TURUN YLIOPISTO

Haittaohjelmien tunnistus koneoppimisen avulla

LuK-tutkielma
Tietojenkäsittelytiede
Tietotekniikan laitos, Teknillinen tiedekunta
Kandidaatintutkielma
Jessica Koskinen
Maaliskuu 2025

LuK-tutkielma
Tietotekniikan laitos, Teknillinen tiedekunta
Turun yliopisto

Tutkinto-ohjelma: Tietojenkäsittelytiede
Tekijä: Jessica Koskinen
Otsikko: Haittaohjelmien tunnistus koneoppimisen avulla
Sivumäärä: 22 sivua
Päivämäärä: Maaliskuu 2025

Haittaohjelmien määrän räjähdysmäinen kasvu on yksi kyberturvallisuuden suurimmista haasteista nykypäivänä. Näiden ohjelmien torjuminen on tärkeää niiden aiheuttaessa vakavia tietoturvauhkia, taloudellisia menetyksiä ja järjestelmien toimimattomuutta. Perinteiset tunnistusmenetelmät eivät ole riittävän tehokkaita kehittyneiden haittaohjelmien tunnistuksessa, mikä on johtanut uusien menetelmien käyttöönottoon. Koneoppimismalleja on alettu hyödyntää yhä enemmän haittaohjelmien tunnistuksessa niiden mahdollistaessa tiedon analysoinnin ja poikkeavuuksien havaitsemisen tehokkaasti.

Tämä tutkielma on kirjallisuuskatsaus, jossa tutustutaan ohjattuun, ohjaamattomaan ja puoliohjattuun oppimiseen perustuvien koneoppimismenetelmien hyödyntämiseen haittaohjelmien tunnistuksessa. Tutkielmassa tarkastellaan koneoppimismallien kehitysprosessia ja toimintaperiaatteita. Kirjallisuuskatsauksen keskeisenä tavoitteena on selvittää, mitä etuja ja haasteita koneoppimisen hyödyntämisellä on. Lisäksi tutkielmassa tarkastellaan viimevuotisia tutkimuksia, jotka keskittyvät koneoppimismenetelmien suorituskyvyn arviointiin todellisissa tunnistustilanteissa ohjelmien piirteisiin ja toimintaan perustuen. Tuloksia arvioidaan hyödyntämällä neljää mittaria: täsmällisyyttä, tarkkuutta, herkkyyttä ja F1-arvoa.

Tulosten perusteella havaitaan, että koneoppimismenetelmät tuottavat hyvin tarkkoja tuloksia haitallisten ohjelmien tunnistuksen tehostamiseksi ja automatisoimiseksi. Menetelmät vaativat suuren määrän laadukasta dataa ja kouluttamista, jotta ne suoriutuvat tehokkaasti ja tarkasti ohjelmien kehittyessä jatkuvasti. Tarkasteltujen tutkimusten perusteella suorituskyvyltään tarkimmiksi koneoppimismenetelmiksi osoittautuivat puupohjaiset mallit, erityisesti satunnaismetsäalgoritmi. Koneoppimisen soveltamista reaaliaikaisessa tunnistuksessa ja mallien kykyä reagoida tuntemattomiin haittaohjelmiin tulee tutkia lisää, jotta on mahdollista kehittää entistä tehokkaampia menetelmiä uhkien tunnistukseen.

Asiasanat: tekoäly, koneoppiminen, koneoppimisalgoritmi, haittaohjelma, kyberturvallisuus

Sisällysluettelo

1 Johdanto

Haittaohjelmilla (engl. *malware*) tarkoitetaan ohjelmistoja, joiden tarkoituksena on vahingoittaa, tuhota tai varastaa tietoa tietokoneilta tai tietojärjestelmistä (Mhara ym., 2024). Suurin osa verkossa olevista uhkista on jonkinlaisia haittaohjelmia (Sarker, 2023). Haittaohjelmat voivat levitä, kun käyttäjät lataavat tiedostoja epäluotettavilta sivustoilta, avaavat haitallisia sähköpostiliitteitä tai klikkaavat vahingollisia linkkejä. Haittaohjelmien tekijät voivat myös sisällyttää haitallista koodia muihin ohjelmiin tai ohjelmistojen päivityksiin sekä käyttää hyväkseen järjestelmien haavoittuvuuksia. (Patel & Ghosh, 2024; Sharma & Arora, 2021.)

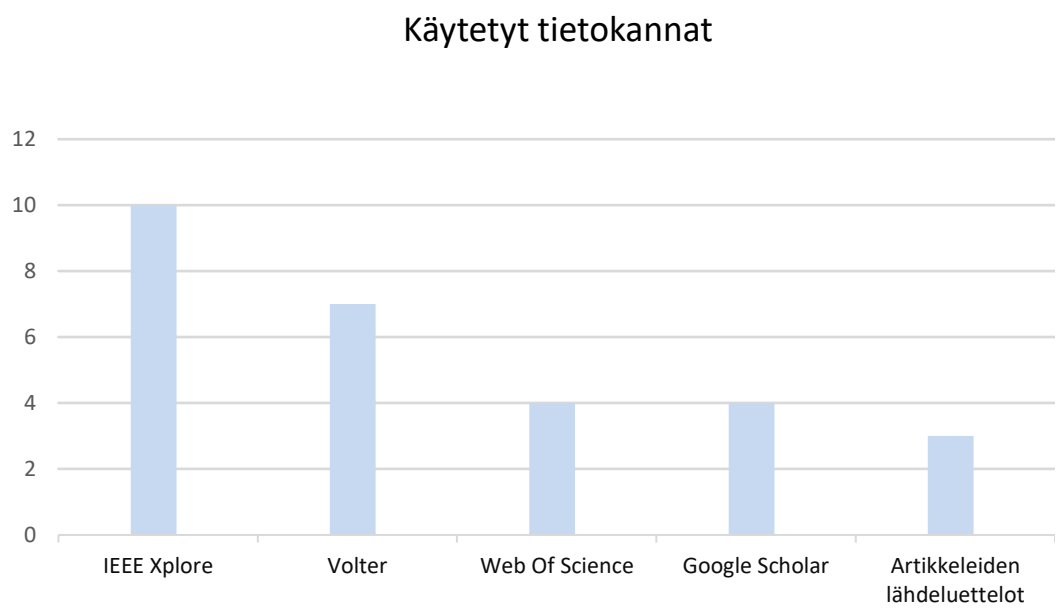
Teknologian kehitys ja digitaalisen informaation nopea leviäminen ovat johtaneet kyberuhkien määrän merkittävään kasvuun. Haittaohjelmat ovat yleistyneet samalla kehittyen yhä monimutkaisemmiksi ja vaikeammin havaittaviksi. Arvioiden mukaan päivittäin syntyy jopa noin miljoona uutta haittaohjelmamuunnosta, joista suurin osa on olemassa olevien haittaohjelmien kehittyneitä versioita. (Aslan ym., 2021.) Nämä ohjelmat muodostavat merkittävän uhan niin yksilöiden kuin organisaatioidenkin tietoturvalle. Tehokas kybertorjunta on keskeinen osa organisaatioiden kyberturvallisuusstrategiaa, ja se vaatii menetelmien jatkuvaa kehittämistä ja sopeutumista muuttuviin uhkiin (Nour & Said, 2024).

Perinteiset haittaohjelmien tunnistusmenetelmät, kuten allekirjoituspohjainen ja käyttäytymispohjainen tunnistus, ovat olleet pitkään keskeisimpiä keinoja haitallisten ohjelmien tunnistuksessa ja torjunnassa. Näillä menetelmillä on kuitenkin rajoitteita erityisesti monimutkaisten ja suorituksen aikana muuttuvien haittaohjelmien tunnistuksessa. Kehittyneet haittaohjelmat käyttävät eri menetelmiä piiloutuakseen ja jäädäkseen havaitsemattomiksi, jolloin perinteiset menetelmät eivät yksinään riitä torjumaan niitä (Sharma & Arora, 2021). Haitallisten ohjelmien määrän räjähdysmäinen kasvu on johtanut lukuisten uusien tunnistusmenetelmien käyttöönottoon. Viime vuosina koneoppiminen on noussut keskeiseksi keinoksi haittaohjelmien tunnistuksessa, tarjoten tehokkaampia ja kehittyneempiä menetelmiä.

Tutkielman tavoitteena on kartoittaa, kuinka koneoppimista voidaan hyödyntää haittaohjelmien tunnistuksessa nykypäivänä. Tutkielma on rajattu tarkastelemaan ohjattuun, ohjaamattomaan ja puoliohjattuun oppimiseen pohjautuvia koneoppimismenetelmiä, joita voidaan hyödyntää haittaohjelmien tunnistuksessa. Tutkielman pyrkii tarjoamaan kattavan yleiskuvan koneoppimismenetelmien eduista ja haitoista, sekä algoritmien tarkkuudesta haittaohjelmien tunnistuksessa eri ympäristöissä. Tämän tutkielman tavoitteiden pohjalta luodut tutkimuskysymykset ovat seuraavat:

TK1. Mitkä ovat koneoppimisen edut ja haasteet haittaohjelmien tunnistuksessa?
TK2. Kuinka tarkasti eri koneoppimisalgoritmit tunnistavat haittaohjelmia?

Tutkielma on toteutettu kirjallisuuskatsauksena. Tietoaineisto on kerätty tietokannoista IEEE Xplore, Web of Science, Volter ja Google Scholar. Keskeisimpänä hakulauseena käytettiin *(AI OR "Artificial Intelligence" OR" Machine Learning" OR ML) AND ("Cybersecurity" OR "Cyber Security") AND (Malware OR "Malware Detection")*. Hakulause tuotti jokaisessa käytetyssä tietokannassa satoja tuloksia, jonka jälkeen artikkeleita karsittiin otsikoiden ja julkaisuvuosien perusteella. Hakuvaiheessa rajattiin pois ennen vuotta 2018 julkaistut artikkelit, jotta haku tuottaisi mahdollisimman ajankohtaisia tuloksia. Tämän jälkeen jäljelle jääneistä aineistoista luettiin tiivistelmä ja yhteenveto, joiden perusteella aineistoa karsittiin. Lopulta tutkielmaan valittiin artikkelit, joka käsittelivät monipuolisesti koneoppimisen roolia ja algoritmien suorituskykyä haittaohjelmien tunnistuksessa. Lisäksi muutamia artikkeleita kerättiin tutkimusartikkeleiden lähdeluetteloista. Kaaviossa 1 on esitetty tutkielmassa hyödynnettyjen artikkeleiden alkuperä.



Kaavio 1: Tutkielmassa hyödynnettyjen artikkeleiden alkuperä

Seuraavassa luvussa käsitellään haittaohjelmia ja niiden toiminnan analysointia sekä perinteisten tunnistusmenetelmien toimintaperiaatteita teoreettisesti. Kolmannessa luvussa tarkastellaan koneoppimisen käsitettä sekä koneoppimismallien kehittämistä ja toimintaperiaatteita. Lisäksi käsitellään koneoppimisen asemaa haittaohjelmien tunnistuksessa nykypäivänä sekä sen tarjoamia etuja ja haasteita. Luvussa 4 arvioidaan ja vertaillaan koneoppimisalgoritmien suorituskykyä eri tilanteissa viimevuotisiin tutkimuksiin perustuen. Lopuksi yhteenvedossa, luvussa 5, esitetään tutkielman tärkeimmät havainnot sekä vastataan tutkielman tutkimuskysymyksiin.

2 Haittaohjelmat ja perinteiset tunnistusmenetelmät

Tässä luvussa tarkastellaan yleisimpiä haittaohjelmia, niiden toiminnan analysointia ja perinteisiä tunnistusmenetelmiä. Nämä asettavat perustan koneoppimispohjaisille haittaohjelmien tunnistusmenetelmille, joita käsitellään myöhemmin tutkielmassa.

2.1 Haittaohjelmat

Haittaohjelmia luokitellaan muun muassa niiden leviämistavan, toimintaperiaatteen ja tavoitteen perusteella (Song ym., 2024). Haittaohjelmatyypeillä on eri piirteitä ja näiden ominaisuuksien tunnistaminen on tärkeää, jotta haittaohjelmia voidaan havaita ja torjua. Haittaohjelmatyyppien lukumäärää on vaikeaa arvioida, määrän kasvaessa ja uusien variaatioiden syntyessä jatkuvasti. Yleisemmin tunnettuja haittaohjelmatyyppejä ovat muun muassa virus, mato, troijalainen sekä mainos-, vakoilu- ja kiristysohjelmat.

- **Virus** on haitallinen koodi, joka ei kykene toimimaan itsenäisesti, vaan tarvitsee isäntäohjelman tai -tiedoston toimiakseen. Kun virus on päässyt järjestelmään, se pystyy kopioimaan itseään muihin ohjelmiin. (Faruk ym., 2021.) Virukset käynnistyvät yleensä käyttäjän toimenpiteiden, kuten haitallisen tiedoston avaamisen, seurauksena (Bazrafshan ym., 2013).

- **Mato** (engl. *computer worm*) on ohjelma, joka suorittaa itsenäisesti haitallisia toimintoja ilman käyttäjän lupaa. (Mhara ym., 2024.)

- **Troijalainen** (engl. *Trojan*) on haittaohjelma, joka näyttää tavalliselta ohjelmalta, mutta suorittaa haitallisia toimintoja ohjelman käynnistyessä (Song ym., 2024).

- **Mainosohjelma** (engl. *adware*) on ei-toivottu ohjelmisto, joka toistaa automaattisesti mainoksia käyttäjän tietokoneella ohjelmiston asennuksen tai suorituksen aikana. Mainosohjelmat voivat ohjata käyttäjän verkkosivustoille, joilla he eivät halua vierailla. (Song ym., 2024.)

- **Vakoiluohjelma** (engl. *spyware*) on haitallinen ohjelmisto, joka asennetaan käyttäjän tietokoneelle keräämään tietoja ja välittämään niitä eteenpäin ilman käyttäjän suostumusta. Modernin vakoiluohjelman tunnistaminen on vaikeaa, sillä ne kykenevät päivittämään jatkuvasti koodiaan. (Akhtar & Feng, 2022.)

- **Kiristysohjelma** (engl. *ransomware*) on haittaohjelma, joka salaa järjestelmän tai henkilökohtaiset tiedostot ja vaatii lunnaita pääsyn palauttamiseksi (Song ym., 2024).

Yksittäisten haittaohjelmatyyppien lisäksi on olemassa hybridihaittaohjelmia, jotka koostuvat useista haittaohjelmatyypeistä. Nämä ohjelmat ovat erittäin vaarallisia ja voivat aiheuttaa laajaa vahinkoa järjestelmille. (Sharma & Arora, 2021.) Lisäksi haittaohjelmat voivat luoda itsestään uusia variaatioita kiertääkseen tunnistusmekanismeja samalla hyödyntäen jo olemassa olevaa koodia ja resursseja (Ucci ym., 2019).

Haittaohjelmat ovat kasvava ongelma, koska ne yleistyvät ja kehittyvät jatkuvasti entistä monimutkaisemmiksi. Ne käyttävät hyväkseen järjestelmien haavoittuvuuksia ja mukautuvat nopeasti uusiin teknologioihin, mikä tekee niistä vaikeammin havaittavia. Haittaohjelmat ovat merkittävä uhka sekä tietojärjestelmille että käyttäjille (Faruk ym., 2021). Haittaohjelmat voivat aiheuttaa merkittäviä vahinkoja, kuten taloudellisia menetyksiä, tiedon menetystä sekä organisaatioiden ja yksilöiden maineen vahingoittumista.

Globalisaation myötä haittaohjelmien seuraukset voivat levitä nopeasti ja vaikuttaa laajoihin verkostoihin. Yksi tunnetuimmista haittaohjelmista on vuonna 2017 levinnyt WannaCry-kiristysohjelma, joka vaikutti yli 230 000 tietokoneen toimintaan 150 maassa aiheuttaen vakavia häiriöitä terveydenhuolto- ja yritysverkostoille maailmanlaajuisesti. (Djenna ym., 2023; Mhara ym., 2024.) Haittaohjelmien seuraukset voivat heijastua kansainvälisesti vaikuttaen sekä yksilöiden että organisaatioiden toimintaan, mikä korostaa niiden laajamittaisia ja monialaisia seurauksia nykypäivänä.

2.2 Haittaohjelmien toiminnan analysointi

Haittaohjelmien tunnistuksen keskeinen vaihe on niiden toiminnan analysointi kohdejärjestelmässä. Analysoinnin tavoitteena on kerätä mahdollisimman paljon arvokasta tietoa, jonka perusteella voidaan tunnistaa haitallisen ohjelman toiminnot ja käyttäytyminen pyrkien estämään samankaltaisten kyberhyökkäysten kohdistumista tietojärjestelmiin (Djenna ym., 2023). Analysointimenetelmät voidaan yleisesti jakaa staattiseen ja dynaamiseen analyysiin sekä näiden yhdistelmään, hybridianalyysiin (Song ym., 2024).

Staattinen analyysi pyrkii havaitsemaan haittaohjelmat analysoimalla tiedoston sisältöä ja rakennetta suorittamatta itse koodia. Staattisia piirteitä ovat esimerkiksi ohjelmointirajapinnan kutsut (engl. *application programming interface*, API), operaatiokoodit, Portable Executable -otsikot (PE-otsikot) ja merkkijonot. Nämä ominaisuudet antavat tärkeää tietoa, jonka perusteella haittaohjelmia voidaan tunnistaa. (Song ym., 2024.) Staattinen menetelmä on yksikertainen, nopea ja tehokas tunnettuja haittaohjelmia vastaan. Se ei kuitenkaan aina toimi odotetusti, sillä edistyneet haittaohjelmat, kuten polymorfiset variantit, voivat automaattisesti muokata itseään toimimaan haitallisesti suorituksen

aikana. Tämä heikentää staattisten analyysin tehokkuutta ja toimivuutta, sillä menetelmä ei sisällä tietoa haittaohjelman jatkuvasta toiminnasta. (Apruzzese ym., 2023; Djenna ym., 2023.)

Dynaaminen analyysi vie enemmän aikaa verrattuna staattiseen analyysiin, sillä tarkkailee haittaohjelman toimintaa suorituksen aikana. Yleensä analysointi suoritetaan reaaliajassa hyödyntämällä kontrolloitua ympäristöä, kuten virtuaalikonetta, ohjelmiston toimintojen seuraamiseen (Akhtar & Feng, 2022). Dynaamisia piirteitä ovat esimerkiksi järjestelmäkutsut sekä verkkoliikenne- ja rekisterimuutokset. Tämä menetelmä tarjoaa tärkeää tietoa ohjelmien haitallisesta käytöksestä suorituksen aikana sekä sen myötä edistää merkittävästi edistyneiden ja monimutkaisten haittaohjelmien tunnistusta. (Damodaran ym., 2017.)

Hybridianalyysissa yhdistyvät staattisen ja dynaamisen analyysin hyödyt. Ohjelmistoa tarkastellaan aluksi staattista analyysia hyödyntäen, jolloin etsitään haittaohjelman tunnisteet. Tämän jälkeen ohjelma voidaan suorittaa sen todellisen käyttäytymisen analysoimiseksi. (Sharma & Arora, 2021.) Hybridianalyysi antaa monipuolisesti tietoa ohjelmiston toiminnasta perustuen sekä staattisiin että dynaamisiin piirteisiin.

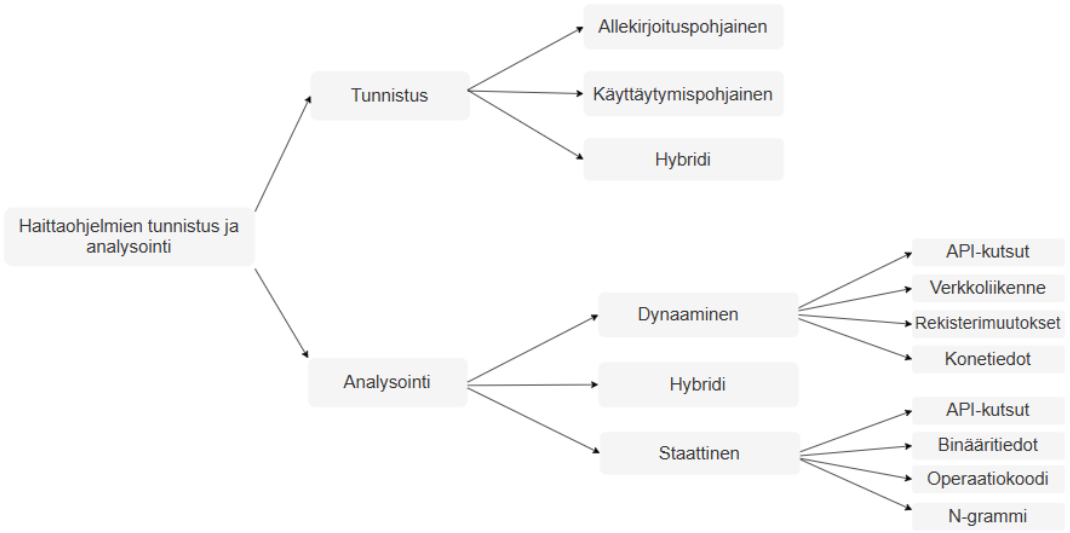
2.3 Perinteiset haittaohjelmien tunnistusmenetelmät

Haittaohjelmien havaitsemisen ja torjunnan tarkoitus on suojata järjestelmää haitallisilta hyökkäyksiltä. Kyberuhkien varhainen havaitseminen on keskeistä aiheutuvien vahinkojen minimoimiseksi. Haittaohjelmien tunnistusmenetelmät voidaan jakaa kahteen pääkategoriaan: allekirjoituspohjaisiin ja käyttäytymispohjaisiin menetelmiin. Allekirjoituspohjainen tunnistus on yleisimmin käytetty tunnistusmenetelmä virustorjuntaohjelmistoissa. (Souri & Hosseini, 2018.) Allekirjoitus on tavujono, jolla voidaan tunnistaa tietokannassa määriteltyjä haittaohjelmia. Allekirjoitukseen pohjautuva menetelmä auttaa tunnistamaan ja havaitsemaan haittaohjelmia etsimällä niihin viittaavia tunnisteita ohjelmista. Tässä menetelmässä kehittäjät käyttävät virustietokannan allekirjoituksia sekä vertailevat tutkittavan tiedoston ja tietokannan tietoja toisiinsa haittaohjelmien havaitsemiseksi. (Faruk, 2021.)

Allekirjoituspohjaisen tunnistuksen etuja ovat nopea tunnistusprosessi, helppokäyttöisyys ja laaja saatavuus. Tämä tunnistusmenetelmä on tehokas tunnettujen haittaohjelmien tunnistuksessa, mutta sen avulla ei voida tunnistaa edistyneitä haittaohjelmia johtuen menetelmän kyvystä tunnistaa vain tietokannassa määritettyjä haittaohjelmia. Allekirjoituspohjaiset tunnistusmenetelmät ovat tehottomia polymorfisia haittaohjelmia vastaan, jotka muuntavat koodiaan jokaisessa tartunnassa. (Bazrafshan ym., 2013.) Allekirjoituspohjaiset tietokannat ovat julkisia, jolloin ne ovat yleisesti tunnettuja. Tämän vuoksi haittaohjelmien kehittäjät voivat helposti kiertää tunnistamisen hyödyntäen salaus- ja harhautustekniikoita. Tietokannan säännöllinen päivitys uusien haitallisten uhkien ilmetessä voi edistää

haittaohjelmien tunnistusta tilapäisesti, mutta nykyajan kehittyneitä haittaohjelmia vastaan pelkkä allekirjoitukseen perustuva tunnistusmenetelmä ei ole riittävä. (Souri & Hosseini, 2018.) Tietokantojen manuaalinen päivitys voi kestää useita päiviä, joiden kuluessa haittaohjelmat voivat aiheuttaa vakavia seurauksia (Azeez ym., 2021).

Käyttäytymispohjainen tunnistus perustuu järjestelmän käyttäytymisen tarkkailuun prosessin suorituksen aikana. Menetelmä erottelee haittaohjelmat harmittomista ohjelmista vertailemalla niiden toimintaa. Käyttäytymispohjaisissa järjestelmissä haittaohjelmien ja harmittomien ohjelmien käyttäytymistä analysoidaan koulutusvaiheessa ja näille luodaan mallit. Tämän jälkeen testausvaiheessa suoritettava ohjelma voidaan luokitella haitalliseksi tai harmittomaksi ohjelmaksi näiden mallien perusteella. (Damodaran ym., 2017.) Käyttäytymispohjainen tunnistus on monimutkaisempi ja vie enemmän aikaa verrattuna allekirjoituspohjaiseen menetelmään, koska se tarkastelee monipuolisemmin, mitä ohjelma tekee pelkän sisällön tarkastelun sijaan. Tämä antaa syvällisempää tietoa haittaohjelman toiminnasta. Käyttäytymispohjaisella tunnistuksella voidaan tunnistaa myös kehittyneitä haittaohjelmia, jotka kykenevät suorituksen aikana generoimaan koodiaan välttääkseen tunnistuksen (Mohapatra ym., 2022). Haittaohjelmien toiminnan analysointi ja tunnistus on esitetty tiivistetysti kuvassa 1.



Kuva 1: Haittaohjelmien tunnistus ja analysointi. Kuva muokattu lähteestä Djenna ym., 2023.

3 Koneoppimismenetelmät haittaohjelmien tunnistuksessa

Teknologian kehittyessä haittaohjelmat käyttävät eri keinoja piiloutuakseen ja kiertääkseen perinteiset tunnistusmenetelmät. Tämä tekee menetelmistä riittämättömiä nykyaikaisia uhkia vastaan ja luo tarpeen tehokkaille ratkaisuille datan käsittelemiseksi ja haittaohjelmien tunnistamiseksi. Tekoälyteknologian kehittyessä tunnistusprosessit ovat muuttuneet entistä edistyneemmiksi ja automatisoidummiksi. (Song ym., 2024) Tässä luvussa käsitellään koneoppimisen käsite sekä sen menetelmien kehitysprosessia ja toimintaperiaatteita. Lisäksi arvioidaan koneoppimisen asemaa haittaohjelmien tunnistuksessa ottaen huomioon sen tarjoamat edut ja haasteet.

3.1 Koneoppiminen

Koneoppiminen (engl. *machine learning*) on tekoälyn osa-alue, joka keskittyy kehittämään algoritmeja ja tilastollisia malleja, jotka pystyvät analysoimaan dataa ja tekemään sen pohjalta päätöksiä. Koneoppiminen käyttää erilaisia algoritmeja riippuen ratkaistavasta ongelmasta ja oppimisprosessiin liittyvistä muuttujista. (Ozkan-Ozay ym., 2024.) Tämä tutkielma on rajattu käsittelemään ohjattuun, ohjaamattomaan ja puoliohjattuun oppimiseen perustuvia koneoppimismenetelmiä.

Ohjatussa oppimisessa malli koulutetaan merkityn datan pohjalta, eli jokaiselle annetulle syötteelle on etukäteen määritelty oikea tulos. Ohjattu oppiminen vaatii kehittäjän ohjausta prosessin aikana datan merkitsemiseen ja algoritmin toiminnan sääntöjen määrittämiseen. Algoritmi oppii aiemmasta datasta ja voi hyödyntää sitä uusissa tilanteissa. (Ozkan-Ozay ym., 2024.) Haittaohjelmien tunnistuksessa koulutusdata sisältää sekä haittaohjelmien että harmittomien ohjelmien piirteitä. Koulutusdatan perusteella malli oppii luokittelemaan ohjelmat haittaohjelmiksi tai harmittomiksi ohjelmiksi. Koska malli oppii vain tarjotusta merkitystä datasta, sitä on haastavaa soveltaa koulutusaineistosta poikkeavalle tietoaineistolle (Song ym., 2024). Ohjatun oppimisen mallit kykenevät kuitenkin tunnistamaan haittaohjelmien variantteja perustuen tunnettujen haittaohjelmien ja uudenlaisten varianttien välisiin yhtäläisyyksiin (Ozkan-Ozay ym., 2024). Ohjattua oppimista voidaan hyödyntää luokittelu- ja regressiotehtävissä.

Ohjaamaton oppiminen soveltuu tilanteisiin, joissa hyödynnetään merkitsemätöntä dataa (Sarker ym., 2021). Menetelmässä ohjelma pyrkii etsimään malleja tuntemattomasta syötedatasta. (Song ym., 2024.) Se analysoi dataa ja pyrkii tekemään johtopäätöksiä havaitakseen siinä piilevät rakenteet. Ohjaamattomaan oppimiseen perustuvia algoritmeja voidaan käyttää datan analysointiin, muuttujien vähentämiseen sekä ryhmittelyyn. (Ozkan-Osay ym., 2024.) Ryhmittelymenetelmät voivat järjestellä dataa samankaltaisten piirteiden perusteella, jolloin poikkeavat ja epänormaalit käyttäytymismallit sekä luvattomat tietokäytöt erottuvat helpommin. Ohjaamaton oppiminen on erityisen tehokasta

tuntemattomien uhkien tunnistuksessa, koska se pystyy analysoimaan monimutkaista dataa ja löytämään siitä poikkeamia ilman ennakkotietoja mahdollisista uhkista (Sarker ym., 2021). Tämä korostaa ohjaamattoman oppimisen merkitystä haittaohjelmien tunnistuksessa, koska haittaohjelmat usein sisältävät uudenlaisia epäsäännönmukaisuuksia, joita ohjatun oppimisen menetelmät eivät välttämättä havaitse.

Puoliohjattu oppiminen on koneoppimisen menetelmä, joka yhdistää ohjatun ja ohjaamattoman oppimisen ominaisuuksia koneoppimismallien kouluttamiseen (Song ym., 2024). Puoliohjattu oppiminen vaatii pienen määrän merkittyä dataa, mutta pystyy samalla hyödyntämään laajasti myös merkitsemätöntä dataa. Tämä tekee siitä tehokkaan menetelmän tilanteissa, joissa datan merkitseminen on kallista tai aikaa vievää (Ozkan-Ozay ym., 2024). Puoliohjattu oppiminen yhdistää ohjatun ja ohjaamattoman oppimisen etuja ja sitä voidaan käyttää parantamaan merkittävästi oppimisprosessin tarkkuutta. Puoliohjattu menetelmä sopii luokittelu- ja ryhmittelytehtäviin (Song ym., 2024).

3.2 Kehitysprosessi ja toimintaperiaatteet

Koneoppimismallien kehitys on monivaiheinen prosessi, joka useimmissa tapauksissa alkaa tietoaineiston ja hyödynnettävän koneoppimismenetelmän määrityksellä. Tämän jälkeen keskeisiä vaiheita ovat aineiston esikäsittely, piirteiden valinta, mallin koulutus ja testaus sekä sen toiminnan arviointi. Kuvassa 2 on esitetty kehitysprosessin keskeisimmät vaiheet.

Kuva 2: Koneoppimismenetelmän kehitysprosessi

Koneoppimismenetelmien toiminta perustuu hyödynnettävään tietoaineistoon ja siitä opittaviin malleihin, joiden perusteella voidaan tunnistaa haittaohjelmat ja tarjota ennusteita uusista ohjelmista (Azeez ym., 2021). Hyödynnettävän datan tulee olla monipuolista sisältäen tietoa sekä haitallisista että harmittomista ohjelmista. Datan tulee kuvastaa tarkasti todellista ympäristöä ja sisältää riittävästi vaihtelua, jotta koneoppimismallit voivat oppia tunnistamaan monipuolisesti eri haittaohjelmia. (Abhijna ym., 2024.)

Tietoaineisto jaetaan esikäsittelyssä koulutus- ja testiaineistoihin sekä muunnetaan sellaiseen muotoon, että koneoppimismallit pystyvät hyödyntämään sitä tehokkaasti (Mankar ym., 2024). Piirteiden suodattaminen vain tunnistuksen kannalta tärkeisiin piirteisiin on ratkaisevaa virheiden minimoinnin ja

laskennallisen tehon optimoinnin kannalta (Abhijna ym., 2024). Toisaalta, jos piirteitä karsitaan liikaa, malli ei opi tunnistamaan olennaisia eroja haittaohjelmien ja harmittomien ohjelmien välillä. (Akhtar & Feng, 2022.) Piirteiden valinnassa voidaan hyödyntää koneoppimismenetelmiä luokittelemaan piirteitä niiden sisältämän informaation perusteella (Elkilany & Chu, 2025). Piirteitä analysoimalla koneoppimismalli oppii tunnistamaan haittaohjelmille tyypillisiä malleja ja tekemään ennusteita tuntemattomasta datasta. Haittaohjelmien tunnistuksessa hyödynnettäviä piirteitä ovat pääasiassa tiedoston metatiedot ja käyttäytymiseen liittyvät tiedot (Song ym., 2024).

Kun tietoaineisto on purettu piirteiksi, ne muunnetaan matemaattisiksi arvoiksi piirrevektoreiden avulla. Malli oppii erottamaan haittaohjelmat harmittomista näytteistä analysoimalla ja vertailemalla niiden piirrevektoreita. (Song ym., 2024.) Koneoppimismenetelmien luokittelu- ja tunnistusprosessi voidaan jakaa koulutus- ja testausvaiheeseen (Abhijna ym., 2024). Koulutusvaiheessa data luokitellaan haitalliseksi ja harmittomaksi. Algoritmeja voidaan myös kouluttaa luokittelemaan haittaohjelmia niiden tyypin mukaan, jolloin mahdollisia luokkia voi olla useampia. Testausvaiheessa koneoppimismalli saa syötteeksi tuntemattoman tietoaineiston, jonka sisältämät ohjelmat se luokittelee koulutusvaiheessa oppimiensa mallien perusteella. (Abhijna ym., 2024.)

Mallien käyttökelpoisuutta voidaan arvioida niiden vaatimien resurssien perusteella. Koneoppimismenetelmien koulutus- ja suoritusaikaan vaikuttaa esimerkiksi mallin monimutkaisuus, piirteiden lukumäärä, tietoaineiston koko ja käytettävissä oleva laitteisto (Azeem ym., 2024). Koneoppimismenetelmän suorituskykyä arvioidaan myös tulosmittareiden avulla, joiden tulokset antavat tärkeää informaatiota mallien toiminnasta ja tarvittavasta jatkokehityksestä. Tutkimuksissa yleisimmin käytössä olevia mittareita ovat täsmällisyys (engl. *accuracy*), tarkkuus (engl. *precision*), herkkyys (engl. *recall*) ja F1-tulos (engl. *F1-score*). Täsmällisyys kuvaa prosenttiosuutta kaikista oikein luokitelluista tapauksista sisältäen sekä haittaohjelmien että harmittomien ohjelmien luokittelun. Tarkkuus ilmaisee oikein tunnistettujen haittaohjelmien osuuden kaikista koneoppimismallin haittaohjelmiksi luokittelemista ohjelmista. Herkkyys määrittää osuuden, kuinka monta haittaohjelmaa tunnistetaan kaikista aineistoissa olevista haittaohjelmista. F1-tulos on tarkkuuden ja herkkyyden harmoninen keskiarvo. (Elkilany & Chu, 2025; Mankar ym. 2024.) Lisäksi koneoppimismalleja voidaan arvioida väärien positiivisten ja negatiivisten tulosten perusteella. Väärä positiivinen kuvaa tilannetta, jossa koneoppimismenetelmä luokittelee ohjelman haitalliseksi, vaikka se on todellisuudessa harmiton. Väärä negatiivinen viittaa puolestaan tapaukseen, jossa haittaohjelma luokitellaan harmittomaksi ja jää tällöin havaitsematta. (Patil & Deng, 2020.) Tarkkuuden kasvaessa väärien positiivisten määrä pienenee. Samoin herkkyyden ja väärien negatiivisten arvojen välillä on yhteys: mitä korkeampi herkkyys, sitä vähemmän järjestelmä jättää tunnistamatta todellisia haittaohjelmia. (Apruzze ym., 2023.)

3.3 Rooli tunnistuksessa

Koneoppimismenetelmiä hyödynnetään haittaohjelmien luokittelussa ja tunnistuksessa yhä enemmän, koska ne mahdollistavat kehittyneiden tunnistusmallien luomisen. Viime vuosina koneoppimisen hyödyntäminen kyberturvallisuudessa on edistynyt ja sen roolin odotetaan kasvavan entisestään kyberuhkien yleistymisen ja monimutkaistumisen myötä. (Patil & Deng, 2020.) Toisin kuin staattisiin allekirjoituksiin perustuvat perinteiset menetelmät, koneoppimismenetelmät analysoivat ohjelmistojen toimintaa tunnistaakseen sekä tunnetut haittaohjelmat että niiden kehittyneet variantit. Tämä tekee koneoppimisesta erityisen tehokkaan työkalun kehittyvien kyberuhkien torjunnassa (Nour & Said, 2024). Koneoppimista hyödyntäviä viruksentorjuntaohjelmistoja ovat esimerkiksi Norton, McAfee ja Avast Antivirus (Nachaat, 2023).

Koneoppimismenetelmiä sovelletaan tunnistusjärjestelmissä ohjelmistojen toiminnan analysoinnissa ja haitallisen toiminnan havaitsemisessa (Nour & Said, 2024). Algoritmit käsittelevät suuria tietomääriä koulutusvaiheessa tunnistaakseen kaavoja ja poikkeamia, jotka viittaavat kyberuhkiin. Testausvaiheessa nämä opitut mallit auttavat haittaohjelmien havaitsemisessa (Ozkan-Ozay ym., 2024). Koneoppimismenetelmien kehittäminen vaatii suuren määrän laadukasta dataa ja näiden menetelmien toimivuus riippuu datasta, jolla ne on koulutettu ja testattu. Suurikokoisen tietojoukon käsittely vaatii tehokkaita laskentaresursseja, optimoituja algoritmeja ja sopivia tiedonkäsittelymenetelmiä tarkkojen tulosten saavuttamiseksi (Azeem ym., 2024). Koneoppimismenetelmät edellyttävät jatkuvaa mallien päivittämistä, jotta ne pysyvät ajantasaisina ja suorituskykyisinä.

Koneoppimisen hyödyntäminen on parantanut haittaohjelmien havaitsemisen tarkkuutta ja tehokkuutta merkittävästi. Koneoppimiseen perustuvat virustorjuntaohjelmat kykenevät analysoimaan suuria tietomääriä, tunnistamaan malleja sekä potentiaalisia uhkia nopeammin ja tarkemmin verrattuna perinteisiin menetelmiin. Lisäksi nämä menetelmät voivat tunnistaa myös uusia ja muuttuvia haittaohjelmia, joita perinteiset menetelmät eivät välttämättä havaitse. (Nachaat, 2023.) Koneoppimisen hyödyntäminen on keskeisessä asemassa, kun luodaan tehokkaampia tunnistusjärjestelmiä, joilla perinteisten menetelmien rajoitukset voidaan ylittää.

Yksinkertaiset koneoppimismallit ovat riittävän tehokkaita tunnistamaan osan nykyajan haittaohjelmista. Syväoppimista hyödyntävät menetelmät tulevat kehittymään tulevaisuudessa entisestään ja niiden asema haittaohjelmien tunnistuksessa tulee kasvamaan uhkien kehittyessä monimutkaisemmiksi (Song ym., 2024). Jatkuva ohjelmistojen kehitys ja laadukkaan datan hyödyntäminen ovat keskeisiä tekijöitä koneoppimispohjaisten haittaohjelmien tunnistusjärjestelmien tehokkuuden varmistamisessa.

3.4 Koneoppimisen tarjoamat edut

Koneoppimisesta on tullut keskeinen työkalu kyberturvallisuudessa, sillä tarve nopeammille, tehokkaammille ja joustavammille tunnistusmenetelmille kasvaa jatkuvasti (Ibekwe ym., 2023). Koneoppimisalgoritmit parantavat uhkien tunnistuksen tarkkuutta pystyessään tunnistamaan haittaohjelmien rakenteellisia ja toiminnallisia piirteitä, jotka saattavat jäädä ihmisanalyytikoilta havaitsematta. Lisäksi hyödyntämällä useita koneoppimismalleja samanaikaisesti voidaan saada entistä tarkempia ja luotettavampia tuloksia. (Nachaat, 2023.) Toisin kuin allekirjoituspohjaiset menetelmät, koneoppimiseen perustuvat ratkaisut eivät ole rajoittuneita aiemmin tunnettuihin uhkiin. Tämä mahdollistaa uusien ja ennestään tuntemattomien haittaohjelmien tunnistamisen tehden koneoppimisesta erityisen tehokkaan ratkaisun kehittyvien uhkien tunnistuksessa pitkällä aikavälillä. (Ozkan-Ozay ym., 2024.)

Koneoppimismenetelmät ovat nopeita kouluttaa ja suoriutuvat annetuista tehtävistä tehokkaasti. Azeem ym. (2024) tekemässä tutkimuksessa mitattiin mallien koulutukseen ja suoritukseen vaadittavaa aikaa. Tutkimuksessa satunnaismetsäalgoritmin koulutusaika 49 piirteen suhteen oli 8,08 sekuntia. Saman verran aikaa kului uuden syötteen luokittelussa. Piirteiden määrän vähentyessä koulutus- ja suoritusajat pienenivät entisestään vain muutamaan sekuntiin. (Azeem ym., 2024.) Lähes reaaliaikaisen data-analyysin avulla koneoppimisalgoritmit voivat havaita ja torjua uhkia paljon nopeammin kuin perinteiset sääntöpohjaiset järjestelmät. Tämä edistää haittaohjelmien varhaista havaitsemista, mikä on keskeistä seurausten minimoimiseksi. Koneoppimisalgoritmit kykenevät analysoimaan suuria tietomääriä, tunnistamaan malleja ja luokittelemaan ohjelmia tehokkaasti käyttäytymisen tai opittujen piirteiden perusteella. (Patel & Ghosh, 2024.)

Koneoppimisen avulla voidaan automatisoida monia aikaa vieviä tehtäviä, kuten uhkien havaitseminen ja luokittelu sekä toiminnan analysointi. Nämä vaiheet vaativat perinteisesti paljon manuaalista työtä, jolloin automatisointi vapauttaa kyberturvallisuusammattilaisilta aikaa. Automaatio vähentää myös inhimillisten virheiden riskiä ja tehostaa yleisesti kyberturvajärjestelmien toimintaa. Haittaohjelmat muuttuvat nopeasti, jolloin koneoppimismenetelmät voivat sopeutua ja oppia uusista uhkista jatkuvan koulutuksen ja päivityksen myötä. Tämä dynaaminen mukautumiskyky parantaa järjestelmän pitkän aikavälin toimivuutta ja tehokkuutta. (Nachaat, 2023.)

Koneoppimismenetelmien hyödyntäminen haittaohjelmien tunnistuksessa tarjoaa huomattavia etuja, kuten korkeamman tarkkuuden, mukautumiskyvyn muuttuviin uhkiin, reaaliaikaisen tunnistuksen ja tehokkuuden suurien datamäärien analysoinnissa. Koneoppimismenetelmien etu haittaohjelmien tunnistamisessa pohjautuu niiden kykyyn mukautua uusiin ja muuttuviin uhkiin sekä oppia jatkuvasti. Algoritmien päivitys ja uudelleenkoulutus laadukkaalla datalla mahdollistavat ennakoivan

suojautumisen ja tehokkaan reagoinnin kyberuhkiin tehden koneoppimisesta olennaisen osan nykyaikaista kyberturvallisuutta.

3.5 Koneoppimisen haasteet

Koneoppimismenetelmillä on lukuisten etujen lisäksi myös haasteita, jotka rajoittavat niiden käyttöönottoa. Nämä menetelmät vaativat suuren määrän laadukasta dataa, jonka tulee kuvastaa todenmukaisesti haittaohjelmien ominaisuuksia. Datan tulee olla monipuolista, mutta piirteiden tulee olla tarkoin määriteltyjä, jotta algoritmeista saadaan koulutettua tehokkaita ja tarkkoja samalla minimoiden väärät positiiviset ja negatiiviset tulokset. Koneoppimismenetelmät saattavat merkitä harmittomia ohjelmia haittaohjelmiksi tai olla tunnistamatta hyökkäyksiä, jos käytetty data ei ole riittävän monipuolista ja informatiivista. Lisäksi algoritmeja on jatkuvasti koulutettava uudelleen vastaamaan uusimpien haittaohjelmien piirteitä ja menetelmiä, jotta ne pysyvät tehokkaina ohjelmien kehittyessä. Tehokkaiden koneoppimiseen pohjautuvien tunnistusmenetelmien kehitys on monimutkaista, ja se vaatii paljon aikaa ja resursseja. (Akhtar, 2022.) Lisäksi suureen datan tarpeeseen, sekä koulutus- että testausvaiheessa, liittyy monia tietoturvakysymyksiä.

Organisaatioiden kyky hyödyntää koneoppimisen potentiaalia kärsii usein osaavan henkilöstön puutteesta. Koneoppimisen käyttöönotto haittaohjelmien tunnistuksessa vaatii laajaa teknistä osaamista, kuten datatieteiden, koneoppimisen ja kyberturvallisuuden asiantuntemusta. Teknologian ymmärtämisen puute heikentää kykyä ottaa käyttöön koneoppimismenetelmiä sekä hallita ja valvoa niiden toimintaa tehokkaasti. Koneoppimismallien käyttöönotto kyberturvallisuudessa tuo yleensä mukanaan korkeita kustannuksia, mikä vaikuttaa erityisesti pieniin ja keskisuuriin organisaatioihin. Laitteistojen, ohjelmistojen ja henkilöstön kustannukset voivat nousta nopeasti suuriksi, mikä vaikeuttaa organisaatioiden kykyä ottaa käyttöön ja ylläpitää näitä järjestelmiä. (Nachaat, 2023.)

Koneoppimismallien ylläpitäminen ajan tasalla ja tehokkaina kehittyviä uhkia vastaan ovat jatkuvia haasteita. Koneoppiminen luo myös lisää mahdollisuuksia haitallisille toimijoille hyödyntää koneoppimista entistä vaikeammin havaittavien haittaohjelmien kehittämisessä sekä väistämistekniikoiden luonnissa. (Ozkan-Ozay ym., 2024.) Haasteisiin vastaaminen on keskeistä koneoppimisen tehokkaan hyödyntämisen ja käyttöönoton varmistamiseksi. Laitteistoihin, ohjelmistoihin ja henkilöstön kouluttamiseen investointi parantaa osaamista ja kehittyneen teknologian ymmärtämistä. Kun tekoälymenetelmät kehittyvät ja niiden rakenteet muuttuvat monimutkaisemmaksi, osaamisen ja koulutuksen merkitys kasvaa yhä tärkeämmäksi.

4 Koneoppimisalgoritmien suorituskyky

Tässä luvussa tarkastellaan viimevuotisia tutkimuksia, joissa koneoppimista on hyödynnetty haittaohjelmien tunnistuksessa eri suoritusympäristöissä. Tarkastellut tutkimukset ja niissä käytetyt tietoaineistot, piirteet ja algoritmit on esitetty taulukossa 1. Tutkimusten tuloksia arvioidaan neljän tulosmittarin, täsmällisyyden, tarkkuuden, herkkyyden ja F1-tuloksen, avulla. Kunkin tutkimuksen suorituskyvyltään parhaimman algoritmin tulokset on tiivistetty taulukkoon 2. Osassa tutkimuksista käsiteltiin myös syväoppimiseen perustuvia algoritmeja, mutta nämä algoritmit ja niiden tulokset on rajattu pois tästä tarkastelusta tutkielman keskittyessä vain matalan tason koneoppimismalleihin. Lisäksi tutkimukset saattavat sisältää muitakin vertailuja, mutta tässä osiossa käsitellään vain algoritmien suorituskykyä tunnistaa haittaohjelmia ja luokitella niitä tyyppinsä mukaan. Tutkimusten käyttämät tietoaineistot ja piirteet poikkeavat toisistaan, jolloin tulokset eivät ole täysin vertailukelpoisia. Niiden pohjalta voidaan kuitenkin tehdä suuntaa-antavia johtopäätöksiä koneoppimisalgoritmien suorituskyvystä haittaohjelmien tunnistuksessa nykypäivänä.

4.1 Tarkastellut tutkimukset

Kumar ym. (2024) tutkivat haittaohjelmien tunnistusta kuudella eri koneoppimismenetelmällä. Tutkimuksessa hyödynnettiin CIC-MalMem-2022-tietoaineistoa, joka koostuu 58 596 tiedostosta. Näistä puolet sisältävät haittaohjelmia ja puolet ovat harmittomia näytteitä. Haittaohjelmat koostuivat kiristys- ja vakoiluohjelmista sekä troijalaisista. Alun perin tietoaineisto sisälsi 57 piirrettä, joista lopulta tutkimukseen valittiin 18. Keskeisimmät piirteet valittiin käyttämällä korrelaatiopohjaista piirteenvalintaa, minkä tarkoituksena oli vähentää piirteitä paremman suorituskyvyn ja tunnistustarkkuuden saavuttamiseksi. Tietoaineisto jaettiin satunnaisesti 80 % koulutusdataan ja 20 % testidataan, ja algoritmien suorituskykyä tarkkailtiin sekä koulutus- että testausvaiheessa.

Azeem ym. (2024) tekemä tutkimus perustuu UNSW-NB15-tietoaineistoon, joka koostuu 2 540 044 reaaliaikaisesta verkkotapahtumasta, joista osa on normaaleja ja osa viittaa haittaohjelmiin. Tietojoukko sisältää yhdeksän eri hyökkäystyyppiä. Tutkimuksessa koneoppimismallien suorituskykyä haittaohjelmien tunnistuksessa ja luokittelussa arvioitiin neljässä eri tapauksessa, joista ensimmäisessä algoritmien toiminnan tarkastelussa hyödynnettiin jokaista 49:ää piirrettä. Seuraavaksi malleja testattiin poistamalla piirrejoukosta 10 tärkeintä piirrettä. Tämän jälkeen piirrejoukosta vähennettiin yhteensä 20 keskeisintä piirrettä. Piirteiden valinnassa hyödynnettiin entropiaan perustuvaa valintamenetelmää. Viimeisessä vaiheessa poistetiin satunnaisesti piirteitä ja arvioitiin koneoppimismallien suorituskykyä jäljelle jääneiden piirteiden perusteella. Piirteiden lukumäärään ei otettu kantaa. Tutkimuksessa ei myöskään käsitelty sitä, miten tietoaineisto on jaettu koulutus- ja testidataan.

Patel ja Ghosh (2024) kehittivät Android-haittaohjelmien tunnistukseen tarkoitetun koneoppimiseen perustuvan menetelmän nimeltä AMD-XAI-ML. Mallia hyödynnettiin binääriluokittelussa, jossa haittaohjelmat pyrittiin erottamaan harmittomista ohjelmista. Tutkimuksessa käytettiin CICAndMal2019-tietoaineistoa, joka jaettiin 75 % koulutusdataan ja 25 % testidataan. Tietoaineisto sisälsi sekä staattisia että dynaamisia piirteitä. Tutkimuksessa hyödynnettiin selitettävää tekoälyä (engl. *explainable artificial intelligence*, XAI) tulkitsemaan haittaohjelmien tunnistuksen kannalta keskeisiä piirteitä. Selitettävän tekoälyn analyysi keskittyi 20 keskeisimpään piirteeseen, mutta tutkimuksessa ei mainita tunnistuksessa hyödynnettyjen piirteiden kokonaismäärää.

Azeez ym. (2021) tutkimuksessa arvioitiin viiden koneoppimisalgoritmin tarkkuutta haittaohjelmien tunnistuksessa. Tutkimuksessa hyödynnettiin Windows PE -tiedostoista kerättyä dataa, joka koostui haitallisista ja harmittomista ohjelmista. Algoritmien tarkoitus oli erottaa haittaohjelmat harmittomista ohjelmista ilman haittaohjelmien tarkempaa luokittelua. Tietoaineisto jaettiin 80 % koulutusdataan ja 20 % testidataan. Aineisto sisälsi alun perin 77 piirrettä, joista lopulta tarkasteltiin 55:tä keskeisintä, jotka osoittautuivat merkityksellisimmiksi tiedoston haitallisuuden tai harmittomuuden tunnistamisessa. Keskeisimpiä piirteitä olivat muun muassa osioiden lukumäärä, sisääntulopisteen osoite ja käyttöjärjestelmän pääversio.

Elkilany ja Chu (2025) tekemässä tutkimuksessa arvioitiin koneoppimismallien hyödyntämistä haittaohjelmien reaaliaikaisessa tunnistuksessa ja luokittelussa. Tutkimuksen ensimmäisessä osassa tarkasteltiin koneoppimismenetelmien suorituskykyä haittaohjelmien tunnistuksessa. Algoritmien koulutuksessa ja testauksessa hyödynnettiin Kaggle-sivustolta löytyvää julkista PE-header data -tietoaineistoa. Tunnistuksessa analysoitiin 54:ää piirrettä ja niihin liittyviä arvoja: entropiaa, keskiarvoa ja keskihajontaa. Tutkimuksen toisessa osassa algoritmien suorituskykyä arvioitiin eri haittaohjelmatyyppien luokittelussa. Tarkoituksena oli luokitella haittaohjelmat niiden tyyppin mukaan perustuen haittaohjelmatiedoston sisältöön ja piirteisiin. Tietoaineistona käytettiin Microsoftin tietoaineistoa (*Microsoft malware classification challenge BIG 2015)*. Se sisältää 10 868 haittaohjelmatiedostoa, jotka edustavat yhdeksää eri haittaohjelmatyyppiä. Tutkimuksen kummassakin osassa tietoaineistosta 80 % käytettiin mallien koulutukseen ja 20 % mallien testaukseen.

Taulukko 1: Tutkimusten hyödyntämät tietoaineistot ja niiden jako koulutus- ja testidataan sekä tarkasteltavien piirteiden lukumäärä ja käytetyt luokittelijat.

Artikkeli	**Tietoaineisto**	**Tietoaineiston jako**	**Piirteiden lukumäärä**	**Luokittelijat**
Kumar ym., 2024	CIC-MalMem-2022	80 % koulutukseen, 20 % testaukseen	18	Päätöspuu, satunnaismetsä, K-lähin naapuri, logistinen regressio, Naïve Bayes, tukivektorikone
Azeem ym., 2024	UNSW-NB15	-	1. 49 2. 39 3. 29 4. -	Päätöspuu, satunnaismetsä, K-lähin naapuri, logistinen regressio, Extra tree
Patel & Ghosh, 2024	CICAndMal2019	75 % koulutukseen, 25 % testaukseen	-	Päätöspuu, satunnaismetsä, K-lähin naapuri, logistinen regressio, Extra tree, Naïve Bayes, gradienttivahvistus, tukivektorikone, XGBoost
Azeez ym., 2021	Windows PE -tiedostoista kerätty data	80 % koulutukseen, 20 % testaukseen	55	Päätöspuu, satunnaismetsä, Naïve Bayes, gradienttivahvistus, AdaBoost
Elkilany & Chu, 2025	PE-header data Kaggle ja Microsoft malware classification challenge BIG 2015	80 % koulutukseen, 20 % testaukseen	54	Satunnaismetsä, K-lähin naapuri, Naïve Bayes, gradienttivahvistus, tukivektorikone

4.2 Suorituskyky

Kumar ym. (2024) tekemässä tutkimuksessa mallien tarkkuutta arvioitiin koulutus- ja testivaiheissa erikseen. Koulutusvaiheessa K-lähin naapuri -algoritmi suoriutui parhaiten täsmällisyydellä 99,95 % ja tarkkuudella 99,97 %, päätöspuun arvot jäi hieman alhaisemmaksi niiden ollessa 99,94 % ja 99,96 %. Loputkin tarkastellut algoritmit saavuttivat yli 99 %:n täsmällisyyden. Testausvaiheessa K-lähin naapuri- ja päätöspuualgoritmit saavuttivat 99,98 %:n täsmällisyyden ja 100 %:n tarkkuuden. Korkea tarkkuusarvo osoittaa, että kaikki haittaohjelmiksi tunnistetut ohjelmat olivat todellisuudessakin haittaohjelmia. 99,97 %:n herkkyyden perusteella vain pieni osa haittaohjelmista jäi tunnistamatta. Muidenkin koneoppimismenetelmien tulokset olivat testivaiheessa lupaavia kaikkien saavuttaessa vähintään 99 %:n täsmällisyyden ja herkkyyden sekä 98,6 %:n tarkkuuden. Tämä tukee kaikkien tarkasteltujen algoritmien käyttökelpoisuutta haittaohjelmien tunnistuksessa. Tutkimuksen tulokset osoittavat, että erityisesti K-lähin naapuri- ja päätöspuualgoritmit toimivat hyvin, kun tietoaineiston piirteet on tarkasti määritelty. Tutkimuksen perusteella kaikki piirteet eivät ole yhtä merkityksellisiä haittaohjelmien tunnistuksen kannalta ja tarkka piirteiden rajaaminen voi parantaa algoritmien suorituskykyä ja tarkkuutta.

Azeem ym. (2024) tutkimuksessa Extra tree- ja satunnaismetsäalgoritmit saavuttivat parhaimmat tulokset. Tutkimuksen ensimmäisessä osassa tarkasteltiin jokaista 49:ää piirrettä, jolloin satunnaismetsäalgoritmi saavutti suurimman täsmällisyyden, sen ollessa 97,68 %. Kun analyysissa poistetiin 10 merkittävintä piirrettä, satunnaismetsä säilyi edelleen suorituskyvyltään tarkimpana algoritmina, mutta täsmällisyys laski 97,35 %:iin. Suorituskyky laski myös muilla malleilla hieman, mutta pudotus ei ollut merkittävä. Kun 20 keskeisintä piirrettä poistettiin, koneoppimismallien täsmällisyys laski edelleen erityisesti lineaarisilla malleilla. Satunnaismetsä säilytti kuitenkin suhteellisen korkean, 96,78 %:n, täsmällisyyden, mutta tulokset korostavat sitä, että tärkeiden piirteiden poistaminen vaikuttaa negatiivisesti mallien suorituskykyyn. Tutkimuksen viimeisessä vaiheessa poistettiin satunnaisesti tuntematon määrä piirteitä, jolloin Extra tree -algoritmi saavutti jokaisella tarkastellulla mittarilla 99,98 %:n tuloksen. Myös muiden algoritmien suorituskyky parani huomattavasti, ja jokainen algoritmi saavutti yli 99 %:n tuloksen kaikilla mittareilla. Tämä viittaa siihen, että piirteiden valinta satunnaisesti voi antaa hyödyllisiä tuloksia. Extra tree- ja satunnaismetsäalgoritmit suoriutuivat tutkimuksen jokaisessa vaiheessa parhaiten, mikä korostaa niiden soveltuvuutta ja tehokkuutta tunnistuksessa eri kokoisiin piirrejoukkoihin perustuen. Tutkimuksen tulokset osoittavat, että piirteiden valinta on keskeinen tekijä koneoppimisalgoritmin suorituskyvyn parantamisessa.

Patel ja Ghosh (2024) havaitsivat tutkimuksessaan, että XGBoost-algoritmi suoriutui tarkimmin, saavuttaen 98,54 %:n täsmällisyyden, tarkkuuden, herkkyyden ja F1-arvon. Extra tree- ja satunnaismetsäalgoritmien tulokset jäivät hieman alhaisimmiksi, täsmällisyysarvojen ollessa 98,52 % ja

98,42 %. Tulokset osoittavat, että kyseiset algoritmit tunnistavat haittaohjelmat tarkasti samalla minimoiden väärät positiiviset ja negatiiviset tapaukset. Tutkimuksen tulokset tukevat useimpien koneoppimismallien käyttökelpoisuutta Android-haittaohjelmien tunnistuksessa. Tarkastelluista algoritmeista heikoiten suoriutui Naïve Bayes -luokitin 70,29 %:n täsmällisyydellä. Tutkimuksessa hyödynnetty selitettävä tekoäly auttaa ymmärtämään tunnistusprosessin kannalta keskeisimpiä ominaisuuksia, mikä parantaa koko tunnistusprosessin ja tulosten tulkittavuutta.

Azeez ym. (2021) tutkimuksessa satunnaismetsäalgoritmi saavutti 99,24 %:n täsmällisyyden sekä 99 %:n tarkkuuden ja 98 %:n herkkyyden. Täsmällisyys viittaa siihen, että algoritmin virheellisesti luokittelemien ohjelmien osuus oli hyvin vähäinen. Tutkimus osoittaa, että satunnaismetsäalgoritmi on käyttökelpoinen, kun tarkasteltavia ominaisuuksia on runsaasti. Päätöspuu- ja gradienttivahvistusalgoritmi suoriutuivat tarkasti saavuttaen yli 98 %:n täsmällisyyden ja 99 %:n tarkkuuden. Algoritmien tuottamat herkkyysarvot jäivät huomattavasti alhaisimmiksi niiden ollessa 95 % ja 93 %. Tämä osoittaa, että näiltä menetelmiltä jäi satunnaismetsäalgoritmiin verrattuna enemmän haittaohjelmia tunnistamatta. Huomattavasti heikoiten algoritmeista suoriutui Naïve Bayes, täsmällisyyden ollessa vain 32,53 %.

Elkilany ja Chu:n (2025) tekemän tutkimuksen ensimmäisessä osassa arvioitiin koneoppimisalgoritmien tarkkuutta tunnistaa haittaohjelmat harmittomista ohjelmista. Parhaiten suoriutui satunnaismetsä saavuttaen lupaavia tuloksia: 99,67 %:n täsmällisyyden, 100 %:n tarkkuuden ja 99,03 %:n herkkyyden. Tämä osoittaa, että algoritmi suoriutui tarkasti erottamaan haittaohjelmat ja harmittomat ohjelmat toisistaan, samalla minimoiden väärät positiiviset ja negatiiviset tulokset. K-lähin naapuri- ja gradienttivahvistusalgoritmi suoriutuivat hieman heikommin, mutta saavuttivat yli 99 %:n täsmällisyyden. Tämä osoittaa myös näiden algoritmien käyttökelpoisuuden haittaohjelmien tunnistuksessa. Heikoiten suoriutuivat tukivektorikone ja Naïve Bayes 66,73 %:n täsmällisyysarvolla. Tutkimuksen toisessa osassa arvioitiin koneoppimismenetelmien suorituskykyä haittaohjelmien luokittelussa yhdeksän eri haittaohjelmatyypin mukaan. Haittaohjelmien luokittelussa gradienttivahvistusalgoritmi suoriutui parhaiten saavuttaen 99,47 %:n täsmällisyyden, 98,33 %:n tarkkuuden ja 97,52 %:n herkkyyden. Luokittelun tulokset olivat yleisesti hieman heikompia kuin pelkän haittaohjelman tunnistuksen tulokset, mutta osoittavat, että monet koneoppimismallit ovat käyttökelpoisia myös haittaohjelmatyyppien luokittelussa. Heikoiten luokittelusta suoriutui tukivektorikone, täsmällisyyden ollessa vain 59,98 %.

Taulukko 2 Tarkasteltujen tutkimusten tarkimmat luokittelijat sekä niiden suorituskyvyn täsmällisyys-, tarkkuus-, herkkyys- ja F1-arvot.

Artikkeli	Tarkin luokittelija	Täsmällisyys	Tarkkuus	Herkkyys	F1-tulos
Kumar ym., 2024	1. K-lähin naapuri 2. K-lähin naapuri ja päätöspuu	1. 99,95 % 2. 99,98 %	1. 99,97 % 2. 100 %	1. 99,92 % 2. 99,97 %	1. 99,94 % 2. 99,98 %
Azeem ym., 2024	1. Satunnaismetsä 2. Satunnaismetsä 3. Satunnaismetsä 4. Extra tree	1. 97,68 % 2. 97,35 % 3. 96,78 % 4. 99,98 %	1. 97,68 % 2. 97,35 % 3. 96,77 % 4. 99,98 %	1. 97,68 % 2. 97,35 % 3. 96,78 % 4. 99,98 %	1. 97,68 % 2. 97,35 % 3. 96,78 % 4. 99,98 %
Patel & Ghosh, 2024	XGBoost-algoritmi	98,54 %	98,54 %	98,54 %	98,54 %
Azeez ym., 2021	Satunnaismetsä	99,24 %	99 %	98 %	98 %
Elkilany & Chu, 2025	1. Satunnaismetsä 2. Gradienttivahvistus	1. 99,67 % 2. 99,47 %	1. 100 % 2. 98,33 %	1. 99,03 % 2. 97,52 %	1. 99,51 % 2. 97,93 %

4.3 Johtopäätökset

Tarkasteltujen tutkimusten perusteella koneoppimismenetelmiä hyödynnetään laajasti haittaohjelmien tunnistuksessa. Keskeisin käyttökohde on ohjelmien luokittelu haitallisuuden ja harmittomuuden sekä haittaohjelmatyypin mukaan. Tutkimusten perusteella voidaan todeta, että koneoppimismenetelmät antavat hyvin tarkkoja ja luotettavia tuloksia haittaohjelmien tunnistuksessa. Tutkimusten tulosten välillä ei ole suuria eroavaisuuksia, mikä viittaa siihen, että yleisellä tasolla koneoppimismallien suorituskyky on hyvä. Erityisesti puupohjaiset algoritmit osoittivat käyttökelpoisuutensa ja tarkkuutensa. Satunnaismetsäalgoritmi antoi lupaavia tuloksia jokaisessa tutkimuksessa, mikä korostaa sen soveltuvuutta haittaohjelmien tunnistukseen. Satunnaismetsä suoriutui tarkasti erityisesti tilanteissa, joissa dataominaisuuksia oli runsaasti.

K-lähin naapuri, päätöspuu, gradienttivahvistus, Extra tree ja XGBoost -algoritmit antoivat myös lupaavia tuloksia. XGBoost-algoritmin suorituskykyä arvioitiin vain yhdessä ja Extra tree -algoritmin kahdessa tutkimuksessa, joten niiden soveltuvuudesta muihin tapauksiin ei voida tehdä johtopäätöksiä. Näiden algoritmien tulosten tarkkuuden varmistamiseksi tarvitaan laajempaa tutkimusta niiden

suorituskyvystä. Tutkimuksissa yleisesti heikoiten suoriutui Naïve Bayes -algoritmi, joka saavutti useassa tutkimuksessa merkittävästi alhaisempia tuloksia verrattuna muihin malleihin.

Yksikään tarkasteltu koneoppimisalgoritmi ei tarjonnut 100 %:n täsmällisyyttä, mikä viittaa siihen, että tarkimmatkaan ratkaisut eivät tarjoa täydellistä suojaa haittaohjelmilta. Mallien tarkkuuteen vaikuttavat suuresti määritellyt piirteet. Piirteiden lukumäärä ei suoraan vaikuta suorituskykyyn, vaan keskeistä on piirteiden tarkka rajaus ja niiden sisältämä informaatio. Tarkimmat tulokset saavutetaan sekä staattisia että dynaamisia piirteitä hyödyntäen, koska hybridianalyysi kuvaa kattavasti sekä haittaohjelmien ominaisuuksia että käyttäytymistä.

Koneoppimismenetelmien rooli haittaohjelmien tunnistuksessa tulee kasvamaan entisestään tulevaisuudessa. Koneoppiminen on osoittanut korkeaa tarkkuutta haittaohjelmien tunnistuksessa, mutta kehitystyön on jatkuttava, jotta menetelmien soveltuvuutta ja tarkkuutta voidaan tulevaisuudessa parantaa. Tarvitaan lisätutkimusta koneoppimismallien suorituskyvystä reaaliaikaisessa käytössä. Erityisesti on tutkittava, miten koneoppimismalleja saadaan päivitettyä reaaliajassa uusien uhkien ilmaantuessa. Lisäksi tutkimukset keskittyvät pääosin ohjatun oppimisen algoritmeihin, joten ohjaamattomaan ja puoliohjattuun oppimiseen perustuvien menetelmien suorituskyvystä reaalitilanteissa tarvitaan lisätutkimusta. Tulevaisuuden tutkimus tulee keskittymään yhä enemmän syväoppimiseen ja itseohjautuviin algoritmeihin. Haittaohjelmien tunnistukseen on keskeistä kehittää menetelmiä, jotka pystyvät mukautumaan itsenäisesti ja reaaliaikaisesti tuntemattomiin uhkiin ilman manuaalisia toimenpiteitä.

5 Yhteenveto

Koneoppimisen hyödyntäminen haittaohjelmien tunnistuksessa on yleistynyt kyberuhkien määrän lisääntyessä. Tässä tutkielmassa käsiteltiin koneoppimisen soveltamista haittaohjelmien tunnistuksessa erityisesti tarkkaillen sen roolia ja suorituskykyä nykyajan kyberturvallisuudessa sekä koneoppimismenetelmien tarjoamia etuja ja haasteita perinteisiin tunnistusmenetelmiin verrattuna.

Perinteiset haittaohjelmien tunnistusmenetelmät eivät riitä vastaamaan kasvavaa tunnistustarvetta. Tämä johtuu siitä, että ne eivät usein ole tarpeeksi tehokkaita eivätkä pysty tunnistamaan kehittyneitä haittaohjelmia, jotka käyttävät edistyneitä tekniikoita piiloutuakseen suojausjärjestelmiltä. Koneoppimiseen perustuvat haittaohjelmien tunnistusmenetelmät voivat tarjota joustavampia ja tehokkaampia keinoja haittaohjelmien tunnistukseen yhä monimutkaisemmassa uhkaympäristössä.

Yksi koneoppimismenetelmien keskeisimpiä etuja on kyky käsitellä ja analysoida suurta tietomäärää tehokkaasti jopa reaaliajassa. Lisäksi koneoppimismenetelmät pystyvät tunnistamaan haittaohjelmien kehittyneitä variantteja, jotka saattaisivat jäädä perinteisiltä menetelmiltä tunnistamatta. Automatisointi vähentää inhimillisten virheiden määrää, tehostaa tunnistusprosessia ja vapauttaa resursseja muihin kyberturvallisuustehtäviin.

Koneoppimisen hyödyntäminen vaatii suuren määrän laadukasta dataa, jotta menetelmät pysyvät tehokkaina ja toimivina muuttuvassa ympäristössä. Lisäksi koneoppimismalleja tulee kouluttaa ja testata jatkuvasti, jotta voidaan varmistua niiden toimivuudesta. Koneoppimismenetelmien kehittäminen ja ylläpitäminen tuo mukanaan suuria kustannuksia ja vie resursseja sen vaatiessa monipuolista osaamista. Tehokkaimmatkaan koneoppimismenetelmät eivät toimi täydellisesti, vaan ne voivat luokitella ja tunnistaa ohjelmia virheellisesti. Tekoälyn yleistyminen luo uusia työkaluja myös haitallisille toimijoille, jotka pystyvät hyödyntämään koneoppimista kehittäessään yhä tehokkaampia ja vaikeammin havaittavia haittaohjelmia.

Tutkielmassa tarkasteltiin viittä tutkimusta, joissa koneoppimismenetelmien suorituskykyä arvioitiin eri ympäristöissä ja erilaisten piirteiden suhteen. Tuloksia arvioitiin täsmällisyyden, tarkkuuden, herkkyyden ja F1-tuloksen perusteella. Tarkasteltujen tutkimusten perusteella suorituskyvyltään parhaimmiksi koneoppimisalgoritmeiksi osoittautuivat satunnaismetsä, päätöspuu, gradienttivahvistus, Extra tree, K-lähin naapuri ja XGBoost. Erityisesti satunnaismetsäalgoritmi tuotti lupaavia tuloksia jokaisessa tarkastellussa tutkimuksessa. Tutkimusten tuloksia on keskeistä arvioida useilla eri mittareilla, jotta selviää oikein tunnistettujen ohjelmien lisäksi myös väärien negatiivisten ja positiivisten tulosten osuus. Algoritmit saavuttivat yleisesti tarkkoja tuloksia pienillä eroilla, mutta yhdessäkään tutkimuksessa ei saavutettu 100 %:n täsmällisyyttä. Tämän pohjalta voimme tehdä

johtopäätöksen, että suorituskyvyltään parhaimmatkin menetelmät tekevät virheitä eivätkä suoriudu täydellisesti haittaohjelmien tunnistuksesta.

Vaikka koneoppimismenetelmien suorituskyky on tarkkaa, ne eivät yksinään riitä takaamaan täydellistä suojaa haittaohjelmia vastaan. Näitä menetelmiä voidaan kuitenkin käyttää perinteisten suojausmallien tukena tunnistuksen tehostamiseksi. Tämänhetkiset tutkimukset perustuvat pääasiassa ohjattuun oppimiseen pohjautuvien algoritmien kykyyn luokitella ohjelmat oikein määriteltyjen piirteiden perusteella. Koneoppimisen soveltamista haittaohjelmien reaaliaikaisessa tunnistuksessa ja menetelmien kykyä reagoida tuntemattomiin ohjelmiin tulee tutkia laajemmin lisää, jotta tulevaisuudessa on mahdollista kehittää vielä tehokkaampia menetelmiä uhkien tunnistukseen.

Lähteet

Abhijna, C. D., Aishwarya, A. S., & Raghuramegowda, S. M. (2024). "Malware Detection using Machine Learning", *2024 Second International Conference on Advances in Information Technology (ICAIT)*, Chikkamagaluru, Karnataka, India, 1-5, https://doi.org/10.1109/ICAIT61638.2024.10690638

Akhtar, M. S., & Feng, T. (2022). "Malware Analysis and Detection Using Machine Learning Algorithms", *Symmetry*, *14*(11), 2304. https://doi.org/10.3390/sym14112304

Apruzzese, G., Laskov, P., Montes de Oca, E., Mallouli, W., Brdalo Rapa, L., Grammatopoulos, A. V., & Di Franco, F (2023). "The role of machine learning in cybersecurity", *Digital Threats: Research and Practice*, *4*(1), 1-38. https://doi.org/10.1145/3545574

Aslan, Ö., Ozkan-Okay, M., & Gupta, D. (2021). "Intelligent Behavior-Based Malware Detection System on Cloud Computing Environment", in *IEEE Access*, 9, 83252-83271. https://doi.org/10.1109/ACCESS.2021.3087316

Azeez, N. A., Odufuwa, O. E., Misra, S., Oluranti, J., & Damaševičius, R. (2021). "Windows PE Malware Detection Using Ensemble Learning", *Informatics*, *8*(1), 10. https://doi.org/10.3390/informatics8010010

Azeem, M., Khan, D., Iftikhar, S., Bawazeer, S., & Alzahrani, M. (2024). "Analyzing and comparing the effectiveness of malware detection: A study of machine learning approaches", *Heliyon*, *10*(1). doi: https://doi.org/10.1016/j.heliyon.2023.e23574

Bazrafshan, Z., Hashemi, Fard, S. M. H., & Hamzeh, A. (2013). "A survey on heuristic malware detection techniques", *The 5th Conference on Information and Knowledge Technology*, Shiraz, Iran, 113-120. https://doi.org/10.1109/IKT.2013.6620049

Damodaran, A., Troia, F. D., Visaggio, C. A., Austin, T. H., & Stamp, M. (2017). "A comparison of static, dynamic, and hybrid analysis for malware detection", *Journal of Computer Virology and Hacking Techniques*, *13*, 1-12. https://doi.org/10.1007/s11416-015-0261-z

Djenna, A., Bouridane, A., Rubab, S., & Marou, I. M. (2023). "Artificial Intelligence-Based Malware Detection, Analysis, and Mitigation", *Symmetry*, *15*(3), 677. https://doi.org/10.3390/sym15030677

Elkilany, A.R. & Chu, Y.B. (2025). "Elucidation on the performance of various machine learning models for real-time malware detection, malware classification and network packet screening", *Mach. Learn. Comput. Sci. Eng* 1, 9. https://doi.org/10.1007/s44379-024-00010-y

Faruk, M. J. H., Shahriar, H., Valero, M., Barsha, F. L., Sobhan, S., Khan, M. A., Whitman, M., Cuzzocrea, A., Lo, D., Rahman, A. & Wu, F. (2021). "Malware Detection and Prevention using Artificial IntelligenceTechniques", *IEEE International Conference on Big Data* Orlando, FL, USA, 2021, 5369-5377. https://doi.org/10.1109/BigData52589.2021.9671434

Ibekwe, U. U., Mbanaso, U. M. & Nnanna, N. A. (2023). "A Critical Review of The Intersection of Artificial Intelligence and Cybersecurity", *2023 2nd International Conference on Multidisciplinary Engineering and Applied Science (ICMEAS)*, Abuja, Nigeria, 1-6. https://doi.org/10.1109/ICMEAS58693.2023.10379362.

Kumar, S., Shersingh, Kumar, S., & Verma, K. (2024). "Malware classification using machine learning models". Procedia Comput. Sci. 235, C (2024), 1419–1428. https://doi.org/10.1016/j.procs.2024.04.133

Mankar, N. P., Sakunde, P. E., Zurange, S., Date, A., Borate, V., & Mali, Y. K. (2024). "Comparative Evaluation of Machine Learning Models for Malicious URL Detection", *2024 MIT Art, Design and Technology School of Computing International Conference*, Pune, India, 2024, 1-7. https://doi.org/10.1109/MITADTSoCiCon60330.2024.10575452

Mhara, M. A. A., Abdulrahman, A. A., & Baroud, A. A. (2024). "Cyber Attacks And Threats: Study Of The Types Of Cyber Attacks: Hacking, Viruses, Targeted Attacks, And Electronic Espionage", *Int. J. Electr. Eng. and Sustain.*, 22(4), 38–47. https://ijees.org/index.php/ijees/article/view/102

Mohapatra, N., Satapathy, B., Mohapatra B. & Mohanta B. K. (2022). "Malware Detection using Artificial Intelligence", *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, 1-6. https://doi.org/10.1109/ICCCNT54827.2022.9984218

Nachaat, M. (2023). "Current trends in AI and ML for cybersecurity: A state-of-the-art survey", *Cogent Engineering*, *10*(2). https://doi.org/10.1080/23311916.2023.2272358

Nour, S. M., & Said, S. A., (2024). "Harnessing the Power of AI for Effective Cybersecurity

Defense”, *6th International Conference on Computing and Informatics (ICCI)*, New Cairo - Cairo, Egypt, 98-102. https://doi.org/10.1109/ICCI61671.2024.10485059

Ozkan-Okay, M., Akin, E., Aslan, Ö., Kosunalp, S., Iliev, T., Stoyanov, I., & Beloev, I. (2024). "A Comprehensive Survey: Evaluating the Efficiency of Artificial Intelligence and Machine Learning Techniques on Cyber Security Solutions", in *IEEE Access*, 12, 12229-12256. https://doi.org/10.1109/ACCESS.2024.3355547

Patil, R., & Deng, W. (2020). "Malware Analysis using Machine Learning and Deep Learning techniques", *SoutheastCon*, Raleigh, NC, USA, 2020, 1-7. https://doi.org/10.1109/SoutheastCon44009.2020.9368268

Patel, A., & Ghosh, S. M. (2024). "AMD-XAI-ML: Android Malware Detection based on an Explainable AI using Machine Learning for Smart Computing Environment", *2024 OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0*, Raigarh, India, 1-6. https://doi.org/10.1109/OTCON60325.2024.10687629

Sarker, I. H. (2023). "Multi-aspects AI-based modeling and adversarial learning for cybersecurity intelligence and robustness: A comprehensive overview", *Security and Privacy*, *6*(5), e295. https://doi.org/10.1002/spy2.295

Sarker, I. H., Furhad, M. H., & Nowrozy, R. (2021). "Ai-driven cybersecurity: an overview, security intelligence modeling and research directions", *SN Computer Science*, *2*(3), 173. https://doi.org/10.1007/s42979-021-00557-0

Sharma, N., & Arora, B. (2021). "Data mining and machine learning techniques for malware detection ", In *Rising Threats in Expert Applications and Solutions: Proceedings of FICR-TEAS 2020*, Springer Singapore, 557-567. https://doi.org/10.1007/978-981-15-6014-9_66

Song, J., Choi, S., Kim, J., Park, K., Park, C., Kim, J., & Kim, I. (2024). "A study of the relationship of malware detection mechanisms using Artificial Intelligence", *ICT Express*, *10*(3), 632-649. https://doi.org/10.1016/j.icte.2024.03.005

Souri, A., & Hosseini, R. (2018). "A state-of-the-art survey of malware detection approaches using data mining techniques", *Human-centric Computing and Information Sciences*, *8*(1), 1-22. https://doi.org/10.1186/s13673-018-0125-x

Ucci, D., Aniello, L., & Baldoni, R. (2019). "Survey of machine learning techniques for malware analysis", *Computers & Security*, *81*, 123-147, ISSN 0167-4048. https://doi.org/10.1016/j.cose.2018.11.001