Ennakoiva kyberuhkatiedustelu tekoälyn avulla

Laura Vesamäki

Opinnäytetyö, AMK
Helmikuu 2025
Tieto- ja viestintätekniikan tutkinto-ohjelma (AMK)

jamk | Jyväskylän ammattikorkeakoulu
University of Applied Sciences

jamk

Kuvailulehti

Vesamäki, Laura

Ennakoiva kyberuhkatiedustelu tekoälyn avulla

Jyväskylä: Jyväskylän ammattikorkeakoulu. Helmikuu 2025, 72 sivua.

Tieto- ja viestintätekniikan tutkinto-ohjelma. Opinnäytetyö AMK.

Julkaisun kieli: suomi

Julkaisulupa avoimessa verkossa: kyllä

Tiivistelmä

Kyberuhkien lisääntyessä ja tietomäärän kasvaessa tarvitaan tehokkaampia keinoja ennakoimaan ja ehkäisemään tulevia uhkia. Tekoälyn avulla kyberuhkatiedustelusta pystytään tekemään entistä nopeampaa ja tehokkaampaa.

Opinnäytetyön tavoitteena oli tuottaa tutkimus, joka tarjoaa tietoa siitä, mitä on ennakoiva kyberuhkatiedustelu ja kuinka tekoälyä voidaan hyödyntää ennakoivassa kyberuhkatiedustelussa.

Opinnäytetyö toteutettiin laadullisena tutkimuksena, joka sisälsi kirjallisen tutkimuksen sekä käytännön toteutuksen. Kirjallisessa tutkimuksessa tutkittiin, millaisia asioita kyberuhkatiedusteluun kuuluu, mitkä ovat yleisimpiä uhkia sekä mitä on tekoäly. Työssä selvitettiin, miten tekoälyä voidaan käyttää kyberuhkatiedustelussa sekä millaisia tekoälymalleja on olemassa. Lisäksi etsittiin ja analysoitiin saatavilla olevia valmiita datasettejä kyberuhkatiedosta. Käytännön osuudessa käsiteltiin löydettyä dataa, arvioitiin datan ominaisuuksien tärkeyttä sekä kokeiltiin tutkittuja tekoälymalleja käsitellyn datan avulla.

Työn tuloksena syntyi kattava tutkimus ennakoivasta kyberuhkatiedustelusta ja siitä, miten tekoälyä voidaan käyttää ennakoivan kyberuhkatiedustelun kehittämisessä. Saatujen tulosten perusteella voitiin huomata, että vaikka tekoälyn käytöllä onkin etuja, sen käytöllä on myös haittoja. Käytännön osuudesta saaduista tuloksista huomattiin, että osa tekoälytyökaluista ja -malleista vaatii käytettävältä laitteistolta enemmän laskentatehoa.

Työn johtopäätöksinä voitiin pitää tekoälyn tarpeellisuutta ennakoivan kyberuhkatiedon kehittymisessä, yhteistyötä ja tiedonjakamista yli toimiala- ja maarajojen sekä automaattisten järjestelmien lisäämistä ja uusien tekoälymallien käyttöönottoa. Jatkotutkimusaiheiksi tutkimukselle löytyi uhkatoimijoiden profilointi tekoälyn avulla sekä tekoälytyökalujen integrointi nykyisiin kyberuhkatiedustelun työkaluihin.

Avainsanat (asiasanat)

Kyberuhkatiedustelu, Kyberturvallisuus, Tekoäly, Koneoppiminen, Syväoppiminen

Muut tiedot (salassa pidettävät liitteet)

-

jamk

Description

Vesamäki, Laura

Predictive Cyber Threat Intelligence using AI

Jyväskylä: JAMK University of Applied Sciences, February 2025, 72 pages.

Degree Programme in Information and Communication Technology. Bachelor´s thesis.

Permission for open access publication: Yes

Language of publication: Finnish

Abstract

As cyber threats increase, and as the amount of data grows, there is a need for more effective ways to predict and prevent future threats. With the use of AI cyber threat intelligence could be even more efficient and faster.

The aim of the thesis was to produce research of predictive cyber threat intelligence and investigate how to use AI to improve it.

The thesis carried out as a qualitative study that included a written study and a practical part. The study examined what kind of elements are included in cyber threat intelligence, what are the most common cyber threats and what is AI. The research also explored how to use AI in cyber threat intelligence and what kind of AI models there are. The study also searched for and analyzed the available datasets of threat intelligence. The practical part of the thesis included preprocessing of the found datasets, evaluating the most important features, and experimenting of the AI models with the preprocessed datasets.

The thesis resulted comprehensive research about predictive cyber threat intelligence using AI. The results showed that even though AI has its advantages, it also has disadvantages. The results of the practical part confirmed that some of the AI tools and models needed more computing power than the others.

The conclusions of the thesis were the necessity of AI usage in cyber threat intelligence, cooperation and exchange of information between industries and countries as well as more automated systems and the use of new AI models. Further research topics were profiling threats actors with AI and integration of AI tools into existing cyber threat intelligence tools.

Keywords/tags (subjects)

Cyber threat intelligence, Cyber security, Artificial intelligence, Machine learning, Deep learning

Miscellaneous (Confidential information)

-

Sisältö

Kuviot

Taulukot

Sanasto

IOC	Indicator of Compromise. Tieto, joka osoittaa, että kyberuhka on voinut tunkeutua järjestelmään. Nämä tarjoavat tärkeää tietoa tietomurrosta tai muusta tietoturvaloukkauksesta (Mitä vaarantumisindikaattorit (IOC:t) ovat? N.d).
IoT	Internet of Things. Laite, joka voidaan liittää langattomasti verkkoon (Mikä on esineiden internet (IoT)? N.d).
IP-osoite	Yksilöllinen tunniste, jonka avulla laite voidaan tunnistaa ja tavoittaa Internetissä tai paikallisessa verkossa (IP-osoite – määritelmä ja selitys n.d).
URL	Verkko-osoite, jonka avulla tietty verkkosivu tai tiedosto löytyy Internetistä (URL-osoite n.d.).
Verkkotunnus	Tunnus, joka on luotu helpottamaan verkkosivun löytämistä. Esimerkiksi google.com on verkkotunnus. (What is a domain name? | Domain name vs. URL n.d.)

1 Johdanto

1.1 Työn tausta

Kyberuhkien kehittyminen ja niiden kasvava määrä tekee perinteisistä menetelmistä riittämättömiä uhkien tunnistamiseen ja ehkäisyyn. Hyökkäystapojen muuttuessa ja uusien tapojen ilmestyessä, on erityisen tärkeää löytää uusia keinoja kyberuhkien ennakoimiseen ja tunnistamiseen, jotka ovat entisiä nopeampia ja tehokkaampia. Uusien keinojen avulla uhkiin voidaan reagoida nopeammin ja suojautua paremmin. Ennakoiva kyberuhkatiedustelu nouseekin keskeiseen rooliin modernissa kyberturvallisuudessa. (What is the future of cybersecurity? 2024.)

Samaan aikaan myös tekoäly kehittyy. Tekoälyn avulla pystytään automatisoimaan suurien datamäärien analysointia, tunnistamaan uhkia sekä ennakoimaan hyökkäysmalleja. Käyttämällä koneja syväoppimista voidaan havaita datassa hienovaraisia muutoksia ja poikkeamia, joita perinteiset menetelmät eivät pysty havaitsemaan. Tämä auttaa uusien tai muuttuvien uhkien tunnistamisessa. Koska kyberuhat ja hyökkäysmallit muuttuvat, nykyiset uhkamallit eivät aina pysty käsittelemään tuntemattomia hyökkäysmalleja tehokkaasti. (What is the future of cybersecurity? 2024.)

1.2 Työn tavoitteet ja rajaus

Tämän opinnäytetyön tavoitteena on tutkia, miten tekoälyä voidaan hyödyntää ennakoivien mallien kehittämisessä kyberuhkatiedusteluun. Tutkimuksen avulla voidaan löytää uusia ratkaisuja ja innovaatioita, joita voidaan hyödyntää sekä akateemisessa maailmassa että käytännön sovelluksissa. Työ myös tarjoaa arvokasta tietoa organisaatioille siitä, miten kyberturvallisuutta voidaan parantaa ja tehostaa tekoälyn avulla.

Uudet ja nopeasti muuttuvat uhat lisäävät organisaatioiden alttiutta kyberhyökkäyksille, sillä perinteiset kyberuhkatiedustelussa käytetyt menetelmät eivät aina riitä tunnistamaan niitä. Työn tavoitteena onkin tutkia, miten tekoälyä voidaan hyödyntää ennakoivien mallien kehittämisessä kyberuhkatiedustelussa. Tutkimuksen tarkoituksena on tuottaa pohja uusille kehitysmahdollisuuksille, joiden avulla voidaan parantaa organisaatioiden kyberturvallisuutta.

Työssä keskitytään teoreettiseen analyysiin sekä konkreettiseen tekoälymallien kokeiluun. Varsinaista uuden prototyypin kehittämistä työ ei sisällä. Tarkoituksena on analysoida yleisimpiä kyberuhkia ja niiden piirteitä, nykyisiä uhkamalleja sekä ennakoivia analyysimenetelmiä ja tekoälymalleja. Työn käytännön osuus keskittyy kyberuhkatiedustelun tekniseen osa-alueeseen.

2 Tutkimusmenetelmä ja -kysymykset

Tämän opinnäytetyön tutkimusmenetelmänä on laadullinen tutkimus, jossa on myös piirteitä tutkimuksellisesta kehittämistyöstä. Laadullisella tutkimuksella tarkoitetaan sellaista tutkimusta, jonka tarkoituksena on vastata sellaisiin kysymyksiin, joilla selvitetään mistä on kyse. Tutkimuksellinen kehittämistyö tai soveltava tutkimus on puolestaan tutkimus, jossa esimerkiksi kehitetään käytännön sovellus olemassa olevan tiedon perusteella. (Hämäläinen 2024.)

Työssä on tarkoituksena esitellä aiheeseen liittyvää teoriaa, työkaluja sekä tekniikoita, joiden avulla voidaan kehittää tutkittavaa aihetta ja halutaan myös selvittää mitä on ennakoiva kyberuhkatiedustelu. Kuitenkin työ ei ole täysin kehittämistyö, sillä tarkoituksena ei ole tuottaa varsinaista ratkaisua vaan lopputuloksena syntyy kehittämisideoita ja -ehdotuksia. Tässä työssä on myös tarkoituksena löytää vastauksia näihin kysymyksiin:

1. Mitä on ennakoiva kyberuhkatiedustelu?
2. Miten ennakoivaa kyberuhkatiedustelua voidaan toteuttaa tekoälyn avulla?

3 Kyberuhkatiedustelu

3.1 Kyberturvallisuus

Kyberturvallisuudella tarkoitetaan laitteiden, tietojärjestelmien ja tiedon turvallisuuden takaamista kyberuhkia vastaan. Sillä pyritään ehkäisemään ja estämään kyberuhkien aiheuttamia vahinkoja. Kyberturvallisuus usein sekoitetaan tietoturvallisuuteen, mutta tietoturvallisuudessa tietoa suojellaan laajemmin niin fyysisesti kuin digitaalisesti. Esimerkiksi torjutaan väärää tietoa tai tiedon levittämistä. (Mitä on kyberturvallisuus? N.d.)

Kyberturvallisuus kattaa kaikki IT infrastruktuurin osat. Näitä osia ovat muun muassa tekoälyturvallisuus, verkkoturvallisuus, laiteturvallisuus sekä ohjelmistoturvallisuus. Esimerkiksi tekoälyturvallisuuden tarkoituksena on estää tai lieventää tekoälysovelluksiin ja -järjestelmiin kohdistuvia tai tekoälyä haitalliseen toimintaan käyttäviä kyberuhkia ja verkkohyökkäyksiä erilaisin toimenpitein sekä teknologioin. Verkkoturvallisuus puolestaan ehkäisee ja estää luvattoman pääsyn verkkoon ja verkkoresursseihin. Laiteturvallisuus suojelee erilaisia laitteita, kuten tietokoneita, niihin kohdistuvilta uhkilta. Laiteturvallisuuteen kuuluu esimerkiksi virustorjuntaohjelmat. Sovellusturvallisuus suojelee sovelluksia ja niihin liittyviä tietoja luvattomalta käytöltä. Se auttaa myös tunnistamaan ja lieventämään puutteita tai haavoittuvuuksia sovellussuunnittelussa. (Lindemulder & Kosinski 2024; AZ STUDIO 2024.)

3.2 Kyberuhkatiedustelu

Kyberuhkatiedustelulla tarkoitetaan sellaista tiedustelua, jossa kerätään, prosessoidaan ja analysoidaan kerättyä dataa, jotta voidaan ymmärtää uhkatekijöiden motiiveja, kohteita sekä hyökkäyskäyttäytymistä. Uhkatiedustelun avulla pystytään muuttamaan toimintaa reaktiivisesta ennakoivaksi sekä tietoturvallisten päätösten tekeminen on nopeampaa ja tietoisempaa. Kyberuhkatiedustelusta hyötyvät erilaisissa kyber- ja tietoturvan parissa työskentelevät henkilöt, jotka tarvitsevat uhkatietoa erilaisissa tilanteissa. Siitä hyötyvät myös yritysten sidosryhmät, joiden on helpompaa tehdä sijoituspäätöksiä, pienentää riskejä ja parantaa tehokkuutta, koska uhkista ja uhkatekijöistä saadaan enemmän tietoa ja niitä ymmärretään paremmin. (Baker 2023.)

Uhkien kehittyessä jatkuvasti, yrityksiltä vaaditaan nopeaa sopeutumista. Tätä helpottaa tiedustelussa käytetty kuusi portainen sykli, joka tarjoaa tiimeille mahdollisuuden optimoida resurssit ja vastata tehokkaasti nykyaikaisiin uhkiin. Nämä kuusi porrasta ovat: suunnittelu, tiedon keräys, prosessointi, analysointi, tulosten esittely ja palaute. Näissä vaiheissa määritellään vaatimukset, jotka halutaan selvittää, kerätään tietoa määritysten mukaisesti, minkä jälkeen saatu tieto prosessoidaan, analysoidaan ja esitetään sidosryhmille. Lopuksi sidosryhmät antavat palautteen, jonka perusteella tehdään mahdollisia muutoksia tulevaisuudessa tehtäviin kyberuhkatiedustelun operaatioihin. (Baker 2024.)

Kyberuhkatiedustelu voidaan jakaa neljään osaan sen mukaan, millaisesta uhkatiedustelusta on kyse. Näitä ovat strateginen, taktinen, tekninen sekä operatiivinen tiedustelu. Strateginen uhkatiedustelu on tarkoitettu lähinnä johtotason turvallisuusasiantuntijoille, joiden tarkoituksena on ohjata korkean tason organisaatiostrategiaa. Strategisessa uhkatiedustelussa tarkoituksena on luoda yleiskuva organisaation uhkakuvasta ja se ei ole niin tekninen. Se tarjoaa tietoa organisaation haavoittuvuuksista ja riskeistä sekä ennaltaehkäisevistä toimista, uhkatoimijoista, heidän tavoitteistaan ja mahdollisten hyökkäysten vakavuudesta. (What is Threat Intelligence in Cybersecurity? 2024.)

Taktinen tiedustelu kohdistuu tarkemmin uhkatoimijoiden käyttämiin tekniikoihin ja toimintatapoihin. Se on tarkoitettu turvallisuustiimille lisäämään heidän ymmärrystänsä hyökkäyskohteista. Taktinen tiedustelu antaa apuja puolustusstrategioiden suunnitteluun ja se sisältää tiedot turvallisuusjärjestelmien haavoittuvuuksista, joita mahdolliset hyökkääjät voivat hyödyntää. (What is Threat Intelligence in Cybersecurity? 2024.)

Tekninen uhkatiedustelun ydin on hyökkäysten yksityiskohtien tunnistaminen. Tarkoituksena on löytää IOC:t, joiden avulla voidaan havaita, että hyökkäys on tapahtunut. Näitä tietoja ovat muun muassa ilmoitetut IP-osoitteet, tietojenkalasteluun käytettyjen sähköpostiviestien sisältö, haittaohjelmanäytteet sekä väärennetyt URLit. Tiedon jakaminen ja milloin se tapahtuu, on kriittinen osa teknistä tiedustelua, sillä osa IOC-tiedoista vanhentuvat muutamassa päivässä. Tällaisia IOC-tietoja ovat esimerkiksi haitalliset IP-osoitteet sekä väärennetyt URLit. (What is Threat Intelligence in Cybersecurity? 2024.)

Operatiivinen tiedustelu keskittyy keräämään tietoa hyökkäyksistä. Sen tarkoituksena on tuottaa tietoa hyökkäyksen tyypistä, sen motiiveista, ajoituksesta ja sen toteutuksesta. Operatiivisessa tiedustelussa tiedon keräys voi olla haastavaa, sillä se tapahtuu useimmiten käyttämällä erilaisia hakkereiden suosimia keskustelufoorumeita tai chathuoneita. Keskustelut ovat usein yksityisiä tai salattuja, manuaalinen tiedon keräys erilaisten viestintäkanavien valtavasta datamäärästä on haastavaa sekä uhkatoimijat saattavat tarkoituksellisesti käyttää sellaista kieltä, jota ulkopuoliset eivät ymmärrä. (What is Threat Intelligence in Cybersecurity? 2024.)

3.3 Kyberuhkatoimijat

3.3.1 Kyberuhkatoimijat

Uhkatoimija on yleiskäsite sellaiselle henkilölle tai ryhmälle, joka on uhka kyberturvallisuudelle. Heidän tarkoituksenansa on tahallisesti aiheuttaa vahinkoa digitaalisille laitteille tai järjestelmille. Vahinkoa aiheutetaan hyväksikäyttämällä erilaisia haavoittuvuuksia järjestelmissä, verkossa ja ohjelmistoissa, joiden avulla kyberhyökkäykset toteutetaan. (What is a threat actor? 2023.)

Uhkatoimijat voidaan luokitella neljään pääluokkaan, joita ovat kyberrikolliset, haktivistit, valtion rahoittamat hyökkääjät sekä sisäpiirin uhkat. Kyberrikolliset toimivat yleensä taloudellisen hyödyn vuoksi. Kyberrikollisten käyttävät yleisimmin erilaisia lunnasohjelmatroijalaisia (ransomware) sekä tietojenkalasteluhuijauksia (phishing). Nykyään kyberrikollisuus on järjestäytyneempää kuin koskaan aiemmin. Vuonna 2023 se nousi kannattavimmaksi laittomaksi teollisuudenalaksi ohittamalla jopa huumekaupan. Esimerkiksi Yhdysvalloissa vuonna 2015 uhrit maksoivat yli 24 miljoonaa dollaria lunnasohjelmatroijalaisia käyttäville ryhmille. (What is a threat actor? 2023; Borges 2024.)

Haktivistit poikkeavat kyberrikollisista siten, että he eivät yleensä toimi taloudellisen hyödyn vuoksi. Heitä motivoi poliittiset ja yhteiskunnalliset tavoitteet, kuten sananvapaus ja ihmisoikeudet. Haktivistit uskovat olevansa oikeutettuja hyökkäämään valtion virastoja, organisaatioita tai yksityishenkilöitä vastaan paljastaakseen salaisuuksia tai arkaluonteisia tietoja. (What is a threat actor? 2023.)

Kansallisvaltiot ja hallitukset rahoittavat hyökkääjiä, joiden tarkoituksena on hyökätä toisen valtion kriittistä infrastruktuuria vastaan, kerätä luottamuksellista tietoa ja aiheuttaa taloudellista vahinkoa. Ne eivät ole yhtä yleisiä kuin kyberrikollisuus tai haktivismi, mutta ovat yksi tärkeimmistä uhkatekijöistä, sillä hyökkääjiä motivoi yleensä enemmän tieto kuin raha. Hyökkääjät ovat yleensä hyvin rahoitettuja, minkä avulla hyökkäyksistä voidaan tehdä monimutkaisempia ja haastavampia havaita. Valtioiden rahoittamien hyökkääjien käyttämistä taktiikoista, tekniikoista ja menettelytavoista (tactics, techniques and procedures, TTPs) käytetään termiä edistynyt jatkuva uhka (Advanced Persistent Threat, APT). (What is a threat actor? 2023; Borges 2024.)

Sisäpiirin uhkat aiheuttavat vahinkoa esimerkiksi organisaation sisältä käsin. He voivat aiheuttaa vahinkoa niin tahallisesti kuin tahattomasti. Tahallisesti toimiva sisäpiirin uhka on esimerkiksi organisaatiossa toimiva henkilö, joka hyväksikäyttää valtuuksiaan saavuttaakseen henkilökohtaista hyötyä, aiheuttaakseen vahinkoa tai varastaakseen tietoa. Tahattomasti toimiva uhka on puolestaan sellainen, joka huolimattomuudellaan aiheuttaa vaaraa. He voivat esimerkiksi käyttää heikkoja salasanoja, käsitellä arkaluontoisia tietoja väärin tai asentavat vaarantuneita ohjelmistoja. Sisäpiirin uhkia on vaikea havaita ja ne ovat olleet vastuussa joistakin historian merkittävimmistä tietomurroista. (What is a threat actor? 2023; Borges 2024.)

3.3.2 Erilaisia kyberuhkatoimijoita

RansomHub

RansomHub on maailmanlaajuinen ransomware-as-a-service (RaaS)- ryhmä, joka siis tarkoittaa sitä, että se toimittaa lunnasohjelmia palveluna. Ransomhubia pidetään tämän hetken lunnasohjelmatoiminnan aktiivisimpana ja merkittävimpänä toimijana. Ryhmä havaittiin ensimmäistä kertaa helmikuussa 2024 ja se on hyökännyt useille eri aloille sekä maantieteellisille alueille. Kuitenkin RansomHub ei hyökkää sellaisia organisaatioita kohtaan, jotka toimivat Itsenäisten valtioiden yhteisön, Kuuban, Pohjois-Korean tai Kiinan alueella. RansomHubin kohteina ovat arvokkaat organisaatiot ja se harjoittaa muiden lunnasohjelmaryhmien tavoin niin sanottua ”suurriistan metsästystä”. Ryhmä on vielä nuori, mutta yhtenä sen hyökkäyksistä on syyskuussa 2024 tehty hyökkäys Planned Parenthood of Montana- järjestöä kohtaan, jossa varastettiin arkaluontoista tietoa 100GB verran. RansomHubin tiedetään käyttävän EDRKillShifter-työkalua ja hyödyntäneen Zerologon-haavoittuvuutta. (Morgan 2024.)

IntelBroker

IntelBroker on tunnetuin pimeän verkon BreachForums-foorumin jäsen. Hänen tiedetään erikoistuvan tunnistavan ja myyvän käyttöoikeuksia vaarantuneisiin järjestelmiin, arkaluontoisen tiedon vuotaminen sekä mahdollisesti kiristys. IntelBroker havaittiin ensimmäistä kertaa loppuvuodesta 2022 ja tuli tunnetuksi alkuvuodesta 2023, ku hänen väitettiin murtaneen Weee Grocery Service-palvelun. Muita murtoja ovat murrot Europoliin, Autotraderiin, Volvoon sekä Hilton Hotelleihin.

Vuonna 2023 IntelBroker liittyi rasistiseen kyberrikollisryhmä CyberNiggersiin, joka erikoistuu erilaisiin hyökkäyksiin Yhdysvaltoihin, EU:hun, Etelä-Afrikkaan sekä Intiaan. IntelBrokerin identiteettiä ei voi varmasti tietää, mutta hän on mahdollisesti serbialainen, joka toimii itsenäisesti ja mahdollisesti käyttää iranilaisia haittaohjelmamuunnelmia. Hänen kohteenaan on erityisesti Yhdysvaltojen puolustusalan korkean profiilin kohteet. (Dark Web Profile: IntelBroker 2024.)

Gamaredon APT

Gamaredon APT on APT-ryhmä, joka on toiminut kybervakoilukampanjoissa vuodesta 2013 lähtien. Se tunnetaan myös nimillä Primitive Bear, Actinium sekä UAC-0010. Ryhmällä on läheiset yhteydet Moskovan turvallisuuspalveluun (FSB) ja sen toiminta on johdonmukaisesti ollut linjassa Venäjän strategisten etujen kanssa. Gamaredon APT hyökkäykset kohdistuvat geopoliittisiin kohteisiin, erityisesti Ukrainan hallitukseen, armeijaan ja kriittiseen infrastruktuuriin, sekä Natomaihin. Ryhmän eniten käyttämät hyökkäystavat ovat kohdistettu tietojen kalastelu (spearphishing), aseistetut USB-asemat sekä laillisten palveluiden hyväksikäyttö. Työkaluina Gamaredon APT käyttää muun muassa GammaLoadia, Pterodoa sekä GammaSteeliä. (Dark Web Profile: Gamaredon APT 2024.)

3.4 Kyberuhkatieto

Kyberuhkatietoa ovat sellaiset tiedot, joiden avulla saadaan käsitys menneistä, nykyisistä ja mahdollisista tulevista kyberuhkista. Näitä tietoja ovat muun muassa vaarantumisindikaattorit, tiedot uhkatoimijoiden operaatioista (TTPs), uhkatoimijoiden profiilit, haavoittuvuustiedot sekä sosiaalisesta mediasta ja pimeästä verkosta saatavat tiedot. Vaarantumisindikaattoreihin kuuluu esimerkiksi sellaiset IP-osoitteet, URLit tai verkkotunnukset, jotka liittyvät haitalliseen toimintaan ja tietojenkalastelusivuille, haittaohjelmien hashit sekä sähköpostiosoitteet tai aiheet, joita käytetään tietojenkalastelukampanjoissa. (Cyber Threat Intelligence: A Comprehensive Guide n.d.)

Toimijoiden profiilit antavat erilaista tietoa motiiveista, kyvyistä ja taidoista sekä aiemmin toteutetuista kampanjoista tai operaatioista. Menneet operaatiot tarjoavat tietoa toimijoiden käyttämistä tietyistä menetelmistä, tekniikoista ja tavoista, joilla he suorittavat operaatiot. Haavoittuvuustiedot ovat ohjelmistojen tai laitteistojen heikkouksia, joita voidaan hyödyntää. Nämä tiedot koostu-

vat esimerkiksi haavoittuvuustunnisteista (CVE numerot), tieto järjestelmistä ja ohjelmistoista, joihin haavoittuvuus vaikuttaa sekä mahdollisista haavoittuvuudesta aiheutuvista vaikutuksista ja niihin käytettävistä lieventämisstrategioista. Erilaiset foorumit, sosiaalisen median alustat ja pimeä verkko voivat tarjota tietoa missä ja miten uhkatoimijat kommunikoivat keskenään, jakavat työkaluja tai myyvät varastettua tietoa. (Cyber Threat Intelligence: A Comprehensive Guide n.d.)

4 Kyberuhat

4.1 Yleisimmät uhat ja niiden piirteet

Kyberuhat vaihtelevat selkeästi havaittavista sähköpostihuijauksista huomaamattomiin koodin osiin, jotka voivat elää verkossa kuukausista jopa vuosiin ennen kuin ne havaitaan. Uhkat ovat erilaisia hyökkäyksiä riippuen mitä hyökkäyksellä halutaan saavuttaa. Hyökkäyksessä voidaan esimerkiksi yrittää päästä tietokoneverkon tai järjestelmän sisälle, jota kautta halutaan muokata, varastaa, tuhota tai paljastaa tietoa. Kohteet vaihtelevat yksittäisistä yksityishenkilöistä aina suuriin yrityksiin sekä valtioiden hallituksiin. Uhkat voidaan jakaa useampaan alueeseen riippuen niiden toteutustavasta. Näitä ovat esimerkiksi haittaohjelmat, tietojenkalastelu, palvelunestohyökkäykset, SQL injektiot sekä hyökkäykset IoT-laitteisiin. (Types of cyberthreats 2024.)

Haittaohjelmat

Haittaohjelmalla tarkoitetaan sellaista haitallista ohjelmaa, joka on luotu tuottamaan vahinkoa joko järjestelmälle tai sen käyttäjille. Ne ovat yleisimpiä kyberuhkia ja nykyaikaisissa kyberhyökkäyksissä käytetään lähes aina jonkinlaista haittaohjelmaa. Näillä ohjelmilla halutaan saada luvaton pääsy verkkoon tai järjestelmään ja saadaan aikaan vahinkoa esimerkiksi tuhoamalla ja varastamalla arkaluontoista tietoa tai kriittisiä tiedostoja. Yleisimpiä haittaohjelmia ovat kiristysohjelmat, Troijan hevonen, vakoiluohjelmat ja madot. (Baker 2024.)

Kiristysohjelmien tarkoituksena on saada lukittua käyttäjän tiedot tai laite, minkä avulla käyttäjää kiristetään maksamaan rahat vastineeksi tietojen tai laitteen avaamisesta. Troijan hevosilla tarkoitetaan hyödylliseksi ohjelmaksi naamioitua haittaohjelmaa, jonka käyttäjä lataa laitteelleen. Tällaiset haittaohjelmat voivat luoda esimerkiksi takaoven, jota kautta hyökkääjällä on pääsy laitteelle tai ladata muita haittaohjelmia sen jälkeen, kun se on päässyt käyttäjän verkkoon. Vakoiluohjelmat

puolestaan keräävät arkaluontoista tietoa kuten käyttäjänimiä, salasanoja ja pankkitunnuksia käyttäjän tietämättä. Madoilla tarkoitetaan itseään kopioivia ohjelmia, jotka leviävät automaattisesti sovelluksiin sekä laitteisiin ja ne eivät vaadi ihmisen toimia. (Types of cyberthreats 2024.)

Sosiaalinen manipulointi ja tietojenkalastelu

Sosiaalisen manipuloinnin tarkoituksena on saada uhri paljastamaan luottamuksellista tietoa, uhkaamalla uhrin omaa tai uhrin organisaation taloudellista hyvinvointia tai muulla tavoin vaarantaa henkilökohtaista tai organisaation turvallisuutta. Hyökkääjä pyrkii manipuloimaan uhria käyttämällä voimakkaita motivointikeinoja kuten rakkautta, rahaa tai pelkoa (Baker 2024). Tunnetuin sosiaalisen manipuloinnin muoto on tietojenkalastelu, jossa houkutellaan uhria jakamaan henkilökohtaista tietoa, käyttäjätunnuksia tai lähettämään rahaa erilaisten sähköpostien, tekstiviestien tai puheluiden muodossa (Types of cyberthreats 2024).

Muita sosiaalisen manipuloinnin muotoja ovat spear phishing, jossa hyödynnetään uhrin julkisia sosiaalisen median profiileja, jotta huijaus olisi uskottavampi, whale phishing, jonka uhrit ovat yritysten johtajia tai rikkaita henkilöitä sekä yrityssähköpostien vaarantuminen, jossa huijaukset toteuttajat esittävät yrityksen johtohenkilöä tai muuta luotettua liikekumppania huijatakseen uhria lähettämään rahaa tai jakamalla arkaluontoista tietoa. Tähän kuuluu myös verkkotunnushuijaukset (DNS spoofing), joissa luodaan nettisivuja, joiden verkkotunnukset muistuttavat olemassa olevia verkkotunnuksia. Näiden avulla pyritään huijaamaan uhria antamaan henkilökohtaista tietoa. (Types of cyberthreats 2024.)

Palvelunestohyökkäykset

Palvelunestohyökkäyksissä (DoS) tarkoituksena on häiritä liiketoimintaa tukkimalla verkkosivusto, sovellus tai järjestelmä suurilla määrillä vääriä pyyntöjä, jotka aiheuttavat kohteen hidastumista tai estävät kokonaan kohteen käytön oikeilta käyttäjiltä. Usein näiden hyökkäysten tarkoituksena ei ole aiheuttaa tietojen menetystä, vaan halutaan aiheuttaa yritykselle ajan, rahan sekä muiden resurssien menetystä ja ne ratkeavat usein ilman lunnaiden maksua. (Baker 2024.) DoS-hyökkäykset eroavat DDoS-hyökkäyksistä (hajautettu palvelunestohyökkäys) siten, että hyökkäyksen lähteenä

on yksi järjestelmä, kun taas DDoS-hyökkäyksissä lähteenä on useita järjestelmiä. DDoS-hyökkäykset ovat myös DoS-hyökkäyksiä nopeampia sekä niitä on vaikeampi estää ja jäljittää. (Difference between DOS and DDOS attack 2024.)

SQL injektio

SQL (Structured Query Language) tarkoittaa strukturoitua kyselykieltä, jota käytetään tietokantajärjestelmien kyselyihin, käyttämiseen ja hallintaan. Näitä tietokantajärjestelmiä käytetään erilaisissa verkkosovelluksissa. SQL injektiot ovat hyökkäystapa, jossa hyväksikäytetään näitä tietokantajärjestelmiä. Esimerkiksi normaalisti käyttäjä hakee verkkosovelluksesta tietyn käyttäjän profiilin, mihin käytetään SQL kyselyä ja tuloksena käyttäjä pääsee katsomaan toisen käyttäjän kirjasuositukset. SQL injektiohyökkäyksessä tätä edellä mainittua toimintoa hyväksikäytetään ja tuloksena hyökkääjä saa selville sellaisia tietoja, joihin normaalisti hyökkääjällä ei ole pääsyä, kuten käyttäjien tunnukset. Tämä vaatii kuitenkin sen, että SQL kyselyitä ei ole suojattu kunnolla. SQL injektioiden vaikutukset voivat vaihdella tietojen paljastamisesta aina haitallisen koodin levittämiseen sovelluksen käyttäjille. (Understanding SQL Injection n.d.)

5 Tekoäly

Tekoäly on kattotermi sellaiselle toiminnalle, joka liitetään ihmisiin ja jotka ohjelmisto, digitaalinen tietokone tai tietokonepohjainen robotti suorittaa. Tällaisia toimintoja ovat esimerkiksi kuviomäärittely (pattern regocnition) tai oppiminen aiemmasta. (Copeland 2024.) Tekoälyn on tarkoitus auttaa esimerkiksi suurten data määrien käsittelyssä, jossa ihmisillä kestäisi vuosia (Kolari & Kallio 2023).

Tällä hetkellä tekoäly voidaan jakaa kahteen osaan: heikkoon ja vahvaan tekoälyyn. Heikolla tekoälyllä tarkoitetaan sellaista tekoälyä, joka voi suorittaa vain sille määrättyjä tehtäviä. Esimerkiksi erilaiset chattibotit, sähköpostin roskapostisuodatus tai GPS-sovellukset ovat heikkoja tekoälyjä. Vahva tekoäly puolestaan pystyy ajattelemaan ja toimimaan itsenäisesti kuten ihminen, mutta tätä ei ole kyetty vielä luomaan. (Kolari & Kallio 2023.)

5.1 Koneoppiminen

Brownin (2021) mukaan koneoppiminen on yksi tapa käyttää tekoälyä. Koneoppimisen määritteli 1950-luvulla, tekoälypioneerinakin tunnettu, Arthur Samuel. Hän määrittelin sen opiskelualaksi, joka antaa tietokoneille mahdollisuuden oppia ilman niiden ohjelmointia erikseen. Tällä tarkoitetaan sitä, että tietokoneet oppivat ohjelmoimaan itsensä kokemuksen kautta käyttämällä niille annettua dataa. (Brown 2021.)

Koneoppimisprosessi alkaa ensimmäisenä koulutuksessa tarvittavan datan keräämisellä ja valmistelulla. Dataa kannattaa kerätä mahdollisimman paljon, jotta lopputuloksesta tulisi hyvä. Se miten data valmistellaan, riippuu siitä, mitä koneoppimismallia käytetään. Valmisteltu tieto jaetaan koulutus- ja testausosiin, joista koulutusaineisto syötetään mallille. Koneoppimismalli kouluttaa itsensä esimerkiksi löytämään toistuvia kuvioita tai tekemään ennusteita sille syötetyn koulutusaineiston perusteella. Testausosaa käytetään mallin tarkkuuden testaukseen tilanteessa, jossa sille annetaan uutta tietoa. Ihminen pystyy säätämään käytettyä mallia esimerkiksi muuttamalla erilaisia arvoja tulosten perusteella, mikä auttaa mallia saavuttamaan tarkempia tuloksia. (Brown 2021.)

5.1.1 Koneoppimistyypit

Koneoppimismalleja koulutetaan käyttämällä koneoppimisalgoritmeja, joiden perusteella koneoppimistyypit voidaan jakaa neljään osaan. Jaottelussa olennaisena osana on myös se, onko ihminen käsitellyt dataa. Näitä tyyppejä ovat valvottu, valvomaton, osittain valvottu ja vahvistettu oppiminen. (Lawton 2024.)

Valvotussa oppimisessa malli koulutetaan datalla, jonka ihminen on merkinnyt (labeled) (Brown 2021). Merkitsemisellä tarkoitetaan datan merkitsemistä sellaisilla tunnisteilla, jotka ovat merkityksellisiä ja luokittelevat datassa olevat elementit tai tulokset (Awan 2023). Esimerkiksi datasetti Titanicin matkustajista voidaan jakaa luokkiin ”Selviytyi”, ”Sukupuoli” ja ”Ikä”. Valvottu oppiminen on kaikista yleisin tyyppi. Tämän tyypin heikkoutena on otsikointi, joka vaatii aikaa ja vaivaa. Valvotun oppimisen käytetyimmät algoritmit ovat luokittelu (classification) ja regressio (regression). Luokittelualgoritmit nimensä mukaisesti luokittelevat asioita tai tapahtumia annetun datan perus-

teella. Luokittelualgoritmit vaihtelevat yksinkertaisista kyllä/ei tai selviytyi/ei-selviytyi monimutkaisempiin, asiat useampaan luokkaan yhdistäviin kuten laiva, auto tai lentokone. Luokittelualgoritmeihin kuuluu päätöspuut (decision trees), logistinen regressio (logistic regression) ja satunnaismetsä (random forest). Regressioalgoritmit puolestaan tunnistavat tietojoukossa olevien muuttujien välisiä suhteita, esimerkiksi miten selviytyminen Titanicista korreloi iän tai sukupuolen kanssa. Regressioalgoritmeihin kuuluu päätöspuut, lineaarinen regressio ja monimuuttujaregressio. (Lawton 2024.)

Valvomattomassa oppimisessa malli koulutetaan sellaisella datalla, jota ei ole merkitty. Tällainen malli pystyy tunnistamaan sellaisia kuvioita, joita ihminen ei huomaa tai osaa etsiä ja on erityisen hyödyllinen data-analysoinnissa, jonka tarkoituksena on löytää paras tapa ratkaista ongelma. (Brown 2021.) Esimerkiksi valvomaton oppiminen voi olla erityisen hyödyllinen tässä opinnäytetyössä. Yleisimmät oppimisalgoritmit valvomattomassa oppimisessa ovat klusterointi ja ulottuvuuksien vähentäminen (dimensional reduction). Klusterialgoritmien tehtävänä on yhdistellä samankaltaisia tietojoukkoja erilaisten kriteerien perusteella ja ulottuvuuksien vähentämisen avulla voidaan esittää annettu tietojoukko pienemmällä määrällä ominaisuuksia säilyttäen kuitenkin alkuperäisen tiedon merkitykselliset ominaisuudet (Lawton 2024; Murel & Kavlakoglu 2024.)

Osittain valvottu oppiminen yhdistää valvottua ja valvomatonta oppimista siten, että siinä käytetään merkittyä dataa, jonka merkintä suoritetaan käyttämällä valvomattoman oppimisen algoritmeja. Vahvistettu oppiminen tapahtuu niin sanotun yrityksen ja erehdyksen kautta, jossa yritetään löytää paras mahdollinen tapa luomalla palkkiojärjestelmä. Palkkiojärjestelmän avulla koneelle kerrotaan aina, milloin oikea valinta on tehty, mikä auttaa konetta jatkossa tekemään oikeita valintoja. (Lawton 2024.)

5.2 Syväoppiminen

Syväoppiminen on yksi koneoppimisen haaroista, joka käyttää keinotekoisia neuroverkkoja (artificial neural network) oppimiseen. Neuroverkot koostuvat useasta toisiinsa kytkettyjen solmujen (nodes) kerroksista, joissa jokainen solmu vastaa datan tietyn ominaisuuden oppimisesta. Esimerkiksi kuvien tunnistuksessa, ensimmäisen kerroksen solmujen vastuulla on tunnistamaan kuvien reunat, seuraavan tason solmut tunnistavat kuvissa olevia muotoja ja niin edelleen. Kuten koneoppimisessa, syväoppimisessakin on erilaisia malleja. Kaikista yleisimpiä syväoppimismalleja ovat

konvoluutioinen neuroverkko (CNN), syvä vahvistusoppiminen (deep reinforcement learning) ja rekursiivinen neuroverkko (RNN). (What is Deep Learning? N.d.)

CNN on kuvien tunnistuksessa ja prosessoinnissa käytetty malli, joka soveltuu erinomaisesti tunnistamaan objekteja. Malli tunnistaa hyvin sellaisetkin objektit, jotka eivät näy kokonaan tai ovat vääristyneitä. Syvää vahvistusoppimista käytetään robotiikassa ja pelien pelaamisessa. Se toimii vahvistetun oppimisen tapaan palkkiojärjestelmää käyttämällä. RNN puolestaan on luonnollisen kielen prosessoinnissa ja puheen tunnistuksessa käytetty malli, joka on hyvä tunnistamaan lauseen kontekstin. Tätä mallia käytetään tekstin luomisessa tai käännöstöissä. (What is Deep Learning? N.d.)

6 Ennakoiva kyberuhkatiedustelu

6.1 Ennakoivan analyysin merkitys kyberuhkatiedustelussa

Ennakoivassa analyysissä nimensä mukaisesti ennakoidaan tulevaisuutta analysoimalla trendejä ja malleja datasta. Käytetty data koostuu erilaisista kohteista saaduista tiedoista kuten verkkoliikenteestä, käyttäjien käyttäytymisestä, tietoturvalokeista ja uhkatiedustelutiedoista. Tämän avulla voidaan tunnistaa toistuvia toimintatapoja tai käyttäytymistä, jotka kertovat mahdollisesta hyökkäyksestä ennen kuin se tapahtuu. (Bartels 2024.)

Ennakoiva analyysi antaa useita etuja kyberuhkatiedusteluun kuten kyberhyökkäysten ennakointi ja ehkäisy, resurssien optimointi sekä kyky reagoida paremmin kyberhyökkäyksiin. Ennakoivan analyysin avulla kyberhyökkäykset voidaan ennaltaehkäistä kokonaan tai lieventää niistä koituvia seurauksia. Resurssien optimoinnilla organisaatiot voivat priorisoida ne tietoturvallisuuden osat, jotka ovat kaikista kriittisimmät ja samalla vahvistaa kyberturvallisuusstrategiaansa. Ennakoivan analyysin avulla voidaan luoda hyökkäysskenaarioita, joiden avulla voidaan valmistaa reagointisuunnitelmia hyökkäysten varalle. (Bartels 2024.)

6.2 Nykyiset uhkamallit

Nykyiset uhkamallit voidaan luokitella allekirjoituksiin (signature-based), poikkeamiin ja heuristiikkaan perustuviin malleihin. Allekirjoituspohjaisella mallilla tarkoitetaan sellaista mallia, joka perustuu tiedettyihin uhkiin. Näillä uhkilla on tietynlaiset ”allekirjoitukset”, joiden perusteella ne voidaan tunnistaa. Näiden mallien pohjana käytetään valmiiksi koottua listaa, joka sisältää tiedetyt vaarantumisindikaattorit. Vaarantumisindikaattoreita voivat olla esimerkiksi haitallisia verkkotunnuksia, tunnettuja tavusekvenssejä (byte sequences) tai haitallista verkkohyökkäyskäyttäytymistä. Allekirjoituspohjaiset mallit voivat käsitellä nopeasti tunnettuja hyökkäyksiä ja väärien positiivisten tulosten määrä on matala. Vaikka nämä mallit pystyvät tunnistamaan haitalliset tapahtumat nopeasti ja tarkasti, se ei kuitenkaan pysty tunnistamaan nollapäivähyökkäyksiä ja muita uusia hyökkäystapoja. (Election Security Spotlight – Signature-Based vs Anomaly-Based Detection n.d.)

Poikkeamiin perustuvia malleja käytetään tunnistamaan muutokset käyttäytymisessä. Nämä mallit koulutetaan ensin normalisoidulla perustasolla, johon sitten toimintaa verrataan. Normaalista poikkeavat tapahtumat aiheuttavat hälytyksen. Normaalista poikkeavia tapahtumia voivat olla esimerkiksi käyttäjän kirjautuminen virka-ajan ulkopuolella, suuri määrä uusia IP-osoitteita yrittää yhdistää verkkoon tai uusi laite on lisätty verkkoon ilman lupaa. Toisin kuin allekirjoituksiin pohjautuvat mallit, poikkeamiin perustuvat mallit voivat tunnistaa nollapäivähyökkäyksiä. Heikkoutena näillä malleilla on suurempi väärien positiivisten määrä. Näiden poissulkeminen voi johtaa resurssien ja ajankäytön lisääntymiseen. (Election Security Spotlight – Signature-Based vs Anomaly-Based Detection n.d.)

Heuristiikkaan perustuvilla malleilla tarkoitetaan sellaisia malleja, jotka käyttävät uhkien tunnistukseen erilaisia epäilyttäviä piirteitä, joita voi löytyä tunnistamattomista, uusista viruksista, tunnettujen uhkien muunnelluista versioista sekä tunnetuista haittaohjelmista. Nämä mallit ovat harvoja malleja, joita käytetään tunnistamaan suuria määriä uusia uhkia. Ne ovat myös harvoja malleja, joiden avulla voidaan tunnistaa polymorfisia viruksia. Polymorfisilla viruksilla tarkoitetaan sellaista haitallista koodia, joka muuttuu ja sopeutuu jatkuvasti. Jotta heuristiikkaan perustuvat mallit ovat mahdollisimman tehokkaita, ne täytyy virittää huolellisesti siten, että uusien haittojen tunnistus on parasta mahdollista ilman, että se tuottaa vääriä positiivisia tuloksia. Tämän vuoksi heuristisia malleja käytetään muiden mallien kanssa. (What is Heuristic Analysis? N.d.)

7 Tekoälyn soveltaminen kyberuhkatiedustelussa

7.1 Tekoälyn soveltamisalueet kyberuhkatiedustelussa

Tekoälyä voidaan hyödyntää kyberuhkatiedustelun eri vaiheissa kuten tiedon keräyksessä, prosessoinnissa sekä tuotannossa. Tiedon keräyksessä tekoäly pystyy automatisoimaan ja tehostamaan datan keräysprosessia sekä skannaamaan suuria määriä dataa eri lähteistä, kuten avoimen lähdekoodin tietoa, pimeän verkon foorumeita, sosiaalista mediaa ja raportteja, tunnistaakseen olennaiset tiedot. Prosessointivaiheessa tekoäly pystyy havaitsemaan ja poistamaan kerätystä datasta kaksoiskappaleet tai muuten tarpeettomat osat. Prosessoinnin jälkeen tekoäly voi merkitä poikkeavuudet käsitellystä datasta ja korreloida ne tunnettujen uhkien tai uhkatekijöiden kanssa. (AI in Threat Intelligence n.d.)

Tuotantovaiheessa tekoälyä voidaan käyttää kokoamaan, tiivistämään ja muotoilemaan olennaiset tiedot ihmisille luettavissa oleviksi ja ytimekkäiksi uhkaraporteiksi. Tekoäly pystyy myös helpottamaan monikielistä uhkaraportointia luomalla tiedustelutiedot eri kielille. Viimeisessä vaiheessa, palautteen annossa, tekoälyä pystytään mukauttamaan muokkaamalla käytettyjä algoritmeja ja prosesseja sen perusteella millaista palautetta sidosryhmät antavat. Tämän avulla saadaan kohdennettua uhkatiedustelua organisaation tarpeisiin sekä vastaamaan tiettyihin uhkiin tehokkaammin. Tämä myös mahdollistaa organisaatioille jatkuvan uhkien havaitsemisvalmiuksien kehityksen sekä resurssien tehokkaamman kohdentamisen tärkeimpiin ja kiireellisimpiin turvallisuusuhkiin suojautumiseen. (AI in Threat Intelligence n.d.)

7.2 Uhka-analyysi ja ennustemallit

Kyberuhkatiedustelun yksi olennaisista osista on uhka-analyysi. Uhka-analysoinnilla tarkoitetaan sellaista analysointia, jossa mahdollisia uhkia etsitään ja arvioidaan sekä määritetään suojattava omaisuus (assets). Tämän avulla organisaatiot tietävät mitä turvallisuusuhkia heillä on ja mitkä järjestelmän osat ovat haavoittuvaisia. Suojattavalla omaisuudella voidaan tarkoittaa esimerkiksi käyttäjien laitteita, kuten tietokoneet ja työasemat, verkkolaitteita, kuten reitittimet ja kytkimet, kaikenlaista dataa, kuten tietokannat, tallennetut ja arkistoidut tiedot, ohjelmistoja, kuten käyttöjärjestelmä ja asiakasohjelmat sekä palveluja, kuten sovellukset ja IP-palvelut. Mahdollisia uhkia

voivat olla esimerkiksi erilaiset varkaudet, haittaohjelmat, fyysiset vahingot, luvaton pääsy tietoihin, laitteistoihin tai palveluihin, hajautettu palvelunestohyökkäys, tietojen, laitteistojen tai ohjelmistojen vahingoittuminen sekä tietovuoto. (Lange 2024.)

Uhka-analysoinnissa käytetään erilaisia menetelmiä, joita ovat muun muassa riskien arviointi ja haavoittuvuusanalyysi. Riskien arvioinnilla tarkoitetaan prosessia, jossa tunnistetaan, arvioidaan ja priorisoidaan uhat niiden mahdollisten vaikutusten ja toteutumisen todennäköisyyden perusteella (Gillis 2023). Haavoittuvuusanalyysissa tarkastellaan järjestelmät, tietoverkot ja ohjelmistot sellaisten tietoturvaheikkouksien tunnistamiseksi, joita uhka voisi hyödyntää (Vulnerability Assessment n.d.).

Tekoälyn hyödyntäminen uhka-analyysissa nopeuttaa ja mahdollistaa suurten datamäärien käyttöä, minkä avulla voidaan luoda tarkempia ennustemalleja. Riskien arvioinnissa tekoälyn avulla pystytään havaitsemaan malleja ja ennustamaan mahdollisia uhkia aiempia tapahtumia koskevien tietojen perusteella, mikä parantaa uhkien havaitsemisen tarkkuutta ja nopeutta (What Is the Role of AI in Threat Detection? N.d.). Haavoittuvuusanalyysissa tekoälyn avulla voidaan automatisoida haavoittuvuuksien havaitseminen esimerkiksi skannaamalla ja analysoimalla sovelluksia, konfiguraatioita ja verkkoliikennettä. Tämän avulla voidaan vähentää esimerkiksi virheellisten konfiguraatioiden, päivittämättömien sovellusten tai hyväksikäytettävien heikkouksien aiheuttamia haavoittuvuuksia. (AI in Cybersecurity: Revolutionizing threat detection and defense 2023.)

Ennustemallit

Kyberuhkien ennustaminen on haastavaa ja hidasta ilman tekoälyä. Tekoälyn avulla voidaan luoda ennusteita niin historiallisesta kuin reaaliaikaisesta datasta. Tämä takaa paremmat mahdollisuudet suojautua tulevilta kyberhyökkäyksiltä. Uhkien ennustamiseen voidaan luoda tekoälypohjaisia ennustemalleja. Näitä ennustemalleja voidaan toteuttaa käyttämällä esimerkiksi koneoppimismalleista satunnaismetsiä tai tukivektorikoneita (Support Vector Machines) tai syväoppimismalleista konvoluutioneuroverkkoja tai rekursiivisia neuroverkkoja. Näiden tekoälymallien avulla voidaan luoda reaaliaikaisia ennusteita mahdollisista uhkista aiempien kyberhyökkäyksien perusteella. (Dhanushkodi & Thejas 2024.)

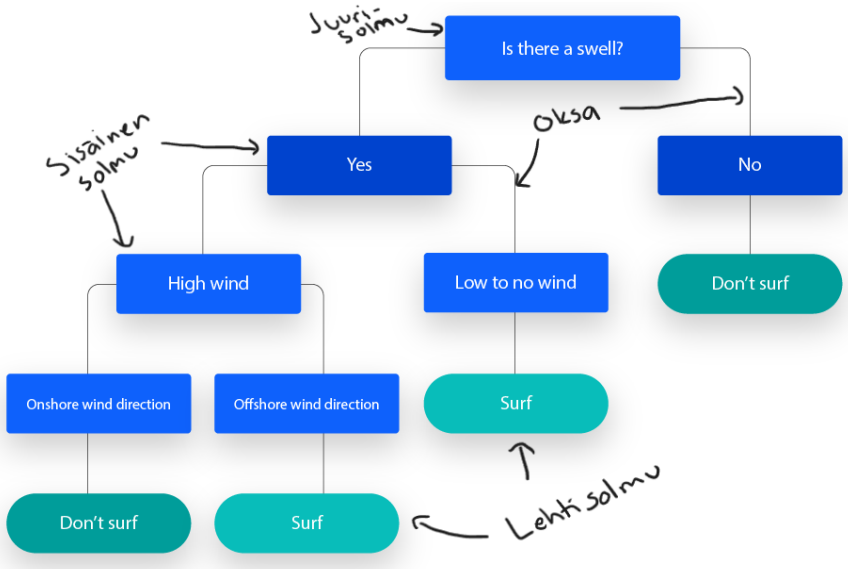
8 Tekoälymallit kyberuhkatiedustelussa

8.1 Poikkeamien havaitsemistekniikat

Poikkeavuuksilla tarkoitetaan normaalista tai odotetusta poikkeavaa käyttäytymistä tai toimintaa. Esimerkiksi tavallisesti työntekijä kirjautuu tietokoneelle arkiaamuisin ja kirjautuu ulos iltapäivällä, mutta lokitietojen perusteella viimeisen kuukauden aikana on huomattu epätavallisia kirjautumisia viikonloppu öisin. Kyberuhkatiedustelussa poikkeavuuksien havaitsemista voidaan hyödyntää tunnettujen kyberuhkatekijöiden toimintatapojen tunnistamisessa kyberhyökkäyksissä. Poikkeavuuksien havaitsemiseen voidaan hyödyntää koneoppimismalleista päätöspuita, satunnaismetsiä, k-lähintä naapuria (k-NN) ja naiivi Bayesia (Naive Bayes). (Tuychiev 2023.)

Päätöspuut

Päätöspuu on valvotun oppimisen algoritmi, jota käytetään sekä luokittelun että regression tehtäviin. Päätöspuulla on hierarkkinen rakenne, joka muodostuu juurisolmusta, oksista, sisäisistä solmuista sekä lehtisolmuista. (Ks. kuvio 1.) Päätöspuut ovat helppotulkintaisia, joka auttaa tärkeimpien ominaisuuksien valinnassa ja se on joustavampi tiedon suhteen, kuin muut luokittelijat eli se tarvitsee vähemmän tiedon esikäsittelyä. Kuitenkin päätöspuilla on myös heikkouksia. Ne ovat alttiimpia ylisovittamiseen (overfitting), jos ne ovat monimutkaisia ja pienetkin vaihtelut datassa voivat vaikuttaa merkittävästi lopputulokseen. (What is a decision tree? N.d.) Ylisovittamisella tarkoitetaan tilannetta, jossa malli opettelee ulkoa sille syötetyn koulutusdatan, jonka perusteella se saa täydet pisteet, mutta se ei pysty ennustamaan sellaisesta datasta, jota se ei ole vielä nähnyt (Cross-validation: evaluating estimator performance n.d.).



Kuvio 1 Päätöspuun rakenne (What is a decision tree? N.d., muokattu)

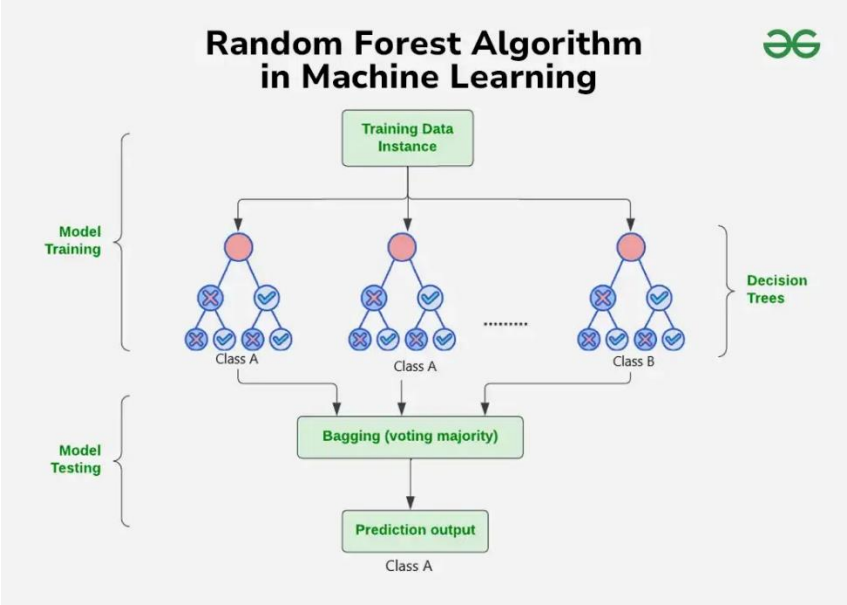
Päätöspuut toteutetaan siten, että aluksi muutetaan datasta tekstimuotoiset muuttujat numeraaliseen muotoon. Tämän jälkeen data jaetaan ominaisuuksiin, joiden avulla ennustus tehdään, sekä kohteeseen, jonka arvot halutaan ennustaa. Tämän jälkeen päätöspuu koulutetaan jaetulla datalla. Alla on esimerkkitoteutus päätöspuusta. (Machine Learning – Decision Tree n.d.)

```
import pandas
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt

df = pandas.read_csv("data.csv")

d = {'UK': 0, 'USA': 1, 'N': 2}
df['Nationality'] = df['Nationality'].map(d)
d = {'YES': 1, 'NO': 0}
df['Go'] = df['Go'].map(d)

features = ['Age', 'Experience', 'Rank', 'Nationality']

X = df[features]
y = df['Go']

dtree = DecisionTreeClassifier()
dtree = dtree.fit(X, y)

tree.plot_tree(dtree, feature_names=features)
```

Kuvio 2 Esimerkkitoteutus päätöspuusta (Machine Learning - Decision Tree n.d.)

Satunnaismetsät

Satunnaismetsä on algoritmi, jossa joukko päätöspuita tekee ennusteet. Päätöspuut koulutetaan datan satunnaisilla osilla, minkä jälkeen tulokset yhdistetään ja luodaan keskiarvo niistä. (Ks. kuvio 3.) Päätöspuiden valinnat ovat toisistaan riippumattomia ja jokaisella on eri osa datasta. Jotta tulokset ovat mahdollisimman tarkat, datamäärän tulee olla tarpeeksi suuri. (Random Forest Algorithm in Machine Learning 2025.)



Kuvio 3 Satunnaismetsän toimintamalli (Random Forest Algorithm in Machine Learning 2025)

Satunnaismetsä toteutetaan siten, että tarvittava data jaetaan ominaisuuksiin (features) ja luokkaan (label). Tämän jälkeen ominaisuudet ja luokka jaetaan koulutukseen ja testaamiseen, joiden avulla malli koulutetaan ja testataan. Seuraavaksi malli luodaan ja koulutetaan koulutukseen tarkoitetulla datan osalla. Viimeisenä testataan malli testaamiseen tarkoitetulla osalla. Testaamisen tulosten avulla mallia voidaan arvioida esimerkiksi tarkkuuden osalta. Kuvio 4 näyttää esimerkkitoteutuksen satunnaismetsämallista.

```
data = pd.read_csv('urls/urls.csv', sep=';')
df = pd.DataFrame(data)

y = df_test['label']
X = df_test['domain']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_vectorized, y, test_size=0.2, random_state=42)
# Train Random Forest
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
```

Kuvio 4 Esimerkkitoteutus satunnaismetsästä

8.2 Tietojenkalastelun tunnistaminen

Tekoälyn avulla tietojenkalastelun voi tunnistaa analysoimalla sähköpostiviestien sisältöä tai esimerkiksi vertaamalla niiden sisältämiä verkko-osoitteita tiedossa oleviin vaarallisiin osoitteisiin. Tekoälymallit voidaan kouluttaa tunnistamaan yleisimmin tietojenkalastelussa käytettäviä varoitusmerkkejä, kuten kirjoitusvirheet, vastaanottajan kiristäminen tai URLit. Näiden perusteella mallit laskevat todennäköisyyden sille, onko kyseessä tietojenkalastelu ja pitäisikö se estää. (Phishing Detection Techniques n.d.)

Tietojenkalastelun tunnistamiseen voidaan käyttää hyvin erilaisia tekoälymalleja aina koneoppimismalleista suuriin kielimalleihin (large language models, LLMs). Tarkemmin näistä malleista käydään läpi syväoppimismallit BERT (Bidirectional Encoder Representations from Transformers) ja DistilBERT sekä koneoppimismalli logistinen regressio. Logistinen regressio on tilastollinen algoritmi, joka ennustaa kategorisen muuttujan tulosta. Sen tuloksena on todennäköisyysarvo välillä 0–1. Sitä voidaan käyttää sekä sellaisissa tilanteissa, joissa on kaksi mahdollista tulosta, esimerkiksi ”voitto” ja ”häviö”, että tilanteissa, joissa on kolme tai useampi mahdollinen tulos, esimerkiksi ”sairaus 1”, ”sairaus 2” ja ”sairaus 3”. Logistinen regressio -malli toteutetaan jakamalla data koulutukseen ja testaukseen, minkä jälkeen se koulutetaan ja testataan. (Ks. kuvio 5.) (Logistic Regression in Machine Learning 2025.)

```
#Load the following dataset
X, y = load_breast_cancer(return_X_y=True)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20,
random_state=23)

clf = LogisticRegression(max_iter=10000, random_state=0)
clf.fit(X_train, y_train)

acc = accuracy_score(y_test, clf.predict(X_test)) * 100
print(f"Logistic Regression model accuracy: {acc:.2f}%")
```

Kuvio 5 Logistinen regressio -malli (Logistic Regression in Machine Learning 2025.)

BERT on Googlen tekemä luonnollisen kielen prosessointiin (natural language processing, NLP) tarkoitettu malli. Se on toiminut pohjana useille muille malleille, kuten DistilBERT. BERT-mallit pystyvät ymmärtämään ilmaisujen vivahteita paremmin kuin muut luonnollisen kielen prosessointiin käytetyt mallit. Esimerkiksi lauseesta ”Matti tarvitsee lääkettä apteekista. Hänen päänsä on kipeä,

joten voitko hakea hänelle särkylääkettä?” BERT ymmärtää paremmin, että ”Matti”, ”hänen” ja ”hänelle” ovat kaikki sama henkilö. Annettu esimerkki on kirjoitettu suomeksi lukijan avuksi, mutta BERT on koulutettu englannin kielellä. DistilBERT on pienempi, nopeampi ja kevyempi versio BERTistä. (BERT n.d.)

DistilBERT toteutetaan yksinkertaisimmillaan lataamalla haluttu esikoulutettu malli ja tokenisoija. Tämän jälkeen syötetään data tokenisoijalle, joka muuntaa sen sopivaan muotoon. Tämä muunnettu data syötetään mallille, joka tekee ennusteen syötteestä. Kuvio 6 esimerkissä halutaan saada selville, onko ”Wow! What a surprise!” -lause positiivinen vai negatiivinen.

```
import torch
from transformers import DistilBertTokenizer, DistilBertForSequenceClassification

tokenizer = DistilBertTokenizer.from_pretrained("distilbert-base-uncased-finetuned-sst-2-english")
model = DistilBertForSequenceClassification.from_pretrained("distilbert-base-uncased-finetuned-sst-2-english")

inputs = tokenizer("Wow! What a surprise!", return_tensors="pt")

with torch.inference_mode():
    logits = model(**inputs).logits

predicted_class_id = logits.argmax().item()
model.config.id2label[predicted_class_id]
```

>> Output: “POSITIVE”

Kuvio 6 Esimerkki DistilBERTistä (Drishti 2022.)

8.3 Aikasarja-analyysi uhkien ennustamisessa

Aikasarja-analyysi on analyysityyppi, jossa analysoinnin kohteena on aikajaksot tietyltä aikaväliltä kerätystä datasta. Sen tarkoituksena on selvittää mahdolliset kehityssuunnat, mallit ja muutokset aikaväliltä. Aikasarja-analyysin avulla voidaan ennustaa tulevia trendejä, tunnistamaan malleja ja poikkeamia, vähentämään riskejä sekä tehdä strategisia suunnitelmia. Sen neljä tärkeintä osaa ovat trendi, kausivaihtelu, syklisyys ja epäsäännöllisyys. (Time Series Analysis and Forecasting 2024.)

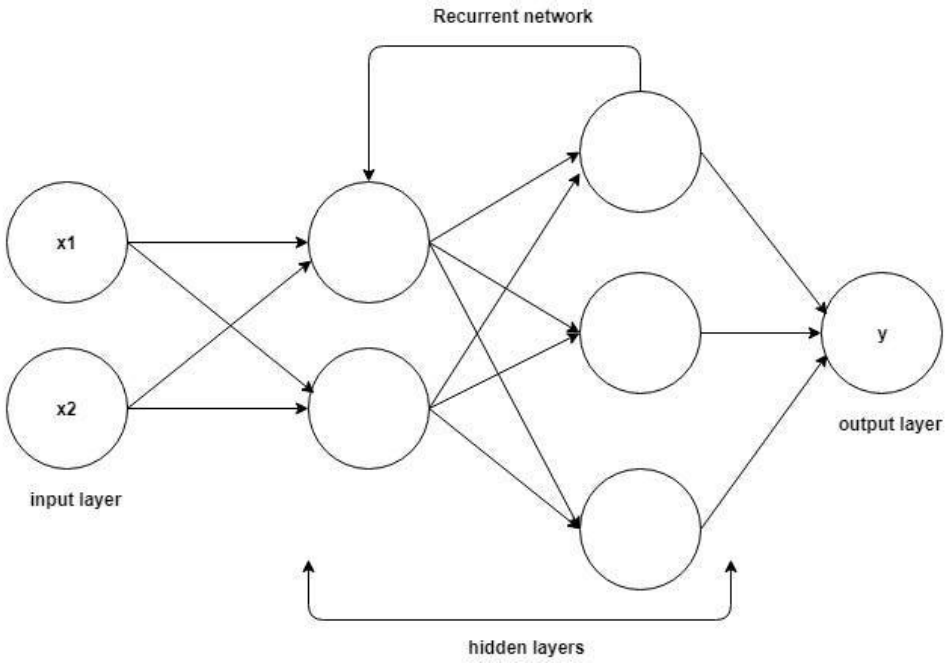
Trendi on pitkän aikavälin liikettä tai suuntautuneisuutta ajan kuluessa. Se kuvastaa yleistä taipumusta kasvaa, laskea tai pysyä vakaana. Kausivaihtelu on jaksottaista vaihtelua tai malleja, joita

esiintyy säännöllisin väliajoin datassa. Kausivaihteluun vaikuttavat erilaiset tekijät kuten vuodenajat, juhlapyhät sekä suhdanteet. Syklisyys on puolestaan pidempiaikaista vaihtelua ja sillä ei ole kiinteää ajanjaksoa kuten kausivaihtelulla. Nämä vaihtelut voivat kestää useita vuosia ja ne liittyvät yleensä taloudelliseen toimintaan. Epäsäännöllisyys on aikavälillä olevaa ennalta arvaamatonta tai satunnaista vaihtelua, jotka eivät johdu trendeistä, kausivaihtelusta tai suhdannevaihtelusta. Epäsäännöllisyys voi johtua satunnaisista tapahtumista, mittausvirheistä tai muista ennalta arvaamattomista tekijöistä. (Time Series Analysis and Forecasting 2024.)

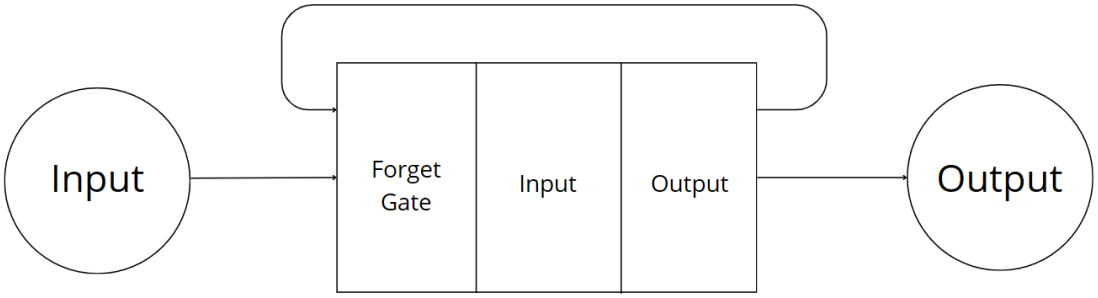
Kyberuhkatiedustelussa aikasarja-analyysia voidaan hyödyntää tunnistamaan erilaisia hyökkäystrendejä ja -malleja sekä ennustamaan mahdollisia hyökkäystrendejä. Aikasarja-analyysin avulla voidaan muun muassa analysoida miten erilaiset geopoliittiset tapahtumat ovat vaikuttaneet aiempiin hyökkäystrendeihin, miten erilaiset hyökkäysmallit ovat muuttuneet ajan kuluessa ja millä todennäköisyydellä tietynlainen hyökkäysskenaario tapahtuu. (Bolen 2024.)

Aikasarja-analyysimallit

Aikasarja-analyysin voi suorittaa useilla eri menetelmillä, mutta tässä keskitytään miten RNN- ja LSTM-syväoppimismalleilla voidaan toteuttaa aikasarja-analyysi. Tavallinen neuroverkko toimii siten, että syötteet (input) ja tuotokset (output) toimivat itsenäisesti, kun taas puolestaan RNN toimii siten, että aiemman vaiheen tuotos on seuraavan vaiheen syötös. (Ks. kuvio 7.) Tämä soveltuu hyvin sellaisiin tilanteisiin, joissa aiempien vaiheiden konteksti on olennainen. Pitkän lyhytkestoisen muistin verkot (LSTM) on yksi RNN:n variaatioista. Se poikkeaa RNN:stä siten, että siinä on erityinen muistiyksikkö, jonka avulla se voi oppia peräkkäisten tietojen pitkän aikavälin riippuvuuksia. (Ks. kuvio 8.) Tämän vuoksi LSTM on tunnettu aikasarjaennusteiden käytössä. (Introduction to Recurrent Neural Networks 2024; What is LSTM – Long Short Term Memory? 2024.)



Kuvio 7 RNN:n arkkitehtuuri (De 2018)



Kuvio 8 LSTM:n arkkitehtuuri

Aikasarja-analyysissä mallien luominen aloitetaan datan esikäsittelyllä, jonka jälkeen data jaetaan yhtä pitkiin syötesarjoihin. Jaksot muodostuvat syötöksestä ja tavoitteesta, joka on jaksoa seuraava arvo. Syötösdata muokataan malleihin sopivaan muotoon. Seuraavaksi rakennetaan malli halutulla määrällä erilaisia kerroksia. Kerrokset sisältävät vähintään yhden syötöskerroksen ja yhden tuotoskerroksen. Mallien tarkkuutta säädellään kerrosten määrällä ja niiden muuttujilla. Tämän jälkeen malli kootaan halutuilla muuttujilla sekä koulutetaan jaksotetulla datalla. Koulutuksen jälkeen malli ennustaa jokaisen syötesarjan seuraavan arvon. (Ks. kuvio 9 ja kuvio 10.)

```
# Data is preprocessed in variable data
data = pd.read_csv('data.csv', sep=',')
data = pd.DataFrame(data)

# Prepare data for RNN (input shape: [samples, time_steps, features])
X = []
y = []
sequence_length = 30
for i in range(len(data) - sequence_length):
    X.append(data[i:i+sequence_length])
    y.append(data[i+sequence_length])

X = np.array(X)
y = np.array(y)

# Reshape input for RNN
X = X.reshape((X.shape[0], X.shape[1], 1))

# RNN model
rnn = Sequential([
    Input(shape=(sequence_length, 1)),
    SimpleRNN(64, activation='relu'),
    Dense(X.shape[2])  # Output layer
])

# Compile model
rnn.compile(optimizer='adam', loss='mse')

# Train model
rnn.fit(X, y, epochs=20, batch_size=32)

# Make predictions
predictions = rnn.predict(X)
```

Kuvio 9 Esimerkki RNN-mallista

```
# Data is preprocessed in variable data
data

# Split data to sequences
X = []
y = []
sequence_length = 10
for i in range(len(data) - sequence_length):
    X.append(data[i:i+sequence_length])
    y.append(data[i+sequence_length])

X = np.array(X)
y = np.array(y)

# Reshape input for LSTM
X = X.reshape((X.shape[0], X.shape[1], 1))

# LSTM model
model = Sequential([
    LSTM(128, activation='relu', input_shape=(sequence_length, 1)),
    Dropout(0.5),
    Dense(1)  # Output layer
])

optimizer = Adam(learning_rate=0.001)
model.compile(optimizer=optimizer, loss='mse')

# Train
model.fit(X, y, epochs=10, batch_size=32)

# Make predictions
predictions = model.predict(X)
```

Kuvio 10 Esimerkki LSTM-mallista

9 Data, sen esikäsittely ja olennaisuus

9.1 Työssä käytetyt datalähteet

Suoraan saatavia datasettejä kyberuhkatiedosta, jotka olisivat sellaisenaan sopinut tekoälymallien testaukseen ei ollut suoraan saatavilla. Työn tarkoituksena ei ollut myöskään tuottaa omia datasettejä, joten datasetit kerättiin erilaisilta sivustoilta, jotka tarjosivat valmiita settejä tai tarvittava data generoitiin synteettisesti tekoälyn luomien skriptien pohjalta. Työssä käytetty data saatiin Kagglesta, Aalto-yliopiston verkkosivuilta, UNSW Sydneyn tutkimusportaalista ja synteettisesti generoimalla.

Kaggle on maailman suurin tietojenkäsittelijöiden yhteisö, joka tarjoaa mahdollisuuden kilpailla, tehdä yhteistyötä, oppia ja jakaa omia töitä tietojenkäsittelyyn, tekoälyyn ja koneoppimiseen liittyen (Kaggle n.d.). Kagglessa on saatavilla lukemattomia datasettejä erilaisiin käyttö tarkoituksiin, joita sivuston käyttäjät ovat sinne lisänneet (How to Use Kaggle n.d.). Aalto-yliopisto on Suomessa toimiva yliopisto, joka tarjoaa tutkimusportaalin, jossa on tietoa tutkijoista, tutkimuksesta ja julkaisuista. Siellä on saatavilla erilaisia tutkimusaineistoja ja ohjelmistoja tietyin lisenssiehdoin. (Aalto-yliopiston tutkimusportaali n.d.) UNSW Sydney on Sydneyssä toimiva yliopisto, joka tarjoaa Aalto-yliopiston tavoin tutkimusportaalin kautta erilaisia tutkimusaineistoja (About us n.d). Työssä käytetty aineisto on vapaasti käytettävissä akateemisiin töihin (Nour 2021).

Generoidut datat on luotu vain esimerkiksi käytännön osuuteen eikä niiden tarkoituksena ole kuvastaa todellisia tilanteita vain olla tukemassa mallien toimintaa. Datan generoimiseen käytettiin skriptejä, jotka luotiin ChatGPT:n avulla. Skriptien avulla generoitiin kaksi erilaista datasettiä kyberhyökkäyksiin, joista yhdessä hyökkäykset tapahtuivat satunnaisesti ja toisessa erilaisilla hyökkäystyypeillä oli erilaiset, toistuvat hyökkäyssyklit. Kummassakin datasetissä on käytetty samoja hyökkäystyyppejä, kuten DDoS, haittaohjelma ja SQL injektio.

9.2 Datan esikäsittely

Datan esikäsittely on oleellinen osa tekoälymallien tekemisessä. Esikäsittelyssä raakadata käsitellään siten, että lopputuloksena on siisti, jäsennelty data. Esikäsittely sisältää kaikkea tyhjien arvojen käsittelystä muuttujien normalisointiin ja muuntamiseen (encoding). (ML | Data Preprocessing in Python. 2025.)

Tyhjien arvojen käsittelyllä tarkoitetaan sitä, että datasta etsitään mahdolliset tyhjät arvot ja ne korvataan sopivilla arvoilla, kuten muiden arvojen keskiarvolla. Normalisointi puolestaan tarkoittaa muuttujien skaalausta, jonka avulla parannetaan mallien johdonmukaisuutta ja vertailukelpoisuutta. Esimerkiksi numeroarvot muunnetaan välille 0–1. Yleisimmin käytettyjä tekniikoita ovat Z-Score Normalisation ja Min-Max Scaling. (Data Normalization Machine Learning 2024.)

Muuntamisella tarkoitetaan kategorisen tai tekstimuotoisen datan muuntamista numeraaliseen muotoon, jotta algoritmit voivat käsitellä dataa. Tekstimuotoinen data voi olla joko sellaista, jolla ei ole järjestystä, kuten sukupuoli tai eläinlajit, tai sellaista, jolla on järjestys, kuten tyytyväisyysluokitus. Tämän perusteella data muunnetaan käyttämällä sopivaa tekniikkaa. Jos järjestystä ei ole, voidaan käyttää esimerkiksi One Hot Encoding -tekniikkaa. Muussa tapauksessa voidaan käyttää esimerkiksi Ordinal Encoding -tekniikkaa. (Kumar 2024.)

URLit

Tässä käytetty datasetti on Aalto-yliopiston tutkimusportaalista. URLit sisältävän datasetin esikäsittely aloitettiin tarkastelemalla millaista tietoa se sisältää. (Ks. kuvio 11.) Seuraavaksi etsittiin mahdolliset tyhjät arvot ja poistettiin ne. Tyhjät arvot poistettiin, koska datasetti sisälsi vain ”domain”- ja ”label”-ominaisuudet, joiden korvaamista ei voinut tehdä. Koska kyseessä oli URLien luokittelu, URLien siistiminen tai muuntaminen ei ollut tarpeellista.

	domain	label
0	nobell.it/70ffb52d079109dca5664cce6f317373782/...	1.0
1	www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrc...	1.0
2	serviciosbys.com/paypal.cgi.bin.get-into.herf....	1.0
3	mail.printakid.com/www.online.americanexpress....	1.0
4	thewhiskeydregs.com/wp-content/themes/widescre...	1.0

Kuvio 11 Alkuperäiset URLit

Sähköpostit

Tässä käytetty datasetti on Kagglesta. Sähköpostien käsittely aloitettiin tutkimalla mitä tietoja data sisältää ja mitkä niistä ovat sellaisia, joita tarvitaan myöhemmin. (Ks. kuvio 12.) Sähköpostien luokittelua haluttiin lähteä kokeilemaan sähköpostien aiheiden perusteella, joten jäljelle jätettiin vain ”subject”- ja ”label”-ominaisuudet. Seuraavana vaiheena datan käsittelyssä oli tekstidatan puhdistus. Puhdistuksessa tekstimuotoisesta tiedosta käsiteltiin NaN-arvot, poistettiin HTML-tagit, erikoismerkit ja teksti muutettiin kokonaan pieniksi kirjaimiksi. (Ks. kuvio 13 ja kuvio 14.)

	sender	receiver	date	subject	body	label	urls
0	Jeff Chan <ociwu@surbl.org>	zqoqi@spamassassin.apache.org	Tue, 05 Aug 2008 18:31:40 -0600	Re: what are the criteria for being listed in\...	Also, the sa-blacklist inclusion policy is at:...	0	1
1	Freeman Cabrera <ChadwickzerothRosario@million...	twanna_patient@gvc.ceas-challenge.cc	Wed, 06 Aug 2008 04:32:27 -0200	Guaranteed Erection Fast	\nSize DOES matter - change your life today!\n...	1	1
2	Daily Top 10 <notelliu1989@arte-m.de>	user8.2-ext1@gvc.ceas-challenge.cc	Wed, 06 Aug 2008 15:31:48 +0200	CNN.com Daily Top 10	>+=+...	1	1
3	Chris Poteet <jhsluxg@siolon.com>	PownceAPI <zgdjhgesv@googlegroups.com>	Wed, 06 Aug 2008 08:39:40 -0700	[PownceAPI] Dates in API?	\nLet's say I wanted to return the actual date...	0	1
4	Gene Oneil <auralx8@pinnacleframe.com>	user8.1@gvc.ceas-challenge.cc	Fri, 08 Aug 2008 11:15:14 +0200	Re:	\nLove is the beginning of all the joy which n...	1	1

Kuvio 12 Alkuperäinen sähköpostidata

	subject	label
0	re what are the criteria for being listed in\t...	0
1	guaranteed erection fast	1
2	cnncom daily top 10	1
3	pownceapi dates in api	0
4	re	1

Kuvio 13 Sähköpostidata puhdistuksen ja tarpeettomien kenttien poiston jälkeen

```
# Cleaning data
def clean_email(text):
    if pd.isnull(text):  # Handle NaN values
        return ""
    # Remove HTML tags
    text = re.sub(r'<.*?>', '', text)
    # Remove special characters
    text = re.sub(r'[^\w\s]', '', text)
    # Convert to lowercase
    return text.lower()

X['subject'] = X['subject'].apply(clean_email)
```

Kuvio 14 Datan puhdistukseen käytetty funktio

Verkkoliikenne

Tässä osiossa käytettiin UNSW Sydneyn tutkimusportaalista löytyvää datasettiä. Datasetti sisälsi yhteensä 46 eri ominaisuutta. Käsittely aloitettiin tutkimalla millaista tietoa ominaisuudet pitävät sisällään. Tarkempaa tietoa ominaisuuksista löytyy liitteestä 1. Ennen muutoksia ”label”- ja ”type”-ominaisuudet tallennettiin omiin muuttujiin ja sekä poistettiin ne datasetistä. Tämän jälkeen poistettiin sellaiset ominaisuudet, jotka eivät tarjonneet oleellista tietoa, kuten ”src_ip” ja ”dst_ip”. Seuraavaksi muunnettiin sellaisten ominaisuuksien arvot, jotka sisälsivät arvot ”T”, ”F” tai ”-”. ”F”- ja ”-”-arvot muutin nolliksi (0) ja ”T”-arvot ykkösiksi (1). (Ks. kuvio 15.)

Viimeisenä etsittiin numeraaliset ominaisuudet, jotka skaalattiin käyttämällä MinMaxScaler-metodia sekä muunnettiin tekstimuotoiset ominaisuudet käyttämällä TargetEncoder-metodia. TargetEncoder valittiin siksi, koska monet tekstimuotoiset muuttujat sisälsivät useita kategorioita, mikä

olisi kasvattanut ominaisuuksien määrän todella suureksi jollain toisella metodilla. Lopputuloksen esikäsittelystä näkyy kuviolla 16.

```
df['ssl_established'] = df['ssl_established'].apply(lambda x: 1 if x == 'T' else 0)
df['dns_AA'] = df['dns_AA'].apply(lambda x: 1 if x == 'T' else 0)
df['dns_RD'] = df['dns_RD'].apply(lambda x: 1 if x == 'T' else 0)
df['dns_rejected'] = df['dns_rejected'].apply(lambda x: 1 if x == 'T' else 0)
df['ssl_resumed'] = df['ssl_resumed'].apply(lambda x: 1 if x == 'T' else 0)
df['dns_RA'] = df['dns_RA'].apply(lambda x: 1 if x == 'T' else 0)
```

Kuvio 15 Arvojen muuttaminen

	src_port	dst_port	duration	src_bytes	dst_bytes	missed_bytes	src_pkts	src_ip_bytes	dst_pkts	dst_ip_bytes	...	http_status_code
0	0.858226	0.001222	1.216892e-08	0.000000e+00	0.000000	0.000000	0.000157	0.000070	0.000008	6.944804e-07	...	0.00000
1	0.543242	0.001222	6.187115e-08	0.000000e+00	0.000000	0.000000	0.000157	0.000070	0.000008	6.944804e-07	...	0.00000
2	0.789873	0.140483	2.657701e-07	1.588619e-07	0.000005	0.000008	0.000470	0.000292	0.000041	3.076548e-05	...	0.49505
3	0.709494	0.001222	2.362907e-04	0.000000e+00	0.000000	0.000000	0.000235	0.000102	0.000016	1.388961e-06	...	0.00000
4	0.000320	0.000336	1.176258e-09	0.000000e+00	0.000000	0.000000	0.000157	0.000050	0.000008	5.092856e-07	...	0.00000

Kuvio 16 Verkkoliikennedata esikäsittelyn jälkeen

Aikasarjat

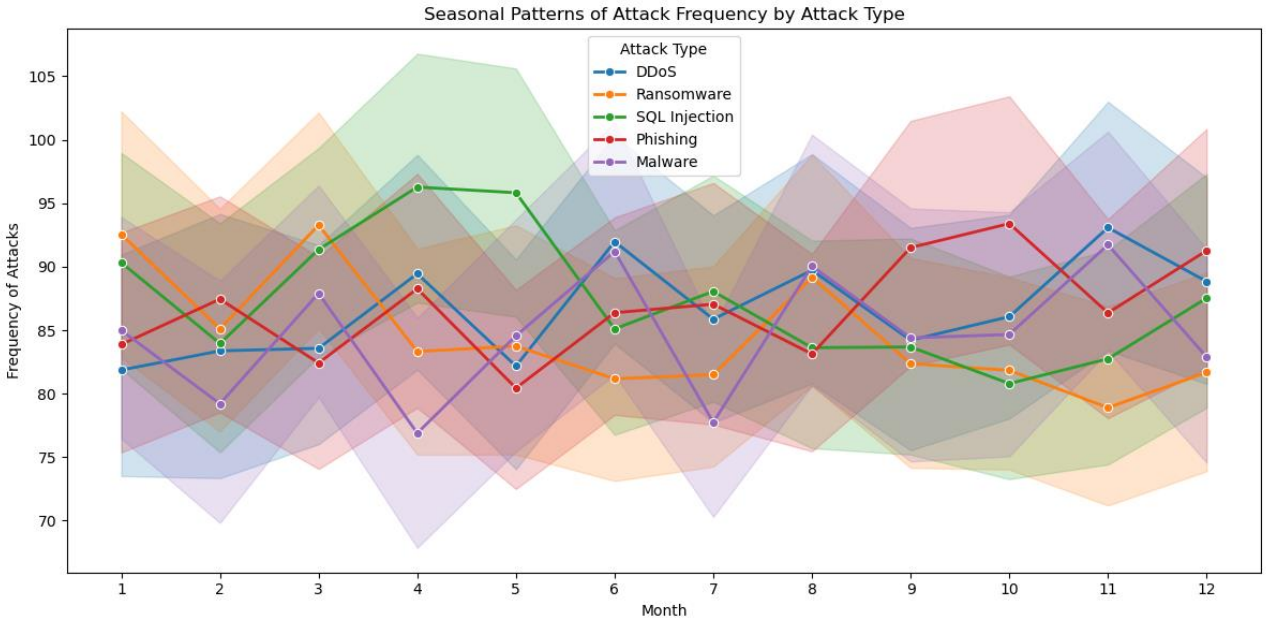
Aikasarja-analyysissä käytettiin kappaleessa ”10.2 Työssä käytetyt datalähteet” mainittuja generoituja datasettejä. Datasetit on generoitu aikavälille 2015–2024. Koska datasetit ovat generoitu, esikäsittelyä ei hirveästi tarvittu. (Ks. kuviot 17 ja 18.) Datasetit esikäsiteltiin ainoastaan muuntamalla päivämäärän sopivaan muotoon sekä ryhmittämällä ne päivämäärän ja hyökkäystyypin mukaan. Jaksollisuutta havainnollistaa kuviot 19–22.

```
          Date    Attack Type  Frequency
0   2015-01-01     Ransomware          6
1   2015-01-01  SQL Injection         82
2   2015-01-01           DDoS         37
3   2015-01-02  SQL Injection         91
4   2015-01-02  SQL Injection         79
(18218, 3)
```

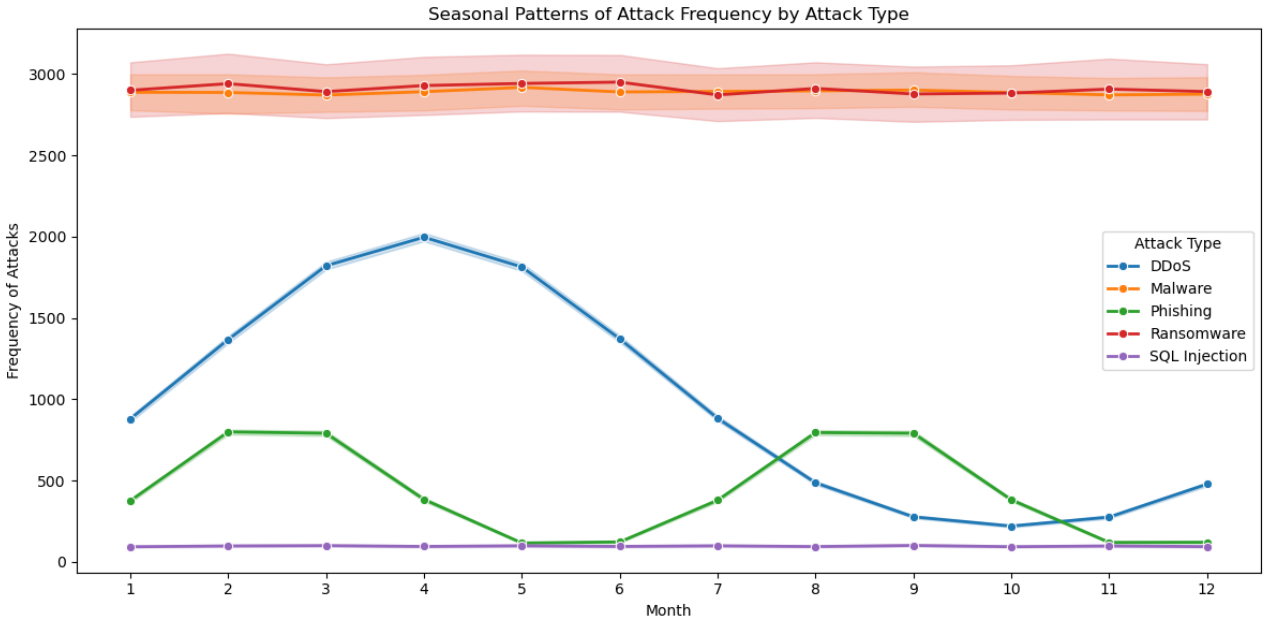
Kuvio 17 Satunnaisesti luotu datasetti ennen esikäsittelyä

```
         Date Attack Type  Frequency
0  2015-01-01        DDoS         33
1  2015-01-01        DDoS         26
2  2015-01-01        DDoS         20
3  2015-01-01        DDoS         26
4  2015-01-01        DDoS         29
(598118, 3)
```

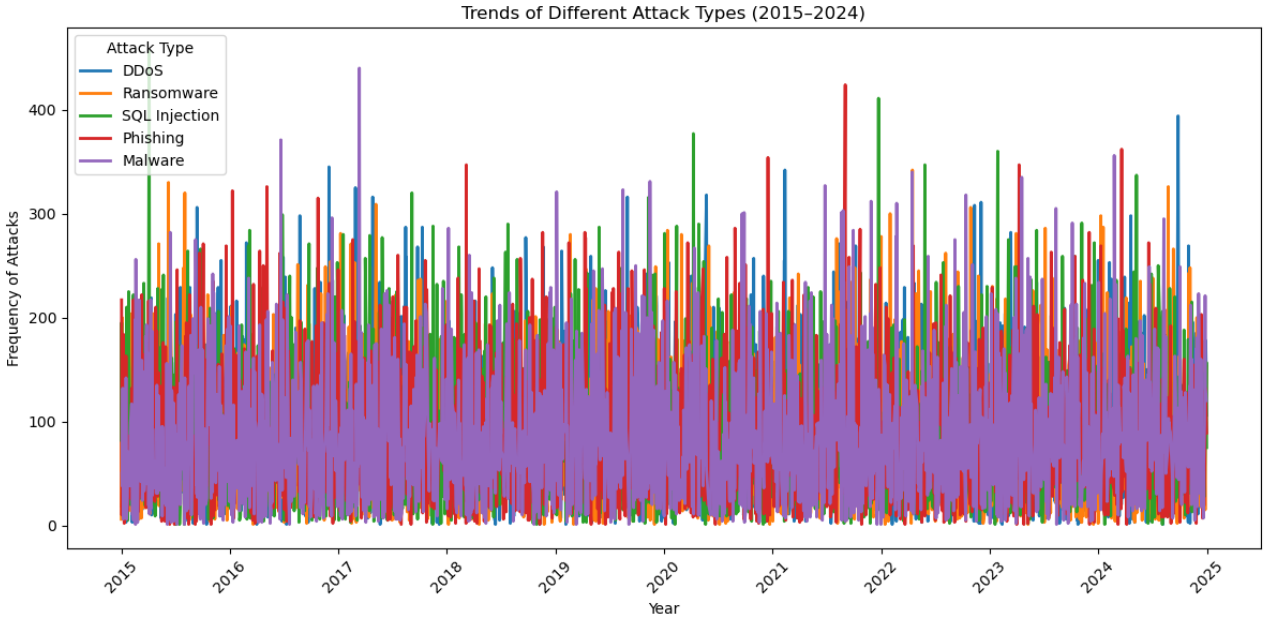
Kuvio 18 Jaksollinen datasetti ennen esikäsittelyä



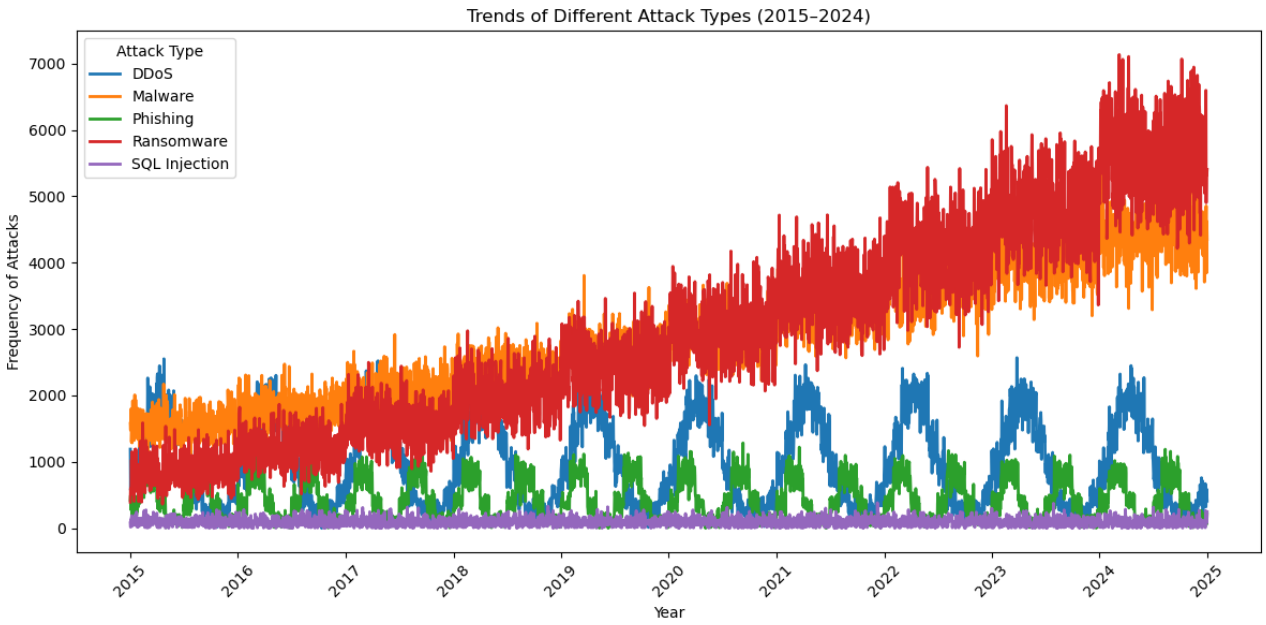
Kuvio 19 Jaksottoman datasetin trendit kuukausittain



Kuvio 20 Jaksollisen datasetin trendit kuukausittain



Kuvio 21 Jaksottoman datasetin trendit koko aikajaksolta



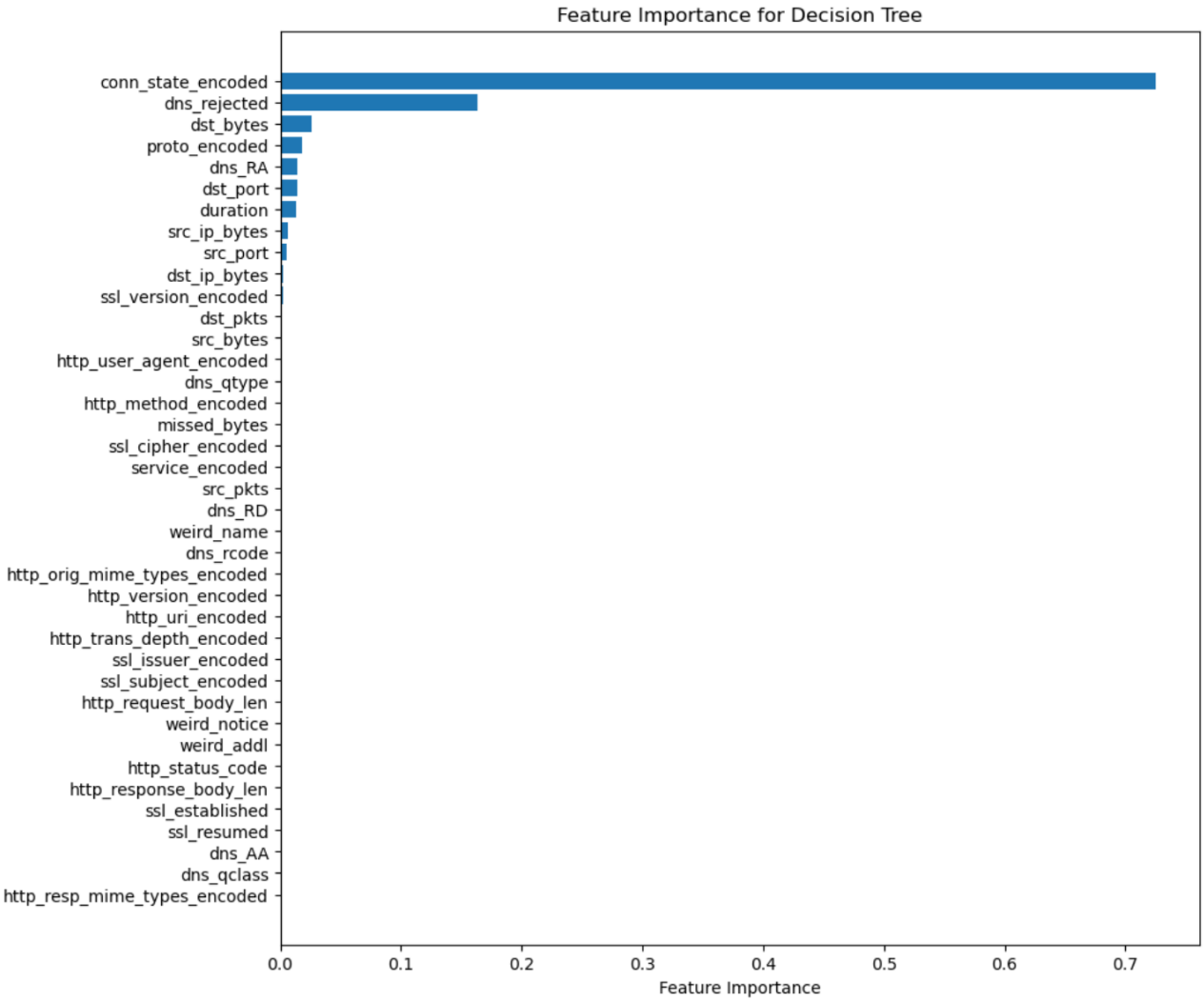
Kuvio 22 Jaksollisen datasetin trendit koko aikajaksolta

9.3 Datan olennaisuus

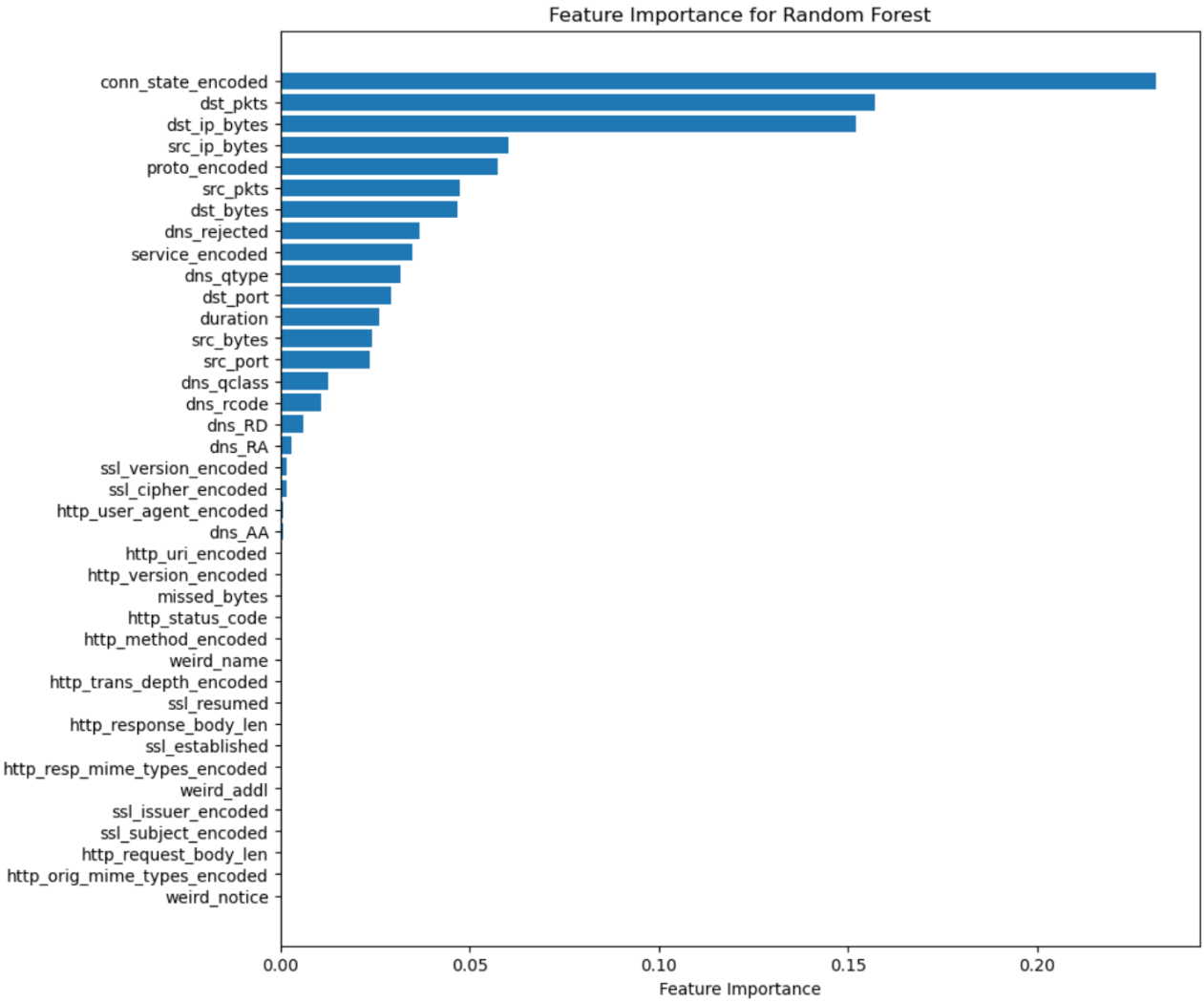
Datan olennaisuus (feature importance) on tärkeä osa koneoppimista. Se kertoo miten hyödyllisiä datan eri ominaisuudet ovat kohdemuuttujan ennustamisessa. Tämän avulla datasetistä hyödynnetään vain sellaiset ominaisuudet, jotka ovat lopputuloksen kannalta olennaisia. Ominaisuuksien olennaisuus on tärkeää käytettäessä etenkin päätöspuita tai satunnaismetsiä ja se voidaan toteuttaa usealla tavalla. Esimerkiksi päätöspuu- ja satunnaismetsä-malleilla on olemassa suoraan metodit, joiden avulla saadaan selville tärkeimmät ominaisuudet. Vaihtoehtoisesti voidaan käyttää permutaatio ominaisuus olennaisuus -metodia (Permutation Feature Importance). (Understanding Feature Importance and Visualization of Tree Models 2024.)

Käytännön osuudessa selvitettiin ominaisuuksien olennaisuuden päätöspuulle sekä satunnaismetsälle verkkoliikennedatalla. Olennaisuudet selvitettiin siten, että ensimmäisenä rakennettiin mallit ja sen jälkeen ne koulutettiin. Seuraavaksi käytettiin DecisionTreeClassifier- ja RandomForestClassifier-mallien metodia ”feature_importances_”, jonka avulla saatiin selville mitkä ominaisuuksista on kaikista tärkeimpiä lopputuloksen kannalta. Näiden mallien olennaisimmat ominaisuudet erosivat toisistaan siten, että päätöspuulla näitä oli vähemmän kuin satunnaismetsällä. (Ks. kuviot 23 ja

24.) Myös se, mitkä ominaisuuksista olivat tärkeimpiä hieman erosi mallien välillä, mutta molemmissa malleissa kaikista tärkeimpänä ominaisuutena oli ”conn_state_encoded”. Tulosten perusteella mallien koulutukseen voidaan ottaa vain kaikista olennaisimmat ominaisuudet, koska muilla ominaisuuksilla ei ole lopputuloksen kannalta väliä. Tämä vähentää myös käsiteltävät datan määrää.



Kuvio 23 Ominaisuuksien olennaisuus päätöspuumallille



Kuvio 24 Ominaisuuksien olennaisuus satunnaismetsämallille

10 Tekoälymallien käytännön testaaminen

10.1 Käytetyt työkalut

Käytännön osuus toteutettiin käyttämällä ohjelmointikielenä Pythonia ja sen erilaisia kirjastoja kuten scikit-learn, pandas, numpy sekä matplotlib. Ohjelmointiin käytettiin Anaconda Navigatoria ja JupyterLabiä. Ohjelmoinnin apuna käytettiin myös ChatGPT:tä.

Python kirjastot

Scikit-learn on avoimen lähdekoodin koneoppimiskirjasto. Se tarjoaa erilaisia työkaluja muun muassa datan esikäsittelyyn sekä mallien valintaan, arviointiin ja sovittamiseen. (Getting Started n.d.)

Pandas on datan analysointiin ja käsittelyyn tarkoitettu Python-paketti. Sen avulla voidaan käsitellä esimerkiksi puuttuvia arvoja, yhdistellä datasettejä sekä ryhmitellä dataa. (Package overview n.d.) NumPy on tieteelliseen laskentaan tarkoitettu kirjasto, joka tarjoaa erilaisia työkaluja aina erilaisista listaobjekteista niille suoritettaviin peruslineaarialgebran ja tilastollisiin toimintoihin (NumPy documentation n.d.). Matplotlib on erilaisiin visuaalisointeihin suunnattu kirjasto. Sen työkalujen avulla voidaan toteuttaa staattisia, animoituja sekä interaktiivisia esityksiä datasta. (Matplotlib: Visualization with Python n.d.)

Anaconda Navigator ja JupyterLab

Anaconda Navigator on käyttöliittymä, joka on tarkoitettu sovellusten käynnistykseen sekä conda-pakettien, - ympäristöjen ja -kanavien hallintaan. Conda on ilmainen, avoimen lähdekoodin sovellus pakettien ja ympäristöjen hallintaan. (Anaconda Navigator n.d.) Anaconda Navigatorin avulla käytettiin JupyterLabiä sekä asennettiin tarvittavia paketteja. JupyterLab on ilmainen kirjoitussovellus ja muokkausympäristö, jossa voidaan luoda ja selittää koodia sekä tutkia ja visualisoida dataa laskennallisiin muistikirjoihin (JupyterLab Documentation n.d.).

10.2 Tietojenkalastelu

Tietojenkalastelu sähköpostien luokitteluun käytettiin sekä syväoppimismallia DistilBERT että koneoppimismallia logistinen regressio. DistilBERT valittiin toteutukseen siksi, koska se on BERT-mallia nopeampi ja vaatii vähemmän muistia. Tässä toteutuksessa käytettiin apuna A Visual Notebook to Using BERT for the First Time -tutoriaalia. (A Visual Notebook to Using BERT for the First TIme.ipynb n.d.)

DistilBERT

Datan esikäsittelyn jälkeen, datasetistä valittiin 4 000 riviä, koska koko datasetin käyttö olisi ollut hidasta. Seuraavaksi ladattiin esikoulutettu DistilBERT-malli sekä DistilBERT-tokenisoija. Näille käytettiin esikoulutettuja painoja ”distilbert-base-uncased”. Esikoulutetuilla painoilla tarkoitetaan numeerisia arvoja, jotka määrittelevät neuronien välisten yhteyksien voimakkuuden ja suunnan neuroverkoissa. Ne vastaavat biologisten neuroverkkojen synapseja. (Weights n.d.)

Seuraavaksi data jaettiin tokenisoijalla sanoihin ja alisanoihin, jotka muutettiin mallille sopivaan formaattiin. Tokenisoitu data sisälsi listan, jossa oli listat jokaisesta tokenisoidusta otsikosta. Nämä listat muunnettiin yhtä pitkiksi lisäämällä lyhyempiin täytettä. Mallille jouduttiin tämän jälkeen luomaan muuttuja, jonka avulla malli sivuutti lisätyt täytteet. Tämän jälkeen pystyttiin ajamaan malli tokenisoidulla datalla, jonka lopputuloksena oli ”last_hidden_states”. Lopputuloksesta otettiin talteen vain se osa, jota tarvittiin jatkossa. Haluttu osa tallennettiin muuttujaan ”features”. Ennen datan jakamista koulutusta ja testausta varten, luotiin ”labels”-muuttuja, johon tallennettiin datasetissä olevat ”labels”-ominaisuuden arvot. Nämä muuttujat jaettiin koulutukseen ja testaukseen, joita käytettiin logistisessa regressiossa. (Katso liite 2, jossa on kokonaisuudessaan tässä käytetty koodi.)

Logistinen regressio

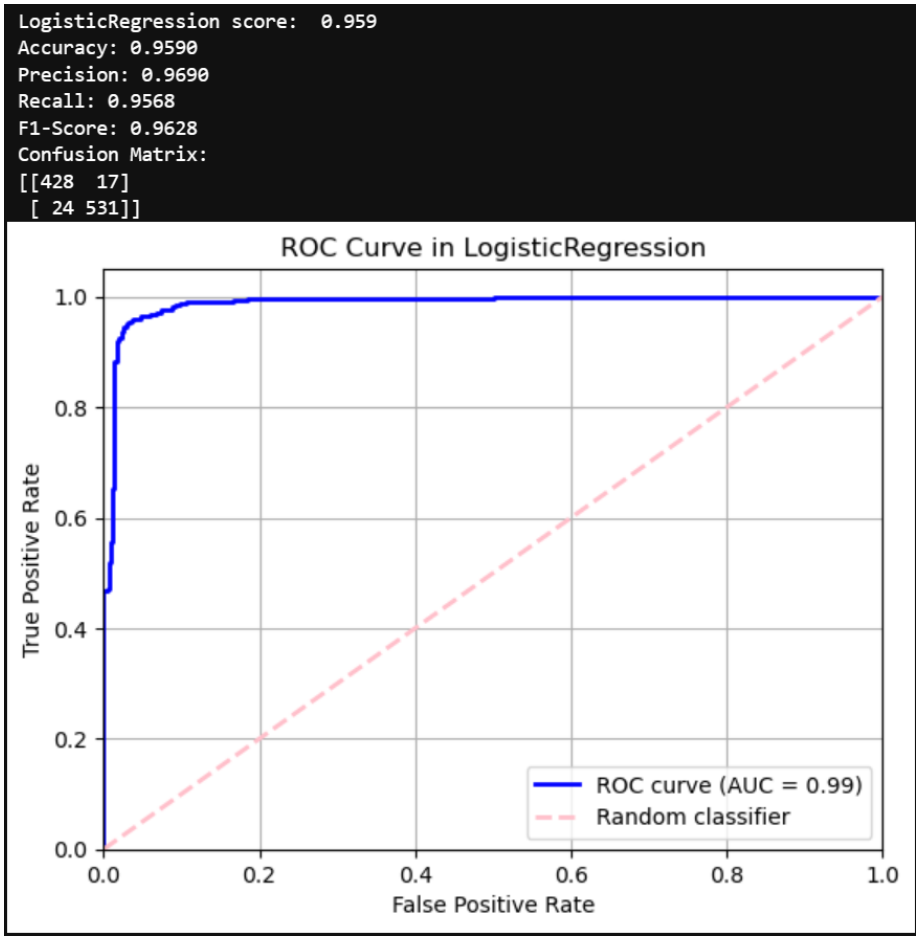
Tämä osa aloitettiin etsimällä paras arvo parametrille C käyttämällä GridSearchCV-työkalua. (Ks. kuvio 25.) Logistisessa regressiossa parametri C määrittää regularisointivahvuuden (regularization strength) (LogisticRegression n.d.). Regularisointi vähentää liian hyvien tulosten määrää (overfitting) (Regularization in Machine Learning 2024). GridSearchCV-työkalu on parametrien säätämisessä käytetty työkalu, joka kokeilee kaikki mahdolliset parametriyhdistelmät (Tuning the hyper-parameters of an estimator n.d.).

```
# Find best C parameter for LogisticRegression

param_grid = {'C': [0.01, 0.1, 1, 10, 100]}
clf = LogisticRegression(penalty='l2', solver='lbfgs', max_iter=1000)

# Perform Grid Search
grid_search = GridSearchCV(clf, param_grid, cv=5, scoring='accuracy')
grid_search.fit(train_features, train_labels)
best_param = grid_search.best_params_['C']
```

Kuvio 25 Paras arvo muuttujalle C

Parhaan arvon löytämisen jälkeen malli koulutettiin koulutusdatalla, minkä jälkeen mallin tarkkuutta testattiin testidatan avulla. Malli sai tarkkuudeksi 0.9590 eli 95,9 %, joka on erinomainen.

(Ks. kuvio 26.) Sekavuusmatriisi antoi tulokseksi vääriä positiivisia 17 kappaletta ja vääriä negatiivisia 24 kappaletta. Väärillä positiivisilla tarkoitetaan sitä, että sähköposti oli luokiteltu tietojenkalasteluksi sen ollessa oikeasti harmiton sähköposti ja väärä negatiivinen on puolestaan päinvastoin.



Kuvio 26 Logistinen regressio sähköpostien luokittelussa

10.3 Haitalliset URLit

Haitallisten URLien luokitteluun valittiin koneoppimismalli satunnaismetsä sekä syväoppimismalli LSTM. Luokittelussa haluttiin kokeilla miten koneoppimismalli ja syväoppimismalli eroavat toisistaan ja saadaanko aikaan erilaisia tuloksia.

Satunnaismetsä

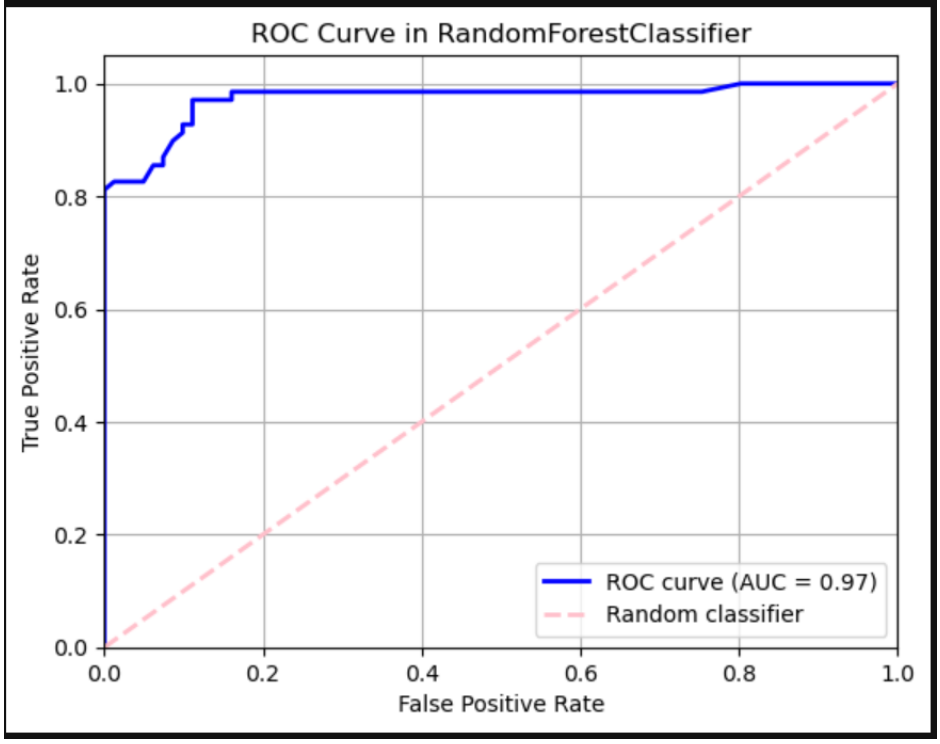
Datan esikäsittelyn jälkeen satunnaismetsämallin luominen aloitettiin valitsemalla datasta satunnaisesti tietty määrä rivejä. Mallia kokeiltiin rivimäärillä 1 000 ja 30 000, koska koko datasetin käyttäminen olisi ollut hidasta. Seuraavaksi valittu data vektorisoitiin TfidVectorizer-työkalulla mallille sopivaan muotoon. Vektorisoinnilla tarkoitetaan tiedon muuttamista numeraaliseen muotoon. Tämän jälkeen mallille valittiin paras ”test_size”-parametrin arvo käyttämällä funktiota ”test_best_test_size”. (Ks. kuvio 27.) ”test_size”-parametrin perusteella käytettävä data jaetaan koulutukseen ja testaukseen.

```
def test_best_test_size(X,y):
    sizes = [0.1, 0.15, 0.2, 0.25, 0.30, 0.35, 0.4, 0.45, 0.5]

    results = {}
    for size in sizes:
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=size, random_state=42)

        # Train Random Forest
        model = RandomForestClassifier(n_estimators=100, random_state=42)
        model.fit(X_train, y_train)

        # Evaluate
        y_pred = model.predict(X_test)
        accuracy = accuracy_score(y_test, y_pred)
        results[size] = accuracy

    return results
```

Kuvio 27 ”test_best_test_size”-funktio

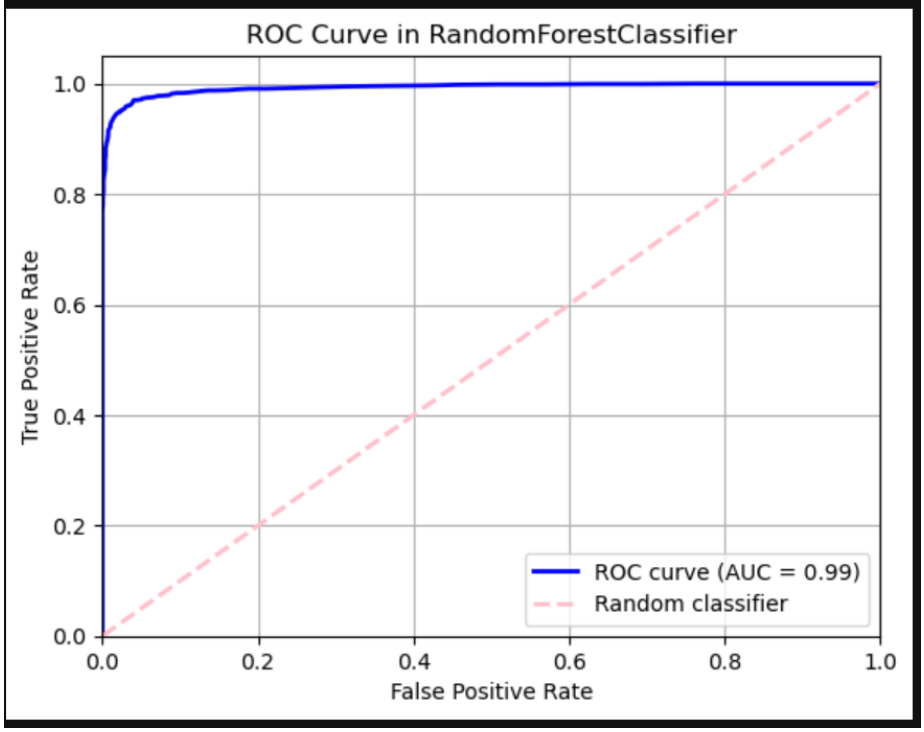
Kun paras arvo parametrille löydettiin, vektorisoitu data jaettiin saadun tuloksen perusteella koulutukseen ja testaukseen. Tämän jälkeen malli koulutettiin koulutukseen tarkoitetulla datalla. Lopuksi malli testattiin ja saatuja tuloksia arvioitiin käyttämällä esimerkiksi luokitteluraporttia sekä sekavuusmatriisia. (Ks. kuvio 28 ja kuvio 29.) Tuloksista selviää, miten suuremmalla datamäärällä mallin tarkkuus paranee. Suuremmalla datamäärällä mallin tarkkuus oli 96 %, joka oli parempi kuin pienemmällä datamäärällä. Luokkien (0 ja 1) väliset erot tarkkuudessa (precision) ja tunnistuksessa (recall) olivat myös pienempiä suuremmalla datamäärällä. (Ks. Liite 3, joka sisältää toteutuksen.)

```
Classification Report for 1000 rows:

              precision    recall  f1-score   support

         0.0       0.84      1.00      0.91        81
         1.0       1.00      0.77      0.87        69

    accuracy                           0.89       150
   macro avg       0.92      0.88      0.89       150
weighted avg       0.91      0.89      0.89       150


Confusion Matrix for 1000 rows:

[[81  0]
 [16 53]]
```



Kuvio 28 Tarkkuus 1 000 rivillä

```
Classification Report for 30000 rows:

              precision    recall  f1-score   support

         0.0       0.94      0.98      0.96      1460
         1.0       0.98      0.94      0.96      1540

    accuracy                           0.96      3000
   macro avg       0.96      0.96      0.96      3000
weighted avg       0.96      0.96      0.96      3000


Confusion Matrix for 30000 rows:

[[1436   24]
 [  89 1451]]
```



Kuvio 29 Tarkkuus 30 000 rivillä

LSTM-malli

LSTM-malliin valittiin satunnaisesti 10 000 riviä esikäsitellystä datasta. Koko datasettiä ei käytetty sillä, se olisi ollut satunnaismetsämallin tavoin hidasta. Valittu data muunnettiin tokenisoijan avulla mallille sopivaan muotoon. Tämän jälkeen käytettiin RandomSearch-työkalua löytämään sopivimmat arvot mallissa käytettäviin kerroksiin. (Ks. kuvio 30.) Tähän meni yhteensä noin yhdeksän tuntia, joka vahvisti sitä, että koko datasetin käyttäminen olisi ollut todella hidasta. Tämän jälkeen malli rakennettiin ja koulutettiin parhailla löydetyillä arvoilla. Lopuksi malli testattiin ja mallin tarkkuudeksi saatiin 95,4 %. (Ks. Liite 4, joka sisältää toteutuksen.)

```
tuner = RandomSearch(
    build_model,
    objective='val_accuracy',
    max_trials=5,
    executions_per_trial=1,
    directory='lstm_tuning',
    project_name='lstm_hyperparam'
)
```

Kuvio 30 RandomSearch-työkalu parametrien löytämiseen

10.4 Verkkoliikenne

Verkkoliikennedatan luokitteluun valittiin koneoppimismallit päätöspuu sekä satunnaismetsä. Mallit valittiin siksi, koska haluttiin vertailla mallien toimivuutta keskenään. Mallit olivat myös tuttuja.

Päätöspuu

Datan esikäsittelyn jälkeen data jaettiin koulutukseen ja testaukseen, minkä jälkeen etsittiin parhaimmat parametrien arvot käyttämällä RandomizedSearchCV-työkalua. (Ks. kuvio 31.) RandomizedSearchCV-työkalu on tarkoitettu parametrien säätämiseen, mutta GridSearchCV:stä poiketen se kokeilee tietyn määrän parametriyhdistelmiä valitsemalla satunnaisesti parametriarvot määritellyistä arvoista (Tuning the hyper-parameters of an estimator n.d.). Seuraavaksi malli koulutettiin saaduilla arvoilla. Lopuksi mallia arvioitiin käyttämällä classification_report-, accuracy_score- ja confusion_matrix- työkaluja.

```
# Define hyperparameter grid
dt_param_grid = {
    'max_depth': range(1,20),
    'min_samples_split': range(2,20),
    'min_samples_leaf': range(1,20)
}

# Perform Randomized Search
dt_search = RandomizedSearchCV(
    dt,
    param_distributions=dt_param_grid,
    n_iter=100,
    cv=5,
    scoring='accuracy',
    random_state=42
)
```

Kuvio 31 RandomizedSearchCV-työkalu päätöspuun käytössä

Satunnaismetsä

Satunnaismetsämalli toteutettiin samalla tyylillä kuin päätöspuumalli. Aluksi etsittiin parhaat parametrin arvot RandomizedSearchCV-työkalulla ja sen jälkeen malli koulutettiin saaduilla arvoilla. Tämän jälkeen mallia testattiin ja sen tarkkuutta arvioitiin samoilla työkaluilla kuin päätöspuumalliakin.

Tulokset

Päätöspuun tarkkuudeksi saatiin 0.99776… eli pyöristettynä 99,78 %. Vääriä positiivisia oli 21 kappaletta ja vääriä negatiivisia 46. Mallin tarkkuus oli lähes 100 %, joka voi mahdollisesti viitata ylisovittamiseen. Satunnaismetsämallin tarkkuudeksi saatiin 99,88 %, joka oli hieman parempi kuin päätöspuumallilla. Väärien positiivisten ja negatiivisten määrät olivat pienemmät kuin päätöspuulla. Koska mahdollisuus ylisovittamiseen oli olemassa, malleja arvioitiin käyttämällä ristiin validointiin tarkoitettua cross_val_score-työkalua. Ristiin validoinnin avulla voidaan luotettavammin arvioida mallin toimivuutta tuntemattomalla datalla. Malli on silloin luotettava, jos ristiin validoinnin tulosten keskiarvo on korkea ja keskihajonta on matala. Molemmat mallit saivat korkeat tulokset keskiarvosta sekä matalat keskihajonnasta, joten näiden tulosten perusteella malleja voidaan pitää luotettavina. (Ks. taulukko 1.)

Taulukko 1 Päätöspuun ja satunnaismetsän vertailu

	Päätöspuu	**Satunnaismetsä**
Tarkkuus	99,78 %	99,88 %
Väärät positiiviset	21	17
Väärät negatiiviset	46	19
Ristiin validoinnin keskiarvo	0.996	0.998
Ristiin validoinnin keskihajonta	0.001	0.000

10.5 Erilaiset hyökkäystyypit aikasarja-analyysissä

Aikasarja-analyysiin valittiin syväoppimismallit RNN ja LSTM. Mallit valittiin siksi, koska niitä voidaan hyödyntää aikasarja-analyysissä sekä ne soveltuvat jaksollisen datan käsittelyyn. Tässä kokeilussa malleja käytettiin kahden datasetin luokitteluun. Aluksi datasetit jaettiin 30 päivän jaksoihin, jotka sitten jaettiin koulutukseen ja testaukseen. (Ks. kuvio 32.)

```
def create_sequences(data, seq_len):
    X, y = [], []
    for i in range(len(data) - seq_len):
        X.append(data.iloc[i:i + seq_len, :])
        y.append(data.iloc[i + seq_len, :])
    return np.array(X), np.array(y)

# Create sequences with length of 30
seq_len = 30
X,y = create_sequences(df_final, seq_len)

print("Shape of sequences 30 days:", X.shape)
print("Shape of targets 30 days:", y.shape)

data_len = math.ceil(len(X) * 0.8)

# Split to train and test
X_train = X[:data_len]
y_train = y[:data_len]
X_test = X[data_len:]
y_test = y[data_len:]
```

Kuvio 32 Datan jaksottaminen ja jakaminen

RNN ja LSTM

Ensimmäisenä lähdettiin säätämään parametrien arvoja käyttämällä RandomSearch-työkalua. Tämän jälkeen mallit koulutettiin ja testattiin sekä jaksottomalla että jaksollisella aikasarjadatasetillä. RNN tuloksia jaksottomaan datasettiin verrattiin LSTM-mallin tuloksiin samalla datasetillä. Myös jaksollisen datasetin tuloksia verrattiin toisiinsa.

Tulokset

Mallit tuottivat paremman tarkkuuden jaksollisella datalla, joka vahvistaa sitä, että mallit toimivat hyvin jaksollisen datan kanssa ja ne löysivät käytetystä jaksollisesta datasta jonkin verran jaksollisuutta. (Ks. taulukko 2.) Tulokset jaksottomasta datasta kertovat siitä, että datassa ei ollut selkeitä

jaksoja, joista mallit olisivat voineet oppia. Mikä olikin tarkoituksena, että jaksottomassa datasetissä hyökkäykset tapahtuvat satunnaisesti. Kuitenkin molempien tulokset olivat heikkoja, joka viittaa mahdollisesti alisovittamiseen (underfitting) eli koulutukseen käytetty datasetti ei ollut tarpeeksi laaja, mallit olivat liian yksinkertaisia tai itse generoitu datasetti ei ollut ominaisuuksiltaan sopiva. Tulokset kuitenkin olivat heikkoja molemmilla dataseteillä. (Ks. liitteet 5 ja 6, joissa visualisoinnit mallien ennusteista ja todellisista arvoista.)

Taulukko 2 Aikasarja-analyysin tulosten vertailu

Malli	**Jaksollinen Data Tarkkuus**	**Jaksoton Data Tarkkuus**
RNN	50 %	15 %
LSTM	51 %	22 %

11 Tulokset ja pohdinta

11.1 Keskeiset havainnot ja niiden merkitys

Tutkimuksen perusteella voidaan todeta, että tekoäly antaa etua erilaisissa kyberuhkatiedustelun osissa. Se on tehokas työkalu, jonka avulla voidaan suorittaa tehtäviä nopeammin ja tehokkaammin kuin mitä ihminen pystyisi. Myös suurien datamassojen käsittely on huomattavasti tehokkaampaa tekoälyn avulla, minkä ansiosta saadaan tuotettua ja koottua kyberuhkatietoa hyvin erilaisista lähteistä. Tekoälyn avulla kyberuhkatietoa saadaan tekstimuotoisen datan lisäksi myös kuvista ja videoista. Hyvin toteutetuilla tekoälymalleilla voidaan saavuttaa hyvinkin tarkkoja tuloksia, joiden avulla voidaan lisätä esimerkiksi kyberturvallisuutta tietojenkalastelukampanjoita vastaan.

Käytännön osuus kuitenkin osoitti myös sen, että vaikka tekoäly onkin erinomainen apu, se ei ole ilmaista. Mallien koulutus ja pyörittäminen vaatii huomattavan määrän laskentatehoa, mikä vaikuttaa käytettävissä oleviin resursseihin. Väärät positiiviset voivat tuhlata resursseja antamalla vääriä hälytyksiä ja väärät negatiiviset puolestaan voivat jättää järjestelmät haavoittuviksi.

Suurimpia haasteita työssä oli oikeanlaisen datan löytäminen ja mallien toteutus. Vaikka tietoa esimerkiksi eri uhkatoimijoista, kampanjoista ja tekniikoista oli saatavilla, sitä ei ollut mahdollista työstää työhön sopivaksi työhön käytettävän ajan puitteissa. Tämän vuoksi työhön jouduttiinkin käyttämään osittain generoitua dataa, joka ei vastaa todellisuutta. Datan saatavuus rajoittikin työtä siten, että käytännön osuus keskittyi lähinnä vain kyberuhkatiedustelun tekniseen osaan.

Mallien toteuttamisessa haasteena oli tiettyjen mallien ja työkalujen vaatima laskentateho, jonka vuoksi kaikkea ei pystytty toteuttamaan tai siihen meni huomattavasti enemmän aikaa kuin oli suunniteltu. Mikä vaikutti työn aikataulutukseen. Esimerkkinä tästä URLien tunnistuksessa käytetyn LSTM-mallin parametrien säätäminen kesti noin yhdeksän tuntia ja siinä käytettiin vain noin 10 % datasetistä.

11.2 Tutkimuksen tavoitteiden saavuttaminen

Työn tavoitteena oli tuottaa kirjallinen tutkimus siitä, millaisia asioita ennakoivaan kyberuhkatiedusteluun kuuluu sekä miten tekoälyä voidaan hyödyntää ennakoivan kyberuhkatiedustelun toteutuksessa ja kehityksessä. Kirjallisen tutkimuksen tukena ja tekoälyn käyttöä kyberuhkatiedusteluun demonstroi työn käytännön osuus. Sen tarkoituksena oli hyödyntää saatavilla olevaa kyberuhkatiedusteluun liittyvää dataa ja kokeilla käytännössä erilaisia tekoälymalleja.

Työ tuotti vastaukset asetettuihin tutkimuskysymyksiin, joita oli:

1. Mitä on ennakoiva kyberuhkatiedustelu?
2. Miten ennakoivaa kyberuhkatiedustelua voidaan toteuttaa tekoälyn avulla?

Kirjallisen tutkimuksen avulla voidaan todeta, että ennakoiva kyberuhkatiedustelu on tulevien trendien ennustamista sekä hyökkäysmallien ja -tapojen tunnistamista aiemman datan perusteella. Nykyisin käytetyt uhkamallit eivät pysty ennakoimaan kovinkaan tehokkaasti puutteellisten tai koko ajan muuttuvien tietojen takia. Myös sopeutuminen uusiin uhkiin on vaikeaa nykyisille uhkamalleille. Työssä saatiin selville, että tekoälyn avulla voidaan toteuttaa sellaisia työkaluja, jotka pystyvät tuottamaan ennusteita ja analysoimaan tietoa, vaikka se olisikin puutteellista tai tiedot

muuttuisivat. Tekoäly pystyy kehittämään ennakoivaa kyberuhkatiedustelua muun muassa nopeuttamalla datan käsittelyä, analysoimalla dataa erilaisista lähteistä sekä tunnistamaan varhaisia merkkejä lähestyvästä hyökkäyksestä.

Käytännön osuudessa saatiin etsittyä kyberuhkatiedusteluun soveltuvaa dataa, jota päästiin soveltamaan tekoälymallien käyttöön. Tekoälymalleja päästiin käyttämään erilaisilla dataseteillä. Käytännön osuudesta kuitenkin jäi puuttumaan esimerkiksi erilaisten lokien analysointi sekä kyberuhkatoimijoiden ja niiden toimintamallien perusteella tehdyt analyysit ja profiloinnit.

11.3 Työn luotettavuus

Työssä käytettiin laajasti erilaisista lähteistä löytyvää ajantasaista tietoa. Lähteinä pyrittiin käyttämään esimerkiksi ohjelmistojen virallisia sivustoja, joissa on oikeelliset tiedot. Käytännön osuudessa käytetyt datasetit olivat sellaisia, jotka eivät vaatineet erityisiä lisenssejä tai ne oli tuotettu skriptin avulla. Termien käännösvirheiden tai virheellisten tulkintojen ehkäisemiseksi suomennettujen termien jälkeen mainittiin englanninkieliset vastineet.

Tekoälyä käytettiin työssä Jamkin ohjeistuksia noudattaen. Sitä käytettiin työssä lähinnä käytännön osuudessa erilaisten koodien tuottamiseen tai korjaamiseen ja auttamaan ongelmatilanteissa. Tekoälyä käytettiin tarpeen mukaan myös englanninkielisten lähdetekstien käännöksissä tai sopivien suomenkielisten ammattitermien löytämisessä. Tilanteen mukaan tekoälyn antamat vastaukset tarkistettiin luotettavista lähteistä.

12 Yhteenveto

12.1 Tekoälyn tulevaisuus kyberuhkatiedustelussa

Tulevaisuudessa tekoälyn merkitys kasvaa entisestään. Sitä pystytään hyödyntämään lukemattomilla tavoilla eikä sen käyttö rajoitu pelkästään kyberpuolustukseen. On äärimmäisen tärkeää ottaa huomioon tekoälyn hyödyntäminen rikolliseen tai muuhun haitalliseen käyttöön. Tämä tarkoittaa kyberuhkatiedustelun kannalta sitä, että tulevaisuudessa uhkat ovat vaikeammin tunnistettavia tekoälyllä ja niistä tulee entistä edistyneempiä. Tekoälyn avulla myös hyökkäysten määrät ja vaikutukset kasvavat.

Tekoälymallien kehittyessä, ne vaativat yhä enemmän laskentatehoa, joka lisää myös energiatarpeen kasvua. Erittäin tehokkaat tekoälyratkaisut kasvattavat organisaatioiden resurssien tarvetta, mikä vaikuttaa siihen, etteivät kaikki organisaatiot pääse hyödyntämään tekoälyä tarvitsemallaan tavalla. Mikä pakottaakin tuottamaan uusia innovaatioita, joiden avulla tekoälyä pystytään hyödyntämään esimerkiksi kyberuhkatiedusteluun ja jotka olisivat kaikkien saatavilla.

Lisääntynyt tekoälyn käyttö vaikuttaa kyberturvallisuuteen. Ihmiset ovat alkaneet käyttää entistä enemmän tekoälyä, joka kasvattaa kyberturvallisuuden tarvetta. Tekoälylle syötetään arkaluontoista tietoa ja sen tuottamaan sisältöä, esimerkiksi koodia, kopioidaan suoraan käyttöön tarkistamatta sen luotettavuutta. Tekoälyn avulla pystytään myös luomaan yhä realistisempia kuvia ja videoita, joiden avulla voidaan suorittaa erilaisia hyökkäyksiä. Tämä kasvattaa tarvetta kuvien ja videoiden analysoinnille kyberuhkatiedustelussa.

12.2 Kehittämisideat ja jatkotutkimus

Ennakoivan kyberuhkatiedustelun kehittäminen tekoälyn avulla olisi hyvä aloittaa siitä, että tekoäly ratkaisut olisivat mahdollisimman usean käytettävissä. Erilaisia ratkaisuja voitaisiin toteuttaa yhteistyössä erilaisten tahojen kanssa. Esimerkiksi kyberuhkatiedusteluun perehtyneet toisivat esille kriittisimmät kehityskohteen, joiden perusteella tekoälyratkaisuja tarjoavat kehittäisivät sopivia ratkaisuja. Yhteistyön ei pitäisi rajoittua pelkästään eri alojen välille, vaan se pitäisi laajentaa koskemaan myös eri maita. Bolen (2024) toteaa, että kyberuhkatiedustelu edellyttää tulevaisuudessa entistä avoimempaa ja yhteistyöhön perustuvaa mallia, sillä usein uhkatoimijat toimivat yli rajojen, joka altistaa organisaatioita, hallituksia ja toimialoja, koska kaikilla ei ole samaa tietoa.

Tekoälyn avulla voidaan lisätä automaattisia järjestelmiä kyberuhkatiedusteluun kuten reagointijärjestelmiä, jotka häiriötilanteessa automaattisesti voivat estää havaitun epätavallisen toiminnan. Tekoälymalleja voitaisiin kehittää tunnistamaan geopoliittisten konfliktien vaikutuksia hyökkäyksiin ja näiden pohjalta luoda ennusteita, miten tulevaisuudessa hyökkäykset tapahtuvat geopoliittisten konfliktien aikana. Hyödyntämällä NLP-malleja erilaisten sosiaalisen median alustojen ja pimeän verkon keskustelupalstojen tiedon keräykseen ja analysointiin voidaan tuottaa uhkatietoa esimerkiksi uusista uhkatoimijoista.

Konkreettisempana jatkotutkimuksena tälle työlle olisi esimerkiksi luoda olemassa olevasta kyberuhkatiedosta datasettejä, joiden avulla voitaisiin rakentaa tekoälymalleja uhkatekijöiden tunnistamiseen, profilointiin tai tunnistamaan esimerkiksi, miten hyökkäyksissä käytetyt tekniikat ovat muuttuneet ajansaatossa ja onko havaittavissa tiettyjä trendejä. Myös esimerkiksi voitaisiin tutkia, miten tekoälyä käytetään nykyisissä kyberuhkatiedusteluun käytettävissä työkaluissa ja miten tekoälytyökaluja voitaisiin näihin integroida.

Lähteet

A Visual Notebook to Using BERT for the First TIme.ipynb. N.d. Tiedosto. Colab. Viitattu 2.2.2025. https://colab.research.google.com/github/jalammar/jalammar.github.io/blob/master/notebooks/bert/A_Visual_Notebook_to_Using_BERT_for_the_First_Time.ipynb#scrollTo=izA3-6kffbdT.

Aalto-yliopiston tutkimusportaali. N.d. Aalto-yliopisto. Verkkopalvelu. Viitattu 27.1.2025. https://research.aalto.fi/fi/.

AI in Cybersecurity: Revolutionizing threat detection and defense. 2023. Artikkeli. Viitattu 9.12.2024. https://datasciencedojo.com/blog/ai-in-cybersecurity/.

AI in Threat Intelligence. N.d. Artikkeli Silobreakerin verkkosivuilla. Viitattu 9.12.2024. https://www.silobreaker.com/glossary/ai-in-threat-intelligence/.

Al-Subaiey, A., Al-Thani, M., Alam, N. A., Antora, K. F., Khandakar, A., & Zaman, S. A. U. 2024. Novel Interpretable and Robust Web-based AI Platform for Phishing Email Detection. Artikkeli. ArXiv.org. Viitattu 29.1.2025. https://doi.org/10.48550/arXiv.2405.11619.

Anaconda Navigator. N.d. Dokumentti. Anaconda. Viitattu 2.2.2025. https://docs.anaconda.com/navigator/.

Awan, A. 2023. What is Labeled Data? Artikkeli. Viitattu 7.11.2024. https://www.datacamp.com/blog/what-is-labeled-data.

AZ STUDIO. 2024. What is Cybersecurity? Types, Threats and Cyber Safety Tips. Artikkeli. Viitattu 30.1.2025. https://medium.com/@az065483/what-is-cybersecurity-types-threats-and-cyber-safety-tips-c3a2b333faf6.

Baker, K. 2023. What is Cyber Threat Intelligence? Artikkeli. Julkaistu 23.03.2023. Viitattu 7.10.2024. https://www.crowdstrike.com/cybersecurity-101/threat-intelligence/.

Baker, K. 2024. 12 Most Common Types of Cyberattacks. Artikkeli. Julkaistu 14.05.2024. Viitattu 7.10.2024. https://www.crowdstrike.com/cybersecurity-101/cyberattacks/most-common-types-of-cyberattacks/.

Bartels, J. 2024. Harnessing Predictive Analytics In Cybersecurity. Artikkeli. Viitattu 16.1.2025. https://www.biia.com/harnessing-predictive-analytics-in-cybersecurity/.

BERT. N.d. Artikkeli. NVIDIA. Viitattu 1.2.2025. https://www.nvidia.com/en-us/glossary/bert/.

Bolen, S. 2024. Cyber Threat Intelligence 2025: How AI and Human Expertise Will Revolutionize Cybersecurity. Artikkeli. Viitattu 26.1.2025. https://medium.com/@scottbolen/cyber-threat-intelligence-2025-how-ai-and-human-expertise-will-revolutionize-cybersecurity-73f7c9a7c711.

Borges, E. 2024. 4 Main Threat Actor Types Explained for Better Proactive Defense. Artikkeli. Viitattu 30.1.2025. https://www.recordedfuture.com/threat-intelligence-101/threat-actors/threat-actor-types.

Brown, S. 2021. Machine learning, explained. Artikkeli. MIT Sloan School of Managementin verkkosivut. Viitattu 7.11.2024. https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained.

Copeland, B.J. 2024. artificial intelligence. Artikkeli. Britannica- tietosanakirja. Päivitetty 6.11.2024. viitattu 7.11.2024. https://www.britannica.com/technology/artificial-intelligence.

Cyber Threat Intelligence: A Comprehensive Guide. N.d. Artikkeli Palo Alton verkkosivuilla. Viitattu 16.1.2025. https://www.paloaltonetworks.com/cyberpedia/cyber-threat-intelligence.

Dark Web Profile: Gamaredon APT. 2024. Blogi. Viitattu 30.1.2025. https://socradar.io/dark-web-profile-gamaredon-apt/.

Dark Web Profile: IntelBroker. 2024. Blogi. Viitattu 30.1.2025. https://socradar.io/dark-web-profile-intelbroker/.

Data Normalization Machine Learning. 2024. Artikkeli. Päivitetty 4.11.2024. Viitattu 27.1.2025. https://www.geeksforgeeks.org/what-is-data-normalization/.

De, D. 2018. RNN or Recurrent Neural Network for Noobs. Artikkeli. Viitattu 26.1.2025. https://medium.com/hackernoon/rnn-or-recurrent-neural-network-for-noobs-a9afbb00e860.

Dhanushkodi, K & Thejas, S. 2024. AI Enabled Threat Detection: Leveraging Artificial Intelligence for Advanced Security and Cyber Threat Mitigation. Tutkielma. IEEE. Julkaistu 8.11.2024. Viitattu 12.12.2024. https://ieeexplore.ieee.org/document/10747338.

Difference between DOS and DDOS attack. 2024. Artikkeli. Viitattu 9.2.2025. https://www.geeksforgeeks.org/difference-between-dos-and-ddos-attack/.

DistilBERT. N.d. Dokumentti. Hugging Face. Viitattu 1.2.2025. https://huggingface.co/docs/transformers/model_doc/distilbert#transformers.DistilBertModel.

Dristhi. 2022. Introduction to DistilBERT in Student Model. Blogi. Viitattu 9.2.2025. https://www.analyticsvidhya.com/blog/2022/11/introduction-to-distilbert-in-student-model/.

Election Security Spotlight – Signature-Based vs Anomaly-Based Detection. N.d. Artikkeli. Viitattu 16.1.2025. https://www.cisecurity.org/insights/spotlight/cybersecurity-spotlight-signature-based-vs-anomaly-based-detection.

Getting Started. N.d. Dokumentti. scikit-learn. Viitattu 1.2.2025. https://scikit-learn.org/stable/getting_started.html.

Gillis, A. 2023. risk assessment. Artikkeli TechTargetin verkkosivuilla. Päivitetty lokakuussa 2023. Viitattu 9.12.2024. https://www.techtarget.com/searchsecurity/definition/risk-assessment.

Hämäläinen, R. 2024. Ongelma ja ratkaisu. Luentomateriaali Jyväskylän ammattikorkeakoulun Moodle-verkkotyötilassa. Viitattu 24.10.2024. https://moodle.jamk.fi.

How to Use Kaggle. N.d. Dokumentti Kagglen verkkosivuilla. Viitattu 27.1.2025. https://www.kaggle.com/docs/datasets.

Introduction to Recurrent Neural Networks. 2024. Artikkeli. Päivitetty 15.11.2024. Viitattu 26.1.2025. https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/.

IP-osoite – määritelmä ja selitys. N.d. Artikkeli Kasperskyn verkkosivuilla. Viitattu 18.11.2024. https://www.kaspersky.fi/resource-center/definitions/what-is-an-ip-address.

JupyterLab Documentation. N.d. Dokumentti. Viitattu 3.2.2025. https://jupyterlab.readthedocs.io/en/latest/.

Kagglen etusivu. N.d. Kaggle. Verkkosivu. Viitattu 27.1.2025. https://www.kaggle.com/.

Kolari, J. & Kallio, A. 2023. Tekoäly 123. Docendo. Viitattu 21.10.2024. https://janet.finna.fi, Ellibslibrary.

Kumar, V. 2024. 08] Data Encoding In Machine Learning: Different Data Encodings: OneHotEncoding(), OrdinalEncoding() & LabelEncoding(). Artikkeli. Viitattu 27.1.2025. https://medium.com/@vinodkumargr/08-data-encoding-in-machine-learning-different-data-encodings-onehotencoding-65f736d31ecf.

Lange, K. 2024. What Is Threat Analysis? Artikkeli. Viitattu 9.12.2024. https://www.splunk.com/en_us/blog/learn/threat-analysis.html.

Lawton, G. 2024. The different types of machine learning explained. Artikkeli. Viitattu 7.11.2024. https://www.techtarget.com/searchenterpriseai/tip/Types-of-learning-in-machine-learning-explained.

Lindemulder, G. & Kosinski, M. 2024. What is cybersecurity? Artikkeli. Viitattu 30.1.2025. https://www.ibm.com/think/topics/cybersecurity.

LogisticRegression. N.d. Dokumentti. scikit-learn. Viitattu 2.2.2025. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

Marchal, S. 2014. PhishStorm - phishing / legitimate URL dataset. Datasetti. Aalto-yliopisto. urlset(v.zip). Viitattu 29.1.2025. https://doi.org/10.24342/f49465b2-c68a-4182-9171-075f0ed797d5.

Matplotlib: Visualization with Python. N.d. Verkkosivu. Matplotlib. Viitattu 1.2.2025. https://matplotlib.org/.

Mikä on esineiden internet (IoT)? N.d. Artikkeli SAP-verkkosivuilla. Viitattu 18.11.2024. https://www.sap.com/finland/products/artificial-intelligence/what-is-iot.html.

Mitä on kyberturvallisuus? N.d. Artikkeli. F-Securen verkkosivut. Viitattu 2.10.2024. https://www.f-secure.com/fi/articles/what-is-cyber-security.

Mitä vaarantumisindikaattorit (IOC:t) ovat? N.d. Artikkeli Microsoftin verkkosivuilla. Viitattu 18.11.2024. https://www.microsoft.com/fi-fi/security/business/security-101/what-are-indicators-of-compromise-ioc.

ML | Data Preprocessing in Python. 2025. Artikkeli. Päivitetty 17.1.2025. Viitattu 27.1.2025. https://www.geeksforgeeks.org/data-preprocessing-machine-learning-python/.

Morgan, C. 2024. 5 Critical Threat Actors You Need to Know About. Blogi. Viitattu 30.1.2025. https://www.reliaquest.com/blog/5-critical-threat-actors-you-need-to-know-about/.

Moustafa, N & Slay, J. 2015. Military Communications and Information Systems Conference (MilCIS). Tutkimus. IEEE. Verkkoliikennedatasetti. Viitattu 29.1.2025. https://ieeexplore.ieee.org/abstract/document/7348942.

Moustafa, N & Slay, J. 2016. Information Security Journal: A Global Perspective. Artikkeli. Verkkoliikennedatasetti. Viitattu 29.1.2025. https://www.tandfonline.com/doi/abs/10.1080/19393555.2015.1125974.

Moustafa, N. 2021. The UNSW-NB15 Dataset. Datasetin esittelysivu. Viitattu 29.1.2025. https://research.unsw.edu.au/projects/unsw-nb15-dataset.

Moustafa, N. 2024. Description of Network Features. PDF-dokumentti tekijän OneDrivessa. Liite 1. Viitattu 29.1.2025. https://unsw-my.sharepoint.com/personal/z5025758_ad_unsw_edu_au/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fz5025758%5Fad%5Funsw%5Fedu%5Fau%2FDocuments%2FTON%5FIoT%20datasets&ga=1.

Moustafa, N., Creech, G. & Slay, J. 2017. Big Data Analytics for Intrusion Detection System: Statistical Decision-Making Using Finite Dirichlet Mixture Models. Springer, Cham. Verkkoliikennedatasetti. Viitattu 29.1.2025. https://doi.org/10.1007/978-3-319-59439-2_5.

Moustafa, N., Slay, J. & Creech, G. 2017. Novel Geometric Area Analysis Technique for Anomaly Detection Using Trapezoidal Area Estimation on Large-Scale Networks. Artikkeli. Verkkoliikennedatasetti. Viitattu 29.1.2025. https://ieeexplore.ieee.org/abstract/document/7948715.

Murel, J. & Kavlakoglu, E. 2024. What is dimensionality reduction? Artikkeli. Viitattu 7.11.2024. https://www.ibm.com/topics/dimensionality-reduction.

NumPy documentation. N.d. Dokumentti. NumPy. Viitattu 1.2.2025. https://numpy.org/doc/stable/.

Package overview. N.d. Dokumentti. pandas. Viitattu 1.2.2025. https://pandas.pydata.org/docs/getting_started/overview.html.

Phishing Detection Techniques. N.d. Artikkeli. Viitattu 1.2.2025. https://www.checkpoint.com/cyber-hub/threat-prevention/what-is-phishing/phishing-detection-techniques/.

Random Forest Algorithm in Machine Learning. 2025. Artikkeli. Päivitetty 16.1.2025. Viitattu 27.1.2025. https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/.

Regularization in Machine Learning. 2024. Artikkeli. Viitattu 3.2.2025. https://www.geeksforgeeks.org/regularization-in-machine-learning/.

Sarhan, M., Layeghy, S., Moustafa, N. & Portmann, M. 2020. Tutkimus. Verkkoliikennedatasetti. Viitattu 29.1.2025. https://doi.org/10.48550/arXiv.2011.09144.

Time Series Analysis and Forecasting. 2024. Artikkeli. Päivitetty 13.8.2024. Viitattu 26.1.2025. https://www.geeksforgeeks.org/time-series-analysis-and-forecasting/.

Tuning the hyper-parameters of an estimator. N.d. Dokumentissa User Guide. scikit-learn. Viitattu 8.2.2025. https://scikit-learn.org/stable/modules/grid_search.html.

Tuychiev, B. 2023. A Comprehensive Introduction to Anomaly Detection. Artikkeli DataCampin verkkosivuilla. Viitattu 16.1.2025. https://www.datacamp.com/tutorial/introduction-to-anomaly-detection.

Types of cyberthreats. 2024. Artikkeli. IBM:n verkkosivut. Julkaistu 25.03.2024. Viitattu 7.10.2024. https://www.ibm.com/think/topics/cyberthreats-types.

Understanding Feature Importance and Visualization of Tree Models. 2024. Artikkeli. Päivitetty 4.6.2024. Viitattu 27.1.2025. https://www.geeksforgeeks.org/understanding-feature-importance-and-visualization-of-tree-models/.

Understanding SQL Injection. N.d. Dokumentti. Cisco. Viitattu 9.2.2025. https://sec.cloudapps.cisco.com/security/center/resources/sql_injection.html.

URL-osoite. N.d. Dokumentti. Google. Viitattu 11.2.2025. https://support.google.com/google-ads/answer/14095?hl=fi.

Vulnerability Assessment. N.d. Artikkeli. Viitattu 9.12.2024. https://www.blackduck.com/glossary/what-is-vulnerability-assessment.html.

Weights. N.d. Artikkeli. TED AI San Francisco. Viitattu 2.2.2025. https://tedai-sanfrancisco.ted.com/glossary/weights/.

What is a decision tree? N.d. Artikkeli IBM:n verkkosivuilla. Viitattu 29.1.2025. https://www.ibm.com/think/topics/decision-trees.

What is a domain name? | Domain name vs. URL. N.d. Artikkeli. Viitattu 11.2.2025. https://www.cloudflare.com/learning/dns/glossary/what-is-a-domain-name/.

What is a threat actor? 2023. Artikkeli IBM:n verkkosivuilla. Viitattu 16.1.2025. https://www.ibm.com/think/topics/threat-actor.

What is Deep Learning? N.d. Dokumentti Google Cloudissa. Viitattu 14.11.2024. https://cloud.google.com/discover/what-is-deep-learning.

What is Heuristic Analysis? N.d. Artikkeli. Viitattu 16.1.2025. https://www.kaspersky.com/resource-center/definitions/heuristic-analysis.

What is LSTM – Long Short Term Memory? 2024. Artikkeli. Päivitetty. 10.6.2024. Viitattu 26.1.2025. https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/.

What is the future of cybersecurity? 2024. Blogi-kirjoitus. Julkaistu 28.5.2024. Viitattu 25.9.2024. https://fieldeffect.com/blog/what-is-the-future-of-cyber-security.

What Is the Role of AI in Threat Detection? N.d. Artikkeli Paloalton verkkosivuilla. Viitattu 9.12.2024. https://www.paloaltonetworks.com/cyberpedia/ai-in-threat-detection#ai.

Liitteet

Liite 1. Description of Network Features

Service profile: Connection activity			
ID	Feature	Type	Description
1	ts	Time	Timestamp of connection between flow identifiers
2	src_ip	String	Source IP addresses which originate endpoints' IP addresses
3	src_port	Number	Source ports which Originate endpoint's TCP/UDP ports
4	dst_ip	String	Destination IP addresses which respond to endpoint's IP addresses
5	dst_port	Number	Destination ports which respond to endpoint's TCP/UDP ports
6	proto	String	Transport layer protocols of flow connections
7	service	String	Dynamically detected protocols, such as DNS, HTTP and SSL
8	duration	Number	The time of the packet connections, which is estimated by subtracting 'time of last packet seen' and 'time of first packet seen'
9	src_bytes	Number	Source bytes which are originated from payload bytes of TCP sequence numbers
10	dst_bytes	Number	Destination bytes which are responded payload bytes from TCP sequence numbers
11	conn_state	String	Various connection states, such as S0 (connection without replay), S1 (connection established), and REJ (connection attempt rejected)
12	missed_bytes	Number	Number of missing bytes in content gaps

Service profile: Statistical activity			
ID	Feature	Type	Description
13	src_pkts	Number	Number of original packets which is estimated from source systems
14	src_ip_bytes	Number	Number of original IP bytes which is the total length of IP header field of source systems
15	dst_pkts	Number	Number of destination packets which is estimated from destination systems

16	dst_ip_bytes	Number	Number of destination IP bytes which is the total length of IP header field of destination systems

Service profile: DNS activity			
ID	Feature	Type	Description
17	dns_query	string	Domain name subjects of the DNS queries
18	dns_qclass	Number	Values which specifies the DNS query classes
19	dns_qtype	Number	Value which specifies the DNS query types
20	dns_rcode	Number	Response code values in the DNS responses
21	dns_AA	Bool	Authoritative answers of DNS, where T denotes server is authoritative for query
22	dns_RD	Bool	Recursion desired of DNS, where T denotes request recursive lookup of query
23	dns_RA	Bool	Recursion available of DNS, where T denotes server supports recursive queries
24	dns_rejected	Bool	DNS rejection, where the DNS queries are rejected by the server

Service profile: SSL activity			
ID	Feature	Type	Description
25	ssl_version	String	SSL version which is offered by the server
26	ssl_cipher	String	SSL cipher suite which the server chose
27	ssl_resumed	Bool	SSL flag indicates the session that can be used to initiate new connections, where T refers to the SSL connection is initiated
28	ssl_established	Bool	SSL flag indicates establishing connections between two parties, where T refers to establishing the connection
29	ssl_subject	String	Subject of the X.509 cert offered by the server
30	ssl_issuer	String	Trusted owner/originator of SLL and digital certificate (certificate authority)

Service profile: HTTP activity			
ID	Feature	Type	Description
31	http_trans_depth	Number	Pipelined depth into the HTTP connection

32	http_method	String	HTTP request methods such as GET, POST and HEAD
33	http_uri	String	URIs used in the HTTP request
35	http_version	String	The HTTP versions utilised such as V1.1
36	http_request_body_len	Number	Actual uncompressed content sizes of the data transferred from the HTTP client
37	http_response_body_len	Number	Actual uncompressed content sizes of the data transferred from the HTTP server
38	http_status_code	Number	Status codes returned by the HTTP server
39	http_user_agent	Number	Values of the User-Agent header in the HTTP protocol
40	http_orig_mime_types	String	Ordered vectors of mime types from source system in the HTTP protocol
41	http_resp_mime_types	String	Ordered vectors of mime types from destination system in the HTTP protocol

Service profile: Violation activity			
ID	Feature	Type	Description
42	weird_name	String	Names of anomalies/violations related to protocols that happened
43	weird_addl	String	Additional information is associated to protocol anomalies/violations
44	weird_notice	bool	It indicates if the violation/anomaly was turned into a notice

Service profile: Data labelling			
ID	Feature	Type	Description
45	label	Number	Tag normal and attack records, where 0 indicates normal and 1 indicates attacks
46	type	String	Tag attack categories, such as normal, DoS, DDoS and backdoor attacks, and normal records

Liite 2. Tietojenkalastelu DistilBERT-mallilla

```
import transformers as ppb
import torch
import numpy as np

# Model and tokenizer classes
model_class, tokenizer_class = (ppb.DistilBertModel, ppb.DistilBertTokenizer)

# Load pretrained model and tokenizer
model = model_class.from_pretrained('distilbert-base-uncased')
tokenizer = tokenizer_class.from_pretrained('distilbert-base-uncased')

# Tokenize subjects
tokenized = X['subject'].apply((lambda x: tokenizer.encode(x, add_special_tokens=True)))

# Convert list of lists to list
max_len = 0
for i in tokenized.values:
    if len(i) > max_len:
        max_len = len(i)

# Add padding
padded = np.array([i + [0]*(max_len-len(i)) for i in tokenized.values])

# For ignoring padding
attention_mask = np.where(padded != 0, 1, 0)
attention_mask.shape

# Inputs with padding
input_ids = torch.tensor(padded)

# Ignore padding
attention_mask = torch.tensor(attention_mask)

# Run model with tokenized, padded data with ignoring padding
with torch.no_grad():
    last_hidden_states = model(input_ids, attention_mask=attention_mask)

# Slice CLS from last_hidden_states. The first token of every subject
features = last_hidden_states[0][:,0,:].numpy()
labels = X['label']
```

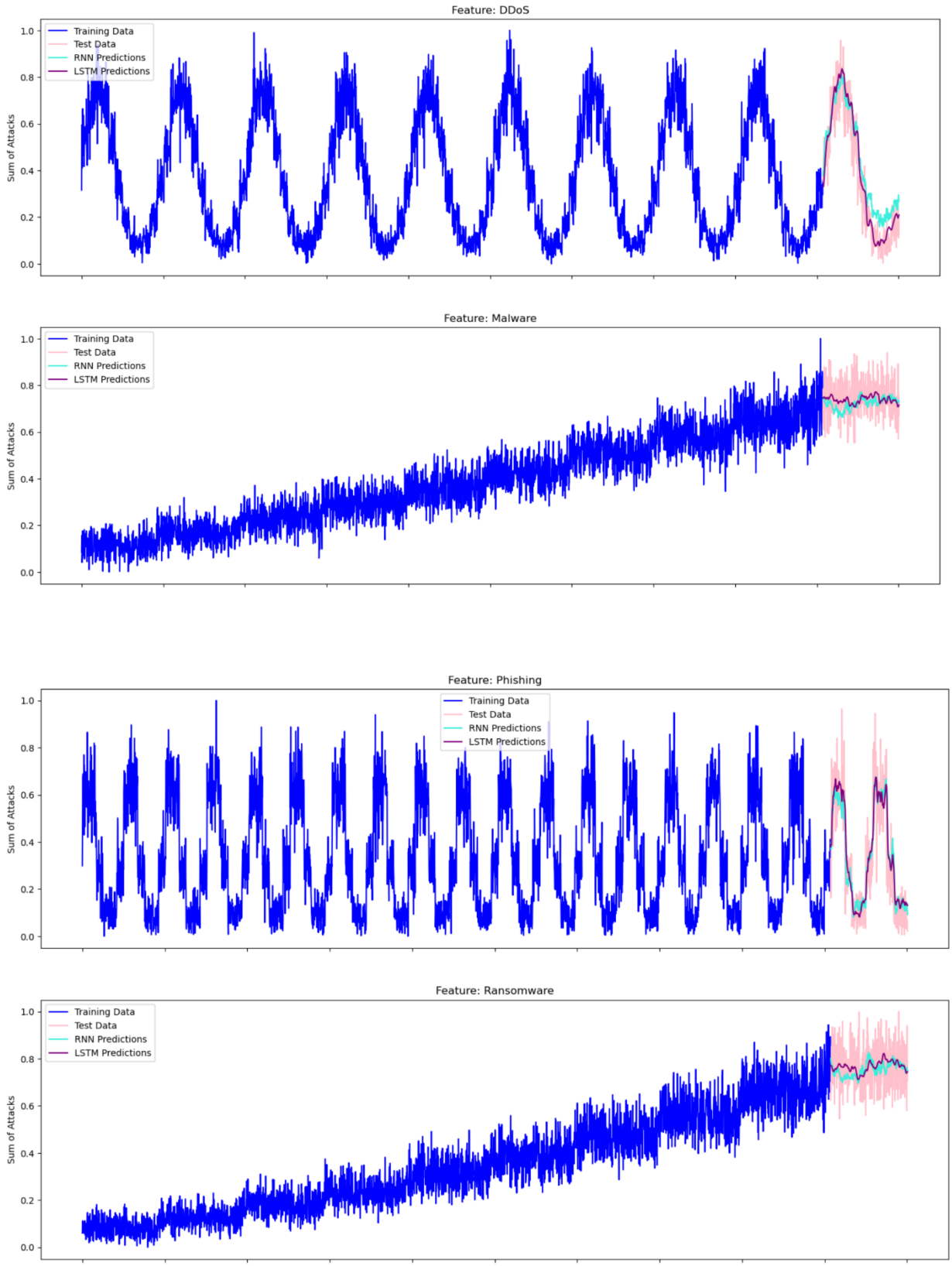
Liite 3. URLien tunnistaminen satunnaismetsämallilla

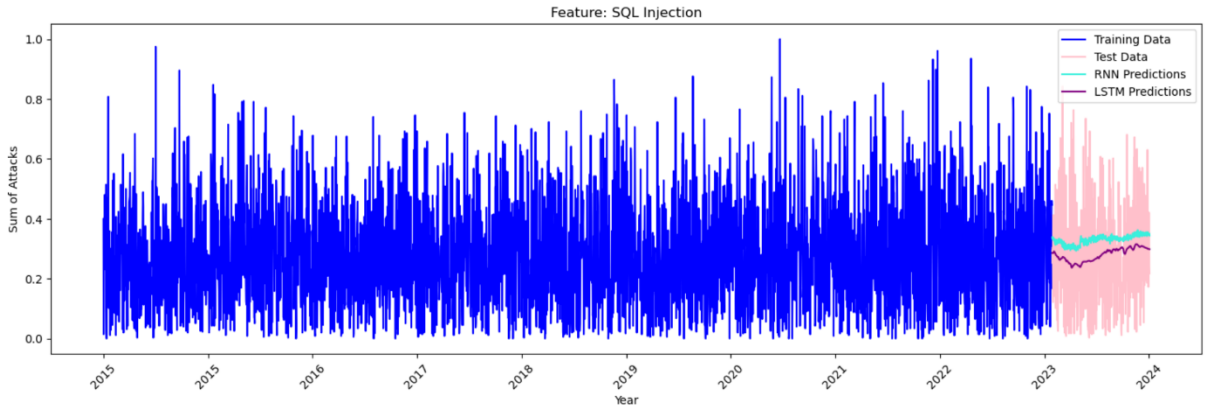
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, roc_curve, auc, roc_auc_score, confusion_matrix
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

data = pd.read_csv('urls.csv', sep=';')
df = pd.DataFrame(data)

# Find all null values and delete them
df = df.dropna()
df = df.sample(frac=1, random_state=42).reset_index(drop=True)

## The original data has over 90 000 rows so model was tested by 1 000 and 30 000 rows.
rows = 1000
df_final = pd.DataFrame(df.head(rows))

y = df_final['label']
X = df_final['domain']
X = X.astype(str)

# Convert URLs to feature vectors using TD-IDF
vectorizer = TfidfVectorizer(analyzer='char', ngram_range=(3,5))
X_vectorized = vectorizer.fit_transform(X)

# Train-test split
size = 0.15 # Result of test_best_test_size-function for 1000 rows
X_train, X_test, y_train, y_test = train_test_split(X_vectorized, y, test_size=size, random_state=42)

# Train Random Forest
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Evaluate
y_pred = model.predict(X_test)
print(f'Classification Report for {rows} rows: \n')
print(f'{classification_report(y_test, y_pred)} \n')
print(f'Confusion Matrix for {rows} rows: \n')
print(f'{confusion_matrix(y_test, y_pred)} \n')

# ROC curve with prediction of probabilities
y_pred_probs = model.predict_proba(X_test)[:,1]
fpr, tpr, thresholds = roc_curve(y_test, y_pred_probs)
roc_auc = roc_auc_score(y_test, y_pred_probs)

# Visualization for ROC curve
plt.figure()
plt.plot(fpr, tpr, color='blue', lw=2, label=f'ROC curve (AUC = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], color='pink', lw=2, linestyle='--', label='Random classifier')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve in RandomForestClassifier')
plt.legend(loc='lower right')
plt.grid()
plt.show()
```

Liite 4. URLien tunnistaminen LSTM-mallilla

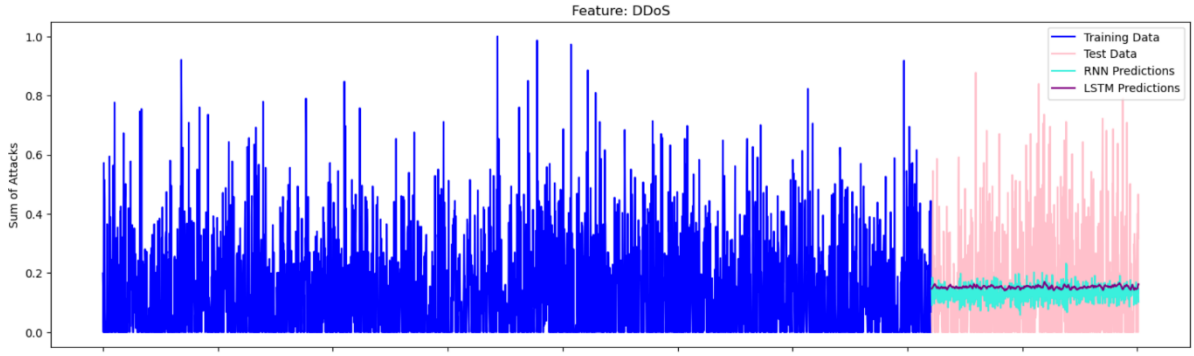
```
# Split data to features (urls) and labels (y)
urls = df['domain']
y = df['label']

# Tokenize URLs
num_words = 10000
tokenizer = Tokenizer(char_level=True, num_words=num_words)
tokenizer.fit_on_texts(urls)
X = tokenizer.texts_to_sequences(urls)

# Pad sequences to the same length
max_len = 100
X = pad_sequences(X, maxlen=max_len, padding='post')

# Function for RandomSearch
def build_model(hp):
    num_words = 10000

    model = Sequential([
        Embedding(input_dim=num_words, output_dim=128),
        Bidirectional(LSTM(units=hp.Int('units', min_value=32, max_value=256, step=32), return_sequences=True)),
        Dropout(rate=hp.Float('dropout', min_value=0.1, max_value=0.5, step=0.1)),
        Bidirectional(LSTM(units=hp.Int('units_2', min_value=32, max_value=256, step=32), return_sequences=True)),
        Dropout(rate=hp.Float('dropout', min_value=0.1, max_value=0.5, step=0.1)),
        Bidirectional(LSTM(units=hp.Int('units_3', min_value=32, max_value=256, step=32), return_sequences=True)),
        Dropout(rate=hp.Float('dropout', min_value=0.1, max_value=0.5, step=0.1)),
        Bidirectional(LSTM(units=hp.Int('units_4', min_value=32, max_value=256, step=32))),
        Dense(units=hp.Int('dense_units', min_value=32, max_value=128, step=32), activation='relu'),
        Dropout(rate=hp.Float('dropout', min_value=0.1, max_value=0.5, step=0.1)),
        Dense(1, activation='sigmoid')
    ])

    model.compile(optimizer=Adam(hp.Choice('learning_rate', [1e-3, 1e-4, 1e-5])), loss='mse', metrics=['accuracy'])

    return model

# Build RandomSearch
tuner = RandomSearch(
    build_model,
    objective='val_accuracy',
    max_trials=5,
    executions_per_trial=1,
    directory='lstm_tuning',
    project_name='lstm_hyperparam'
)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.15, random_state=42)

# Find best parameters
tuner.search(X_train, y_train, epochs=10, validation_data=(X_test, y_test), batch_size=1000)

# Get the best params
best_hps = tuner.get_best_hyperparameters(num_trials=1)[0]

# Build and train the best model
best_model = tuner.hypermodel.build(best_hps)
best_model.fit(X_train, y_train, epochs=10, validation_data=(X_test, y_test), batch_size=1000)

# Evaluate
_, accuracy = best_model.evaluate(X_test, y_test)
print("Accuracy:", accuracy)
```

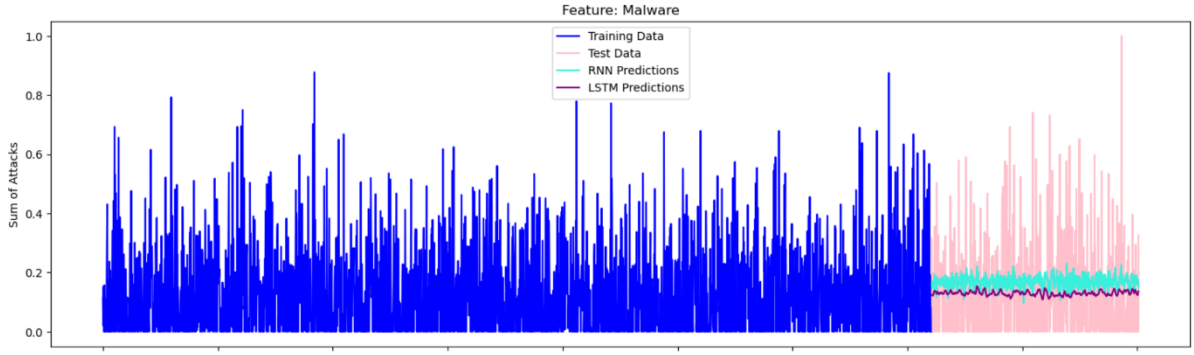
Liite 5. Jaksollisen datasetin tulokset

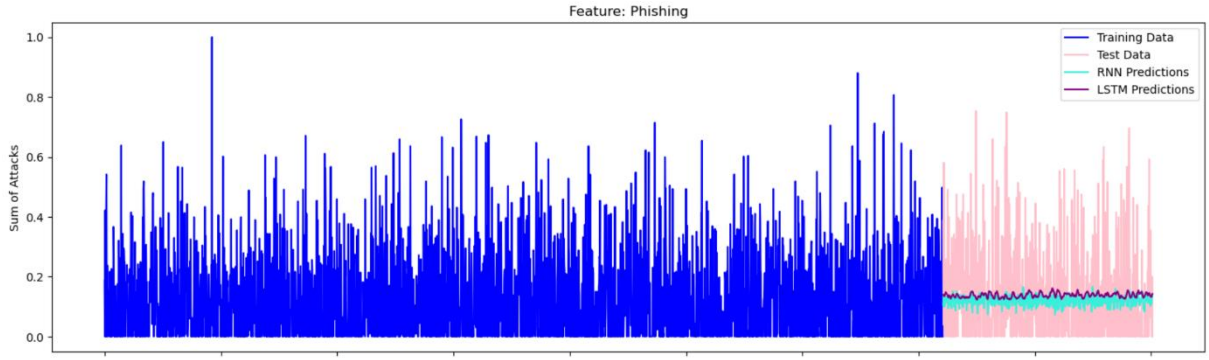


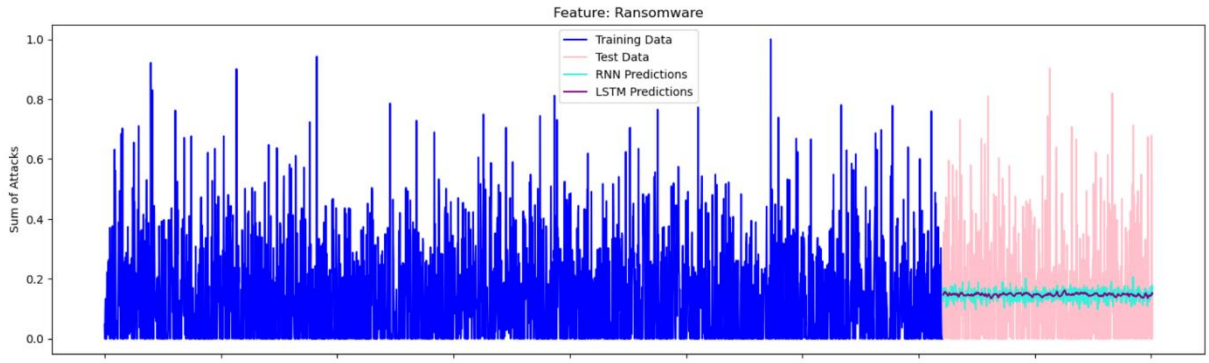

Feature: SQL Injection
Training Data
Test Data
RNN Predictions
LSTM Predictions
Sum of Attacks
1.0
0.8
0.6
0.4
0.2
0.0
2015
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
Year

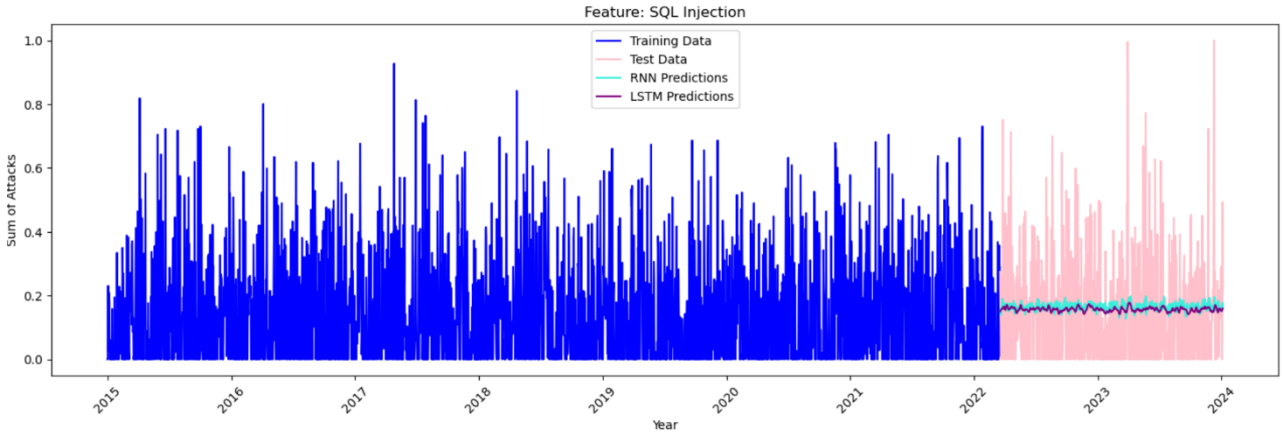
Liite 6. Jaksottoman datasetin tulokset










Feature: SQL Injection
Training Data
Test Data
RNN Predictions
LSTM Predictions
Sum of Attacks
1.0
0.8
0.6
0.4
0.2
0.0
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
Year