Jani Reinikainen

FAMDAD ja poikkeamien tunnistaminen IoT-verkkoliikenteestä

Tietotekniikan
Pro gradu -tutkielma
7. maaliskuuta 2022

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Kokkolan yliopistokeskus Chydenius

Tekijä: Jani Reinikainen
Yhteystiedot: jani.reinikainen@iki.fi
Puhelinnumero: -
Ohjaaja: Risto Honkanen ja Ismo Hakala
Työn nimi: FAMDAD ja poikkeamien tunnistaminen IoT-verkkoliikenteestä
in English: FAMDAD and IoT Network Traffic Anomaly Detection
Työ: Tietotekniikan Pro gradu -tutkielma
Sivumäärä: 93
Tiivistelmä: Vaikka esineiden internet onkin tapana hyödyntää internetiä vielä suhteellisen uusi, kasvaa käytössä olevien IoT-laitteiden määrä jatkuvasti. Samalla kun nämä laitteet tulevat yhä enemmän osaksi jokapäiväistä elämäämme, korostuu niiden tietoturvan merkitys. Poikkeamia verkkoliikenteestä tunnistava tunkeutumisen havaitsemisjärjestelmä voi osaltaan parantaa tietoturvaa hälyttämällä poikkeavasta verkkoliikenteestä. Tässä työssä selvitettiin, miten hyvin FAMDAD-menetelmä soveltuu puoliohjattuun ja ohjaamattomaan poikkeaman tunnistukseen ensisijaisesti IoT-verkoista kerätyistä liikennevirtatietueisiin pohjautuvista aineistoista. Työn empiirisen osuuden tulosten perusteella FAMDAD-menetelmällä saatujen tulosten ei voitu osoittaa poikkeavan tilastollisesti merkitsevästi Mahalanobiksen etäisyydellä ja autoenkoodereihin perustuneella menetelmällä saaduista tuloksista.
Avainsanat: autoenkooderi, FAMD, FAMDAD, IoT, Mahalanobiksen etäisyys, PCA, poikkeaman tunnistus, TCP/IP, tietoturva, tunkeutumisen havaitsemisjärjestelmä
Abstract: Although the Internet of Things is still relatively new, the number of IoT devices in use is constantly increasing. As these devices become more and more ubiquitous, the importance of their security is being emphasized. An intrusion detection system that detects anomalies from network traffic can improve security by alerting about anomalous network traffic. The suitability of FAMDAD for semi-supervised and unsupervised anomaly detection from network traffic flow record based data collected primarily from IoT networks was investigated in this work. Based on the results of the empirical comparison, it could not be shown that the results obtained using FAMDAD differ statistically significantly from the results obtained by Mahalanobis distance and simple autoencoders.
Keywords: autoencoder, FAMD, FAMDAD, information security, intrusion detection system, IoT, Mahalanobis distance, PCA, anomaly detection, TCP/IP

Sanasto

AE	Autoencoder
AMQP	Advanced Message Queuing Protocol
ARP	Address Resolution Protocol
AUC	Area Under the ROC Curve
CoAP	Constrained Application Protocol
DDoS	Distributed Denial of Service
DDS	Data Distribution Service
DNS	Domain Name System
DoS	Denial of Service
DTLS	Datagram Transport Layer Security
FAMD	Factor Analysis of Mixed Data
FAMDAD	Factor Analysis of Mixed Data for Anomaly Detection
FN	False Negatives
FP	False Positives
FPR	False Positive Rate
HIDS	Host-based Intrusion Detection System
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IDPS	Intrusion Detection and Prevention Systems
IDS	Intrusion Detection System
IIoT	Industrial Internet of Things
IoT	Internet of Things
IP	Internet Protocol
IPFIX	IP Flow Information Export
IPS	Intrusion Prevention System
KDD	Knowledge Discovery and Data Mining
K-NN	K-Nearest Neighbor

LPWAN	Low Power Wide Area Network
LR-WPAN	Low-Rate Wireless Personal Area Network
MAC	Media Access Control
MIT	Massachusetts Institute of Technology
MITM	Man-in-the-middle
MQTT	Message Queuing Telemetry Transport
NAT	Network Address Translation
NBA	Network Behavior Analysis
NDP	Neighbor Discovery Protocol
NIDS	Network Intrusion Detection System
PCA	Principal Component Analysis
RFID	Radio Frequency Identification
SPAD	Simple Probabilistic Anomaly Detector
SPE	Squared Prediction Error
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TN	True Negatives
TP	True Positives
TPR	True Positive Rate
UDP	User Datagram Protocol
UNSW	University of New South Wales
WIDS	Wireless Intrusion Detection System
WLAN	Wireless Local Area Network
XSS	Cross-Site Scripting
ZCA	Zero-Phase Component Analysis

Sisältö

1 Johdanto

Tietoliikenne- ja siruteknologian nopea kehittyminen yhdessä lisääntyneen kysynnän kanssa on mahdollistanut internetiin yhdistettyjen esineiden sovelluskohteiden määrän kasvamisen usealla eri toimialalla [149]. Samalla kun nämä IoT-laitteet (Internet of Things, IoT) ovat kehittyneet ja niitä on ryhdytty käyttämään logistiikka-alan lisäksi myös muun muassa maataloudessa, teollisuudessa sekä älykkäissä kodeissa ja kaupungeissa, on niiden määrä kasvanut nopeasti [48, 149]. IoT-laitteiden jokapäiväistyminen tekeekin niiden tietoturvasta tärkeän tutkimusalueen [48].

Vaikka esineiden internetin tietoturvauhat voivat eri sovelluskohteissa olla erilaisia, on hyvä huomata, että internetin käyttöön IoT-laitteissa liittyy aina vastaavia uhkia kuin perinteisissä tietokoneissa. Koska kaikki laitevalmistajat eivät aina ole ottaneet näitä uhkia tosissaan, on osaan laitteista voitu murtautua esimerkiksi telnet-yhteyden yli laitteen oletussalasanan avulla [10]. Tietoturvauhilta suojautumisen lisäksi voidaan myös pyrkiä löytämään merkkejä niiden toteutumisesta. Jos tällaisia merkkejä etsivä tunkeutumisen havaitsemisjärjestelmä hyödyntää vain verkkoliikennettä, ei tarkkailtaviin IoT-laitteisiin tarvitse tehdä muutoksia.

Verkkoliikennettä ainakin jollain tasolla hyödyntäviä tunkeutumisen havaitsemisjärjestelmiä on useita, ja ne eroavat toisistaan esimerkiksi käyttämänsä analysointimenetelmän osalta [15]. Tunnettuja väärinkäytöksiä etsivät järjestelmät pyrkivät löytämään analysoimistaan tietolähteistä hyökkäyksiin liittyviä sormenjälkiä. Vaikka ne tuottavat vain vähän vääriä hälytyksiä, eivät ne voi löytää niille uusia väärinkäytöksiä. Poikkeamia eli myös täysin uusia hyökkäyksiä tunnistavan järjestelmän kyky erottaa väärinkäytökset normaalista riippuu sekä sen tietolähteiden tuottamista tiedoista että sen käyttämästä poikkeaman tunnistusmenetelmästä.

Denningin [39] vuonna 1987 esittämän tilastollista poikkeaman tunnistusmenetelmää hyödyntäneen tunkeutumisen havaitsemisjärjestelmän julkaisun jälkeen on tehty valtava määrä tutkimuksia, joissa ollaan esitetty tai vertailtu eri tekniikoita hyödyntäviä poikkeaman tunnistusmenetelmiä tunkeutumisen havaitsemisessa. Eräs viime aikoina esitetty poikkeaman tunnistusmenetelmä, jota ei vaikuttaisi olevan juurikaan tutkittu sen julkaisemisen jälkeen, on Davidowin ja Mattesonin [35] esittämä pääkomponenttianalyysiin perustuva FAMDAD-menetelmä.

1.1 Tutkimuskysymys

Tässä työssä pyritään selvittämään, miten hyvin Davidowin ja Mattesonin [35] esittämä FAMDAD soveltuu puoliohjattuun ja ohjaamattomaan poikkeaman tunnistukseen ensisijaisesti IoT-verkoista kerätyistä liikennevirtatietueisiin pohjautuvista aineistoista. Menetelmää arvioidaan sekä teoreettisesti että empiirisesti vertaamalla sitä Mahalanobiksen etäisyyteen ja yksinkertaiseen autoenkoodereita hyödyntäneeseen menetelmään. Suorituskykymittarina vertailussa käytetään menetelmien tuottamista anomaliapisteytyksistä laskettavaa ROC-käyrän alle jäävää pinta-alaa.

Vaikka juuri FAMDAD-menetelmää useassa eri aineistossa Mahalanobiksen etäisyyteen [84] tai autoenkoodereihin vertailevia tutkimuksia ei näyttäisikään olevan tehty, on pääkomponenttianalyysiä ja autoenkoodereita hyödyntäneitä menetelmiä käytetty verkkoliikenteestä tapahtuvaan poikkeaman tunnistukseen jo muun muassa Lakhina et al. [78] ja Hawkins et al. [58] toimesta. Pääkomponenttianalyysiä ja autoenkoodereita hyödyntäneiden menetelmien avulla saatuja tuloksia on myös vertailtu empiirisesti [27]. Tosin ainakin harvemmin vertailuun on käytetty useampaa myös IoT-laitteiden verkkoliikenteestä muodostettua aineistoa.

1.2 Tutkielman rakenne

Aluksi luvussa 2 luodaan katsaus esineiden internetiin, ja verrataan sitä perinteiseen internetiin. Kun esineiden internetin käsite on määritelty tarkemmin, käydään luvussa vielä läpi sen arkkitehtuureja, tietotoliikennettä ja tietoturvaa. Luvussa 3 tarkastellaan muun muassa verkkohyökkäyksiä havaitsevia tunkeutumisen havaitsemisjärjestelmiä lähinnä NIST 800-94 -erityisjulkaisun mukaisesti. Työn yleisemmän osuuden päättää luku 4, jossa luodaan tarkempi katsaus poikkeamien, joiksi tunkeutumisetkin voidaan nähdä, tunnistamiseen verkkoliikenteestä.

Luvussa 5 tutustutaan tarkemmin sekä FAMDAD-menetelmään että tässä työssä sen vertailukohtina käytettyihin muihin menetelmiin. Luvussa 6 taas esitellään ensin valittujen menetelmien vertailuun käytetyt aineistot. Tämän jälkeen tutkimuskysymykseen haetaan vastausta vertaamalla FAMDAD-menetelmällä saatuja tuloksia sekä Mahalanobiksen etäisyydellä että autoenkoodereihin perustuneella yksinkertaiseen menetelmällä saatuihin tuloksiin. Ennen luvun 7 yhteenvetoa luvussa 6 käydään vielä läpi vertailluilla menetelmillä puoliohjatussa ja ohjaamattomassa poikkeaman tunnistuksessa saadut tulokset.

2 Esineiden internet

Tapamme hyödyntää internetiä on kehittynyt pitkälti laitteiden ehdoilla [82]. Ensimmäisessä vaiheessa käyttäjien oli mentävä internetiin kytketyn laitteen, joka oli tyypillisesti henkilökohtainen tietokone, luokse päästäkseen internetiin. Langattomien teknologioiden kehittyessä ja yleistyessä internetiin saattoi jo päästä sijainnista riippumatta. Nykyisin internetiä hyödyntävät myös useat erilaiset esineet tai laitteet, jotka voivat lähettää sensoreidensa avulla keräämiään mittaustuloksia internetissä oleviin palveluihin ja toimia niiltä saamiensa ohjeiden mukaisesti.

Alkujaan esineiden internet eli IoT on Kevin Ashtonin 1990-luvulla keksimä termi, jota hän käytti vuonna 1999 laatimassaan esityksessä, jossa hän yhdisti internetin ja radiotaajuisen etätunnistuksen (Radio Frequency Identification, RFID) käytön Procter & Gamble -yrityksen toimitusketjussa [14, 48]. Eräs uudempi määritelmä IoT-laitteelle on järjestelmä, joka sisältää joukon sensoreita ja toimilaitteita ja joka voi liittyä internetiin joko suoraan tai jonkin toisen laitteen välityksellä [48]. Mitään universaalia määritelmää IoT:lle ei kuitenkaan ole [82, 145].

Vaikka esineiden internet onkin tapana hyödyntää internetiä vielä suhteellisen uusi, kasvaa käytössä olevien IoT-laitteiden määrä jatkuvasti. IDC:n ennusteen mukaan vuonna 2025 maailmassa on 55,7 miljardia yhdistettyä laitetta [63], joka tarkoittaa yli kymmenen prosentin vuosittaista kasvua yhdistettyjen laitteiden määrässä, kun vertailukohtana käytetään vuotta 2010, jolloin internetiin yhdistettyjen laitteiden määrän arvioidaan olleen 12,5 miljardia [48]. IDC ennustaa lisäksi IoT-laitteiden, joista se odottaa 75 % olevan yhteydessä johonkin IoT-alustaan, tuottavan 73,1 miljardia teratavua tietoa vuodessa vuoteen 2025 mennessä [63].

Vaikka periaatteessa älytelevision, matkapuhelimen ja näppäimistöllä varustetun tietokoneenkin on mahdollista täyttää aiemmin esitetty IoT-laitteen määritelmä, voidaan perinteisen internetin ja esineiden internetin väliltä löytää myös eroavaisuuksia. Esimerkiksi toisin kuin perinteisessä internetissä, jonka sisältö on tyypillisesti ollut ihmisten tuottamaa ja kuluttamaa, esineiden internetissä myös laitteet itse tuottavat sisältöä [44]. Usein tällaiset IoT-laitteet ovat myös suunniteltu jotain tiettyä käyttötarkoitusta varten, jolloin niiden normaali toiminta on helpommin tunnistettavissa kuin yleiskäyttöisen ihmisen operoiman laitteen [56].

Tavallisten esineiden, kuten lamppujen, videokameroiden, jääkaappien ja sähkömittareiden, yksilöimistä ja varustamista sensoreilla ja toimilaitteilla sekä verkkomoduuleilla voidaan myös pitää IoT:n perusajatuksena [145]. Esineiden internetin laitteita onkin useita, ja ne ovat varsin erilaisia, sillä sekä yksinkertaisella passiivisella RFID-tunnisteella varustettu pakkaus että useilla eri sensorilla ja toimilaitteella varustettu matkapuhelin voivat olla IoT-laitteita. Niitä käytetään myös useilla eri toimialoilla, joista suurin on julkispalvelut, jonka osuutta nostaa älykäs sähkön mittaus, noin 24 %:n osuudella IoT-laitekannasta [53]. Eniten sisältöä taas tuottavat turvallisuuteen ja videovalvontaan liittyvät IoT-laitteet [63].

2.1 Esineiden internetin arkkitehtuuri

Esineiden internetiä voidaan hyödyntää hyvin erilaisissa sovelluskohteissa useilla eri toimialoilla [82]. Esimerkiksi kannettava RFID-lukija voi lukea pakkaukseen kiinnitetyn RFID-tunnisteen, ja hakea siihen liittyvät tiedot langattoman lähiverkon (Wireless Local Area Network, WLAN) kautta sovellukseen liittyvästä pilvipalvelusta, ja esittää ne käyttäjälle lukijan käyttöliittymässä. Toisaalta IoT-laite voi olla myös esimerkiksi sääasema, joka vain lähettää keräämänsä mittaustiedot säännöllisesti suoraan LoRaWAN-yhdyskäytävän kautta internetissä olevalle LoRaWAN-alustalle. Käytettyjen laitteiden ja verkkojen lisäksi sovellukset voivat erota toisistaan myös esimerkiksi vasteaikavaatimusten suhteen [82].

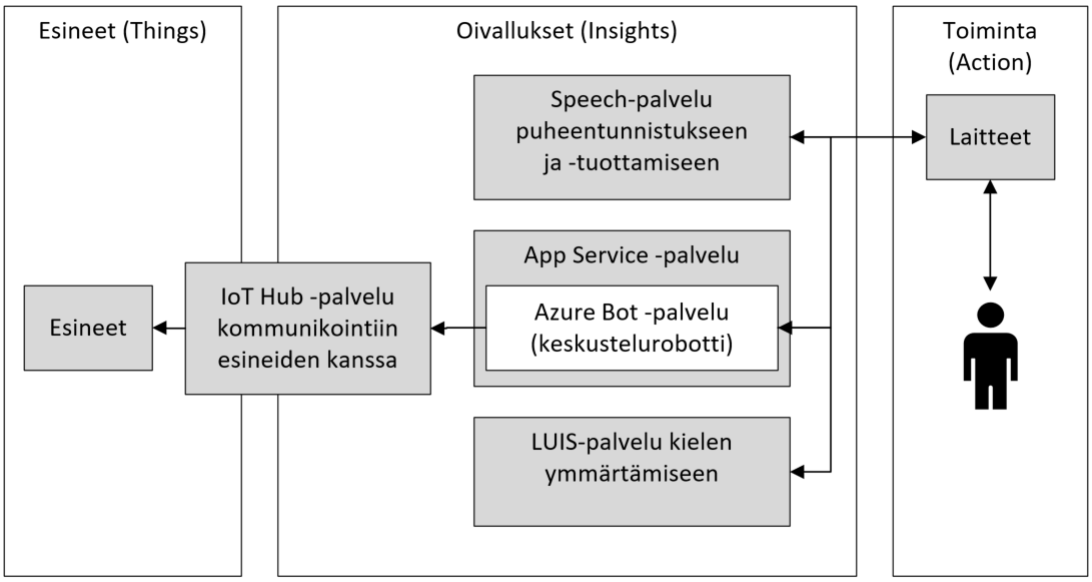
Esineiden internetille onkin esitetty sekä useita arkkitehtuureja että referenssiarkkitehtuureja [5, 80, 82]. Tyypillisesti esitetyt arkkitehtuurit ovat kerrosarkkitehtuureja [5, 80], joista ehkä yleisimmässä on kolme kerrosta, jotka alhaalta ylöspäin lukien ovat havainnointi-, verkko- ja sovelluskerros [80]. Tässä havainnointikerrokseen voidaan katsoa kuuluvan esineiden internetin esineiden toiminnot, jotka usein liittyvät reaalimaailman aistimiseen ja muuttamiseen sensoreiden ja toimilaitteiden avulla. Verkkokerroksen tarkoituksena taas on välittää tietoa sekä esineiden internetin esineiden että muiden järjestelmien välillä. Sovelluskerroksen puolestaan voidaan katsoa tarjoavan verkkokerroksen päällä olevat palvelut ja toiminnot, joita tarvitaan muun muassa tietojen tallentamiseen ja analysointiin.

Myös erilaisia referenssiarkkitehtuureja on useita, ja niitä on määritelty useiden eri tahojen toimesta [82]. Esimerkiksi IEEE-standardissa 2413-2019 määritellään arkkitehtuurikehikko, jonka osana kuvataan IoT-järjestelmän komponenttien valmiudet, jotka niiden olisi toivottavaa tarjota standardien rajapintojen muodossa, jotta

myös eri tavoilla toteutetuista komponenteista saataisiin muodostettua toimivia järjestelmiä [64]. Standardissa lueteltuihin valmiuksiin sisältyvät havainnoinnin, toimimisen sekä tiedon tallentamisen, siirtämisen ja prosessoinnin lisäksi myös muun muassa käyttöliittymän ja rajapinnan tarjoaminen.

Havainnointivalmiuden omaavat komponentit saavat syötteenään aistimassaan muodossa olevaa energiaa, jonka ne muuttavat sähköenergiaksi, jolle vielä tehdään analogia-digitaalimuunnos [64, 34]. Toimimisvalmiuden omaavat komponentit taas muuttavat digitaalisessa muodossa olevaa tietoa sellaisessa muodossa olevaksi energiaksi, jolla ne pystyvät muuttamaan ympäristönsä tilaa suunnitellusti. Tietojen tallentamiseen, siirtämiseen ja prosessointiin liittyvät toiminnot käyttävät syötteenään ja palauttavat tulosteenaan vain digitaalisessa muodossa olevaa tietoa. Kerrosarkkitehtuureissa havainnointi- ja toimimisvalmius sijaitsisivat havainnointikerroksella, josta usein käytetään myös nimeä laitekerros [80].

Kaupallisista toimijoista esimerkiksi Microsoft ja Amazon ovat myös laatineet omia referenssiarkkitehtuurejaan tarjoamiaan pilvipalvelualustoja hyödyntäville IoT-sovelluksille [7, 90]. Kuvassa 2.1 on esitetty Azure IoT -referenssiarkkitehtuuriin ja lähteeseen [91] pohjautuva arkkitehtuuri sovellukselle, joka hyödyntää ääniohjausta IoT-laitteiden (esineet) ohjaamiseen. Siinä ei hyödynnetä referenssiarkkitehtuurin sisältynyttä reunalaskentaa, jolla pilvessä tapahtuvan laskennan viivettä saadaan pienennettyä suorittamalla sitä lähempänä IoT-laitteita verkon reunalla [64].



Kuva 2.1: Microsoftin referenssiarkkitehtuurin perustuva arkkitehtuuri

2.2 Esineiden internetin tietoliikenne

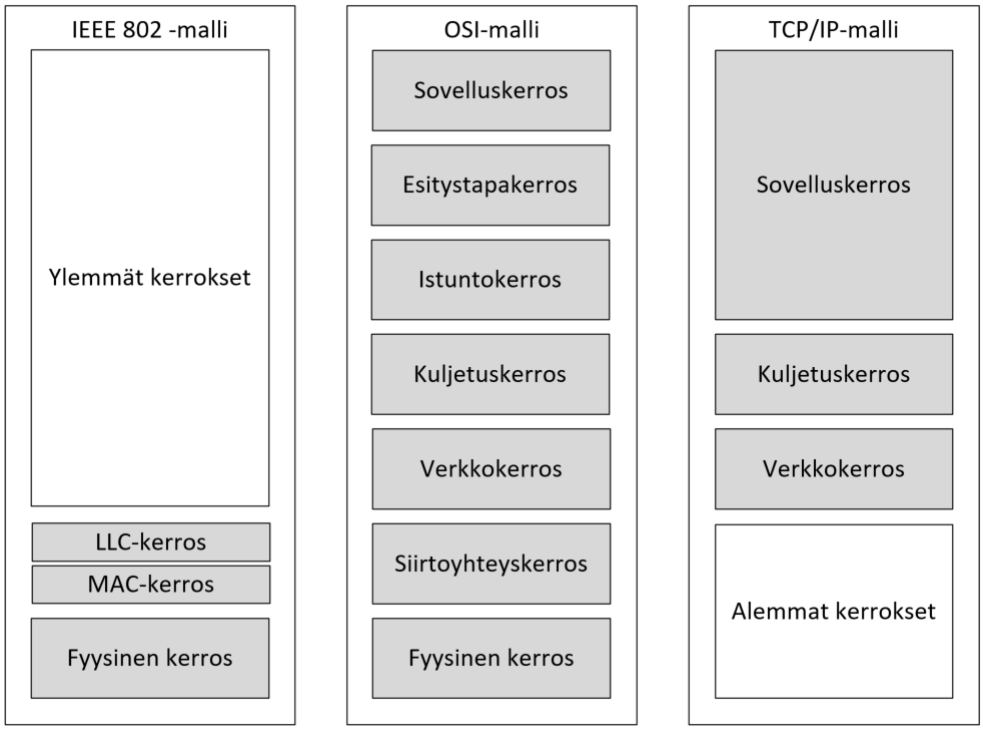
Esineiden internetin esineet voidaan suunnitella käyttämään joitain useista eri verkkoteknologioista, joilla kaikilla on omat hyvät ja huonot puolensa. Esimerkiksi siinä missä älytelevisio saattaa yleiskäyttöisten tietokoneiden tapaan hyödyntää Ethernet- tai WLAN-verkkoa, voi kaukaista tulivuorta monitoroiva IoT-laite käyttää sijoituspaikan asettamien rajoitteiden ohjaamana esimerkiksi jotain alhaisen tehon suurialueverkkoa (Low Power Wide Area Network, LPWAN). Sovelluksissa on siis huomioitava, että eri verkkoteknologiat eroavat toisistaan muun muassa kantaman, kaistanleveyden, luotettavuuden, viiveen ja energiatehokkuuden suhteen [127].

Eräitä IoT-laitteiden alhaisemmat prosessointi- ja energiaresurssit huomioivia tiedonsiirtoteknologioita RFID:n lisäksi ovat IEEE 802.15.4 -standardissa määritelty matalanopeuksinen langaton likiverkko (Low-Rate Wireless Personal Area Network, LR-WPAN) sekä LoRa ja Sigfox, jotka molemmat ovat lisenssivapailla radiotaajuuksilla toimivia LPWAN-teknologioita [65, 110]. Myös perinteiset Bluetooth-, WLAN- ja matkapuhelinverkkoteknologiat ovat kehittyneet paremmin IoT-laitteiden tarpeet huomioiviksi muun muassa Bluetooth Low Energy, IEEE 802.11ah, IEEE 802.11ax, 3GPP eMTC ja 3GPP NB-IoT määritysten myötä [110].

Radiotaajuisessa etätunnistuksessa RFID-lukija lukee RFID-tunnisteen sisältämät tiedot radioaaltojen avulla. Tunnisteet ovat aktiivisia, jos niillä on oma virtalähde, ja passiivisia, jos ne indusoivat tarvitsemansa energian lukijan antenniinsa lähettämistä radioaalloista. Esimerkiksi toimitusketjujen hallinnassa RFID-tunnisteita voidaan, toisin kuin viivakoodeja, lukea yhtä aikaa ilman näköyhteyttä. Tunnisteisiin myös mahtuu enemmän tietoa kuin viivakoodeihin, ja osaan niistä voidaan myös tallentaa tietoja RFID-lukijan avulla. Tyypillisesti RFID-tunnisteet kuitenkin sisältävät vain jonkin esineen yksilöivän tunnisteen [143].

IEEE 802.15.4 -standardi määrittää kuvassa 2.2 esitetystä IEEE 802 -mallista, jossa kuvataan 802-sarjan standardien laajuus, fyysisen kerroksen lisäksi myös siirtotielle pääsyn hallintaan (Media Access Control, MAC) liittyvän MAC-kerroksen [65]. Koska se on määritelty alhaista tiedonsiirtonopeutta vaativille laitteille, joilla on rajalliset energiavarat, ja koska se tukee tähtitopologian lisäksi myös vertaisverkkotopologiaa, soveltuu se käytettäväksi myös langattomissa sensoriverkoissa, joissa viestejä reititetään laitteilta toisille. Verkon muodostus ja reititys jätetään standardissa kuitenkin ylempien kerrosten vastuulle. Muun muassa verkon pienestä hyötykuorman koosta johtuen, sen päällä ei voida suoraan käyttää IPv6-protokollaa, ja sitä varten IETF onkin määrittänyt 6LoWPAN-sovituskerroksen [76].

Vaikka sekä LoRa että Sigfox määrittelevät IEEE 802.15.4 -standardin tavoin lisenssivapailla radiotaajuuksilla toimivan IoT-laitteille sopivan verkon fyysisen kerroksen, eroavat ne LPWAN-verkkoina siitä kuitenkin muun muassa siten, että niiden kantamaa on kasvatettu tiedonsiirtonopeuden kustannuksella [65, 129]. Kohinaisen kanavan kapasiteetti, joka siis asettaa teoreettisen ylärajan tiedonsiirtonopeudelle, saadaan Shannon-Hartley-teoreeman mukaan $C = B\,log_2(1 + \frac{S}{N})$, kun B on kaistanleveys ja $\frac{S}{N}$ signaali-kohinasuhde vastaanottimessa, joka luonnollisesti laskee lähetysetäisyyden kasvaessa [129]. LoRaWAN puolestaan määrittelee Semtechin patentoimaa LoRa-modulaatiota hyödyntävän fyysisen kerroksen päälle soveltuvan MAC-kerroksen [88, 129]. LoRan tavoin myös Sigfox on patentoitu [88].



Kuva 2.2: IEEE 802 -malli, OSI-malli ja TCP/IP-malli [66, 119]

Riippumatta siitä millaisia teknologioita ja protokollia IoT-laitteiden verkoissa käytetään, joissain vaiheessa niiden tuottaman tai tarvitseman tiedon on reitityttävä Internet-protokollaa (Internet Protocol, IP) verkkokerroksella hyödyntävässä verkossa, sillä esineiden internetin esineiden on oltava vähintään epäsuorasti yhteydessä internetiin. Kuvassa 2.2 on esitetty OSI-mallin lisäksi myös IEEE 802 -malli sekä internetissä käytetty TCP/IP-malli, joka ei määrittele fyysistä kerrosta eikä sil-

le pääsystä vastaavaa siirtoyhteyskerrosta [37, 108, 112, 119]. TCP/IP-malli eroaa OSI-mallista myös siten, että istuntojen ja erilaisten esitystapojen, kuten merkistökoodausten, hallinnalle ei ole siinä määritelty erillisiä kerroksia [108].

2.2.1 TCP/IP-mallin verkko- ja kuljetuskerros

TCP/IP-mallissa verkkokerroksen tärkein protokolla on IP, sillä siinä muiden protokollien tietoa välitetään paketeissa, jotka sisältävät lähettäjän eli lähteen osoitteen lisäksi myös reitityksen kannalta oleellisen tiedon vastaanottajan eli kohteen osoitteesta. Käytännössähän reitittimet eli laitteet, jotka välittävät IP-paketteja eri verkkojen välillä, perustavat reitityspäätöksensä juuri paketin kohdeosoitteen alkuosaan tai siitä johdettuihin tietoihin. Lähdeosoite puolestaan kertoo IP-pakettiin mahdollisesti kohdistuviin vastauspaketteihin tulevan kohdeosoitteen. [20]

Tällä hetkellä internet-protokollasta on käytössä kaksi eri versiota, joista uudemmassa IPv6:ssa osoitteen pituutta on kasvatettu vanhemman IPv4:n 32 bitistä 128 bittiin [37]. Vaikka 32 bitin jono voikin saada vain $2^{32} \approx 4,3$ miljardia eri arvoa, voidaan IPv4-osoitteiden rajallisuutta kiertää käyttämällä osassa verkkoja yksityiseen käyttöön tarkoitettuja IP-osoitteita, joiden ei tarvitse olla globaalisti yksilöiviä [116]. Koska tällaisia osoitteita sisältäviä paketteja ei voida reitittää internetiin, täytyy niille tehdä osoitemuunnos (Network Address Translation, NAT), jossa yksityinen IP-osoite korvataan julkisella tai päin vastoin liikenteen suunnasta riippuen, olettaen, että tällaisen osoitteen omaavan laitteen täytyy olla yhteydessä internetiin [131].

Muita TCP/IP-mallin verkkokerroksen protokollia ovat muun muassa IPv4:ään liittyvät ICMP ja ARP sekä IPv6:een liittyvät ICMPv6 ja NDP. Näistä sekä ICMP että ICMPv6 ovat IP:n päällä toimivia protokollia, joita käytetään esimerkiksi pakettien välittämiseen liittyvien virheiden raportointiin ja selvittämiseen [20, 32]. Osoitteiden selvittämiseen (Address Resolution Protocol, ARP) ja naapurien löytämiseen (Neighbor Discovery Protocol, NDP) tarkoitettuja protokollia käytetään IP-osoitteita vastaavia siirtoyhteyskerroksen osoitteita selvitettäessä [20, 97]. MAC-osoitteita tarvitaan, kun IP-paketti sisällytetään käytetyn siirtoyhteyskerroksen kehykseen [20].

Kuljetuskerroksen tärkeimmät protokollat ovat puolestaan UDP (User Datagram Protocol) ja TCP (Transmission Control Protocol). Vaikka sekä UDP-paketit että TCP-segmentit sisältävät laitteen sisällä olevan sovellusprosessin yksilöintiin tarvittavan lähde- ja kohdeportin ja vaikka ne molemmat sisällytetään IP-pakettien hyötykuormaan, eroavat protokollat toisistaan siinä, että UDP tarjoaa epäluotettavan ja TCP luotettavan kuljetuskerroksen. Koska TCP-yhteyden muodostukseen tarvitaan kol-

mitiekättelyä, jossa yhteyden muodostaja lähettää kaksi ja yhteyden kohde yhden IP-paketin, ja koska muodostetussa yhteydessä, joka pitää vielä erikseen sulkea, lähetetyt tiedot kuitataan tiedonsiirron luotettavuuden takaamiseksi, on TCP raskaampi kuin yhteydetön UDP varsinkin pieniä tietomääriä siirrettäessä. [20]

2.2.2 TCP/IP-mallin sovelluskerros

Sovelluskerroksen protokollat toimivat TCP/IP-mallin kuljetuskerroksen päällä, ja niiden ohjelmointiin voidaan muun muassa Windows- ja Unix-käyttöjärjestelmissä käyttää pistoke-ohjelmointirajapintaa, jonka avulla sovellusohjelmat voivat esimerkiksi muodostaa ja sulkea TCP-yhteyksiä sekä lähettää ja vastaanottaa UDP-paketteja ja TCP-segmenttejä [75, 92]. Sovellukset voivat sen avulla myös ilmoittaa haluavansa käsitellä tiettyyn porttiin saapuvia TCP-yhteyksiä tai UDP-paketteja [75]. Esimerkiksi HTTP-palvelinohjelma voi ilmoittaa käynnistyessään haluavansa käsitellä kaikki tunnettuun porttiin 80 saapuvat yhteydet.

Luonnollisesti myös eri sovellusprotokollille voidaan toteuttaa ohjelmistokirjastoja. Eräitä tärkeitä sovelluskerroksen protokollia ovat tyypillisesti TCP:n päällä toimivat TLS (Transport Layer Security) ja HTTP (Hypertext Transfer Protocol) sekä yleensä UDP:n päällä toimiva DNS (Doman Name System) [47, 93, 118]. Näistä TLS 1.3 tarjoaa muille sovelluskerroksen protokollille salatun tiedonsiirtokanavan, jota muodostettaessa ainakin asiakas tunnistaa palvelimen [118]. DNS-protokollaa taas tarvitaan esimerkiksi, kun HTTP-pyyntöä TLS-kanavan yli valmisteleva laite selvittää DNS-palvelimelta HTTP-palvelimen DNS-osoitetta vastaavan IP-osoitteen.

Vaikka HTTP-pohjaiset palvelut ovatkin internetissä suosittuja, ei TCP:tä kuljetuskerroksella hyödyntävä HTTP sellaisenaan sovi kovin hyvin käytettäväksi rajoitetut resurssit omaavien IoT-laitteiden muodostamissa verkoissa. HTTP:n kaltainen UDP:n päällä toimiva CoAP (Constrained Application Protocol) soveltuukin sitä paremmin tällaisiin ympäristöihin. Vaikka esimerkiksi välityspalvelimen onkin mahdollista suorittaa protokollamuunnos CoAP:n ja HTTP:n välillä, eroavat protokollat toisistaan esimerkiksi siten, että CoAP-pyynnössä määritetään, halutaanko siihen kuittaus. Koska CoAP toimii UDP:n päällä, käytetään siinä TLS:n sijaan myös UDP:n päällä toimivaa DTLS:ää (Datagram Transport Layer Security). [124]

Muita esineiden internetin sovelluksissa käytettyjä sovelluskerroksen protokollia ovat muun muassa julkaise ja tilaa -mallin toteuttavat MQTT (Message Queuing Telemetry Transport), DDS (Data Distribution Service) ja AMQP (Advanced Message Queuing Protocol). Esimerkiksi MQTT-protokollaa käyttävässä IoT-sovelluksessa

havainnointivalmiuden omaavat laitteet voisivat muodostaa TCP-yhteyden MQTT-välittäjään, ja julkaista sille tiettyyn aiheeseen liittyviä viestejä, jotka se sitten edelleen välittäisi erillisten TCP-yhteyksien avulla tilaajille eli laitteille, jotka ovat ilmoittaneet olevansa kiinnostuneita kyseisistä aiheista [132].

2.3 Esineiden internetin tietoturva

Koska esineiden internetin esineet ovat ainakin epäsuorasti yhteydessä internetiin, voidaan IoT-sovelluksiin kohdistaa niiden ominaispiirteitä hyödyntävien hyökkäysten lisäksi myös samanlaisia hyökkäyksiä kuin perinteisiin internetiä hyödyntäviin sovelluksiin. Esineiden internetille ominaiset tietoturvahaasteet voivat liittyä muun muassa IoT-laitteiden rajoitettuihin resursseihin, verkkoteknologioiden suureen kirjoon, asetusten ja salausavainten jakamiseen, ohjelmistojen päivittämiseen sekä yksityisyyden suojaamiseen ja tietovuotojen estämiseen [52, 122]. Osa haasteista näkyy myös OWASP-järjestön julkaisemalla kymmenen tärkeimmän IoT-järjestelmän elinkaaren aikana vältettävän asian listalla [137]:

1. Helposti murrettavien, julkisten tai kovakoodattujen salasanojen käyttö
2. Tarpeettomien tai haavoittuvien verkkopalveluiden ajaminen IoT-laitteissa
3. Haavoittuvat tausta- ja oheisjärjestelmien rajapinnat sekä Web-käyttöliittymät
4. Turvallisen ohjelmistopäivitysmekanismin puuttuminen
5. Haavoittuvien tai vanhentuneiden ohjelmistokomponenttien käyttö
6. Henkilökohtaisten tietojen huolimaton tai luvaton käyttö
7. Tiedonsiirto tai -tallentaminen ilman asiallista pääsynvalvontaa ja salausta
8. Laitehallinnan puuttuminen
9. Haavoittuvien oletusasetusten käyttö
10. IoT-laitteiden puutteellinen fyysinen suojaus

Vaikka tunnettujen salasanojen käyttäminen ja tarpeettomien verkkopalveluiden ajaminen IoT-laitteissa kuulostavatkin varsin itsestään selvästi huonoilta ratkaisuilta, liittyvät ne silti lukuisiin tietoturvaloukkauksiin. Esimerkiksi vuonna 2016 Mirai-haittaohjelma onnistui levittäytymään useisiin ssh- tai telnet-palvelua ajaneisiin IoT-laitteisiin, joiden joukossa oli muun muassa reitittimiä, valvontakameroita ja tulos-

timia, yksinkertaisesti kirjautumalla niihin tuntemansa 62 käyttäjätunnus-salasanaparin avulla [10, 74]. Onnistuneen kirjautumisen jälkeen Mirai lähetti sekä laitteen IP-osoitteen että käyttäjätunnuksen ja salasanan erilliselle raportointipalvelimelle, joka käynnisti varsinaisen hyökkäyksen, jossa laitteelle sopiva haittaohjelmaversio ladattiin internetistä käyttöjärjestelmän wget- tai tftp-komennolla [10].

Asennuttuaan Mirai muodosti yhteyden saastuneiden IoT-laitteiden muodostamaa bottiverkkoa hallinnoivaan palvelimeen pyrkien samalla myös aktiivisesti etsimään muita haavoittuvia laitteita muodostamalla uusia ssh- tai telnet-yhteyksiä satunnaisesti valitsemiinsa IP-osoitteisiin. Vuonna 2016 Mirai-bottiverkkoja käytettiinkin useisiin hajautettuihin palvelunestohyökkäyksiin, joissa niitä hallinnoivat tahot komensivat sadat tuhannet saastuneet IoT-laitteet suorittamaan muun muassa SYN- ja HTTP-tulvalla toteutettuja hyökkäyksiä, joiden eräitä kohteita olivat Krebs on Security -sivusto sekä Dyn-nimipalvelu. [10]

Tilanteen ongelmallisuutta lisäsi myös se, että vaikka IoT-laitteen salasanan olisikin käynyt muuttamassa Web-käyttöliittymässä tai vastaavassa, ei se olisi välttämättä poistanut tunnettujen salasanojen käytön aiheuttamaa ongelmaa, sillä joissain laitteissa esimerkiksi telnet-yhteydessä voitiin edelleen käyttää laitteeseen kovakoodattua järjestelmän pääkäyttäjän salasanaa [25, 74]. Toki näissäkin tilanteissa haittaohjelmalta olisi voinut vielä välttyä, jos IoT-laitteen ja internetin välissä olisi ollut palomuuri, joka olisi suodattanut IoT-laitteen ajamien tarpeettomien verkkopalveluiden käyttämiin TCP-portteihin internetistä lähetetyt IP-paketit.

2.3.1 Eräitä IoT-laitteisiin kohdistettuja verkkohyökkäyksiä

IoT-sovelluksiin kohdistetut verkkohyökkäykset voivat liittyä useisiin OSI-mallin kerroksiin, ja niissä voidaan hyödyntää järjestelmän tai sen osien suunnittelussa, määrittelyssä, toteutuksessa ja konfiguroinnissa syntyneitä haavoittuvuuksia [62, 125]. Ne voivat siis vaihdella esimerkiksi langattoman verkon fyysisellä kerroksella tapahtuvasta radiosignaalin häirinnästä aina sovelluskerroksella suoritettuun puskurinylivuotohyökkäykseen, jossa ohjelmointivirheen sisältävä ohjelma saadaan sopivalla syötteellä ylikirjoittamaan pinosta sille varatun tilan lisäksi myös aliohjelman paluuosoite, ja näin suorittamaan hyökkääjän haluamia komentoja [135].

Erilaisia verkkohyökkäyksiä eli tietoverkon kautta tapahtuvia tekoja tai toimintoja, joilla pyritään tietoverkon, tietojärjestelmän, laitteen tai datan vahingoittamiseen tai oikeudettomaan käyttöön on useita, ja niissä hyödynnetyt haavoittuvuudet riippuvat luonnollisesti itse IoT-sovelluksesta [98, 62]. Hyökkäyksen yhteydes-

sä käyttäjä tai kuten Mirai-bottiverkon yhteydessä ohjelma, suorittaa toimenpiteitä saavuttaakseen haluamansa tavoitteen, joka voi olla käyttöoikeuksien laajentaminen, luvaton tietoon pääsy tai sen muuttaminen, järjestelmän normaalin toiminnan estäminen tai sen resurssien luvaton käyttö [62].

Luotaus ja skannaus ovat toimenpiteitä, joiden avulla hyökkääjä saa kerättyä tietoa hyökkäyksen mahdollisesta kohteesta tai mahdollisista kohteista [62]. Esimerkiksi TCP-yhteyden muodostuksen aloittavalla SYN-segmentillä toteutetussa skannauksessa hyökkääjä saa selvitettyä ilman kolmitiekättelyn loppuun viemistä, ajetaanko tietyn IP-osoitteen mukaisen kohteen tietyssä portissa verkkopalvelua, johon se voisi muodostaa TCP-yhteyden [22]. Myös Mirai-bottiverkon laitteet hyödynsivät tämän kaltaista skannausta, tosin lähinnä sen nopeuden vuoksi, etsiessään uusia laitteita, joihin ne saattoivat muodostaa telnet- tai ssh-yhteyden [10].

Tulviminen taas on toimenpide, jolla hyökkääjä pyrkii ylikuormittamaan hyökkäyksen kohteena olevan järjestelmän ja näin vaarantamaan sen saatavuuden [62]. Koska eri tavalla toteutettuja erilaisia verkkopalveluja on useita, myös niihin kohdistuvia tulvimiseen perustuvia palvelunestohyökkäyksiä (Denial of Service, DoS) on useita. Myös TCP-kolmitiekättelyä voidaan hyödyntää SYN-tulvalla toteutetussa DoS-hyökkäyksessä, sillä palvelin joutuu varaamaan resursseja jokaiselle aloitetulle kättelylle joksikin aikaa [22]. Hajautetuissa palvelunestohyökkäyksissä (Distributed Denial of Service, DDoS) hyökkäykseen osallistuu useita laitteita [22].

Tunnistautuminen ja sen ohittaminen ovat toimenpiteitä, joiden avulla hyökkääjä saa ainakin osittaisen pääsyn hyökkäyksen kohteena olevaan järjestelmään [62]. Siinä missä tunnistautumisella tarkoitetaan esimerkiksi käyttäjätunnus- ja salasanalistojen avulla järjestelmään kirjautumista, niin sen ohittamisella tarkoitetaan esimerkiksi järjestelmään jätetyn takaoven tai tunnetun haavoittuvuuden hyödyntämistä tunnistautumisen ohittamiseen [62]. Eräitä Web-käyttöliittymistä usein löytyviä haavoittuvuuksia ovat XSS (Cross-Site Scripting) ja erityyppiset injektiot, jotka tyypillisesti aiheutuvat siitä, että käyttäjän syöte lisätään sellaisenaan osaksi jotain rakennetta, kuten HTML tai SQL, jolla on oma kielioppinsa, jossa asianmukaisesti koodaamattomalla syötteellä saattaa olla epätoivottu merkitys [136].

Verkon laite voi myös esittää olevansa jokin toinen laite [62]. Tällaisessa toimenpiteessä hyökkääjän ohjaama laite voi esimerkiksi esittää olevansa kyseisen aliverkon oletusyhdyskäytävä eli laite, jolle aliverkkoon kuulumattomille laitteille osoitetut IP-paketit osoitetaan siirtoyhteyskerroksella [20]. Koska IPv4-verkoissa ARP-kyselyt ja -vastaukset ovat salaamattomia, riittää hyökkääjälle lähettää uhrille ARP-

vastaus, jossa se ilmoittaa oman MAC-osoitteensa vastaavan oletusyhdyskäytävän IP-osoitetta [96]. Näin hyökkääjä pääsee uhrin ja pakettien varsinaisen vastaanottajan väliin, ja voi tarvittaessa aloittaa MITM-hyökkäyksen.

Mies välissä -hyökkäys (Man-in-the-middle, MITM) voidaan suorittaa myös esimerkiksi sopivasti ajoitetulla salaamattomalla DNS-vastauksella, jossa hyökkääjä esittää olevansa näkemässään DNS-kyselyssä kysyttyä DNS-nimeä vastaava laite sijoittamalla IP-osoitteensa vastaukseen [42]. Toisen laitteen esittämisen lisäksi MITM-hyökkäyksissä käytettyjä yleisiä toimenpiteitä ovat ainakin tietojen lukeminen, kopiointi ja muokkaaminen [62]. Niiden avulla hyökkääjä voi muun muassa muokata osapuolten toisilleen lähettämien sanomien sisältämiä tietoja.

Lukemisen, kopioinnin ja muokkaamisen lisäksi muita yleensä tiedostoihin kohdistettuja toimenpiteitä ovat niiden varastaminen ja poistaminen [62]. Esimerkiksi Mirain tavoin IoT-laitteisiin levinneet BrickerBot ja Silex ovat haittaohjelmia, jotka poistivat laitteiden flash-muistissa olleet tiedot ylikirjoittamalla ne [23]. Koska IoT-laitteissa ei useinkaan säilytetä mitään kovin arvokasta, eivät ne ole niin otollisia kohteita kiristyshaittaohjelmille, jotka vaativat maksua salaamiensa tietojen avaamiseen tarvittavasta avaimesta, kuin yleiskäyttöiset tietokoneet [23].

2.3.2 Tietoliikenteen salaaminen

Koska verkkopalveluihin kohdistetut hyökkäykset välittyvät TCP/IP-verkoissa IP-pakettien päällä, voidaan niiltä pyrkiä suojautumaan haavoittuvuuksien paikkaamisen lisäksi myös estämällä hyökkäykseen liittyvien pakettien pääsy itse palveluun esimerkiksi palomuurien ja tunkeutumisen estämisjärjestelmien avulla [8]. Ilman tietoliikenteen salausta näillä keinoilla ei kuitenkaan voida estää esimerkiksi langattoman viestinnän salakuuntelua radiovastaanottimella tai hyökkääjän kontrolloiman IP-paketteja reitittävän laitteen suorittamaa MITM-hyökkäystä.

Tietoliikenteen turvallinen salaaminen tarjoaakin IP-paketteja keskenään vaihtaville tahoille keinon varmistua siitä, että hyökkääjä voi purkaa kaappaamansa verkon yli siirretyn hyötykuorman salauksen vain tuntemalla siinä käytetyt salausavaimet. Tietojen salaamiseen riittää periaatteessa symmetrinen salaus, jossa tietojen salaukseen ja salauksen purkamiseen käytetään samaa avainta. Esimerkiksi kertaavaimeen perustuvassa täydellisessä salauksessa salattavan bittijonon b_p ja sen pituisen täysin satunnaisen symmetrisen salausavaimen b_k avulla saadaan salattu bittijono $b_s = b_p \oplus b_k$ ja siitä edelleen alkuperäinen bittijono $b_p = b_s \oplus b_k$. [41]

Jos salausjärjestelmä on turvallinen, mahdollistaa se myös lähettäjän tunnistamisen, koska tällöin hyökkääjä ei voi muodostaa haluamiaan salaamattomia bittijonoja vastaavia salattuja bittijonoja tuntematta viestinnässä käytettyä salausavainta. Symmetrisen salauksen ongelmana on kuitenkin salausavainten turvallinen sopiminen eri osapuolten välillä. Periaatteessa ongelma voidaan ratkaista käyttämällä symmetristen avainten välittämiseen eri tahojen välillä julkisen avaimen järjestelmää, jossa tietojen salaukseen ja sen purkamiseen käytetään eri avaimia. [41]

Symmetrisessä salauksessa käytetty avain voitaisiin siis esimerkiksi välittää turvattoman verkon yli salaamalla se ensin vastaanottajan julkisella avaimella, ja sen jälkeen vielä lähettäjän yksityisellä avaimella [41]. Koska julkisen avaimen järjestelmässä julkiset avaimet ovat nimensä mukaan julkisia, saa vastaanottaja purettua ulomman salauksen lähettäjän julkisella avaimella ja sisemmän salauksen omalla yksityisellä avaimellaan, olettaen, että salaus voidaan myös purkaa julkisella avaimella, kun se on salattu yksityisellä avaimella [41]. Menetelmä kuitenkin antaa hyökkääjälle mahdollisuuden aiemmin tallentamiensa yhteyksien salauksen purkamiseen, jos vastaanottajan yksityinen avain myöhemmin paljastuu.

Yllä kuvattu menetelmä muistuttaakin TLS 1.3 -määrityksestä poistettuja salausalgoritmipaketteja, joissa palvelimen julkista avainta käytetään palvelimen tunnistamisen lisäksi myös asiakkaan palvelimelle välittämän symmetrisen salausavaimen muodostamiseen käytetyn salaisuuden salaukseen [40, 118]. Parempi suoja TLS:ssä palvelimen yksityisen avaimen paljastumista vastaan saadaankin, kun salaisuuden välittämiseen käytetään Diffie-Hellman -avaintenvaihtoa, jossa koko salaisuutta ei missään vaiheessa välitetä verkon yli [117, 118]. Siirtoyhteys- ja sovelluskerroksella suoritetun salauksen lisäksi IoT-sovelluksissa voidaan hyödyntää myös verkkokerroksella tapahtuvaa salausta IPSec-protokollien avulla [70].

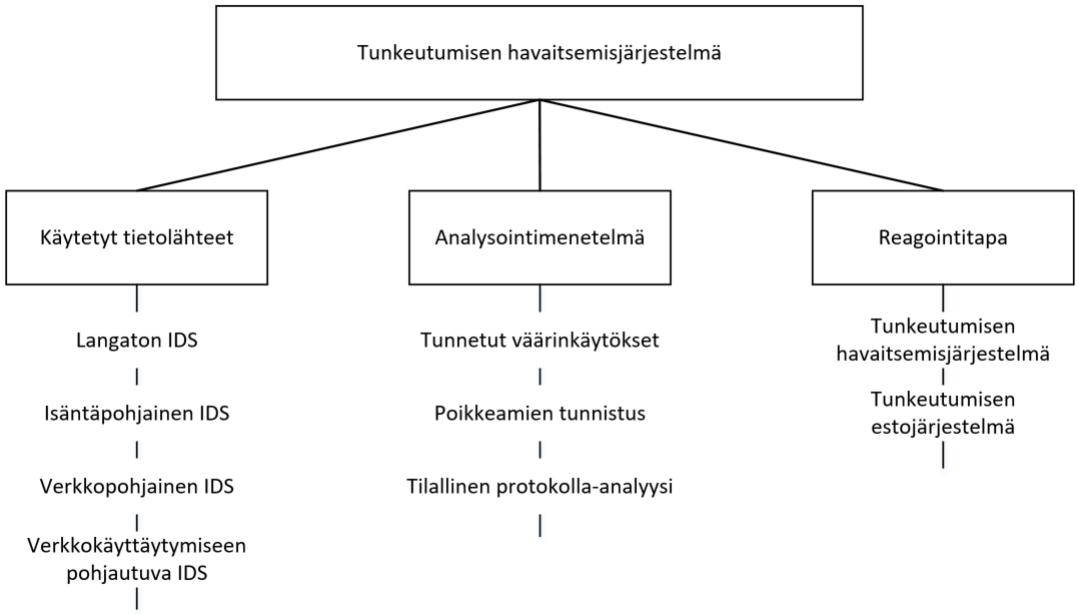
3 Tunkeutumisen havaitsemisjärjestelmät

Tunkeutumisen havaitsemisjärjestelmien yhteydessä tunkeutuminen voidaan määrittää joukoksi tietokoneisiin tai tietoverkkoon kohdistuvia toimenpiteitä, joilla pyritään ohittamaan jokin turvamekanismi tai joilla pyritään rikkomaan tietoturvan perustavoitteita, joihin kuuluvat luottamuksellisuus, eheys ja saatavuus [15]. Tunkeutumisen havaitsemisjärjestelmä (Intrusion Detection System, IDS) on siis ohjelmisto tai laite, jonka tarkoituksena on havaita tällaista epäilyttävää toimintaa tarkkailtavassa järjestelmässä syntyvien tapahtumien perusteella [2, 15]. Syötteenään IDS voi käyttää sekä verkkoliikenteestä että tietokoneista keräämiään tietoja [2].

Tunkeutumisen havaitseminen voidaan myös määrittää prosessiksi, jossa tietokoneita ja tietoverkkoa tarkkailemalla pyritään löytämään merkkejä toiminnasta, joka rikkoo tai uhkaa rikkoa valvottavan järjestelmän tietoturvapolitiikka, hyväksyttävän käytön politiikkaa tai tietoturvakäytänteitä. Jos järjestelmä pyrkii tunkeutumisen havaitsemisen lisäksi myös estämään mahdollisen tunkeutumisen, on kyseessä tunkeutumisen estämisjärjestelmä (Intrusion Prevention System, IPS). Muuten tunkeutumisen havaitsemis- ja estämisjärjestelmät (Intrusion Detection and Prevention Systems, IDPS) ovat toiminnoiltaan varsin samanlaisia. [120]

3.1 Tunkeutumisen havaitsemisjärjestelmien luokittelu

Tunkeutumisen havaitsemisjärjestelmät tarvitsevat tietolähteitä, joiden tuottamia tietoja analysoimalla ne pyrkivät löytämään mahdollisia tunkeutumisia tai jo tapahtuneita tunkeutumisia [15]. Kun mahdollinen tunkeutuminen on havaittu, täytyy järjestelmän vielä reagoida siihen jotenkin. Tunkeutumisen havaitsemiseen ja estämiseen tehdyt järjestelmät voidaankin luokitella näissä perustoiminnoissa olevien erojen perusteella huomioiden myös se, mistä niiden käyttämien tietolähteiden tuottamat tiedot on kerätty [15, 120]. Kuvaan 3.1 on koottu näiden erojen pohjalta muodostettuja luokitteluja. On kuitenkin huomattava, että IDPS-tuotteiden ei ole mikään pakko kuulua vain yhteen luokkaan eri luokituksissa [120].



Kuva 3.1: Tunkeutumisen havaitsemisjärjestelmien luokitteluja

3.1.1 Luokittelu käytettyjen tietolähteiden perusteella

Käyttämiensä tietolähteiden ja sijaintinsa perusteella tunkeutumisen havaitsemisjärjestelmät voidaan luokitella langattomiksi, isäntäpohjaisiksi, verkkopohjaisiksi sekä verkkokäyttäytymisen analysointiin pohjautuviksi [120]. Usein tällainen luokittelu [2, 15, 19, 103] tosin tehdään vain isäntä- ja verkkopohjaisten järjestelmien välillä: periaatteessahan sekä langattomien protokollien että verkkokäyttäytymisen analysointiin perustuvat järjestelmät hyödyntävät tietolähteenään joissain muodossa olevaa verkon tietoliikennettä.

Langaton IDS (Wireless IDS, WIDS) käyttää siis tietolähteenään kuulemaansa radioliikennettä ja siinä käytettyjä langattomia tietoliikenneprotokollia kyeten näin havaitsemaan esimerkiksi luvattoman WLAN-tukiaseman tai päätelaitteen [120]. Esimerkiksi avoimen lähdekoodin Kismet [72] on WIDS, joka pystyy tarkkailemaan sekä 802.11- että Bluetooth-standardin mukaista radioliikennettä. Langattoman tietoliikenteen monitorointiin liittyy myös omat haasteensa: radioaaltojen kantaman ja usean kanavan seuraamisen haasteet on otettava huomioon [120].

Isäntäpohjainen IDS (Host-based IDS, HIDS) puolestaan hyödyntää tietolähteinään valvomastaan koneesta keräämiään tietoja, jotka voivat liittyä esimerkiksi laitteella käynnissä oleviin prosesseihin, sillä oleviin tiedostoihin tai tietoihin, jotka saa-

tettaisiin siirtää verkon yli salattuna [15, 120]. Avoimen lähdekoodin OSSEC [102] on isäntäpohjainen IDPS, joka osaa havaita myös haitta- ja piilohallintaohjelmia. HIDS voi siis sisältää myös virustorjuntaohjelmista tuttuja ominaisuuksia [120].

Sekä verkkopohjainen IDS (Network IDS, NIDS) että verkkokäyttäytymisen analysointiin perustuva IDS (Network Behavior Analysis, NBA) käyttävät tietolähteenään verkkoliikennettä [120]. Näistä NBA-ohjelmistot, joihin Cisco Secure Network Analytics [29] lukeutuu, ovat uudempia, ja ne ovat kehittyneet osin tuotteista, joiden päätarkoituksena on ollut tunnistaa DDoS-hyökkäyksiä, ja osin tuotteista, jotka ovat analysoineet verkon liikennevirtoja yksittäisten pakettien sijaan [120]. Avoimen lähdekoodin Snort [128] on puolestaan suosittu tunnettuja väärinkäytöksiä tunnistava verkkopohjainen IDPS. Tyypillisesti NIDS ja NBA eroavatkin toisistaan myös siten, että NBA etsii tunnettujen väärinkäytösten sijaan poikkeamia normaalista [120].

3.1.2 Luokittelu analysointimenetelmän perusteella

Tunkeutumisen havaitsemisjärjestelmät voidaan luokitella myös käytetyn analysointimenetelmän perusteella. Väärinkäytöksiä etsivä IDS pyrkii löytämään analysoimistaan tietolähteistä tunnettuja väärinkäytöksiä tietyille hyökkäyksille ominaisten sormenjälkien avulla. Koska menetelmä perustuu etukäteen määriteltyihin sormenjälkiin, tunnistaa se tarkasti tunnetut hyökkäykset. Toisaalta tällainen IDS ei voi havaita tunkeutumista, jos sillä ei ole siihen sopivaa sormenjälkeä. Se ei myöskään osaa seurata monimutkaisen viestinnän tilaa, eikä näin yleensä tunnista hyökkäyksiä, jotka aiheuttavat useita tietolähteissä näkyviä tapahtumia. [120]

Poikkeamia tunnistava IDS puolestaan pyrkii löytämään anomalioita analysoimiensa tietolähteiden tuottamista tiedoista. Jotta tällainen IDS voi tunnistaa tunkeutumisen, täytyy tunkeutumisen kuitenkin muuttaa tietolähteiden normaalisti tuottamaa tietoa siten, että se eroaa normaalin käytön tuottamasta tiedosta. Normaalia toimintaa kuvaavan profiilin tai profiilit anomalioita tunnistava IDS voi muodostaa tietolähteiden järjestelmän tavallisen käytön aikana tuottamien tietojen pohjalta. Käytännössä voi kuitenkin olla hankalaa varmistua siitä, että profiilin luontiin käytetty historiallinen tieto ei sisällä tunkeutumisia. Vaikka anomaliapohjaiset tunkeutumisen havaitsemisjärjestelmät voivatkin tunnistaa myös ennestään tuntemattomia hyökkäyksiä, tuottavat ne usein myös runsaasti vääriä hälytyksiä. [120]

Tunkeutumisen havaitsemisjärjestelmä voi hyödyntää menetelmänään myös tilallista protokolla-analyysiä, jossa tarkasteltavan tietoliikenneprotokollan toimintaa verrataan kyseisen protokollan normaaliin toimintaan. Toisin kuin poikkeaman tun-

nistukseen perustuvassa analysoinnissa tässä protokollan normaalin toiminnan profiili perustuu ennalta määrättyihin yleisiin malleihin siitä, miten kyseisen protokollan tulisi normaalisti toimia. Analyysissä voidaan siis huomioida esimerkiksi protokollan kieliopin mukaisten syötteiden muoto sekä tietyssä protokollan tilassa normaalisti käytetyt sanomat. Tunkeilijan havaitsemisjärjestelmät voivat luonnollisesti myös yhdistää eri analysointimenetelmiä. [120]

3.1.3 Luokittelu reagointitavan perusteella

Kun IDPS on havainnut mahdollisen tunkeutumisen, täytyy sen vielä reagoida siihen jotenkin. Siinä missä IDS vain ilmoittaa epäilemistään tunkeutumisista esimerkiksi hälytyksen tai sähköpostin muodossa, IPS myös reagoi havaitsemaansa tunkeutumiseen. Se voi esimerkiksi sulkea TCP-yhteyden lähettämällä RST-paketin yhteyden molemmille osapuolille tai vaikuttaa tietoturvaympäristöön muun muassa konfiguroimalla reitittimen tai palomuurin uudelleen. Se voi myös muuttaa havaitsemiensa pakettien sisältöä tehden niistä vaarattomia. [120]

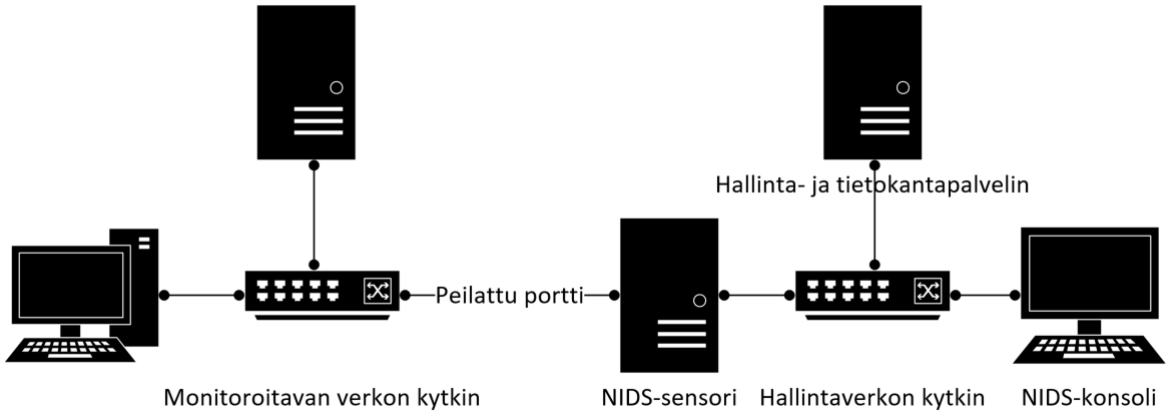
Jos tunkeutumisen estämisjärjestelmän halutaan voivan estää tai muuttaa tarkkailemansa verkkoliikenteen sisältöä, täytyy sen käyttämät sensorit sijoittaa siten, että verkkoliikenne kulkee niiden läpi. Muuten IDPS-sensorit sijoitetaan yleensä siten, että ne monitoroivat vain kopiota varsinaisesta verkkoliikenteestä. Sensorin voidaan sanoa olevan aktiivinen, jos verkon liikenne kulkee sen läpi, ja passiivinen, jos se käsittelee vain kopiota verkon liikenteestä. [120]

3.2 Tunkeutumisen havaitsemisjärjestelmän komponentit

IDS koostuu tyypillisesti sensoreista, agenteista, hallinta- ja tietokantapalvelimista sekä konsoleista. Näistä verkkoliikennettä tarkkailevat sensorit ja isäntäkoneita valvovat agentit keräävät ja käsittelevät tunkeutumisen havaitsemisjärjestelmän käyttämien tietolähteiden tuottamaa informaatiota. Hallintapalvelin taas vastaanottaa tietoa hallinnoimiltaan sensoreita ja agenteilta. Se voi myös analysoida useista eri lähteistä saamaansa tietoa ja löytää tunkeutumisia, joita yksittäiset sensorit ja agentit eivät havainneet. IDS voi toimia myös ilman tietokanta- ja hallintapalvelimia. Jos tietokantapalvelinta käytetään, on sen tehtävänä tallentaa eri tietolähteiltä kerätty ja jalostettu tieto. Hallintakonsoli on käyttöliittymä, jonka kautta järjestelmää voidaan monitoroida, konfiguroida ja päivittää. [120]

Tunkeutumisen havaitsemisjärjestelmän komponentit voivat olla yhteydessä toisiinsa organisaation tavallisen verkon lisäksi myös erillisen hallintaverkon kautta, jolloin IDS saa käyttöönsä tarpeellisen tiedonsiirtokapasiteetin myös esimerkiksi DDoS-hyökkäyksen aikana. Erillisessä verkossa jokaisella komponentilla on oma verkkoliitäntänsä, jonka kautta kulkeva liikenne on täysin erillään tarkkailtavasta liikenteestä. Tällöin IDS ei myöskään ole hyökkääjien havaittavissa. Jos erillistä hallintaverkkoa ei käytetä, saadaan komponenttien välinen viestintä eriytettyä tavallisesta liikenteestä myös virtuaalisen lähiverkon avulla. Tällöin tunkeutumisen havaitsemisjärjestelmän sisäinen liikenne kuitenkin jakaa verkon fyysiset resurssit tarkkailemansa liikenteen kanssa. [120]

Kuvassa 3.2 on esitetty yksinkertainen erillistä hallintaverkkoa hyödyntävä arkkitehtuuri verkkopohjaiselle tunkeutumisen havaitsemisjärjestelmälle, jonka ainoa sensori on sijoitettu sellaiseen valvottavan verkon kytkimen porttiin, johon kytkimen muiden porttien kautta kulkeva liikenne on peilattu. Tällaisessa asennuksessa NIDS voi siis halutessaan analysoida myös siirtoyhteyskerroksen kehyksiä, joita Ethernet-verkoissa käytetään muun muassa verkkokerroksen IP-osoitteiden ja siirtoyhteyskerroksen MAC-osoitteiden välisten yhteyksien muodostamisessa ARP-kyselyjen ja -vastausten avulla [109, 61]. Esimerkiksi Snort voidaan määrittää tunnistamaan ARP-kehysten avulla tehtyjä MITM-hyökkäyksiä [61].



Kuva 3.2: Yksinkertainen NIDS-arkkitehtuuri

Toisin kuin siirtoyhteyskerroksen kehykset verkkokerroksen IP-paketit reitittyvät myös eri verkkojen välillä [112]. Niitä ja niiden kuljettamia ylemmän kerroksen protokollia tarkkailemalla tunkeutumisen havaitsemisjärjestelmän on siis mahdol-

lista seurata eri laitteiden välistä kommunikaatiota silloinkin, kun toinen osapuolista on valvotun verkon ulkopuolella, eikä näin ole tunnistettavissa siirtoyhteyskerroksen [112] osoitetietojen perusteella. Verkkoon asennettu sensori voi IP-pakettien sijaan tarkkailla myös niistä koostettuja liikennevirtatietueita [120].

3.3 Liikennevirtatietueiden hyödyntämien

Tunkeutumisen havaitsemisen yhteydessä käytetyt liikennevirtatietueet sisältävät tyypillisesti ainakin lähteen ja kohteen IP-osoitteen, käytetyn ylemmän kerroksen protokollan, siirrettyjen tavujen ja pakettien määrän sekä yhteyden aloitus- ja lopetushetken [120]. Käytetyn protokollan ollessa TCP tai UDP ne sisältävät tyypillisesti myös lähteen ja kohteen porttinumeron sekä vastaavasti tyypin ja koodin protokollan ollessa ICMP [120]. Useat [60] kytkimet ja reitittimet voivat itsekin tuottaa tällaisia tietueita käsittelemistään paketeista. Niiden jatkokäsittelyn edellyttämään viemiseen keräyspisteiltä on määritelty muun muassa Netflow- ja IPFIX-protokolla [30, 31]. Myös esimerkiksi avoimen lähdekoodin Argus [11] ja Zeek [140] osaavat muodostaa liikennevirtatietueita käsittelemästään verkkoliikenteestä.

Liikennevirtatietueiden käyttäminen yksittäisten pakettien sijaan vähentää reitittimien ja muiden keräyspisteiden läpi kulkevan liikenteen tallentamiseen tarvittavan tilan määrää merkittävästi [60]. Tällöin poikkeamia tunnistavan tunkeutumisen havaitsemisjärjestelmän tarvitsema normaali liikenne vaatii siis myös vähemmän tallennustilaa. Koska liikennevirtatietoja käyttävä IDS ei näe yksittäisten pakettien sisältöä, ei se myöskään voi analysoida niiden sisältöä. Sen on siis rajoituttava niihin tietoihin, joita kerätyt liikennevirtatietueet jo sisältävät. Toki liikennevirtatietueiden muodostamisen yhteydessä on myös mahdollista analysoida IP-pakettien hyötykuormaa, ja muodostaa esimerkiksi sovellustason protokollaan liittyviä tietovirtaa koskevia tietoja [138]. Luonnollisesti se vaatii kuitenkin enemmän resursseja kuin pelkkien otsikkotietojen käsittely, ja niihin liittyvän tilan ylläpito.

Pelkkien kerättyjen liikennevirtatietojen analysointiin perustuva IDS ei kuitenkaan voi havaita poikkeamia reaaliaikaisesti, siellä tietueet ovat valmiita vietäviksi tyypillisesti vasta, kun liikennevirtaan liittyvä TCP-yhteys suljetaan tai kun jokin siihen liittyvä aikaraja ylittyy [60]. Aikarajoilla voidaan rajoittaa liikennevirran enimmäiskestoa sekä sitä, kauanko siihen liittyviä uusia paketteja maksimissaan odotetaan [60]. Aikarajoja tarvitaan jo siksikin, että esimerkiksi UDP [111] ei ole yhteydellinen protokolla.

4 Poikkeamien tunnistaminen verkkoliikenteestä

Usein viitatun Hawkinsin [57] määritelmän mukaan poikkeama on havainto, joka eroaa muista havainnoista niin paljon, että herää epäilys siitä, että se on eri mekanismin muodostama. Havaintoyksikkö voi olla poikkeava yksittäisen muuttujan arvon lisäksi myös, koska siitä tehdyt eri havainnot ovat yhdessä poikkeavia [133]: esimerkiksi 40 kiloa painava 210 cm pitkä ihminen. Kaikki poikkeamat eivät kuitenkaan ole välttämättä mielenkiintoisia sovelluksen näkökulmasta [1, 19]. Esimerkiksi huoltotoimenpiteiden yhteydessä siirretty suuri tiedosto voi aiheuttaa verkkoon epätavallista liikennettä, vaikka ei täytäkään tunkeutumisen määritelmää. Tunkeutumisen havaitsemisessa poikkeamalla on siis rajatumpi määritelmä.

Käytettävissä olevien tietojen luonne vaikuttaa suuresti niihin soveltuviin poikkeaman tunnistusmenetelmiin [133]. Verkkoliikenteestä NIDS saa IP-tasolla [112] kerättyä ainakin lähettäjän ja vastaanottajan osoitteen, paketin koon sekä aikaleiman. Myös tietovirtatasolla tyypillisesti tiedetään ainakin lähteen ja kohteen osoitteet sekä tietovirran koko ja siihen liittyvät aikaleimat [120]. Havaituista paketeista ja tietovirroista muodostetut datapisteet ovat siis tyypillisesti moniulotteisia ja riippuvat toisistaan. Jos niitä käsitellään tunnistuksessa tavallisena havaintomatriisina, jäävät niiden väliset ajalliset ja yhteydelliset riippuvuudet huomioimatta [1, 133].

Poikkeamat voidaankin luokitella [26] pisteanomalioiksi, kontekstuaalisiksi anomalioiksi ja kokoelma-anomalioiksi sen mukaan, miten ne eroavat muista aineiston datapisteistä. Kaikista yksinkertaisimpia anomalioita ovat pisteanomaliat, jotka ovat poikkeamia, koska ne eivät muistuta muita datapisteitä [26]. Esimerkki tällaisesta anomaliasta voisi olla yksittäinen IP-paketti [112], jonka otsikon versiokentässä on arvon 4 sijaan arvo 1. Jos aineiston datapisteet eivät riipu toisistaan ja jos niihin ei löydy kontekstia, voidaan niiden joukosta etsiä vain pisteanomalioita [26].

Kontekstuaalinen anomalia [26] taas on poikkeama, koska se eroaa muista datapisteistä siinä kontekstissa, jossa se esiintyy. Verkkoliikenteessä tällainen poikkeama voisi olla esimerkiksi ilman SYN-lippua lähetetty TCP-yhteyteen [113] liittyvä segmentti, jota ei ole edeltänyt kolmitiekättelyyn liittyvä SYN-lipulla varustettu segmentti. Kokoelma-anomalia on puolestaan joukko toisiinsa liittyviä datapisteitä, jotka yhdessä poikkeavat muista datapisteistä [26]. Esimerkiksi SYN-tulvalla toteu-

tetusta palvelunestohyökkäyksestä muodostuneet useat TCP-segmentit muodostavat tällaisen poikkeaman [100]. Joskus kontekstuaaliset anomaliat voi olla hyödyllistä pelkistää pisteanomalioiksi [26]. Esimerkiksi ilman SYN-lippua lähetetty TCP-segmentti on pisteanomalia tietystä lähdeosoitteesta ja -portista tietylle kohdeosoitteelle ja -portille ensimmäiseksi lähetettyjen segmenttien kontekstissa.

4.1 Poikkeaman tunnistusmenetelmien luokittelu

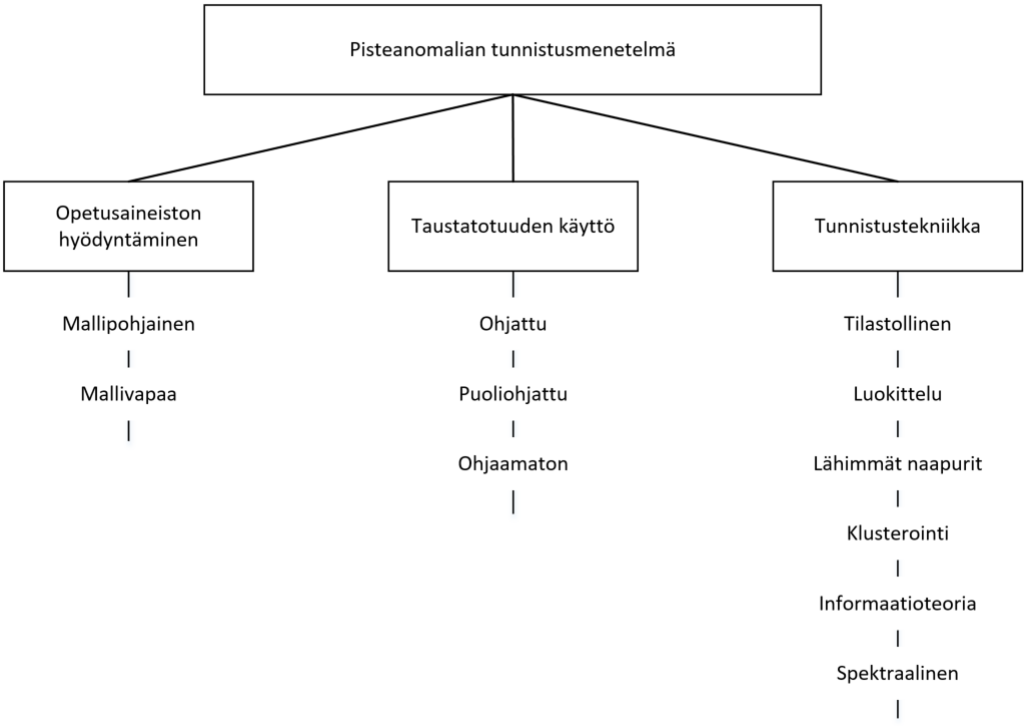
Verkkoliikenteestä voidaan muodostaa varsin eri tyyppisiä aineistoja. Eri koneiden väliset yhteydet eri ajanhetkinä voidaan esimerkiksi esittää graafeina [107]. Koska verkon liikenne on luonteeltaan temporaalista, voidaan siitä myös muodostaa aikasarjoja [107, 146]. Lisäksi verkon laitteilla on sijainti, ja usein myös käyttäjä. Eri tyyppisiin aineistoihin ja ongelmiin sopivia poikkeaman tunnistusmenetelmiä onkin kehitetty useita [133]. Osa niistä on mallipohjaisia ja osa mallivapaita [133].

Mallipohjaiset menetelmät muodostavat opetusaineistosta mallin, jonka perusteella uusien datapisteiden poikkeavuutta voidaan arvioida. Yksiulotteiseen aineistoon malliksi voi sopia esimerkiksi normaalijakauma $X \sim \mathcal{N}(\mu, \sigma^2)$, jolloin datapisteen x poikkeavuus voidaan esittää yksinkertaisesti standardoidun havaintoarvon itseisarvona $|\frac{x-\mu}{\sigma}|$, kun μ on odotusarvo ja σ^2 varianssi. Mallittomissa menetelmissä mitään varsinaista mallia ei muodosteta, ja uuden datapisteen poikkeavuutta kuvaava arvo saattaakin olla esimerkiksi sen etäisyys viidenneksi lähimpään naapuriinsa opetusaineistossa, joka on sisältänyt vain normaaleja datapisteitä. [133]

Eri poikkeaman tunnistusmenetelmät voivat antaa ennusteensa käsittelemiensä datapisteiden poikkeavuudesta eri muodossa. Jotkut menetelmät tuottavat vain ennusteen taustatotuudesta, joka yksinkertaisimmillaan voi tarkoittaa tietoa siitä, onko arvioitava datapiste menetelmän mielestä poikkeama vai ei. Menetelmät saattavat myös tuottaa pisteytyksen, jonka avulla arvioidut datapisteet on mahdollista järjestää eniten poikkeavasta vähiten poikkeavaan. Järjestystä voidaan hyödyntää poikkeamien priorisoinnissa sekä itse menetelmän arvioinnissa. Pisteytyksestä saadaan ennuste taustatotuudesta määrittämällä kaikki tiettyä kynnysarvoa suuremman anomaliapisteytyksen saaneet datapisteet poikkeamiksi. [133]

Vaikka useat poikkeaman tunnistusmenetelmät eivät suoraan sovellukaan kontekstuaalisten ja kokoelma-anomalioiden tunnistamiseen, voidaan niiden tunnistamisessa usein kuitenkin hyödyntää samoja tekniikoita kuin pisteanomalioiden tunnistamisessa [3, 26]. Voidaanhan esimerkiksi kontekstuaalinen anomalia nähdä pis-

teanomaliana omassa kontekstissaan [26]. Kuvassa 4.1 on esitetty pisteanomalioiden tunnistamiseen kehitettyjen menetelmien luokittelu opetusaineiston hyödyntämisen [133], taustatotuuden käytön [19] ja tunnistustekniikan [26] mukaan.



Kuva 4.1: Pisteanomalian tunnistusmenetelmien luokitteluja

4.1.1 Luokittelu taustatotuuden käytön mukaan

Koska poikkeamia tunnistava NIDS tai NBA tyypillisesti muodostaa valvottavan järjestelmän normaalia toimintaa kuvaavan profiilin järjestelmän tavallisen käytön aikana kerätyistä tiedoista, ei sillä tyypillisesti ole esimerkkejä tunkeutumisia sisältävästä liikenteestä, ellei normaalin käytön aikana ole esiintynyt tunkeutumisia [120, 130]. Jos profiilin rakentamisen eli opetusvaiheen aikana IDS olettaa kaiken liikenteen olevan normaalia, tulisi tällaisista tunkeutumisista kerätyt tiedot kuitenkin poistaa opetukseen käytetystä aineistosta.

Poikkeaman tunnistusmenetelmät voidaankin luokitella sen mukaan, miten ne hyödyntävät mahdollisesti saatavilla olevaa taustatotuutta, joka tunkeutumisen havaitsemisen yhteydessä voisi siis tarkoittaa esimerkiksi tietoa siitä, onko kyseessä

tunkeutuminen vai ei [19, 133]. Jos taustatotuus on opetukseen käytettävissä olevan aineiston osalta saatavilla jokaiseen luokkaan, on poikkeaman tunnistus ohjattua [19], ja sitä voidaan lähestyä luokitteluongelmana [133]. Käytännössä edustavasti tunkeutumisia sisältävästä verkkoliikenteestä muodostetun aineiston kerääminen on kuitenkin hankalaa [19]. Ohjattua oppimista on tunkeutumisten havaitsemiseen liittyen hyödynnetty rajatummin muun muassa algoritmisesti muodostettujen vihamielisten verkkotunnusten tunnistamisessa [148].

Puoliohjattu poikkeaman tunnistus eroaa ohjatusta poikkeaman tunnistuksesta siten, että kaikki havaintoyksiköt ovat normaaleja [4, 19, 26]. Tämä vastaa tilannetta, jossa verkkoliikenteestä poikkeamia etsivä IDS olettaa opetusvaiheen aikana kerättyjen tietojen kuvaavan järjestelmän normaalia käyttöä. Kaikkiin havaintoyksiköihin siis liittyy implisiittinen taustatotuus siitä, että ne ovat normaaleja. Normaaliksi oletetun verkkoliikenteen seassa mahdollisesti oleva tunkeutumisista syntynyt liikenne vaikuttaa luonnollisesti tällaista menetelmää käyttävän järjestelmän kykyyn tunnistaa poikkeamia ja siten myös tunkeutumisia [120]. Puoliohjatulla oppimisella toisaalta tarkoitetaan eräänlaista luokittelua, jossa myös sellaisia havaintoyksiköitä pyritään hyödyntämään, joiden taustatotuutta ei tunneta [150].

Ohjaamattomassa poikkeamien tunnistuksessa taustatotuutta ei toisaalta edes tarvita, sillä siinä käytettävissä olevan aineiston oletetaan sisältävän sekä normaaleja että poikkeavia havaintoyksiköitä [19, 133]. Verkkoliikenteestä tunkeutumisia etsittäessä tämä siis vastaisi tilannetta, jossa käsitellyn aineiston oletetaan myös sisältävän niitä. Ohjaamattomassa poikkeamien tunnistuksessa joudutaan kuitenkin tyypillisesti olettamaan, että normaalin taustatotuuden omaavia havaintoyksiköitä on huomattavasti enemmän kuin muun taustatotuuden omaavia [19, 3], jotta sovelluksen kannalta mielenkiintoiset poikkeamat eivät sopisi hyvin havaintoaineiston jakaumaan [26]. Esikäsittelyn yhteydessä suoritetussa aineiston puhdistuksessa käytettynä poikkeaman tunnistus on siis määritelmän mukaan ohjaamatonta.

4.1.2 Luokittelu tunnistustekniikan mukaan

Poikkeaman tunnistusongelmaan vaikuttaa osaltaan käytettävissä olevien tietojen luonne, taustatotuuden saatavuus, etsittävien poikkeamien luonne sekä haluttu ennusteen tyyppi [26]. Pääosin pisteanomalioiden etsimiseen soveltuvat tekniikat voidaan jakaa tilastollisiin, luokitteluun perustuviin, lähimpiin naapureihin pohjautuviin, klusterointia hyödyntäviin, informaatioteoreettisiin sekä spektraalisiin menetelmiin [26]. Vaikka samoja tekniikoita voidaan hyödyntää myös monimutkaisem-

pien anomalioiden tunnistamisessa [3, 26], saattaa kontekstuaalisten ja kokoelmaanomalioiden tunnistaminen vaatia myös erilaisia lähestymistapoja.

Tilastolliset menetelmät [26] voivat olla parametrisia tai parametrittomia. Ne olettavat, että poikkeavat datapisteet esiintyvät kohdissa, joissa niillä on tilastollisen mallin mukaan pieni todennäköisyys esiintyä. Parametrisia menetelmiä käytettäessä aineiston oletetaan noudattavan jotain tunnettua jakaumaa tai jakaumia, jolloin jakauman tai jakaumien parametrit saadaan estimoitua opetusaineistosta [26, 133]. Jos aineistoon on mahdollista sovittaa regressiomalli, voidaan esimerkiksi havaitun selitettävän muuttujan ja selittävien muuttujien arvojen perusteella saadun ennusteen välisen erotuksen eli virheen neliötä käyttää anomaliapistetyksenä [1, 26].

Parametrittomissa tilastollisissa menetelmissä mallin rakenne opitaan aineistosta, joten sitä ei tarvitse tietää etukäteen [26]. Esimerkiksi yksinkertaisessa parametrittomassa histogrammeihin perustuvassa menetelmässä yhden tai useamman aineiston muuttujan määräämä avaruus voidaan jakaa vakiomittaisin osiin, joihin mallin muodostuksessa käytetyn aineiston datapisteet sijoitetaan [1, 26]. Uuden havaintoyksikön anomaliapisteytyksenä voidaan tällöin käyttää esimerkiksi sen sisältäneen avaruuden osan datapisteiden lukumäärän käänteisarvoa, jos se on määritelty. Sileämpi arvio datapisteiden jakaumasta saadaan ydinestimoinnin avulla [1].

Luokittelua voidaan käyttää, kun opetusaineiston havaintoyksiköiden taustatotuus tunnetaan. Yhden luokan luokittelussa yritetään tyypillisesti löytää raja normaalien datapisteiden ympärille, joten siinä riittää, että opetusaineistossa on esimerkkejä yhden taustatotuuden omaavista havaintoyksiköistä. Koska useampiluokkaisen luokittelijan tarkoituksena on erottaa eri taustatotuuden eli luokan omaavat datapisteet toisistaan, tarvitaan sellaisen opettamiseen aineisto, jossa eri taustatotuuksia on enemmän kuin yksi. Jos luokittelijalta saadaan myös eri luokkiin kuulumisen varmuudesta kertova pisteytys, voidaan datapisteet, jotka eivät kuulu varmasti mihinkään luokkaan, määrittää poikkeamiksi, vaikka kaikki luokittelussa käytetyt luokat olisivatkin normaaleja. Eräitä luokittelualgoritmeja ovat neuroverkot, tukivektorikoneet ja päätöspuut. [26]

Lähimmän naapurin menetelmät ovat mallittomia, koska niissä opetusaineistosta ei muodosteta varsinaista mallia [133]. Ne olettavat, että normaalit datapisteet esiintyvät tiheissä naapurustoissa ja että poikkeamat sijaitsevat kaukana toisistaan ja muista datapisteistä [1]. Jotta eri datapisteiden väliset etäisyydet tai samankaltaisuudet saadaan selville, tarvitaan etäisyys- tai samankaltaisuusmittaa, jollaiseksi käy esimerkiksi euklidinen etäisyys [26]. Lähimmän naapurin menetelmät voivat

perustaa anomaliapisteytyksensä datapisteiden suhteelliseen tiheyteen tai datapisteiden ja niiden k:nneksi lähimmän naapurin etäisyyteen [26]. Suhteelliseen tiheyteen perustuvat menetelmät pyrkivät huomioimaan aineiston tiheyden vaihtelua ja toimimaan, vaikka aineistossa olisi vaihtelevan tiheyden omaavia klustereita [26].

Klusteroinnissa datapisteet pyritään sijoittamaan samaan klusteriin, jos ne muistuttavat toisiaan, ja eri klustereihin, jos ne eivät muistuta toisiaan. Siinäkin tarvitaan siis etäisyys- tai samankaltaisuusmittaa. Eri klusterointimenetelmien avulla poikkeamia voidaan löytää eri tavoilla. Esimerkiksi datapisteet, jotka DBSCAN-algoritmi jättää sijoittamatta mihinkään klusteriin, voidaan tulkita poikkeamiksi. Toisaalta esimerkiksi k-means-algoritmin avulla saadaan selville k:n klusterin keskipisteet, jolloin datapisteitä, jotka eivät ole lähellä mitään näistä keskipisteistä, voidaan pitää anomalioina. Myös kaikkia harvoihin tai vähän datapisteitä sisältäviin klustereihin kuuluvia datapisteitä voidaan pitää anomalioina. [26]

Informaatioteoreettisissa menetelmissä normaalien datapisteiden seassa olevien poikkeamien oletetaan vaikuttavan koko aineiston sisältämän informaation määrään [26], jota voidaan mitata esimerkiksi entropian $-\sum_{i=1}^{n} p_i \, log_2(p_i)$ avulla [123]. Aineiston, jossa on 50 punaista, 50 sinistä ja 3 purppuraa datapistettä, entropiaksi saadaan siten $-(\frac{50}{103}log_2(\frac{50}{103}) + \frac{50}{103}log_2(\frac{50}{103}) + \frac{3}{103}log_2(\frac{3}{103})) \approx 1.16$. Toisaalta, jos aineistosta poistetaan purppurat datapisteet, saadaan entropiaksi 1. Purppuroiden datapisteiden aiheuttama informaation lisäys on siten $1.16 - 1 = 0.16$. Eräs informaatioteoreettinen menetelmä perustuukin siihen, että aineiston D osajoukon S anomaliapisteytyksenä käytetään arvoa $I(D) - I(D \setminus S)$, jossa I on funktio, joka ilmaisee datapisteiden sisältämän informaation määrän [26, 133].

Spektraaliset menetelmät taas pyrkivät löytämään havaintoyksiköiden ominaisuuksien funktion, joka kuvaa aineiston datapisteet pienempiulotteiseen avaruuteen säilyttäen kuitenkin samalla suurimman osan niiden varianssista. Menetelmät pyrkivät lisäksi muodostamaan kuvauksen siten, että poikkeamat erottuvat maaliavaruudessa muista datapisteistä [26]. Aineiston datapisteet saadaan kuvattua pienempiulotteiseen avaruuteen muun muassa projisoimalla ne niille tehdyn pääkomponenttianalyysin (Principal Component Analysis, PCA) avulla saatujen ominaisvektoreiden osajoukon virittämään avaruuteen [26, 69]. Jos datapisteiden projektioiden koordinaateista eli pääkomponenteista valitaan vain ne, joiden suuntaan alkuperäisessä aineistossa on vain vähän vaihtelua, saadaan alkuperäisen aineiston korrelaatiorakennetta rikkovat datapisteet erottumaan normaaleista [69].

4.2 Aineiston esikäsittely

Aineiston esikäsittelyn tarkoituksena voidaan pitää raaka-aineiston muuttamista muotoon, jossa se on paremmin valittujen poikkeaman tunnistusmenetelmien hyödynnettävissä [36, 45, 133]. Eräitä aineiston esikäsittelyyn liittyviä tekniikoita ovat koostaminen, otanta, ulottuvuuksien pienentäminen, ominaisuuksien valinta ja muodostaminen sekä muuttujien diskretointi, binarisointi ja muunnokset [133]. Näiden tekniikoiden avulla voidaan siis valikoida datapisteitä, luoda uusia muuttujia sekä valikoida tai muuttaa jo olemassa olevia ominaisuuksia [133].

Aineiston koostamisessa alkuperäisestä aineistosta saadaan pienempi muodostamalla sen datapisteiden osajoukoista uusia datapisteitä, joiden ominaisuudet ovat niiden alkuperäisen aineiston datapisteiden ominaisuuksien funktiota, joista ne on koostettu [133]. Esimerkiksi verkkoliikenteen osalta yksittäisistä IP-paketeista voidaan koostaa eri lähteiden ja kohteiden välistä kommunikaatiota kuvaavia liikennevirtatietueita [36]. Luonnollisesti aineiston koostamisen yhteydessä on myös mahdollista menettää yksityiskohtia, joista olisi voinut olla hyötyä esimerkiksi tunkeutumisen havaitsemisen yhteydessä [133].

Otantaa käytetään yleisesti, kun varsinaiseen analyysin halutaan valita vain jokin mahdollisten havaintoyksiköiden osajoukko [133]. Verkkoliikenteeseen liittyen otantaa tai näytteen ottoa on sekä yksittäisten IP-pakettien että tietovirtatietueiden osalta käytetty vähentämään niiden käsittelyyn ja tallentamiseen vaadittavien resurssien määrää [60]. Otoksen edustavuuteen ja siten tulosten yleistettävyyteen vaikuttaa sekä otoksen koko että käytetty otantamenetelmä [133]. Vaikka yksinkertaisessa satunnaisotannassa jokaisella havaintoyksiköllä on yhtä suuri todennäköisyys tulla valituksi otokseen, ei se silti välttämättä ole paras otantamenetelmä jokaiseen ongelmaan [133]. Esimerkiksi ositetussa otannassa, jossa otanta tehdään eri aliryhmien osalta erikseen, voidaan paremmin varmistua siitä, että eri aliryhmät ovat edustettuina lopullisessa otoksessa [133].

Usein esimerkiksi poikkeaman tunnistusmenetelmien vertailuun on saatavilla aineisto, jonka taustatotuus tunnetaan. Jotta tällaisen aineiston avulla koulutetun menetelmän yleistyvyyttä voidaan arvioida, voidaan sen kouluttamiseen käyttää vain osaa käytettävissä olevista datapisteistä, jolloin jäljelle jääneitä datapisteitä voidaan vielä hyödyntää sen arviointiin [45, 133]. Opetukseen käytettyjen havaintoyksiköiden valintaan voidaan käyttää esimerkiksi yksinkertaista satunnaisotantaa tai ositettua otantaa määrittämällä aliryhmät taustatotuuden mukaan [133]. Tosin, esimerkiksi aikasarjojen yhteydessä datapisteiden järjestys tulee huomioida [43].

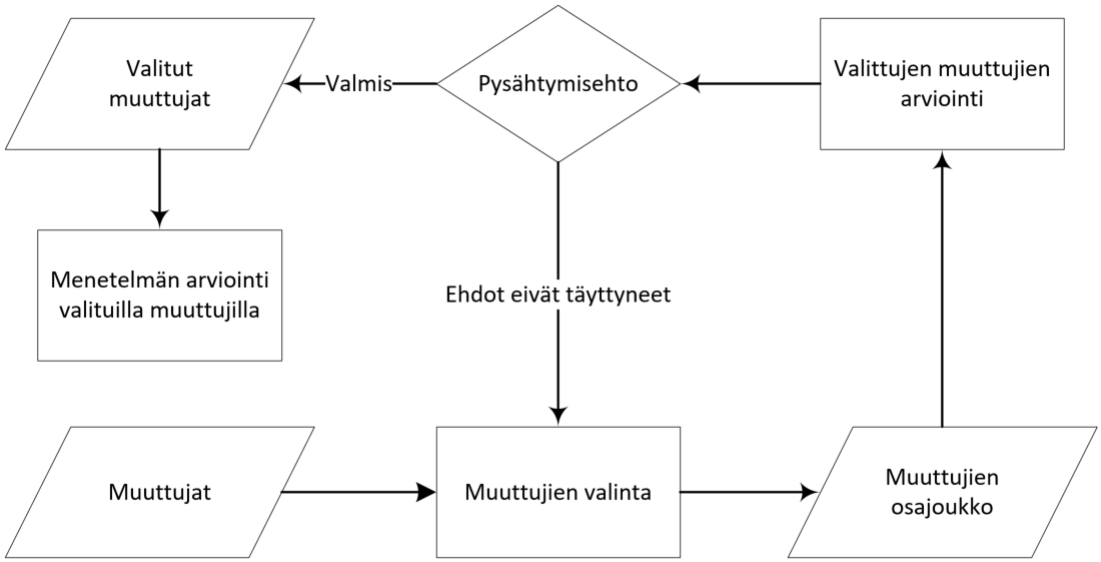
Ulottuvuuksien vähentämisellä tarkoitetaan yleensä menetelmää, jossa aineiston muuttujien määrää pyritään vähentämään käyttämällä alkuperäisten muuttujien sijaan uusia muuttujia, jotka ovat alkuperäisten funktioita [133]. Eräitä ulottuvuuksien vähentämiseen soveltuvia menetelmiä ovat spektraalisten poikkeaman tunnistusmenetelmien yhteydessä mainittu PCA sekä neuroverkkoihin perustuva autoenkooderi (Autoencoder, AE), jonka keskimmäisen piilokerroksen ulostuloja voidaan pitää uusina muuttujina [1]. PCA:ta käytettäessä uusiksi muuttujiksi valitaan taas tyypillisesti alkuperäisen aineiston varianssia hyvin säilyttävä joukko alkuperäisten muuttujien keskenään korreloimattomia lineaarikombinaatioita [69].

Vaikka monet poikkeaman tunnistusmenetelmien käyttämät algoritmit toimivatkin paremmin, kun ulottuvuuksia on vähemmän, on esimerkiksi pääkomponenttianalyysin yhteydessä hyvä muistaa, että alkuperäisen aineiston korrelaatiorakennetta rikkovat datapisteet näkyvät selvimmin juuri niissä pääkomponenteissa, jotka säilyttävät vain vähän alkuperäisen aineiston vaihtelua [26, 69, 133]. Ulottuvuuksien vähentämisellä voidaan myös vähentää dimensioiden kirouksen vaikutusta sekä tehdä aineistosta helpommin visualisoitava [133]. Dimensioiden kirouksella tarkoitetaan sitä, että ulottuvuuksien kasvaessa aineistosta tulee eksponentiaalisesti harvempaa ja samalla myös mahdollisesti vähemmän edustavaa [133]. Esimerkiksi jos aineiston muuttujilla voi olla kaksi eri arvoa, voi kaksiulotteisessa aineistossa olla $2^2 = 4$ erilaista datapistettä ja kolmiulotteisessa $2^3 = 8$.

Aineiston ulottuvuuksien määrää voidaan laskea myös käyttämällä vain osaa sen muuttujista [133]. Esimerkiksi liikennevirtatietueen sisältämä siirrettyjen tavujen määrä ilmaistuna eri kerrannaisyksiköissä sisältää vakiokerrointa vaille saman informaation. Tällaiset toisteiset muuttujat sisältävät saman tai lähes saman informaation kuin mitä jokin muu tai jotkin muut muuttujat yhdessä [133]. Liikennevirtatietueet saattavat myös sisältää muun muassa tietueet yksilöivän tunnisteen, jota voidaan pitää epäoleellisena verkkoliikenteestä tapahtuvan poikkeaman tunnistuksen kannalta. Epäoleellisten ja toisteisten muuttujien poistaminen ei juuri vaikuta aineiston sisältämän hyödyllisen informaation määrään [133].

Teoriassa optimaalisin muuttujien osajoukko tietylle menetelmälle saadaan valitsemalla kaikista muuttujien osajoukoista se, jolle valittu menetelmä tuottaa parhaat tulokset. Käytännössä tällaista lähestymistapaa ei kuitenkaan usein käytetä, vaikka sopiva arviointikriteeri löytyisikin, sillä mahdollisia muuttujien osajoukkoja on 2^n, kun muuttujia on n. Tyypillisesti sopivien osajoukkojen valintaan käytetäänkin eri lähestymistapoja. Joissain menetelmissä, joihin päätöspuutkin lukeutuvat,

muuttujien valinta on sisäänrakennettu. Suodattimiin perustuvissa lähestymistavoissa muuttujat valitaan esimerkiksi pareittaisen korreloimattomuuden perusteella ennen varsinaisen menetelmän suorittamista. Kääreeseen perustuvassa lähestymistavassa ei yleensä käydä läpi kaikkia muuttujien osajoukkoja, mutta muuten se vastaa lähestymistapaa, jolla teoriassa löydetään optimaalisin muuttujien osajoukko. Kuvassa 4.2 on esitetty vuokaavio muuttujien valinnalle silloin, kun se ei ole sisäänrakennettu ja menetelmää on mahdollista arvioida. [133]



Kuva 4.2: Vuokaavio muuttujien valinnasta [133]

Usein aineistoon voidaan myös luoda uusia muuttujia, jotka kuvaavat sitä sen alkuperäisiä muuttujia tehokkaammin. Jos näin saatu aineisto sisältää uusia muuttujia enemmän toisteisia muuttujia, jotka voidaan jättää valitsematta, saadaan samalla myös aineiston ulottuvuuksien määrää laskettua. Esimerkiksi valokuvia esittävistä pikselimatriiseista voidaan sopivalla piirreirrotusmenetelmällä muodostaa uusia muuttujia, jotka soveltuvat raaka-aineistoa paremmin halutun luokittelualgoritmin syötteeksi. Joskus irrotettu piirre voi olla myös alkuperäisten muuttujien yksinkertainen funktio. Piirreirrotuksen lisäksi uusia muuttujia voidaan muodostaa myös kuvaamalla aineiston datapisteet uuteen avaruuteen. Esimerkiksi aikasarjojen yhteydessä voidaan Fourier-muunnoksen avulla saada eri jaksonajan välein toistuvat ilmiöt esiin. [133]

Jotkut poikkeaman tunnistuksessa käytetyt algoritmit voivat vaatia, että havaintoyksiköt sisältävät vain luokittelu- ja järjestysasteikollisia muuttujia. Diskretoinnin avulla jatkuvistakin muuttujista saadaan tällaisia [133]. Tasavälisessä diskretoinnissa alkuperäisen muuttujan pienimmän x_0 ja suurimman x_n arvon muodostama väli voidaan jakaa yhtä suuriin osiin $\{[x_0, x_1],]x_1, x_2], ...,]x_{n-1}, x_n]\}$, joihin liitetään arvo, jonka uusi diskreetti muuttuja saa diskretoitavan muuttujan kuuluessa kyseiselle välille. Jos tasavälinen diskretointi tehdään esimerkiksi opetusaineiston perusteella, saadaan se kattamaan muuttujan koko arvoalue korvaamalla saaduista väleistä välit $[x_0, x_1]$ ja $]x_{n-1}, x_n]$ väleillä $]-\infty, x_1]$ ja $]x_{n-1}, \infty[$.

Tasaisten välien sijaan diskretoinnissa voidaan myös käyttää esimerkiksi välejä, jotka saadaan, kun vaaditaan, että jokaisessa välissä on yhtä monta diskretointiin käytetyn aineiston havaintoa. Diskretointi voidaan tehdä myös kahden tai useamman jatkuvan muuttujan yhteisjakaumalle, kun välien sijaan käytetään muuttujien määräämän avaruuden osia. Ohjatussa diskretoinnissa diskretoitavien muuttujien muodostama avaruus pyritään jakamaan osiin siten, että eri osissa on mahdollisimman paljon vain yhden taustatotuuden omaavia havaintoja. Tällöin aineiston täytyy kuitenkin sisältää eri taustatotuuden omaavia datapisteitä. [133]

Binarisoinnissa alkuperäisestä muuttujasta luodaan uusia muuttujia, jotka voivat saada vain kaksi eri arvoa [133]. Luokittelu- ja järjestysasteikollisten muuttujien binarisointiin voidaan käyttää esimerkiksi one-hot-koodausta, jossa jokaista alkuperäisen muuttujan saamaa eri arvoa kohden luodaan uusi muuttuja, joka saa arvon yksi tai nolla riippuen siitä, saako alkuperäinen muuttuja uutta muuttujaa vastaavan arvon vai ei [1]. Koska binarisoinnissa uusien muuttujien määrä riippuu alkuperäisen muuttujan arvojoukon koosta, tulee aineistosta uusien muuttujien myötä myös helposti hyvin korkeaulotteinen ja harva [1, 133].

Muuttujien muunnoksiin voidaan käyttää yksinkertaisia funktioita tai esimerkiksi standardointia ja normalisointia. Yksinkertaisten funktioiden avulla alkuperäinen muuttuja voidaan muun muassa saada noudattamaan jotain tunnettua jakaumaa tai soveltumaan paremmin halutun menetelmän syötteeksi. Standardointia ja normalisointia taas käytetään tyypillisesti, kun muuttujien arvoista halutaan sellaisia, että niiden mittayksiköt eivät vaikuta käytetyn menetelmän tuloksiin. Standardoinnissa muuttujan keskiarvoksi saadaan nolla ja keskihajonnaksi yksi jakamalla alkuperäiset arvot, joista on vähennetty alkuperäinen keskiarvo, alkuperäisellä keskihajonnalla. Näin esimerkiksi euklidinen etäisyys saadaan paremmin huomioimaan erot muuttujissa, joiden keskihajonta on pieni. [133]

4.3 Etäisyys- ja samankaltaisuusmittoja

Datapisteiden erilaisuus ja samankaltaisuus ovat tärkeitä, koska niitä tarvitaan muun muassa lähimpiin naapureihin perustuvissa ja klusterointia hyödyntävissä poikkeaman tunnistusmenetelmissä. Vaikka ne molemmat periaatteessa ovatkin datapisteiden ominaisuuksien funktiota, eroavat ne kuitenkin ainakin siten, että datapisteiden välisen erilaisuuden kasvaessa niiden samankaltaisuus vähenee [133]. Samankaltaisuus saa tyypillisesti arvoja väliltä $[0, 1]$. Metriikka [101] eli etäisyysfunktio on puolestaan datapisteiden X erilaisuuden mitta $d : X \times X \to \mathbb{R}$, jolle kaikilla $x, y, z \in X$

1. $d(x, y) \geq 0$
2. $d(x, y) = 0 \Leftrightarrow x = y$
3. $d(x, y) = d(y, x)$
4. $d(x, z) \leq d(x, y) + d(y, z)$.

Eri erilaisuus- ja samankaltaisuusmitat soveltuvat erilaisiin tilanteisiin. Esimerkiksi euklidista etäisyyttä, joka on yleisen Minkowskin etäisyyden

$$d(x, y) = \Big(\sum_{k=1}^{n} |x_k - y_k|^p \Big)^{\frac{1}{p}}$$

erikoistapaus, kun $p = 2$, käytetään usein kun aineisto ei ole harvaa, ja sen muuttujat ovat jatkuvia. Se ja muut Minkowskin etäisyydet ovat hyviä valintoja myös, kun etäisyyden halutaan huomioivan erot datapisteiden kaikkien ominaisuuksien suhteen. Jaccardin indeksi ja kosini ovat puolestaan samankaltaisuusmittoja, joita voidaan käyttää, kun datapisteiden nolla-arvoisten komponenttien vastaavuuden ei haluta vaikuttavan niiden samankaltaisuuteen. [133]

Esimerkiksi binäärisistä ominaisuuksista koostuvien datapisteiden $(0, 1, 1, 0)$ ja $(0, 0, 1, 0)$ Jaccardin indeksi, joka tässä on $\frac{1}{2}$, saadaan, kun sellaisten komponenttien, jotka molemmissa datapisteissä saavat arvon 1, määrä jaetaan niiden komponenttien määrällä, jotka ainakin toisessa datapisteistä poikkeavat nollasta. Datapisteiden eli vektorien välinen kosini

$$cos(x, y) = \frac{x^T y}{||x||\,||y||}$$

taas saadaan lineaarialgebran keinoin, kun vektorien pistetulo jaetaan niiden normien tulolla. Eri erilaisuus ja samankaltaisuusmittoja on useita, ja ne voivat myös huomioida esimerkiksi aineiston jakauman. [133]

4.4 Suorituskykymittareita

Jos poikkeaman tunnistukseen käytetyn menetelmän arvioinnissa on mahdollista hyödyntää aineistoa, jonka taustatotuus tunnetaan, voidaan sen ennustuskykyä arvioida sekaannusmatriisin avulla [133]. Koska verkkoliikenteestä suoritettu tunkeutumisen havaitseminen on tyypillisesti joko puoliohjattua tai ohjaamatonta poikkeaman tunnistusta [1, 26], on taustatotuudella yleensä vain kaksi mielekästä arvoa [1, 26, 133]: esimerkiksi normaali ja tunkeutuminen. Koska taustatotuuksia on kaksi, saadaan yleisestä $n \times n$ -sekaannusmatriisista 2×2 -matriisi [46]. Taulukossa 4.1 on esitetty tällainen matriisi. Sen riveiltä näkee ennustetun luokan, ja sen sarakkeilta todellisen luokan eli taustatotuuden [46].

Taulukko 4.1: Sekaannusmatriisi [46]

	Tunkeutuminen	Normaali
Tunkeutuminen	Todelliset positiiviset (True Positives, TP)	Väärät positiiviset (False Positives, FP)
Normaali	Väärät negatiiviset (False Negatives, FN)	Todelliset negatiiviset (True Negatives, TN)

Jos poikkeaman tunnistuksessa käytetty menetelmä kertoo vain, ovatko havaintoyksiköt poikkeamia vai eivät, saadaan TP, FP, FN ja TN vastaavasti oikein tunnistettujen poikkeamien, väärin poikkeamiksi tunnistettujen, väärin normaaleiksi tunnistettujen ja oikein tunnistettujen normaalien havaintoyksiköiden määristä [46]. Toisaalta jos arvioitavan menetelmän avulla on mahdollista saada havaintoyksiköiden poikkeavuutta kuvaavat useampiarvoiset pisteytykset, ovat TP, FP, FN ja TN niihin käytetyn kynnysarvon t funktiota [1]. Näiden funktioiden lausekkeet on esitetty taulukossa 4.2, jossa joukko $S(t)$ on kynnysarvolla t poikkeamiksi ennustetut tapaukset ja jossa joukko G on todelliset poikkeamat.

Taulukko 4.2: Sekaannusmatriisi kynnysarvon t funktiona

	Tunkeutuminen	Normaali
Tunkeutuminen	$\lvert S(t) \cap G \rvert$	$\lvert S(t) \cap \neg G \rvert$
Normaali	$\lvert \neg S(t) \cap G \rvert$	$\lvert \neg S(t) \cap \neg G \rvert$

Sekaannusmatriisin avulla saadaan myös laskettua muita tunnuslukuja. Tarkkuus (Precision)

$$Tarkkuus(t) = \frac{TP(t)}{TP(t) + FP(t)}$$

kertoo, kuinka suuri osa ennustetuista poikkeamista on oikeasti poikkeamia. Vastaavasti todellisten positiivisten osuus (True Positive Rate, TPR)

$$TPR(t) = \frac{TP(t)}{TP(t) + FN(t)}$$

kertoo, kuinka suuri osuus todellisista poikkeamista on ennustettu poikkeamiksi. Todellisten positiivisten osuudesta käytetään myös nimityksiä saanti (Recall) ja osumatarkkuus (Hit Rate). Väärien positiivisten osuuden (False Positive Rate, FPR)

$$FPR(t) = \frac{FP(t)}{FP(t) + TN(t)}$$

avulla voidaan puolestaan arvioida väärien hälytysten määrää, sillä se kertoo, kuinka suuri osuus arviointiin käytetyn aineiston todellisesti normaaleista tapauksista on ennustettu olevan poikkeamia. Virheettömyys (Accuracy)

$$Virheettömyys(t) = \frac{TP(t) + TN(t)}{TP(t) + FP(t) + FN(t) + TN(t)}$$

taas kertoo sen, kuinka suuri osa tehdyistä ennusteista oli oikeita. F-mitta (F-measure)

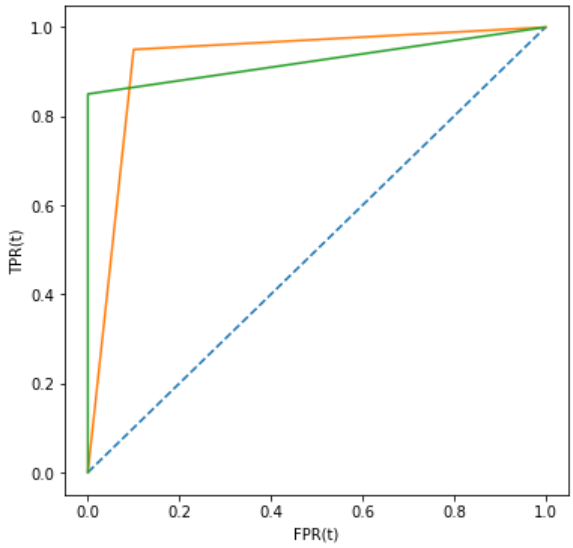
$$F\text{-}mitta(t) = \frac{2}{\frac{1}{Tarkkuus(t)} + \frac{1}{TPR(t)}}$$

on tarkkuuden ja todellisten positiivisten osuuden harmoninen keskiarvo. [46]

Poikkeaman tunnistukseen käytetyn menetelmän arvioinnissa voidaan käyttää myös pisteiden $\{(FPR(t), TPR(t)) \mid t \in R\}$ kautta kulkevaa ROC-käyrää. Koska käytetyssä aineistossa tulisi aina olla eri taustatotuuden omaavia havaintoyksiköitä, kulkee ROC-käyrä pisteiden $(0,0)$ ja $(1,1)$ kautta. Piste $(0,0)$ vastaa tilannetta, jossa kaikkien havaintoyksiköiden ennustetaan olevan normaaleja. Tällöin ei synny vääriä positiivisia, mutta ei toisaalta myöskään todellisia positiivista. Tilannetta, jossa kaikkien havaintoyksiköiden ennustetaan olevan poikkeamia, vastaa piste $(1,1)$. Tällöin ei ole vääriä negatiivisia eikä todellisia negatiivisia. [46]

Kuvassa 4.3 on visualisoitu kahden eri poikkeaman tunnistusmenetelmän ennustuskykyä ROC-käyrien avulla. Molemmat vertailluista menetelmistä ennustavat suoraan havaintoyksiköiden taustatotuutta, eivätkä siten tarjoa erillistä ennusteen varmuudesta [46] kertovaa anomaliapisteytystä. Kuvassa näkyy myös pisteitä

$(0,0)$ ja $(1,1)$ yhdistävä jana, joka edustaa täysin satunnaisen menetelmän ROC-käyrää äärettömän suuressa aineistossa [46]. Täydelliseltä tunnistusmenetelmältä vaaditaan puolestaan, että $]a_n, b_n[\cap]a_p, b_p[= \emptyset$, kun $]a_n, b_n[$ on normaalien ja kun $]a_p, b_p[$ on poikkeamien anomaliapisteytysten arvojoukko. Sitä kuvaava ROC-käyrä kulkisi pisteiden $(0,0)$, $(0,1)$ ja $(1,1)$ kautta.



Kuva 4.3: Kaksi ROC-käyrää

Kuvasta 4.3 nähdään, että menetelmä, jota esittävä ROC-käyrä kulkee pisteen $(0;0,85)$ kautta, löysi 85 % kaikista anomalioista ilman vääriä hälytyksiä. Kuvasta nähdään myös, että toinen menetelmistä löysi 95 % kaikista anomalioista väärien hälytysten osuuden ollessa 10 %, sillä sitä esittävä ROC-käyrä kulkee pisteen $(0,1;0,95)$ kautta. Koska menetelmät eivät tuottaneet anomaliapisteytyksiä, saadaan kynnysarvoa t muuttamalla vain kolme pistettä. Toinen menetelmistä siis löytää enemmän poikkeamia, mutta tuottaa myös vääriä hälytyksiä.

ROC-käyrän ja vaaka-akselin välistä pinta-alaa (Area Under the ROC Curve, AUC) voidaan myös käyttää poikkeaman tunnistusmenetelmien arvioinnissa. Arvioitavan menetelmän AUC on tyypillisesti vähintään $\frac{1}{2}$, joka vastaa täysin satunnaisen menetelmän ROC-käyrän alle jäävää pinta-alaa, ja enintään 1, joka vastaa täydellisen menetelmän ROC-käyrän alle jäävää pinta-alaa. Koska AUC lasketaan pisteistä $(FPR(t_i), TPR(t_i))$ puolisuunnikassäännön avulla, on sen arvojoukko $[0,1]$. Menetelmän AUC on myös todennäköisyys sille, että se pitää satunnaista normaalia tapausta vähemmän poikkeavana kuin satunnaista poikkeamaa. [1, 46]

5 Valitut poikkeaman tunnistamismenetelmät

Luvun 5 tarkoituksena on kuvata Davidowin ja Mattesonin [35] esittämä kvantitatiivisten ja kvalitatiivisten muuttujien faktorianalyysiä (Factor Analysis of Mixed Data, FAMD) hyödyntävä menetelmä (Factor Analysis of Mixed Data for Anomaly Detection, FAMDAD) sekä sen vertailukohtana käytetyt muut menetelmät. Mahalanobiksen etäisyyden lisäksi FAMDAD-menetelmän vertailukohtana tässä työssä käytetään yksinkertaista autoenkoodereita hyödyntävää poikkeaman tunnistusmenetelmää. Valittujen menetelmien varsinainen vertailu eri verkoista kerätyillä poikkeaman tunnistukseen tarkoitetuilla tietojoukoilla suoritetaan luvussa 6.

Koska FAMDAD-menetelmässä suoritetaan kvantitatiivisten ja kvalitatiivisten muuttujien faktorianalyysi, jossa kvantitatiivisia eli välimatka- tai suhdeasteikollisia muuttujia painotetaan vielä niiden huipukkuuteen perustuvilla kertoimilla, käydään tässä luvussa lyhyesti läpi myös FAMD, jossa määrällisille muuttujille tarkoitettua pääkomponenttianalyysiä laajennetaan laadullisiin muuttujiin [105]. Koska FAMD voidaan suorittaa myös tavallisen PCA:n avulla aineistolle, joka on esikäsitelty menetelmän vaatimalla tavalla, esitellään tässä luvussa myös pääkomponenttianalyysi, joka liittyy läheisesti myös Mahalanobiksen etäisyyteen [69, 105].

5.1 Mahalanobiksen etäisyys

Mahalanobiksen etäisyys on multinormaalijakaumaa noudattavaan aineistoon sopiva metriikka, joka toisin kuin esimerkiksi euklidinen etäisyys huomioi myös muuttujien varianssin sekä niiden välisen korrelaation [133]. Kun jakaumaa, jonka kovarianssimatriisi on C, tavallisesti noudattavan aineiston datapisteiden x_i Mahalanobiksen etäisyys lasketaan tavallisen aineiston generoineen jakauman odotusarvoon μ, voidaan metriikkaa

$$Mahalanobis(x_i, x_j) = \sqrt{(x_i - x_j)^T C^{-1} (x_i - x_j)} \tag{5.1}$$

käyttää myös parametrisena tilastollisena poikkeaman tunnistusmenetelmänä [133]. Mahalanobiksen etäisyyden yhteys multinormaalijakauman todennäköisyysmassan

tiheyteen käy ilmi, kun n-ulotteisen normaalijakauman tiheysfunktio [133]

$$\begin{aligned} f(x) =& \frac{1}{\sqrt{(2\pi)^n (det\, C)}} e^{-\frac{(x-\mu)^T C^{-1}(x-\mu)}{2}} \\ =& \frac{1}{\sqrt{(2\pi)^n (det\, C)}} e^{-\frac{Mahalanobis(x,\mu)^2}{2}} \end{aligned}$$

kirjoitetaan siten, että $Mahalanobis(x, \mu)$ esiintyy siinä terminä. Selvästi datapisteen x ja jakauman odotusarvon μ välisen Mahalanobiksen etäisyyden kasvaessa tiheysfunktion f, jossa Neperin luvun kerroin on vakio, saamat arvot pienenevät.

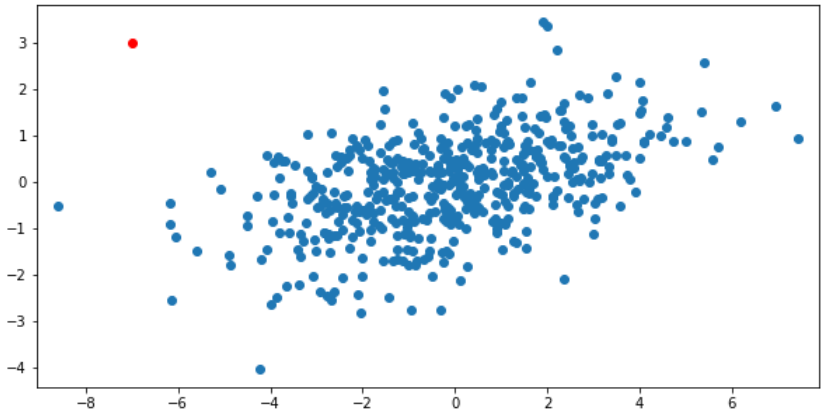
Koska Mahalanobiksen etäisyyden laskemiseen tarvitaan C^{-1}, täytyy sen laskemiseen käytetyn satunnaisvektorin kovarianssimatriisin olla kääntyvä [1]. Kaavasta 5.1 nähdään myös, että jos C tai sitä estimoimaan käytetty otoskovarianssimatriisi S on yksikkömatriisi I, on Mahalanobiksen etäisyys itse asiassa sama kuin euklidinen etäisyys, sillä $\sqrt{(x_i - x_j)^T I^{-1}(x_i - x_j)} = \sqrt{(x_i - x_j)^T (x_i - x_j)}$. Tällöin muuttujien varianssi on siis jo valmiiksi yksi, eivätkä ne korreloi keskenään. Kovarianssimatriisin C käänteismatriisin tehtävänä kaavassa 5.1 onkin valkaista pisteet $x_i - \mu$ ja $x_j - \mu$ ennen niiden välisen euklidisen etäisyyden laskemista.

Multinormaalijakaumaa $\mathcal{N}(\mu, C)$ noudattavasta satunnaisvektorista X keskittämällä saatu satunnaisvektori $X - \mu \sim \mathcal{N}(0, C)$ saadaan valkaistua lineaarikuvauksella $Z = C^{-\frac{1}{2}}(X - \mu)$ [71]. Tällöin $Z \sim \mathcal{N}(0, I)$ [106]. Koska tällaista keskitetyn satunnaisvektorin ZCA-valkaisua (Zero-Phase Component Analysis, ZCA), jota kutsutaan myös Mahalanobis-valkaisuksi, vastaa symmetrinen matriisi $C^{-\frac{1}{2}}$, jolle $C^{-\frac{1}{2}} C^{-\frac{1}{2}} = C^{-1}$ [71, 106], voidaan Mahalanobiksen etäisyys esittää myös keskitettyjen ja valkaistujen pisteiden välisenä euklidisena etäisyytenä, kun merkitään, että z_i ja z_j ovat pisteitä x_i ja x_j vastaavat keskitetyt ja valkaistut pisteet.

$$\begin{aligned} Mahalanobis(x_i, x_j) =& \sqrt{(x_i - x_j)^T C^{-1}(x_i - x_j)} \\ =& \sqrt{(x_i - x_j)^T C^{-\frac{1}{2}} C^{-\frac{1}{2}}(x_i - x_j)} \\ =& \sqrt{(C^{-\frac{1}{2}}(x_i - x_j))^T (C^{-\frac{1}{2}}(x_i - x_j))} \\ =& \sqrt{(C^{-\frac{1}{2}}((x_i - \mu) - (x_j - \mu)))^T (C^{-\frac{1}{2}}((x_i - \mu) - (x_j - \mu)))} \\ =& \sqrt{(C^{-\frac{1}{2}}(x_i - \mu) - C^{-\frac{1}{2}}(x_j - \mu))^T (C^{-\frac{1}{2}}(x_i - \mu) - C^{-\frac{1}{2}}(x_j - \mu))} \\ =& \sqrt{(z_i - z_j)^T (z_i - z_j)} \end{aligned}$$

Edellisestä yhtälöstä nähdään lisäksi, että $Mahalanobis(x, \mu)$ voidaan kirjoittaa myös muodossa $\sqrt{z^T z}$, kun z on satunnaisvektorin X jakaumasta generoitua pistettä x vastaava keskitetty ja valkaistu piste. Koska n-ulotteista normaalijakaumaa noudattavan satunnaisvektorin muuttujien korreloimattomuus tarkoittaa samalla myös niiden riippumattomuutta, on $z^T z = \sum_{i=1}^{n} z_i^2$ itse asiassa tällöin riippumattomien standardinormaalijakautuneiden lukujen neliöiden summa, sillä multinormaalijakaumaa noudattavan satunnaisvektorin yksittäiset satunnaismuuttujat ovat myös normaalijakautuneita [9]. Siten, kun x on generoitu n-ulotteisesta normaalijakaumasta, $Mahalanobis(x, \mu)^2$ noudattaa χ^2-jakaumaa vapausasteella n.

Kuvassa 5.1 on esitetty kaksiulotteisesta normaalijakaumasta $\mathcal{N}(\mu, C)$ generoitu aineisto, jossa on yksittäinen poikkeama. Koska edellä todettiin, että tällaista multinormaalijakaumaa noudattavan aineiston havaintoyksiköiden neliöity Mahalanobiksen etäisyys jakauman odotusarvoon noudattaa χ^2-jakaumaa vapausasteella 2, voidaan datapisteiden poikkeavuuden arvioinnissa hyödyntää χ^2-jakauman kertymäfunktiota [1]. Tässä poikkeaman $(-7, 3)$ ja jakauman odotusarvon $(0, 0)$ Mahalanobiksen etäisyyden neliöksi saadaan 34. Koska $P(X \geq 34) < 10^{-7}$, kun $X \sim \chi_2^2$, nähdään, että on hyvin epätodennäköistä, että poikkeava havaintoyksikkö olisi peräisin samasta jakaumasta kuin muut datapisteet.



$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 5 & 1 \\ 1 & 1 \end{bmatrix}$$

Kuva 5.1: Jakaumaa $\mathcal{N}(\mu, C)$ noudattava aineisto, jossa on poikkeama (-7,3).

Koska parametrisena tilastollisena poikkeaman tunnistusmenetelmänä käytetty Mahalanobiksen etäisyys voidaan laskea, kun μ ja C^{-1} tai niiden estimaatit tunnetaan, ei siihen monista muista menetelmistä poiketen liity asetuksia, joiden optimaalisten arvojen löytämiseen tarvittaisiin tietoa havaintoyksiköiden taustatotuudesta [1]. Koska satunnaisvektorin X affiineilla muunnoksilla $AX + b$ ei ole vaikutusta

Mahalanobiksen etäisyyteen, kun A on täyttä astetta oleva neliömatriisi [121], ei itse asiassa silläkään ole väliä, koostuvatko sen laskemiseen käytetyt vektorit alkuperäisten vai standardoitujen muuttujien arvoista. Tämä on selvää sillä, kun A on diagonaalimatriisi, jonka päälävistäjän arvot ovat alkuperäisten muuttujien keskihajontojen käänteislukuja, on $Z = A(X - \mu) = AX - A\mu$ affiinilla muunnoksella saatu keskitetty satunnaisvektori, jonka muuttujilla on yksikkövarianssi.

Serflingin [121] mainitsema affiineihin muunnoksiin liittyvä ominaisuus voidaan osoittaa, kun ensin tarkastellaan, miten muunnos $AX + b$, jossa $A_{n\times n}$ on täyttä astetta ja siten kääntyvä, vaikuttaa alkuperäisen jakauman kovarianssimatriisin käänteismatriisiin $C^{-1} = \mathbb{E}((X - \mu)(X - \mu)^T)^{-1}$. Koska muunnoksen avulla saadun uuden jakauman kovarianssimatriisin käänteismatriisiksi saadaan

$$\begin{aligned}
&\mathbb{E}((AX + b - \mathbb{E}(AX + b))(AX + b - \mathbb{E}(AX + b))^T)^{-1} \\
=&\mathbb{E}((AX + b - A\mathbb{E}(X) - b)(AX + b - A\mathbb{E}(X) - b)^T)^{-1} \\
=&\mathbb{E}(A(X - \mu)(A(X - \mu))^T)^{-1} \\
=&(A\mathbb{E}((X - \mu)(X - \mu)^T)A^T)^{-1} \\
=&(ACA^T)^{-1}
\end{aligned}$$

odotusarvon lineaarisuuden perusteella, nähdään, että kaikilla i, j

$$\begin{aligned}
Mahalanobis(z_i, z_j) =&\sqrt{(z_i - z_j)^T(ACA^T)^{-1}(z_i - z_j)} \\
=&\sqrt{(Ax_i + b - (Ax_j + b))^T(ACA^T)^{-1}(Ax_i + b - (Ax_j + b))} \\
=&\sqrt{(A(x_i - x_j))^T(ACA^T)^{-1}A(x_i - x_j)} \\
=&\sqrt{(x_i - x_j)^TA^T(A^T)^{-1}C^{-1}A^{-1}A(x_i - x_j)} \\
=&\sqrt{(x_i - x_j)^TC^{-1}(x_i - x_j)} \\
=&Mahalanobis(x_i, x_j),
\end{aligned}$$

kun $z_i = Ax_i + b$ kaikilla i.

Vaikka vain multinormaalijakaumaa noudattavan aineiston Mahalanobiksen etäisyyden neliö jakauman odotusarvoon noudattaa χ^2-jakaumaa, voidaan Mahalanobiksen etäisyyttä käyttää myös muun tyyppisissä aineistoissa anomaliapisteytyksenä, joka ottaa huomioon aineiston painopisteen ja sen datapisteiden välisen etäisyyden lisäksi myös sen korrelaatiorakenteen. On kuitenkin huomattava, että jos aineisto koostuu klustereista, ei menetelmä ole välttämättä kovin tehokas. [1]

5.2 Pääkomponenttianalyysi

Pääkomponenttianalyysi on luultavasti vanhin ja parhaiten tunnettu moniulotteisen datan analysointiin käytetty tekniikka. Tyypillisesti sen avulla pyritään löytämään alkuperäistä keskenään korreloivaa muuttujajoukkoa pienempi joukko keskenään korreloimattomia muuttujia, jotka kuitenkin säilyttävät suurimman osan alkuperäisten muuttujien varianssista. Uudet muuttujat eli alkuperäisen satunnaisvektorin $[X_1, X_2, ..., X_n]^T$ lineaarikombinaatiot $w_1^T X, w_2^T X, ..., w_n^T X$ muodostetaan siten, että jokaisen muuttujan $w_i^T X$ varianssi on mahdollisimman suuri huomioiden se, että ne eivät saa korreloida aikaisempien muuttujien $\{w_j^T X | 1 \leq j < i\}$ kanssa. [69]

Näin saatuja uusia muuttujia $w_1^T X, w_2^T X, ..., w_n^T X$ kutsutaan pääkomponenteiksi, joista usein siis toivottaisiin voida valita vain $m \ll n$ ensimmäistä niin, että loput $w_{m+1}^T X, w_{m+2}^T X, ..., w_n^T X$ eivät enää säilyttäisi paljoakaan alkuperäisten muuttujien varianssista. Toisaalta poikkeaman tunnistuksessa usein juuri nämä viimeiset pääkomponentit $w_i^T X$ ovat mielenkiintoisia, sillä aineiston normaalia korrelaatiorakennetta rikkovien datapisteiden x_j pistetulot $w_i^T x_j$ sijaitsevat kauempana näiden pääkomponenttien odotusarvoista $\mathbb{E}(w_i^T X)$ kuin normaalien datapisteiden. [69]

Ensimmäiseksi pääkomponenttianalyysissa pyritään siis löytämään lineaarikombinaatio $w_1^T X$, jonka varianssi on mahdollisimman suuri [69]. Koska satunnaisvektorin X, minkä tahansa lineaarikombinaation $a^T X$ varianssi on $\mathrm{Var}(a^T X) = a^T C a$, kun C on X:n kovarianssimatriisi, saadaan, että w_1 on ensimmäisen pääkomponentin määräävä vektori, joka maksimoi $\mathrm{Var}(w_1^T X) = w_1^T C w_1$, kun vaaditaan lisäksi, että $||w_1||^2 = w_1^T w_1 = 1$, jotta vektorin w_1 komponentit olisivat äärelliset [68, 69]. Asetetun optimointitehtävän rajoitteet huomioivaksi Lagrangen funktioksi ja sen osittaisderivaatoiksi saadaan siten

$$\begin{aligned}
\mathcal{L}(w_1, \lambda) &= w_1^T C w_1 - \lambda(w_1^T w_1 - 1) \\
\frac{\partial}{\partial w_1}\mathcal{L}(w_1, \lambda) &= (C + C^T)w_1 - \lambda(I + I^T)w_1 = 2Cw_1 - 2\lambda w_1 \\
\frac{\partial}{\partial \lambda}\mathcal{L}(w_1, \lambda) &= 1 - w_1^T w_1.
\end{aligned}$$

Koska $\mathcal{L}$ on jatkuvasti differentioituva ja koska $\nabla(w_1^T w_1 - 1) = 2w_1$ on nolla vain, jos $w_1 = 0$, on optimointitehtävän $\mathrm{argmax}_{w_1 : ||w_1|| = 1} w_1^T C w_1$ ratkaisun oltava molempien osittaisderivaattojen nollakohta [28]. Toisin sanoen vektorin w_1 on toteutettava yhtälöt $2Cw_1 - 2\lambda w_1 = 0 \Leftrightarrow Cw_1 = \lambda w_1$ ja $1 - w_1^T w_1 = 0 \Leftrightarrow w_1^T w_1 = 1$. Koska C on neliömatriisi ja koska w_1 ei saa olla nollavektori, seuraa ensimmäisestä

yhtälöstä, että vektorin w_1 täytyy olla kovarianssimatriisin ominaisvektori ja vastaavasti Lagrangen kertoimen λ sitä vastaava ominaisarvo [28, 69]. Yhtälöistä saadaan myös, että $w_1^T C w_1 = w_1^T \lambda w_1 = \lambda ||w_1||^2 = \lambda$, joten $\mathrm{Var}(w_1^T X) = \lambda$ maksimoidaan, kun λ on C:n suurin ominaisarvo ja w_1 sitä vastaava ominaisvektori [69].

Seuraavat vektorit $w_2, w_3 ..., w_n$ ja siten myös pääkomponentit $w_2^T X, w_3^T X, ..., w_n^T X$ löydetään vastaavasti kuin ensimmäinen, kun lisäksi vaaditaan, että vektoria w_k etsittäessä lineaarikombinaatio $w_k^T X$ ei saa korreloida edellisten jo löydettyjen pääkomponenttien $w_{k-1}^T X, w_{k-2}^T X, ..., w_1^T X$ kanssa [69]. Koska $\mathrm{Cov}(w_2^T X, w_1^T X) = w_2^T C w_1$ [68], saadaan toista pääkomponenttia $w_2^T X$ vastaavan vektorin w_2 etsimiseen tarvittavaksi mainitut lisärajoitteet huomioivaksi Lagrangen funktioksi ja sen osittaisderivaatoiksi

$$\begin{aligned}
\mathcal{L}(w_2, \mu) &= w_2^T C w_2 - \mu_2 (w_2^T w_2 - 1) - \mu_1 (w_2^T C w_1) \\
\frac{\partial}{\partial w_2} \mathcal{L}(w_2, \mu) &= 2 C w_2 - 2 \mu_2 w_2 - \mu_1 C w_1 \\
\frac{\partial}{\partial \mu_2} \mathcal{L}(w_2, \mu) &= 1 - w_2^T w_2 \\
\frac{\partial}{\partial \mu_1} \mathcal{L}(w_2, \mu) &= -w_2^T C w_1.
\end{aligned}$$

Koska $\mathcal{L}$ on jälleen jatkuvasti differentioituva ja koska ehtojen gradienttivektorit $\nabla(w_2^T w_2 - 1) = 2 w_2$ ja $\nabla(w_2^T C w_1) = C w_1$ kohdassa w_2^* ovat lineaarisesti riippumattomia rajoitteen $w_2^T C w_1 = w_2^T \lambda w_1 = \lambda w_2^T w_1 = 0$ ja tiedon $\lambda \neq 0$ perusteella, on ratkaisun oltava osittaisderivaattojen nollakohta [28]. Kun yhtälön $\frac{\partial}{\partial w_2} \mathcal{L}(w_2, \mu) = 0$ molemmat puolet kerrotaan vasemmalta rivivektorilla w_1^T, saadaan rajoitteiden perusteella

$$\begin{aligned}
&2 w_1^T C w_2 - 2 \mu_2 w_1^T w_2 - \mu_1 w_1^T C w_1 = 0 \\
&\quad \Leftrightarrow 2 w_2^T C w_1 - 2 \mu_2 w_2^T w_1 - \mu_1 \lambda = 0 \Rightarrow \mu_1 = 0.
\end{aligned}$$

Kun yhtälöön $\frac{\partial}{\partial w_2} \mathcal{L}(w_2, \mu) = 0$ sijoitetaan $\mu_1 = 0$, saadaan, vastaavasti kuin ensimmäisen pääkomponentin kohdalla, että $\mathrm{Var}(w_2^T X) = \mu_2$ on C:n ominaisarvo ja w_2 sitä vastaava ominaisvektori [69]. Koska pääkomponentit $w_2^T X$ ja $w_1^T X$ eivät saa korreloida, saadaan, että w_2 on kovarianssimatriisin C kohtisuorassa vektoria w_1 vastaan oleva ominaisvektori, jonka ominaisarvo μ_2 on mahdollisimman suuri. Vastaavasti menetellen löydetään loputkin pääkomponentit niitä vastaavien ominaisarvojen mukaan laskevassa järjestyksessä [68, 69]. Jos pääkomponentteja vastaa sama ominaisarvo, ei niiden keskinäistä järjestystä ole määritelty [68].

Kaikki kovarianssimatriisit C ovat $n \times n$-neliömatriiseja ja määritelmänsä mukaan symmetrisiä, joten niille voidaan aina löytää reaaliset ominaisarvot $\lambda_1, \lambda_2, ..., \lambda_n$ sekä niitä vastaavat keskenään ortogonaaliset ominaisvektorit $w_1, w_2, ..., w_n$ [28]. Koska kovarianssimatriisit ovat lisäksi positiivisesti semidefiniittejä

$$x^T C x = x^T \mathbb{E}((X-\mu)(X-\mu)^T)x = \mathbb{E}(((X-\mu)^T x)^T (X-\mu)^T x) \geq 0 \ \forall \ x,$$

tiedetään, että saadut ominaisarvot ovat kelvollisia variansseja [28]. Satunnaisvektorin tai aineiston pääkomponentit määräävät ominaisvektorit voidaan siten aina selvittää ratkaisemalla ensin kovarianssimatriisin tai otoskovarianssimatriisin A ominaisarvoyhtälöstä $Ax = \lambda x \Leftrightarrow (A - \lambda I)x = 0$ saadusta karakteristisesta yhtälöstä $det(A - \lambda I) = 0$ ominaisarvot [68]. Kun ominaisarvot tunnetaan, halutut ominaisvektorit voidaan ratkaista esimerkiksi ominaisarvoyhtälön avulla.

Otetaan esimerkiksi Mahalanobiksen etäisyyden yhteydessä käsitelty multinormaalijakauma $\mathcal{N}(\mu, C)$, jossa

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 5 & 1 \\ 1 & 1 \end{bmatrix}$$

ja etsitään sen pääkomponentit. Ratkaistaan siis ensimmäiseksi kovarianssimatriisin karakteristisen yhtälön $det(C - \lambda I)$ nollakohdat eli matriisin C ominaisarvot. Koska

$$\begin{aligned} det(C - \lambda I) = 0 &\Leftrightarrow det\left(\begin{bmatrix} 5 & 1 \\ 1 & 1 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \right) = 0 \\ &\Leftrightarrow det\left(\begin{bmatrix} 5-\lambda & 1 \\ 1 & 1-\lambda \end{bmatrix} \right) = 0 \\ &\Leftrightarrow (5-\lambda)(1-\lambda) - 1 \cdot 1 = 0 \\ &\Leftrightarrow \lambda^2 - 6\lambda + 4 = 0 \Leftrightarrow (\lambda - 3)^2 = 5, \end{aligned}$$

nähdään, että yhtälön juuret λ_1, λ_2 ovat $3 \pm \sqrt{5} > 0$. Ominaisarvoyhtälöstä taas

$$(C - \lambda I)x = 0 \Leftrightarrow \begin{bmatrix} 5-\lambda & 1 \\ 1 & 1-\lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

saadaan hyödyntämällä, että $\lambda^2 - 6\lambda + 4 = 0$, Gaussin ja Jordanin menetelmällä

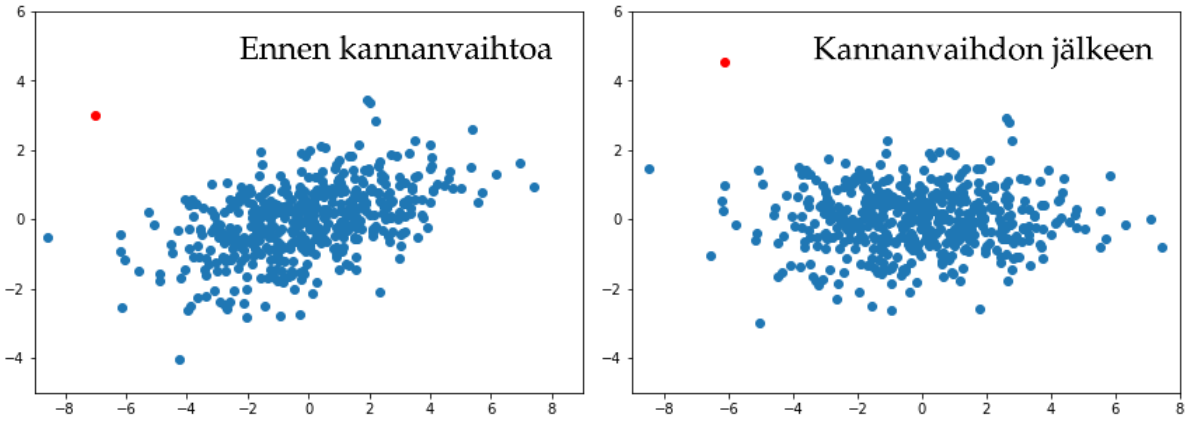
$$\left[\begin{array}{cc|c} 5-\lambda & 1 & 0 \\ 1 & 1-\lambda & 0 \end{array}\right] \Rightarrow \left[\begin{array}{cc|c} 0 & -(\lambda^2 - 6\lambda + 4) & 0 \\ 1 & 1-\lambda & 0 \end{array}\right] \Rightarrow \left[\begin{array}{cc|c} 1 & 1-\lambda & 0 \\ 0 & 0 & 0 \end{array}\right].$$

Ominaisvektorit ovat siis muotoa $[-(1-\lambda)x_2, x_2]^T = x_2[\lambda - 1, 1]^T$. Koska niiden vaaditaan olevan yksikkövektoreita, saadaan, että

$$w_1 = \frac{1}{\sqrt{(2+\sqrt{5})^2 + 1^2}}[2+\sqrt{5}, 1]^T = \left[\frac{2+\sqrt{5}}{\sqrt{10+4\sqrt{5}}}, \frac{1}{\sqrt{10+4\sqrt{5}}}\right]^T$$
$$w_2 = \frac{1}{\sqrt{(2-\sqrt{5})^2 + 1^2}}[2-\sqrt{5}, 1]^T = \left[\frac{2-\sqrt{5}}{\sqrt{10-4\sqrt{5}}}, \frac{1}{\sqrt{10-4\sqrt{5}}}\right]^T.$$

5.2.1 Pääkomponenttianalyysin geometrinen tulkinta

Koska ominaisvektoreiden $\{w_1, w_2, ..., w_n\}$ k-variaatiot muodostavat k-ulotteisen avaruuden ortonormaalin kannan, kun $k > 0$, saadaan yksittäisen datapisteen x_j projektion koordinaattivektori vektoreiden $w_{i1}, w_{i2}, ..., w_{ik}$ virittämässä avaruudessa matriisitulolla $W^T x_j = [w_{i1}, w_{i2}, ..., w_{ik}]^T x_j = [x_j^T w_{i1}, x_j^T w_{i2}, ..., x_j^T w_{ik}]^T$. Jos uudet koordinaatit halutaan selvittää koko satunnaisvektorin jakauman generoineelle aineistolle yksittäisen datapisteen sijaan, saadaan datamatriisin M rivien koordinaatit uudessa kannassa tulona $MW = [x_1, x_2, ..., x_m]^T[w_{i1}, w_{i2}, ..., w_{ik}]$ datamatriisimuodossa eli siten, että sarakevektorit ovat muuttujia ja rivivektorit havaintoyksiköitä.



Kuva 5.2: Kannanvaihto aineistolle, jossa on poikkeama (-7,3).

Kuvassa 5.2 on esitetty, miten Mahalanobiksen etäisyyden yhteydessä esitellyn poikkeaman sisältäneen aineiston M parvikuvio muuttuu, kun sille tehdään edellä kuvatulla tavalla kannanvaihto $MW = M[w_1, w_2]$ aiemmin ratkaistujen ominaisvektoreiden w_1 ja w_2 avulla. Kuvan perusteella uusi parvikuvio vaikuttaa olevan saatu alkuperäisestä kuviosta kiertämällä siten, että suurin varianssi on vaaka-

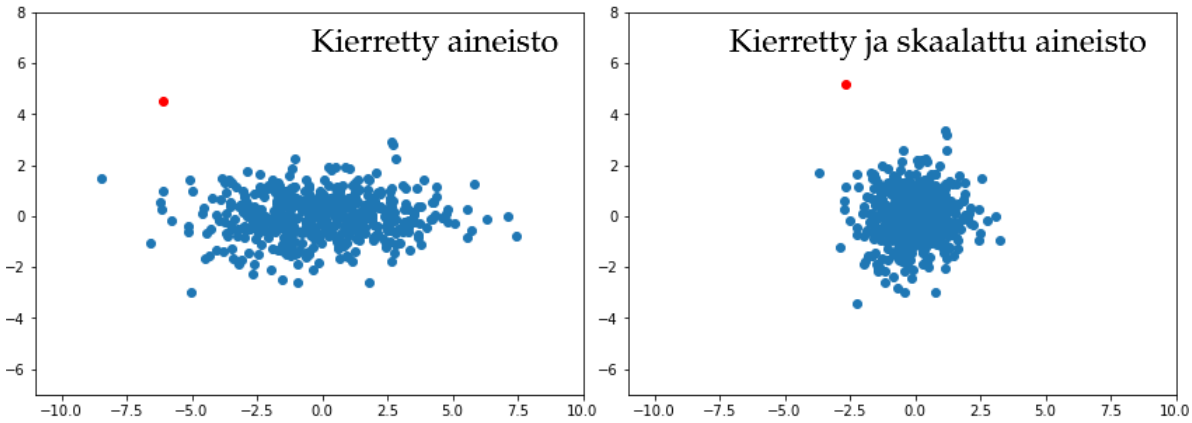
akselin suuntaan, ja näin myös on, sillä $W^T X$ on ortogonaalinen muunnos ja vastaa siten akselien kiertoa tai peilausta [68]. Pientä eroa kierrossa tosin voi olla, sillä otoskovarianssimatriisi ei täysin vastaa satunnaisvektorin X kovarianssimatriisia.

Ortogonaaliseksi kuvauksen $W^T X$ tekee se, että sitä vastaava matriisi W^T on ortogonaalinen [33]. Tämä on ilmeistä, sillä matriisin $W^T W = [w_i^T w_j]$ alkiot ovat ortonormaalien vektoreiden pistetuloja. Ortogonaaliset muunnokset Qx vastaavat geometrisesti akseleiden kiertoa tai peilausta, sillä ne säilyttävät vektoreiden pistetulon $(Q^T x_i)^T (Q^T x_j) = x_i^T Q Q^T x_j = x_i^T Q Q^{-1} x_j = x_i^T x_j$ kaikilla x_i, x_j ja siten myös niiden pituudet ja niiden väliset kulmat [18]. Kuvan 5.2 kannanvaihdossa käytetty matriisi

$$[w_1, w_2]^T = \begin{bmatrix} cos\,\theta & sin\,\theta \\ -sin\,\theta & cos\,\theta \end{bmatrix} = \begin{bmatrix} \frac{2+\sqrt{5}}{\sqrt{10+4\sqrt{5}}} & \frac{1}{\sqrt{10+4\sqrt{5}}} \\ \frac{2-\sqrt{5}}{\sqrt{10-4\sqrt{5}}} & \frac{1}{\sqrt{10-4\sqrt{5}}} \end{bmatrix}$$

on siis kiertomatriisi ja kiertää parvikuviota origon suhteen myötäpäivään noin 13°.

Kuvista 5.2 ja 5.3 nähdään, että akselien kierron jälkeen poikkeama erottuu muista datapisteistä selvemmin saamalla muita datapisteitä suurempia arvoja toisen pääkomponenttinsa osalta. Poikkeamia voidaankin etsiä jokaisen pääkomponentin osalta erikseen [69] esimerkiksi tutkimalla, kuinka monen keskihajonnan päässä odotusarvosta niiden saamat arvot ovat. Kuvassa 5.3 on esitetty, miten edellä esitetty parvikuvio muuttuu, kun satunnaisvektorin $W^T X = [w_1, w_2]^T X$ jakauman generoimien vektoreiden komponentit jaetaan niiden keskihajonnoilla $\sqrt{\mathrm{Var}(w_i^T X)} = \sqrt{\lambda_i}$.



Kuva 5.3: Kierretyn aineiston skaalaus.

Kun $D^{-\frac{1}{2}}$ on diagonaalimatriisi, jonka päälävistäjän arvot ovat $\frac{1}{\sqrt{\lambda_1}}, \frac{1}{\sqrt{\lambda_2}}, ..., \frac{1}{\sqrt{\lambda_n}}$, saadaan keskitetyistä pääkomponenteista $[w_1^T(X-\mu), w_2^T(X-\mu), ..., w_n^T(X-\mu)]^T$

muodostuvasta satunnaisvektorista yleisestikin tällä tavalla skaalattu kuvauksella $D^{-\frac{1}{2}}W^T(X-\mu)$, kunhan satunnaisvektorin X kovarianssimatriisin C ominaisarvot ovat aidosti positiivisia. Alkuperäisen satunnaisvektorin X jakauman generoimien datapisteiden x kuvien, jotka näkyvät kuvan 5.3 oikeanpuoleisessa parvikuviossa, neliöidyt etäisyydet maaliavaruuden origoon saadaan siten pistetulolla

$$\begin{aligned}(D^{-\frac{1}{2}}W^T(x-\mu))^T(D^{-\frac{1}{2}}W^T(x-\mu)) &= (x-\mu)^TWD^{-\frac{1}{2}}D^{-\frac{1}{2}}W^T(x-\mu)\\ &= (x-\mu)^TWD^{-1}W^T(x-\mu).\end{aligned}$$

Koska ominaisarvoyhtälöstä $Cw_i = \lambda_i w_i$ saadaan C:n ominaisarvo- ja spektraalihajotelmaksikin kutsuttu yhtälö $CW = WD \Leftrightarrow C = WDW^{-1} = WDW^T$ [69, 114], on selvää, että $C^{-1} = WD^{-1}W^T$, kun C on kääntyvä, sillä $WDW^TWD^{-1}W^T = I$. Edellä esitetty kuvaus $(x-\mu)^TWD^{-1}W^T(x-\mu) = (x-\mu)^TC^{-1}(x-\mu)$ on siis neliöity Mahalanobiksen etäisyys jakauman odotusarvoon, ja $C^{-\frac{1}{2}} = WD^{-\frac{1}{2}}W^T$ matriisi, jota voidaan käyttää ZCA-valkaisuun [71, 106]. Koska ZCA-valkaisussa matriisi W vain palauttaa satunnaisvektorin alkuperäiseen kantaan, voidaan myös pääkomponenttianalyysin avulla saatua matriisia $D^{-\frac{1}{2}}W^T$ käyttää valkaisuun [71]. Kuvassa 5.3 esitetty kierto ja skaalaus vastaavatkin satunnaisvektorin X valkaisua.

5.2.2 Aineiston standardointi ja anomaliapisteytys

Usein PCA:ssa käytetyt muuttujat standardoidaan myös jo ennen varsinaisten pääkomponenttien selvittämistä, jotta muuttujien eri mitta-asteikot eivät pääsisi vaikuttamaan suurimman vaihtelun suuntiin ja siten pääkomponenttianalyysin avulla saatuihin uusiin muuttujiin. Tyypillisesti aineiston generoinutta jakaumaakaan ei tunneta, ja pääkomponenttien etsimiseen käytetään otoskovarianssimatriisia, joka standardoitujen muuttujien osalta on siis sama kuin otoskorrelaatiomatriisi. Kun $Z_{m\times n}$ on matriisi, jonka rivivektorit z_i ovat havaintoyksiköitä ja jonka sarakevektorit ovat tällaisesta standardoidusta aineistoista X saatuja pääkomponentteja, on aineiston X datapisteiden x_i Mahalanobiksen etäisyyden neliö aineiston keskiarvoon

$$\sum_{k=1}^{p}\frac{z_{ik}^2}{\lambda_k} + \sum_{k=n-q+1}^{n}\frac{z_{ik}^2}{\lambda_k}$$

edellä käytetyin merkinnöin edellyttäen, että $n = p+q$, sillä $z_i = W^Tx_i$. [69]

Koska ensimmäiset p pääkomponenttia sisältävät eniten ja viimeiset q pääkomponenttia vähiten alkuperäisen aineiston varianssista, voidaan niiden avulla löytää

erilaisia poikkeamia [69]. Ensimmäisten pääkomponenttien avulla löydetyt poikkeamat ovat ääriarvoja aineiston normaalin vaihtelun suuntaan ja saattavat siten olla löydettävissä jo alkuperäisten muuttujien avulla. Viimeisten pääkomponenttien avulla löydettävät poikkeamat ovat puolestaan sellaisia, jotka rikkovat alkuperäisen aineiston korrelaatiorakennetta. Riippumatta siitä, mitkä p ja q valitaan, noudattaa yllä esitetty summa likimain χ^2 jakaumaa vapausasteella $p + q$, olettaen, että aineisto, josta pääkomponentit on muodostettu, on multinormaalijakautunut [68].

Esimerkiksi tilastollisessa prosessinohjauksessa järjestelmän käytön aikana suoritettujen yksittäisten mittausten poikkeavuutta voidaan arvioida yllä esitetyn summan avulla hyödyntämällä järjestelmän normaalin toiminnan aikana suoritetuista mittauksista saatua keskiarvovektoria ja otoskovarianssimatriisia [83]. Tällöin summa, johon otetaan tyypillisesti vain p ensimmäistä termiä, on T^2-testisuure, jonka suurella otoskoolla voidaan olettaa noudattavan χ^2-jakaumaa [68, 83]. Summa (Squared Prediction Error, SPE) voidaan myös laskea niin, että summattavat termit, tavallisesti q viimeisintä, jätetään jakamatta niitä vastaavilla ominaisarvoilla, jolloin vältytään tilanteelta, jossa jakajaksi saadaan hyvin pieni ominaisarvo [68, 83].

Pääkomponenttianalyysin avulla verkkoliikenteestä tapahtuvassa poikkeaman tunnistuksessa ollaan hyödynnetty vastaavia periaatteita. Esimerkiksi Lakhina et al. käyttivät sekä ensimmäisen p:n pääkomponentin avulla laskettua T^2-testisuuretta että viimeisen q:n pääkomponentin avulla laskettua SPE:tä anomaliapisteytyksissään [77, 78]. Myös Shyu et al. [126] laskivat erikseen summan ensimmäisen p:n ja viimeisen q:n termin osalta, vaikka jakoivatkin myös viimeiset termit niitä vastaavilla ominaisarvoilla. χ^2-jakauman sijaan he käyttivät opetusaineiston summien empiirisiä jakaumia kynnysarvon asettamiseen.

5.3 Kvantitatiivisten ja kvalitatiivisten muuttujien faktorianalyysi

Kvantitatiivisten ja kvalitatiivisten muuttujien faktorianalyysi on pääkomponenttianalyysiä hyödyntävä menetelmä, jossa määrällisten muuttujien lisäksi myös one-hot-koodauksella binarisoidut laadulliset muuttujat osallistuvat aineiston pääkomponenttien muodostukseen. Jotta määrälliset muuttujat sekä binarisoidut laadulliset muuttujat huomioitaisiin oikeassa suhteessa, esikäsitellään ne eri tavalla. Käytännössä FAMD voidaan suorittaa keskityksen jälkeen tavallisella PCA:lla aineistolle, jossa määrälliset muuttujat on standardoitu ja jossa binarisoidut muuttujat on jaettu niitä vastaavien muuttujien arvojen suhteellisten frekvenssien neliöjuurilla. [105]

Kun FAMD tehdään aineistolle, jossa on vain jatkuvia muuttujia, vastaa se standardoidulle aineistolle tehtyä pääkomponenttianalyysiä. Koska tällaisen aineiston kovarianssimatriisin päälävistäjältä löytyvät varianssit ovat yksikkövariansseja, on niiden summa eli aineiston kokonaisvarianssi yhtä suuri kuin aineiston muuttujien lukumäärä. Lisäksi voidaan osoittaa, että standardoiduista muuttujista koostuvan n-ulotteisen satunnaisvektorin Z kovarianssimatriisin q ensimmäistä ominaisvektoria $W = [w_1, w_2, ..., w_q]$ määräävät sellaisen ortonormaalin kuvauksen $Y = W^T Z$, jonka avulla saadun satunnaisvektorin Z:n komponenttien suhteen laskettujen yhteiskorrelaatiokertoimien neliöiden summa on suurin mahdollinen [69]. Koska pääkomponentit eivät korreloi keskenään, maksivoivat ne

$$\sum_{i=1}^{n} R^2_{Z_i.Y_1,Y_2,...,Y_q} = \sum_{i=1}^{n} \sum_{j=1}^{q} \mathrm{Corr}(Z_i, w_j^T Z)^2.$$

Kun one-hot-koodauksella saatu binaarimuuttuja X jaetaan siihen liittyvän pistetodennäköisyyden $p = P(X = 1)$ neliöjuurella, saadaan uusi muuttuja $Y = p^{-\frac{1}{2}} X$, jonka varianssi on $\mathbb{E}(Y^2) - \mathbb{E}(Y)^2 = (p^{-\frac{1}{2}})^2 p - (p^{-\frac{1}{2}} p)^2 = 1 - p$. Tällöin k eri arvoa saavan laadullisen muuttujan kokonaisvarianssi on siihen liittyvien binäärimuuttujien varianssien summa eli $k - 1$. Laadullisten muuttujien saamien eri arvojen määrä vaikuttaakin FAMD:ssä suuresti aineiston sisältämän kokonaisvarianssin määrään, joka on määrällisten muuttujien K ja binaarimuuttujien lukumäärien summa, josta on vähennetty laadullisten muuttujien Q lukumäärä [105]. Binaarimuuttujista saadut uudet muuttujat Y_i keskitetään vielä ennen PCA:ta. FAMD:ssä [104] pyritään siis löytämään pääkomponentit $w_i^T Z$, jotka jokainen osaltaan maksimoivat

$$\sum_{k \in K} \mathrm{Corr}(k, w_i^T Z)^2 + \sum_{q \in Q} \eta^2(q, w_i^T Z).$$

Koska binaarimuuttujat jaetaan osana esikäsittelyä niitä vastaavien muuttujien arvojen suhteellisten frekvenssien neliöjuurilla, voi jokainen näin saatu ja keskitetty muuttuja $(X - p)p^{-\frac{1}{2}}$ saada sitä suurempia arvoja mitä harvinaisempaa alkuperäisen muuttujan arvoa se vastaa. Koska jokaiseen laadulliseen muuttujaan $q \in Q$ liittyvien binaarimuuttujien arvot summautuvat aina vakioon yksi, ei esikäsitellyn aineiston kovarianssimatriisi ole tyypillisesti täyttä astetta [68]. Ainakin viimeisen $|Q|$:n pääkomponentin varianssi on nolla [105]. Luonnollisestikaan esikäsitellyistä määrällisistä ja one-hot-koodauksella saaduista binaarimuuttujista koostuva aineisto ei noudata multinormaalijakaumaa.

On kuitenkin hyvä huomata, että FAMD:ssä sekä one-hot-koodauksella saaduista binaarimuuttujista B että määrällisistä muuttujista K muodostetulle satunnaisvektorille $X = [K_1, K_2, ..., K_c, B_1, B2, ..., B_d]^T$ suoritettu esikäsittely on itse asiassa affiini muunnos. Tämä on selvää, sillä kun A on diagonaalimatriisi, jonka päälävistäjän arvot ovat $\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, ..., \frac{1}{\sigma_c}, \frac{1}{\sqrt{p_1}}, \frac{1}{\sqrt{p_2}}, ..., \frac{1}{\sqrt{p_d}}$, on $A(X - \mu) = AX - A\mu$ vaadittu esikäsittely, olettaen, että p_i ja σ_i ovat nollasta poikkeavia suhteellisia frekvenssejä ja keskihajontoja [105]. Käytännössä FAMD tulisi siis suorittaa vain sellaiselle aineistolle, josta on jo etukäteen poistettu kaikki vakiomuuttujat.

Koska PCA, kuten luvussa 5.2.1 todettiin, vastaa mahdollisesti esikäsitellyn aineiston kovarianssimatriisin tai sen estimaatin ominaisvektoreista muodostetun matriisin W^T avulla suoritettua aineiston kiertoa, nähdään, että FAMD on kokonaisuudessaankin affiini muunnos $Z = W^T(A(X - \mu)) = W^T AX - W^T A\mu$. Käytännössä tämä tarkoittaa sitä, että kun C on n-ulotteisen muunnoksen kääntyvä kovarianssimatriisi ja kun z_i on alkuperäistä datapistettä x_i vastaava FAMD:n avulla saatu piste ja kun λ_k on pääkomponentin Z_k varianssi, niin

$$Mahalanobis(x_i, \mu)^2 = z_i^T C^{-1} z_i = \sum_{k=1}^{n} \frac{z_{ik}^2}{\lambda_k},$$

koska Z on keskitetty ja koska C on variansseista λ_k koostuva diagonaalimatriisi.

Vaikka satunnaisvektoreiden X ja Z kovarianssimatriisit eivät tyypillisesti olekaan one-hot-koodauksen seurauksena kääntyviä, voidaan Mahalanobiksen etäisyyden neliö jakauman odotusarvoon silti tyypillisesti laskea n-ulotteisen avaruuden sijaan $(n - |Q|)$-ulotteisessa avaruudessa huomioimalla vain ne suunnat joihin kannanvaihdon jälkeen on vaihtelua summalla

$$z_i^T C^- z_i = \sum_{k=1}^{n-|Q|} \frac{z_{ik}^2}{\lambda_k}, \tag{5.2}$$

jossa C^-, jolle $CC^-C = C$, on yleistetty käänteismatriisi [115]. Koska periaatteessa samaan tulokseen päästään myös, kun ennen PCA:ta jokaisen esikäsitellyn laadullisen muuttujan jotain arvoa vastaava binaarimuuttuja poistetaan satunnaisvektorista [17, 115], ei faktorianalyysiin tarkoitettu FAMD ole välttämättä poikkeaman tunnistuksen kannalta kovin mielenkiintoinen, jos sen avulla saatujen pääkomponenttien anomaliapisteytykseen käytetään kaavaa 5.2. On kuitenkin hyvä huomata, että muuttujien esikäsittely ei tyypillisesti ole ortogonaalinen muunnos, ja vaikuttaa siten yleensä PCA:sta saataviin pääkomponentteihin [69].

5.4 FAMDAD-menetelmä

FAMDAD on Davidowin ja Mattesonin [35] esittämä kvantitatiivisten ja kvalitatiivisten muuttujien faktorianalyysiä hyödyntävä ohjaamaton poikkeaman tunnistusmenetelmä, jossa standardoituja määrällisiä muuttujia painotetaan niiden huipukkuuteen perustuvilla kertoimilla. Vaikka FAMDAD periaatteessa vain kuvaakin alkuperäisen aineiston alempiulotteiseen avaruuteen, jossa poikkeamien on tarkoitus erottua selvemmin, voidaan siihen yhdistää myös muita menetelmiä. Davidow ja Matteson käyttivätkin FAMDAD-menetelmällä käsittelemänsä aineiston anomaliapisteytykseen sekä histogrammeihin perustuvaa (Simple Probabilistic Anomaly Detector, SPAD) menetelmää että Isolation Forest -menetelmää.

Koska FAMDAD perustuu kvantitatiivisten ja kvalitatiivisten muuttujien faktorianalyysiin, voidaan sekin suorittaa tavallisen pääkomponenttianalyysin avulla. Ainoa muutos, joka muuttujien esikäsittelyyn tarvitaan FAMD-menetelmään verrattuna, on, että ennen pääkomponenttianalyysin suorittamista jokainen standardoitu määrällinen muuttuja Z täytyy vielä kertoa sen huipukkuuteen $\kappa = \mathbb{E}(Z^4)$ tai tarkemmmin otoshuipukkuuteen b_2 perustuvalla kertoimella $(\frac{b_2}{3})^{\frac{1}{2}}$ [35, 144]. Koska normaalijakauman huipukkuus on 3, korostaa menettely normaalijakaumaa hännäkkäämmin jakautuneiden muuttujien varianssia, ja samalla vähentää niiden muuttujien varianssia, joilla otoshuipukkuus on pienempi kuin kolme [35].

Käytännössä FAMDAD-menetelmä eroaa siis standardoidulle aineistolle, johon laadulliset muuttujat on otettu mukaan ennen standardointia one-hot-koodattuina binaarimuuttujina, tehdystä pääkomponenttianalyysistä vain PCA:ssa käytettyjen muuttujien painokerrointen osalta. Sekä FAMD että FAMDAD voidaankin nähdä eräänlaisina yleistettyinä pääkomponenttianalyyseinä, joissa sekä havainnoille että muuttujille voidaan määrittää painot ja metriikat [69]. FAMDAD-menetelmässä jokaiselle havaintoyksikölle annetaan tyypillisesti sama paino. Esikäsitellyn aineiston datapisteiden z_i ja z_j euklidisesta etäisyydestä saadaan puolestaan metriikka

$$\begin{aligned} d^2(z_i, z_j) = (z_i - z_j)^T (z_i - z_j) &= \sum_{k=1}^{n} (z_{ik} - z_{jk})^2 \\ &= \sum_{k=1}^{c} \frac{b_{2_k}(x_{ik} - x_{jk})^2}{3s_k^2} + \sum_{k=c+1}^{n} \frac{(x_{ik} - x_{jk})^2}{p_k}, \end{aligned}$$

jossa p_k on binarisoidun muuttujan suhteellinen frekvenssi ja jossa b_{2_k} ja s_k^2 ovat määrällisen muuttujan otoshuipukkuus ja otosvarianssi, kun one-hot-koodauksella saatujen alkuperäisten datapisteiden x_i ja x_j esikäsittely huomioidaan [35, 105].

Koska FAMDAD menetelmän on tarkoitus vähentää alkuperäisen aineiston ulottuvuuksien määrää, pitää menetelmässä valita PCA:n avulla saaduista pääkomponenteista jokin osajoukko [35]. Davidow ja Matteson esittivät, että optimaalinen säilytettävien pääkomponenttien määrä riippuu alkuperäisen aineiston muuttujien määrästä. Menetelmänsä varsinaiseen arviointiin he käyttivät sekä viittä ensimmäistä että yhteensä viittä ensimmäistä ja viimeistä varianssia vielä mahdollisesti sisältänyttä pääkomponenttia. Lisäksi he päätyivät asettamaan jatkuville muuttujille käytetyn painokertoimen $(\frac{b_2}{3})^{\frac{1}{2}}$ maksimiarvoksi $(\frac{10}{3})^{\frac{1}{2}}$.

Anomaliapisteytykseen käytetty SPAD on parametriton tilastollinen poikkeaman tunnistusmenetelmä, jossa diskreettien tai diskretoitujen muuttujien X_k anomaliapisteytys perustuu niiden pistetodennäköisyyteen $P(X_k = x_{ik})$. Koska menetelmässä muuttujien oletetaan olevan riippumattomia, saadaan yksittäisen datapisteen x_i todennäköisyys tulona $\prod_{k=1}^{n} P(X_k = x_{ik})$, johon SPAD-menetelmän anomaliapisteytys perustuu. Käytännössä todennäköisyyden sijaan käytetään sen logaritmia, jossa pistetodennäköisyyksien estimoinnissa hyödynnetään Laplace-tasoitusta ja aineistoa, jossa on N datapistettä [13]. Tämä voidaan esittää yhtälönä

$$\sum_{k=1}^{n} log\, \hat{P}(X_k = x_{ik}) = \sum_{k=1}^{n} log\left(\frac{f_{x_{ik}} + 1}{N + |A_{X_k}|}\right),$$

jossa $f_{x_{ik}}$ on arvon x_{xi} frekvenssi ja jossa A_{X_k} on muuttujan X_k arvojoukko [13]. Koska osana FAMDAD-menetelmää suoritettu PCA tekee muuttujista korreloimattomia ja multinormaalijakauman osalta myös riippumattomia, on selvää, että SPAD:n riippumattomuusoletus voi saada vahvistusta tämän tyyppisestä esikäsittelystä.

Isolation Forest -menetelmä [81] puolestaan perustuu siihen, että poikkeavat havaintoyksiköt on normaaleja helpompi eristää aineiston muista datapisteistä niiden komponenttien arvojen perusteella. Opetusvaiheessa menetelmä rakentaa t:n satunnaisotoksen perusteella opetusaineistosta t eri sisäsolmuista ja lehdistä koostuvaa puuta. Sisäsolmuihin liittyy aina kaksi lasta T_l, T_r sekä satunnaisesti valittu muuttuja q ja sen arvoväliltä satunnaisesti valittu arvo p. Ehto $q < p$ jakaa sisäsolmuun liittyvät datapisteet osajoukkoihin, jotka liittyvät lapsisolmuihin T_l ja T_r. Lapsisolmu on lehti, jos siihen liittyvän datapistejoukon koko on yksi tai jos puun maksimisyvyys on saavutettu. Muussa tapauksessa myös se on sisäsolmu.

Isolation Forest -menetelmän anomaliapisteytys perustuu siihen, kuinka monta sisäsolmuihin liittyvää vertailua pisteytettävälle datapisteelle x on tehtävä, ennen kuin sitä vastaava lehtisolmu löydetään. Käytännössä pisteytyksessä käytetään eri metsän puiden polun pituuksien $h_i(x)$ keskiarvoa $h_{avg}(x) = \frac{1}{t}\sum_{i=1}^{t} h_i(x)$. Koska osa

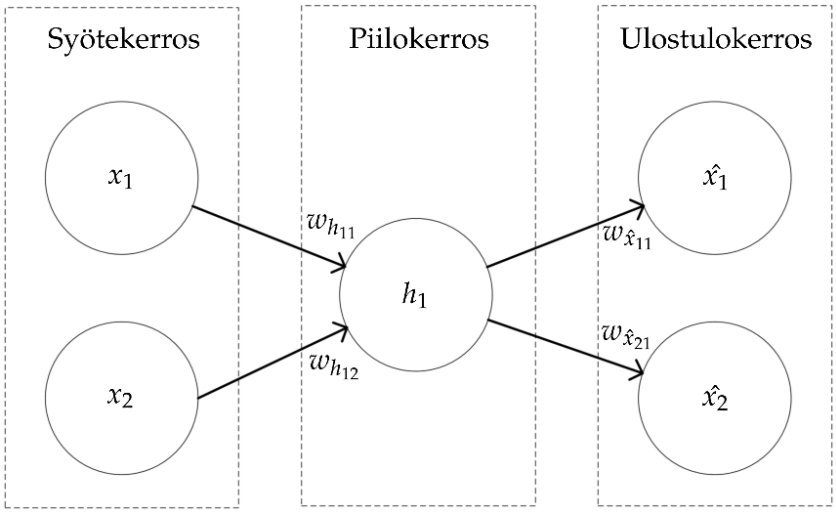
lehdistä voi syntyä polun maksimipituuden ylityttyä, korjataan niihin liittyvien polkujen pituutta lisäämällä niihin $c(k)$, joka vastaa epäonnistuneen binäärihaun keskimääräisen polun pituutta k:n opetusvaiheessa lehtisolmuun kuuluneen datapisteen muodostamassa binääripuussa. Varsinainen anomaliapisteytys saadaan kaavalla

$$s(x, n) = 2^{-\frac{h_{avg}(x)}{c(n)}},$$

jossa n on opetuksessa käytetty satunnaisotoksen koko [81]. Koska Isolation Forest -menetelmä on suunniteltu jatkuville muuttujille, on helppoa nähdä, että FAMDAD-menetelmän tuottamat pääkomponentit voivat soveltua sille paremmin kuin esimerkiksi one-hot-koodauksella binarisoidut muuttujat.

5.5 Autoenkooderit ja neuroverkot

Autoenkooderit ovat neuroverkkoja, jotka on opetettu muodostamaan mahdollisimman tarkka kopio syötteestään jonkin opetusaineiston avulla [54]. Tässä työssä keskitytään vain sellaisiin autoenkoodereihin, joissa neuroverkon syöte- ja ulostulokerroksen välissä on yksi tai useampi piilokerros, joilla on vähemmän neuroneita kuin ulommilla kerroksilla. Koska jokaisen kerroksen neuronien määrä vastaa sen ulottuvuuksien määrä, joutuu tällainen neuroverkko oppimaan syötevektoreiden alempiulotteisen esityksen [133]. Kuvassa 5.4 on esitetty yksinkertaisen autoenkooderin arkkitehtuuri, jossa on syöte- ja ulostulokerroksen lisäksi yksi piilokerros.



Kuva 5.4: Yksinkertainen eteenpäinsyöttävä neuroverkko.

Syötekerroksen neuronit poikkeavat muiden kerrosten neuroneista siten, että ne ovat suoraan neuroverkon syötevektoreiden komponentteja [133]. Muiden kerrosten neuronit saavat eteenpäinsyöttävässä neuroverkossa syötevektorinsa, joiden komponentteja kuvassa 5.4 esittävät niihin piirretyt nuolet, niitä edeltäneen kerroksen neuroneilta [28]. Tällaisen syötevektorin x sekä neuroniin itseensä liittyvän vakiotermin b ja aktivointifunktion f_a avulla saadaan vektorilla w painotetun summan funktio $f_a(w^T x + b)$, joka neuronin kerroksen tyypistä riippuen voi toimia joko muiden neuronien syötteenä tai neuroverkon ulostulovektorin komponenttina [133].

Koska kuvassa 5.4 vektori $x = [x_1, x_2]^T$ toimii piilokerroksen ainoan neuronin syötevektorina, voidaan piilokerros esittää vektorina $h = [f_{a_{h_1}}(w_{h_1}^T x + b_{h_1})]$, joka toimii sitä seuraavan kerroksen neuronien syötteenä. Myös ulostulokerros voidaan esittää vektorina $[f_{a_{\hat{x}_1}}(w_{\hat{x}_1}^T h + b_{\hat{x}_1}), f_{a_{\hat{x}_2}}(w_{\hat{x}_2}^T h + b_{\hat{x}_2})]^T$, jossa on yhtä monta komponenttia kuin kerroksella on neuroneita. Selvästi tällaisen neuroverkon minkä tahansa kerroksen ulostulovektori on helppo laskea, kunhan neuroneiden vakiotermit, painot ja aktivointifunktiot tunnetaan. Koska neuroverkon jokainen neuroni on yhdistetty kaikkiin sitä seuraavan kerroksen neuroneihin, on se täysin kytketty [50].

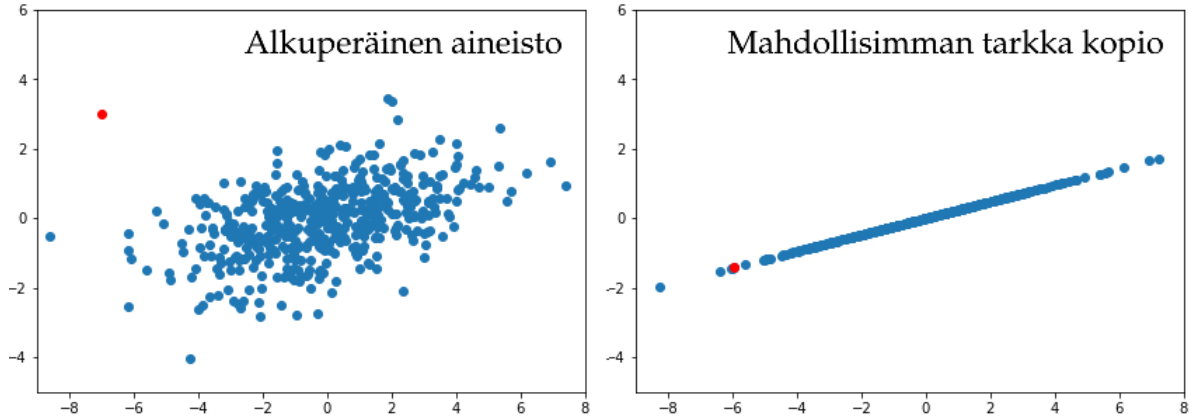
Eteenpäinsyöttävän täysin kytketyn neuroverkon lisäksi muita neuroverkkotyyppejä ovat muun muassa takaisinkytketty neuroverkko, joka hyödyntää syötteenään myös aiemmin käsittelemiensä syötevektoreiden päivittämää sisäistä tilaansa, sekä konvoluutioneuroverkko, joka hyödyntää konvoluutiota ainakin jollain kerroksistaan [54, 133]. Takaisinkytketyt neuroverkot soveltuvat aineistoihin, joissa datapisteiden keskinäisellä järjestyksellä on merkitystä [54]. Konvoluutiota, joka kaksiulotteisen syötteen X ja ytimen K tapauksessa voidaan esittää summana

$$S(i,j) = \sum_m \sum_n [X]_{(i-m)(j-n)} [K]_{mn},$$

käyttävät neuroverkot pystyvät taas hyödyntämään tehokkaasti esimerkiksi kuvissa olevien pikseleiden sijainnin suhteessa naapuripikseleihinsä yllä esitetyn summan avulla [54, 142]. Tässä työssä keskitytään kuitenkin vain eteenpäinsyöttäviin täysin kytkettyihin neuroverkkohin, joihin perustuvia autoenkoodereita voidaan käyttää muun muassa pisteanomalioiden tunnistamiseen verkkoliikenteestä [26, 58].

Koska kuvan 5.4 autoenkooderin syöte- ja ulostulokerroksella on kaksi neuronia ja koska sen piilokerroksella on vain yksi neuroni, nähdään, että sitä voitaisiin käyttää oppimaan luvussa 5.1 esitellyn kaksiulotteista normaalijakaumaa noudattavan aineiston X yksiulotteinen esitys piilokerroksen ulostulossaan. Kuvassa 5.5 onkin esitetty parvikuviona tämän aineiston lisäksi kaikki yllä kuvatun autoenkooderin,

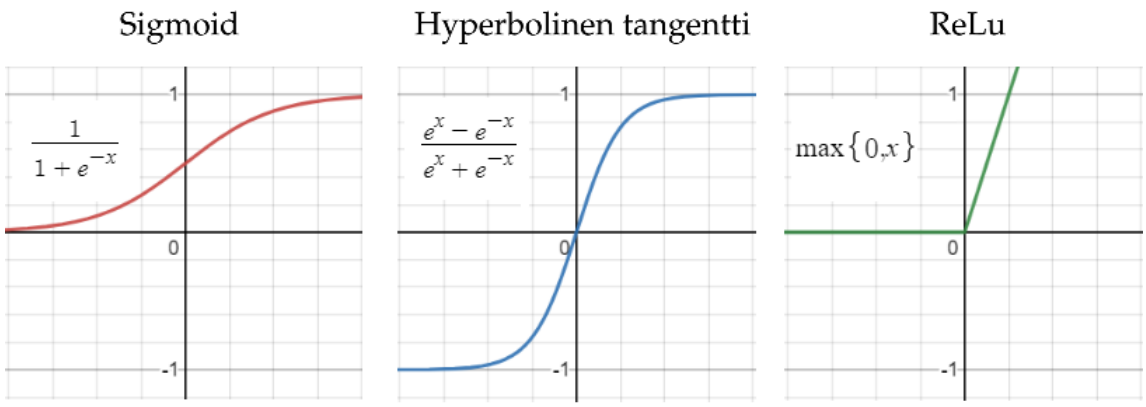
jonka aktivointifunktioksi on asetettu identiteettifunktio, X:stä ilman vakiotermejä opituilla painoilla datapisteistä $x \in X$ muodostamat ulostulovektorit $\hat{x} = [\hat{x_1}, \hat{x_2}]^T$.



Kuva 5.5: Alkuperäinen aineisto ja siitä autoenkooderin muodostama kopio.

Kuvasta 5.5 nähdään, että autoenkooderin muodostaman kopion pisteet ovat likimain alkuperäisten pisteiden projektiot parhaalle mahdolliselle suoralle, joka parvikuvioon voidaan sovittaa pienintä keskineliövirhettä $\frac{1}{|X|}\sum_{x \in X} ||x - \hat{x}||^2$ tavoiteltaessa. Koska autoenkooderissa ei käytetty vakiotermejä ja koska jokaisen neuronin aktivointifunktio on identiteettifunktio, saadaan piilokerroksen neuronin ulostuloksi $w^T x$, jossa painovektori on aineistosta opittu $w \approx [0,963; 0,229]^T$. Vaikka w sattuukin olemaan lähellä aineistosta PCA:n avulla saatavaa ensimmäistä pääkomponenttia vastaavaa ominaisvektoria $w_{pc1} \approx [0,973; 0,232]^T$, ei opetuksessa käytetty gradienttimenetelmä tätä kuitenkaan takaa [16]. Aineistosta lasketun keskineliövirheen minimoimiseksi riittäisi, että piilokerrokselta saataisiin $w^T x = (w_{pc1} c^T)^T x$ ja että ulostulokerrokselta saataisiin $w_{pc1} c^{-1} (w_{pc1} c^T)^T x = w_{pc1} w_{pc1}^T x$ [16, 68].

Koska keskitetystä aineistosta X pääkomponenttianalyysin avulla saatua k:ta ensimmäistä pääkomponenttia vastaavat ominaisvektorit $W = [w_1, w_2, ..., w_k]$ määräävät yleisestikin vain identiteettifunktiota aktivointifunktionaan käyttävän autoenkooderin, jonka piilokerroksella on k neuronia, pienimmän keskineliövirheen tuottaman aineiston kopion $\hat{X} = XWW^T$, eivät tällaiset autoenkooderit ole välttämättä kovinkaan mielenkiintoisia [16, 68]. Tyypillisesti autoenkoodereissa käytetäänkin myös epälineaarisia aktivointifunktioita, joiden ansiosta ne voivat oppia aineiston alempiulotteisen esityksen, joka ei perustu vain muuttujien lineaarisiin riippuvuuksiin [133]. Kuvassa 5.6 on esitetty eräitä yleisiä epälineaarisia aktivointifunktioita.



Kuva 5.6: Eräitä epälineaarisia aktivointifunktioita.

Kaikille näistä funktioista on yhteistä se, että ne ovat jatkuvia ja jatkuvasti derivoituvia, jos ReLu-funktion derivaattana käytetään jompaakumpaa sen toispuoleisista derivaatoista kohdassa nolla, jossa se ei ole derivoituva. Koska sigmoid ja hyperbolinen tangentti kärsivät häviävien gradienttien ongelmasta, joka aiheutuu sitä, että niiden derivaatat lähestyvät nollaa suurilla ja pienillä arvoilla, ei niitä suositella eteenpäinsyöttävien neuroverkkojen piilokerrosten aktivointifunktioiksi. Vaikka ReLu, jota pidetään hyvänä oletusvalinta piilokerroksille, ei kärsikään häviävien gradienttien ongelmista positiivisilla arvoilla, on sen derivaatta negatiivisilla arvoilla nolla. Jos näitä aktivointifunktioita käytetään ulostulokerroksella, eivät niiltä saatavien vektoreiden komponentit voi saada kaikkia reaalilukuarvoja. [54]

Teoreettisesta näkökulmasta on myös hyvä huomata, että jo yhden piilokerroksen sisältävä eteenpäinsyöttävä neuroverkko, joka käyttää jotain yllä mainituista aktivointifunktiosta piilokerroksellaan ja identiteettifunktiota ulostulokerroksellaan, voi universaalin approksimointilauseen mukaan approksimoida mitä tahansa mitallista kuvausta mielivaltaisen pienellä virheellä $\epsilon > 0$, kunhan sen piilokerroksen neuronien määrää ei rajoiteta [54]. Polynomeja aktivointifunktioinaan käyttäville eteenpäinsyöttäville neuroverkoille tätä ominaisuutta ei voida osoittaa [79].

5.5.1 Neuroverkon opettaminen

Sekä tavallisten että syvien eli useita piilokerroksia sisältävien eteenpäinsyöttävien neuroverkkojen opettamiseen käytetään tyypillisesti menetelmää, jossa verkon painot, vakiotermit sekä mahdolliset muut parametrit opitaan minimoimalla verkon

ulostulovektoreihin liittyvää kustannusfunktiota jonkin aineiston suhteen gradientteihin perustuvalla menetelmällä. Vaikka neuroverkot teoriassa pystyvätkin toimimaan universaaleina approksimaattoreina, joudutaan niitä opetettaessa tyypillisesti minimoimaan kustannusfunktiota, joka ei ole konveksi. Teoreettisesti ei siis voida osoittaa, että vaaditun arkkitehtuurin omaavat neuroverkot myös oppisivat mielivaltaisen pienen virheen saavuttamiseksi tarvittavat parametrit. [54]

Vaikka funktion, joka ei ole konveksi, minimiä gradenttipohjaisella menetelmällä etsittäessä voidaankin päätyä lokaaliin minimiin, satulapisteeseen tai kohtaan, jossa gradientti ei edes ole nolla, on löydetty ratkaisu silti usein kohta, jossa kustannusfunktion arvo on riittävän alhainen. Vaikuttaakin siltä, että riittävän suurissa neuroverkoissa kustannusfunktion arvot lokaaleissa minimeissä ovat vain harvoin korkeita. Lisäksi ulottuvuuksien kasvaessa usean tyyppisten satunnaisfunktioiden kriittiset pisteet ovat lokaalien minimien sijaan yhä useammin satulapisteitä, joita gradienttipohjaiset menetelmät onneksi vain harvoin löytävät. [54]

Tavallisessa gradienttimenetelmässä pyritään siis minimoimaan kustannusfunktiota J opetukseen käytetyn n-ulotteisen aineiston X suhteen neuroverkon parametreja θ iteratiivisesti päivittämällä [54]. Esimerkiksi autoenkoodereille soveltuvaa keskineliövirhettä kustannusfunktiona käytettäessä gradienttimenetelmällä ratkaistava optimointiongelma voidaan esittää muodossa

$$\begin{aligned}\underset{\theta}{\operatorname{argmin}}\ J(\theta) &= \underset{\theta}{\operatorname{argmin}}\ \frac{1}{|X|}\sum_{x\in X}||x - f_\theta(x)||^2 \\ &= \underset{\theta}{\operatorname{argmin}}\ \frac{1}{2|X|}\sum_{x\in X}\sum_{k=1}^{n}(x_k - f_\theta(x)_k)^2,\end{aligned}$$

kun f_θ on funktio, joka palauttaa neuroverkon ulostulovektorin annetulla syötteellä x ja parametreilla θ. Gradienttimenetelmän iteratiivisuuden vuoksi $J(\theta)$ lasketaan kuitenkin tyypillisesti X:n satunnaisilla osajoukoilla koko aineiston sijaan [54].

Ennen kuin $J(\theta)$ ja $\nabla J(\theta)$ voidaan laskea ensimmäisen kerran, täytyy neuroverkon parametrit alustaa. Koska minimoitava kustannusfunktio ei tyypillisesti ole konveksi, voivat käytetyt arvot vaikuttaa gradienttimenetelmällä saatuihin tuloksiin. Tyypillisesti neuronien painovektorit alustetaan satunnaisluvuilla tasa- tai normaalijakaumasta, jolloin vältytään tilanteelta, jossa samoin alustettujen neuronien parametrit päivittyvät identtisesti. Muiden parametrien alustukseen käytetään tyypillisesti heuristisesti valittuja vakioita. Esimerkiksi ReLu-aktivointifunktioita käytettäessä vakiotermien asettaminen pieneen nollaa suurempaan arvoon, lisää todennäköisyyttä, että niiden saamat arvot ja derivaatat ovat aluksi positiivisia. [54]

Kun neuroverkon parametrit on alustettu, voidaan uudet parametrit laskea yksinkertaisesti päivittämällä niitä pieni askel η poispäin kustannusfunktion suurimmasta kasvusuunnasta $\theta \leftarrow \theta - \eta \nabla J(\theta)$. Jos askel η eli oppimisnopeus on liian pieni, sopivien parametrien löytämiseen tarvitaan turhan monta päivitystä. Toisaalta jos oppimisnopeus on liian suuri, ei menetelmä löydä sopivia parametreja. Vaikka oppimisnopeus onkin esitetty vakiona, voidaan sitä vaikka pienentää jokaisen iteraatiokerran jälkeen vähän kunnes se saavuttaa jonkin minimiarvon. [54]

Esimerkiksi kuvassa 5.5 esitetyn kopion muodostaneen autoenkooderin parametrit voidaan esittää vektorina $\theta = [w_{h_{11}}, w_{h_{12}}, w_{\hat{x}_{11}}, w_{\hat{x}_{21}}]^T$, joka koostuu pelkistä neuronien painovektoreiden komponenteista. Jos neuroverkko olisi käyttänyt vakiotermejä, olisivat myös ne parametreja. Koska summan derivaatta on derivaattojen summa, saadaan edellä esitetyn kustannusfunktion osittaisderivaatoiksi

$$\begin{aligned}\frac{\partial}{\partial \theta_i} J(\theta) &= \frac{1}{|X|} \sum_{x \in X} \sum_{k=1}^{n} (x_k - f_\theta(x)_k) \frac{\partial}{\partial \theta_i} (x_k - f_\theta(x)_k) \\ &= \frac{1}{|X|} \sum_{x \in X} \sum_{k=1}^{n} (f_\theta(x)_k - x_k) \frac{\partial}{\partial \theta_i} f_\theta(x)_k.\end{aligned}$$

Neuroverkoissa näiden gradientin komponenttien laskemiseen voidaan käyttää vastavirta-algoritmia, jossa neuroverkon ulostulovektorien laskemisen jälkeen eri kerrosten parametrien osittaisderivaatat lasketaan vastavirtaan eli ulostulokerrokselta syötekerrokselle päin [28]. Kun vakiotermi merkitään edelliseltä kerrokselta saatavan vektorin z komponentiksi $z_0 = 1$ ja kun sen arvoa vastaa paino $w_{\hat{x}_0}$, saadaan ulostulokerroksen neuronin k painojen ja vakiotermin osittaisderivaatoiksi

$$\frac{\partial}{\partial w_{\hat{x}_{kr}}} f_\theta(x)_k = f'_{a_{\hat{x}_k}} \left(\sum_{i=0}^{n} w_{\hat{x}_{ki}} z_i \right) \frac{\partial}{\partial w_{\hat{x}_{kr}}} \sum_{i=0}^{n} w_{\hat{x}_{ki}} z_i = f'_{a_{\hat{x}_k}} (w_{\hat{x}_k}^T z) z_r,$$

kun $f'_{a_{\hat{x}_k}}$ on neuronin käyttämän aktivointifunktion derivaatta. Kun viimeiselle piilokerrokselle tulevaan vektoriin x sisällytetään vakiotermi vastaavasti kuin edellä ja kun $f'_{a_{h_r}}$ on tämän kerroksen neuronin r aktivointifunktio, saadaan neuronin painojen ja vakiotermin osittaisderivaatoiksi ketjusäännön avulla

$$\frac{\partial}{\partial w_{h_{rs}}} f_\theta(x)_k = f'_{a_{\hat{x}_k}} (w_{\hat{x}_k}^T z) \frac{\partial}{\partial w_{h_{rs}}} w_{\hat{x}_k}^T z = f'_{a_{\hat{x}_k}} (w_{\hat{x}_k}^T z) w_{\hat{x}_{kr}} f'_{a_{h_r}} (w_{h_r}^T x) x_s.$$

Vastaavasti menetellen voidaan myös selvittää syvempien neuroverkkojen lähempänä syötekerrosta olevien piilokerrosten parametrien osittaisderivaatat. Koska tässä esimerkkinä käytetyssä neuroverkossa oli vain kolme kerrosta ja koska sen

kaikkien neuronien aktivointifunktio oli identiteettifunktio, saadaan sen parametrien osittaisderivaatoiksi aikaisemmin käytetyin merkinnöin

$$\frac{\partial}{\partial\theta_4}J(\theta) = \frac{1}{|X|}\sum_{x\in X}(f_\theta(x)_2 - x_2)h_1$$
$$\frac{\partial}{\partial\theta_3}J(\theta) = \frac{1}{|X|}\sum_{x\in X}(f_\theta(x)_1 - x_1)h_1$$
$$\frac{\partial}{\partial\theta_2}J(\theta) = \frac{1}{|X|}\sum_{x\in X}\sum_{k=1}^{2}(f_\theta(x)_k - x_k)w_{\hat{x}_{k1}}x_2$$
$$\frac{\partial}{\partial\theta_1}J(\theta) = \frac{1}{|X|}\sum_{x\in X}\sum_{k=1}^{2}(f_\theta(x)_k - x_k)w_{\hat{x}_{k1}}x_1.$$

5.5.2 Autoenkooderien anomaliapisteytys

Autoenkooderin syötekerroksen vektoreista x ulostulokerrokselta saatavien mahdollisimman tarkkojen kopioiden $\hat{x}$ anomaliapisteytyksenä voidaan käyttää niiden etäisyyden neliötä $||x - \hat{x}||^2$, josta usein käytetään myös nimitystä uudelleenrakennusvirhe [133]. Koska autoenkoodereita opetettaessa tämä virhe pyritään minimoimaan koko opetusaineiston osalta, joutuvat ne oppimaan identiteettifunktion, joka tuottaa keskimäärin pienimmän uudelleenrakennusvirheen. Uudelleenrakennusvirhe siis tavallaan kertoo, miten hyvin datapisteet vastaavat autoenkooderin opetukseen käytettyä aineistoa neuroverkon oppimien parametrien kannalta.

Vaikka autoenkoodereita voidaan käyttää sekä puoliohjattuun että ohjaamattomaan poikkeaman tunnistukseen, pyrkivät autoenkooderit lähtökohtaisesti oppimaan sellaiset parametrit, joilla kustannusfunktion arvo saadaan minimoitua opetusaineistossa [133]. Käytännössä ne siis voivat poikkeamien luonteesta ja määrästä riippuen oppia myös sellaiset parametrit, jotka tuottavat pienen uudelleenrakennusvirheen myös ainakin osalle opetukseen käytetyn aineiston poikkeamista.

Verkkoliikenteestä tapahtuvassa ohjaamattomassa poikkeaman tunnistuksessa autoenkoodereita ovat käyttäneet muun muassa Hawkins et al. [58] sekä Merrill ja Eskandarian [89]. Hawkins et al. olettivat autoenkoodereiden oppivan painot, jotka tuottavat poikkeamille normaaleja datapisteitä suuremman uudelleenrakennusvirheen. Merrill ja Eskandarian puolestaan esittävät, että autoenkooderit voivat oppia uudelleenrakentamaan myös poikkeamat hyvin, kunhan niiden annetaan päivittää painojaan tarpeeksi monta kertaa. He myös kuvasivat tekniikoita, joilla tätä ongelmaa voidaan pyrkiä välttämään.

6 Menetelmien vertailu

Luvun 6 tarkoituksena on asetetun tutkimuskysymyksen mukaisesti selvittää, miten Davidowin ja Mattesonin [35] esittämä FAMDAD soveltuu puoliohjattuun ja ohjaamattomaan poikkeaman tunnistukseen ensisijaisesti IoT-verkoista kerätyistä liikennevirtatietueisiin pohjautuvista aineistoista. Vaikka heidän mukaansa FAMDAD onkin ohjaamaton menetelmä, käsitellään sitä tässä työssä myös puoliohjattuna menetelmänä. FAMDAD-menetelmän soveltuvuutta arvioidaan valituissa aineistoissa lähinnä vertaamalla sen avulla saatuja tuloksia Mahalanobiksen etäisyyden ja autoenkoodereihin perustuvan menetelmän avulla saatuihin tuloksiin.

Koska FAMDAD periaatteessa vain kuvaa alkuperäisen aineiston alempiulotteiseen avaruuteen, käytetään sen avulla saatujen vektoreiden anomaliapisteytykseen tässä työssä Davidowin ja Mattesonin [35] tapaan sekä histogrammeihin perustuvaa SPAD-menetelmää että Isolation Forest -menetelmää. Ennen menetelmien ja niiden avulla saatujen tulosten tarkempaa läpikäyntiä tässä luvussa kuitenkin esitellään aluksi vertailussa käytetyt aineistot. Koska aineistoja on vain vähän ja koska ne poikkeavat toisistaan myös esimerkiksi muuttujiensa ja normaalin liikenteen profiiliensa osalta, ei vertailussa rajoituta pelkästään tilastollisiin testeihin.

6.1 Menetelmien vertailuun käytetyt aineistot

Poikkeaman tunnistusmenetelmien vertailussa käytettäviksi IoT-verkoista kerätyiksi aineistoiksi tässä työssä valittiin sekä Uuden Etelä-Walesin yliopiston (University of New South Wales, UNSW) julkaisema TON_IoT [6] että Stratosphere tutkimuslaboratorion julkaisema Aposemat IoT-23 [51]. Koska IoT-verkoista kerätyistä liikennevirtatietueista muodostettuja, sekä määrällisiä että laadullisia muuttujia sisältäviä, hyökkäyksiksi ja tavalliseksi liikenteeksi luokiteltuja aineistoja ei juurikaan ollut saatavilla, päädyttiin vertailussa käytettäviksi aineistoiksi valitsemaan myös KDD Cup 1999 [134] sekä uudempi UNSW:n julkaisema UNSW-NB15 [94].

Muita mahdollisia vertailussa käytettäviä IoT-aiheisia liikennevirtatietueisiin perustuvia aineistoja olisivat olleet muun muassa N-BaIoT [87] sekä UNSW:n BoT-IoT [73] ja UNSW-IoT [55]. Näitä aineistoja ei kuitenkaan valittu koska BoT-IoT ei sisäl-

tänyt juurikaan normaalia liikennettä ja toisaalta koska UNSW-IoT ja N-BaloT eivät sisältäneet valmiiksi FAMDAD-menetelmän vertailun kannalta oleellisia laadullisia muuttujia. Vaikka UNSW-NB15 ja KDD Cup 1999 ovatkin koostettu perinteisten verkkojen liikenteestä, voidaan niiden valintaa perustella IoT-sovellusten heterogeenisuudella ja sillä, että kumpaankin niistä liittyviin artikkeleihin löytyi useampi tuhat viittausta esimerkiksi Google Scholar -palvelun perusteella. Taulukossa 6.1 on esitetty menetelmien vertailuun käytettyjen aineistojen perustiedot.

Taulukko 6.1: Käytettyjen aineistojen perustiedot [6, 95, 51, 134]

Aineisto	Luontivuosi	IoT-aineisto	Erillinen opetusaineisto
TON_IoT	2019	Kyllä	Ei
Aposemat IoT-23	2018–2019	Kyllä	Ei
UNSW-NB15	2015	Ei	Kyllä
KDD CUP 1999	1998	Ei	Kyllä

6.1.1 IoT-verkoista kerätyt aineistot

Valituista aineistoista TON_IoT [6] on ainoa, joka sisältää valmiiksi nimenomaan IoT-verkkoihin tarkoitettujen tunkeutumisen havaitsemisjärjestelmien testaamista varten koostetun tietojoukon, jota ei tosin ole valmiiksi jaettu erillisiin opetus- ja testausaineistoihin. Koska aineisto on varsin tuore ja koska se sisältää myös varsin kattavasti luvussa 2.3 mainittuja tietoturvahaasteita kontrolloidussa testiympäristössä hyödyntäneistä verkkohyökkäyksistä syntyneitä liikennevirtatietueita, soveltuu se varsin hyvin valittujen poikkeaman tunnistusmenetelmien vertailuun.

Alsaedi et al. [6] mukaan TON_IoT-aineiston keräämistä varten rakennettu testiympäristö sisälsi muun muassa älytelevision, älypuhelimia sekä muita erinäisillä sensoreilla varustettuja IoT-laitteita, ja sen tarkoituksena oli simuloida tyypillistä useasta kerroksesta koostuvaa keskikokoista reunalaskentaa hyödyntävää IoT- tai IIoT-sovellusta (Industrial Internet of Things, IIoT). Vaikka heidän kuvaamastaan testiympäristöstä kerätty aineisto sisältääkin myös käyttöjärjestelmien lokitietoja sekä IoT-laitteiden keräämiä mittaustuloksia, keskitytään tässä työssä vain sen liikennevirtatietueita sisältäneeseen osaan.

Koska verkkoliikenteestä muodostettu osa TON_IoT-aineistoa ei sisältänyt erillistä opetusaineistoa, jaettiin se yksinkertaisesti puoliohjattujen poikkeaman tunnis-

tusmenetelmien vertailua varten ositetulla otannalla erillisiksi testi- ja opetusaineistoiksi siten, että testaukseen käytettiin noin 20 % sekä aineiston normaaleista että hyökkäyksiä sisältäneistä liikennevirtatietueista. Koska puoliohjattujen menetelmien opettamiseen käytetyn aineiston ei haluttu sisältävän hyökkäyksiä, muodostettiin opetusaineisto pelkästään jäljelle jääneistä normaaleista tietueista. Taulukossa 6.2 on esitetty tarkemmin, miten hyökkäyksistä ja normaalista liikenteestä syntyneet liikennevirtatietueet jakautuivat testi- ja opetusaineistoihin.

Taulukko 6.2: TON_IoT-aineiston liikennevirtatietueiden tyypit

Tyyppi	Opetusaineisto	Testiaineisto	Koko aineisto
Normaali	80 %	20 %	300000
Injektio	-	20 %	20000
Skannaus	-	20 %	20000
XSS	-	20 %	20000
DoS	-	20 %	20000
Kiristyshaittaohjelma	-	20 %	20000
Salasanan arvaus	-	20 %	20000
Takaovi	-	20 %	20000
DDoS	-	20 %	20000
MITM	-	20 %	1043

Koska eri tyyppisistä hyökkäyksistä syntyneet liikennevirtatietueet muodostavat noin kolmasosan koko TON_IoT-aineistosta, eivätkä siten välttämättä ole enää poikkeamia, käytettiin ohjaamattoman poikkeaman tunnistuksen vertailuun hyökkäyksiin liittyneitä datapisteitä vain sen verran, että ne muodostivat noin 2 % aineistosta, johon oli otettu mukaan kaikki normaalit havaintoyksiköt. Valittu poikkeavien ja normaalien datapisteiden suhde ohjaamattoman oppimisen osalta on tällöin myös sama, jota Davidow ja Matteson [35] käyttivät testatessaan menetelmänsä toimintaa valitsemassaan KDD CUP -aineistossa.

Vaikka Aposemat IoT-23 -aineisto [51] onkin koostettu oikeiden IoT-laitteiden verkkoliikenteestä, koostuu se toisin kuin TON_IoT useasta eri aikaan muodostetusta aineistosta, joiden tarkoituksena on ollut lähinnä edustaa joko IoT-laitteiden normaaliin käyttöön liittynyttä verkkoliikennettä tai vastaavasti johonkin hyökkäykseen liittynyttä verkkoliikennettä. Kaikkiaan IoT-23 sisältää kolme täysin normaalista verkkoliikenteestä koostettua aineistoa, jotka ovat laitteiden Somfy-älylukko,

Philips HUE ja Amazon Echo tuottamia. Eri haittaohjelmien tuottamaa verkkoliikennettä sisältävää aineistoa IoT-23:een kuuluu yhteensä kaksikymmentä.

Koska normaalista verkkoliikenteestä koostetut IoT-23-aineistot [51] sisältävät vain vähän liikennevirtatietueita, päädyttiin tässä työssä käyttämään vain yhtä kahdestakymmenestä Raspberry Pi -laitteessa suoritetusta haittaohjelmasta syntyneestä aineistosta. Vaikka tällainen menettely takaakin sen, että aineistossa on tarpeeksi eri taustatotuuden omaavia datapisteistä, jotka perustellusti liittyvät toisiinsa, on valittu CTU-IoT-Malware-Capture-1-1 sisältämiensä havaintoyksiköiden osalta kuitenkin varsin yksipuolinen esimerkiksi TON_IoT-aineistoon verrattuna.

Suurin osa CTU-IoT-Malware-Capture-1-1-aineistossa hyökkäyksiksi merkityistä datapisteistä on syntynyt Hide and Seek -haittaohjelman saastuttaman laitteen suorittamasta skannauksesta. Muita hyökkäyksiksi merkittyjä datapisteitä aineistossa on vain kahdeksan, ja ne on kaikki merkitty laitteen ja bottiverkon välisestä verkkoliikenteestä syntyneeksi. Vaikka Hide and Seek etsiikin uusia haavoittuvia laitteita skannaamalla ja perustuu osin samaan lähdekoodiin kuin Mirai, eroaa se siitä kuitenkin muun muassa siinä, että sen sisäisessä viestinnässä hyödynnetään sen tarpeisiin suunniteltua vertaisverkkoprotokollaa [86].

Koska käytetty aineisto ei luonnollisestikaan sisältänyt erillistä opetusaineistoa, jaettiin se TON_IoT-aineiston tapaan puoliohjattujen poikkeaman tunnistusmenetelmien vertailua varten erillisiin testi- ja opetusaineistoihin. Taulukossa 6.3 on esitetty tarkemmin, miten CTU-IoT-Malware-Capture-1-1-aineiston hyökkäyksistä ja normaalista verkkoliikenteestä syntyneiksi merkityt liikennevirtatietueet jakautuivat testi- ja opetusaineistoihin. Myös valitun IoT-23-aineiston osalta ohjaamattoman poikkeaman tunnistuksen vertailuun valittiin hyökkäyksiin liittyneitä datapisteitä vain sen verran, että ne muodostivat noin 2 % aineistosta, johon oli otettu mukaan kaikki normaalit havaintoyksiköt.

Taulukko 6.3: Aposemat IoT-23 -aineiston liikennevirtatietueiden tyypit

Tyyppi	Opetusaineisto	Testiaineisto	Koko aineisto
Normaali	n. 80 %	n. 20 %	469275
Skannaus	-	n. 20 %	539465
Bottiverkko	-	n. 20 %	8

Koska sekä TON_IoT [21] että Aposemat IoT-23 [51] sisältävät IoT-verkoista samalla avoimen lähdekoodin verkkoliikenteen analysaattorilla muodostettuja liiken-

nevirtatietueita, sisältävät ne paljon samoja muuttujia. Vaikka tietueet koostanut Zeek osaakin analysoida useita eri sovelluskerroksen protokollia ja muodostaa niitä kuvaavia liikennevirtatason tietoja, päädyttiin tässä työssä käyttämään molempien IoT-aiheisten aineistojen osalta vain yleisimpiä liikennevirtatietueisiin liittyneitä muuttujia. Taulukossa 6.4 on listattu valitut muuttujat, jotka siis voidaan muodostaa pelkästään IP-, ICMP-, TCP- ja UDP-otsikkotietojen perusteella [139].

Taulukko 6.4: TON_IoT- ja IoT-23-aineistosta käytetyt muuttujat

Muuttuja	Laadullinen
Protokolla	Kyllä
Yhteyden tila	Kyllä
Yhteyden kesto	Ei
Lähteen lähettämän hyötykuorman tavujen määrä	Ei
Lähteen lähettämien IP-pakettien määrä	Ei
Lähteen lähettämien IP-tavujen määrä	Ei
Kohteen lähettämän hyötykuorman tavujen määrä	Ei
Kohteen lähettämien IP-pakettien määrä	Ei
Kohteen lähettämien IP-tavujen määrä	Ei

Vaikka liikennevirtatietueet tyypillisesti sisältävät myös lähteen ja kohteen IP-osoitteen sekä protokollan tyypistä riippuen myös portin [120], ei näitä laadullisia muuttujia hyödynnetä tässä työssä niiden potentiaalisesti suuren arvojoukon ja dynaamisuuden vuoksi. Näiden ja Zeekin tunnistamaan tarkempaan sovelluskerroksen protokollaan liittyvien muuttujien pois jättämistä voidaan perustella myös sillä, että esimerkiksi TON_IoT-aineistossa useat yksityiset IP-osoitteet ja tarkemmat sovellustason protokollat liittyvät vain hyökkäyksiksi merkittyihin datapisteisiin. Puoliohjatussa tapauksessa tällaiset poikkeamat tunnistettaisiin käytännössä jo one-hot-koodauksen yhteydessä opetusaineistoon kuulumattoman arvon perusteella.

6.1.2 Perinteisistä verkoista kerätyt aineistot

UNSW-NB15 [95] on tunkeutumisen havaitsemisjärjestelmien arviointia varten koostettu tietojoukko, joka sisältää myös valmiin opetus- ja testiaineiston. Vaikka sitä muodostettaessa ei varsinaisesti olekaan huomioitu esineiden internetin käyttötapauksia, sisältää se myös vastaavia hyökkäyksiä kuin esimerkiksi IoT-laitteiden ja

perinteisten laitteiden verkkoliikenteestä muodostettu TON_IoT [6]. Koska UNSW-NB15 sisältää myös muuttujia, joiden arvot eivät kohdistu pelkästään havaintoyksikkönä toimivaan liikennevirtatietueeseen, poikkeaa se aikaisemmin esitellyistä aineistoista muutenkin kuin vain kuvaamansa verkkoliikenteen osalta.

Käytännössä UNSW-NB15-aineiston [94] muuttujat on muodostettu työkalujen Bro, nykyisin Zeek, ja Argus koostamista liikennevirtatason tiedoista. Näistä verkkoliikenteen analysaattoreista valmiiksi saatujen muuttujien ja niiden yksinkertaisten funktioiden lisäksi UNSW-NB15 sisältää siis myös muuttujia, joiden saamat arvot on koostettu juuri ennen havaintoyksikköä muodostuneista liikennevirtatietueista. Eräitä tällaisia muihin datapisteisiin liittyviä muuttujia ovat muun muassa saman kohteen IP-osoitteen ja saman lähteen IP-osoitteen omaavien liikennevirtatietueiden määrä sadan edellisen liikennevirtatietueen joukossa.

Taulukko 6.5: UNSW-NB15-aineiston havaintoyksiköiden tyypit

Tyyppi	Opetusaineisto	Testiaineisto	Koko aineisto
Normaali (Normal)	n. 80 %	n. 20 %	2203575
Yleinen (Generic)	-	n. 20 %	213718
Haavoittuvuudet (Exploits)	-	n. 20 %	28317
Odottaman syöte (Fuzzers)	-	n. 20 %	21517
Tiedustelu (Reconnaissance)	-	n. 20 %	11855
DoS	-	n. 20 %	3863
Haitallinen koodi (Shellcode)	-	n. 20 %	1511
Analysointi (Analysis)	-	n. 20 %	622
Takaovi (Backdoor)	-	n. 20 %	357
Madot (Worms)	-	n. 20 %	174

Koska valmis opetus- ja testiaineisto ovat vain kohtalaisen pieniä otoksia koko aineistosta [95], päädyttiin tässä työssä käyttämään koko aineistoa, joka jaettiin muiden aineistojen tapaan puoliohjattujen poikkeaman tunnistusmenetelmien vertailua varten erillisiin testi- ja opetusaineistoihin. UNSW-NB15-aineiston liikennevirtatietueiden sisältämien protokollien suuresta määrästä johtuen, tässä työssä päädyttiin hyödyntämään vain niitä datapisteitä, joiden protokolla oli TCP tai UDP. Taulukossa 6.5 on esitetty tarkemmin, miten näin valitut hyökkäyksistä ja normaalista liikenteestä syntyneet liikennevirtatietueet muodostivat puoliohjattujen menetelmien vertailuun käytetyn testi- ja opetusaineiston.

Toisin kuin IoT-aiheisten aineistojen osalta UNSW-NB15-aineistosta pyrittiin lähtökohtaisesti hyödyntämään taustatotuuteen liittymättömistä muuttujista kaikkia muita paitsi aikaleimoja, lähteen ja kohteen IP-osoitetta ja porttia sekä tarkempaa liikennevirtatietueeseen yhdistettyä sovellustason protokollaa. Koska TCP-yhteyden muodostukseen kuluneesta kokonaisajasta kertovan muuttujan kuitenkin havaittiin jo dokumentaationkin [95] perusteella olevan kahden muun muuttujan summana niiden lineaarikombinaatio, ei sitäkään päädytty sisällyttämään eri menetelmien vertailuissa käytettyjen muuttujien joukkoon.

UNSW-NB15 osalta ohjaamattomien poikkeaman tunnistusmenetelmien vertailuun käytettiin kaikkia taulukossa 6.5 esitettyjä normaaleja havaintoyksiköitä sekä sen verran siinä listattuja hyökkäyksiä, että niiden osuudeksi vertailuun käytetystä aineistosta saatiin sama, noin 2 %, mitä se IoT-verkoista kerättyjen aineistojen osaltakin oli. Koska sekä puoliohjattujen että ohjaamattomien menetelmien vertailuissa käytetyissä aineistoissa hyödynnettiin laajemman UNSW-NB15-aineiston [94] datapisteitä, sisälsivät ne otoksen kahden päivän aikana kolmen verkon simuloidusta liikenteestä muodostetuista liikennevirtoja kuvaavista havaintoyksiköistä.

KDD Cup 1999 [85, 134] on käytetyistä aineistoista vanhin. Se on Massachusettsin teknillisen korkeakoulun (Massachusetts Institute of Technology, MIT) Lincolnin laboratorion keräämästä 1998 DARPA -aineistosta vuonna 1999 järjestettyä vuosittaista tiedon louhinnan ja tietämyksen muodostuksen (Knowledge Discovery and Data Mining, KDD) kilpailua varten koostettu versio. Koska sen sisältämä Yhdysvaltain ilmavoimien lähiverkkoa simuloivasta ympäristöstä muodostettu valmis testiaineisto noudattaa eri jakaumaa kuin siitä muodostettu valmis opetusaineisto, päädyttiin tässä työssä hyödyntämään vain KDD Cup 1999 -opetusaineistoa.

Taulukko 6.6: KDD CUP 1999 -aineiston havaintoyksiköiden tyypit

Tyyppi	Opetusaineisto	Testiaineisto	Koko aineisto
Normaali	n. 80 %	n. 20 %	972781
DoS	-	n. 20 %	3883370
Luotaus/Skannaus	-	n. 20 %	41102
Etähyökkäys (R2L)	-	n. 20 %	1126
Käyttöoikeuksien korotus (U2R)	-	n. 20 %	52

Valmis KDD Cup 1999 -opetusaineisto jaettiin tässä työssä puoliohjattujen poik-

keaman tunnistusmenetelmien vertailua varten erillisiksi testi- ja opetusaineistoiksi. Taulukossa 6.6 on esitetty tarkemmin, miten hyökkäyksistä ja normaalista verkkoliikenteestä syntyneiksi merkityt liikennevirtatietueet jakautuivat testi- ja opetusaineistoihin. Hyökkäyksiin liittyneitä datapisteitä valittiin valmiista opetusaineistosta, vastaavasti kuin muiden aineistojen osalta, ohjaamattoman poikkeaman tunnistuksen vertailuun vain sen verran, että ne muodostivat noin 2 % aineistosta, johon oli otettu mukaan kaikki valmiin opetusaineiston normaalit havaintoyksiköt.

UNSW-NB15 [94] tavoin myös KDD Cup 1999 [134] sisältää muuttujia, joiden saamat arvot on koostettu eri havaintoyksiköihin liittyvistä liikennevirroista. Sadan edellisen liikennevirtatietueen sijaan tällaisten muuttujien arvot on tosin siinä koostettu kahden sekunnin aikaikkunan avulla. Näistä ja muista valmiin opetusaineiston taustatotuuteen liittymättömistä muuttujista päädyttiin tässä työssä käyttämään valittujen poikkeaman tunnistusmenetelmien vertailuun kaikkia muita paitsi liikennevirran sovellustason protokollaan liittyvää muuttujaa.

6.2 Vertailussa käytettyjen menetelmien toteutus

FAMDAD-menetelmän soveltuvuutta IoT-verkoista kerätyistä liikennevirtatietueista suoritettuun puoliohjattuun ja ohjaamattomaan poikkeaman tunnistukseen arvioidaan tässä työssä vertaamalla sitä Mahalanobiksen etäisyyteen ja autoenkoodereihin perustuvaan menetelmään. Eri menetelmien vertailuun käytetään edellisessä luvussa kuvattuja aineistoja. Koska FAMDAD käytännössä vain kuvaa alkuperäisen aineiston alempiulotteiseen avaruuteen, käytetään sen avulla saatujen vektoreiden anomaliapisteytykseen Davidowin ja Mattesonin [35] tapaan sekä histogrammeihin perustuvaa SPAD-menetelmää että Isolation Forest -menetelmää.

Koska Davidow ja Matteson [35] eivät maininneet käyttäneensä mitään menetelmää, jolla he olisivat pyrkineet poistamaan selvästi poikkeavat havainnot vertailuun käyttämistään aineistoista ennen tai osana FAMDAD-menetelmää, ei tässäkään työssä pyritty erityisesti poistamaan merkittävästi poikkeavia havaintoja osana esikäsittelyä. Jotta mikään opetusaineiston one-hot-koodauksella saaduista binaarimuuttujista ei olisi ollut vakio, päädyttiin puoliohjattuun oppimisen liittyvässä esikäsittelyssä kuitenkin poistamaan havaintoyksiköt, joiden laadullisten muuttujien saamat arvot eivät kuuluneet opetukseen käytettyjen datapisteiden saamien arvojen joukkoon. Mistään vertailuun käytetystä aineistosta ei kuitenkaan lopulta poistettu edes puolta promillea havaintoyksiköistä.

Koska kaikkiin vertailussa käytettyihin menetelmiin Mahalanobiksen etäisyyttä lukuun ottamatta liittyy niiden toimintaa ohjaavia asetuksia eli hyperparametreja, päädyttiin eri aineistoissa kokeilluista hyperparametrien arvojen yhdistelmistä raportoimaan jokaisen menetelmän osalta Campos et al. [24] tapaan sekä keskimääräinen suorituskyky että kyseiseen aineistoon parhaiten sopineella yhdistelmällä saavutettu suorituskyky. Yksi vaihtoehto olisi myös ollut valita eri menetelmille Vanhoeyvelin ja Martenin [141] tapaan sellaiset aineistokohtaiset hyperparametrien arvot, jotka tuottaisivat muissa vertailuun käytetyissä aineistoissa parhaan keskimääräisen sijoituksen. Näin ei kuitenkaan tässä työssä menetelty, koska vertailuun käytettyjä aineistoja, jotka eivät edes välttämättä olleet kaikkien hyperparametrien osalta kovin hyviä vertailukohtia toisilleen, oli vähemmän kuin heillä.

Vertailtujen menetelmien suorituskykymittariksi valittiin AUC, jonka laskemiseen eri hyperparametrien arvojen yhdistelmillä käytettiin puoliohjatussa oppimisessa viisinkertaista ristiinvalidointia, joka tässä työssä toteutettiin siten, että aineisto jaettiin ositetulla otannalla viiteen likimain yhtä suureen osaan, joista jokaista vuorollaan käytettiin testiaineistona muiden osien normaalien datapisteiden toimiessa opetusaineistona. Ristiinvalidoinnin AUC on näin saatujen testiaineistojen keskiarvo, ja siten yksittäisestä testiaineistosta saatua AUC:ia parempi arvio kokeilluilla hyperparametrien arvoilla vastaavista aineistoista opittaessa saavutettavasta suorituskyvystä [67, 133]. Myös ohjaamattoman oppimisen osalta ilmoitettu AUC on viiden suorituskerran, joissa kaikissa on käytetty normaalien datapisteiden lisäksi eri satunnaisotosta aineiston poikkeamista, keskiarvo.

6.2.1 Mahalanobiksen etäisyys

Mahalanobiksen etäisyys on ainoa vertailuun käytetyistä menetelmistä, jonka tuloksiin affiineilla muunnoksilla $AX + b$, joissa $A_{n \times n}$ on täyttä astetta ja b vakio, ei ole vaikutusta [121]. Käytännössä tämä tarkoittaa sitä, että sen avulla saatavien tulosten kannalta on sama, standardoidaanko vertailussa käytetyn aineiston muuttujat vai ei. Koska myös Davidowin ja Mattesonin [35] esittämät muuttujien painotukset ovat affiineja muunnoksia, voitaisiin vertailussa käytetty Mahalanobiksen etäisyys aineiston keskiarvoon periaatteessa laskea myös FAMDAD-menetelmällä saaduista pääkomponenteista luvussa 5.3 esitetyllä kaavalla 5.2.

Mahalanobiksen etäisyyden yhteydessä muuttujien keskiarvot, varianssit ja kovarianssit laskettiin puoliohjatussa tapauksessa opetusaineiston sisältämistä normaaleista datapisteistä ja ohjaamattomassa tapauksessa koko aineistosta. Havainto-

yksiköiden anomaliapisteytyksenä käytettiin siten joko Mahalanobiksen etäisyyttä opetusaineiston keskiarvoon tai koko aineiston keskiarvoon. Ennen lopullisen kovarianssimatriisin muodostamista aineistosta poistettiin vielä kaikki muuttujat, jotka yhdessä jäljellä olevien muuttujien jonkin osajoukon kanssa summautuivat aina johonkin vakioon. Puoliohjatussa tapauksessa nämä säännöt oltaisiin myös voitu tallentaa, ja niitä myöhemmin rikkovat datapisteet merkitä suoraan poikkeamiksi.

6.2.2 FAMDAD-menetelmä

Myös FAMDAD-menetelmän osalta sekä muuttujien esikäsittelyyn tarvittavat suhteelliset frekvenssit ja keskihajonnat että varsinaiset pääkomponentit määräävät ominaisvektorit ja niitä vastaavat ominaisarvot laskettiin puoliohjatussa tapauksessa opetusaineistosta ja ohjaamattomassa tapauksessa koko aineistosta. Jatkuville muuttujille käytetyt painokertoimet rajoitettiin ennen PCA:ta tapahtuvassa esikäsittelyssä Davidowin ja Mattesonin [35] tapaan arvoon $(\frac{10}{3})^{\frac{1}{2}}$. Koska he esittivät että poikkeamat erottuvat parhaiten ensimmäisiä ja viimeisiä pääkomponentteja käytettäessä, kokeiltiin tässä työssä heidän ajatustensa mukaisesti, millaisia tuloksia pääkomponenteilla, joiden ominaisarvoille λ pätee $\lambda \geq 1 + a \vee \lambda \leq 1 - a$, saadaan.

Myös molempiin FAMDAD:ilta saatavien vektoreiden anomaliapisteytykseen käytettyihin menetelmiin liittyy hyperparametreja. Koska kaikilla mahdollisilla hyperparametrien arvojen yhdistelmillä saatavien tulosten selvittämien olisi ollut aikaa vievää, käytettiin Isolation Forest -menetelmän hyperparametrien arvoina niitä, joita Liu et al. [81] käyttivät oletusarvoina. SPAD-menetelmän osalta jokaisen muuttujan arvoalue $[\bar{x_i} - 3s_i, \bar{x_i} + 3s_i]$, jossa $\bar{x_i}$ on muuttujan i keskiarvo ja s_i otoskeskihajonta, päädyttiin Aryalin et al. [12] tapaan jakamaan b:hen yhtä suureen osaan anomaliapisteytystä varten. Edellä $b = \lfloor log_2 N \rfloor + 1$, kun N on pistetodennäköisyyksien estimointiin käytetyn aineiston koko. Taulukossa 6.7 on esitetty tarkemmin, mitä edellä mainitsemattomia hyperparametrien arvoja eri aineistoissa käytettiin.

Taulukko 6.7: FAMDAD-menetelmän yhteydessä käytettyjä hyperparametreja

Menetelmä	Hyperparametri	Kokeillut arvot
FAMDAD	ominaisarvoleikkuri, a	0; 0,1, 0,5; 0,9
Isolation Forest	otoksen koko, n	256
Isolation Forest	puiden määrä, t	100

6.2.3 Autoenkoodereihin perustuva menetelmä

Autoenkoodereihin perustuvan menetelmän esikäsittelyssä aineistojen kvalitatiiviset muuttujat binarisoitiin muiden menetelmien tapaan one-hot-koodauksella. Tämän jälkeen sekä määrälliset että binarisoidut muuttujat standardointiin puoliohjatussa tapauksessa opetusaineistosta ja ohjaamattomassa tapauksessa koko aineistosta laskettujen keskiarvojen ja otoskeskihajontojen avulla. Koska autoenkooderit opetetaan muodostamaan tarkka kopio syötteestään, on identiteettifunktio luonnollinen valinta niiden ulostulokerroksen aktivointifunktioksi, varsinkin, kun muut tässä työssä käsitellyt aktivointifunktiot eivät pystyisi muodostamaan arvoja, jotka ovat yli yhden otoskeskihajonnan alle alkuperäisen muuttujan keskiarvon.

Piilokerrosten aktivointifunktiona tässä työssä käytettiin kaikissa kokeilluissa arkkitehtuureissa ReLu:a. Neuronien painot alustettiin satunnaisluvuilla tasajakaumasta, ja niiden vakiotermeiksi asetettiin ulostulokerroksella nolla ja piilokerroksilla Goodfellowin et al. [54] mainitsema 0,1. Autoenkooderien opetus toteutettiin luvussa 5.5.1 mainitulla tavallisella gradienttimenetelmällä, jossa kustannusfunktiona käytettiin keskineliövirhettä ja jossa oppimisnopeutena käytettiin arvoa 0,001. Koska tässä työssä käytetään eteenpäinsyöttäviin täysin kytkettyihin neuroverkkoihin perustuvia autoenkoodereita, joissa syöte- ja ulostulokerrosten välissä on yksi tai useampi piilokerros, joilla on vähemmän neuroneita kuin ulommilla kerroksilla, täytyy edellä kuvattujen hyperparametrien lisäksi vielä valita ainakin neuroverkon tarkempi arkkitehtuuri, kustannusfunktion gradienttien laskemiseen käytettyjen osajoukkojen koko sekä gradienttimenetelmän lopetusehto.

Hinton ja Salakhutdinov [59] ovat esittäneet, että useamman piilokerroksen käyttö autoenkoodereissa voi johtaa pienempään uudelleenrakennusvirheeseen testiaineistossa tiedon alempiulotteista esitystä tavoiteltaessa. Myös verkkoliikenteestä tapahtuvassa poikkeaman tunnistuksessa on käytetty syviä autoenkoodereita. Esimerkiksi jo Hawkins et al. [58] käyttivät kolmea piilokerrosta etsiessään poikkeamia. Työssään he mainitsivat käyttäneensä arkkitehtuureja, joissa sisimmällä piilokerroksella oli kolme neuronia ulompien piilokerrosten neuronien määrän ollessa joko 35, 40 tai 45. Myös Xu et al. [147] käyttivät useita piilokerroksia verkkoliikenteestä poikkeamia etsiessään. Parhaaseen tuloksen he pääsivät arkkitehtuurilla, jossa autoenkooderin eri kerrosten neuronien määrät olivat 122, 32, 5, 32, 122 sisääntulokerrokselta ulostulokerrokselle päin lueteltuina. Taulukossa 6.8 on esitetty tarkemmin, mitä edellä kuvatuista töistä vaikutteita saaneita arkkitehtuureja autoenkoodereihin liittyen tässä työssä kokeiltiin.

Taulukko 6.8: Autoenkoodereiden yhteydessä kokeillut hyperparametrit

Hyperparametri	Kokeillut arvot
Arkkitehtuuri, kun n muuttujaa	n,$\lfloor \frac{n}{6} \rfloor$,$n$ ja n,$\lfloor \frac{n}{3} \rfloor$,$\lfloor \frac{n}{8} \rfloor$,$\lfloor \frac{n}{3} \rfloor$,$n$ ja n,$\lfloor \frac{n}{2} \rfloor$,$\lfloor \frac{n}{4} \rfloor$,$\lfloor \frac{n}{8} \rfloor$,$\lfloor \frac{n}{4} \rfloor$,$\lfloor \frac{n}{2} \rfloor$,$n$

Kustannusfunktion J gradienttien $\nabla J(\theta)$ laskemiseen käytettyjen erillisten osajoukkojen eli minisatsien kooksi tässä työssä valittiin yksinkertaisesti 32 datapistettä. Käytännössä opetukseen käytetty aineisto jaettiin ensin näihin minisatseihin, joista sitten jokaista vuorollaan hyödynnettiin halutun autoenkooderin parametrien θ päivittämiseen yhdessä oppimisnopeuden η kanssa $\theta \leftarrow \theta - \eta \nabla J(\theta)$. Kun kaikkia minisatseja oli käytetty parametrien päivitykseen eli kun yksi epookki oli saatu valmiiksi, muodostettiin uudet minisatsit seuraavaa epookkia varten. Oppimista jatkettiin vastaavasti korkeintaan sadan epookin ajan, kunnes gradienttimenetelmän lopetusehto jonkin niistä päätteeksi täyttyi.

Lopetusehtona tässä työssä päädyttiin käyttämään aikaista lopetusta, jossa osa opetukseen tarkoitetuista datapisteistä varattiin erilliseksi validointiaineistoksi, josta jokaisen epookin päätteeksi laskettua kustannusfunktion arvoa käytettiin lopetusehdossa. Koska validointiaineistoa ei käytetä parametrien päivittämiseen, antaa siitä laskettu kustannusfunktion arvo paremman kuvan siitä, miten hyvin sen oppima identiteettifunktion approksimaatio yleistyy vastaavanlaisiin datapisteisiin, joilla sitä ei kuitenkaan olla opetettu [54]. Tässä työssä parametrien päivittäminen lopetettiin, kun validointiaineistosta laskettu keskineliövirhe ei ollut laskenut viiteen epookkiin. Näistä korkeintaan sadan epookin päätteeksi saaduista parametreista käytettiin niitä, joilla validointiaineistossa saatiin pienin keskineliövirhe.

Koska vertailussa käytettyjen autoenkoodereiden oli tarkoitus edustaa lähinnä yksinkertaisiin sellaisiin perustuvia menetelmiä, joihin Xu et al. [147] kuvaama menetelmäkin voitaneen lukea, päädyttiin tehtyjä hyperparametrivalintoja pitämään poikkeaman tunnistusmenetelmien vertailun kannalta riittävinä. Käytännössä autoenkoodereiden kykyä oppia muodostamaan mahdollisimman tarkka kopio myös poikkeavista syötteistään ei tässä työssä siis erityisesti rajoitettu edes ohjaamattomassa oppimisessa muuten kuin autoenkoodereiden arkkitehtuureissa olleiden pullonkaulojen ja edellä kuvatun aikaisen lopetuksen avulla.

6.3 Tulosten esittely

Tämän luvun tarkoituksena on arvioida FAMDAD-menetelmää sekä puoliohjatussa että ohjaamattomassa poikkeaman tunnistuksessa ensisijaisesti valituista aineistoista edellisessä luvussa kuvatuilla menetelmillä saatuja tuloksia vertaamalla. Vaikka poikkeaman tunnistusmenetelmien vertailuun käytettyjä aineistoja onkin vain neljä, tutkitaan menetelmien välisten suorituskykyerojen mahdollista tilastollista merkitsevyyttä niissä Demšarin [38] esittämästi Friedmanin [49] testillä. Käytännössä testi tehdään sekä puoliohjatussa että ohjaamattomassa poikkeaman tunnistuksessa erikseen aineistoihin parhaiten soveltuneilla hyperparametreilla saaduille tuloksille ja kaikkien kokeiltujen hyperparametrien tulosten keskiarvoille.

Koska Friedmanin testi perustuu tässä työssä vertailtujen menetelmien, joita on $p = 4$, edellisessä luvussa kuvatulla tavalla lasketun AUC:in perusteella saatuihin keskimääräisiin sijoituksiin r_j vertailuun käytetyissä aineistoissa, joita on $n = 4$, saadaan testin, jonka nollahypoteesin mukaan eri menetelmien keskimääräiset sijoitukset ovat peräisin samasta jakaumasta, testisuureeksi Friedmanin mukaan

$$\chi_r^2 = \frac{12n}{p(p+1)} \sum_{j=1}^{p} \left(r_j - \frac{p+1}{2} \right)^2,$$

joka noudattaa χ^2-jakaumaa vapausasteella $p - 1$, kun n ja p eivät ole liian pieniä. χ_r^2 arvoja t_i vastaavia todennäköisyyksiä $P(\chi_r^2 \geq t_i)$ on laskettu valmiiksi tässä työssä käytetyille pienille n ja p muun muassa Friedmanin toimesta.

6.3.1 Puoliohjatun poikkeaman tunnistuksen tulokset

Koska FAMDAD-menetelmän osalta päädyttiin kokeilemaan pääkomponentteja, joiden ominaisarvot poikkesivat vähintään a verran yhdestä, saatiin sen muodostamien vektoreiden anomaliapisteytykseen käytetyiltä menetelmiltä jokaisen aineiston osalta neljä eri anomaliapisteytystä, ja siten myös neljä edellisessä luvussa kuvatulla tavalla laskettua ROC-käyrän alle jäävää pinta-alaa. Myös autoenkoodereita hyödyntäneen menetelmän osalta jokaiseen aineistoon saatiin kolme eri uudelleenrakennusvirheeseen perustunutta anomaliapisteytystä, sillä sen kanssa kokeiltiin eri arkkitehtuureja. Taulukossa 6.9 on esitetty nämä normaalien ja poikkeavien datapisteiden erottelukyvystä kertovat tulokset Isolation Forest -menetelmän osalta. Sekä siinä että muissa vastaavissa taulukoissa jokaisen aineiston paras AUC, jota ei tosin aina ole helppoa havaita tulosten pyöristämisestä johtuen, on alleviivattu.

Taulukko 6.9: Isolation Forest -menetelmän tulokset

Ominaisarvoleikkuri	TON_IoT	IoT-23	UNSW-NB15	KDD CUP 1999	r_j
$a = 0$	0,879	0,903	0,988	0,997	2,00
$a = 0,1$	0,867	0,903	0,988	0,998	1,75
$a = 0,5$	0,833	0,903	0,988	0,998	2,25
$a = 0,9$	0,769	0,903	0,987	0,994	4,00
Keskiarvo	0,837	0,903	0,988	0,997	

Taulukosta 6.9 nähdään, että TON_IoT-aineistoa lukuun ottamatta saadut tulokset ovat kaikilla kokeilluilla pääkomponenttiyhdistelmillä hyvin samankaltaisia. Paras keskimääräinen sijoitus Isolation Forest -menetelmällä saavutettiin käyttämällä pääkomponentteja, joiden ominaisarvot eivät olleet välillä $]0,9;1,1[$. Vaikka valinta $a = 0,9$ tuottikin aina huonoimman tuloksen, ei eri pääkomponenttiyhdistelmien keskimääräisillä sijoituksilla ole Friedmanin testin mukaan tilastollisesti merkitsevää eroa. Koska $\chi_r^2 = 7,5$ ja $P(\chi_r^2 \geq 7,5) = 0,052$, ei χ_r^2 tosin jää kauas melkein merkitsevästä. Taulukossa 6.10 on esitetty vastaavat tulokset SPAD-menetelmän osalta.

Taulukko 6.10: SPAD-menetelmän tulokset

Ominaisarvoleikkuri	TON_IoT	IoT-23	UNSW-NB15	KDD CUP 1999	r_j
$a = 0$	0,842	0,910	0,989	0,997	2,25
$a = 0,1$	0,797	0,905	0,989	0,998	2,25
$a = 0,5$	0,686	0,904	0,990	0,999	2,00
$a = 0,9$	0,609	0,904	0,989	0,997	3,50
Keskiarvo	0,734	0,906	0,989	0,998	

Myös SPAD-menetelmän osalta nähdään, että kaikilla kokeilluilla pääkomponenttiyhdistelmillä saadut tulokset olivat TON_IoT-aineistoa lukuun ottamatta jokseenkin samankaltaisia. Koska taulukon 6.10 arvoista saadaan $\chi_r^2 = 3,3$ ja toisaalta koska $P(\chi_r^2 \geq 3,3) = 0,389$, ei Friedmanin testin perusteella voida tässäkään löytää tukea sille, että eri pääkomponenttiyhdistelmillä saatujen tulosten välillä olisi tilastollisesti merkitsevää eroa. Vaikka valinta $a = 0,5$ tuottikin parhaan keskimääräisen sijoituksen SPAD:illa, olivat kaikilla pääkomponenteilla saadut tulokset joko parhaita tai lähellä parhaita tuloksia kaikissa aineistoissa molempien anomaliapisteytyk-

seen käytettyjen menetelmien osalta. Taulukossa 6.11 on esitetty vastaavat tulokset autoenkoodereita hyödyntäneen menetelmän osalta.

Taulukko 6.11: Autoenkoodereihin perustuvan menetelmän tulokset

Arkkitehtuuri	TON_IoT	IoT-23	UNSW-NB15	KDD CUP 1999	r_j
$n, \lfloor \frac{n}{6} \rfloor, \ldots$	0,825	0,923	0,976	0,995	2,50
$n, \lfloor \frac{n}{3} \rfloor, \lfloor \frac{n}{8} \rfloor, \ldots$	0,856	0,929	0,987	0,995	2,00
$n, \lfloor \frac{n}{2} \rfloor, \lfloor \frac{n}{4} \rfloor, \lfloor \frac{n}{8} \rfloor, \ldots$	0,875	0,918	0,987	0,997	1,50
Keskiarvo	0,852	0,923	0,983	0,996	

Autoenkoodereiden osalta paras keskimääräinen sijoitus vertailuun käytetyissä aineistoissa saavutettiin arkkitehtuurilla, jossa oli viisi piilokerrosta. Tässäkään eri hyperparametrien arvojen välille ei tosin saatu Friedmanin testin perusteella tilastollisesti merkitsevää eroa, sillä taulukosta 6.11, jossa tällä kertaa oli vain kolme vertailtavaa arkkitehtuuria, saatiin, että $\chi_r^2 = 2$ ja toisaalta että $P(\chi_r^2 \geq 2) = 0,431$. Taulukossa 6.12 on esitetty puoliohjattujen poikkeaman tunnistusmenetelmien tulokset aineistoihin parhaiten soveltuneita hyperparametreja käytettäessä.

Taulukko 6.12: Vertailun tulokset parhailla hyperparametreilla

Menetelmä	TON_IoT	IoT-23	UNSW-NB15	KDD CUP 1999	r_j
Mahalanobis	0,879	0,906	0,984	0,996	3,25
Isolation Forest	0,879	0,903	0,988	0,998	2,25
SPAD	0,842	0,910	0,990	0,999	2,00
Autoenkooderi	0,875	0,929	0,987	0,997	2,50

Vaikka molemmat FAMDAD:ilta saatavia pääkomponentteja hyödyntävät menetelmät ovatkin tällä tavalla vertailtaessa keskimäärin parhaiten sijoittuneita, on hyvä huomata, että menetelmien keskimääräisillä sijoituksilla ei siltikään ole tilastollisesti merkitsevää eroa. Tässä Friedmanin testillä saadaan, että $\chi_r^2 = 2,1$ ja toisaalta että $P(\chi_r^2 \geq 2,1) = 0,649$. Tällainen vertailu on kuitenkin siinä mielessä ongelmallista, että saadut tulokset on periaatteessa mahdollista saavuttaa vain, jos kokeilluista hyperparametrien arvoista, joiden välillä ei tässä voitu osoittaa olevan tilastollisesti merkitsevää eroa, osattaisiin aina valita jokaiseen aineistoon parhaat il-

man että niillä saavutettuja tuloksia tiedettäisiin etukäteen. Varsinkin autoenkoodereihin perustuvien menetelmien osalta tämä on mielenkiintoinen haaste [99].

SPAD- ja Isolation Forest -menetelmän osalta olisi tosin voinut olla myös perusteltua käyttää aina kaikkia pääkomponentteja. Tällöin niiden osalta oltaisiin Mahalanobiksen etäisyyden tapaan saatu jokaisen aineiston osalta vain yksi tulos. Onkin mielenkiintoista havaita, että taulukossa 6.12 esitettyjen menetelmien sijoitukset eivät muuttuisi minkään aineiston osalta, vaikka SPAD ja Isolation Forest käyttäisivät aina kaikkia FAMDAD-menetelmältä saamiaan pääkomponentteja. Toisaalta Davidow ja Matteson nimenomaan esittivät, että poikkeavat havaintoyksiköt erottuvat selvästi ensimmäisillä ja viimeisillä pääkomponenteilla.

Taulukko 6.13: Vertailun tulokset keskimääräisillä hyperparametreilla

Menetelmä	TON_IoT	IoT-23	UNSW-NB15	KDD CUP 1999	r_j
Mahalanobis	0,879	0,906	0,984	0,996	2,25
Isolation Forest	0,837	0,903	0,988	0,997	2,75
SPAD	0,734	0,906	0,989	0,998	2,25
Autoenkooderi	0,852	0,923	0,983	0,996	2,75

Taulukossa 6.13 on esitetty vertailtujen puoliohjattujen poikkeaman tunnistusmenetelmien osalta eri hyperparametrien arvoilla saatujen tulosten keskiarvot aineistoittain. Saadut tulokset siis kuvastavat sitä, millainen suorituskyky vertailluilla menetelmillä keskimäärin saavutettaisiin, jos hyperparametrien arvot arvattaisiin kaikista kokeilluista [24]. Tässä vertailussa on siis hyvä huomata, että jokaiseen menetelmään Mahalanobiksen etäisyyttä lukuun ottamatta liittyi myös sellaisia hyperparametreja, joiden arvoja ei kokeiltu muuttaa lainkaan. Lisäksi pelkästään mahdollisia kokeiltavia arkkitehtuureja olisi jo ollut ääretön määrä.

Vaikka FAMDAD-menetelmältä saamiaan vektoreita anomaliapisteyttänyt Isolation Forest ja autoenkoodereihin perustunut menetelmä olivatkin tässä vertailussa keskimääräisten sijoitustensa perusteella kaksi huonointa menetelmää, eivät erot sijoituksissa olleet Friedmanin testin mukaan kuitenkaan lähelläkään tilastollisesti merkitseviä, sillä $\chi_r^2 = 0,6$ ja $P(\chi_r^2 \geq 0,6) = 0,928$. Puoliohjattujen poikkeaman tunnistusmenetelmien osalta ei siis voida suurella varmuudella sanoa, että FAMDAD yhdistettynä SPAD- tai Isolation Forest -menetelmään olisi sen huonompi tai parempi kuin muut vertailussa mukana olleet menetelmät. Näyttääkin siltä, että se voisi periaatteessa toimia myös puoliohjattuna poikkeaman tunnistusmenetelmänä.

FAMDAD-menetelmän soveltuminen myös puoliohjattuun poikkeaman tunnistukseen ei toisaalta ole kovin yllättävää, kun luvusta 5.4 muistetaan, että se on periaatteessa vain painotetuista määrällisistä muuttujista ja painotetuista one-hot-koodauksella saaduista binaarimuuttujista koostuvalle aineistolle tehty PCA, josta saatujen uusien muuttujien avulla voitaisiin periaatteessa myös laskea Mahalanobiksen etäisyys luvussa 5.3 esitetyllä kaavalla 5.2. Puoliohjatussa poikkeaman tunnistuksessa muuttujien huipukkuuteen perustuvien kerrointen käyttöä FAMDAD-menetelmän yhteydessä ei tosin voida perustella sillä, että aineiston sisältämät sovelluksen kannalta mielenkiintoiset poikkeamat vaikuttaisivat niihin. Käytännössä kertoimet määräytyivät kuitenkin pitkälti myös ohjaamattomassa poikkeaman tunnistuksessa aineiston normaaleiksi merkittyjen datapisteiden perusteella.

Onkin hyvä huomata, että vaikka havainnot vaikuttavat sitä enemmän muuttujien huipukkuuteen, mitä useamman keskihajonnan päässä ne niiden keskiarvoista ovat, on eri jakaumilla jo lähtökohtaisestikin eri huipukkuus [144]. Kolmesta poikkeava huipukkuus voi siis mielenkiintoisten poikkeamien lisäksi olla myös merkki siitä, että muuttuja ei noudata normaalijakaumaa. Esimerkiksi TON_IoT- ja IoT-23-aineistossa kaikkien määrällisten muuttujien kerroin oli sekä puoliohjatussa että ohjaamattomassa poikkeaman tunnistuksessa $(\frac{10}{3})^{\frac{1}{2}}$. Myös muissa aineistoissa hyökkäyksiksi merkittyjen datapisteiden vaikutus oli pientä, ja saattoi ihan yhtä hyvin myös laskea muuttujien huipukkuutta. Vaikka Davidowin ja Mattesonin [35] mukaan kertoimien onkin tarkoitus painottaa hännäkkäästi jakautuneita muuttujia, voisi niiden käytön tarkempi tutkiminen olla mielenkiintoista myös ohjaamattoman poikkeaman tunnistuksen osalta, varsinkin kun heidän työssään vertailemiensa kuvausten välille ei Friedmanin testillä saada tilastollisesti merkitsevää eroa.

6.3.2 Ohjaamattoman poikkeaman tunnistuksen tulokset

Myös ohjaamattomassa poikkeaman tunnistuksessa sekä FAMDAD:in muodostamia vektoreita hyödyntäneet poikkeaman tunnistusmenetelmät että autoenkoodereihin perustunut yksinkertainen menetelmä tuottivat jokaisesta vertailuun käytetystä aineistosta useamman eri tuloksen, jotka esitellään tässä luvussa vastaavasti kuin puoliohjatun poikkeaman tunnistuksen yhteydessä. Käytännössä vertailtavat menetelmät ovat samat, mutta nyt vertailuun käytetyissä aineistoissa on noin kaksi prosenttia hyökkäyksiksi merkittyjä datapisteitä eli tunkeutumisen havaitsemisen kannalta mielenkiintoisia poikkeamia. Taulukossa 6.14 on esitetty ROC-käyrän alle jäävät pinta-alat eri aineistoista Isolation Forest -menetelmän osalta.

Taulukko 6.14: Isolation Forest -menetelmän tulokset

Ominaisarvoleikkuri	TON_IoT	IoT-23	UNSW-NB15	KDD CUP 1999	r_j
$a = 0$	0,874	0,902	0,968	0,863	2,00
$a = 0,1$	0,876	0,901	0,960	0,902	2,50
$a = 0,5$	0,835	0,900	0,954	0,911	3,25
$a = 0,9$	0,778	0,901	0,963	0,955	2,25
Keskiarvo	0,841	0,901	0,961	0,908	

Vertailun tuloksista nähdään, että paras keskimääräinen sijoitus Isolation Forest -menetelmällä saavutettiin kaikkia pääkomponentteja käyttämällä. Koska Friedmanin testistä saadaan $\chi_r^2 = 2,1$ ja koska $P(\chi_r^2 \geq 2,1) = 0,649$, ei eri pääkomponenttiyhdistelmillä saatujen keskimääräisten sijoitusten välillä kuitenkaan voida sanoa olevan tilastollisesti merkitsevää eroa. Taulukosta 6.14 nähdään myös, että kaikkia pääkomponentteja käyttämällä KDD CUP 1999 -aineistosta saatiin tällä kertaa selvästi muita pääkomponenttiyhdistelmiä huonompi AUC.

Taulukko 6.15: SPAD-menetelmän tulokset

Ominaisarvoleikkuri	TON_IoT	IoT-23	UNSW-NB15	KDD CUP 1999	r_j
$a = 0$	0,841	0,900	0,957	0,790	2,25
$a = 0,1$	0,851	0,900	0,955	0,790	2,25
$a = 0,5$	0,756	0,900	0,951	0,824	3,00
$a = 0,9$	0,514	0,899	0,959	0,912	2,50
Keskiarvo	0,741	0,900	0,955	0,829	

SPAD-menetelmän osalta taulukosta 6.15 nähdään, että paras keskimääräinen sijoitus vertailuun käytetyissä aineistoissa saavutettiin sekä käyttämällä kaikkia pääkomponentteja että käyttämällä vain niitä pääkomponentteja, joiden ominaisarvot eivät olleet välillä $]0,9;1,1[$. Koska $\chi_r^2 = 0,9$ ja toisaalta koska $P(\chi_r^2 \geq 0,9) = 0,9$, ei tässäkään eri pääkomponenttiyhdistelmillä saatujen keskimääräisten sijoitusten välillä ollut tilastollisesti merkitsevää eroa. Vaikka $a = 0$ olikin molempien anomaliapisteytykseen käytettyjen menetelmien osalta paras keskimääräinen valinta, on mielenkiintoista havaita, miten erisuuntaisia TON_IoT- ja KDD CUP 1999 -aineistosta saadut tulokset olivat tällä valinnalla verrattuna valintaan $a = 0,9$.

Taulukko 6.16: Autoenkoodereihin perustuvan menetelmän tulokset

Arkkitehtuuri	TON_IoT	IoT-23	UNSW-NB15	KDD CUP 1999	r_j
$n, \lfloor \frac{n}{6} \rfloor, \ldots$	0,840	0,929	0,960	0,659	1,50
$n, \lfloor \frac{n}{3} \rfloor, \lfloor \frac{n}{8} \rfloor, \ldots$	0,851	0,917	0,948	0,501	2,00
$n, \lfloor \frac{n}{2} \rfloor, \lfloor \frac{n}{4} \rfloor, \lfloor \frac{n}{8} \rfloor, \ldots$	0,848	0,911	0,893	0,602	2,50
Keskiarvo	0,846	0,919	0,934	0,587	

Taulukosta 6.16 nähdään, että paras keskimäärinen sijoitus autoenkoodereita hyödyntäneen menetelmän osalta saatiin arkkitehtuurilla, jossa oli vain yksi piilokerros. Vaikka keskimääräisten sijoitusten välillä ei tässäkään ole tilastollisesti merkitsevää eroa, koska $\chi_r^2 = 2$ ja koska $P(\chi_r^2 \geq 2) = 0,431$, on silti mielenkiintoista havaita, että ohjaamattomassa poikkeaman tunnistuksessa eri arkkitehtuurien keskimääräisten sijoitusten järjestys on päin vastainen kuin puoliohjatussa. Lisäksi vaikuttaa siltä, että kaikkien kokeiltujen arkkitehtuurien on mahdollista oppia aivan liian tarkka identiteettifunktio KDD CUP 1999 -aineiston poikkeamille.

Taulukko 6.17: Vertailun tulokset parhailla hyperparametreilla

Menetelmä	TON_IoT	IoT-23	UNSW-NB15	KDD CUP 1999	r_j
Mahalanobis	0,868	0,900	0,941	0,934	2,75
Isolation Forest	0,876	0,902	0,968	0,955	1,25
SPAD	0,851	0,900	0,959	0,912	3,50
Autoenkooderi	0,851	0,929	0,960	0,659	2,50

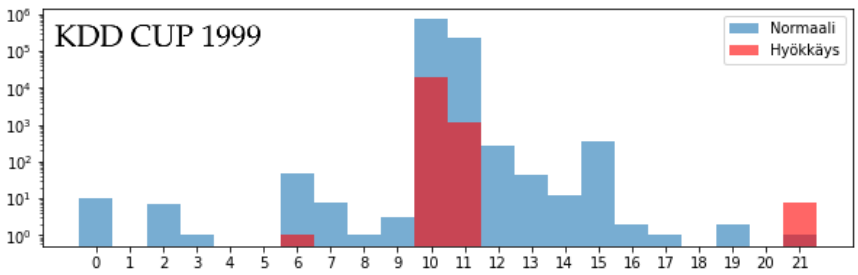
Taulukosta 6.17, jossa on esitetty ohjaamattomien poikkeaman tunnistusmenetelmien tulokset aineistoihin parhaiten soveltuneita hyperparametreja käytettäessä, nähdään, että Isolation Forest oli vertailun paras ja SPAD vertailun huonoin menetelmä keskimääräisten sijoitusten perusteella. Koska $\chi_r^2 = 6,3$ ja toisaalta koska $P(\chi_r^2 \geq 6,3) = 0,094$, ei eri menetelmien välille saatu kuitenkaan tilastollisesti merkitsevää eroa. Isolation Forest -menetelmän osalta on mielenkiintoista havaita, että vaikka sen kanssa oltaisiin käytetty aina kaikkia pääkomponentteja, olisi sen sijoitus muuttunut vain KDD CUP 1999 -aineistossa. Taulukossa 6.18 on esitetty vielä eri hyperparametrien arvoilla saatujen tulosten keskiarvot aineistoittain.

Taulukko 6.18: Vertailun tulokset keskimääräisillä hyperparametreilla

Menetelmä	TON_IoT	IoT-23	UNSW-NB15	KDD CUP 1999	r_j
Mahalanobis	0,868	0,900	0,941	0,934	2,00
Isolation Forest	0,841	0,901	0,961	0,908	2,00
SPAD	0,741	0,900	0,955	0,829	3,25
Autoenkooderi	0,846	0,919	0,934	0,587	2,75

Vaikka vertailtujen menetelmien eri hyperparametrien arvoilla saaduista tuloksista arvaamalla valittavien tulosten odotusarvoilla ei olekaan tässä tilastollisesti merkitsevää eroa, sillä $\chi_r^2 = 2,7$ ja $P(\chi_r^2 \geq 2,7) = 0,508$, on kuitenkin mielenkiintoista havaita, miten hyvin Mahalanobiksen etäisyys sijoittui sekä tässä että vastaavassa puoliohjattujen menetelmien yhteydessä tehdyssä vertailussa. Koska kokeiltujen hyperparametrien valinta oli osin mielivaltainen, on myös hyvä huomata, että se oli ainoa menetelmä, jolla samalla hyperparametrivalinnalla saavutettu tulos jäi kaikissa aineistoissa korkeintaan $0,03$ päähän aineiston parhaasta tuloksesta.

SPAD- ja Isolation Forest -menetelmän osalta on taas mielenkiintoista havaita, että kummallakin paras keskimääräinen sijoitus saavutettiin ohjaamattomassa poikkeaman tunnistuksessa käyttämällä kaikkia pääkomponentteja. Ainoa aineisto, jossa valinta oli selvästi molempien menetelmien osalta huono, oli KDD CUP 1999. Vaikka tässä työssä ei varsinaisesti pyrittykään löytämään tarkempia syitä näille eroille, on esimerkiksi SPAD-menetelmän osalta hyvä muistaa, että se jakaa jokaisen muuttujan arvovälin $[\bar{x}_i - 3s_i, \bar{x}_i + 3s_i]$ yhtä suurin osiin ja käyttää niiden suhteellisia frekvenssejä anomaliapisteytyksessään. Kuvasta 6.1 nähdäänkin, että aina keskitetyn, skaalatun ja kierretyn aineiston koordinaattien harvinaiset arvot eivät välttämättä tarkoita sovelluksen kannalta mielenkiintoisia poikkeamia.



Kuva 6.1: Erään pääkomponentin arvovälien frekvenssit logaritmisella asteikolla.

Käytännössä KDD CUP 1999 -aineistossa pääkomponenttien, joilla yksinään saatiin korkentaan 0,5 AUC SPAD:illa, osuus kaikista käytetyistä pääkomponenteista oli valinnoilla $a = 0$, $a = 0,1$, $a = 0,5$ ja $a = 0,9$ noin 43 %, 45 %, 40 % ja 32 %. Lähellä yhtä olevan ominaisarvon omaavat pääkomponentit eivät siis keskimäärin vaikuttaneet olevan niin hyödyllisiä tässä aineistossa. On tietysti mielenkiintoista kysyä, olisivatko tunkeutumisen havaitsemisen kannalta mielenkiintoiset poikkeamat kuvassa 6.1 voineet olla myös sellaisia, että ne olisivat osuneet arvoväleille, joilla normaaleja datapisteitä ei juuri esiintynyt. SPAD- ja Isolation Forest -menetelmän eri valinnoilla saatujen tulosten järjestys ei olisi muuttunut, vaikka huipukkuuteen perustuva kerroin olisi laskettu vain KDD CUP 1999 normaaleista datapisteistä.

6.3.3 Pohdintaa

Davidowin ja Mattesonin [35] esittämä FAMDAD vaikuttaa lupaavalta poikkeaman tunnistusmenetelmältä sekä puoliohjattuun että ohjaamattomaan poikkeaman tunnistukseen, varsinkin jos se nähdään vain PCA:na, jossa muuttujille voidaan tarvittaessa antaa eri painot ja jossa tarvittaessa voidaan käyttää vain osaa pääkomponenteista. Tällöin se tosin muistuttaa paljon esimerkiksi Aryal et al. [12] esittämää paranneltua SPAD-menetelmää, jossa anomaliapisteytyksessä käytetään alkuperäisten muuttujien lisäksi myös pääkomponenttianalyysin avulla saatuja uusia muuttujia. Aryal et al. julkaisemista puoliohjattuun poikkeaman tunnistukseen liittyvistä tuloksista nähdään myös, että heidän suorittamissaan vertailuissa eri menetelmien keskimääräisille sijoituksille saadaan tilastollisesti merkitsevä ero.

Ensin he vertasivat pääkomponentteja käyttävää SPAD-menetelmää alkuperäisiä muuttujia käyttävään neljään muuhun menetelmään, joiden joukossa oli myös SPAD ja Isolation Forest. Koska he käyttivät viittätoista aineistoa, saadaan heidän taulukoimistaan tuloksista, että $\chi_r^2 \approx 16,37$ ja että $P(\chi_r^2 \geq 16,37) < 0,005$. Koska Friedmanin testin perusteella keskimääräiset sijoitukset eivät suurella varmuudella ole satunnaisia, voidaan eri menetelmien i, j keskimääräisten sijoituserojen $|r_i - r_j|$ tilastollista merkitsevyyttä tutkia Demšarin [38] kuvaamasti Nemenyin testin kriittisen sijoituseron avulla. Jos $|r_i - r_j|$ on ainakin yhtä suuri kuin kriittinen sijoitusero

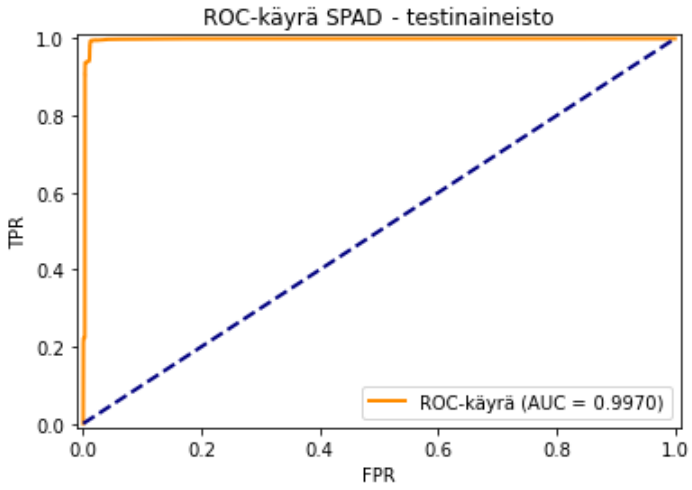
$$CD = q_\alpha \sqrt{\frac{p(p+1)}{6n}},$$

jossa n on aineistojen määrä ja p vertailtavien menetelmien määrä, on se valmiiksi taulukoitujen q_α mukaisella merkitsevyystasolla merkittävä. Heidän vertailussaan

pääkomponentteja käyttänyt SPAD oli parempi kuin alkuperäisiä muuttujia käyttäneet SPAD ja Isolation Forest merkitsevyystasolla, jossa sijoituseroa voidaan pitää virheellisesti merkittävänä korkeintaan 5 % todennäköisyydellä.

Vertailussaan Aryal et al. [12] kokeilivat käyttää pääkomponentteja myös Isolation Forest -menetelmän yhteydessä ja saavuttivatkin niillä parempia tuloksia kuin pelkästään alkuperäisiä muuttujia käyttämällä. Ehkä selkein ero FAMDAD:in ja Aryal et al. kokeilemien menetelmien välillä liittyykin muuttujien skaalaukseen ja siihen, että FAMDAD:issa käytetään vain PCA:sta saatavia pääkomponentteja. Niissä molemmissa kuitenkin anomaliapisteytetään alkuperäisten muuttujien muunnosten ja niistä pääkomponenttianalyysin, jossa muuttujille voidaan antaa eri painot, avulla saatujen muuttujien osajoukkojen muodostamia aineistoja. SPAD:ia ja Isolation Forestia anomaliapisteytykseen käytettäessä voidaan myös löytää poikkeamia, jotka ovat tavallisten datapisteiden muodostamien klustereiden välissä.

Vaikka kaikkia FAMDAD:ilta saatuja pääkomponentteja anomaliapisteyttäneet SPAD ja Isolation Forest olivatkin puoliohjatussa poikkeaman tunnistuksessa muita menetelmiä parempia keskimääräisten sijoitustensa perusteella, löysivät ne vain yhdestä aineistosta yli 20 % poikkeamista, kun raja-arvoa, jolla vääriä hälytyksiä tuli korkeintaan 0,1 %, suuremman anomaliapisteytyksen saaneet datapisteet merkittiin poikkeamiksi. Tunkeutumisen havaitsemisessa ne siis joko tuottaisivat paljon vääriä hälytyksiä tai jättäisivät suuren osan hyökkäyksistä havaitsematta, olettaen, että IDS:n sensoreiden keräämät aineistot muistuttaisivat vertailussa käytettyjä. Kuvassa 6.2 on SPAD-menetelmällä saatu ROC-käyrä KDD CUP 1999 -aineistosta.



Kuva 6.2: SPAD-menetelmällä saatu ROC-käyrä KDD CUP 1999 -aineistosta.

7 Yhteenveto

Tässä työssä selvitettiin, miten Davidowin ja Mattesonin esittämä FAMDAD soveltuu puoliohjattuun ja ohjaamattomaan poikkeaman tunnistukseen ensisijaisesti IoT-verkoista kerätyistä liikennevirtatietueisiin pohjautuvista aineistoista. FAMDAD-menetelmän soveltuvuutta arvioitiin sekä teoreettisesti että empiirisesti vertaamalla sitä Mahalanobiksen etäisyyteen ja autoenkoodereita hyödyntäneeseen menetelmään. Ennen varsinaista vertailua työssä kuitenkin luotiin vielä katsaus esineiden internetiin ja sen tietoturvaan, tunkeutumisen havaitsemisjärjestelmiin sekä erityyppisiin poikkeamiin ja niiden tunnistamiseen verkkoliikenteestä.

Työn teoriaosuudessa osoitettiin, että FAMDAD-menetelmä voidaan nähdä Jolliffen mainitsemana yleistettynä pääkomponenttianalyysinä, jossa havainnoille ja muuttujille voidaan määrittää painot ja metriikat. Käytännössä FAMDAD keskittää, skaalaa ja kiertää aineiston, ja on siten affiini muunnos. Mahalanobiksen etäisyyttä aineiston keskipisteeseen laskettaessa ei siis periaatteessa ole merkitystä, suoritetaanko FAMDAD-menetelmän mukainen PCA vai ei, olettaen, että sitä laskettaessa käytetään kovarianssimatriisin yleistettyä käänteismatriisia. Anomaliapisteytykseen Davidow ja Matteson käyttivätkin sekä SPAD:ia että Isolation Forestia.

Työn empiirisessä osuudessa vertailtujen menetelmien suorituskykyä arvioitiin niiden anomaliapisteytyksistä laskettujen ROC-käyrien alle jäävien pinta-alojen perusteella. Koska SPAD oli puoliohjatun ja Isolation Forest ohjaamattoman poikkeaman tunnistuksen paras menetelmä, vaikuttaa FAMDAD-menetelmä soveltuvan niiden kanssa sekä puoliohjattuun että ohjaamattomaan poikkeaman tunnistukseen vastaavista aineistoista ainakin suunnilleen yhtä hyvin kuin sen verrokkeina toimineet menetelmät. Friedmanin testin perusteella eri menetelmien keskimääräisillä sijoituksilla ei kuitenkaan voitu osoittaa olevan tilastollisesti merkitsevää eroa.

Empiirisen osuuden yhteydessä havaittiin myös, että ohjaamattomassa poikkeaman tunnistuksessa parhaat keskimääräiset sijoitukset sekä SPAD- että Isolation Forest -menetelmällä saavutettiin kaikkia pääkomponentteja käyttämällä. Työssä tuotiin myös Friedmanin testin avulla esiin, että Davidowin ja Mattesonin suorittamassa vertailussa huipukkuuteen perustuvia kertoimia käyttäneiden menetelmien ja muiden vastaavien menetelmien välillä ei vielä ollut tilastollisesti merkitsevää eroa.

Lähteet

[1] AGGARWAL, C. C. *Outlier Analysis*, 2nd ed. Springer International Publishing AG, Cham, Switzerland, 2017.

[2] AHMAD, Z., SHAHID KHAN, A., WAI SHIANG, C., ABDULLAH, J., JA AHMAD, F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies 32*, 1 (2021), e4150.

[3] AHMED, M. Collective Anomaly Detection Techniques for Network Traffic Analysis. *Annals of Data Science 5*, 4 (2018), 497–512.

[4] AISSA, N. B., JA GUERROUMI, M. Semi-supervised Statistical Approach for Network Anomaly Detection. *Procedia Computer Science 83* (2016), 1090–1095.

[5] AL-FUQAHA, A., GUIZANI, M., MOHAMMADI, M., ALEDHARI, M., JA AYYASH, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials 17*, 4 (2015), 2347–2376.

[6] ALSAEDI, A., MOUSTAFA, N., TARI, Z., MAHMOOD, A., JA ANWAR, A. TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems. *IEEE Access 8* (2020), 165130–165150.

[7] AMAZON WEB SERVICES, INC. Architecture Best Practices for IoT. URL `https://aws.amazon.com/architecture/iot/?awsf.content-type=content-type%23reference-arch-diagram`, viitattu 14.2.2022.

[8] ANDERSON, R. J. *Security Engineering: A Guide to Building Dependable Distributed Systems*, 2nd ed. Wiley Publishing, Inc., Indianapolis, IN, USA, 2008.

[9] ANDERSON, T. W. *An Introduction to Multivariate Statistical Analysis*, 3rd ed. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2003.

[10] ANTONAKAKIS, M., APRIL, T., BAILEY, M., BERNHARD, M., BURSZTEIN, E., COCHRAN, J., DURUMERIC, Z., HALDERMAN, J. A., INVERNIZZI, L., KALLITSIS, M., KUMAR, D., LEVER, C., MA, Z., MASON, J., MENSCHER, D., SEAMAN, C., SULLIVAN, N., THOMAS, K., JA ZHOU, Y. Understanding the Mirai Botnet. Julkaisusarjassa *Proceedings of the 26th USENIX Conference on Security Symposium* (Vancouver, BC, Canada, Elokuu 2017), USENIX Association, 1093–1110.

[11] ARGUS. openargus - Home. URL `https://openargus.org/`, viitattu 14.2.2022.

[12] ARYAL, S., BANIYA, A. A., JA SANTOSH, K. Improved histogram-based anomaly detector with the extended principal component features. *arXiv preprint arXiv:1909.12702* (2019).

[13] ARYAL, S., TING, K. M., JA HAFFARI, G. Revisiting Attribute Independence Assumption in Probabilistic Unsupervised Anomaly Detection. Julkaisusarjassa *Proceedings of the 11th Pacific Asia Workshop on Intelligence and Security Informatics* (Auckland, New Zealand, Huhtikuu 2016), Springer, 73–86.

[14] ASHTON, K. That 'Internet of Things' Thing. *RFID Journal 22*, 7 (2009), 97–114.

[15] BACE, R., JA MELL, P. *Intrusion Detection Systems*. Tekninen raportti NIST Special Publication (SP) 800-31, National Institute of Standards and Technology, Gaithersburg, MD, USA, Marraskuu 2001.

[16] BALDI, P., JA HORNIK, K. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks 2*, 1 (1989), 53–58.

[17] BARCELÓ-VIDAL, C., MARTÍN-FERNÁNDEZ, J. A., JA PAWLOWSKY-GLAHN, V. Comment on "Singularity and Nonnormality in the Classification of Compositional Data" by G. C. Bohling, J. C. Davis, R. A. Olea, and J. Harff. *Mathematical Geology 31*, 5 (1999), 581–585.

[18] BASILEVSKY, A. *Applied Matrix Algebra in the Statistical Sciences*. Dover Publications, Inc., Mineola, NY, USA, 2013.

[19] BHUYAN, M. H., BHATTACHARYYA, D. K., JA KALITA, J. K. Network Anomaly Detection: Methods, Systems and Tools. *IEEE Communications Surveys & Tutorials 16*, 1 (2014), 303–336.

[20] BLANK, A. G. *TCP/IP Foundations*. SYBEX Inc., Alameda, CA, USA, 2004.

[21] BOOIJ, T. M., CHISCOP, I., MEEUWISSEN, E., MOUSTAFA, N., JA DEN HARTOG, F. T. ToN_IoT: The Role of Heterogeneity and the Need for Standardization of Features and Attack Types in IoT Network Intrusion Data Sets. *IEEE Internet of Things Journal 9*, 1 (2022), 485–496.

[22] BOSWORTH, S., KABAY, M., JA WHYNE, E. *Computer Security Handbook, Set*, 6th ed. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2014.

[23] BRIERLEY, C., PONT, J., ARIEF, B., BARNES, D. J., JA HERNANDEZ-CASTRO, J. PaperW8: an IoT bricking ransomware proof of concept. Julkaisusarjassa *Proceedings of the 15th International Conference on Availability, Reliability and Security* (Virtual Event, Ireland, Elokuu 2020), ACM, 1–10.

[24] CAMPOS, G. O., ZIMEK, A., SANDER, J., CAMPELLO, R. J. G. B., MICENKOVÁ, B., SCHUBERT, E., ASSENT, I., JA HOULE, M. E. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery 30*, 4 (2016), 891–927.

[25] CERT COORDINATION CENTER. Dahua Security DVRs contain multiple vulnerabilities. URL `https://www.kb.cert.org/vuls/id/800094`, viitattu 14.2.2022.

[26] CHANDOLA, V., BANERJEE, A., JA KUMAR, V. Anomaly detection: A survey. *ACM Computing Surveys 41*, 3 (2009), 1–58.

[27] CHEN, Z., YEO, C. K., LEE, B. S., JA LAU, C. T. Autoencoder-based network anomaly detection. Julkaisusarjassa *2018 Wireless Telecommunications Symposium (WTS)* (Phoenix, AZ, USA, Huhtikuu 2018), IEEE, 1–5.

[28] CHONG, E. K. P., JA ŻAK, S. H. *An introduction to optimization*, 2nd ed. John Wiley & Sons, Inc., New York, NY, USA, 2004.

[29] CISCO. Cisco Secure Network Analytics (formerly Stealthwatch) At-a-Glance. URL `https://www.cisco.com/c/en/us/products/collateral/`

security/stealthwatch/at-a-glance-c45-736510.html, viitattu 14.2.2022.

[30] CLAISE, B., SADASIVAN, G., VALLURI, V., JA DJERNAES, M. *Cisco Systems NetFlow Services Export Version 9*, RFC 3954, Lokakuu 2004.

[31] CLAISE, B., TRAMMELL, B., JA AITKEN, P. *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information*, RFC 7011, Syyskuu 2013.

[32] CONTA, A., DEERING, S., JA GUPTA, M. *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*, RFC 4443, Maaliskuu 2006.

[33] CRAMÉR, H. *Mathematical Methods of Statistics (PMS-9)*, vol. 9. Princeton University Press, USA, 2016.

[34] DARGIE, W., JA POELLABAUER, C. *Fundamentals of Wireless Sensor Networks: Theory and Practice*. John Wiley & Sons, Chichester, United Kingdom, 2010.

[35] DAVIDOW, M., JA MATTESON, D. S. Factor Analysis of Mixed Data for Anomaly Detection. *arXiv preprint arXiv:2005.12129* (2020).

[36] DAVIS, J. J., JA CLARK, A. J. Data preprocessing for anomaly based network intrusion detection: A review. *Computers & Security 30*, 6–7 (2011), 353–375.

[37] DEERING, S. E., JA HINDEN, R. M. *Internet Protocol, Version 6 (IPv6) Specification*, RFC 8200, Heinäkuu 2017.

[38] DEMŠAR, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research 7*, 1 (2006), 1–30.

[39] DENNING, D. E. An Intrusion-Detection Model. *IEEE Transactions on Software Engineering SE-13*, 2 (1987), 222–232.

[40] DIERKS, T., JA RESCORLA, E. *The Transport Layer Security (TLS) Protocol Version 1.2*, RFC 5246, Elokuu 2008.

[41] DIFFIE, W., JA HELLMAN, M. E. Privacy and authentication: An introduction to cryptography. *Proceedings of the IEEE 67*, 3 (1979), 397–427.

[42] DUAN, H., WEAVER, N., ZHAO, Z., HU, M., LIANG, J., JIANG, J., LI, K., JA PAXSON, V. Hold-On: Protecting Against On-Path DNS Poisoning. Julkaisusarjassa *Securing and Trusting Internet Names (SATIN 2012)* (London, UK, Maaliskuu 2012).

[43] ELSWORTH, S., JA GÜTTEL, S. Time Series Forecasting Using LSTM Networks: A Symbolic Approach. *arXiv preprint arXiv:2003.05672* (2020).

[44] ETZION, O. Differences between the IoT and Traditional Internet. URL `https://www.rtinsights.com/differences-between-the-iot-and-traditional-internet/`, viitattu 14.2.2022.

[45] FAMILI, A., SHEN, W.-M., WEBER, R., JA SIMOUDIS, E. Data Preprocessing and Intelligent Data Analysis. *Intelligent Data Analysis 1*, 1 (1997), 3–23.

[46] FAWCETT, T. An introduction to ROC analysis. *Pattern Recognition Letters 27*, 8 (2006), 861–874.

[47] FIELDING, R., GETTYS, J., MOGUL, J., FRYSTYK, H., MASINTER, L., LEACH, P., JA BERNERS-LEE, T. *Hypertext Transfer Protocol – HTTP/1.1*, RFC 2616, Kesäkuu 1999.

[48] FREMANTLE, P., JA SCOTT, P. A survey of secure middleware for the Internet of Things. *PeerJ Computer Science 3* (2017), e114.

[49] FRIEDMAN, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association 32*, 200 (1937), 675–701.

[50] GALEONE, P. *Hands-On Neural Networks with TensorFlow 2.0*. Packt Publishing, Birmingham, UK, 2019.

[51] GARCIA, S., PARMISANO, A., JA ERQUIAGA, M. J. IoT-23: A labeled dataset with malicious and benign IoT network traffic. URL `https://www.stratosphereips.org/datasets-iot23`, viitattu 14.2.2022.

[52] GARCIA-MORCHON, O., KUMAR, S. S., JA SETHI, M. *Internet of Things (IoT) Security: State of the Art and Challenges*, RFC 8576, Huhtikuu 2019.

[53] GARTNER. Gartner Says 5.8 Billion Enterprise and Automotive IoT Endpoints Will Be in Use in 2020. URL `https://www.gartner.com/en/`

newsroom/press-releases/2019-08-29-gartner-says-5-8-billion-enterprise-and-automotive-io, viitattu 14.2.2022.

[54] GOODFELLOW, I., BENGIO, Y., JA COURVILLE, A. *Deep Learning*. The MIT Press, Cambridge, MA, USA, 2016.

[55] HAMZA, A., GHARAKHEILI, H. H., BENSON, T. A., JA SIVARAMAN, V. Detecting Volumetric Attacks on loT Devices via SDN-Based Monitoring of MUD Activity. Julkaisusarjassa *Proceedings of the 2019 ACM Symposium on SDN Research* (San Jose, CA, USA, Huhtikuu 2019), ACM, 36–48.

[56] HAMZA, A., GHARAKHEILI, H. H., JA SIVARAMAN, V. Combining MUD Policies with SDN for IoT Intrusion Detection. Julkaisusarjassa *Proceedings of the 2018 Workshop on IoT Security and Privacy* (Budapest, Hungary, Elokuu 2018), ACM, 1–7.

[57] HAWKINS, D. M. *Identification of outliers*. Chapman and Hall, London, 1980.

[58] HAWKINS, S., HE, H., WILLIAMS, G., JA BAXTER, R. Outlier Detection Using Replicator Neural Networks. Julkaisusarjassa *International Conference on Data Warehousing and Knowledge Discovery* (Aix-en-Provence, France, Syyskuu 2002), Springer, 170–180.

[59] HINTON, G. E., JA SALAKHUTDINOV, R. R. Reducing the Dimensionality of Data with Neural Networks. *Science 313*, 5786 (2006), 504–507.

[60] HOFSTEDE, R., ČELEDA, P., TRAMMELL, B., DRAGO, I., SADRE, R., SPEROTTO, A., JA PRAS, A. Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX. *IEEE Communications Surveys & Tutorials 16*, 4 (2014), 2037–2064.

[61] HOU, X., JIANG, Z., JA TIAN, X. The detection and prevention for ARP Spoofing based on Snort. Julkaisusarjassa *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)* (Taiyuan, China, Lokakuu 2010), vol. 5, IEEE, V5–137–V5–139.

[62] HOWARD, J. D., JA LONGSTAFF, T. A. *A Common Language for Computer Security Incidents*. Tekninen raportti SAND98-8667, Sandia National Laboratories, Albuquerque, NM, USA, Lokakuu 1998.

[63] IDC. IoT Growth Demands Rethink of Long-Term Storage Strategies, says IDC. URL https://www.idc.com/getdoc.jsp?containerId=prAP46737220, viitattu 14.2.2022.

[64] IEEE STD 2413-2019. *IEEE Standard for an Architectural Framework for the Internet of Things (IoT)*, 2020.

[65] IEEE STD 802.15.4-2020. *IEEE Standard for Low-Rate Wireless Networks*, 2020.

[66] INTERNATIONAL STANDARD ISO/IEC 8802-2:1998. *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 2: Logical link control*, 1998.

[67] JAMES, G., WITTEN, D., HASTIE, T., JA TIBSHIRANI, R. *An Introduction to Statistical Learning with Applications in R*, 2nd ed. Springer Science & Business Media, New York, NY, USA, 2021.

[68] JOHNSON, R. A., JA WICHERN, D. W. *Applied Multivariate Statistical Analysis*, 6th ed. Pearson Education, Inc., Upper Saddle River, NJ, USA, 2007.

[69] JOLLIFFE, I. T. *Principal Component Analysis*, 2nd ed. Springer-Verlag New York, Inc., New York, NY, USA, 2002.

[70] KENT, S., JA SEO, K. *Security Architecture for the Internet Protocol*, RFC 4301, Joulukuu 2005.

[71] KESSY, A., LEWIN, A., JA STRIMMER, K. Optimal Whitening and Decorrelation. *The American Statistician 72*, 4 (2018), 309–314.

[72] KISMET. Alerts and WIDS. URL https://www.kismetwireless.net/docs/readme/alerts_and_wids/, viitattu 14.2.2022.

[73] KORONIOTIS, N., MOUSTAFA, N., SITNIKOVA, E., JA TURNBULL, B. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Future Generation Computer Systems 100* (2019), 779–796.

[74] KREBS, B. Who Makes the IoT Things Under Attack? URL https://krebsonsecurity.com/2016/10/who-makes-the-iot-things-under-attack/, viitattu 14.2.2022.

[75] KUROSE, J. F., JA ROSS, K. W. *Computer Networking: A Top-Down Approach*, 6th ed. Pearson Education, Inc., Upper Saddle River, NJ, USA, 2013.

[76] KUSHALNAGAR, N., MONTENEGRO, G., JA SCHUMACHER, C. *IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals*, RFC 4919, Elokuu 2007.

[77] LAKHINA, A., CROVELLA, M., JA DIOT, C. Characterization of Network-Wide Anomalies in Traffic Flows. Julkaisusarjassa *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement* (Taormina, Sicily, Italy, Lokakuu 2004), ACM, 201–206.

[78] LAKHINA, A., CROVELLA, M., JA DIOT, C. Diagnosing Network-Wide Traffic Anomalies. *ACM SIGCOMM Computer Communication Review 34*, 4 (2004), 219–230.

[79] LESHNO, M., LIN, V. Y., PINKUS, A., JA SCHOCKEN, S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks 6*, 6 (1993), 861–867.

[80] LIN, J., YU, W., ZHANG, N., YANG, X., ZHANG, H., JA ZHAO, W. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet of Things Journal 4*, 5 (2017), 1125–1142.

[81] LIU, F. T., TING, K. M., JA ZHOU, Z.-H. Isolation Forest. Julkaisusarjassa *2008 Eighth IEEE International Conference on Data Mining* (Pisa, Italy, Joulukuu 2008), IEEE, 413–422.

[82] LYNN, T., ENDO, P. T., RIBEIRO, A. M. N. C., BARBOSA, G. B. N., JA ROSATI, P. The Internet of Things: Definitions, Key Concepts, and Reference Architectures. Kirjassa *The Cloud-to-Thing Continuum: Opportunities and Challenges in Cloud, Fog and Edge Computing*, T. Lynn, J. G. Mooney, B. Lee, ja P. T. Endo, Eds. Springer Nature Switzerland AG, 2020, ch. 1, ss. 1–22.

[83] MACGREGOR, J. F., JA KOURTI, T. Statistical process control of multivariate processes. *Control Engineering Practice 3*, 3 (1995), 403–414.

[84] MAHALANOBIS, P. C. On the Generalised Distance in Statistics. Julkaisusarjassa *Proceedings of the National Institute of Sciences of India* (1936), National Institute of Science of India, 49–55.

[85] MAHONEY, M. V., JA CHAN, P. K. An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection. Julkaisusarjassa *International Workshop on Recent Advances in Intrusion Detection* (Pittsburgh, PA, USA, Syyskuu 2003), Springer, 220–237.

[86] MANUEL, J. Searching for the Reuse of Mirai Code: Hide 'N Seek Bot. URL https://www.fortinet.com/blog/threat-research/searching-for-the-reuse-of-mirai-code--hide--n-seek-bot, viitattu 14.2.2022.

[87] MEIDAN, Y., BOHADANA, M., MATHOV, Y., MIRSKY, Y., SHABTAI, A., BREITENBACHER, D., JA ELOVICI, Y. N-BaIoT–Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders. *IEEE Pervasive Computing 17*, 3 (2018), 12–22.

[88] MEKKI, K., BAJIC, E., CHAXEL, F., JA MEYER, F. Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT. Julkaisusarjassa *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)* (Athens, Greece, Maaliskuu 2018), IEEE, 197–202.

[89] MERRILL, N., JA ESKANDARIAN, A. Modified Autoencoder Training and Scoring for Robust Unsupervised Anomaly Detection in Deep Learning. *IEEE Access 8* (2020), 101824–101833.

[90] MICROSOFT. Azure IoT reference architecture. URL https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/iot, viitattu 14.2.2022.

[91] MICROSOFT. Control IoT devices using a voice assistant. URL https://docs.microsoft.com/en-us/azure/architecture/solution-ideas/articles/iot-controlling-devices-with-voice-assistant, viitattu 14.2.2022.

[92] MICROSOFT. Windows Sockets 2. URL https://docs.microsoft.com/en-us/windows/win32/winsock/windows-sockets-start-page-2, viitattu 14.2.2022.

[93] MOCKAPETRIS, P. *Domain Names – Implementation and Specification*, RFC 1035, Marraskuu 1987.

[94] MOUSTAFA, N., JA SLAY, J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). Julkaisusarjassa *2015 Military Communications and Information Systems Conference (MilCIS)* (Canberra, ACT, Australia, Marraskuu 2015), IEEE, 1–6.

[95] MOUSTAFA, N., JA SLAY, J. The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Information Security Journal: A Global Perspective 25*, 1–3 (2016), 18–31.

[96] NACHREINER, C. Anatomy of an ARP Poisoning Attack. URL `https://web.archive.org/web/20130321005829/http://www.watchguard.com/infocenter/editorial/135324.asp`, viitattu 14.2.2022.

[97] NARTEN, T., NORDMARK, E., SIMPSON, W. A., JA SOLIMAN, H. *Neighbor Discovery for IP version 6 (IPv6)*, RFC 4861, Syyskuu 2007.

[98] OLIN, P., KOIVUNIEMI, M., LEHTO, M., LUUKKAINEN, K., MAGD, N., NEVASTE, N., NIINIKORPI, S., RAUTIO, J., RISTOLAINEN, M., SJÖROOS, M., TUOVINEN, J., KOUKI, P., JA SUHONEN, S. *Kyberturvallisuuden sanasto*. Sanastokeskus TSK ry, Helsinki, 2018.

[99] ORDWAY-WEST, E., PARVEEN, P., JA HENSLEE, A. Autoencoder Evaluation and Hyper-Parameter Tuning in an Unsupervised Setting. Julkaisusarjassa *2018 IEEE International Congress on Big Data (BigData Congress)* (San Francisco, CA, USA, Heinäkuu 2018), IEEE, 205–209.

[100] OSANAIYE, O., CHOO, K.-K. R., JA DLODLO, M. Distributed denial of service (DDoS) resilience in cloud: Review and conceptual cloud DDoS mitigation framework. *Journal of Network and Computer Applications 67* (2016), 147–165.

[101] O'SEARCOID, M. *Metric Spaces*. Springer Science & Business Media, London, 2007.

[102] OSSEC. About OSSEC HIDS. URL `https://www.ossec.net/about/`, viitattu 14.2.2022.

[103] OTHMAN, S. M., ALSOHYBE, N. T., BA-ALWI, F. M., JA ZAHARY, A. T. Survey on Intrusion Detection System Types. *International Journal of Cyber-Security and Digital Forensics 7*, 4 (2018), 444–463.

[104] PAGÈS, J. Analyse factorielle de données mixtes. *Revue de Statistique Appliquée 52*, 4 (2004), 93–111.

[105] PAGÈS, J. *Multiple Factor Analysis by Example Using R*. CRC Press, Boca Raton, FL, USA, 2015.

[106] PETERSEN, K. B., JA PEDERSEN, M. S. *The Matrix Cookbook*. Tekninen raportti, Technical University of Denmark, Lyngby, Denmark, Marraskuu 2012.

[107] PINCOMBE, B. Anomaly Detection in Time Series of Graphs using ARMA Processes. *Asor Bulletin 24*, 4 (2005).

[108] PISCITELLO, D. M., JA CHAPIN, A. L. *Open systems networking: TCP/IP and OSI*. Addison-Wesley Publishing Company, Reading, MA, USA, 1993.

[109] PLUMMER, D. C. *An Ethernet Address Resolution Protocol*, RFC 826, Marraskuu 1982.

[110] POPLI, S., JHA, R. K., JA JAIN, S. A Survey on Energy Efficient Narrowband Internet of Things (NBIoT): Architecture, Application and Challenges. *IEEE Access 7* (2019), 16739–16776.

[111] POSTEL, J. *User Datagram Protocol*, RFC 768, Elokuu 1980.

[112] POSTEL, J. *Internet Protocol*, RFC 791, Syyskuu 1981.

[113] POSTEL, J. *Transmission Control Protocol*, RFC 793, Syyskuu 1981.

[114] PUNTANEN, S., STYAN, G. P. H., JA ISOTALO, J. *Matrix Tricks for Linear Statistical Models: Our Personal Top Twenty*. Springer, Berlin, Heidelberg, 2011.

[115] RAO, C. R., JA MITRA, S. K. Generalized inverse of a matrix and its applications. Julkaisusarjassa *Berkeley Symposium on Mathematical Statistics and Probability* (1972), vol. 6.1, University of California Press, Berkeley, 601–620.

[116] REKHTER, Y., MOSKOWITZ, R. G., KARRENBERG, D., DE GROOT, G. J., JA LEAR, E. *Address Allocation for Private Internets*, RFC 1918, Helmikuu 1996.

[117] RESCORLA, E. *Diffie-Hellman Key Agreement Method*, RFC 2631, Kesäkuu 1999.

[118] RESCORLA, E. *The Transport Layer Security (TLS) Protocol Version 1.3*, RFC 8446, Elokuu 2018.

[119] RUSSELL, A. L. The internet that wasn't. *IEEE Spectrum 50*, 8 (2013), 39–43.

[120] SCARFONE, K., JA MELL, P. *Guide to Intrusion Detection and Prevention Systems (IDPS)*. Tekninen raportti NIST Special Publication (SP) 800-94, National Institute of Standards and Technology, Gaithersburg, MD, USA, Helmikuu 2007.

[121] SERFLING, R. Equivariance and invariance properties of multivariate quantile and related functions, and the role of standardisation. *Journal of Nonparametric Statistics 22*, 7 (2010), 915–936.

[122] SETHI, M., SARIKAYA, B., JA GARCIA-CARRILLO, D. Secure IoT Bootstrapping: A Survey. URL `https://datatracker.ietf.org/doc/html/draft-sarikaya-t2trg-sbootstrapping-11`, viitattu 14.2.2022.

[123] SHANNON, C. E. A mathematical theory of communication. *The Bell System Technical Journal 27*, 3 (1948), 379–423.

[124] SHELBY, Z., HARTKE, K., JA BORMANN, C. *The Constrained Application Protocol (CoAP)*, RFC 7252, Kesäkuu 2014.

[125] SHIREY, R. W. *Internet Security Glossary, Version 2*, RFC 4949, Elokuu 2007.

[126] SHYU, M.-L., CHEN, S.-C., SARINNAPAKORN, K., JA CHANG, L. A Novel Anomaly Detection Scheme Based on Principal Component Classifier. Julkaisusarjassa *Proceedings of the ICDM Foundation and New Direction of Data Mining workshop* (2003), 172–179.

[127] SIKIMIĆ, M., AMOVIĆ, M., VUJOVIĆ, V., SUKNOVIĆ, B., JA MANJAK, D. An Overview of Wireless Technologies for IoT Network. Julkaisusarjassa *2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH)* (East Sarajevo, Bosnia and Herzegovina, Maaliskuu 2020), IEEE, 1–6.

[128] SNORT. Snort – Network Intrusion Detection & Prevention System. URL `https://www.snort.org/`, viitattu 14.2.2022.

[129] SOMMER, P., MARET, Y., JA DZUNG, D. Low-Power Wide-Area Networks for Industrial Sensing Applications. Julkaisusarjassa *2018 IEEE International Conference on Industrial Internet (ICII)* (Seattle, WA, USA, Lokakuu 2018), IEEE, 23–32.

[130] SOMMER, R., JA PAXSON, V. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. Julkaisusarjassa *2010 IEEE Symposium on Security and Privacy* (Oakland, CA, USA, Toukokuu 2010), IEEE, 305–316.

[131] SRISURESH, P., JA HOLDREGE, M. *IP Network Address Translator (NAT) Terminology and Considerations*, RFC 2663, Elokuu 1999.

[132] SULTANA, T., JA WAHID, K. A. Choice of Application Layer Protocols for Next Generation Video Surveillance Using Internet of Video Things. *IEEE Access 7* (2019), 41607–41624.

[133] TAN, P.-N., STEINBACH, M., KARPATNE, A., JA KUMAR, V. *Introduction to Data Mining*, 2nd ed. Pearson Education, Inc., New York, NY, USA, 2019.

[134] TAVALLAEE, M., BAGHERI, E., LU, W., JA GHORBANI, A. A. A detailed analysis of the KDD CUP 99 data set. Julkaisusarjassa *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications* (Ottawa, ON, Canada, Heinäkuu 2009), IEEE, 1–6.

[135] TEIXEIRA, F. A., PEREIRA, F. M. Q., WONG, H.-C., NOGUEIRA, J. M. S., JA OLIVEIRA, L. B. SIoT: Securing Internet of Things through distributed systems analysis. *Future Generation Computer Systems 92* (2019), 1172–1186.

[136] THE OWASP FOUNDATION. OWASP Top 10 – 2017. URL `https://raw.githubusercontent.com/OWASP/Top10/master/2017/OWASP%20Top%2010-2017%20(en).pdf`, viitattu 14.2.2022.

[137] THE OWASP IOT SECURITY TEAM. 2018 OWASP IoT Top 10. URL `https://owasp.org/www-pdf-archive/OWASP-IoT-Top-10-2018-final.pdf`, viitattu 14.2.2022.

[138] THE ZEEK PROJECT. About Zeek – Book of Zeek (git/master). URL `https://docs.zeek.org/en/master/about.html`, viitattu 14.2.2022.

[139] THE ZEEK PROJECT. base/protocols/conn/main.zeek – Book of Zeek (git/master). URL `https://docs.zeek.org/en/master/scripts/base/protocols/conn/main.zeek.html`, viitattu 14.2.2022.

[140] THE ZEEK PROJECT. The Zeek Network Security Monitor. URL `https://zeek.org/`, viitattu 14.2.2022.

[141] VANHOEYVELD, J., JA MARTENS, D. *Towards a scalable anomaly detection with pseudo-optimal hyperparameters*. Tekninen raportti 2018-012, Faculty of Business and Economics, University of Antwerp, Antwerp, Belgium, Lokakuu 2018.

[142] VÉSTIAS, M. P. Convolutional Neural Network. Kirjassa *Research Anthology on Artificial Neural Network Applications*, Management Association, Information Resources, Ed. IGI Global, 2021, ch. 77, ss. 1559–1575.

[143] WEINSTEIN, R. RFID: a technical overview and its application to the enterprise. *IT Professional 7*, 3 (2005), 27–33.

[144] WESTFALL, P. H. Kurtosis as Peakedness, 1905–2014. R.I.P. *The American Statistician 68*, 3 (2014), 191–195.

[145] WHITMORE, A., AGARWAL, A., JA DA XU, L. The Internet of Things–A survey of topics and trends. *Information Systems Frontiers 17*, 2 (2015), 261–274.

[146] WU, Q., JA SHAO, Z. Network Anomaly Detection Using Time Series Analysis. Julkaisusarjassa *Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services – (icas-isns'05)* (Papeete, France, Lokakuu 2005), IEEE, 42–42.

[147] XU, W., JANG-JACCARD, J., SINGH, A., WEI, Y., JA SABRINA, F. Improving Performance of Autoencoder-Based Network Anomaly Detection on NSL-KDD Dataset. *IEEE Access 9* (2021), 140136–140146.

[148] YADAV, S., REDDY, A. K. K., REDDY, A. L. N., JA RANJAN, S. Detecting Algorithmically Generated Malicious Domain Names. Julkaisusarjassa *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement* (Melbourne, Australia, Marraskuu 2010), ACM, 48–61.

[149] ZHAO, K., JA GE, L. A Survey on the Internet of Things Security. Julkaisusarjassa *2013 Ninth International Conference on Computational Intelligence and Security* (Emeishan, China, Joulukuu 2013), IEEE, 663–667.

[150] ZHU, X. *Semi-Supervised Learning Literature Survey*. Tekninen raportti TR1530, University of Wisconsin-Madison Department of Computer Sciences, Madison, WI, USA, Syyskuu 2005.