Koneoppimispohjaiset poikkeamien havainnointimenetelmät suoratoistodatassa

Turun yliopisto
Tietotekniikan laitos
TkK-tutkielma
Teknillinen tiedekunta
Helmikuu 2026
Jasmin Kaseva

TURUN YLIOPISTO
Tietotekniikan laitos

Jasmin Kaseva: Koneoppimispohjaiset poikkeamien havainnointimenetelmät suoratoistodatassa

TkK-tutkielma, 25 s.
Teknillinen tiedekunta
Helmikuu 2026

Suoratoistavien sovellusten määrä kasvaa, ja samalla niiden datan määrä lisääntyy. Poikkeamien havainnointi (engl. Anomaly detection, AD) vaatii yhä useammin koneoppimista menetelmien taustalla suoriutumaan suoratoistodatan haasteista, kuten reaaliaikasuudesta ja moniulotteisuudesta. Tämä tutkielma on toteutettu kirjallisuuskatsauksena, jossa käydään läpi vuosina 2021-2025 julkaistua aihetta käsittelevää aineistoa. Tarkastelun kohteena on suoratoiston sovellusalueen merkitys AD-menetelmää valittaessa, sekä aiheen tulevaisuuden tutkimussuunnat.
Tutkielman perusteella AD-menetelmät soveltuvat eri tavoin eri sovellusalueisiin, ja sovellusalueen merkitys menetelmää valitessa tulee kasvamaan. Soveltuvuuteen vaikuttaa koneoppiminen AD-menetelmän taustalla, suoratoistodatan ominaisuudet ja kontekstisidonnaisuus, sekä AD:n tarpeet ylipäätään. Tulevaisuudessa kontekstisidonnaisuus tulee todennäköisesti kasvamaan, ja saatavilla oleva opetusdata tulee vaikuttamaan algoritmien ja menetelmien kehitykseen. Samaan aikaan tarvitaan yhtenäisyyttä menetelmien tutkimiseen, jotta saadaan vertailukelpoisia tuloksia tulevaisuuden kirjallisuuskatsauksiin.

Asiasanat: syväoppiminen, verkossa oppiminen, poikkeama, datan mallintaminen, suoratoisto

Sisällys

1 Johdanto

Poikkeamien havainnointi (engl. Anomaly detection, AD) on kaikenlaisessa datankäsittelyssä kriittinen osa-alue [1]. *Poikkeama* (engl. anomaly tai outlier) nimensä mukaisesti tarkoittaa normaalista poikkeavaa tapahtumaa tai arvoa, joiden nopea ja tehokas havainnointi muun muassa ennaltaehkäisee virhetilanteita ja nopeuttaa päätöksentekoa [2].

Kyberturvallisuudessa esimerkiksi yksityisyyden suojaaminen ja haittaohjelmien havainnointi helpottuvat reaaliaikaisella poikkeamien havainnoinnilla [3]. Finanssialalla poikkeamat voivat liittyä esimerkiksi pankkikorttihuijauksiin, konkurssin ennakointiin tai sijoittamisen riskeihin [4]. Erityisen tärkeitä sovelluskohteita ovat myös esimerkiksi terveysalalla diagnostiikka, potilaan monitorointi ja radiologia. Teollisuudessa AD aikaisessa vaiheessa on tärkeää valmistamisen laadun kannalta. [1][5]

Erilaisten dataa suoratoistavien sovellusten yleistyessä datan määrä kasvaa ja samalla sen käsittely hankaloituu. Verkkoliikenne, *asioiden internet* (engl. Internet of Things, IoT), teolliset ja autonomiset systeemit ovat vain muutama esimerkki sovelluskohteista, joissa poikkeamia havainnoidaan jatkuvasta datavirrasta. Suoratoiston tuomat haasteet, kuten rajallinen muistikapasiteetti ja reaaliaikaisuus [6], on vaatinut yhä enemmän koneoppimisen hyödyntämistä AD:ssä.

Tässä tutkimuksessa tutkitaan AD-menetelmiä suoratoistodatassa, keskittyen erityisesti koneoppimispohjaisten menetelmien näkökulmaan. Tarkoituksena on selvittää, miten koneoppimista on hyödynnetty ja miten sen avulla on pystytty kehit-

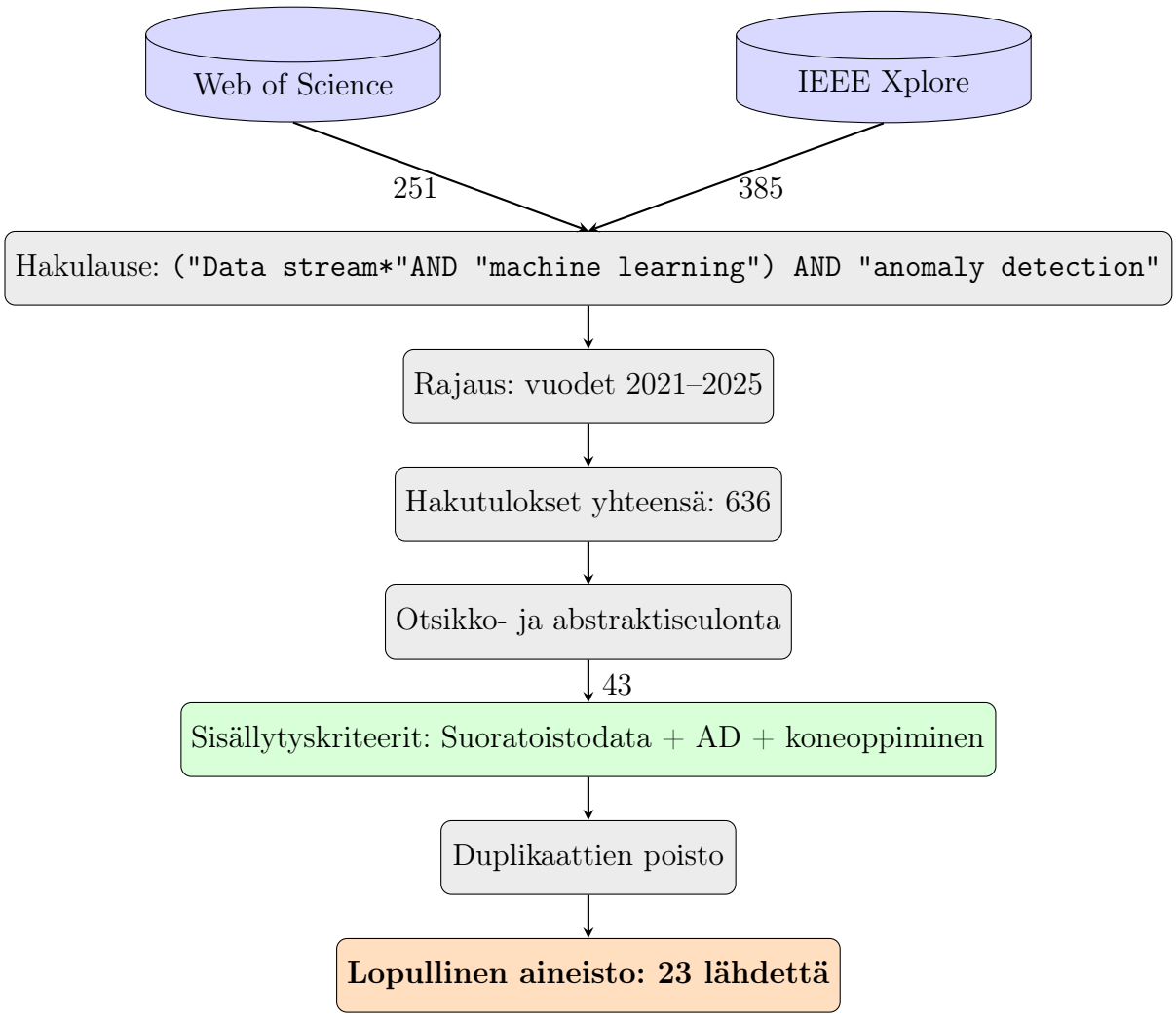
tämään AD-menetelmiä. Samalla selvitetään miten suoratoistodata eri sovelluskohteissa vaikuttaa menetelmän valintaan, ja mihin suuntaan tutkimus ja kehitys on aiheen osalta matkalla. Tämän työn tutkimuskysymykset ovat:

TK1. Miten suoratoistodatan sovellusalue vaikuttaa AD-menetelmän valintaan?

TK2. Mitä tulevaisuuden kehityssuuntia on havaittavissa kirjallisuuskatsauksen aineiston pohjalta?

Kirjallisuuskatsaukseen kerätty aineisto on haettu Web of Science sekä IEEE tietokannoista hakulauseella: `("Data stream*"AND "machine learning") AND "anomaly detection"` kuvan 1.1 mukaisesti. Suuren hakutulosmäärän takia tulokset on rajattu vuosiin 2021-2025, jolloin saadaan myös katsaus viimeisimmistä tutkimuksista aiheen tiimoilta. Näillä parametreilla lähteitä saatiin 636 kappaletta, jotka käytiin otsikko- ja abstraktitasolla läpi. 43 lähdettä luettiin kokonaisuudessaan läpi, ja näistä työhön valittiin mukaan aiheeseen sopivia konferenssijulkaisuja ja artikkeleita, jotka käsittelivät nimenomaan suoratoistodatan AD-menetelmiä ja koneoppimisen hyödyntämistä. Mukaan ei otettu sellaisia aineistoja, jotka eivät käsitelleet kaikkia kolmea teemaa. Duplikaattien poistamisen jälkeen tutkimusaineiston lopullinen määrä on 23 lähdettä.

Tutkielma etenee seuraavasti: Luvussa 2 taustoitetaan suoratoistodataa ja sen tuomia haasteita AD:en, sekä koneoppimisen tarpeellisuutta ja kykyjä ratkaista haasteiden tuomia ongelmia. Luvussa 3 analysoidaan tutkimusaineistoa, ja etsitään vastauksia edellä esitettyihin tutkimuskysymyksiin. Lopuksi 4 luvussa pohditaan työn tuloksia ja luvussa 5 esitetään työn yhteenveto ja vastataan työn tutkimuskysymyksiin.



Kuva 1.1: Kirjallisuuskatsauksen aineistonhakuprosessi

2 Taustoitus

Tässä luvussa taustoitetaan työn kannalta olennaiset termit ja käsitteet, joiden jälkeen luvussa 3 siirrytään itse tutkimuksen pariin. Aluksi esitellään suoratoistodata, sen merkitys ja ominaisuudet. Tämän jälkeen tarkastellaan poikkeamia; niiden määrittelyä, ominaisuuksia sekä havainnointia. Seuraavaksi esitellään suoratoiston tuomat haasteet AD:en, ja sen jälkeen taustoitetaan koneoppimisen merkitystä uusien algoritmien kehityksessä. Lopuksi on vielä koko taustan yhteenveto.

2.1 Suoratoistodata

Suoratoistodata on tyypillisesti jatkuvasti päivittyvää datavirtaa, joka voi olla monenlaista verkkoliikenteestä sensoridataan [6]. Ero staattiseen dataan on se, että data voi olla reaaliaikaista, ja usein dataa on niin paljon, ettei kaikkea pysty tallentamaan käsittelyä varten. Eräs tapa mallintaa suoratoistodataa on massiivinen ja dynaaminen yksiulotteinen vektori, jossa jokainen ulottuvuus vastaa jotakin havaittavaa suuretta [7]. Suoratoistodata on kuitenkin usein luonteeltaan moniulotteista, jolloin datalla on useampia eri ominaisuuksia [8].

Suoratoistossa on muutama eri tapa päivittää arvoja; *aikasarjamallissa* (engl. time-series model) jokainen uusi arvo on *seuraava arvo* eikä menneisyyttä voi muuttaa. Näin toimii esimerkiksi sensoridata sekunti sekunnilta. *Kassakonemalli* (engl. cash register model) sallii arvon lisäyksen, mutta ei poistamista. Tämä malli seuraa esimerkiksi IP-pakettien määrää siirtymissä, tai sitä kuinka monta kertaa web-

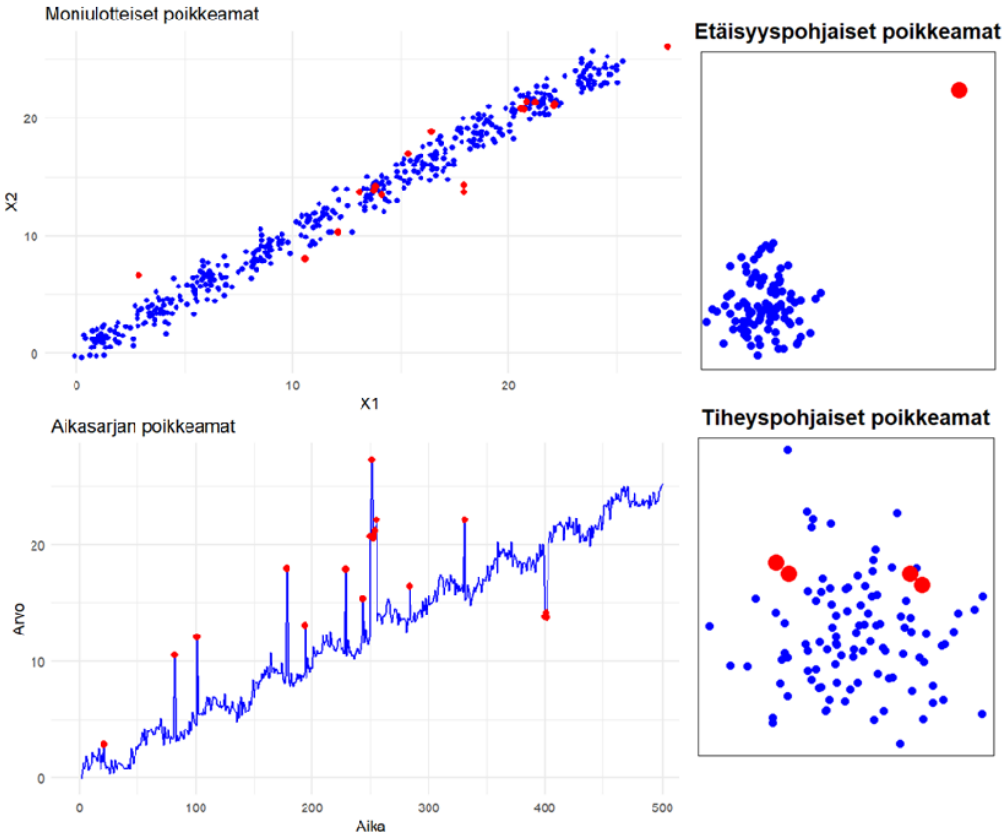
sivua on ladattu. Kolmas malli on *kääntöporttimalli* (engl. turnstile model) jossa sekä arvon lisäys että poistaminen on sallittua. Tämä on joustavin ja yleisin tapa päivittää suoratoistodataa, ja sen avulla voidaan esimerkiksi seurata IP-yhteyksien aktiivisuutta (sulkeminen = poisto, avaus = lisäys jne.) [7]

2.2 Poikkeamat ja niiden havainnointi

Poikkeamat ovat normaalin suhteen huomattavat variaatiot. Määritelläksemme poikkeaman jossakin systeemissä, on ensin määriteltävä sen normaali tila tai käyttäytyminen. Normaalista tilasta voidaan havainnoida dataa, jonka avulla systeemiä kuvaillaan. [1] Keskeinen ongelma poikkeamien havainnoinnissa on se, ettei ole olemassa yksinkertaista määritelmää, jonka avulla voitaisiin arvioida, kuinka samanlaiset tai erilaiset jotkin datapisteet ovat toisistaan [5]. Täytyy myös ymmärtää minkälaisia poikkeamia datassa on läsnä, tämä auttaa selittämään dataa, ja mahdollistaa uuden tiedon kehitystä [9].

Mutta miten määritellään normaali data? Perinteisessä tilastollisessa ajattelussa normaalia kuvataan usein jakauman keskeisten tunnuslukujen aritmeettisen keskiarvon, mediaanin ja moodin avulla. Epänormaalin tai multimodaalisen jakauman kohdalla, kuten esimerkiksi suoratoistodatassa, tämä määritelmä ei enää päde. Normia kuvataankin silloin pisteiden joukkona yhden pisteen sijaan. [5]

Suoratoistodatassa AD on usein joko äkillisen muutoksen havaitsemista aikasarjoista tai ns. uutuuden ja muutoksen havaitsemista moniulotteisesta datasta. Esimerkiksi aikasarjadatassa poikkeama voi olla kontekstuaalinen tai kollektiivinen, riippuen siitä onko havainto tietyllä ajanhetkellä kokenut yhtäkkisen muutoksen, vai onko datan malli muuttunut tietyllä aikajaksolla [8]. Tilannetta, jossa syötetyn datan ja mallin välinen suhde muuttuu ajan myötä kutsutaan *konseptimuutokseksi* (engl. concept drift) [10].



Kuva 2.1: Erilaisten poikkeamien mallinnus simuloidulla datalla

Kuvassa 2.1 on esitetty esimerkki erilaisten poikkeamien visualisoinnista. Esimerkkiä varten on annettu OpenAI GPT-5.2 kielimallille kehoite luoda yksinkertainen RStudiossa käsiteltävä datasetti ja skriptit. Näin yksittäisistä poikkeamista on saatu kuvat, jotka on koottu yhteen. Poikkeamat näkyvät erilaisen datan joukossa punaisina pisteinä, ja monet poikkeamat sijaitsevat suhteellisen kaukana muista pisteistä tai tiheistä alueista. Esimerkiksi aikasarjassa selkeät piikit muuhun dataan verrattuna voidaan kokea poikkeavina ja etäisinä, kun taas moniulotteisessa datassa etäisyyden lisäksi alueen tiheys vaikuttaa poikkeaman luokitteluun.

2.3 AD-menetelmien toimintaperiaatteet

AD-menetelmät voidaan luokitella niiden toimintaperiaatteen perusteella esimerkiksi läheisyys- (etäisyys, klusterointi ja tiheyspohjaiset), mallipohjaisiin-, eristämis- ja

hybridimenetelmiin. [1]. Ei ole kuitenkaan yhtä ainoaa tapaa luokitella algoritmeja ja menetelmiä.

Klusterointipohjaisissa menetelmissä keskitytään klustereiden rajalla tai ulkopuolella olevien poikkeamien havaitsemiseen datasta [5]. Pisteet, jotka eivät sijaitse klusterissa voivat mahdollisesti olla poikkeavia. Tunnettu menetelmä on k-means algoritmi. Etäisyyteen pohjautuvat menetelmät perustuvat lähimmän naapurin etäisyyksiin. Tällöin poikkeavina pidetään suuria etäisyyksiä lähimpiin naapureihin. Tunnettuja etäisyyteen perustuvia algoritmeja ovat k-NN (engl. k Nearest Neighbour) ja r-NN (engl. r Nearest Neighbour). Tiheyteen perustuvissa menetelmissä otetaan etäisyyden lisäksi huomioon pisteiden jakauman tiheys, jolloin poikkeamia tarkastellaan suhteessa datapisteiden tai klustereiden yleiseen tiheyteen. Tunnettu tiheyspohjainen algoritmi on LOF (engl. Local Outlier Factor). [11]

Mallipohjaisissa lähestymistavoissa dataa kuvaavan mallin avulla voidaan ymmärtää ja tarkasti kuvailla datasettiä. Tämä mahdollistaa sellaisten datapisteiden tunnistamisen, jotka eivät ole yhtenäisiä kuvailun kanssa [5]. Eristämiseen perustuvat ja puu-menetelmät pohjautuvat rekursiiviseen partitiointiin, jossa erilaiset tapaukset eristetään toisistaan. Tällöin poikkeamia sisältävät *puun oksat* ovat selkeästi lyhyempiä, koska datapisteet ovat erilaisia normaaliin dataan verrattuna, ja voivat näin ollen mahdollisesti olla poikkeamia. Yksi yleisimpiä eristämiseen perustuvia menetelmiä on Isolation Forest. [12]

Aikasarjoihin ja moniulotteiseen suoratoistodataan on edellä mainittujen menetelmien pohjalta luotu monimutkaisempia algoritmeja. Esimerkiksi kNN ja LOF menetelmistä on luotu suoratoistoon paranneltuja versioita tai hybridimenetelmiä.

2.4 Suoratoistodatan haasteet

Vaikka suoratoistodataa voidaan joissain sovelluksissa tallentaa jälkikäsittelyyn, useimpien lähestymistapojen soveltaminen suoratoistoon ei kuitenkaan ole yksin-

kertaista, koska ne usein vaativat pääsyn koko datajoukkoon [8]. AD-tekniikoiden on oltava tarpeeksi tehokkaita napatakseen pienetkin määrät poikkeamia suuresta datamäärästä, sekä pitkillä että lyhyillä aikaväleillä, aikasarjoissa ja moniulotteisessa datassa.

Suoratoistossa ja erityisen suurissa datamäärissä tyypillinen tilastollinen poikkeavuuksien havaitseminen ei ole toimivaa. Siinä kärsii sekä tarkkuus että nopeus, koska tämä vaatisi kohtuuttoman määrän muistia ja prosessointia kaiken datan läpikäymiseen ja vertailuun [7]. Lisäksi reealiaikainen AD moniulotteisessa virtaavassa datassa on käytännössä epäkäytännöllistä toteuttaa perinteisten tilastollisten menetelmien avulla [8]. Tästä syystä on kehitetty erilaisia koneoppimista hyödyntäviä menetelmiä, jotka pystyvät sovelluskohteesta riippuen löytämään poikkeamia hyvin nopeasti tai jopa reaaliajassa. Algoritmin tehokas ja oikeanlainen hyödyntäminen parantaa turvallisuutta, vähentää riskejä, edesauttaa älykästä päätöksentekoa ja tuottaa säästöjä, kun taas väärinkäyttö voi johtaa päinvastaisiin vaikutuksiin [13].

2.5 Koneoppimisen menetelmien hyödyntäminen

Tässä työssä koneoppimismenetelmillä viitataan yleisiin oppimismalleihin, kuten neuroverkkoihin, klusterointimenetelmiin ja puupohjaisiin algoritmeihin. AD-menetelmällä tarkoitetaan sellaista menetelmää tai järjestelmää, jonka tavoitteena on tunnistaa poikkeamia datasta. Joissakin tapauksissa koneoppimismenetelmää voidaan käyttää AD:en, jolloin se toimii samalla myös AD-menetelmänä.

Koneoppiminen AD-menetelmien taustalla vaikuttaa erityisesti siihen, miten nopeasti datasta saadaan luotua malli eli miten nopeasti data opitaan. Näin ollen AD suoratoistodatasta edellyttää koneoppimisen hyödyntämistä [1]. Koneoppiminen on työkalu jota käytetään nopeuttamaan ja automatisoimaan datasta oppimista. Koneoppimisalgoritmit voivat oppia datasta käyttämällä tilastollisia menetelmiä, ilman että niitä ohjelmoidaan täsmällisesti [14]. Erityisesti kun käytetty data on monimut-

kaista ja dynaamista, myös tarvittavat koneoppimismenetelmät monimutkaistuvat. Esimerkiksi ohjattua oppimista, jossa datan alku- ja loppuarvot ovat tiedossa, voi olla vaikeaa hyödyntää suoratoiston AD:ssä ainakaan itsenäisesti [10]. Lisäksi valmiiksi merkittyä data, jossa poikkeamat ovat jo tiedossa tai määritelty, on todella kallista ja vaatii yleensä ihmisen panosta [9].

Ohjaamattomassa oppimisessa sen sijaan arvot tai muuttujat eivät ole etukäteen tiedossa [10]. Tämä on tyypillinen piirre muun muassa suoratoistodatalle, joka voi myös muuttaa muotoaan ajan myötä. Esimerkiksi klusterointimenetelmät, joiden tavoitteet ovat samat kuin ohjatun oppimisen luokittelulla, muodostavat klustereita datasta opitun mallin perusteella. AD voidaankin nähdä luokitteluongelman variaationa, jossa luokitellaan *normaali* ja *poikkeama* [1].

Verkossa oppiminen (engl. online learning) kuvaa mallin päivitystapaa. Tällöin malli oppii jatkuvasti datavirrasta [15], mutta oppiminen voi silti olla ohjattua tai ohjaamatonta mallista riippuen. Malli pyrkii oppimaan ja päivittämään parhaan ennustajan tulevaisuudelle jokaisessa vaiheessa [16]. Verkossa oppivat mallit ovat erityisen soveltuvia esimerkiksi jatkuviin datavirtoihin.

Syväoppiminen keskittyy neuroverkkoihin, ja algoritmeihn, joiden avulla koulutetaan neuroverkkoja kuten *LSTM* (engl. Long Short Term Memory) tai *CNN*:ää (engl. Convolutional Neural Network). Neuroverkko koostuu toisiinsa linkittyneistä neuroneista, jotka toimivat prosessointiyksikköinä [10]. Neuronien väliset yhteydet lähettävät signaalia, joka mukautuu oppimisprosessin määrittämän painotuksen mukaan. Yksinkertaisimmillaan neuroverkossa on yksi kerros, mutta niiden määrä voi vaihdella. Syväoppimisessa voidaan hyödyntää ohjattua tai ohjaamatonta oppimista, jolloin datasta luodaan neuroverkko, jossa on enemmän kuin yksi piilotettu kerros [10]. Syväoppiminen on erityisen tehokas moniulotteisen ja suuren datan kanssa, jonka takia se suoriutuu hyvin suoratoistodatan haasteista.

2.6 Taustan yhteenveto

Yhteenvetona voidaan todeta, että monimutkaisen suoratoistodatan määrän kasvaessa, AD:n tärkeys korostuu. Poikkeaman erottelu normaalista datasta on välttämätöntä muun muassa kyberhyökkäysten, laitevikojen tai muiden kriittisten tilanteiden reaaliaikaisessa tai ennakoivassa havaitsemisessa. Myös sen ymmärtäminen mikä on poikkeavaa on tärkeää, etenkin kun kyse on suoratoistodatan kaltaisesta ympäristöstä.

Poikkeamia voidaan havaita monesta eri näkökulmasta esimerkiksi etäisyyden tai klusteroinnin avulla. AD-menetelmien erilaiset toimintaperiaatteet tuovat esiin sekä vahvuuksiaan, että rajoitteitaan, jotka korostuvat erityisen herkästi suoratoistodatan kohdalla. Näin ollen monet nykyaikaiset AD-menetelmät ovat menetelmien yhdistelmiä tai mukautettuja versioita.

Suoratoistodatan ominaisuudet kuten reaaliaikaisuus, korkeauloitteisuus ja rajoitettu muistikapasiteetti, asettavat AD-menetelmille monia haasteita. Näiden haasteiden takia perinteiset tilastolliset AD-menetelmät eivät riitä, ja koneoppiminen on noussut AD-menetelmien kehityksessä vahvaan rooliin. Koneoppimisen avulla datan mallintaminen helpottuu ja nopeutuu, jolloin poikkeamien määrittely datasta selkeytyy, ja AD-menetelmien suorituskyky suoratoistodatassa tehostuu.

Taustassa esiteltyjä konsepteja tullaan hyödyntämään seuraavissa luvuissa kirjallisuuskatsauksen aineiston tarkastelussa ja tulkinnassa. Tutkimusaineistossa esiintyneitä AD-menetelmiä on jaoteltu luvussa 3.2 oppimistapojen ja toimintaperiaatteiden mukaan, ja esitelty tarkemmin taulukossa 3.3.

3 Poikkeamien havainnointi suoratoistodatassa

Kirjallisuuskatsaukseen on otettu mukaan yhteensä 23 vuosina 2021-2025 julkaistua lähdettä, jotka käsittelevät koneoppimista suoratoistodatan AD-menetelmissä. Aineistossa on mukana vertailevia tutkimuksia, artikkeleita ja konferenssijulkaisuja joissa arvioidaan jo olemassa olevien AD-menetelmien suorituskykyä suoratoistodatassa, sekä tutkimuksia, joissa vertaillaan itse kehitettyä menetelmää olemassa oleviin menetelmiin.

Seuraavaksi esitellään aineistossa esiintyneet sovellusalueet, sekä aineistossa sovellusalueittain käytettyä dataa ja datasettejä. Sen jälkeen tarkastellaan aineistoissa esiintyviä AD-menetelmiä, vertaillaan aineistoissa saatuja tutkimustuloksia, ja tutkitaan niiden kontekstisidonnaisuutta. Tavoitteena on vertailla aineistossa saatuja tuloksia toisiinsa, ja perustella sovellusalueen merkitys AD-menetelmän valinnassa, sekä aiheen tulevaisuuden tutkimussuunnat.

3.1 AD:n sovellusalueet ja datasetit

Tutkimuksessa esiin tulleet sovellusalueet olivat asioiden internet, teollinen asioiden internet (engl. Industrial Internet of Things, IIoT), kyberturvallisuus, verkkoliikenne, blockchain, sensorit, teollisuus, ajalliset prosessit ja yleiskäyttöiset tutkimusympäristöt. Tutkimukset keskittyivät pitkälti johonkin tiettyyn sovellusalueeseen, jonka

Taulukko 3.1: Katsauksen tutkimukset sovelluskohteittain

Lähde	IoT	IIoT	Kyberturva	Verkkoliikenne	Blockchain	Sensorit	Teollisuus	Ajalliset prosessit	Yleiskäyttöiset
Agyemang (2024) [17]									x
Al-Amri et al. (2021) [18]	x								
Belacel et al. (2022) [19]									x
Choi et al. (2021) [20]				x		x			
Diaz Rivera et al. (2021) [21]				x					
Duraj et al. (2025) [22]									x
Fosic et al. (2023) [23]				x					
Gerz et al. (2022) [24]		x							
Gomes et al. (2025) [25]									x
Gong (2023) [26]								x	
Jain et al. (2022) [27]			x						
Kim et al. (2022) [28]					x				
Lei et al. (2023) [29]									x
Liu et al. (2021) [30]						x	x	x	
Maleki (2021) [31]	x								
Nixon et al. (2024) [32]			x						
Ray et al. (2024) [33]	x								
Ribeiro et al. (2021) [34]			x						
Suryadevara et al. (2025) [35]						x			
Togbe et al. (2021) [36]									x
Velásquez et al. (2022) [37]							x		
Yahyaoui et al. (2021) [38]	x								
Yang et al. (2023) [39]		x							

sisällä AD-menetelmien vertailua tehtiin. Taulukko 3.1 esittelee aineiston jakautumista eri sovellusalueille.

IoT ja IIoT koostuu sensori- ohjain, ja prosessoridatasta, jotka ovat keskenään yhteydessä. Dataa virtaa jatkuvalla syötöllä suuria määriä, ja tyypillisesti data on korkeaulotteista, meluisaa ja joskus osittain puutteellista. Poikkeamien tunnistaminen on tärkeää laitevikojen, kyberuhkien ja suorituskyvyn heikkenemisen tunnistamisessa. [18], [31]

Kyberturvallisuudessa ja verkkoliikenteessä poikkeamat ovat huomattavasti normaalia dataa harvinaisempia. Poikkeama voi olla kyberhyökkäys tai muu luvaton toiminta, mutta toisaalta myös muutos verkkoliikenteen datan luonteessa ajan myötä. Usein kyberturvan kannalta väärä hälytys (false positive) on pienempi paha kuin huomaamatta jäänyt poikkeama (false negative). [27], [32]

Teollisissa järjestelmissä ja ajallisissa prosesseissa poikkeamia havaitaan prosessien valvonnan, ennakoivan huollon tai turvallisuuden valvonnan yhteydessä. Näiden lisäksi myös sensorit ovat vahvasti aikasidonnaisia, jolloin data on luonteeltaan jatkuvaa ja ajallisesti riippuvaa. Virhemarginaali on myös muihin sovellusalueihin verrattuna hieman pienempi. [26], [30]

Yleiskäyttöiset tutkimusympäristöt keskittyvät tutkimaan menetelmiä teoreettisen taustan kontekstissa selkeän sovellusalueen sijaan. Tämä mahdollistaa eri menetelmien vertailun, mutta niissä ei juuri oteta huomioon tosielämän monimutkaisuutta tai kontekstisidonnaisuutta. [25], [36]

Katsauksen tutkimukset käyttivät laajaa valikoimaa simuloituja tai todellisia datasettejä algoritmien koulutukseen ja testaukseen. Taulukossa 3.2 datasetit on jaoteltu neljään pääryhmään tutkimuksen sovellusalueen perusteella: 1. IoT ja IIoT, 2. Kyberturva, verkkoliikenne ja blockchain, 3. Sensorit teollisuus ja ajalliset prosessit, sekä 4. Yleiskäyttöiset. Jokaisella sovellusalueella oli useampi datasetti jolla algoritmeja koulutettiin ja testattiin, ja tutkimukset olivat aika tasaisesti jakautuneet käsittelemään eri sovellusalueita.

Kyberturvaan ja verkkoliikenteeseen liittyvät datasetit kuten KDDCUP99, NSL-KDD, UNSW-NB15 ja CTU-13 sisältävät korkeaulotteista verkkoliikenteen tapahtuma- tai pakettidataa, joissa poikkeamat voivat olla esimerkiksi hyökkäyksiä tai epänormaalia liikennettä [21], [27], [28]. Teolliset ja sensoridataan liittyvät datasetit kuten SWAT, WADI sekä todellinen tai simuloitu sensoridata sisältävät tyypillisesti aikasarjadataa useista sensoreista, jolloin poikkeamat liittyvät esimerkiksi lai-

Taulukko 3.2: Aineistossa käytettyjä datasettejä sovellusalueen mukaan

Sovellusalue	Datasetit	Lähteet
1. IoT, IIoT	KDDCUP99, NAB	[18]
	N-Balot	[31]
	Yahoo! s5	[33]
	ASG	[38]
	IoTID20	[39]
2. Kyberturva, verkkoliikenne, blockchain	CTU-13	[27]
	NSL-KDD, CIDDS-2017	[28]
	Normal	[20]
	UNSW-NB15	[21]
3. Sensorit, teollisuus, ajalliset prosessit	SWAT, WADI, MSL	[20]
	Sensoridata	[35]
	VANET	[26]
	-	[37]
4. Yleiskäyttöiset	Shuttle, SMTP	[17]
	Bank, Forest Cover, HTTP	[19]
	KPI, PEMS, RealTweets	[22]
	Yahoo! s5	[25]
	-	[29], [36]

tevikoihin tai prosessihäiriöihin tai tarkoituksellisiin häiriöihin [20], [35]. Yleiskäyttöiset vertailudatasetit kuten Shuttle, Forest Cover ja SMTP ovat olleet laajassa käytössä algoritmien perussuorituskyvyn arvioinnissa [17], [19].

Tutkimusaineistosta voidaan havaita, että sovellusalue vaikuttaa jo algoritmin koulutukseen ja testaukseen, ja sitä kautta valmiin datasetin valintaan tehtävää varten. Ei ole myöskään nähtävissä datasettiä, joka olisi ollut käytössä jokaisella sovellusalueella, mikä korostaa havainnon merkitystä.

3.2 Menetelmien vertailu

Tutkimusaineistossa esiintyneitä algoritmeja ja niiden suorituskykyä tarkasteltiin vertailemalla aineistoissa saatuja tutkimustuloksia. Vertailussa on otettu huomioon tutkimusten sovellusalueet, olosuhteet, rajoitteet, ja arviointikriteerit. Vertailun tu-

loksista on voitu tehdä tulkintoja esimerkiksi yleisimmistä ja suorituskykyisimmistä menetelmistä, sekä menetelmien kontekstisidonnaisuudesta.

Eri algoritmeilla on erilaiset tavat saavuttaa sama tavoite; havaita poikkeama nopeasti normaalin datan joukosta. Algoritmit on taulukossa 3.3 luokiteltu sekä oppimis- että toimintaperiaatteen mukaan, jolloin voidaan tutkia mitkä oppimis- ja toimintatavat toimivat yleisesti ottaen tai tietyssä kontekstissa parhaiten. Samalla nähdään miten kukin algoritmi oppii mallin, ja minkä perusteella se määrittelee poikkeaman tässä aineistossa, eli miten algoritmeja on aineistossa käytetty.

Luvussa 2.5 esiteltiin oppimistapoja ja mallin päivitystapoja koneoppimisen kontekstissa. Näistä oppimistavoista tutkimusaineistossa eniten käytettiin ohjaamatonta (kuudessa tutkimuksessa), ja vähiten semi-ohjattua (yhdessä tutkimuksessa) ja ohjattua (kahdessa tutkimuksessa). Taulukossa 3.3 on oppimistavan ja toimintatavan lisäksi mainittu oppimisympäristö *verkossa*, joka kuvaa mallin päivitystapaa. Verkossa päivittyviä menetelmiä oli viisi kappaletta, joista kolme oli puumenetelmiin perustuvia (iForestASD, HST ja ASTREAM), DenStream perustuu tiheyteen, ja CluStream klusterointiin. Nämä menetelmät on rakennettu erityisesti konseptimuutoksiin soveltuviksi.

Taulukko 3.3: Algoritmien luokittelu oppimis- ja toimintaperiaatteen mukaan

Oppimistavat							**Toimintaperiaatteet**							**Lähteet**
Algoritmi	Ohjattu	Ohjaamaton	Semi-ohjattu	Verkossa	Hybridi		Tilastollinen	Etäisyyspohjainen	Raja-arvopohjainen	Klusterointipohjainen	Eristämisohjainen	Rekonstruktiopohjainen	Ennustepohjainen	**Viitteet**
k-NN	x							x						[17], [21]
Random Forest	x										x			[23], [38]
Isolation Forest		x									x			[17], [24], [36]
One-Class SVM			x						x					[17], [27], [28]
k-means		x								x				[21], [24], [27]
DBSCAN		x								x				[24]
DenStream				x						x				[34]
CluStream				x						x				[25]
HST				x							x			[33]
iForestASD				x							x			[33], [36]
ASTREAM				x							x			[39]
PCA		x										x		[17]
Autoencoder		x										x		[20], [31], [32]
LSTM		x											x	[20], [22], [26], [29], [35]
CNN		x											x	[22]
AE+LOF					x			x				x		[37]
LSTM-AE					x							x	x	[19], [22], [31]
iForest+Clustering					x					x	x			[25], [39]

Taulukon 3.3 mukaan yksikään aineistossa esiintyvä menetelmä ei ole tilastollinen, mikä luonnollisesti johtuu siitä, että aineistossa suoratoistodataa ei ole kokonaisuudessaan saatavilla. Yleisimmät algoritmit olivat joko klusterointiin tai eristämiseen pohjautuvia. Kuitenkin eniten tutkittu menetelmä on LSTM tai LSTM-pohjainen (kahdeksan tutkimusta), kun taas Isolation Forest, One-Class SVM, k-means ja Autoencoder tulivat kukin esiin kolmessa eri tutkimuksessa.

Tutkimusaineistossa korostuivat erityisesti LSTM-pohjaiset menetelmät [20], [22], [26], [29], [35], Isolation Forest [17], [24], [36] sekä erilaiset hybridimallit [19],

[22], [25], [31], [37], [39]. LSTM-pohjaiset menetelmät olivat yleisimpiä mm ajallista dataa ja yleiskäyttöisissä tutkimuksissa, ja saavuttivat monissa tapauksissa korkeimmat F1-tulokset. Isolation Forest on ollut erityisesti IoT, IIoT, teollisuus ja sensoridataan liittyvissä tutkimuksissa esillä.

Koska AD-menetelmiä oli pelkästään näissä tutkimuksissa mainittuna yli 30 kpl, kaikkia menetelmiä ei ole vertailtu keskenään, samassa kontekstissa, eikä välttämättä edes samoilla arviointimenetelmillä. Yleisimmin käytetty arviointimenetelmä on F1-tulos, jota käytti 15 lähdettä, osa pelkästään, ja osa jonkun muun mittarin lisäksi. F1-tulos ja tarkkuus antavat luotettavia tuloksia erityisesti tasapainoisen datasetin kohdalla, mutta epätasapainoinen datasetti voi antaa harhaanjohtavia tuloksia, mikä osaltaan vaikuttaa tutkimusten väliseen luotettavaan arviointiin [40].

AD-menetelmiä vertailtiin F1-tuloksen lisäksi myös havainnoinnin tarkkuuden ja palauttamisen osalta. Useissa tutkimuksissa otettiin myös huomioon menetelmän laskennallinen kuormitus [19], [23], [39], joka on olennainen suuren datamäärän kanssa. Käytetyt arviointimittarit ja testausolosuhteet vaihtelivat huomattavasti.

3.3 Tulevaisuuden tutkimukset ja kehitys

Aineistossa esiteltiin erilaisia suuntia jatkotutkimusta ja kehitystä varten. Taulukossa 3.4 on esitetty tutkimuksissa esiintyneet tutkimussuunnat neljään pääluokkaan jaoteltuna: hybridimallit ja menetelmät, reaaliaikaisuus ja skaalautuvuus, aikasarja ja ennustaminen, sekä tulkittavuus ja luotettavuus.

Hybridimallit ja menetelmät ovat hyvien tulosten takia luonnollisesti jatkokehityksen kohteena [24], [31], [36]. Tulevaisuudessa pyritään yhdistämään menetelmien parhaita puolia datan käsittelystä mallin oppimiseen ja päivittämiseen ja vielä edelleen toimintatapaan. Näissä tutkimuksissa paino oli myös erityisesti juuri hybridimenetelmien kehityksessä, myös muuhun kuin suoratoistodataan [19].

Taulukko 3.4: Aineistossa esiintyneitä tulevaisuuden tutkimussuuntia

Lähteet	Hybridimallit ja menetelmät	Reaaliaikaisuus ja skaalautuvuus	Aikasarja ja ennustaminen	Tulkittavuus ja luotettavuus
Al-Amri et al. (2021) [18]				x
Diaz Rivera et al. (2021) [21]				x
Togbe (2021) [36]	x		x	
Choi et al. 2021 [20]				x
Maleki et al. (2021) [31]	x	x		
Yahyaoui et al. (2021) [38]		x		x
Ribeiro et al. (2021) [34]		x		x
Liu et al. (2021) [30]		x	x	
Kim et al. (2022) [28]		x		
Gerz et al. (2022) [24]	x		x	x
Belacel et al. (2022) [19]	x	x		
Velasquez et al. (2022) [37]			x	x
Fosic et al. (2023) [23]		x		x
Lei et al. (2023) [29]		x		
Yang et al. (2023) [39]			x	x
Gong (2023) [26]				x
Nixon et al. 2024 [32]				x
Agyemang (2024) [17]				x
Ray et al. (2024) [33]			x	x
Suryadevara et al. (2025) [35]		x		

Reaaliaikaisuus ja skaalautuvuus tuli esiin useissa tutkimuksissa. Reaaliaikaisuus on tulvaisuudessa lähes perus ominaisuus ennakoivan havainnoinnin rinnalla. Skaalautuvuus taas kattaa sekä teknisen suorituskyvyn, että sen, kuinka hyvin menetelmä siirtyy sovellusalueelta toiseen. [23], [29], [31], [34], [38]

Aikasarja ja ennustaminen on kehityskohteina tutkimuksissa, joissa jatkon kannalta tärkeää on ennakoida poikkeamien esiintymisen aiheuttamia ongelmia, eli ennakoiva AD on tärkeää. Näissä tutkimuksissa oli myös suuri aika- ja kontekstiriippuvuus [33], [37], [39].

Tulkittavuus ja luotettavuus oli tärkeä tutkimussuunta suurimmassa osassa tutkimuksia. Tässä tarkoitetaan muun muassa menetelmien ja mallien testaamista ja koulutusta myös oikealla datalla ja todellisen maailman olosuhteissa [24], [33], [34]. Myös tulosten vertailukelpoisuus ja luotettavuus on riippuvainen testauksen olosuhteista ja objektiivisuudesta. Esimerkiksi menetelmän koulutus ja/tai testaus yhdellä synteettisellä datalla verrattuna oikeisiin tilanteisiin, vaikuttaa tulosten luotettavuuteen. [20], [21], [26], [32]

3.4 Analyysin yhteenveto

Suoratoistodata sovellusalueella on tutkimuksen perusteella erittäin suuri merkitys AD-menetelmän valinnalla. Suoratoistadalla on useita erilaisia sovellusalueita, joissa on merkittäviä eroja suoratoiston menetelmistä ja syötetystä datasta. Nämä erot ja haasteet vaikuttavat siihen, että pelkästään suoratoistoon tarkoitettu AD-menetelmä ei ole jokaisessa kontekstissa sopiva, vaan vaaditaan hyvinkin erilaisia menetelmiä sovelluskohteittain. Jo algoritmin kouluttaminen ohjaa usein työstämään algoritmia tietynlaisten tarkoitukseen sopivien datasettien kanssa, koska suoratoistodatassa poikkeamat ovat hyvin harvinaisia, ja suuri määrä merkittyä dataa on rajallisesti saatavilla.

Näin ollen vaikka erilaiset toimintaperiaatteet soveltuvat eri tavoin suoratoiston tuomiin haasteisiin, oikeanlaisen AD-menetelmän valinta korostuu käytännössä. Koska sovelluskohteita on niin monia erilaisia, ja teknologia kehittyy jatkuvasti, myös poikkeamat kehittyvät ja muuttuvat. Yleiskäyttöisten sekä tiettyyn systeemiin tai sovelluskohteeseen räätälöityjen menetelmien kehittäminen on tärkeää. Aineistosta sai kuitenkin kattavan katsauksen koneoppimispohjaisiin AD-menetelmiin, suoratoistodatan sovellusalueen merkitykseen ja ehdotettuihin tulevaisuuden tutkimussuuntiin.

4 Pohdinta

LSTM-pohjaiset mallit pärjäsivät erityisesti F1-tulosten osalta useissa eri tutkimuksissa, mutta suuren aikavaatimuksen takia menetelmä on hidas. Tämä rajoittaa menetelmän käyttöä reaaliaikaisessa ympäristössä. Esimerkiksi LSTM-AE kärsii teollisessa ympäristössä äänekkäästä datasta, laskennan monimutkaisuudesta ja kyvyttömyydestä verkossa oppimiseen [31]. Isolation Forest taas tarjosi kohtuullista suorituskykyä kevyemmällä laskennallisella kuormituksella. Useissa tutkimuksissa korkea havainnointitarkkuus saavutettiin kuitenkin merkittävällä laskennallisella kustannuksella. Tämä vaikuttaa suoraan menetelmien soveltuvuuteen reaalikäytössä.

LSTM, Isolation Forest ja hybridi-menetelmiä käytettiin laajalti eri sovellusalueilla, mikä viittaa niiden yleiseen soveltuvuuteen suoratoistodatan AD-menetelminä. Aineistosta nousi selkeästi esiin se, että tiettyyn käyttökohteeseen kehitetty menetelmä oli monissa tutkimuksissa [19], [21], [30], [37] pärjännyt erityisen hyvin verrattavina olleita AD-menetelmiä vastaan.

Vaikka räätälöityjä hybridimuotoisia menetelmiä esiintyi useilla eri sovellusalueilla, ne olivat omalla sovellusalueellaan huomioineet sen kontekstin yleiskäyttöisen menetelmien sijaan. Tätä havaintoa tukee myös se, että AD-menetelmät, jotka olivat tiettyä käyttökohdetta varten suunniteltu, suoriutuivat johdonmukaisesti muita AD-menetelmiä paremmin [29], [39]. Samaan aikaan ei löytynyt AD-menetelmää, joka olisi jokaisella sovellusalueella paras.

Tutkimuksissa käytetty oppimistapa vaikutti siihen, minkälaista dataa tai datasettejä oli mahdollista hyödyntää algoritmien koulutuksessa ja testauksessa. Ohjatut ja semi-ohjatut menetelmät edellyttivät merkattua dataa, jolloin menetelmän käytössä voi tulla vastaan oikean ja synteettisen datan erot. Ohjaamatonta oppimista ja online-päivittämistä hyödyntävät menetelmät taas käyttivät datasettejä pitkälti testaamiseen, koska käyttötarkoitus ja sovelluskohde vaatii menetelmän suoriutumista ilman merkattua dataa. Tämä vahvistaa käsitystä siitä, että ajan myötä mahdollisesti muuttuva ja dynaaminen data on ohjannut luomaan algoritmeja, jotka eivät vaadi tietoa datan alku- tai loppuarvoista etukäteen.

Jotkut tutkimukset vertailivat algoritmeja vain yhden datasetin avulla, kun taas osa tutkimuksista vertaili eri algoritmien suorituskykyä eri dataseteillä, mikä mahdollistaa kriittisemmän arvioinnin. Samojen datasettien käyttö edesauttaa jonkin verran tutkimusten välistä vertailua, etenkin silloin, kun myös arviointimenetelmät ovat olleet yhtenäisiä eri tutkimuksissa. Vertailun eroavuuksiin vaikuttaa myös se, että eri tutkimuksissa mitattiin eri asiaa (tarkkuus, palauttaminen) sovelluskohteesta riippuen. Sovellusalueilla on selkeitä eroja esimerkiksi datan rakenteessa ja poikkeamien luonteessa, mikä osaltaan selittää miksi ei ole olemassa yhtä kaikkiin tapauksiin sopivaa AD-menetelmää. Sovellusalueen kontekstisidonnaisuus samalla heikentää tulosten keskinäistä arviointia, koska kaikkia AD-menetelmiä ei ole testattu ja arvioitu samoissa olosuhteissa tai välttämättä edes samoilla kriteereillä.

Aineistossa nousi systemaattisesti esiin samat teemat, kun tarkasteltiin niissä esitettyjä jatkotutkimusten aiheita. Ensinnäkin ohjattua tai semi-ohjattua oppimista hyödyntävien AD-menetelmien kehitys vaatisi paljon nykyistä enemmän merkattua dataa. Rajattu määrä merkattua dataa rajoittaa ja haastaa myös hyvin pärjänneiden AD-menetelmien kehitystä [31]. Toinen huomio on se, että suuremmat datasetit tyypillisesti parantavat havainnoinnin tarkkuutta, koska silloin normaali data voi-

daan esittää tarkemmin [17]. Isommat datasetit kuitenkin vaikuttaisivat suoraan algoritmin opetusajan ja muistinkäytön kasvamiseen.

Hybridimenetelmien hyvät tulokset tarkoittavat, että niitä kehitetään eteenpäin jatkossakin. Erityisesti online-päivittämisen ja stream [21], [25], [33], [36], [39](CluStream, ASTREAM) menetelmien yhteensopivuus on tärkeä kehityksen ja tutkimuksen kohde. Voidaan havaita, että reaalisaikaisuus on tulevaisuudessa perusoletus eikä vain toivottava ominaisuus [36], [39]. Samoin myös konseptimuutos on otettava perusominaisuutena, ja se on huomioitava menetelmien jatkokehityksessä.

Kuten työssä on jo aiemmin todettu, sovellusalueella on suuri merkitys AD-menetelmän valintaan, jolloin sovellusaluekohtaista optimointia tulee tapahtumaan myös tulevaisuudessa [22], [30], [33]. Yleiskäyttöisten mallien sijaan jatkossa keskitytään enemmän kontekstisidonnaisiin ratkaisuihin.

Tutkimukset, joissa käsiteltiin tai vertailtiin samoja AD-menetelmiä, tuottivat samantyyppisiä tuloksia, mikä viittaa tutkimusten johdonmukaisuuteen. Menetelmien ja tulosten vertailu oli mahdollista erityisesti niissä tutkimuksissa, joissa käytettiin samoja arviointimittareita, kuten F1-tulosta. F1-tuloksissa oli kuitenkin vaihtelua esimerkiksi käytetyn datasetin mukaan. Tämä korostuu etenkin silloin, kun otetaan huomioon esimerkiksi algoritmin aikavaatimus tai sovelluskohteen tarpeet.

Kuten aiemmin todettiin, työ osoitti, että AD-menetelmän valinta on kontekstidonnaista, ja tämä tulee korostumaan tulevaisuudessa. Täytyy olla selvää mihin algoritmi kykenee ja minkälaisia poikkeamia sillä voidaan havaita [2]. Esimerkiksi [17] huomauttaa että menetelmän soveltuvuus voi myös riippua siitä, kuinka paljon tärkeämpää on saada jokainen poikkeama kiinni, kuin välttää vääriä hälytyksiä. Aineiston rajoitteiden takia työn johtopäätökset ovat itsessäänkin kontekstisidonnaisia, eikä voi varmuudella todeta, ettei koskaan tulisi olemaan suoratoistodataan tarkoitettua yleiskäyttöistä AD-metelmää, joka olisi kaikilla mittareilla ja olosuhteissa tehokas.

Tulevaisuudessa tutkimuksia varten olisi hyvä kehittää yhtenäisempiä toimintatapoja AD-menetelmien arviointiin, jolloin tulosten vertailu helpottuu, ja algoritmin opetukseen ja testaamisen käytetyn datan valinta on läpinäkyvämpää. Aineistossa eri tutkimusten ja artikkelien erot arviointimittareissa hankaloittivat vertailua menetelmien suorituskyvyistä eri konteksteissa. Tätä enemmän vertailua hankaloittaa oikean datan puute, koska synteettisellä datalla testaaminen on subjektiivista ennakkotiedon ansiosta. Sensoridataa lukuun ottamatta harvemmin testejä tehtiin tositilanteissa, jolloin todellinen suorituskyky jäi epäselväksi, vaan lähinnä suuntaa-antavaksi.

Kokonaisuutena tarkasteltuna työ vahvistaa käsitystä siitä, että suoratoistodatan poikkeamien havainnointi edellyttää yhä enemmän sovelluskohtaisesti optimoituja, reaaliaikaisia ja adaptiivisia menetelmiä. Samaan aikaan näiden menetelmien arviointi vaatii yhä yhtenäisempiä käytäntöjä.

5 Yhteenveto

Tässä työssä tutkittiin koneoppimispohjaisten AD-menetelmien käyttöä ja soveltuvuutta suoratoistodataan. Suoratoistodatan määrä kasvaa jatkuvasti, ja virtaavan datan reaaliaikaisuutta pidetään perusoletuksena, jolloin erilaisten poikkeamien nopean ja tehokkaan havainnoinnin tärkeys korostuu. Työn keskiössä oli eri AD-menetelmien soveltuvuus suoratoistodatan erilaisiin sovellusaluisiin, ja koneoppiminen AD-menetelmien taustalla.

Ensimmäiseen tutkimuskysymykseen vastattiin luvussa 3.2 menetelmien vertailun yhteydessä. Työn pohjalta voidaan todeta, että suoratoistodatan sovellusalue ehdottomasti vaikuttaa AD-menetelmän valintaan. Suoratoistodatan eri sovellusalueilla on erilaisia teknisiä ominaisuuksia, kuten datan päivitys, moniulotteisuus ja reaaliaikaisuus, sekä eri asioiden tärkeyden painotus (väärä hälytys vs. huomaamattomat poikkeamat). Nämä vaikuttaa siihen, mikä AD-menetelmä sovellusalueelle soveltuu.

Toiseen tutkimuskysymykseen vastattiin luvussa 3.3, jossa käsiteltiin tulevaisuuden tutkimusta ja kehitystä aiheen tiimoilta. Koneoppimispohjaisten suoratoistodataan soveltuvien AD-menetelmien kehityksessä ja tulevaisuuden tutkimuksessa on kirjallisuuskatsauksen kohdalla selkeitä suuntia. Saatavilla oleva data määrittää esimerkiksi menetelmien koulutusta ja testausta eri sovellusalueilla, ja hybridimenetelmiä kehitetään eteenpäin myös tulevaisuudessa erityisesti online-päivittämisen

hyödyntämiseksi. Lisäksi yhä useammin vaaditaan konseptimuutoksen ja reaaliaikaisuuden huomioimista suoratoistodatan perusominaisuutena.

Työssä tuotiin esiin koneoppimisen edellytys suoratoistodatan AD-menetelmän taustalla. Siinä myös huomautettiin, että menetelmän toimintaperiaate on yhtä tärkeä kuin oppimisperiaate, ja menetelmien välisen suorituskyvyn arvioiminen tulisi olla nykyistä yhtenäisempää ja läpinäkyvämpää. Jatkotutkimuksissa tulisikin kiinnittää sovellusaluekontekstin ja algoritmien kehityksen lisäksi huomiota myös aiheen käytäntöjen ja termistön yhtenäistämiseen, jolloin saadaan vertailukelpoisia tutkimuksia. Aiheen nopea kehitys vaatii kirjallisuuskatsausten lisäksi lisää empiiristä tutkimusta, jossa tulokset ovat vertailukelpoisia.

Lähdeluettelo

[1] C. C. Aggarwal, "An Introduction to Outlier Analysis", teoksessa *Outlier Analysis*, C. C. Aggarwal, toim., Cham: Springer International Publishing, 2017, s. 1–34. DOI: 10.1007/978-3-319-47578-3_1.

[2] R. Foorthuis, "On the nature and types of anomalies: a review of deviations in data", *International Journal of Data Science and Analytics*, vol. 12, nro 4, s. 297–331, 2021. DOI: 10.1007/s41060-021-00265-1.

[3] A. L. Buczak ja E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection", *IEEE Communications Surveys & Tutorials*, vol. 18, nro 2, s. 1153–1176, 2016. DOI: 10.1109/COMST.2015.2494502.

[4] W. Cao ja L. Cao, "Financial Crisis Forecasting via Coupled Market State Analysis", *IEEE Intelligent Systems*, vol. 30, nro 2, s. 18–25, maaliskuu 2015. DOI: 10.1109/MIS.2015.4.

[5] K.G.Mehrotra, C. K. Mohan ja H. Huang, *Anomaly Detection Principles and Algorithms. in Terrorism, Security, and Computation.* Cham: Springer International Publishing, 2017. DOI: 10.1007/978-3-319-67526-8.

[6] J. Gama, *Knowledge Discovery from Data Streams.* Chapman & Hall/CRC, 2010, s. 255, ISBN: 978-1-4398-2611-9.

[7] M. Garofalakis, J. Gehrke ja R. Rastogi, "Data Stream Management: A Brave New World", teoksessa *Data Stream Management: Processing High-Speed Data Streams*, M. Garofalakis, J. Gehrke ja R. Rastogi, toim., Berlin, Heidelberg: Springer, 2016, s. 1–9. DOI: 10.1007/978-3-540-28608-0_1.

[8] C. C. Aggarwal, "High-Dimensional Outlier Detection: Time Series and Multidimensional Streaming Outlier Detection", teoksessa *Outlier Analysis*, Springer New York, 2017, s. 225–265. DOI: 10.1007/978-1-4614-6396-2_5.

[9] A. B. V. Chandola ja V. Kumar, "Anomaly detection: A survey", *ACM Comput. Surv.*, vol. 41, s. 1–58, 2009. DOI: 10.1145/1541880.1541882.

[10] P. Z. C. Janiesch ja K. Heinrich, "Machine learning and deep learning", *Electron Markets*, vol. 31, s. 685–695, 2021. DOI: 10.1007/s12525-021-00475-2.

[11] C. C. Aggarwal, "High-Dimensional Outlier Detection: Proximity-Based Outlier Detection", teoksessa *Outlier Analysis*, Springer New York, 2017, s. 101–133. DOI: 10.1007/978-1-4614-6396-2_5.

[12] C. C. Aggarwal, "High-Dimensional Outlier Detection: The Subspace Method," in Outlier Analysis", teoksessa *Outlier Analysis*, Springer New York, 2017, s. 135–167. DOI: 10.1007/978-1-4614-6396-2_5.

[13] Z. Zamanzadeh Darban, G. I. Webb, S. Pan, C. Aggarwal ja M. Salehi, "Deep Learning for Time Series Anomaly Detection: A Survey", *ACM Computing Surveys*, vol. 57, nro 1, 15:1–15:42, 2024. DOI: 10.1145/3691338.

[14] C. Shah. "A Hands-On Introduction to Machine Learning". Cambridge Aspire, viitattu 17. joulukuuta 2025. url: https://www.cambridge.org/highereducation/books/a-hands-on-introduction-to-machine-learning/3E57313A963BF7AF5C6330EB88ADAB2E.

[15] S. Albers, "Online algorithms: a survey", *Mathematical Programming, Series B*, vol. 97, nro 1, s. 3–26, heinäkuu 2003. DOI: 10.1007/s10107-003-0436-0.

[16] S. C. H. Hoi, D. Sahoo, J. Lu ja P. Zhao. ”Online Learning: A Comprehensive Survey”. Preprint (arXiv). eprint: arXiv:1802.02871, viitattu 17. joulukuuta 2025. url: https://arxiv.org/abs/1802.02871.

[17] E. F. Agyemang, ”Anomaly detection using unsupervised machine learning algorithms: A simulation study”, *Scientific African*, vol. 26, e02386, 2024. DOI: 10.1016/j.sciaf.2024.e02386.

[18] R. Al-amri, R. K. Murugesan, M. Man, A. F. Abdulateef, M. A. Al-Sharafi ja A. A. Alkahtani, ”A Review of Machine Learning and Deep Learning Techniques for Anomaly Detection in IoT Data”, *Applied Sciences*, vol. 11, nro 12, s. 5320, 2021. DOI: 10.3390/app11125320.

[19] N. Belacel, R. Richard ja Z. M. Xu, ”An LSTM Encoder-Decoder Approach for Unsupervised Online Anomaly Detection in Machine Learning Packages for Streaming Data”, teoksessa *2022 IEEE International Conference on Big Data (Big Data)*, 2022, s. 3348–3357. DOI: 10.1109/BigData55660.2022.10020872.

[20] K. Choi, J. Yi, C. Park ja S. Yoon, ”Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines”, *IEEE Access*, vol. 9, s. 120 043–120 065, 2021. DOI: 10.1109/ACCESS.2021.3107975.

[21] J. J. D. Rivera, T. A. Khan, W. Akbar, M. Afaq ja W.-C. Song, ”An ML Based Anomaly Detection System in real-time data streams”, teoksessa *2021 International Conference on Computational Science and Computational Intelligence (CSCI 2021)*, New York, 2021, s. 1329–1334. DOI: 10.1109/CSCI54926.2021.00270.

[22] A. Duraj, P. S. Szczepaniak ja A. Sadok, ”Detection of Anomalies in Data Streams Using the LSTM-CNN Model”, *Sensors*, vol. 25, nro 5, s. 1610, 2025. DOI: 10.3390/s25051610.

[23] I. Fosic, D. Zagar, K. Grgic ja V. Krizanovic, "Anomaly detection in NetFlow network traffic using supervised machine learning algorithms", *Journal of Industrial Information Integration*, vol. 33, s. 100 466, 2023. DOI: 10.1016/j.jii.2023.100466.

[24] F. Gerz, T. R. Bastuerk, J. Kirchhoff, J. Denker, L. Al-Shrouf ja M. Jelali, "A Comparative Study and a New Industrial Platform for Decentralized Anomaly Detection Using Machine Learning Algorithms", teoksessa *2022 International Joint Conference on Neural Networks (ijcnn)*, New York, 2022. DOI: 10.1109/IJCNN55064.2022.9892939.

[25] H. M. Gomes, N. Gunasekara ja Y. Sun, "Machine Learning on the Fly: A Hands-On Tutorial for Streaming Data", teoksessa *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, 2025, s. 4513–4516. DOI: 10.1109/ICDE65448.2025.00342.

[26] L. Gong, "Research on Anomaly Detection in Time-Series Streaming Data", teoksessa *2023 3rd International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)*, 2023, s. 332–337. DOI: 10.1109/CEI60616.2023.10528086.

[27] M. Jain, G. Kaur ja V. Saxena, "A K-Means clustering and SVM based hybrid concept drift detection technique for network anomaly detection", *Expert Systems with Applications*, vol. 193, s. 116 510, 2022. DOI: 10.1016/j.eswa.2022.116510.

[28] J. Kim et al., "A Machine Learning Approach to Anomaly Detection Based on Traffic Monitoring for Secure Blockchain Networking", *IEEE Transactions on Network and Service Management*, vol. 19, nro 3, s. 3619–3632, 2022. DOI: 10.1109/TNSM.2022.3173598.

[29] T. Lei, C. Gong, G. Chen, M. Ou, K. Yang ja J. Li, ”A novel unsupervised framework for time series data anomaly detection via spectrum decomposition”, *Knowledge-Based Systems*, vol. 280, s. 111 002, 2023. DOI: 10.1016/j.knosys.2023.111002.

[30] K. Liu, W. Mao, H. Shi ja X. Liang, ”A New Unsupervised Online Early Fault Detection Framework of Rolling Bearings Based on Granular Feature Forecasting”, *IEEE Access*, vol. 9, s. 159 684–159 698, 2021. DOI: 10.1109/ACCESS.2021.3132353.

[31] S. Maleki, S. Maleki ja N. R. Jennings, ”Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering”, *Applied Soft Computing*, vol. 108, s. 107 443, 2021. DOI: 10.1016/j.asoc.2021.107443.

[32] C. Nixon, M. Sedky, J. Champion ja M. Hassan, ”SALAD: A split active learning based unsupervised network data stream anomaly detection method using autoencoders”, *Expert Systems with Applications*, vol. 248, s. 123 439, 2024. DOI: 10.1016/j.eswa.2024.123439.

[33] S. K. Ray ja S. Susan, ”Performance Analysis of Online Machine Learning Frameworks for Anomaly Detection in IoT Data Streams”, teoksessa *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2024, s. 1–5. DOI: 10.1109/ICCCNT61001.2024.10724326.

[34] A. de R. L. Ribeiro, R. Y. C. Santos ja A. C. A. Nascimento, ”Anomaly Detection Technique for Intrusion Detection in SDN Environment using Continuous Data Stream Machine Learning Algorithms”, teoksessa *2021 IEEE International Systems Conference (SysCon)*, 2021, s. 1–7. DOI: 10.1109/SysCon48628.2021.9447092.

[35] G. Suryadevara, P. Udayaraju, P. Pachipulusu, M. Gayathri, M. Sitharam ja V. D. Kumar, ”Transfer Learning Model for Anomaly Detection in Data Streaming - Data Engineering Perspective”, teoksessa *2025 International Conference on Machine Learning and Autonomous Systems (ICMLAS)*, 2025, s. 250–258. DOI: 10.1109/ICMLAS64557.2025.10968941.

[36] M. U. Togbe, Y. Chabchoub, A. Boly, M. Barry, R. Chiky ja M. Bahri, ”Anomalies Detection Using Isolation in Concept-Drifting Data Streams”, *Computers*, vol. 10, nro 1, s. 13, 2021. DOI: 10.3390/computers10010013.

[37] D. Velásquez et al., ”A Hybrid Machine-Learning Ensemble for Anomaly Detection in Real-Time Industry 4.0 Systems”, *IEEE Access*, vol. 10, s. 72 024–72 036, 2022. DOI: 10.1109/ACCESS.2022.3188102.

[38] A. Yahyaoui, H. Lakhdhar, T. Abdellatif ja R. Attia, ”Machine learning based network intrusion detection for data streaming IoT applications”, teoksessa *2021 21st ACIS International Winter Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD-Winter 2021)*, 2021, s. 51–56. DOI: 10.1109/SNPDWinter52325.2021.00019.

[39] Y. Yang et al., ”ASTREAM: Data-Stream-Driven Scalable Anomaly Detection With Accuracy Guarantee in IIoT Environment”, *IEEE Transactions on Network Science and Engineering*, vol. 10, nro 5, s. 3007–3016, 2023. DOI: 10.1109/TNSE.2022.3157730.

[40] D. Chicco ja G. Jurman, ”The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation”, *BMC genomics*, vol. 21, s. 6–13, 2020. DOI: 10.1186/s12864-019-6413-7.