaa koko verkkoa, saada luvaton pääsy tiedostoihin tai oikeuksiin, tai yksinkertaisesti ohjelmistojen väärinkäyttö (Portnoy, Eskin ja Stolfo 2001). Kendall (1999) luokittelee tunkeutumisen neljään luokkaan: Palvelunesto (DoS), Remote to Local (R2L), User to Root (U2R) ja Probing. DoS:ssa hyökkääjä pyrkii estämään käyttäjiä pääsemästä tiettyyn palveluun. Kun hyökkääjä yritti saada valtuutetun pääsyn kohdejärjestelmään joko hankkimalla paikallisen pääsyn tai ylentämällä käyttäjän pääkäyttäjäksi, nämä hyökkäykset luokiteltiin R2L:ksi ja U2R:ksi. Lopuksi, ”probing” määritellään hyökkäykseksi, joka etsii aktiivisesti järjestelmän haavoittuvuuksia. Vaikka monet tunkeutumiseksi miellettävät tapahtumat ovat luonteeltaan haitallisia, kaikki eivät kuitenkaan sitä ole. Käyttäjä voi esimerkiksi kirjoittaa väärin tavoiteltavan resurssin osoitteen ja yrittää tarkoituksettomasti muodostaa yhteyden epätarkoituksenmukaiseen järjestelmään ilman asianmukaista valtuutusta (Scarfone ja Mell 2007). Tunkeutumisten ja tunkeutumisyritysten havaitseminen ja niihin reagoiminen järjestelmän suojaamiseksi ovat keskeinen osa-alue jokaisessa tietoturvallisessa tietojärjestelmässä.

Tunkeutumisen havaitseminen on prosessi, jossa seurataan tietokonejärjestelmässä tai tietoverkossa tapahtuvia tapahtumia ja analysoidaan niitä mahdollisten tapahtumien varalta, jotka ovat tietokoneen suojauskäytäntöjen, hyväksyttävän käytön käytäntöjen tai tietoturvakäytäntöjen rikkomuksia tai rikkomisen uhkia (Scarfone ja Mell 2007). Jotta tunkeutumisen havaitseminen ja tunkeutumisen havaitsemisjärjestelmän toteuttama toiminta olisi ylipäätään mahdollista, on haitallisen, tunkeutumiseen liittyvän, informaation oltava ainakin jollain

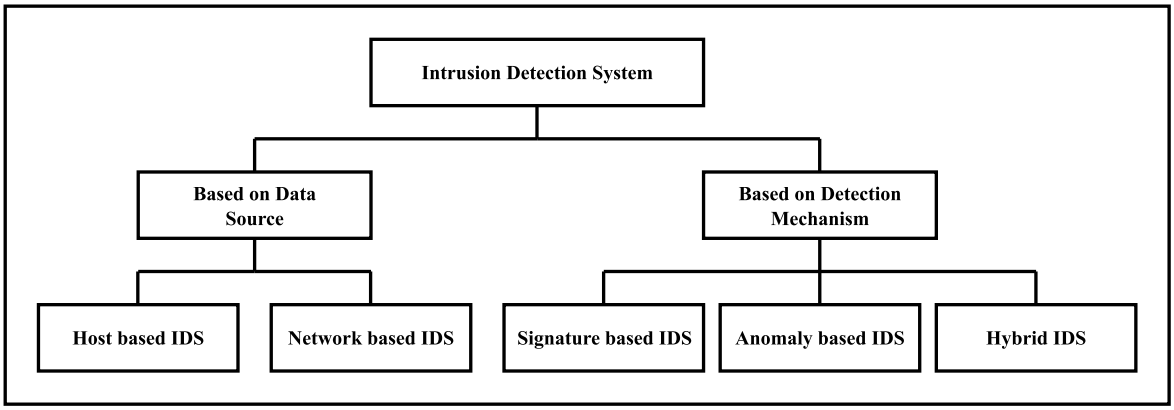
tasolla erilaista verrattuna järjestelmässä olevaan ei-haitalliseen informaatioon. Tunkeutumistapahtumassa generoituvan informaation on siis vähintään jollain tasolla oltava perustavanlaatuisesti erilaista kuin järjestelmän normaalista toiminnasta generoituvan informaation. Mukherjee, Heberlein, ja Levitt (1994) esittävät kaikkien tunkeutumisen havaitsemisjärjestelmien perustuvan uskomukseen, että tunkeilijan käyttäytyminen eroaa huomattavasti laillisen käyttäjän käyttäytymisestä, ja että monet luvattomat toimet ovat havaittavissa.

Tunkeutumisen havaitsemisjärjestelmä (*engl. Intrusion detection system, IDS*) automatisoi prosessin, jossa seurataan tietokonejärjestelmässä tai tietoverkossa tapahtuvia tapahtumia ja analysoidaan niitä tunkeutumisen merkkien varalta. Tunkeutumisen havaitsemisjärjestelmän päätavoite on tunnistaa kaikenlainen luvaton pääsy, väärinkäyttö tai tietojen vioittuminen siinä tietojärjestelmässä, jossa tunkeutumisen havaitsemisjärjestelmä toimii (Yedukondalu ym. 2021). Se on ohjelmisto- tai ohjelmisto- ja laitteistopohjainen ratkaisu, joka automatisoi tunkeutumisen havaitsemisprosessin (Scarfone ja Mell 2007). Käytännönläheisempänä määritelmänä voidaan esittää tunkeutumisen havaitsemisjärjestelmän pyrkivän havaitsemaan tiettyyn järjestelmään kohdistuvat luvattomien tahojen hyökkäykset. Tunkeutumisen havaitsemisjärjestelmän teoreettiseen toimintaperiaatteeseen kuuluu ainoastaan potentiaalisten tunkeutumisten havaitseminen ja niistä ilmoittaminen, ei niiden estäminen tai niihin reagointi minkäänlaisilla vastatoimenpiteillä. Kendallin (1999) kuvauksen mukaan tunkeutumisen havaitsemisjärjestelmä yleensä ilmoittaa ihmisanalyytikolle mahdollisesta tunkeutumisesta eikä ryhdy jatkotoimiin, mutta jotkin uudemmat järjestelmät ryhtyvät aktiivisiin toimenpiteisiin tunkeilijan pysäyttämiseksi havaitsemishetkellä. Yleisesti tunkeutumisen havaitsemisjärjestelmä välittää tiedon havaitsemistaan mahdollisista tunkeutumisista tai niiden yrityksistä eteenpäin, joko erilliselle tunkeutumisen estojärjestelmälle (*engl. Intrusion prevention system, IPS*) tai kyseisen järjestelmän tietoturvasta vastaavalle taholle.

3.1 Järjestelmien luokittelu

Tunkeutumisen havainnointijärjestelmiä pystytään kategorisoimaan eri luokkiin niiden toteuttaman tunnistusmenetelmän sekä järjestelmän sijoituskohteen mukaan (Lakshminarayana, Philips ja Tabrizi 2019). Tämä yleisimmin sovellettu kategorisointitapa on havainnollistettu Kuva 2. Tunnistusmenetelmällä tarkoitetaan järjestelmän toimintamallin

logiikkaa, jonka avulla järjestelmä havaitsee ja tunnistaa onko sille syötetty informaatio haitallista vai ei-haitallista. Järjestelmän sijoituskohteella tarkoitetaan monitorointiympäristöä tai sijaintia järjestelmän arkkitehtuurissa, johon järjestelmä sijoitetaan havaitsemaan poikkeavia tapahtumia.



Kuva 2. Järjestelmien kategorisointi (Sulaiman ym. 2021)

Sijoituskohde määrittelee sen millaista informaatiota järjestelmä tulee käsittelemään ja minkälaisista lähteistä tämä informaatio on peräisin. Anwar ym. (2017) määritelmän mukaan isäntäpohjaiset tunkeutumisen havaitsemisjärjestelmät asennetaan tiettyyn koneeseen, kuten palvelimeen ja mobiililaitteisiin, ja ne valvovat kyseisen laitteen käyttöjärjestelmän tarkastustietoja tunnistaakseen mahdolliset tunkeutumiset ja niiden yritykset. Lisäksi ne havaitsevat, mitkä ohjelmat käyttävät mitäkin järjestelmän osaa tai resursseja. Verkkopohjaiset tunkeutumisen havaitsemisjärjestelmät ottavat toisenlaisen näkökulman ja siirtävät painopisteensä laskennallisesta infrastruktuurista (isännät ja niiden käyttöjärjestelmät) viestintäinfrastruktuuriin (verkko ja sen protokollat) (Vigna ja Kemmerer 1998), ja valvovat verkkoliikennettä minkä tahansa kahden tietokoneen tai määriteltyjen verkkosegmenttien välillä kaikenlaisten tunkeutumisten varalta. Sijoituskohteen ja monitorointiympäristön mukaan järjestelmät havainnoivat tunkeutumisyrityksiä hyvin erityyppisestä informaatiosyötteestä. Verkkopohjaiset tunkeutumisen havainnointijärjestelmät analysoivat verkkoliikennedataa, kun taas isäntäpohjaiset analysoivat isäntäjärjestelmän generoimia lokitiedostoja. García-Teodoro ym. (2009) kuvailevat järjestelmien eroja niiden tarkastelemien informaatiotyyppien mukaan; Isäntäpohjainen tunkeutumisen havaitsemisjärjestelmän analysoi tapahtumia, kuten prosessitunnisteita ja järjestelmäkutsuja, jotka liittyvät pääasiassa

käyttöjärjestelmätietoihin. Toisaalta verkkopohjainen tunkeutumisen havaitsemisjärjestelmän analysoi verkkoon liittyviä tapahtumia: liikenteen määrää, IP-osoitteita, palveluportteja, sekä sovellusten ja protokollien toimintaa. Bukac, Tucek ja Deutsch (2012) huomauttavat, että viime vuosien aikana isäntäpohjaiset tunkeutumisen havainnointijärjestelmät (HIDS) eivät olleet tietoturvatutkimuksissa päähuomiossa ja myös tämä tutkimus keskittyy tutkimaan ainoastaan tietoverkkopohjaisia tunkeutumisen havainnointijärjestelmiä.

Etenkin verkkopohjaisiatunkeutumisen havainnointijärjestelmiä on perustoimintaperiaatteiltaan kahdenkaltaisia; tunnisteisiin pohjautuvia ja anomalioihin pohjautuvia. Tunnistepohjainen tunkeutumisen havaitsemisjärjestelmä tuottaa vasteen, jos jokin ennalta määritetty tunniste (*engl. signature*) havaitaan vastaavan verkon nykyistä käyttäytymistä (Amoli ym. 2015). Anomaliapohjainen järjestelmä oppii ensin tietoliikennettä analysoimalla, minkälainen liikenne on normaalia, turvallista tietoliikennettä ja mikä haitallista liikennettä, eli poikkeavia anomalioita. Bolzoni ja Etalle (2008) kuvailevat mallien eroja niiden etujen kautta; Anomaliapohjaisilla järjestelmillä on se etu, että toisin kuin tunnistepohjaiset järjestelmät ne pystyvät havaitsemaan nollapäivähyökkäykset, koska uudet hyökkäykset voidaan havaita heti niiden tapahtuessa. Toisaalta anomaliapohjaiset, toisin kuin tunnistepohjaiset järjestelmät, vaativat koulutusvaiheen ja luokittelurajojen huolellisen asettamisen, mikä tekee niiden käyttöönotosta ja implementoinnista huomattavasti monimutkaisempaa. Vaikka tunnistepohjaista havaitsemista suositaan yleensä kaupallisissa tuotteissa sen ennakoitavuuden ja suuren tarkkuuden vuoksi, akateemisessa tutkimuksessa anomaliapohjainen havaitseminen on tyypillisesti nähty tehokkaammaksi menetelmäksi, koska sillä on teoreettinen potentiaali puuttua uusiin ja tuntemattomiin hyökkäyksiin (Tavallaee ym. 2009). Vaikka tutkimuksissa usein korostetaan tietyllä ratkaisulla saavutettua tehokkuutta, on jokaisella näistä lähestymistavoista omat etunsa ja haittansa kustannusten, suorituskyvyn ja muiden mittareiden suhteen (Dina ja Manivannan 2021).

3.2 Tunnistepohjaiset järjestelmät

García-Teodoro ym. (2009) määritelmän mukaan tunnistepohjaiseen tunkeutumisenhavainnointiin perustuvat mallit (myös allekirjoituspohjainen tai väärinkäyttöpohjainen) etsivät ennalta määritettyjä malleja tai tunnisteita (*engl. signatures*) analysoiduista tiedoista. Khraisat

ym. (2019) tiivistävät järjestelmän käytännön toiminnan; kun tunkeutumisen tunniste täsmää edellisen tunkeutumisen tunnisteen kanssa, joka on jo olemassa tunnistetietokannassa, havaitsemisesta määritelty hälytyssignaali laukeaa. Jokaista järjestelmään saapuvaa datan instanssia verrataan siis ennalta määritettyyn tunnisteiden tietokantaan. Järjestelmän ei siis tarvitse tunnistaa erikseen, millaista on normaali, ei-haitallinen data. Järjestelmä ainoastaan tarkastelee syötettyä dataa ja tunnistaa, sisältääkö se tunnisteen vai ei (Uddin ym. 2013).

Tätä tarkoitusta varten määritellään järjestelmälle etukäteen tunnettuja hyökkäyksiä vastaava tunnistetietokanta. Tunniste on malli, kaava tai merkkijono, joka vastaa tunnettua hyökkäystä tai uhkaa (Liao ym. 2013). Tunnistetietokanta koostuu näistä ennalta määritellyistä tunnisteista ja näin tunnistetietokannan sisältö määrittelee ne tunnisteet, jotka tunnistepohjaisen tunkeutumisenhavainnointimallin on mahdollista havaita. Käytännössä tunnistetietokantaa on päivitettävä jatkuvasti uusimpien uhkien havaitsemiseksi (Chkirbene ym. 2020). Tämä tietokannan päivittämistiheys on oltava siis suorassa suhteessa uusien tunkeutumismallien syntymisen määrään, jotta järjestelmä pysyy tietoturvallisena. Yadav ja Sharma (2021) korostavatkin tunnistepohjaisen tunkeutumisenhavainnointi järjestelmän suurimman haittapuolen olevan, että se voi havaita tunkeutumisyrityksen vain, jos se sopii tietokannassa olemassa olevaan trendiin. Tämän takia tunnistepohjaiset järjestelmät ovat haavoittuvaisia uusille hyökkäyksille, kunnes niiden tunnistetietokanta päivitetään ajan tasaiseksi. tunnistepohjaisen tunkeutumisenhavainnointi järjestelmän suurimmat edut kuitenkin ovat tunnettujen hyökkäysten havaitsemisen suuri tarkkuus (Hindy ym. 2020) sekä helpompi toteutus ja konfigurointi etenkin suurissa ympäristöissä (Kruegel ja Toth 2003).

Uusien kehittyvien kyberuhkien tunnistamisen kyvyttömyys ja manuaalisten allekirjoitustietokantapäivitysten tarve rajoittavat allekirjoituspohjaisten tunnistusjärjestelmien tehokkuutta (Ullah ja Mahmoud 2021). Lisäksi nykypäivän hyökkäykset ovat luonteeltaan muuttuvia ja mukautuvia, jonka takia yksittäisten tunnisteiden on vaikea vastata kaikkia mahdollisia hyökkäyksiä. Esimerkiksi SQL Injektio -hyökkäysten esittämän korkean polymorfismin vuoksi on mahdotonta tuottaa muutamia yleisiä allekirjoituksia niiden havaitsemiseksi, ja useimmat näistä hyökkäyksistä jäävät huomaamatta (Bolzoni ja Etalle 2008). Yhdeksi vastaukseksi näihin haasteisiin on kehitetty anomaliapohjaisia tunkeutumisenhavainnointi järjestelmiä.

3.3 Anomaliapohjaiset järjestelmät

Ennalta määritettyjen ja ylläpidettävien tunnisteiden sijaan anomaliapohjainen tunkeutumisen havainnointi luottaa toisenlaiseen havainnointimenetelmään. Anomaliapohjaiset järjestelmät pyrkivät havainnoimaan anomalioita, eli normaalista poikkeavaa liikennettä ottamatta kantaan siihen mikä epänormaalin liikenteen aiheuttaja tai syy on. Liao ym. (2013) määrittelevät tunkeutumisen havaitsemisen kontekstissa anomalian olevan poikkeama odotettuun käyttäytymiseen, joka on johdettu säännöllisten toimintojen, verkkoyhteyksien, isäntien tai käyttäjien seurannasta koostuvasta informaatiosta.

Anomaliapohjaisia järjestelmiä opetetaan tunnistamaan haitallisia anomalioita altistamalla ne tietoliikennedatalle, jota analysoimalla järjestelmät tulevat kouluttautuva tai mukautuvat verkon ja verkkosolmujen normaaliin ja epänormaaliin käyttäytymiseen Analysointivaiheen jälkeen järjestelmä nostaa hälytyksiä, kun sen tietoliikennedatasta oppimat kynnysarvot ylittyvät, jolloin se pitää näitä toimia epänormaalina ja tuottaa jonkin vasteen tai ilmoituksen. (Amoli ym. 2015). Tämä tarkoittaa, että anomalioihin perustuva tunkeutumisen havaitseminen havaitsee epänormaalia liikennettä sen sijaan, että se havaitsisi tiettyjä hyökkäyksiä, hyökkäyksen ominaisuuksien perusteella (Le Jeune, Goedeme, ja Mentens 2021). Kaikki normaaliksi ja tavalliseksi opitusta informaatiosta poikkeavat, eli epänormaalit informaatiosyötteet luokitellaan poikkeavuuksiksi. Bolzoni ja Etalle (2008) jakavat anomaliapohjaiset tunkeutumisen havaitsemisjärjestelmät kahteen toiminnallisuuden vaiheeseen. Näissä järjestelmissä algoritmi rakentaa (ns. koulutusvaiheen aikana) ensin tilastollisen mallin legitiimistä, eli hyökkäämättömästä verkkokäyttäytymisestä; myöhemmin (havaitsemisvaiheessa) syötettä verrataan aiemmin luotuun malliin etäisyysfunktiolla, ja kun mitattu etäisyys ylittää tietyn kynnyksen, syöte katsotaan poikkeavaksi eli hyökkäykseksi.

Näiden ominaisuuksien perusteella tunnistepohjaiset järjestelmät ovat erittäin tehokkaita löytämään ennalta määritettyjä hyökkäysmalleja, mutta ovat käytännössä aseettomia uusien tuntemattomien hyökkäysten edessä. Anomaliapohjaiset tunkeutumisen havainnointijärjestelmät mahdollistava entuudestaan tuntemattomien hyökkäysmallien havaitsemisen, sillä hyökkäysmallia ei tarvitse entuudestaan olla määritelty, vaan riittää että se on jollain tasolla poikkeavaa normaalista, hyväntahtoisesta tietoliikennedatasta. Tämä mahdollistaa teoriassa

tuntemattomien uhkien kuten nollapäivähaavoittuvuuksien tunnistamisen ja niiltä suojautumisen. Anomalioihin perustuvien havaitsemistekniikoiden tärkein etu on niiden kyky havaita aiemmin näkymättömiä tunkeutumistapahtumia. Huolimatta muodollisten allekirjoitusmääritysten todennäköisestä epätarkkuudesta, väärien positiivisten (*engl. false positive, virheellisesti hyökkäyksiksi luokiteltujen tapahtumien*) määrä anomalioihin perustuvissa tunkeutumisen havaitsemisjärjestelmissä on yleensä korkeampi kuin allekirjoituksiin perustuvissa (Uddin ym. 2013).

Tungjaturasopon ja Piromsopa (2018) mukaan anomalioihin perustuvat tunkeutumisen havaitsemisjärjestelmät voidaan edelleen luokitella kolmeen ryhmään tunnistustekniikoiden perusteella. Ne ovat: tilastollinen, tietopohjainen ja koneoppimiseen perustuva. Näistä koneoppimispohjaisia järjestelmiä ja algoritmeja on aihealueen tutkimuksissa yleisesti pidetty potentiaalisesti yhtenä tehokkaimmista tavoista ja työkaluista käsitellä tämän päivän ja tulevaisuuden hyökkäyksiä ja tunkeutumisia. Hindy ym. (2020) tekemän kartoituksen mukaan koneoppimisalgoritmeja sovellettiin 97,25 %:ssa tutkituista tunkeutumisen havainnointijärjestelmien kehittämiseen kohdistuneesta tutkimuksesta.

Verkon tunkeutumisen havaitsemisen tapauksessa tekoälymallin pitäisi pystyä tarkastamaan verkkoliikennettä ja havaitsemaan kaikki hyökkäyksiin liittyvät epänormaalit kuviot, kuten muutokset datanopeudessa ja saapumisajoissa tai epätäydelliset pyynnöt tietyistä verkko-osoitteista. Tällaisten mallien kouluttaminen vaatii merkittyä dataa tai synteettisesti luotuja tietojoukkoja ei-haitallisten ja haitallisten mallien binääristä tai moniluokkaista tunnistamista varten (Sabeel ym. 2021).

4 Koneoppiminen tunkeutumisen havaitsemisessa

Koneoppimistekniikoita ja niiden sovellutuksia käytetään yhä enemmän ja enemmän yhtenä lähestymistapana poikkeamien havaitsemiseen (Nassif ym. 2021). Koneoppimistekniikoiden soveltaminen anomaliapohjaisten tunkeutumisenhavaitsemisjärjestelmien kehittämisessä on ollut jatkuvasti kasvava trendi ja nykyinen järjestelmien tutkimus ja kehitys keskittyy laajalti erilaisten koneoppimistekniikoiden ympärille. Erilaisten koneoppimismetodien kehittyessä, on niitä pyritty hyödyntämään nopeasti myös tunkeutumisen havaitsemisen tutkimuksessa. Koneoppimisen perusperiaate, joka sallii sen tehdä itsenäisiä päätelmiä monimutkaisesta ja laajasta data määrästä, vastaa luonteeltaan vahvasti anomaliapohjaisen tunkeutumisen havainnoinnin metodologiaa ja tekee näin tunkeutumisen havainnoinnista selkeän sovellutuksen kohteen koneoppimiselle. Jhan (2013) mukaan useat eri tutkijat ja tutkimusryhmät ovat tutkineet tunkeutumisen havaitsemisjärjestelmien suunnitteluun useita koneoppimisalgoritmeja, mukaan lukien neuroverkot (*engl. neural networks*), päätöspuut (*engl. decision trees*), tukivektorikoneet (*engl. Support Vector Machines*) ja Bayes-verkot.

Koneoppimisessa käytetään erilaisia algoritmeja jäsentämään jonkin tietyn ongelman (esim. tunkeutumisen havaitseminen) ratkaisemiseen liittyvää dataa, oppimaan siitä ja tämän oppimisen avulla datasta löydetään malleja, jotka ovat hyödyllisiä tämän tietyn ongelman ratkaisemisessa (Dina ja Manivannan 2021). Koneoppiminen mahdollistaa ongelmanratkaisun ilman, että järjestelmää olisi spesifisti koodattu ratkaisemaan juuri tiettyä ongelmaa. Vaikka tutkimuksesta löytyy huomattavasti erilaisia koneoppimiseen perustuvia järjestelmäratkaisuja, soveltuvuus kaupallisiin järjestelmiin on vasta alkuvaiheessa (Vinayakumar ym. 2019).

Sommerin ja Paxsonin (2010) mukaan tunkeutumisen havaitsemisen kontekstissa koneoppimisella tarkoitetaan algoritmeja, jotka ensin opetetaan referenssisyötteen avulla "oppimaan" sen erityispiirteet (joko valvottuna tai ilman valvontaa), jotta niitä voidaan sitten käyttää aiemmin näkemättömällä syötteellä varsinaista tunnistusprosessia varten. Järjestelmän harjoitusvaiheessa koneoppimismalli määrittelee harjoitusdatan pohjalta normaalin liikenteen mallin. Testausvaiheessa opittua mallia sovelletaan uuteen syötteeseen, eli testausjoukkoon, ja jokainen testausjoukon instanssi luokitellaan järjestelmän toimesta joko normaaliksi tai poikkeavaksi (Buczak ja Guven 2016). Edelleen verkkopohjaisten tunkeutumisen

havainnointijärjestelmän kontekstissa syötteet ovat jossakin määritellyssä muodossa olevaa tietoliikennedataa. Luokittelutehtäviä voidaan pitää joko binäärisen luokittelun ongelmina (*engl. binary classification*), jossa syöte luokitellaan joko normaaliksi tai haitalliseksi, tai moniluokkaisina ongelmina (*engl. multi-class classification*), jossa haitallisella liikenteellä on useampia malleja kuten esimerkiksi normaali, DoS, probe, R2L ja U2R (Devulapalli 2021). Käytännössä moniluokkainen luokittelu jakaa haitallisen liikenteen vielä omiin alaluokkiinsa esimerkiksi eri hyökkäystyyppien mukaan.

Koneoppimistekniikat luokitellaan yleensä kolmeen laajaan luokkaan, jotka ovat ohjattu oppiminen, ohjaamaton oppiminen ja vahvistusoppiminen (Farah ym. 2015). Nämä luokat eroavat toisistaan perustavanlaatuisesti niiden oppimismenetelmien ja koulutusdatan perusteella. Ohjatussa oppimisessa harjoitusdata on valmiiksi merkitty ja luokiteltu (*engl. labeled*). Ohjaamattomassa oppimisessa harjoitusdataa ei ole millään tavalla merkitty tai luokiteltu ja koneoppimismalli etsii datasta merkitsemättömiä toistuvuuksia ja kaavoja, jonka perusteella se tekee luokittelunsa. Ohjaamaton oppiminen ei vaadi merkittyä tietojoukkoa, sillä se luottaa siihen oletukseen, että hyökkäykset ovat jonkin verran poikkeavaa dataa, minkä jälkeen ne voidaan ryhmitellä omaksi joukokseen (Dang 2021). Tämä olettamus aiheuttaa kuitenkin omat ongelmansa, sillä ei ole olemassa tarkkaa määritelmää hyökkäysliikenteen mallille. Tällöin hyökkäysliikenne ei välttämättä generoi datan malleja, jotka ohjaamattoman oppimisen algoritmien on mahdollista havaita. Vahvistusoppimisessa data on myös merkitsemätöntä, mutta koneoppimismenetelmä saa ”palautetta” tekemistään luokitteluista. Palautteen perusteella vahvistusoppimismalli pystyy suuntaamaan omaa oppimistaan.

Eri koneoppimismallit ja tekniikat suoriutuvat eri tavoilla riippuen syötteen ominaisuuksista, eli valituista harjoitus- ja testaustietojoukoista. Samankaltaiset lähestymistavat, oppimistekniikat ja syötteen ominaisuudet eivät aina takaa eivät aina takaa samoja tuloksia useille eri mahdollisten tuntemattomien hyökkäysten luokille (Gurung, Kanti Ghose, ja Subedi 2019). Koneoppimismenetelmiä hyödyntävän luokittelijan suoriutuminen on siis aina sidonnainen tietojoukkoon ja sen attribuutteihin ja suorituskyky vaihtelee harjoitustietojoukon mukaan. Suleiman ja Issac (2018) päättelevät, että tunkeutumisen havaitsemisjärjestelmille ei ole olemassa yhtä täydellistä koneoppimismallia, koska niillä on keskenään erilaisia ja ainutlaatuisia ominaisuuksia.

Kumarin, Guptan ja Aroran (2021) mukaan tunkeutumisen havaitsemisen kontekstissa, kuten myös useassa muussa koneoppimista hyödyntävässä sovellutuksessa, koneoppimismallien suoriutumista ja tehokkuutta arvioidaan seuraavilla parametreilla:

- Oikea positiivinen (*engl. true positive, TP*). Eli oikeellisesti normaaliksi luokitellut instanssit.
- Oikea negatiivinen (*engl. true negatives, TN*). Eli oikeellisesti poikkeuksiksi luokitellut instanssit.
- Väärä positiivinen (*engl. false positives, FP*). Eli väärin normaaliksi luokitellut instanssit.
- Väärä negatiivinen (*engl. false negatives, FN*). Eli väärin hyökkäyksiksi luokitellut instanssit.

Näiden parametrien avulla on mahdollista laskea järjestelmän toiminnan tehokkuutta ja tarkkuutta kuvaavia arvoja, sen testausvaiheessa suorittaman luokittelun pohjalta. (Dina and Manivannan 2021) mukaan tärkeimmät tehokkuutta ja tarkkuutta kuvaavat arvot ovat ”accuracy”, ”precision” sekä ”recall”. Accuracy on niiden tapahtumien osuus, jotka järjestelmä luokittelee täysin oikein ja Precision on niiden tapahtumien osuus, jotka algoritmi luokittelee tunkeutumiseksi ja jotka ovat oikeasti tunkeutumisia. Recall on niiden todellisten tunkeutumisten osuus, jotka algoritmi luokitteli tunkeutumisiksi.

4.1 Koneoppimismenetelmät tunkeutumisen havaitsemisessa

Laajassa kartoituksessaan Amarudin, Ferdiana, ja Widyawan (2020) tunnistavat tunkeutumisen havaitsemisjärjestelmissä käytettyjen useiden menetelmien joukosta kuusi eniten käytettyä koneoppimismenetelmää ja niiden osuutta tutkimuksesta, jotka ovat: k-NN=7 %, RF=7 %, NB=15 %, DT=17 %, NN=20 % ja SVM = 34 %.

Jokaisella koneoppimismenetelmällä on omat heikkoutensa ja vahvuutensa, ja ne suoriutuvat eri tavoin erilaisista harjoitus- ja syötejoukoista. Niiden oppimisprosesseissa ja päätöksentekomalleissa on fundamentaalisia eroja, jonka takia onkin tärkeää valita tilanteeseen ja käsiteltävään tietojoukkoon sopiva koneoppimismenetelmä.

4.1.1 Tukivektorikone

Tunkeutumisen havaitsemisjärjestelmissä yleisimmin käytetty koneoppimismalli on tukivektorikone (*engl. support-vector machine, SVM*). SVM on luokittelija, joka perustuu erottavan hypertason (*engl. hyperplane*) löytämiseen kahden luokan välisestä piirreavaruudesta siten, että hypertason ja kunkin luokan lähimpien datapisteiden välinen etäisyys maksimoidaan (Ganesh 2019). Tukivektorikoneiden metodologialla konstruoitu luokittelija erottaa äärellisen alueen, jossa normaalia liikennettä kuvaavat objektit ovat ja kaiken muun tilan oletetaan sisältävän poikkeamat (Farah ym. 2015). SVM tarjoaa hyvän yleistyskyvyn suhteellisen pienelläkin harjoitustietojoukolla (Mishra ym. 2019).

4.1.2 Naïve Bayes

Naïve Bayes on valvotun oppimisen menetelmä, joka perustuu Bayesin teoreemaan, ja joka pyrkii ennustamaan syötteen luokan tietojoukon attribuuttien perusteella. Naïve Bayes menetelmässä oletus, että kunkin attribuutin todennäköisyydet luokkaan nähden ovat riippumattomia kaikista muista attribuuttiarvoista, on erittäin vahva (Netti ja Radhika 2015). Naïve Bayes olettaa, että attribuutit ovat ehdollisesti riippumattomia ja yrittää siten estimoida luokkaehtoisen todennäköisyyden. Naive Bayes tuottaa usein hyviä tuloksia luokittelussa, jossa on olemassa yksinkertaisempia suhteita (Farah ym. 2015).

4.1.3 Päätöspuu

Valvotun oppimisen menetelmä päätöspuu (*engl. decision tree, DT*) on puumainen rakenne, jossa on lehtiä (*engl. leaves*), jotka edustavat luokituksia ja oksia (*engl. branches*), jotka puolestaan edustavat niihin luokitteluihin johtavien ominaisuuksien konjunktiota. Esimerkki merkitään (luokitellaan) testaamalla sen ominaisuuden (attribuutin) arvoja päätöspuun solmuja vastaan. (Buczak ja Guven 2016.) Päätöspuita on käytetty laajasta etenkin luokittelua vaativiin tehtäviin, joten se on yleisesti nähty sopivaksi menetelmäksi luokittelua edellyttävään tunkeutumisen havaitsemiseen (Yang ym. 2022)

4.1.4 k-nearest Neighbor

K-nearest neighbor-algoritmin (*k-NN*) päätoimintamalli on seuraava: Se laskee etäisyyden uudesta syötteestä jo tunnettuihin instansseihin määrittääkseen *k* lähintä naapuria. Kun *k* lähintä naapuria on kerätty, yksinkertainen enemmistö näiden *k* lähimmän naapurin luokista on syötteelle annettava luokan ennuste. Jos syötteen *k* lähimmät naapurit kuuluvat samaan luokkaan, kuuluu myös kyselyinstanssi. Muussa tapauksessa uuden syötteen tulos luokitellaan *k* lähimmän naapurin luokan enemmistön perusteella. (Xiao and Ding 2012). KNN on valvotun oppimisen menetelmä, muista poiketen ei sisällä varsinaista koulutusvaihetta. Se tallentaa kaikki instanssit etäisyyslaskelmiin ja käyttää etäisyyslaskentaan Euklidisia, Manhattanin tai Minkowskin funktioita (Karatas, Demir, ja Sahingoz 2020).

4.1.5 Satunnaismetsä

Satunnaismetsät (*engl. random forest, RF*) hyödyntävät päätöksenteossaan päätöspuu algoritmeja. Ne käyttävät ensemble-menetelmää, jossa ne luovat jonkin joukon päätöspuita ja yhdistävät ne saadakseen paremman tarkkuuden ja vakauden omaan päätelmäänsä. Mitä useampaa päätöspuuta satunnaismetsä hyödyntää, sitä parempi on sen luokittelun tehokkuus. (Faker ja Dogdu 2019.) Satunnaismetsät koostuvat aina satunnaisesta ryhmästä päätöspuita, joita se hyödyntää päätöksenteossaan. Yksinkertaistettuna, jokainen satunnaismetsän päätöspuu tuottaa oman ratkaisunsa, eli luokkaennusteen. Yhdistettynä eniten ääniä saanut luokka tulee koko satunnaismetsän ennusteeksi, eli menetelmän tuottamaksi vastaukseksi. (Chindove ja Brown 2021). Se on yksi suosituimmista menetelmistä, koska se tarjoaa nopeasti nopeita ja tarkkoja tuloksia jopa sekaisille, epätäydellisille ja meluisille tietojoukoille (Karatas, Demir ja Sahingoz 2020).

4.1.6 Neuroverkot

Tyypillisesti neuroverkot (*engl. neural networks*) on järjestetty useisiin kerroksiin, jotka koostuvat useista toisiinsa kytketyistä solmuista, jotka sisältävät aktivointifunktion. syöte esitetään verkolle sen ensimmäisen, eli syöttökerroksen kautta, joka kommunikoi yhteen tai useampaan piilotettuun kerrokseen, jossa painotettujen yhteyksien järjestelmän kautta

varsinainen käsittely tapahtuu. Piilotetut kerrokset linkittyvät sitten tuloskerrokseen, joka tuottaa luokittelutuloksen verkon ulostulona. (Farah ym. 2015). Syöttökerros ottaa vastaan syötteen ulkopuolelta, kun taas tuloskerros reagoi syöttökerrokseen syötettyyn syötteeseen oman toiminnallisuutensa perusteella. Piilotetut kerrokset toimivat välittäjinä syöte- ja tuloskerroksen välillä ja muuttavat syötettä tietyllä tavalla. Nämä kerrokset voivat olla osittain tai kokonaan linkitettyinä (Maseer ym. 2021).

5 Tietojoukot

Koneoppimistutkimukseen perustuvan tunkeutumisen havaitsemisen pääpiirteenä on tunnistaa yhtäläisyyksiä ja piirteitä tietojoukosta ja rakentaa sen perusteella tunkeutumisen havaitsemisen malli (Khraisat ym. 2019). Tunkeutumisen havaitsemisjärjestelmien kontekstissa tietojoukoilla tarkoitetaan jossain tietyssä muodossa olevaa tietoliikennettä kuvaavaa dataa, joka on koostettu yhdeksi tai useammaksi joukoksi. Tätä tietojoukkoa käytetään koneoppimismallin koulutustietojoukkona sekä näiden mallien testaukseen ja arviointiin. Riippuen käytettävästä menetelmästä, tietojoukko on saatettu jakaa erillisiin koulutus- ja testaustietojoukkoihin, tai tietojoukkoa voidaan käyttää sellaisenaan. Esimerkiksi ohjatussa oppimisessa luokittelutehtävä vaatii kaksi erillistä tietojoukkoa: yksi koulutusta ja toinen testausta varten. Oppimisalgoritmi luo säännöt koulutustietojoukolle ja sen suorituskykyä mitataan sen testausjoukon luokittelutuloksilla. (Cieslak, Chawla ja Striegel 2006). Lisäksi, Tietojoukot antavat meille mahdollisuuden arvioida ehdotettujen menetelmien kykyä havaita tunkeileva liikennemallit (Khraisat ym. 2019). Käytännössä tietojoukkoa, tai jonkin tietojoukon osajoukkoa, käytetään joko koneoppimispohjaisen tunkeutumisen havaitsemisjärjestelmän koulutukseen, sen toiminnan arvioimiseen tai molempiin näistä tehtävistä. Aiempi tietojoukkoja itsessään käsittelevä tutkimus on keskittynyt lähinnä muutamaan systemaattiseen katsaukseen, jotka pureutuvat analysoimaan eri tietojoukkojen ominaisuuksia ja miten niitä on käytetty. Laajaa tietojoukkoja tarkasti vertailevaa tutkimusta ei ole tehty, vaikka pienimuotoisia vertailuja on tehty osana muuta tutkimusta, esimerkiksi tietojoukon valinnan perusteluna. Bhuyan, Bhattacharyya ja Kalita (2015) luettelevat seuraavat tärkeimmät syyt perustellessaan vakiotietojoukon tärkeyttä: kokeiden toistettavuus, uusien lähestymistapojen validointi, eri lähestymistapojen vertailu, malliparametrien viritys, ulottuvuus tai ominaisuuksien määrä tietojoukossa. Nämä mahdollistavat selkeän jatkuvuuden tutkimuksessa verrattuna tilannekohtaisesti kerättyyn dataan.

Koska tietojoukolla on suora vaikutus sekä järjestelmän toiminnallisuuden ja logiikan muodostumiseen että sen suoriutumisen arvioimiseen, on erittäin tärkeää, että käytössä oleva tietojoukko on laadukas ja tarpeeseen sopiva. Hyvän tietojoukon valitseminen tunkeutumisen havaitsemisjärjestelmän kouluttamiseksi ja testaamiseksi on ratkaiseva parametri, ja on selvää, että tietojoukot vaikuttavat tämän alan tutkimukseen, koska joidenkin niiden sisältöä

pidetään vanhentuneena tai ylimääräistä tietoa sisältävänä (Halbouni ym. 2022). Tietojoukon sisällöllinen laatu vaikuttaa viime kädessä jokaisen tunkeutumisen havaitsemisjärjestelmän luotettavuuteen (Moustafa ja Slay 2015). Koska koulutustietojoukko määrittää koneälypohjaisen tunkeutumisen havaitsemisjärjestelmän suorituskyvyn, on tärkeintä toteuttaa koulutustietojoukko, joka sisältää suuren määrän rikasta dataa verkkotunkeutumisesta mahdollisimman pienellä redundanssilla (Kim ja Pak 2022). tunkeutumisen havaitsemisjärjestelmän tehokkuutta arvioidaan niiden hyökkäysten tunnistamisen suorituskyvyn perusteella. Tämä vaatii kattavan tietojoukon, joka sisältää normaalia ja epänormaalia käyttäytymistä (Moustafa ja Slay 2015). Koska tietojoukkoja käytetään tunkeutumisen havaitsemisjärjestelmien toiminnan laadukkuuden evaluoimiseen, on luotettavan arvion saamiseksi myös tietojoukon oltava laadukas. Hyvä testaustietojoukko vastaa mahdollisimman läheisesti sen verkkoympäristön liikennettä, johon tunkeutumisen havaitsemisjärjestelmä on sijoitettu, jotta arviointi vastaisi reaalimaailman tilannetta.

Nykypäivänä hyvien ja kuvaavien tietojoukkojen puute haittaa tunkeutumisen havaitsemisen tutkimusta tai vähintäänkin vaikeuttaa merkittävästi tunkeutumisen havaitsemismenetelmien arviointia ja niiden suorituskyvyn vertailua (Małowidzki, Bereziński ja Mazur 2015). Hyvien kokonaisvaltaisten tietojoukkojen puute on johtanut tilanteeseen, jossa aihealueen tutkimus ja kehitys joutuu tyytymään suhteellisen vanhoihin sekä laadultaan vajavaisiin tietojoukkoihin. Toisaalta, Shiravi ym. (2012) mukaan täydellisen tietojoukon luominen on itsessään mahdotonta, mutta suunnittelemalla systemaattinen lähestymistapa tietojoukon luomiseen, voidaan mahdollisesti luoda erilaisia riittäviä tietojoukkoja eri tilanteisiin, ja jos se tehdään oikein, se vähentää tehokkaasti täydellisten tietojoukkojen tarvetta.

Jo julkaistut tietojoukot ovat luonteeltaan staattisia, eli niihin ei enää julkaisun jälkeen voida lisätä uutta dataa. Staattiset tietojoukot eivät näin ollen pysty mukautumaan kehittyvään verkkoliikenteeseen, sillä tietojoukon sisältö ei voi muuttua kuvaamaan uudentyyppistä verkkoliikennettä. Staattisen tietojoukon ei ole siis mahdollista pysyä hyvänä, vaikka se julkaisunsa ajankohtana olisi sitä ollutkin. Małowidzki, Bereziński ja Mazur (2015) ehdottavatkin yhdeksi hyvän tietojoukon ominaisuudeksi mahdollisuutta täydentää tietojoukkoa uudenlaisella haittaliikenteellä. Tietojoukkojen dynaamisen luonteen puolesta puhuvat myös Gharib ym. (2016), joiden mukaan staattisista tietojoukoista pois siirtyminen on edellytys

muunnettavissa, laajennettavissa ja toistettavissa olevien sekä moderneja liikennemalleja kuvaavien tietojoukkojen luomiselle.

Tietojoukko koostuu kahdesta liikennedatakomponentista: normaalia liikennettä kuvaavasta ja haitallista liikennettä eli poikkeamia kuvaavasta Małowidzki, Bereziński ja Mazur (2015). Tietoliikennettä tallennetaan yleensä joko pakettipohjaisessa tai flow-pohjaisessa mallissa. Pakettipohjaiset tiedot sisältävät täydelliset hyötykuormatiedot ja sitä tallennetaan useimmiten pakettipohjaisessa formaatissa ja ne sisältävät koko hyötykuorman, kun taas flow-pohjaiset tiedot ovat aggregoidumpia ja sisältävät yleensä vain metatietoja verkkoyhteyksistä tai sessioista, eivätkä ne yleensä sisällä varsinaista hyötykuormaa (Ring ym. 2019). Flow-pohjaiset tiedot kuvaavat yhtä sessiota, joka koostuu yhdestä tai useammasta paketista, jotka liittyvät samaan yhteyteen ja toimintoon. Se on sarja paketteja kahden verkossa olevan pisteen välillä, jotka jakavat tietyt yhteiset attribuutit tietyn aikaikkunan sisällä (Sarhan ym. 2021). Pakettipohjainen tietojoukko koostuu nimensä mukaisesti paketeista, jotka ovat pakettikytkentäisessä verkossa liikkuvan datan perusyksikkö.

Tietojoukkoja on tutkimuksessa luokiteltu useisiin eri kategorioihin niiden ominaisuuksien perusteella. Tunkeutumisen havaitsemisjärjestelmien tutkimuksessa käytettävillä tietojoukoilla on useita eri malleja ja ne voivat olla keskenään hyvinkin erilaisia. Ne kuitenkin sisältävät usein tiettyjä yhteisiä muuttujia, joiden avulla niitä on pystytty kategorisoimaan. Tapa, jolla tietojoukko on muodostettu jakaa tietojoukot sekä synteettisiin että realistisiin tietojoukkoihin. Synteettiset tietojoukot luodaan tutkimuksen generoimasta verkkoliikenteestä vastaamaan tiettyjä tarpeita, ehtoja tai testejä, jotka tietojoukon tulee täyttää, kun taas realistiset tietojoukot luodaan seuraamalla todellista verkkoliikennettä realistisella tavalla, kuten esimerkiksi jonkin organisaation päivittäistä verkkoliikennettä (Bhuyan, Bhattacharyya ja Kalita 2014). Małowidzkin, Berezińskin ja Mazurin (2015) mukaan todellinen liikenne voidaan havaita tuotantoverkossa tai laboratoriossa, eli erityisesti valmistetussa ympäristössä haitallisten sovellusten suorittamiseen ja tarkkailuun. Erona on, että jälkimmäisessä tapauksessa verkkotopologia ja konfiguraatio ovat yleensä spesifisti luotu tietylle tarkoitukselle, mikä voi johtaa erilaiseen käyttäytymiseen ja sen seurauksena erilaiseen havaittuun liikenteeseen. Realistisella tietojoukolla pystytään tutkimuksessa jäljittelemään reaalimaailman verkkoliikennettä erittäin tarkasti. Tietojoukon synteettisyys mahdollistaa toisaalta

tietojoukon sisällön optimoinnin suoritettavaa tehtävää varten, mutta samalla menetetään testauksen realistisuutta ja saatetaan joko tahallisesti tai tahattomasti vaikuttaa tietojoukon sisällön laatuun. Muita tärkeimpiä luokittelutapoja tietojoukoille on sen tasapainoisuus, eli normaalin ja haittaliikenteen suhde (Sun ym. 2020), tietojoukon sisältämät hyökkäystyypit, tietojoukon sisältämien instanssien määrä sekä muoto, tietojoukon saatavuus ja tietojoukon ikä. Lisäksi jokainen tietojoukko sisältää sille uniikin määrän erilaisia ominaisuuksia, jotka kyseisen tietojoukon kokoajat ovat siihen sisällyttäneet.

Hyvin harvat tutkimukset käyttävät ei-julkisia tai omia aineistojaan, joka osoittaa, että nämä julkiset tietojoukot tunnistetaan tunkeutumisen havainnoinnin vakiotietojoukoiksi (Tsai ym. 2009). Amarudin, Ferdiana ja Widyawan (2020) esittivät tutkimuksissa käytettyjen tietojoukkojen suhteen olevan julkisia tietojoukkoja 79 % ja yksityisiä tietojoukkoja 21 %. Toisaalta etenkin yksityisen ja kaupallisen puolen tutkimus saattaa käyttää julkista tutkimusta enemmän omia, ei-julkisia tietojoukkojaan, pitääkseen tuotekehitys prosessinsa salaisena. Ei-julkisen tietojoukon käyttöä voidaan perustella sekä kaupallisesta että tietoturvallisesta näkökulmasta. Khraisat ym. (2019) korostavat etenkin kaupallisten tuotteiden käyttämien tietojoukkojen saatavuuden hankaluutta tietosuojaongelmien vuoksi. Aiemmista tutkimuksista käy ilmi tutkimuksen keskittyminen suureksi osaksi vain tiettyjen tietojoukkojen ympärille Tunnettujen ja julkisesti saatavilla olevien tietojoukkojen lisäksi on olemassa yksityisiä tietojoukkoja, joiden sisältö ja ominaisuudet ovat kaupallisista tai tietoturvasyistä ainoastaan tietojoukot omistavien tahojen hallinnassa. Näitä tietojoukkoja on sivuttu yksittäisissä tutkimuksissa, mutta tämän tutkimuksen osalta ei ole relevanttia kiinnittää näihin tietojoukkoihin tai niistä tehtyihin tutkimuksiin huomiota.

Julkisesti saatavilla olevista tietojoukoista Kumarin ym. (2021) mukaan KDD-99 tietojoukko on selkeästi tutkimuksissa eniten käytetty ja toiseksi käytetyin on KDD-99 tietojoukosta johdettu NSL-KDD. Muita tutkimuksissa hyödynnetyimpiä tietojoukkoja ovat UNSW-NB15, DARPA98 ja Kyoto2006+. Uudemmissa tutkimuksissa huomioita ovat saaneet erityisesti uudet CIC-IDS2017 ja CSE-CIC-IDS2018 tietojoukot, jotka ovat heti julkaisujensa jälkeen alkaneet houkutella tutkijoita analysoimaan niitä ja kehittämään omia mallejaan niiden pohjalta (Ghurab ym. 2021). Hindy ym. (2020) mukaan eri kuusi tutkimuksessa eniten käytettyä julkista tietojoukkoa ja niiden käytön prosentuaaliset osuudet kaikesta

aihealueen tutkimuksesta ovat; KDD-99 50.5 %, NSL-KDD 17.2 %, Kyoto2006+ 3.0 %, unsw-nb15 3.0 %, DARPA (yhteenlaskettu tietojoukon eri versiot; DARPA 1998, DARPA 1999 ja DARPA 2000) 6.0 % ja CIC-IDS2017 3.0 %. KDD-99 tietojoukon massiivinen yliedustus aihealueen tutkimuksessa saattaa indikoida, etteivät uudemmat tietojoukot ole pystyneet tuottamaan merkittävästi laadukkaampaa vaihtoehtoa. Yhden tietojoukon yliedustus tutkimuksessa saattaa olla merkki myös siitä, että samaa tietojoukkoa sovelletaan todennäköisesti hyvin erilaisiin toteutuksiin ja verkkoympäristöihin, jolloin saadut tulokset saattavat olla myös hyvin erilaisia. Divekarin ym. (2018) mukaan ihannetapauksessa tietojoukko olisi aina verkkoympäristökohtainen. Vaihtoehtojen puute on kuitenkin johtanut useisiin tutkimuksiin, joissa keskitytään KDD-99 tietojoukon käyttöön benchmark tietojoukkona, vaikka tietojoukon valinta on olennainen osa nykyaikaisen koneoppimistekniikoita käyttävän tunkeutumisen havaitsemisjärjestelmän turvallisuutta. Olisi kuitenkin erittäin haastavaa ja luoda jokaiseen tilanteeseen oma sopiva tietojoukkonsa. Ei ole olemassa vakiintunutta määritelmää sille, mitä ominaisuuksia tietojoukon tulisi sisältää, jonka seurauksena jokainen käytettävissä oleva tietojoukko on luotu omalla ainutlaatuisella joukolla verkko-ominaisuuksia (Sarhan ym. 2021).

5.1 KDD-99

KDD Cup 1999, eli KDD-99 -tietojoukko luotiin suodattamalla vuoden 1998 DARPA-tietojoukon pakettikaappausta käytettäväksi International Knowledge Discovery and Data Mining Tools Competition -kilpailussa (Sharafaldin ym. 2017). Raakadata, josta tietojoukko on muodostettu sisältää neljä gigatavua pakattua binääristä tcpdump-dataa seitsemän viikon aikana kerätystä verkkoliikenteestä (Sahu, Sarangi ja Jena 2014). Tietojoukko luotiin vuonna 1999, jonka jälkeen siitä on tullut laajimmin käytetty tietojoukko anomaliapohjaisen tunkeutumisenhavaitsemisen tutkimuksessa (Ghurab ym. 2021).

KDD-99 tietojoukko koostuu 4 898 431 yksittäisestä yhteysinstanssista, joista jokainen sisältää 41 ominaisuutta, ja jotka ovat merkitty joko normaaliksi tai hyökkäykseksi, joissa hyökkäykseksi merkitty sisältää täsmälleen yhden tietyn hyökkäystyyppi (Ghurab ym. 2021).Tietojoukon sisältämät hyökkäyksiksi luokitellut yhteydet kuuluvat johonkin seuraavista neljästä kategoriasta: Denial of Service Attack (DoS), User to Root Attack (U2R),

Remote to Local Attack (R2L) tai Probing Attack (Tavallaee ym. 2009). Hyökkäysten ja normaalin liikenteen jakautuminen tietojoukossa on nähtävissä Taulukko 1. Tietojoukkoon sisältyvät 41 ominaisuutta on luokiteltu seuraaviin kolmeen luokkaan: 1) perusominaisuudet, 2) liikenneominaisuudet ja 3) sisältöominaisuudet. Perusominaisuudet saadaan TCP/IP-yhteydestä. Liikenneominaisuudet on jaettu kahteen ryhmään isännän (*eng. host*) tai palvelun (*eng. service*) mukaisesti. Sisältöominaisuudet koskevat epäilyttävää toimintaa tiedoissa (Ferrag ym. 2020). Yhteysinstanssitasolla tietojoukko sisältää perusattribuutteja TCP-yhteyksistä ja korkean tason attribuutteja, kuten epäonnistuneiden kirjautumisten lukumäärän, mutta ei IP-osoitteita (Ring ym. 2019).

	Training set	Prosenttiosuus	Test set	Prosenttiosuus
Normal	972 781	19.85 %	60 593	19.48 %
DoS	3 883 390	79.27 %	231 455	74.41 %
Probe	41 102	0.83 %	4 166	1.33 %
R2L	1 106	0.02 %	14 570	4.68 %
U2R	52	0.001 %	245	0.07 %
Yhteensä	4 898 431	100 %	311 029	100 %

Taulukko 1. Sisällön jakautuminen KDD-99 tietojoukossa (Dina ja Manivannan 2021)

KDD-99 tietojoukon ajantasaisuus on tunnetun ja laajasti käytetyn tietojoukon suurin ongelma. Yli 20 vuotta vanhan tietoliikenneinformaation kyky vastata nykypäivän verkkoliikenteen ominaisuuksia ja kuvata alati kehittyviä tunkeutumismetodeja on kyseenalaistettava. Toinen laajasti tunnistettu ongelma on tietojoukon sisältämä redundanttien instanssien määrä. Ghurab ym. (2021) mukaan analyysi KDD-99 tietojoukon koulutus- ja testaustietojoukosta paljasti, että noin 78 % ja 75 % verkkopaketeista toistuu sekä koulutus- että testijoukkojen sisällössä.

5.2 DARPA98

MIT Lincoln Laboratory loi ensimmäisen tunkeutumisen havaitsemistietojoukon vuonna 1998, ja se nimettiin DARPAksi DARPA-rahoitteisen projektin puitteissa (Thakkar ja Lohiya 2020). Cunningham ym. (1999) muodostivat tietojoukon simuloimalla ilmavoimien tukikohdan verkkoa laboratorioympäristössä. He loivat sekä normaalia että hyökkäysliikennettä ympäristöönsä synteettisesti. Harjoitustietojoukko muodostui tämän simuloidun verkkoliikenteen kaappaamisesta kuuden viikon ajalta ja testitietojoukko kahden viikon kaappauksen ajalta. DARPA tietojoukoista on ilmestynyt myöhempinä vuosina uusia tietojoukkoja, jotka ovat olleet vanhan tietojoukon paranneltuja versioita, mutta sisällöltään muuten hyvin samankaltaisia. DARPA98 tietojoukko on toiminut useiden muiden tietojoukkojen perustana ja esimerkiksi laajasti käytetyt KDD-99 sekä NSL-KDD tietojoukot perustuvat suoraan DARPA98:n sisältämään verkkoliikennedataan. DARPA98 oli ensimmäinen rakentava yritys tunkeutumisen havaitsemiseen käytettävän tietojoukon luomiseksi (Haider ym. 2017)

DARPA98 on pakettimuotoista raakadataa ja tietojoukko on synteettisesti muodostettu. Synteettisyytensä takia tietojoukko ei edusta todellista verkkoliikennettä ja sisältää epäsäännöllisyyksiä, kuten väärien positiivisten (*eng. false positive*) tulosten puuttumista (Sharafaldin, Habibi Lashkari, and Ghorbani 2018). Normaali liikenne tietojoukossa koostuu toiminnoista, kuten sähköpostin lähettämisen ja vastaanottamisen, verkkosivustojen selaamisen, tiedostojen lähettämisen ja vastaanottamisen FTP:llä, telnetin käyttöön, IRC-viestimiseen sekä reitittimen etävalvontaan SNMP:n avulla. Haitallinen liikenne puolestaan koostuu hyökkäyksistä kuten, DoS, guess password, buffer overflow, remote FTP, syn flood, Nmap, and rootkit. (Sharafaldin, Habibi Lashkari ja Ghorbani 2018). Kerätyt verkkopaketit muodostivat yhteensä noin neljä gigatavua sisältäen noin 4 900 000 tietuetta. Kahden viikon testitiedoissa oli noin 2 miljoonaa yhteystietuetta, joista jokaisessa oli 41 ominaisuutta ja ne luokiteltiin normaaliksi tai haitalliseksi. (Khraisat ym. 2019). Vaikka DARPA tietojoukkoa ja sen johdonnaisia on pidetty tunkeutumisen havaitsemisjärjestelmissä tietojoukkojen kultastandardina, se on vanhentunut reaalimaailman verkkojen normaalin liikenteen ja hyökkäyskäyttäytymisen kannalta (Haider ym. 2017).

5.3 NSL-KDD

Tavallaee ym. (2009) kehittivät NSL-KDD tietojoukon alun perin vastaamaan KDD-99 tietojoukkoon kohdistunutta kritiikkiä ja pyrki ainakin osittain ratkaisemaan KKD-99 tietojoukon suurimpia ongelmia. NSL-KDD tietojoukko on sisällöltään johdettu suoraan KDD-99 tietojoukon sisällöstä valitsemalla mukaan vain tietyt KDD-99 tietojoukon tietueet ja se voidaan mieltää parannelluksi versioksi edeltäjästään. Koska NSL-KDD-tietojoukon rakenne on pohjimmiltaan sama kuin KDD-99 -tietojoukon, myös NSL-KDD sisältää samat hyökkäysmallit ja normaalin liikenteen ja samat 41 ominaisuutta (Shone ym. 2018). NSL-KDD tietojoukon hyökkäystyyppien ja normaalin liikenteen jakautuminen on esitelty Taulukko 2.

	Koulutusjoukko	Prosentti-osuus	Testausjoukko	Prosenttiosuus
Normal	67 342	53.458 %	9 710	43.075 %
DoS	45 927	36.458 %	7 457	33.080 %
Probe	11 656	9.253 %	2 421	10.740 %
R2L	995	0.790 %	2 754	12.217 %
U2R	52	0.041 %	200	0.887 %
Yhteensä	125 972	100 %	22 542	100 %

Taulukko 2. Sisällön jakautuminen NSL-KDD tietojoukossa (Divekar ym. 2018)

Verrattuna KDD-99 tietojoukkoon, NSL-KDD:n käytöllä on tutkimuksessa saavutettu huomattavasti parempia tuloksia, koska aineisto ei sisällä redundantteja tietueita ja tietueiden jakautuminen on tasapuolisempaa (Al-Daweri ym. 2020). Ferrag ym. (2020) esittävät NSL-KDD tietojoukon sisältävän seuraavat parannukset aiempiin tietojoukkoihin verrattuna: Redundantit instanssit ovat poistettu, joten tietojoukko ei sisällä enää ylimääräisiä instansseja sekä tietojoukkoon valittujen tietueiden lukumäärä on järjestetty prosenttiosuutena alkuperäisistä tietueista, joten tietojoukon sisältämien tietueiden määrä on kohtuullinen.

5.4 Kyoto2006+

Kyoto2006+ perustuu kolmen vuoden ajalta kerättyihin realistisiin liikennetietoihin (marraskuu 2006 - elokuu 2009) ja se sisältää noin 93 miljoonaa sessiota. Jokainen sessio koostuu 14 tilastollisesta ominaisuudesta, jotka on johdettu KDD-99 tietojoukosta sekä 10 lisäominaisuudesta, joita voidaan käyttää lisäanalyyseihin ja verkkopohjaisten tunkeutumisen havainnointijärjestelmien arviointiin. (Song ym. 2011). Tietojoukon tarjoamat 10 lisäominaisuutta mahdollistavat liikenteen tarkemman tarkastelun paremman ymmärryksen saamiseksi tietoverkosta.

Session kuvaus	Sessioiden määrä	Sessioiden prosentuaalinen määrä
Tuntematon hyökkäys	425 719	0.46 %
Tunnettu hyökkäys	42 617 536	45.79 %
Normaali	50 033 015	53.75 %
Yhteensä	93 076 270	100 %

Taulukko 3. Sessioiden jakautuminen Kyoto2006+ tietojoukossa (Song ym. 2011)

Kyoto2006+ tietojoukon suurin eroavaisuus muihin nähden on, että se ei ole synteettisesti luotu tietojoukko, vaan se sisältää aitoa reaalimaailmasta kerättyä tietoliikenneinformaatiota vastaten näin täysin reaalimaailman liikenteen ominaisuuksia tiettynä ajanhetkenä. Se on koottu käyttämällä honeypotteja, darknet-antureita, sähköpostipalvelimia ja web-indeksointirobottia (Verma, Bhandari ja Singh 2020). Tietojoukko ei koostu puhtaasti raa'asta tietoliikennedatasta, vaan BroIDS (*nyk. Zeek*) ohjelmaa, joka on avoimen lähdekoodin verkkoliikenteen analysaattori, käytettiin muuntamaan raaka pakettimuotoinen tietoliikennedata sessiokohtaiseksi dataksi (Song ym. 2011).

Koska tietojoukko on koostettu honeypot-palvelimiin kohdistuvasta oikeasta tietoliikenteestä, ei sille ole voitu tehdä samanlaista manuaalista merkitsemistä ja anonymisaatiota, kuten synteettisesti koostetuille tietojoukoille. (Ghurab ym. 2021). Manuaalisella

merkitsemisellä tarkoitetaan yksittäisen yhteysinstanssin merkitsemistä, joko normaaliksi tai haitalliseksi yhteysinstanssiksi. Manuaalisen luokittelun sijaan dataa kerätessä honeypot-palvelimilla toimineet tunkeutumisen havaitsemisjärjestelmä ja antivirus ohjelmat merkitsivät liikenteen joko normaaliksi tai haitalliseksi (Song ym. 2011). Liikenteen jakautuminen on esitetty Taulukko 3.

5.5 unsw-nb15

Vastatakseen KDD-99 ja NSL-KDD-tietojoukoista tunnistettuihin ongelmiin sekä benchmark-tietojoukkojen saatavuudellisiin haasteisiin Moustafa ja Slay (2015) esittelivät uuden hybridin tietojoukon UNSW-NB15, joka yhdistää sekä todellista liikennettä että synteettisesti luotua hyökkäystoimintaliikennettä. UNSW-NB15 on varsin uusi tietojoukko, joka on vielä viime vuosiinkin asti nähty modernin tietoliikennekäyttäytymisen tarjoajana (Ghurab ym. 2021). Tietojoukko on muodostettu synteettisessä ympäristössä UNSW:n kyberturvallisuuslaboratoriossa hyödyntämällä IXIA verkkoliikennegeneraattoria, jolla on generoitu sekä normaalia liikennettä että hyökkäyksiä kuvaavaa liikennettä. Hyökkäysliikenteen mallit on johdettu *common vulnerability exposures (CVE)* informaatiota kokoavalta sivulta. (Vinayakumar ym. 2019.) Yleisesti tunnettujen CVE tietojen hyödyntäminen synteettisesti generoidussa tietojoukossa takaa, että tietojoukon sisältämä hyökkäysdata on varmasti relevanttia verrattuna reaalimaailman hyökkäysliikenteeseen.

Yhteyden kategoria	Yhteyksien määrä	Yhteyksien % määrä
Normaali	2 218 761	87.35 %
Fuzzers	24 246	0.95 %
Reconnaissance	13 987	0.55 %
Shellcode	1 511	0.06 %
Analysis	2 677	0.11 %
Backdoors	2 329	0.09 %

DoS	16 353	0.64 %
Exploits	44 525	1.75 %
Generic	215 481	8.48 %
Worms	174	0.01 %
Yhteensä	2 540 044	100 %

Taulukko 4. Sisällön jakautuminen UNSW-NB15 tietojoukossa (Moustafa ja Slay 2015)

Tietojoukossa on yhteensä 2 540 044 nimettyä yhteystietuetta, joista jokainen on merkitty joko normaaliksi tai hyökkäykseksi. Normaalien yhteysinstanssien koko edustaa 87 % (2 218 761 instanssia) tietojoukon koosta, kun taas hyökkäykseksi kategorisoitujen yhteysinstanssien 13 % (321 283 tietuetta) (Al-Daweri ym. 2020). Yhteysinstanssien tarkempi jakautuminen eri hyökkäyskategorioihin ja niiden osuuksiin tietojoukossa on nähtävissä Taulukko 4. Normaalin ja hyökkäysliikenteen jakautuminen osoittaa tietojoukon olevan epätasapainoinen. Tietojoukko on saatavilla sekä kokonaisena että pienempänä osajoukkona, joka on jaettu koulutus- (82 332 tietuetta) ja testausjoukkoihin (175 341 tietuetta) (Vinayakumar ym. 2019).

5.6 CIC-IDS2017

Canadian Institute for Cybersecurityn Sharafaldin, Habibi Lashkari ja Ghorbani (2018) kehittivät CIC-IDS2017-tietojoukon, sillä heidän mukaansa aiemmin kehitetyt tietojoukot ovat sekä vanhentuneita että epäluotettavia käyttää. Sharafaldin, Habibi Lashkari ja Ghorbani (2018) käyttivät CIC-IDS2017-tietojoukon suunnittelussa hyväkseen Gharibin ym. (2016) luomaa viitekehystä, jonka 11 ominaisuutta ovat kriittisiä kattavan ja kelvollisen IDS-tietojoukon kannalta. Tähän viitekehykseen kuuluvat ominaisuudet ovat täydellinen verkkokonfiguraatio (*eng. Complete Network configuration*), liikenteen kokonaisvaltainen kuvaus (*eng. Complete Traffic*), tietojoukon merkitseminen (*eng. Labeled dataset*), koko vuorovaikutuksen kuvaus (*eng. Complete Interaction*), kaiken liikenteen kuvaaminen (*eng. Complete*

Capture), protokollien monimuotoisuus (*eng. Available Protocols*), hyökkäysten monimuotoisuus (*eng. Attack Diversity*), anonymiteetti, heterogeenisyys, johdetut ominaisuudet (*eng. Feature Set*) sekä metatiedot.

CIC-IDS2017 on synteettisesti koostettu tietojoukko. Synteettisyydestään huolimatta tietojoukko on mielletty hyvin realistiseksi. Realistisuutta saatiin aikaa käyttämällä A- ja B-profiilijärjestelmää normaalin taustaliikenteen ja hyökkäysliikenteen luomisessa. (Maseer ym. 2021) B-profiilijärjestelmä profiloi ihmisten käyttäytymismalleja ja generoi naturalistista, normaaliksi kategorisoitavaa liikennettä (Azzaoui ja Boukhamla 2020). Yhteysinstanssien kokonaismäärä tietojoukossa on yhteensä 2 830 108 instanssia, josta normaalia liikennettä on 83,3 % (2 358 036 instanssia), kun taas hyökkäykseksi kategorisoitavaa liikennettä on 16,7 % (471 454 instanssia) (Abdulhammed ym. 2019). Tietojoukon sisällön ja sen sisäisten luokittelujen määrät ja osuudet koko tietojoukossa on nähtävissä Taulukko 5. Tietojoukko on luokittelultaan erittäin epätasapainoinen, joka saattaa vaikuttaa tietojoukkoa käyttävän järjestelmän luokittelutoiminnon puolueellisuuteen (Maseer ym. 2021).

Yhteyden kategoria	Yhteyksien määrä	Yhteyksien % määrä
Normaali	2 358 036	83,34 %
DoS Hulk	231 073	8,17 %
Port Scan	158 930	5,62 %
DDoS	41 835	1,48 %
DoS GoldenEye	10 293	0,36 %
FTP Patator	7 938	0,28 %
SSH Patator	5 897	0,21 %
DoS Slow Loris	5 796	0,20 %
DoS Slow HTTP Test	5 499	0,19 %

Botnet	1 966	0,07 %
Web Attack: Brute Force	1 507	0,05 %
Web Attack: XSS	625	0,02 %
Infiltration	36	0,001 %
Web Attack: SQL Injection	21	0,0007 %
HeartBleed	11	0,0004 %
Yhteensä	2 829 463	100 %

Taulukko 5. Sisällön jakautuminen CIC-IDS2017-tietojoukossa (Abdulhammed ym. 2019)

CIC-IDS2017-tietojoukko on saanut osakseen myös kritiikkiä ja siitä on löydetty parannusta vaativia ominaisuuksia. Panigrahi ja Borah (2018) mainitsevat tietojoukon merkittävimmiksi puutteiksi datan hajanaisuuden, datan valtavan määrän, puutteellisen merkitsemisen sekä luokittelun korkean epätasapainoisuuden. Azzaoui ja Boukhamla (2020) puolestaan huomauttavat, että ylimääräisistä tietueista, epäolennaisista ominaisuuksista, tyhjistä tai tuntemattomista arvoista johtuen, CIC-IDS2017-tietojoukon datan esikäsittely on tarpeellista, jos sitä aiotaan käyttää.

6 Ratkaisun innovointi/rakennus

Tietojoukkojen ominaisuuksien analysoiminen on aiemmassa tutkimuksessa tunnistettu tutkimuksen kannalta erittäin tärkeäksi, mutta toteutukseltaan monimutkaiseksi ja haasteelliseksi. Jotta eri tunkeutumisen havaitsemistietojoukkoja voitaisiin vertailla rinnakkain ja auttaa tutkijoita löytämään sopivia tietojoukkoja omiin spesifeihin arviointiskenaarioihinsa, on tarpeen määritellä tietojoukkojen yhteiset ominaisuudet arvioinnin perustaksi (Ring ym. 2019). Tietojoukkojen ominaisuuksien arvioiminen luo perustan saatavilla olevien tietojoukkojen vertailulle ja tiettyihin tutkimuksiin ja sovellutuksiin sopivien tietojoukkojen oikeelliselle tunnistamiselle. Käytettävissä olevissa tietojoukoissa on näin ollen oltava yhteisiä ominaisuuksia, jotta voidaan ymmärtää erilaisten tunkeutumisen havainnointitietotietojoukkojen luotettavuus ja tunnistaa sopivat tietojoukot käytettäväksi tietyssä arviointiskenaariossa (Hnamte ja Hussain 2021). Tällä perusteella tietojoukkojen vertailu ja sen kautta sopivimman tietojoukon valinta voidaan suorittaa valitsemalla nämä yhteiset ja tärkeät ominaisuudet ja muuttamalla ne keskenään vertailukelpoiseen muotoon. tietojoukkojen sisäisten ominaisuuksien eroavaisuudet saattavat johtua eri tietojoukkojen eri käyttötarkoituksista ja niitä ei aina ole luotu samoja kriteerejä silmällä pitäen. Tästä syystä tietojoukkojen eri ominaisuudet saavat hyvin erilaisia painoarvoja eri tietojoukkojen kesken, joka vaikeuttaa oikeellisten mittareiden muodostamista kaikki tietojoukot huomioiden. Tässä tutkimuksessa luotavan mallin pääajatus on tarjota malli, jonka avulla vertailulle relevantit ominaisuudet voidaan muuttaa vertailtavaan muotoon. Malli tarjoaa myös tavan vertailla ominaisuuksia keskenään ja tavan soveltaa sitä tietojoukon valintatilanteessa.

Eri tietojoukkojen ominaisuuksien vertailuun ja tietojoukkojen evaluointiin on käytetty ja esitelty useita erilaisia metodeja sekä kriteerejä. Nämä on useissa tutkimuksissa kuitenkin sovitettu yksittäisen tutkimuksen tarpeisiin ja niitä analysoidaan vain suhteessa juuri sillä hetkellä tehtävään tutkimukseen. Aiemmista tutkimuksista löytyy kuitenkin muutamia tutkimuksia, jotka pyrkivät luomaan yleiskäyttöisiä vertailumittareita, joilla voidaan arvioida eri tietojoukkojen ominaisuuksia. Ring ym. (2019) ehdottavat, että tulisi tietojoukoissa esiintyviä ominaisuuksia tulisi yhtenäistää ja yleistää sen sijaan, että ottaisi niitä kaikkia yksittäin huomioon vertailua suunniteltaessa. Hnamte ja Hussain (2021) mukaan, jotta voidaan arvioida erilaisia tunkeutumisen havaitsemisjärjestelmien tietojoukkoja rinnakkain ja löytää

niistä yhtäläisyyksiä tietyssä arviointitilanteessa, on tärkeää luoda yhteisiä ominaisuuksia arvioinnin pohjaksi. Tästä syystä malliin valitut mittarit mittaavat vain attribuutteja, jotka ovat läsnä jokaisessa vertailun kohteena olevassa tietojoukossa.

Hyvän tietojoukon määritelmä on usein riippuvainen tutkittavasta näkökulmasta, mutta on esitetty muutamia, jokaisella ”hyvällä tietojoukolla” esiintyviä yhteisiä piirteitä. Nämä piirteet ovat läsnä useimmissa tietojoukkojen laatuun keskittyvissä tutkimuksissa, jonka takia näistä oli luontaista johtaa tässäkin tutkimuksessa käytetyt vertailumallit, sillä myös tämän tutkimuksen mallin perimmäisenä tarkoituksena on löytää tiettyyn tarkoitukseen sopivin ja paras tietojoukko. Hyvän tietojoukon määrittelyä selvittävässä tutkimuksessaan Małowidzki, Bereziński ja Mazur (2015) nostavat esiin tietojoukon ajantasaisuuden sekä tasapainosuhteen yksinä merkittävimmistä laatuun vaikuttavista tekijöistä. Ajantasaisuudella tarkoitetaan käytännössä vain tietojoukon keräämisen ajankohtaa, jonka mukaan, mitä uudempi tietojoukko on, sitä paremmin se kuvaa nykypäivän verkkoliikenteen skenaarioita (Ghurab ym. 2021). Shiravin ym. (2012) mukaan on välttämätöntä, että tietojoukon sisältämä liikenne näyttää ja käyttäytyy mahdollisimman realistisesti, jolloin hyvän tietojoukon on oltava mahdollisimman realistinen. Useat tutkimukset, kuten (Ring ym. 2019), (Gharib ym. 2016) ja (Gumusbas ym. 2021) alleviivaavat tietojoukon sisällöllisiä ominaisuuksia, jotka kuvaavat tietojoukon sisältämien yhteysinstanssien kuvausta sekä muodostumista, niin kerätystä datasta kuin datan keräystavoistakin johtuen. Nämä sisällölliset ominaisuudet määrittelevät tietojoukon kuvaavuutta ja joko rajoittavat tai parantavat datan perusteella tapahtuvaa oppimista. Edellä mainittujen tutkimusten perusteella vertailumallin mittareiden mittaamat ominaisuudet ovat: tasapainoisuus, realismin taso, ajankohtaisuus ja sisäiset ominaisuudet. Näitä mittaamalla saadaan luotettava kuvaus tietojoukon laadusta ja käyttökelpoisuudesta.

Vertailumalli muodostettiin entuudestaan tunnettuja mittareita ja arviointiviitekehyksiä yhdistelemällä ja muodostamalla näiden pohjalta yksittäinen ja yhtenäinen malli, joka ottaa huomioon kaikki tietojoukon laatuun ja sopivuuteen merkittävästi vaikuttavat tekijät ja attribuutit. Arviointiviitekehyksien tuottamat numeeriset ja vertailukelpoiset arvot normalisointiin samalle asteikolla, joka mahdollistaa helpommin havainnollistettavan ja vertailua helpottavamman mallin. Vertailumalliin valitut evaluaatiomenetelmät ja arviointiviitekehykset valittiin aiemmissa tutkimuksissa tietojoukon tärkeimmiksi ominaisuuksiksi

valittujen ominaisuuksien perusteella. Jokainen vertailumalliin mukaan otettu mittari mittaa ”hyvän tietojoukon” edellyttämiä ominaisuuksia ja mittarit pyrittiin valitsemaan periaatteella, jonka mukaan mittarit eivät mittaisi keskenään päällekkäisiä asioita. Tällä tavalla estettiin mittauskohteiden redundanttisuus ja sitä kautta tietyn ominaisuuden tai ominaisuuksien painotusten ylikorostuminen.

6.1 Epätasapainosuhde

Aiemmassa tutkimuksessa usein esiintyvä ongelma on tietojoukkojen epätasapainoisuus. Arqane ym. (2021) mukaan tietojoukko voidaan määritellä epätasapainoiseksi, kun se sisältää epätasaisen jakauman näytteitä, esimerkiksi kun haitallisten ja normaalien näytteiden suhde on 1:10. Näytteellä tarkoitetaan yksittäistä tietojoukon jäsentä, joka saattaa olla yksittäinen paketti, yksittäinen flow tai jokin muu tietoliikennedatan muodostama yksikkö. Tietojoukkojen epätasapainoisuus on yksi tärkeistä ominaisuuksista, johon tulee kiinnittää huomiota, kun suoritetaan minkäänlaista vertailua eri tietojoukkojen kesken. Tietojoukon tasapainoisuus ei vastaa reaalimaailman tietoliikenteessä esiintyvää haitallisen ja normaalin tietoliikenteen jakaumaa, vaan sillä pyritään mahdollisimman tarkasti vaikuttamaan käsillä olevaan tunkeutumisenhavaitsemisjärjestelmään. Epätasapainoinen tietojoukko saattaa vääristää järjestelmän luokittelumekanismin käyttämiä tarkkuus- ja todennäköisyysmittauksia, joka vähentää järjestelmän tehokkuutta ja lisää sen keskimääräistä tarkkuutta (Cieslak, Chawla ja Striegel 2006). Karatas, Demir, ja Sahingoz (2020) esittävät, että lähes kaikki tietojoukot ovat epätasapainossa erilaisilla epätasapainosuhteilla ja että tätä epätasapainosuhdetta tulisi pienentää järjestelmän vääristymien minimoimiseksi.

Koska eri koneoppimismenetelmät myös reagoivat tietojoukkojen epätasapainoisuuksiin eri tavoin, voidaan epätasapainosuhteen tuntemisen avulla tehdä tehokkaampaa valintaa johonkin järjestelmään sopivasta tietojoukosta tai jollekin tietojoukolle sopivasta menetelmästä. Esimerkiksi Khoshgoftaar, Golawala ja Hulse (2007) huomasivat tutkimuksessaan, että satunnaismetsään perustuva menetelmä on huomattavasti muita testattuja menetelmiä parempi sietämään epätasapainoisia harjoitustietojoukkoja. Zhao, Zhang ja Li (2012) puolestaan havaitsivat, että epätasapainoisissa tietojoukoissa SVM-menetelmään perustuva luokittelu on pahasti vinoutunut kohti positiivista luokkaa, mikä johtaa suureen määrään vääriä

negatiivisia. Epätasapainoisuuden huomioiminen tietojoukon valinnassa edellyttää tapaa vertailla keskenään eri tietojoukkojen epätasapainoisuuksia, jotta tietojoukon valinta voisi olla perusteltu jollekin menetelmälle. Tässä ratkaisussa epätasapainoisuutta mitataan epätasapainosuhteella. Jonkin tietojoukon epätasapainosuhde voidaan määritellä enemmistöluokan esiintymien lukumäärän suhteeksi vähemmistöluokan esiintymien lukumäärään. Epätasapainosuhde noudattaa siis kaavaa $Epätasapainosuhde = \frac{enemmistöluokan\ esiintymät}{vähemmistöluokan\ esiintymät}$. (Abdulhammed ym. (2019). Tässä tapauksessa enemmistö- ja vähemmistöluokat kuvaavat tietojoukon normaalin ja hyökkäysliikenteen osuuksia. Epätasapainosuhteen avulla saadaan selvitettyä vakioitu arvo minkä tahansa tietojoukon epätasapainoisuudesta, kunhan sen sisältö on tunnettu ja luokiteltu. Esimerkiksi NSL-KDD-tietojoukossa on 67 342 normaalia instanssia ja 58 630 tunkeutumisinstanssia. NSL-KDD-tietojoukon epätasapainosuhteen arvo on siis 1,148. KDD-99 tietojoukossa arvo on puolestaan 4,045. Täysin tasapainoisessa tietojoukossa suhteen arvo on 1.

6.2 Realismi

Shiravin ym. (2012) mukaan, jotta saadaan selkein mahdollinen kuva verkon kautta tapahtuvien hyökkäysten todellisista ominaisuuksista, ei tietojoukossa saa olla ei-toivottuja ominaisuuksia verkon tai verkkoliikenteen kannalta. Tästä syystä on välttämätöntä, että sekä normaali että poikkeava liikenne tietojoukossa näyttää ja käyttäytyy mahdollisimman realistisesti. Tietoliikennettä sisältävän tietojoukon realistisuus ei ole itsessään vertailukelpoinen muuttuja, vaan tietojoukon realistisuus on tulkinnanvarainen ja useista eri tekijöistä, kuten liikenteen keräys- ja generointitavoista koostuva arvio. Tietojoukon realistisuus on kuitenkin erittäin tärkeä muuttuja järjestelmän suorituskyvyn tehokkuuden kannalta (Hnamte and Hussain 2021), joten tietojoukkoja valittaessa ja vertailtaessa on kyettävä ottamaan huomioon myös niiden realistisuuden taso. Haider ym. (2017) kehittivät Sugenon sumean päättelymallin (*eng. Sugeno Fuzzy Inference Systems*) (Sugeno and Yasukawa 1993) pohjalta IDS-tietojoukon realismin laadulle laskennallisen arviointimetriikan. Perustana Haiderin ym. (2017) esittämällä metriikalle on sumentaminen (*eng. fuzzification*), joka on prosessi, jolla kvantifioidaan kvalitatiivinen subjekti, tässä tapauksessa IDS-tietojoukon realismin laatu.

Haiderin ym. (2017) malli edellyttää kahta eri syötejoukkoa, jotka on koostettu joukoista erilaisia tietojoukon realistisuutta kuvaavia ominaisuuksia. Taulukko 6 syötejoukko X kuvaa mahdollisesti realistisen tietojoukon sisällöllisiä ominaisuuksia, jotka vaikuttavat realistisuuden arviointiin. Taulukko 7 kuvataan syötejoukko Y, joka esittää tietojoukon luomisympäristöön liittyvän muuttujan. Haider ym. (2017) antavat syötejoukoille määritelmät $X = \{x_1, x_2 \dots x_6\}$ ja $Y = \{y_1, y_2\}$, sekä näille joukolle syötejäsenyysfunktiot $F_1(x_k)$ ja $F_2(y_l)$. Jäsenyysfunktion $F_1(x_k)$ tehtävänä on antaa ennalta määrätylle syötteelle arvo, joka määräytyy sillä, että jos aineistolla on maksimirealismi, realismin todennäköisyys on 1 ja jokaisella joukon X jäsenellä on yhtä suuri, 1/6 (= *0.16*) osuus realismin aikaansaamisesta. Jäsenyysfunktion $F_2(y_l)$ tehtävänä on jälleen antaa ennalta määritetylle syötteelle arvo, joka syötejoukossa Y on reaalimaailman verkoissa generoiduille tietojoukoille 1 ja synteettisesti tai testiympäristössä generoidulle tietojoukolle 0.5. (Haider ym. 2017.) Kuten Haider ym. (2017) huomauttavatkin, voidaan syötejoukkojen sisältämien elementtien määrää laajentaa ja jäsenyysfunktioiden tuottamat arvot voidaan laskea tarpeen vaatiessa uudelleen. Aihealueen tutkimus ei kuitenkaan ole määritellyt luotettavia sisällöiltään laajempia tai vaihtoehtoisia syötejoukkoja, joten tässä tutkimuksessa käytetään vain Haiderin ym. 2017) tutkimuksen mukaisia syötejoukkoja. Sumentamiselle tyypillisellä tavalla voidaan myös tietojoukon realismia kuvaavalle kvantitatiiviselle arvolle antaa myös kielelliset termit, josta Haider ym. (2017) käyttämät arvot ovat nähtävissä Taulukko 8. Arvon ilmaisu kielellisin termein ei kuitenkaan ole vertailun kannalta pakollista, sillä keskinäiseen vertailuun riittää tieto siitä, että realismi kasvaa, kun lähestytään arvoa 1.

Ominaisuus	Kuvaus	$F_1(x_k)$
x_1	Verkkoliikenteen pakettien kokonainen kaappaus	0.16
x_2	Sisältää maksimimäärän mahdollisia hyökkäyksiä	0.16
x_3	Hyökkäyskäyttäytyminen nykyaikaista	0.16
x_4	Ajoitukset ja toimialat huomioiva reaalimaailman mukainen liikennedynamiikka	0.16

x_5	Kyberinfrastruktuurin suorituskyvyn ylläpito liikenteen kaappauksen aikana	0.16
x_6	Sisältää totuustiedot liikenteen merkintäprosessille	0.16

Taulukko 6. Syötejoukko X, eli sisällölliset ominaisuudet (Haider ym. 2017)

Ominaisuus	Kuvaus	$F_2(y_l)$
y_1	Tuotantoverkko tai muu todellinen verkko	1
y_2	Synteettinen verkko tai testialusta	0.5

Taulukko 7. Syötejoukko Y, eli luomisympäristön ominaisuudet ym. 2017)

Realismin arvon laskeminen aloitetaan tarkastelemalla, mitkä syötejoukon X ja syötejoukon Y ominaisuudet kuvaavat tarkastelun kohteena olevaa tietojoukkoa. Löydetyistä X ja Y joukon ominaisuuksista muodostetaan parit, joita kutsutaan säännöiksi, Sugenon sumeassa päättelymallissa säännön i, arvo z_i saadaan kaavalla 1, jossa $a = b = c = N$ ja $N =$ löydettyjen sääntöjen määrä. Säännön i arvon painotus w_i saadaan kaavalla 2, jonka jälkeen saatujen arvojen ja painotusten avulla voidaan laskea säännöille painotettu keskiarvo. Painotettu keskiarvo normalisoidaan välille $0 \leq R \leq 1$ jakamalla keskiarvo suurimmalla mahdollisella saavutettavalla realismin arvolla, joka on näillä muuttujilla 12.96. Painotetun keskiarvon laskeminen ja sen normalisointi lopullisen realismin arvoksi saadaan kaavalla 3. Näin tietojoukon realismin taso R, on kvantifioitu muuttujaksi, jonka arvo on $0 \leq R \leq 1$, jossa 0 on pienin mahdollinen realismin taso ja 1 suurin. (Haider ym. 2017).

$$z_i = a[F_1(x_k)] + b[F_2(y_1)] + c \quad (1)$$

$$w_i = AndMethod\ [F_1(x_k),\ F_2(y_1)] \quad (2)$$

$$R = \frac{\sum_{i=1}^{N} w_i z_i}{\sum_{i=1}^{N} w_i} \quad (3)$$

Realismin taso	Kielellinen termi realismin tasolle
$0 \leq R < 0.10$	Ei realistinen
$0.08 < R \leq 0.30$	Matalan tason realistisuus
$0.28 < R \leq 0.60$	Keskitason realistisuus
$0.58 < R \leq 0.97$	Keskikorkean tason realistisuus
$0.95 < R \leq 1$	Korkean tason realistisuus

Taulukko 8. Kielelliset ilmaukset realistisuuden eri tasoille (Haider ym. 2017)

Haiderin ym. (2017) malli kärsii kuitenkin tulkinnanvaraisuuden ongelmasta, jossa arvioinnin kohteena olevasta tietojoukosta syötejoukon X ominaisuuksien löytäminen jää arvioinnin tekijän tulkinnan vastuulle. Ominaisuuksien löytäminen on myös riippuvainen siitä, miten hyvin tietojoukon luomiseen ja datan keräämiseen liittyvät prosessit on dokumentoitu. Jos kaikkia tietojoukon luomiseen vaikuttaneita muuttujia ei ole dokumentoitu, on mahdollista, että realistisuuteen vaikuttavat ominaisuudet jäävät arvioinnin ulkopuolelle ja vaikuttavat näin realismia kuvaavan arvon luotettavuuteen.

6.3 Tietojoukon sisällöllinen laatu

Eri tietojoukkojen sisällölliset ominaisuudet ja niiden tavat kuvata liikennettä eroavat usein hyvin radikaalisti toisistaan. Sisällön kuvaamiseen ja mahdolliseen sisältöön vaikuttavat olennaisesti tiedon tallennus- ja esitysmuoto, tiedonkeruuympäristö sekä kerääjien valinta siitä, mitä he haluavat tietojoukossaan esittää. Tietojoukkojen sisäisten ominaisuuksien keskinäinen vertailu on erittäin tärkeää, jotta pystytään arvioimaan tietojoukon vaikutusta kehitettävään järjestelmään ja jotta on selkeää millaisia puutteita tai mahdollisuuksia mikäkin tietojoukko aiheuttaa.

Gharibin ym. (2016) malli luo tietojoukon ominaisuuksien kokoelman sekä numeerisen arvon kokoelmaan kuuluville ominaisuuksille tarkastelun kohteena olevien tietojoukkojen

vertailun perustaksi. Gharib ym. (2016) arvioivat tietojoukkoja 11 kriteeristä muodostuvan kehysmallin avulla. Tämän kehysmallin sisältävät kriteerit ovat; täydellinen verkkokonfiguraatio (*eng. Complete Network configuration*), liikenteen kokonaisvaltainen kuvaus (*eng. Complete Traffic*), tietojoukon merkitseminen (*eng. Labeled dataset*), koko vuorovaikutuksen kuvaus (*eng. Complete Interaction*), kaiken liikenteen kuvaaminen (*eng. Complete Capture*), protokollien monimuotoisuus (*eng. Available Protocols*), hyökkäysten monimuotoisuus (*eng. Attack Diversity*), anonymiteetti, heterogeenisyys, johdetut ominaisuudet (*eng. Feature Set*) sekä metatiedot. Täydellinen verkkokonfiguraatio kuvaa, että tietojoukon keräysympäristön tulisi koostua täydellisestä ja realistisesta verkkoympäristöstä, joka sisältää kaikki normaaleihin tuotantoverkkoihin kuuluvat laitteet. Liikenteen kokonaisvaltainen kuvaus tarkoittaa, että sisältääkö liikenteen kuvaus synteettistä liikennettä vai onko liikenteen kuvaus kokonaisvaltaisesti realistista. Tietojoukon merkitseminen tarkoittaa, että tietojoukon instanssit sisältävät merkinnän kyseisen instanssin kuulumisesta joko normaaliin tai hyökkäysliikenteeseen. Koko vuorovaikutuksen kuvauksella tarkoitetaan sekä verkon sisäisen että verkosta toiseen kulkevan liikenteen kuvaamista tietojoukon keräysympäristössä. Kaiken liikenteen kuvaaminen tarkoittaa nimensä mukaisesti, että tietojoukkoon on sisällytetty kaikki verkossa tapahtunut liikenne, eikä sitä ole millään tavalla karsittu. Protokollien ja hyökkäysten monimuotoisuus kuvaa vastaako tietojoukon sisältämä liikenne reaalimaailmassa esiintyviä protokollia sekä hyökkäyksiä. Anonymiteetillä tarkoitetaan sisältääkö tietojoukon instanssit sekä IP-osoitteet, että hyötykuorman. Heterogeenisissä tietojoukoissa instansseja on kerätty useammasta kuin yhdestä lähteestä. Ominaisuuksien johtaminen tarkoittaa onko tietojoukon instansseista johdettu erilaisia vertailukelpoisia ominaisuuksia, kuten esimerkiksi palvelut, portit sekä lähde ja kohde IP-osoitteet. (Gharib ym. 2016). Ring ym. (2019) kuvaavat omassa tutkimuksessaan hyvin samanlaiset kriteerit ja mallin tietojoukkojen sisäisten ominaisuuksien arvioinnille. Tämän mallin suurin eroavaisuus on kuitenkin, että se tarjoaa vain luettelon yksittäisistä vertailtavista kriteereistä, ilman tapaa muuttaa kriteerejä laskennalliseen muotoon. Myös Ring ym. (2019) toteavat tutkimuksessaan tietojoukon ratkaisevimpien ominaisuuksia olevan datan merkitseminen ja datan formaatti tietojoukon sisällä. Nämä havainnot tukevat Gharibin ym. 2016) luoman arviointikehyksen käyttämistä.

Gharib ym. (2016) mittaavat kehysmallinsa kriteerien toteutumista tietojoukossa kaavan 4 avulla. Kaavassa n on arviointikriteerien lukumäärä, joka tässä kehysmallissa on 11, ja m on kertoimien lukumäärä jokaiselle kriteerille. Ehdotetussa kehyksessä kahdelle kriteerille "hyökkäykset" ja "protokollat" m:n arvo on 7 ja 5, mutta muilla kriteereillä $m = 1$. Hyökkäysten ja protokollien arvot 7 ja 5 on johdettu reaalimaailmassa eniten esiintyvien hyökkäyksien ja protokollien mukaan. Eniten esiintyvät hyökkäykset jakautuvat kategorioihin, joita ovat: Browser, Bruteforce, DoS, Scan, DNS, Backdoor ja ”muut”, kun taas yleisimmät protokollat ovat http, https, ssh, ftp ja email. Tällä perusteella ”hyökkäysten monimuotoisuus” ja ”protokollien monimuotoisuus” kriteerit koostuvat useasta kertoimesta, jotka kuvaavat tiettyjä protokollia ja hyökkäyksiä, kun taas muut kriteerit arvioidaan binäärisellä 0 ja 1 asteikolla. (Gharib ym. 2016) Hyökkäysten ja käytetyimpien protokollien päivittäminen malliin onnistuu, jos tulevaisuudessa hyökkäysten ja protokollien jakautuminen globaalissa verkkoliikenteessä on tässä esitellystä poikkeavaa. Esitelty malli on siis adaptiivinen tulevaisuuden muuttuviin vaatimuksiin ja pystyy näin vastaamaan modernien tietoverkkojen alati muuttuvaan luonteeseen.

$$\sum_{i=1}^{n} W_i \left(\sum_{j=i}^{m} V_j * F_j \right) \tag{4}$$

Kaavan avulla saadaan vertailukelpoinen numeerinen arvo, joka kuvaa kriteerien toteutumista tietyssä tietojoukossa. Kullakin kehysmallin kriteerille on annettu painotettu arvo sen tärkeyden perusteella. Niiden kriteerien arvot, jotka jokin tietty tietojoukko täyttää lasketaan yhteen, jolloin saadaan keskinäisen vertailun mahdollistava numeerinen arvo, joka kuvaa kriteerien toteutumista tarkastelun kohteena olevassa tietojoukossa. Mallin tuottama arvo sijoittuu välille 0 ja 1, jossa arvolla 1 kaikki kriteerit täyttyvät ja arvolla 0 yksikään kriteeri ei täyty. Mitä lähempänä tietojoukon saama arvo on arvoa 1, sitä paremmin se tämän mallin mukaan sopii tunkeutumisen havaitsemisjärjestelmien tutkimuksessa ja kehityksessä käytettäväksi. Gharib ym. 2016)

6.4 Ajankohtaisuus

Kuten aiemmin mainittiin, useat tutkimukset korostavat tietojoukon nykyaikaisuuden merkitystä tietojoukkoja arvioidessa. Hyvän tietojoukon tulee olla uusi, koska tällainen tietojoukko mallintaa realistisimmin nykyisen Internet-liikenteen ja sisältää hyökkäysjäljet, jotka ovat tyypillisiä nykyaikaisille haittaohjelmille (Małowidzki, Bereziński ja Mazur 2015). Tästä syystä tietojoukon ajankohtaisuus arvioidaan sen iän perusteella omana yksittäisenä mittarinaan. Tietojoukon sisällön nykyaikaisuutta arvioidaan osittain jo realistisuutta arvioivassa mittarissa, mutta ajankohtaisuuden arvo nousee aihealueen tutkimuksessa niin tärkeään asemaan, ettei sen arvioiminen vain osana toista mittaria olisi ollut riittävää. Aikaisemmat tutkimukset alleviivaavat tietojoukon iän ja ajantasaisuuden merkitystä laadukkaan tutkimuksen ja kehityksen tekemisessä (Ghurab ym. (2021); Małowidzki, Bereziński ja Mazur (2015); Divekar ym. (2018)) Vanhemmissa tietojoukoissa huolenaiheeksi on noussut etenkin jopa kymmeniä vuosia sitten kerätyn tietoliikenneinformaation vastaavuus nykypäivän liikennemalleihin. Jotta mittariston graafiseen esitykseen ja sitä kautta nopeaan keskinäiseen vertailuun saatiin mukaan ajankohtaisuuden mittari, niin tietojoukon ajankohtaisuutta arvioiva ja vertaileva mittari ottaa huomioon vertailun kohteena olevien tietojoukkojen luomisvuodet ja arvottaa niiden perusteella tietojoukot järjestykseen. Mitä uudempi tietojoukko on, sitä paremman arvon se mittarin mitta-asteikolla saa.

7 Ratkaisun testaus

Ratkaisun testaus toteutetaan keräämällä valituista tietojoukoista vertailumittareiden avulla vertailuarvot. Testauksessa haetaan ensin vertailuarvot jokaisesta tietojoukosta, jonka jälkeen niitä on mahdollista vertailla keskenään kokonaisuuksina. Tietojoukot, joilla testausta suoritetaan sisältävät lähes samat tietojoukot kuin kappaleessa 5 esiteltiin, eli KDD-99, NSL-KDD, Kyoto2006+, unsw-nb15 ja CIC-IDS2017. Tämän lisäksi testaukseen on otettu mukaan muita aihealueen tutkimuksessa käytettyjä tietojoukkoja, joita ei muuten ole vielä tämän tutkimuksen piirissä esitelty. Vertailuun lisätyt tietojoukot ovat: ISCXIDS2012 (Shiravi ym. 2012), Twente (Sperotto ym. 2009), CIDDS-001 (Ring ym. 2017) ja CSE-CIC-IDS2018 ("A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018)" 2018). Huomionarvoista on, että CSE-CIC-IDS2018 tietojoukko on tuotettu pitkälti CIC-IDS2017 tietojoukon metodologian mukaisesti. Tuomalla testaukseen mukaan myös muita kuin kaikkein käytetyimmät tietojoukot, pyrittiin selvittämään, että onko käytetyimmissä tietojoukoissa, jotain merkittävästi erilaista, jonka takia juuri nämä tietojoukot ovat suositun aseman saavuttaneet. Suuremmalla joukolla mallin testaaminen tuottaa myös enemmän merkittävää tietoa mallin toiminnasta ja mahdollistaen samalla selkeämmän kuvan muodostamisen tietojoukkojen tilasta. Hindyn ym. (2020) mukaan nämä ovat tutkimuksessa käytetyimmät tietojoukot, joten ne antavat todennäköisesti relevanteimman kuvan vertailumallin soveltuvuudesta. Käytettävät tietojoukot hajautuvat myös luontiajankohdaltaan pitkälle aikavälille sekä uusimpia että kaikkein vanhimpia verkkotunkeutumisen havainnointiin luotuja tietojoukkoja.

Tiedot vertailumittareihin kerättiin artikkeleista ja dokumentaatioista, jotka käsittelivät kyseisen tietojoukon keruuta ja keruuympäristön luontia sekä tietojoukoista itsestään. Tietojoukon generointia koskevat tutkimukset eivät minkään tietojoukon kohdalla tarjonneet kaikkia mittareita tyydyttäviä tietoja, joten tietojoukkojen läpikäyminen ja tiedon hakeminen tietojoukoista oli välttämätöntä kaiken tarpeellisen tiedon saamiseksi. Toteutetun mallin testauksessa valitut tietojoukot asetettiin vertailumittareihin ja näiden mittareiden tuottamat tulokset esiteltiin niin, että ne mahdollistavat tehokkaan keskinäisen vertailun. Mittareiden toimita on avattu taulukoihin, joista jokainen edustaa yhtä mittaria, johon jokainen tietojoukko on sijoitettu. Taulukko 9 kuvataan tietojoukkojen epätasapainosuhteiden mittari. Taulukko 10 on kuvattu realismin tason mittari. Taulukko 11 kuvaa tietojoukon sisällöllisten

ominaisuuksien mittaria. Tietojoukkojen luomisvuodet on listattu Taulukko 12. Tietojoukon ajankohtaisuus.

Tietojoukko	Esiintymät	Epätasapainosuhde
KDD-99	$\frac{3\ 925\ 650}{972\ 781}$	4,04
NSL-KDD	$\frac{67\ 342}{58\ 630}$	1,15
Kyoto2006+	$\frac{50\ 033\ 015}{43\ 043\ 255}$	1,16
unsw-nb15	$\frac{2\ 218\ 761}{321\ 283}$	6,91
CIC-IDS2017	$\frac{2\ 358\ 036}{471\ 427}$	5,00
ISCXIDS2012	$\frac{2\ 381\ 532}{68\ 792}$	34,62
Twente	$\frac{7\ 721\ 692}{6\ 448\ 440}$	1,20
CIDDS-001	$\frac{28\ 051\ 907}{3\ 907\ 268}$	7,12
CSE-CIC-IDS2018	$\frac{2\ 856\ 035}{1\ 669\ 364}$	1,71

Taulukko 9. Tietojoukkojen epätasapainosuhteet

Tietojoukko	N	Säännöt	z_i	w_i	R
KDD-99	2	(x_1, y_1) (x_6, y_1)	4.32 4.32	0.16 0.16	0.33
NSL-KDD	2	(x_1, y_1) (x_6, y_1)	4.32 4.32	0.16 0.16	0.33
Kyoto2006+	1	(x_6, y_1)	2.16	0.16	0.16
unsw-nb15	4	(x_2, y_1) (x_3, y_2) (x_4, y_2) (x_6, y_1)	8.64 6.64 6.64 8.64	0.16 0.08 0.08 0.16	0.54

CIC-IDS2017	5	(x_1, y_2) (x_2, y_2) (x_3, y_2) (x_4, y_2) (x_6, y_2)	8.30 8.30 8.30 8.30 8.30	0.08 0.08 0.08 0.08 0.08	0.64
ISCXIDS2012	3	(x_1, y_1) (x_3, y_2) (x_6, y_1)	6,48 4,98 6,48	0,16 0,08 0,16	0.47
Twente	3	(x_1, y_1) (x_3, y_1) (x_6, y_1)	6,48 6,48 6,48	0,16 0,16 0,16	0.50
CIDDS-001	4	(x_1, y_2) (x_3, y_2) (x_4, y_2) (x_6, y_2)	6,32 6,32 6,32 6,32	0.08 0.08 0.08 0.08	0.49
CSE-CIC-IDS2018	5	(x_1, y_2) (x_2, y_2) (x_3, y_2) (x_4, y_2) (x_6, y_2)	8.30 8.30 8.30 8.30 8.30	0.08 0.08 0.08 0.08 0.08	0.64

Taulukko 10. Tietojoukkojen realismin tasot Haiderin ym.(2017) mallin mukaan.

Tietojoukko	Laskukaava	Pisteet
KDD-99	0.05*1 + 0.05*0 + 0.1*1 + 0.05*1 + 0.05*1+ 0.25 * (0.1 + 0.0 + 0.04 + 0.08 + 0.04) + 0.25 * (0.0 + 0.19 + 0.16 + 0.03 + 0.0 + 0.0 + 0.2) + 0.05*0 + 0.05*0 + 0.05*1 + 0.05*1	0.56
NSL-KDD	0.05*1 + 0.05*0 + 0.1*1 + 0.05*1 + 0.05*0+ 0.25 * (0.1 + 0.0 + 0.04 + 0.08 + 0.04) + 0.25 * (0.0 + 0.19 + 0.16 + 0.03 + 0.0 + 0.0 + 0.2) + 0.05*0 + 0.05*0 + 0.05*1 + 0.05*1	0.51
Kyoto2006+	0.05*1 + 0.05*0 + 0.1*1 + 0.05*1 + 0.05*1 + 0.25 * (0.1 + 0.74 + 0.04 + 0.08 + 0.04) + 0.25 * (0.36 + 0.19 + 0.16 + 0.03 + 0.03 + 0.03 + 0.2) + 0.05*0 + 0.05*0 + 0.05*1 + 0.05*1	0.85
unsw-nb15	0.05*1 + 0.05*1 + 0.1*1 + 0.05*1 + 0.05*1 + 0.25 * (0.1 + 0.00 + 0.04 + 0.08 + 0.04) + 0.25 * (0.36 + 0.19	0.77

	+ 0.16 + 0.03 + 0.03 + 0.03 + 0.2) + 0.05*1 + 0.05*0 + 0.05*1 + 0.05*1	
CIC-IDS2017	0.05*1 + 0.05*1 + 0.1*1 + 0.05*1 + 0.05*1 + 0.25 * (0.1 + 0.74 + 0.04 + 0.08 + 0.04) + 0.25 * (0.36 + 0.19 + 0.16 + 0.03 + 0.03 + 0.03 + 0.2) + 0.05*1 + 0.05*1 + 0.05*1 + 0.05*1	1
ISCXIDS2012	0.05*1 + 0.05*0 + 0.1*1 + 0.05*1 + 0.05*1 + 0.25 * (0.1 + 0.00 + 0.04 + 0.08 + 0.04) + 0.25 * (0.36 + 0.19 + 0.16 + 0.03 + 0.03 + 0.00 + 0.2) + 0.05*0 + 0.05*1 + 0.05*0 + 0.05*1	0.66
Twente	0.05*1 + 0.05*1 + 0.1*1 + 0.05*1 + 0.05*1 + 0.25 * (0.1 + 0.00 + 0.04 + 0.08 + 0.00) + 0.25 * (0.0 + 0.19 + 0.0 + 0.03 + 0.00 + 0.00 + 0.2) + 0.05*1 + 0.05*1 + 0.05*0 + 0.05*1	0.61
CIDDS-001	0.05*1 + 0.05*0 + 0.1*1 + 0.05*1 + 0.05*1 + 0.25 * (0.1 + 0.74 + 0.04 + 0.00 + 0.04) + 0.25 * (0.0 + 0.19 + 0.16 + 0.03 + 0.0 + 0.0 + 0.0) + 0.05*1 + 0.05*0 + 0.05*1 + 0.05*1	0.73
CSE-CIC-IDS2018	0.05*1 + 0.05*1 + 0.1*1 + 0.05*1 + 0.05*1 + 0.25 * (0.1 + 0.74 + 0.04 + 0.08 + 0.04) + 0.25 * (0.36 + 0.19 + 0.16 + 0.03 + 0.03 + 0.03 + 0.2) + 0.05*1 + 0.05*1 + 0.05*1 + 0.05*1	1

Taulukko 11. Numeeriset arvot tietojoukkojen sisällön laadulle Gharibin ym. (2016) mallin mukaisesti.

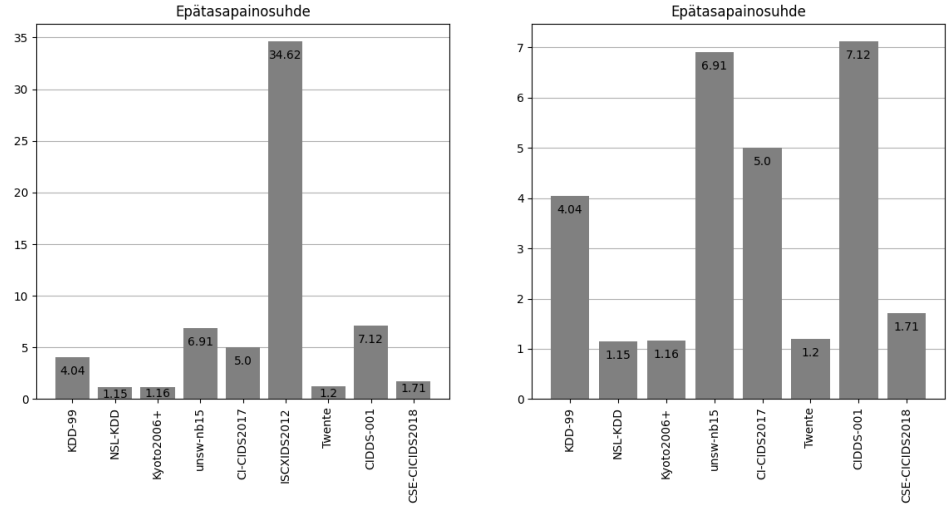
Tietojoukko	Valmistumisvuosi
KDD-99	1999
NSL-KDD	2009
Kyoto2006+	2006
unsw-nb15	2015
CIC-IDS2017	2017
ISCXIDS2012	2012
Twente	2009
CIDDS-001	2017

CSE-CIC-IDS2018	2018

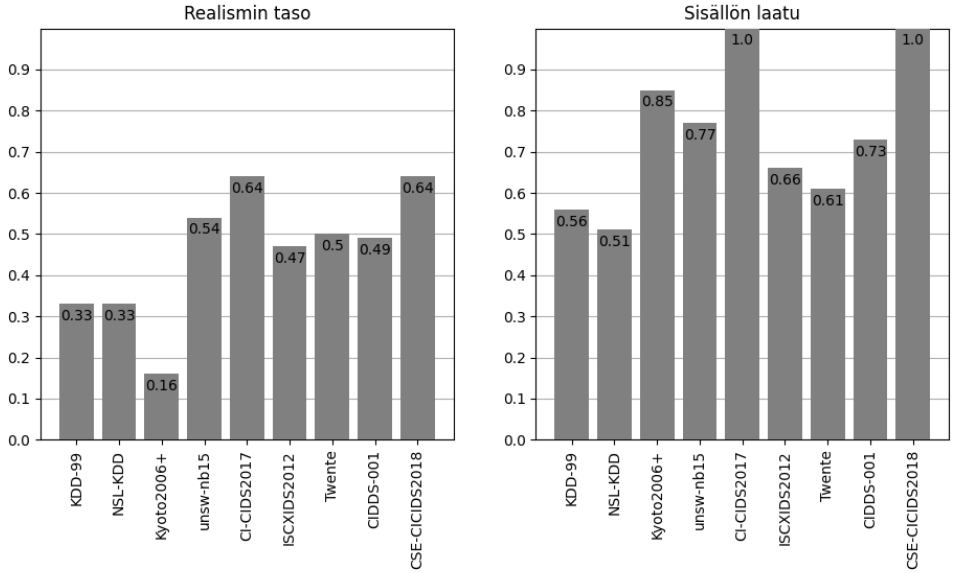
Taulukko 12. Tietojoukon ajankohtaisuus.

Taulukoiden tuottamat tulokset on havainnollistettu graafiseen muotoon, jotta mittarin tuottamien arvojen keskinäinen vertailu olisi selkeämpää ja helpompaa. Graafinen esitysasu mahdollistaa nopean eri tietojoukkojen keskinäisen paremmuuden vertailun jollain tietyllä mittarilla. Sekä epätasapainosuhdetta että ajankohtaisuutta mittaavien mittareiden tulokset on standardisoitu välille $[0,1]$. Epätasapainoisuus on standardisoitu niin, että eniten epätasapainossa oleva tietojoukko saa arvon 0, ja vähiten epätasapainossa, eli näin ollen tällä kriteerillä laadullisesti paras, tietojoukko saa arvon 1. Erityistä huomioitavaa tässä on että vaikka jokin tietojoukko saa parhaan arvon keskinäisessä vertailussa, ei se kerro kyseisen tietojoukon välttämän olevan täysin tasapainossa, vaan ainoastaan, että se on vertailtavista vaihtoehdoista tasapainoisin. Tietojoukkojen epätasapainosuhteet on esitetty Kuva 3. Epätasapainosuhteet on selkeyden vuoksi esitetty kahdessa eri kuviossa, joissa toisessa on mukana kaikki esitetyt tietojoukot ja toisesta on poistettu tietojoukko ISCXIDS2012. ISCXIDS2012 -tietojoukolla on niin korkea epätasapainosuhde, että muiden tietojoukkojen väliset erot jäävät verrattain epäselviksi, vaikka ne ovat todellisuudessa merkittäviä eroja. Jotta muiden keskinäiset erot olisivat myös helposti nähtävissä, esitetään myös kaavio ilman tätä. Edellä mainitusta syystä on vertailua vääristävä tietojoukko jätetty pois myös muista kaavioista, jotka sisältävät epätasapainosuhteen graafisen esityksen.

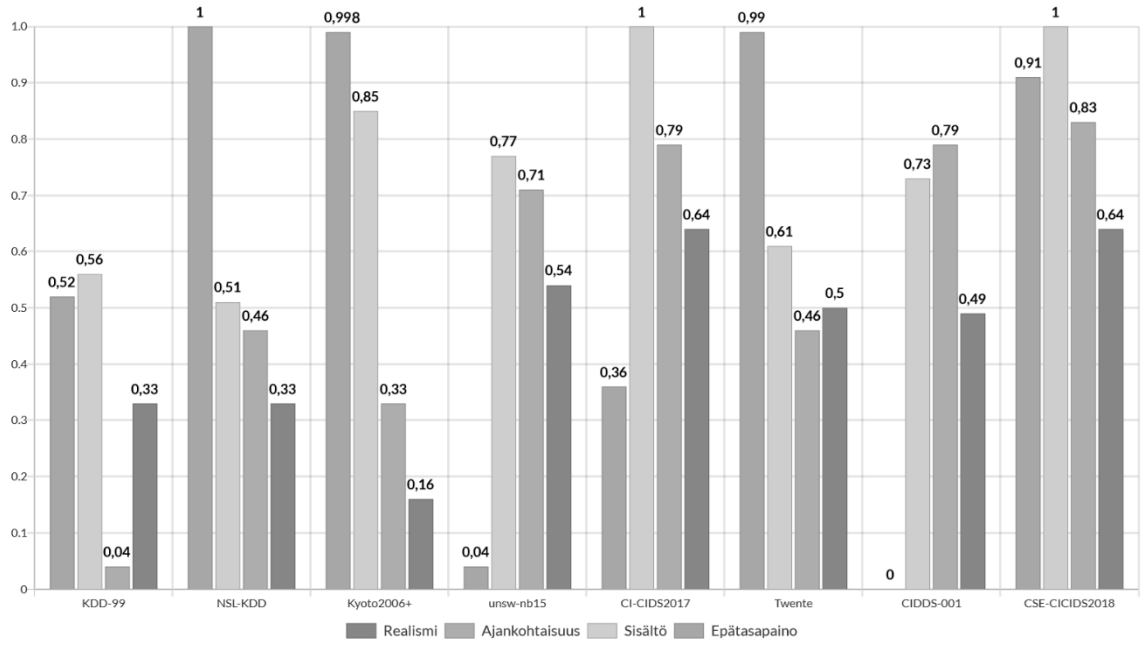
Luodun mittarin graafiseen esitystapaan ajankohtaisuutta on arvioitu asteikolla $[0,1]$. Tietojoukkojen julkaisuvuodet on standardisoitu asteikolle niin, että matalin mahdollinen arvo 0, kuvaa vuotta 1998, jolloin ensimmäiset tunkeutumisen havaitsemiseen liittyvät tietojoukot julkaistiin ja suurin arvo 1 kuvaa nykyistä vuotta 2022. Näin ollen mitä uudempi tietojoukko on, sitä lähempänä se on arvoa yksi. Muiden yksittäisten mittareiden eli realismin tason ja sisällöllisen laadun arvot ovat nähtävissä Kuva 4. Kuva 5 kuvaa kaikkien luotujen mittareiden tuottamia arvoja tässä konstruktion testauksessa käytetyille tietojoukoille. Kuva 6 on eritelty jokaisen mittarin tuottaman arvon lisäksi kaikki yhden tietojoukon saamat arvot yhteenlaskettuna, joka mahdollistaa tietojoukkojen laadun nopean ja kokonaisvaltaisen vertailun.



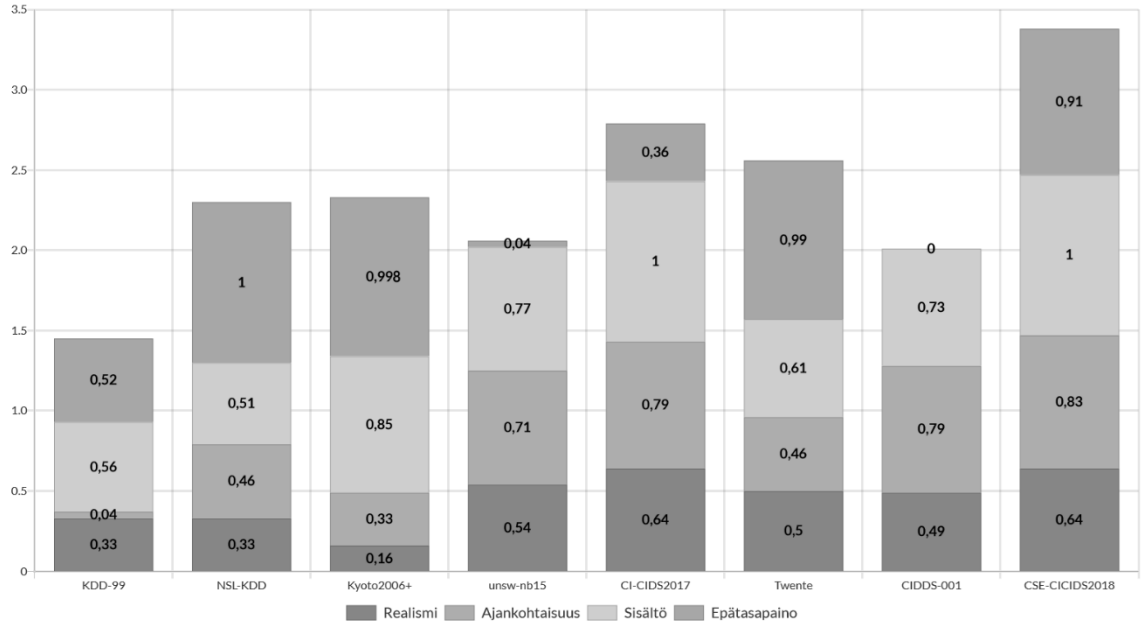
Kuva 3. Tietojoukkojen epätasapainosuhteet.



Kuva 4. Tietojoukkojen realismin taso ja sisällön laatu



Kuva 5. Vertailumittareiden tulokset tietojoukoille



Kuva 6. Vertailumittareiden yhteenlasketut tulokset tietojoukoille

Luotujen mittareiden ja niiden avulla kerättyjen arojen avulla on mahdollista helposti vertailla tietojoukkoja keskenään ja valita niistä sopivin. Jotta tietojoukkoja vertaillessa saataisiin mahdollisimman tarkka vertailutulos, on kiinnitettävä huomiota sekä vertailumittareiden

yhteenlasketuille tuloksille että yksittäisten mittareiden yksittäisille arvoille. Pelkästään yhteenlaskettuja arvoja vertailemalla on helppo saada nopea kokonaiskuva siitä, miten tietojoukot mittaristolla sijoittuvat ja suuret erot on helppo huomata. On esimerkiksi yhteenlaskettujen pisteiden avulla selkeää, että CSE-CIC-IDS2018 on saanut huomattavasti korkeammat pisteet kuin KDD-99 ja on näin ollen selkeästi parempi, ilman syvällisempää tarkastelua. Kuitenkin on syytä yhteenlaskettujen tulosten lisäksi syytä tarkastella vertailumittareiden tuottamien tulosten jakautumiseen tietyn tietojoukon kohdalla, etenkin jos vertailtavien tietojoukkojen yhteenlaskettujen pisteiden välillä ei ole suurtakaan eroa. Esimerkiksi on syytä huomioida, että vaikka NSL-KDD ja UNSW-NB15 ovat yhteenlaskettujen pisteiden osalta verrattain lähellä toisiaan, on niin pisteytyksen jakautuminen hyvin erilainen. NSL-KDD tietojoukon pisteistä suuri osa tulee siitä, että tietojoukko on varsin laadukas epätasapainosuhteen suhteen, kun taas UNSW-NB15 saa vastaavasta mittarista vain 0,04 pistettä. Näin vertailun olennaiseksi osaksi muodostuu myös pisteiden jakautumisen tarkastelu ja analysointi. Yhteenlaskettu pistemäärä saattaa siis kuvata yhden mittarin yliedustusta tietojoukkoa analysoitaessa.

8 Teoreettinen kontribuutio

Lukan (2001) mukaan konstruktiivisessa tutkimuksessa on mahdollista saavuttaa teoreettista kontribuutiota kahdella eri tavalla; uudella konstruktiolla itsellään sekä uuden konstruktion takana olevien riippuvuussuhteiden kautta. Tämän tutkimuksen tieteellinen ja teoreettinen kontribuutio keskittyy vahvasti etenkin jälkimmäiseen kontribuution tapaan testatessaan ja soveltaessaan aiempien tutkimusten luomia arviointitapoja uuden yksittäisen konstruktion luomisessa. Tämä noudattaa Lukan (2001) näkemystä siitä, että konstruktiivinen tutkimusprojekti on areena olemassa olevan rakenteita ja prosesseja koskevan teoreettisen tietämyksen soveltamiselle, testaamiselle ja kehittämiselle. Tässä tutkimuksessa on ollutkin erityisesti läsnä juuri olemassa olevien teoreettisten rakenteiden soveltaminen, testaaminen ja uudelleen arviointi.

Tämän tutkimuksen tuottama uutuusarvo on keskittynyt kahteen eri osa-alueeseen. Ensimmäinen uutuusarvoa tuottava ominaisuus on tämän tutkimuksen osoittama todistus siitä, että tietojoukkojen keskinäinen laskennallinen arvioiminen niiden laadullisten ominaisuuksien perusteella on sekä mahdollista, että tämän mallin avulla verrattain helposti toteutettavissa. Toinen ominaisuus on jo tämän tutkimuksen konstruktion testausosiossa toteutettu vertailu, joista on mahdollista nähdä suoraan aihealueen tutkimuksen suosituimpien tietojoukkojen keskinäinen vertailu ja sen tulokset ilman, että vertailua pitäisi toteuttaa mallin pohjalta itse. Tästä näkökulmasta pelkkä tässä tutkimuksessa suoritettu mallin testaaminen tuottaa merkittävää uutuusarvoa paitsi todistaessaan vertailumallin toimivuuden, niin myös tarjotessaan laajan näkymän tietojoukkojen jo mitattuun laatuun. Tämän tutkimuksen testauksessa tuotettua keskinäistä vertailua on mahdollista käyttää sellaisenaan apuna tulevaisuuden tutkimuksissa tietojoukkoa valittaessa, olettaen, että tutkijat ovat valitsemassa tietojoukkoja tässä vertailussa esiintyvien tietojoukkojen joukosta. Kuten Lukka (2001) toteaa; tietämyksen kehittymisen kannalta yleensä konstruktiivinen tutkimus on luonnostaan sovelias pienentämään käytännön ja tutkimuksen välistä kuilua. Tämän tutkimuksen tuottama vertailumalli onkin pohjimmaiselta tarkoitukseltaan tuottamassa käytännön ongelmaan sovellettavan mallin, joka on toteutettu vahvasti aiemman tietojoukkoja analysoivan ja vertailevan tutkimuksen päälle.

Tietojoukkojen keskinäisen vertailun ja tietojoukkojen ominaisuuksien vertailun teorian keskinäinen kysymys on, että voidaanko kvantitatiivisia mittauksia käyttää näiden ominaisuuksien laadun mittaamiseen (Shiravi ym. 2012). Aiempien tietojoukkoja analysoineiden tutkimustapojen, kuten (Haider ym. 2017) ja (Gharib ym. 2016), yhteen liittäminen ja näiden avulla saatujen tulosten kvantifioiminen yhtenäisesti vertailtavaan muotoon, sekä hyvän tietojoukon kriteerien mukaisen laajemman vertailumittariston suunnittelu, luominen ja testaaminen on sekä aiemman tutkimuksen tuottaman tiedon verifikaatiota, sekä suoraan sen päälle rakentavaa ja niiden teoriaa laajentavaa. Tuotettu konstruktio on ensimmäinen tunkeutumisen havaitsemisjärjestelmä-tietojoukoille tehty laskennallisen vertailun mahdollistava vertailumalli, joka arvioi tietojoukkojen laadullisia ominaisuuksia usean eri mittarin ja näkökulman kautta. Tämä paitsi osoittaa aiemman tutkimuksen tuottaman teorian soveltamisen ja toimivuuden käytännön sovellutuksessa, että rakentaa täysin uuden laajemman kokonaisuuden näiden teorioiden päälle.

Lukan (2001) määritelmän mukaan konstruktiivisen tutkimusotteen soveltamista voidaan pitää integroituna yrityksenä, joka käyttää olemassa olevaa perustietämystä soveltavan empiirispainotteisen tutkimusprosessin eräänä syötteenä ja palaa taas prosessin lopussa perustietämyksen tasolle analysoidakseen sen suhteen saavutettavaa kontribuutiota. Konstruktion rakentaminen aiemman tutkimuksen tuottamien mittareiden ja teorioiden avulla sekä konstruktion toteutus ja testaaminen oikeilla tietojoukoilla osoittaa, että tietojoukkojen keskinäinen laskennallinen vertailu on sekä mahdollista että pääosin linjassa aiempien tutkimusten mukaisten arvioiden kanssa. Esimerkiksi Arqane ym. (2021) tutkimuksessaan toteavat perinteisten tietojoukkojen kuten DARPA98 ja KDD-99 olevan huono valinta tunkeutumisen havaitsemisjärjestelmien tutkimukseen, kun taas tietojoukot, kuten UNSW-NB15, CIC-IDS2017 ja CSE-CIC-IDS 2018, suositellaan riittäviksi. Vastaavanlaisiin päätelmiin on tullut suuri määrä aihealueen tutkimusta, mutta perustelu tietojoukon kritisoimiselle on ollut pääasiassa vain tietojoukon ikä. Tässä tutkimuksessa tuotettu vertailumalli osoittaa aiemman tutkimuksen olleen pääosin oikeassa uudempien tietojoukkojen laadukkuudesta, mutta luotu vertailumalli on päätynyt tähän tulokseen vertailemalla useampaa laskennallista mittaria, kuin vain tietojoukon ikää. Tutkimuksen teoriaa verifioiva osuus korostuu myös siinä, että osana tuotetun konstruktion testausta, laajennetaan aiemman tutkimuksen luomia mittareita

ja teoriaa myös sellaisiin ympäristöihin, joissa sitä ei vielä aikaisemmin ole testattu. Esimerkiksi Haiderin ym. (2017) tutkimusta laajennetaan koskemaan useampaa tietojoukkoa, kuin mitä aiemmassa tutkimuksessa on tehty sekä heidän malliaan laajennetaan tietojoukkojen keskinäiseen vertailuun asti.

9 Yhteenveto

Tämän tutkimuksen tavoitteena oli selvittää, millä erilaisilla kriteereillä koneoppimispohjaisissa tunkeutumisen havaitsemisjärjestelmissä käytettäviä tietojoukkoja voidaan arvioida ja vertailla keskenään. Tietojoukkojen keskinäisen vertailun mahdollistaminen tuottaisi arvokasta tietoa tietojoukkojen valintaprosesseihin ja auttaisi tulevaisuudessa tutkijoita ja kehittäjiä valitsemaan parhaan mahdollisen tietojoukon. Aiempi tietojoukkoja keskenään vertaileva tutkimus on ollut varsin vähäistä, ja tavoitteena oli toteuttaa ensimmäinen tietojoukkoja laajemmin keskenään vertaileva arviointimalli.

Metodin valinta oli tämän tutkimuksen puitteissa hyvin suoraviivainen, sillä tämän tutkimuksen edellyttämät toimintamallit vastaavat konstruktiivisen tutkimusotteen sisältämiä menetelmiä varsin tarkasti. Aiempaan teoriapohjaan nojaava, reaalimaailman ongelmaan ratkaisua tarjoavan mallin suunnitteluun, toteutukseen ja lopulta sen käytännön toiminnan arviointiin ei löytynyt konstruktiivista tutkimusotetta sopivampaa menetelmää.

Tietojoukkojen keskinäiselle vertailulle perustan muodostavat ne kriteerit, jotka ovat yleisesti mielletty hyvien tietojoukkojen perustaksi ja joita hyviltä tietojoukoilta edellytetään olevan. Näitä kriteereitä tunnistettiin neljä ja ne olivat tietojoukon epätasapainoisuus, tietojoukon realismin taso, tietojoukon ajankohtaisuus sekä tietojoukon sisällöllinen laatu. Nämä ominaisuudet ovat teoriassa mahdollista tunnistaa jokaisesta tunkeutumisen havaitsemisjärjestelmien tutkimukseen luodosta tietojoukosta tai muista tietoverkkoliikennettä sisältävästä tietojoukosta. Näiden ominaisuuksien soveltaminen vertailuun ei kuitenkaan ole täysin ongelmatonta. Ominaisuuksien tunnistaminen tietojoukosta, edellyttää tietojoukon tehneiden tahojen huolellista dokumentointia tietojoukon luomisprosessista sekä tietojoukon sisällöstä. Tietojoukon sisältöä on myös hyvä päästä tarkastelemaan ennen arviointia, jotta sisällölliset ominaisuudet voidaan tunnistaa tai varmentaa.

Jokaiseen eri kriteerin vertailumittariin liittyy myös tiettyjä heikkouksia, jotka ovat syytä huomioida, kun tietojoukkoja vertaillaan keskenään mallin avulla. Realismin mittari kärsii tulkinnanvaraisuuden ongelmasta, jossa arvioinnin kohteena olevasta tietojoukosta realismia kuvaavien ominaisuuksien löytäminen jää arvioinnin tekijän tulkinnan vastuulle. Tietojoukon realistisuus on tulkinnanvarainen ja useista eri tekijöistä, kuten liikenteen keräys- ja

generointitavoista koostuva arvio. Tietojoukon uutuus ei myöskään tässä tutkimuksessa luodun mallin mukaan ole takuu sille, että tietojoukko on laadukkaampi. Sisällöllistä laatua mittaava kaava painottaa muuttujia tämänhetkisten verkkoliikenteen trendien mukaisesti, joten sitä on päivitettävä samalla kun reaalimaailman liikenteen trendit muuttuvat.

Huolimatta siitä, että mittarit sisältävät tiettyjä heikkouksia, oli kuitenkin mahdollista toteuttaa tietojoukkoja ja niiden laatua keskenään laskennallisesti vertaileva malli, joka mahdollistaa parhaimman tietojoukon valitsemisen vertailun kohteista. Vertailumalli ottaa huomioon tietojoukkojen laatua tärkeimmin kuvaavat neljä kriteeriä ja muodostaa näille numeerisen arvon. Löydettyjen ominaisuuksien suora vertailu ei olisi ollut mahdollista ilman, että nämä ominaisuudet muutettiin laskennalliseen muotoon, jotta kvantitatiivisia mittauksia pystyttiin käyttämään näiden ominaisuuksien laadun mittaamiseen. Nämä arvot on mahdollista esittää graafisessa tai numeerisessa muodossa, jolloin tietojoukkojen laadullisista ominaisuuksista voi muodostaa tehokkaasti kattavan yleiskuvan. Keskinäisen laadun vertailun avulla voidaan valita käsillä olevista tietojoukoista laadukkain ja sitä kautta tuottaa laadukkaampaa tutkimusta. Vertailumallia sovellettiin yhdeksään tunnettuun tietojoukkoon ja tulokset esitettiin numeerisessa ja graafisessa muodossa. Vertailun tulokset tukevat aiempaa tutkimusta ja ovat linjassa yleisen näkemyksen kanssa siitä, mitkä tietojoukot ovat suositeltuja käyttää. Tietojoukot, kuten KDD-99 ja NSL-KDD ovat vertailussa verrattain matalalla. Tietojoukkoja on laajasti kritisoitu niiden ajankohtaisuuden ja realismin puutteesta, mikä on nähtävissä myös tämän tutkimuksen tuottamasta mallista. CIC-IDS2017 ja CSE-CIC-IDS2018 menestyivät tällä mittarilla erityisen hyvin, mikä tukee käsitystä siitä, että uudemmat tietojoukot ovat pystyneet ottamaan huomioon vanhempien tietojoukkojen puutteet, sekä siitä että ne kuvaavat paremmin nykypäivän reaalimaailman liikennettä. Vertailun tuloksista on nähtävissä, että tietojoukkojen laskennallisen laadun määrä on tässä tutkimuksessa esitellyllä mittaristolla kasvanut ja useimmissa tapauksissa uudet tietojoukot ovat laadukkaampia vanhoihin verrattuna. Huomionarvoista on kuitenkin, että tietojoukkojen epätasapaino on jatkuva ongelma riippumatta tietojoukon ajankohtaisuudesta tai muista muuttujista.

Tutkimuksessa puutteena on tietojoukkojen vertailun tekeminen ainoastaan teoreettisella tasolla. Tietojoukkojen suoriutumista ei arvioitu lainkaan implementoimalla niitä varsinaiseen

tunkeutumisen havaitsemisjärjestelmään, jonka takia tietojoukon todellista suoriutumista ei voitu varmentaa. Haasteeksi muodostuu myös vertailukriteerien johtaminen aiemmasta tutkimuksesta. Koska vertailumittari on riippuvainen tutkimuksesta ja dokumentaatiosta, joka on tuotettu, kun tietojoukkoja on luotu, ei ole mahdollista ottaa vertailuun mukaan tietojoukkoja, joista näitä dokumentteja ei ole olemassa.

Tulevaisuuden tutkimuksessa tietojoukkoja tulisi vertailla myös testaamalla niitä eri käytännön koneoppimismenetelmillä. Näin mahdollistettaisiin teoreettisiin ominaisuuksiin perustuvan vertailututkimuksen lisäksi tietojoukkojen käytännön toiminnallisuuksien vertailu, ja nähtäisiin myös, miten tietojoukot soveltuvat eri koneoppimismenetelmiin. Lisäksi tämä tutkimus kannustaa kehittämään entistä parempia ja nykyaikaisempia tietojoukkoja tunkeutumisen havaitsemisjärjestelmien tutkimukseen ja kehitykseen, sillä myös tämä tutkimus osoittaa, että täydellistä tietojoukkoa ei ole toistaiseksi pystytty luomaan.

Lähteet

"A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018)." 2018. Canadian Institute for Cybersecurity, University of New Brunswick. https://registry.opendata.aws/cse-cic-ids2018/.

Abdulhammed, Razan, Hassan Musafer, Ali Alessa, Miad Faezipour, and Abdelshakour Abuzneid. 2019. "Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection." *Electronics* 8 (3): 322. https://doi.org/10.3390/electronics8030322.

Al-Daweri, Muataz Salam, Khairul Akram Zainol Ariffin, Salwani Abdullah, and Mohamad Firham Efendy Md. Senan. 2020. "An Analysis of the KDD99 and UNSW-NB15 Datasets for the Intrusion Detection System." *Symmetry* 12 (10): 1666. https://doi.org/10.3390/sym12101666.

Amarudin, Ridi Ferdiana, and Widyawan. 2020. "A Systematic Literature Review of Intrusion Detection System for Network Security: Research Trends, Datasets and Methods." In *2020 4th International Conference on Informatics and Computational Sciences (ICICoS)*, 1–6. Semarang, Indonesia: IEEE. https://doi.org/10.1109/ICICoS51170.2020.9299068.

Amoli, Payam Vahdani, Timo Hamalainen, Gil David, and Mikhail Zolotukhin. 2015. "Unsupervised Network Intrusion Detection Systems for Zero-Day Fast- Spreading Attacks and Botnets," 13.

Anwar, Shahid, Jasni Mohamad Zain, Mohamad Fadli Zolkipli, Zakira Inayat, Suleman Khan, Bokolo Anthony, and Victor Chang. 2017. "From Intrusion Detection to an Intrusion Response System: Fundamentals, Requirements, and Future Directions." *Algorithms* 10 (2): 39. https://doi.org/10.3390/a10020039.

Arqane, Aouatif, Omar Boutkhoum, Hicham Boukhriss, and Abdelmajid El Moutaouakkil. 2021. "A Review of Intrusion Detection Systems: Datasets and Machine Learning Methods." In *The 4th International Conference on Networking, Information Systems Amp Security.*, 1–6. KENITRA AA Morocco: ACM. https://doi.org/10.1145/3454127.3456576.

Azzaoui, Hanane, and Akram Boukhamla. 2020. “Two-Stages Intrusion Detection System Based On Hybrid Methods.” In *Proceedings of the 10th International Conference on Information Systems and Technologies*, 1–7. Lecce Italy: ACM. https://doi.org/10.1145/3447568.3448512.

Bhuyan, Monowar H., D. K. Bhattacharyya, and J. K. Kalita. 2014. “Network Anomaly Detection: Methods, Systems and Tools.” *IEEE Communications Surveys & Tutorials* 16 (1): 303–36. https://doi.org/10.1109/SURV.2013.052213.00046.

Bhuyan, Monowar H, Dhruba K Bhattacharyya, and Jugal K Kalita. 2015. “Towards Generating Real-Life Datasets for Network Intrusion Detection,” 19.

Bolzoni, Damiano, and Sandro Etalle. 2008. “Approaches in Anomaly-Based Network Intrusion Detection Systems.” In *Intrusion Detection Systems*, 38:1–16. Advances in Information Security. Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-77265-3_1.

Buczak, Anna L., and Erhan Guven. 2016. “A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection.” *IEEE Communications Surveys & Tutorials* 18 (2): 1153–76. https://doi.org/10.1109/COMST.2015.2494502.

Bukac, Vit, Pavel Tucek, and Martin Deutsch. 2012. “Advances and Challenges in Standalone Host-Based Intrusion Detection Systems.” In , 12. https://doi.org/10.1007/978-3-642-32287-7_9.

Chindove, Hatitye, and Dane Brown. 2021. “Adaptive Machine Learning Based Network Intrusion Detection.” In *Proceedings of the International Conference on Artificial Intelligence and Its Applications*, 1–6. Virtual Event Mauritius: ACM. https://doi.org/10.1145/3487923.3487938.

Chkirbene, Zina, Aiman Erbad, Ridha Hamila, Amr Mohamed, Mohsen Guizani, and Mounir Hamdi. 2020. “TIDCS: A Dynamic Intrusion Detection and Classification System Based Feature Selection.” *IEEE Access* 8: 95864–77. https://doi.org/10.1109/ACCESS.2020.2994931.

Cieslak, D.A., N.V. Chawla, and A. Striegel. 2006. “Combating Imbalance in Network Intrusion Datasets.” In *2006 IEEE International Conference on Granular Computing*, 732–37. Atlanta, GA, USA: IEEE. https://doi.org/10.1109/GRC.2006.1635905.

“Cisco Annual Internet Report (2018–2023) White Paper.” 2020. White paper. Cisco. https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

Cunningham, R. K., R. P. Lippmann, D. J. Fried, S. L. Garfinkel, I. Graf, K. R. Kendall, S. E. Webster, D. Wyschogrod, and M. A. Zissman. 1999. “Evaluating Intrusion Detection Systems Without Attacking Your Friends: The 1998 DARPA Intrusion Detection Evaluation:” Fort Belvoir, VA: Defense Technical Information Center. https://doi.org/10.21236/ADA526274.

Dang, Quang-Vinh. 2021. “Evaluating Machine Learning Algorithms for Intrusion Detection Systems Using the Dataset CIDDS-002.” In *2021 4th International Conference on Computer Science and Software Engineering (CSSE 2021)*, 112–18. Singapore Singapore: ACM. https://doi.org/10.1145/3494885.3494906.

Devulapalli, Saurabh. 2021. “A MACHINE LEARNING APPROACH FOR UNIFORM INTRUSION DETECTION.” THE PURDUE UNIVERSITY GRADUATE SCHOOL.

Dina, Ayesha S, and D Manivannan. 2021. “Intrusion Detection Based on Machine Learning Techniques in Computer Networks.” *Internet of Things*, 18.

Divekar, Abhishek, Meet Parekh, Vaibhav Savla, Rudra Mishra, and Mahesh Shirole. 2018. “Benchmarking Datasets for Anomaly-Based Network Intrusion Detection: KDD CUP 99 Alternatives.” *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*, October, 1–8. https://doi.org/10.1109/CCCS.2018.8586840.

Faker, Osama, and Erdogan Dogdu. 2019. “Intrusion Detection Using Big Data and Deep Learning Techniques.” In *Proceedings of the 2019 ACM Southeast Conference*, 86–93. Kennesaw GA USA: ACM. https://doi.org/10.1145/3299815.3314439.

Farah, Nutan, Md. Avishek, Faisal Muhammad, Abdur Rahman, Musharrat Rafni, and Dewan Md. 2015. “Application of Machine Learning Approaches in Intrusion Detection System: A Survey.” *International Journal of Advanced Research in Artificial Intelligence* 4 (3). https://doi.org/10.14569/IJARAI.2015.040302.

Ferrag, Mohamed Amine, Leandros Maglaras, Sotiris Moschoyiannis, and Helge Janicke. 2020. “Deep Learning for Cyber Security Intrusion Detection: Approaches, Datasets, and Comparative Study.” *Journal of Information Security and Applications* 50 (February): 102419. https://doi.org/10.1016/j.jisa.2019.102419.

Ganesh, Vaishnavi. 2019. “A Review on Artificial Intelligence Methods For Cyber Intrusion Detection” 6 (5): 5.

García-Teodoro, P., J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez. 2009. “Anomaly-Based Network Intrusion Detection: Techniques, Systems and Challenges.” *Computers & Security* 28 (1–2): 18–28. https://doi.org/10.1016/j.cose.2008.08.003.

Gharib, Amirhossein, Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. 2016. “An Evaluation Framework for Intrusion Detection Dataset.” In *2016 International Conference on Information Science and Security (ICISS)*, 1–6. Pattaya, Thailand: IEEE. https://doi.org/10.1109/ICISSEC.2016.7885840.

Ghurab, Mossa, Ghaleb Gaphari, Faisal Alshami, Reem Alshamy, and Suad Othman. 2021. “A Detailed Analysis of Benchmark Datasets for Network Intrusion Detection System.” *Asian Journal of Research in Computer Science*, April, 14–33. https://doi.org/10.9734/ajrcos/2021/v7i430185.

Gumusbas, Dilara, Tulay Yldrm, Angelo Genovese, and Fabio Scotti. 2021. “A Comprehensive Survey of Databases and Deep Learning Methods for Cybersecurity and Intrusion Detection Systems.” *IEEE Systems Journal* 15 (2): 1717–31. https://doi.org/10.1109/JSYST.2020.2992966.

Gurung, Sandeep, Mirnal Kanti Ghose, and Aroj Subedi. 2019. “Deep Learning Approach on Network Intrusion Detection System Using NSL-KDD Dataset.” *International Journal*

of Computer Network and Information Security 11 (3): 8–14. https://doi.org/10.5815/ijcnis.2019.03.02.

Haider, W., J. Hu, J. Slay, B.P. Turnbull, and Y. Xie. 2017. "Generating Realistic Intrusion Detection System Dataset Based on Fuzzy Qualitative Modeling." *Journal of Network and Computer Applications* 87 (June): 185–92. https://doi.org/10.1016/j.jnca.2017.03.018.

Hindy, Hanan, David Brosset, Ethan Bayne, Amar Kumar Seeam, Christos Tachtatzis, Robert Atkinson, and Xavier Bellekens. 2020. "A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems." *IEEE Access* 8: 104650–75. https://doi.org/10.1109/ACCESS.2020.3000179.

Hnamte, Vanlalruata, and Jamal Hussain. 2021. "An Extensive Survey on Intrusion Detection Systems: Datasets and Challenges for Modern Scenario." In *2021 3rd International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)*, 1–10. Kuala Lumpur, Malaysia: IEEE. https://doi.org/10.1109/ICECIE52348.2021.9664737.

Jha, Jayshree. 2013. "Intrusion Detection System Using Support Vector Machine." *International Journal of Applied Information Systems*, 6.

Karatas, Gozde, Onder Demir, and Ozgur Koray Sahingoz. 2020. "Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset." *IEEE Access* 8: 32150–62. https://doi.org/10.1109/ACCESS.2020.2973219.

Kendall, Kristopher. 1999. "A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems." MASSACHUSETTS INSTITUTE OF TECHNOLOGY. https://archive.ll.mit.edu/ideval/files/kkendall_thesis.pdf.

Khoshgoftaar, Taghi M., Moiz Golawala, and Jason Van Hulse. 2007. "An Empirical Study of Learning from Imbalanced Data Using Random Forest." In *19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007)*, 310–17. Patras, Greece: IEEE. https://doi.org/10.1109/ICTAI.2007.46.

Khraisat, Ansam, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. 2019. “Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges.” *Cybersecurity* 2 (1): 20. https://doi.org/10.1186/s42400-019-0038-7.

Kim, Taehoon, and Wooguil Pak. 2022. “Robust Network Intrusion Detection System Based on Machine-Learning With Early Classification.” *IEEE Access* 10: 10754–67. https://doi.org/10.1109/ACCESS.2022.3145002.

Kruegel, Christopher, and Thomas Toth. 2003. “Using Decision Trees to Improve Signature-Based Intrusion Detection.” In *Recent Advances in Intrusion Detection*, edited by Giovanni Vigna, Christopher Kruegel, and Erland Jonsson, 2820:173–91. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-45248-5_10.

Kumar, Satish, Sunanda Gupta, and Sakshi Arora. 2021. “Research Trends in Network-Based Intrusion Detection Systems: A Review.” *IEEE Access* 9: 157761–79. https://doi.org/10.1109/ACCESS.2021.3129775.

Lakshminarayana, Deepthi Hassan, James Philips, and Nasseh Tabrizi. 2019. “A Survey of Intrusion Detection Techniques.” In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 1122–29. Boca Raton, FL, USA: IEEE. https://doi.org/10.1109/ICMLA.2019.00187.

Le Jeune, Laurens, Toon Goedeme, and Nele Mentens. 2021. “Machine Learning for Misuse-Based Network Intrusion Detection: Overview, Unified Evaluation and Feature Choice Comparison Framework.” *IEEE Access* 9: 63995–15. https://doi.org/10.1109/ACCESS.2021.3075066.

Liao, Hung-Jen, Chun-Hung Richard Lin, Ying-Chih Lin, and Kuang-Yuan Tung. 2013. “Intrusion Detection System: A Comprehensive Review.” *Journal of Network and Computer Applications* 36 (1): 16–24. https://doi.org/10.1016/j.jnca.2012.09.004.

Lukka, Kari. 2001. “Konstruktiivinen Tutkimusote.” 2001. https://metodix.fi/2014/05/19/lukka-konstruktiivinen-tutkimusote/.

———. 2003. “The Constructive Research Approach.” In *Case Study Research in Logistics*, 83–101. B. Turku School of Economics and Business Administration.

Małowidzki, Marek, Przemysław Bereziński, and Michał Mazur. 2015. “Network Intrusion Detection: Half a Kingdom for a Good Dataset.”

Maseer, Ziadoon Kamil, Robiah Yusof, Nazrulazhar Bahaman, Salama A. Mostafa, and Cik Feresa Mohd Foozy. 2021. “Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset.” *IEEE Access* 9: 22351–70. https://doi.org/10.1109/ACCESS.2021.3056614.

Mishra, Preeti, Vijay Varadharajan, Uday Tupakula, and Emmanuel S. Pilli. 2019. “A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection.” *IEEE Communications Surveys & Tutorials* 21 (1): 686–728. https://doi.org/10.1109/COMST.2018.2847722.

Moustafa, Nour, and Jill Slay. 2015. “UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set).” In *2015 Military Communications and Information Systems Conference (MilCIS)*, 1–6. Canberra, Australia: IEEE. https://doi.org/10.1109/MilCIS.2015.7348942.

Mukherjee, B., L.T. Heberlein, and K.N. Levitt. 1994. “Network Intrusion Detection.” *IEEE Network* 8 (3): 26–41. https://doi.org/10.1109/65.283931.

Nassif, Ali Bou, Manar Abu Talib, Qassim Nasir, and Fatima Mohamad Dakalbab. 2021. “Machine Learning for Anomaly Detection: A Systematic Review.” *IEEE Access* 9: 78658–700. https://doi.org/10.1109/ACCESS.2021.3083060.

Nehinbe, J. O. 2011. “A Critical Evaluation of Datasets for Investigating IDSs and IPSs Researches.” In *2011 IEEE 10th International Conference on Cybernetic Intelligent Systems (CIS)*, 92–97. London: IEEE. https://doi.org/10.1109/CIS.2011.6169141.

Netti, Kalyan, and Y Radhika. 2015. “A Novel Method for Minimizing Loss of Accuracy in Naive Bayes Classifier.” In *2015 IEEE International Conference on Computational*

Intelligence and Computing Research (ICCIC), 1–4. Madurai, India: IEEE. https://doi.org/10.1109/ICCIC.2015.7435801.

Oyegoke, Adekunle. 2011. "The Constructive Research Approach in Project Management Research." *International Journal of Managing Projects in Business* 4 (4): 573–95. https://doi.org/10.1108/17538371111164029.

Pal Singh, Amrit, and Manik Deep Singh. 2014. "Analysis of Host-Based and Network-Based Intrusion Detection System." *International Journal of Computer Network and Information Security* 6 (8): 41–47. https://doi.org/10.5815/ijcnis.2014.08.06.

Panigrahi, Ranjit, and Samarjeet Borah. 2018. "A Detailed Analysis of CICIDS2017 Dataset for Designing Intrusion Detection Systems." *International Journal of Engineering*, January, 5.

Portnoy, Leonid, Eleazar Eskin, and Salvatore Stolfo. 2001. "Intusion Detection with Unlabeled Data Using Clustering." http://ids.cs.columbia.edu/sites/default/files/cluster-ccsdmsa01.pdf.

Ring, Markus, Sarah Wunderlich, Dominik Grüdl, Dieter Landes, and Andreas Hotho. 2017. "Flow-Based Benchmark Data Sets for Intrusion Detection," 10.

Ring, Markus, Sarah Wunderlich, Deniz Scheuring, Dieter Landes, and Andreas Hotho. 2019. "A Survey of Network-Based Intrusion Detection Data Sets." *Computers & Security* 86 (September): 147–67. https://doi.org/10.1016/j.cose.2019.06.005.

Sabeel, Ulya, Shahram Shah Heydari, Khalid Elgazzar, and Khalil El-Khatib. 2021. "Building an Intrusion Detection System to Detect Atypical Cyberattack Flows." *IEEE Access* 9: 94352–70. https://doi.org/10.1109/ACCESS.2021.3093830.

Sahu, Santosh Kumar, Sauravranjan Sarangi, and Sanjaya Kumar Jena. 2014. "A Detail Analysis on Intrusion Detection Datasets." In *2014 IEEE International Advance Computing Conference (IACC)*, 1348–53. Gurgaon, India: IEEE. https://doi.org/10.1109/IAdCC.2014.6779523.

Sarhan, Mohanad, Siamak Layeghy, Nour Moustafa, and Marius Portmann. 2021. “NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems.” In , 371:117–35. https://doi.org/10.1007/978-3-030-72802-1_9.

Scarfone, K A, and P M Mell. 2007. “Guide to Intrusion Detection and Prevention Systems (IDPS).” NIST SP 800-94. 0 ed. Gaithersburg, MD: National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-94.

Sharafaldin, Iman, Amirhossein Gharib, Arash Habibi Lashkari, and Ali A. Ghorbani. 2017. “Towards a Reliable Intrusion Detection Benchmark Dataset.” *Software Networking* 2017 (1): 177–200. https://doi.org/10.13052/jsn2445-9739.2017.009.

Sharafaldin, Iman, Arash Habibi Lashkari, and Ali A. Ghorbani. 2018. “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization:” In *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 108–16. Funchal, Madeira, Portugal: SCITEPRESS - Science and Technology Publications. https://doi.org/10.5220/0006639801080116.

Shiravi, Ali, Hadi Shiravi, Mahbod Tavallaee, and Ali A. Ghorbani. 2012. “Toward Developing a Systematic Approach to Generate Benchmark Datasets for Intrusion Detection.” *Computers & Security* 31 (3): 357–74. https://doi.org/10.1016/j.cose.2011.12.012.

Shirey, R. 2007. “Internet Security Glossary, Version 2.” RFC4949. Request for Comments. RFC Editor. https://doi.org/10.17487/rfc4949.

Shone, Nathan, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi. 2018. “A Deep Learning Approach to Network Intrusion Detection.” *IEEE Transactions on Emerging Topics in Computational Intelligence* 2 (1): 41–50. https://doi.org/10.1109/TETCI.2017.2772792.

Sommer, Robin, and Vern Paxson. 2010. “Outside the Closed World: On Using Machine Learning for Network Intrusion Detection.” In *2010 IEEE Symposium on Security and Privacy*, 305–16. Oakland, CA, USA: IEEE. https://doi.org/10.1109/SP.2010.25.

Song, Jungsuk, Hiroki Takakura, Yasuo Okabe, Masashi Eto, Daisuke Inoue, and Koji Nakao. 2011. “Statistical Analysis of Honeypot Data and Building of Kyoto 2006+ Dataset for

NIDS Evaluation." In *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security - BADGERS '11*, 29–36. Salzburg, Austria: ACM Press. https://doi.org/10.1145/1978672.1978676.

Sperotto, Anna, Ramin Sadre, Frank van Vliet, and Aiko Pras. 2009. "A Labeled Data Set for Flow-Based Intrusion Detection." In *IP Operations and Management*, 5843:39–50. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-04968-2_4.

Sugeno, M., and T. Yasukawa. 1993. "A Fuzzy-Logic-Based Approach to Qualitative Modeling." *IEEE Transactions on Fuzzy Systems* 1 (1): 7. https://doi.org/10.1109/TFUZZ.1993.390281.

Sulaiman, Noor Suhana, Akhyari Nasir, Wan Roslina Wan Othman, Syahrul Fahmy Abdul Wahab, Nur Sukinah Aziz, Azliza Yacob, and Nooraida Samsudin. 2021. "Intrusion Detection System Techniques : A Review." *Journal of Physics: Conference Series* 1874 (1): 012042. https://doi.org/10.1088/1742-6596/1874/1/012042.

Suleiman, Mohammed F., and Biju Issac. 2018. "Performance Comparison of Intrusion Detection Machine Learning Classifiers on Benchmark and New Datasets." In *2018 28th International Conference on Computer Theory and Applications (ICCTA)*, 19–23. Alexandria, Egypt: IEEE. https://doi.org/10.1109/ICCTA45985.2018.9499140.

Sun, Lijian, Yun Zhou, Yanjuan Wang, Cheng Zhu, and Weiming Zhang. 2020. "The Effective Methods for Intrusion Detection With Limited Network Attack Data: Multi-Task Learning and Oversampling." *IEEE Access* 8: 185384–98. https://doi.org/10.1109/ACCESS.2020.3029100.

Tavallaee, Mahbod, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. 2009. "A Detailed Analysis of the KDD CUP 99 Data Set," 8.

Thakkar, Ankit, and Ritika Lohiya. 2020. "A Review of the Advancement in Intrusion Detection Datasets." *Procedia Computer Science* 167: 636–45. https://doi.org/10.1016/j.procs.2020.03.330.

Tsai, Chih-Fong, Yu-Feng Hsu, Chia-Ying Lin, and Wei-Yang Lin. 2009. “Intrusion Detection by Machine Learning: A Review.” *Expert Systems with Applications*, 7.

Tungjaturasopon, Praiya, and Krerk Piromsopa. 2018. “Performance Analysis of Machine Learning Techniques in Intrusion Detection.” In *Proceedings of the 2018 VII International Conference on Network, Communication and Computing - ICNCC 2018*, 6–10. Taipei City, Taiwan: ACM Press. https://doi.org/10.1145/3301326.3301335.

Uddin, Mueen, Azizah Abdul Rehman, Naeem Uddin, Jamshed Memon, Raed Alsaqour, and Suhail Kazi. 2013. “Signature-Based Multi-Layer Distributed Intrusion Detection System Using Mobile Agents,” 10.

Ullah, Imtiaz, and Qusay H. Mahmoud. 2021. “Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks.” *IEEE Access* 9: 103906–26. https://doi.org/10.1109/ACCESS.2021.3094024.

Verma, J., A. Bhandari, and G. Singh. 2020. “REVIEW OF EXISTING DATA SETS FOR NETWORK INTRUSION DETECTION SYSTEM.” *Advances in Mathematics: Scientific Journal* 9 (6): 3849–54. https://doi.org/10.37418/amsj.9.6.64.

Vigna, G., and R.A. Kemmerer. 1998. “NetSTAT: A Network-Based Intrusion Detection Approach.” In *Proceedings 14th Annual Computer Security Applications Conference (Cat. No.98EX217)*, 25–34. Phoenix, AZ, USA: IEEE Comput. Soc. https://doi.org/10.1109/CSAC.1998.738566.

Vinayakumar, R., Mamoun Alazab, K. P. Soman, Prabaharan Poornachandran, Ameer Al-Nemrat, and Sitalakshmi Venkatraman. 2019. “Deep Learning Approach for Intelligent Intrusion Detection System.” *IEEE Access* 7: 41525–50. https://doi.org/10.1109/ACCESS.2019.2895334.

Virtanen, Aila. 2006. “Konstruktiivinen tutkimusote.” *Ammattikasvatuksen aikakauskirja* 8: 46–52.

Wang, Cheng-Ru, Rong-Fang Xu, Shie-Jue Lee, and Chie-Hong Lee. 2018. “Network Intrusion Detection Using Equality Constrained-Optimization-Based Extreme Learning

Machines." *Knowledge-Based Systems* 147 (May): 68–80. https://doi.org/10.1016/j.knosys.2018.02.015.

Xiao, Xingjiang, and Huafeng Ding. 2012. "Enhancement of K-Nearest Neighbor Algorithm Based on Weighted Entropy of Attribute Value." In *2012 5th International Conference on BioMedical Engineering and Informatics*, 1261–64. Chongqing, China: IEEE. https://doi.org/10.1109/BMEI.2012.6513101.

Yadav, Mukesh Kumar, and Krishna Pal Sharma. 2021. "Intrusion Detection System Using Machine Learning Algorithms: A Comparative Study." In *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)*, 415–20. Jalandhar, India: IEEE. https://doi.org/10.1109/ICSCCC51823.2021.9478086.

Yang, Zhen, Xiaodong Liu, Tong Li, Di Wu, Jinjiang Wang, Yunwei Zhao, and Han Han. 2022. "A Systematic Literature Review of Methods and Datasets for Anomaly-Based Network Intrusion Detection." *Computers & Security* 116 (May): 102675. https://doi.org/10.1016/j.cose.2022.102675.

Yedukondalu, G., G. Hima Bindu, J. Pavan, G. Venkatesh, and A. SaiTeja. 2021. "Intrusion Detection System Framework Using Machine Learning." In *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, 1224–30. Coimbatore, India: IEEE. https://doi.org/10.1109/ICIRCA51532.2021.9544717.

Zhao, Shu-Juan, Hua-Peng Zhang, and Lei Li. 2012. "A New Algorithm for Imbalanced Datasets in Presence of Outliers and Noise." In *2012 8th International Conference on Natural Computation*, 30–34. Chongqing, Sichuan, China: IEEE. https://doi.org/10.1109/ICNC.2012.6234723.