VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

Justas Dargis

ĮSIBROVIMO APTIKIMAS ANALIZUOJANT KOMPIUTERIO LYGMENS ŽURNALINIUS ĮRAŠUS IŠ AWSCTD DUOMENŲ RINKINIO

INTRUSION DETECTION BY ANALYZING COMPUTER-LEVEL LOG ENTRIES FROM THE AWSCTD DATASET

Baigiamasis magistro darbas

Informacijos ir informacinių technologijų saugos studijų programa, valstybinis kodas 6211BX014
Informatikos inžinerijos studijų kryptis

Vilnius, 2023

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

Justas Dargis

ĮSIBROVIMO APTIKIMAS ANALIZUOJANT KOMPIUTERIO LYGMENS ŽURNALINIUS ĮRAŠUS IŠ AWSCTD DUOMENŲ RINKINIO

INTRUSION DETECTION BY ANALYZING COMPUTER-LEVEL LOG ENTRIES FROM THE AWSCTD DATASET

Baigiamasis magistro darbas

Informacijos ir informacinių technologijų saugos studijų programa, valstybinis kodas 6211BX014
Informatikos inžinerijos studijų kryptis

Vadovas dr. Dainius Čeponis
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

Lietuvių kalbos konsultantas dr. Vaida Buivydienė
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

Vilnius, 2023

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

Studijų kryptis: Informatikos inžinerija

Studijų programa: Informacijos ir informacinių technologijų sauga , valstybinis kodas 6211BX014

Specializacija: Informacijos ir informacinių technologijų sauga

TVIRTINU
Katedros vedėjas
Nikolaj Goranin
2023-05-26

MAGISTRO BAIGIAMOJO DARBO UŽDUOTIS

Nr. ITSfm-21–3703

Vilnius

Studentas (-ė): Justas Dargis

Baigiamojo darbo tema: Įsibrovimo aptikimas analizuojant kompiuterio lygmens žurnalinius įrašus iš AWSCTD duomenų rinkinio

Baigiamojo darbo užbaigimo terminas pagal numatytą studijų kalendorinį grafiką.

BAIGIAMOJO DARBO UŽDUOTIS:

Tikslas: Pasiūlyti įsibrovimo aptikimo būdą kompiuterio lygmenyje, paremtą žurnalinių įrašų analize.

Uždaviniai:
1. Atlikti IDS ir jose naudojamų įsilaužimo aptikimo metodų apžvalgą.
2. Apžvelgti giliojo mokymosi, mašininio mokymosi ir kitus metodus, kurie yra naudojami žurnalinių įrašų analizei.
3. Papildyti AWSCTD duomenų rinkinį įvykdžius kenksmingą programinį kodą.
4. Atlikti duomenų rinkinio analizę pritaikant pasirinktus mašininio ar giliojo mokymosi metodus.
5. Pateikti atliktos analizės ir gautų rezultatų išvadas.

Planuojamas rezultatas: Literatūros analizė, AWSCTD duomenų rinkinio papildymas ir giliojo mokymosi metodo pritaikymas įsibrovimo aptikimui.

Vadovas Dainius Čeponis

Vilniaus Gedimino technikos universitetas
Fundamentinių mokslų fakultetas
Informacinių sistemų katedra

ISBN | ISSN
Egz. sk. .....
Data .....-.....-.....

Antrosios pakopos studijų **Informacijos ir informacinių technologijų saugos** programos magistro baigiamasis darbas

Pavadinimas	**Įsibrovimo aptikimas analizuojant kompiuterio lygmens žurnalinius įrašus iš AWSCTD duomenų rinkinio**
Autorius	**Justas Dargis**
Vadovas	**Dainius Čeponis**

Kalba: lietuvių

Anotacija

Magistro baigiamojo darbo tikslas – pasiūlyti įsibrovimo aptikimo būdą kompiuterio lygmenyje, paremtą žurnalinių įrašų analize. Atliekant literatūros analizę apžvelgiamos įsibrovimo aptikimo sistemos – jų tipai, naudojami metodai įsibrovimams aptikti. Detaliai analizuojamos kompiuterio lygmenyje veikiančios įsibrovimo aptikimo sistemos – nagrinėjama, kokias technikas bei kokius duomenis jos naudoja. Taip pat apžvelgiami įsibrovimo aptikimo tyrimuose, kompiuterio lygmenyje, naudojami duomenų rinkiniai ir metodai taikomi žurnalinių įrašų analizei. Tiriamojoje darbo dalyje apžvelgiamas AWSCTD duomenų rinkinys, aprašomi įžvelgti trūkumai bei apibūdinamas metodas taikomas duomenų rinkiniui papildyti. Taip pat aprašomas siūlomas metodas programų elgsenos duomenims atvaizduoti ir vektorizuoti. Galiausiai aprašomas atliktas bandymas aptinkant įsibrovimus ir apžvelgiami gauti rezultatai. Darbą sudaro 5 dalys: įvadas, susijusios literatūros analizė, siūlomo metodo aprašymas, išvados, literatūros sąrašas. Darbo apimtis – 83 p. teksto be priedų, 35 iliustr., 15 lent., 59 bibliografiniai šaltiniai.

Prasminiai žodžiai: Įsibrovimo aptikimas, IDS, žurnaliniai įrašai, žurnalinių įrašų analizė, natūralios kalbos apdorojimas, gilusis mokymasis, kenksmingas programinis kodas, AWSCTD

Vilnius Gediminas Technical University
Faculty of Fundamental Sciences
Department of Information Systems

ISBN | ISSN
Copies No. .....
Date .....-.....-.....

Master Degree Studies **Information and Information Technologies Security** study programme Master Graduation Thesis

Title	**Intrusion Detection by Analyzing Computer-level log Entries from the AWSCTD Dataset**
Author	**Justas Dargis**
Academic supervisor	**Dainius Čeponis**

Thesis language: Lithuanian

Annotation

The objective of the Master's thesis is to propose an intrusion detection method at the computer level based on the analysis of logs. Through a literature analysis, the paper gives an overview of intrusion detection systems - their types and methods used to detect intrusions. A detailed analysis of host-based intrusion detection systems is performed, where the techniques and type of data used were examined. It also covers the datasets used in host-based intrusion detection research and the methods used to analyze log records. The exploratory part of the thesis gives an overview of the AWSCTD dataset, describes the shortcomings identified, and describes the method used to augment the dataset. It also describes the proposed method for representing and vectorizing behavioral data of the executed programs. Finally, the intrusion detection experiment is described, and the results obtained are summarized. The thesis consists of 5 parts: introduction, analysis of related literature, description of the proposed method, conclusions, list of references. The volume of the thesis is 83 pages of text without appendices, 35 illustrations, 15 tables, 59 references.

Keywords: Intrusion detection, IDS, system logs, log analysis, natural language processing, deep learning, malware, AWSCTD

Turinys

1. Įvadas 10
2. Susijusios literatūros analizė 13
    2.1. Įsibrovimo aptikimo sistemos 13
        2.1.1. Tinklo lygmens įsibrovimo aptikimo sistemos 15
        2.1.2. Kompiuterio lygmens įsibrovimo aptikimo sistemos 17
    2.2. Įsibrovimo aptikimo metodų tipai 20
        2.2.1. Parašais paremtas įsibrovimo aptikimas 20
        2.2.2. Anomalijomis paremtas įsibrovimo aptikimas 21
    2.3. Įsibrovimo aptikimo technikos 23
        2.3.1. Statistiniais metodais paremta 23
        2.3.2. Mašininiu mokymu paremta 24
        2.3.3. Giliuoju mokymu paremta 25
    2.4. HIDS duomenų rinkiniai 26
    2.5. Žurnalinių įrašų analizės metodai 30
        2.5.1. Žurnalinių įrašų struktūrizavimas 31
        2.5.2. Žurnalinių įrašų vektorizavimas 33
        2.5.3. Anomalijų aptikimas 38
    2.6. Skyriaus apibendrinimas 44
3. Siūlomo metodo aprašymas 46
    3.1. Duomenys 46
        3.1.1. AWSCTD duomenų rinkinys 46
        3.1.2. OSSEC HIDS 48
        3.1.3. AWSCTD OSSEC žurnaliniai įrašai 50
    3.2. Duomenų rinkinio papildymas 52
    3.3. Duomenų paruošimas 57
    3.4. Duomenų vektorizavimas 59
    3.5. Įsibrovimų aptikimų vertinimas 64
        3.5.1. Įsibrovimo aptikimas analizuojant pradinius AWSCTD HIDS duomenis 72
    3.6. Skyriaus apibendrinimas 74

Išvados 76

Literatūros sąrašas 78

Paveikslų sąrašas

2.1 pav. Įsibrovimo aptikimų sistemų klasifikavimas 14
2.2 pav. NIDS architektūros schema (Elrawy ir kt., 2018) 15
2.3 pav. HIDS architektūros schema (Elrawy ir kt., 2018) 17
2.4 pav. Parašais paremtos įsibrovimo aptikimo sistemos veikimo schema (Khan ir kt., 2020) 21
2.5 pav. Anomalijomis paremtos įsibrovimo aptikimo sistemos veikimo schema (Khan ir kt., 2020) 22
2.6 pav. Giliuoju mokymu paremtos IDS veikimo schema (Aldweesh ir kt., 2020) 26
2.7 pav. Žurnalinių įrašų analizės pagrindiniai etapai (Wang ir kt., 2020) 31
2.8 pav. Žurnalinio įrašo pavertimas į struktūrizuotą šabloną (Zhu ir kt., 2018) 32
2.9 pav. Spell metodo veikimas (Du & Li, 2016) 33
2.10 pav. CBOW ir skip-gram modeliai (Mikolov, Chen, ir kt., 2013) 36
2.11 pav. LogRobust semantinis vektorizavimas (Zhang ir kt., 2019) 38
2.12 pav. LSTM veikimas (Olah, 2015) 41
2.13 pav. LogRobust anomalijų aptikimo modelis (Zhang ir kt., 2019) 43
3.1 pav. KPK vykdymo ir duomenų rinkimo komponentų schema (Čeponis & Goranin, 2018) 47
3.2 pav. Dažniausios kenkėjiškų programų kategorijos ir šeimos AWSCTD duomenų rinkinyje (Čeponis & Goranin, 2018) 47
3.3 pav. Skirtingi įvykiai ir jų pasiskirstymas duomenų rinkinyje 51
3.4 pav. Kenkėjiškų bandinių AWSCTD OSSEC dažniausi pranešimai su kintamosiomis žurnalo dalimis 51
3.5 pav. Švarių bandinių AWSCTD OSSEC dažniausi pranešimai su kintamosiomis žurnalo dalimis 52
3.6 pav. Programos bandinio duomenys papildytame duomenų rinkinyje 55
3.7 pav. Programos bandinio duomenys originaliame duomenų rinkinyje 55
3.8 pav. Dažniausiai pasikartojantys įvykiai papildytame duomenų rinkinyje 56
3.9 pav. HIDS pranešimų atvaizdavimas dokumente 59
3.10 pav. Dokumento vektorizavimo pavyzdys 64
3.11 pav. MLP modelio architektūra 69
3.12 pav. CNN modelio architektūra 69
3.13 pav. SVM 71
3.14 pav. RF 71
3.15 pav. MLP 71
3.16 pav. CNN 71
3.17 pav. SVM 72
3.18 pav. RF 72
3.19 pav. MLP 72
3.20 pav. CNN 72
3.21 pav. CNN painiavos matrica. 0 – nekenksmingi, 1 – kenksmingi 73
3.22 pav. CNN painiavos matrica. 0 – AdWare, 1 – Trojan, 2 – Downloader, 3 – WebToolbar, 4 - Benign 73

Lentelių sąrašas

2.1 lentelė. NIDS ir HIDS palyginimas 19
2.2 lentelė. Įsibrovimo aptikimo metodų tipų palyginimas (Liu & Lang, 2019) 22
2.3 lentelė. Atakų tipai naudoti ADFA-LD duomenų rinkiniui generuoti (Borisaniya & Patel, 2015) 27
2.4 lentelė Atakos naudotos ADFA-WD duomenų rinkinio duomenims sugeneruoti (Borisaniya & Patel, 2015) 28

2.5 lentelė. NGIDS duomenų rinkinio atributų aprašymas 29
2.6 lentelė. Kompiuterio lygmens įsibrovimo aptikimo duomenų rinkinių palyginimas 29
3.1 lentelė. OSSEC įspėjamųjų pranešimų pavyzdžiai 49
3.2 lentelė. Bandinių ir įvykių statistika 50
3.3 lentelė. Papildyto duomenų rinkinio bandinių ir įvykių statistika 56
3.4 lentelė. KPK tipų pasiskirstymas duomenų rinkinyje 58
3.5 lentelė. NLP modelių privalumai ir trūkumai 62
3.6 lentelė. Bandyme naudojamas duomenų kiekis pagal klases 67
3.7 lentelė. Klasifikavimo modelių hiperparametrai 67
3.8 lentelė. Klasifikavimo rezultatai 70
3.9 lentelė. Klasifikavimo rezultatai su originaliu duomenų rinkiniu 73

Santrumpos

IDS (angl. *intrusion detection system*) – įsibrovimo aptikimo sistema.

HIDS (angl. *host-based intrusion detection system*) – įsibrovimo aptikimo sistema veikianti kompiuterio lygmenyje.

NIDS (angl. *network-based intrusion detection system*) – įsibrovimo aptikimo sistema veikianti tinklo lygmenyje.

OS (angl. *operating system*) – operacinė sistema.

VM (angl. *virtual machine*) – virtuali mašina.

SIDS (angl. *signature-based intrusion detection system*) – parašais paremta įsibrovimo aptikimo sistema.

AIDS (angl. *anomaly-based intrusion detecetion system*) – anomalijomis paremta įsibrovimo aptikimo sistema.

ML (angl. *machine learning*) – mašininis mokymasis.

DL (angl. *deep learning*) – gilusis mokymasis.

NLP (angl. *Natural Language Processing*) – natūralios kalbos apdorojimas.

NN (angl. *Neural Network*) – neuroninis tinklas.

CNN (angl. *Convolutional Neural Network*) – konvoliucinis neuroninis tinklas.

LSTM (angl. *Long Short-Term Memory*) – ilgalaikė trumpalaikė atmintis.

KPK (angl. *Malware*) – kenksmingas programinis kodas.

API (angl. *application programming interface*) – aplikacijų programavimo sąsaja.

MLP (angl. *Multilayer perceptron)* – daugiasluoksnis perceptronas.

RF (angl. *Random forest*) – atsitiktinių miškų algoritmas.

SVM (angl. *Support vector machine*) – atraminių vektorių mašinos algoritmas.

RNN (angl. *Recurrent neural network*) – rekurentinis neuroninis tinklas.

AWSCTD (angl. *Attack-Caused Windows OS System Calls Traces Dataset*) – atakų sukeltų Windows sisteminių funkcijų kvietimų duomenų rinkinys.

TF-IDF (angl. *Term Frequency-Inverse Document Frequency*) – terminų dažnis – atvirkštinis dokumento dažnis.

BERT (angl. *Bidirectional Encoder Representations from Transformers*) – dvikryptis transformacinio kodavimo būdas.

JSON (angl. *JavaScript Object Notation*) – JavaScript objektų žymėjimas

1. Įvadas

Internetas ir kompiuteriai yra mūsų kasdieninio gyvenimo dalis ir nepakeičiami įrankiai. Šiuos įrankius naudojame daugelyje gyvenimo sričių – švietime, darbinėje aplinkoje ar pramogoms. Sparčiai besivystant technologijoms, technologijos atneša ne tik didesnę naudą žmonėms, bet taip pat atsiranda ir naujų kibernetinių atakų, kurios padaro žalą žmonėms ar organizacijoms. 2020 m., kurie yra laikomi pandemijos metais, kai daug žmonių dirbo nuotoliniu būdu, kibernetinių incidentų skaičius Lietuvoje išaugo ketvirtadaliu lyginant su ankstesniais metais. Kibernetinių incidentų augimo tendencija yra stebima kiekvienais metais ir, tikėtina, kad incidentų ateityje tik daugės (Krašto apsaugos ministerija, 2021).

Apsisaugoti nuo kibernetinių atakų organizacijose yra naudojama įvairi techninė ar programinė įranga, pavyzdžiui ugniasienės, antivirusų programinė įranga ar įsibrovimo aptikimo sistemos (IDS). Ugniasienė yra vienas iš naudojamų mechanizmų, tačiau ji ne visais atvejais gali apsaugoti tinklo nuo atakų, nes aptinka jas tik už tinklo ribų. Pastaraisiais metais, atakų nukreiptų į tinklą ar kompiuterius skaičius sparčiai išaugo. Todėl mokslininkų susidomėjimas IDS, kas yra alternatyvi saugumo priemonė, išaugo (Ethala ir kt., 2013). IDS gali aptikti įsibrovimus tiek tinkle, tiek kompiuteryje. IDS yra skirstomos į 2 pagrindinius tipus, pagal jų veikimo vietą – tinkle veikiančios (NIDS) ir kompiuteryje veikiančios (HIDS). NIDS analizuoja tinklo duomenis ir gali vienu metu stebėti daug kompiuterių bei nustatyti išorines grėsmes, prieš joms išplintant tinkle. Tuo tarpu HIDS analizuoja duomenis kompiuterio lygmenyje. Duomenų šaltinis tokiose sistemose dažnai yra sistemos žurnaliniai įrašai, sisteminių funkcijų kvietimų įrašai, „Windows“ registrų pakeitimai ar failų pasikeitimai failų sistemoje (Ozkan-Okay ir kt., 2021). Žurnaliniai įrašai yra puikus duomenų šaltinis iš kurio galima identifikuoti anomalijas, tačiau, kad panaudoti šiuos duomenis yra reikalinga juos tinkamai apdoroti, dėl ko žurnalinių įrašų analizė nėra dažnai taikomas metodas anomalijoms aptikti HIDS. Todėl yra reikalinga ištirti žurnalinių įrašų analizės metodus, kurie būtų praktiški ir juos būtų galima panaudoti aptinkant įsibrovimus kompiuteriuose naudojant HIDS.

Tyrimo objektas – įsilaužimų aptikimo metodai kompiuterio lygmenyje.

Darbo tikslas – pasiūlyti įsibrovimo aptikimo būdą kompiuterio lygmenyje, paremtą žurnalinių įrašų analize.

Darbo uždaviniai:

1. Atlikti IDS ir jose naudojamų įsilaužimo aptikimo metodų apžvalgą.

2. Apžvelgti giliojo mokymosi, mašininio mokymosi ir kitus metodus, kurie yra naudojami žurnalinių įrašų analizei.
3. Papildyti AWSCTD duomenų rinkinį vykdant kenkėjiškų programų bandinius.
4. Atlikti duomenų rinkinio analizę pritaikant pasirinktus mašininio ar giliojo mokymosi metodus.
5. Pateikti atliktos analizės ir gautų rezultatų išvadas.

Temos naujumas – įsibrovimo aptikimas, paremtas tinklo duomenimis, yra gana aktyviai nagrinėjamas ir šiai užduočiai atlikti yra pakankamas kiekis kokybiškų, laisvai prieinamų duomenų rinkinių. Kompiuterio lygmens duomenų, įsibrovimams aptikti, kurie būtų kokybiški ir laisvai prieinami nėra tiek daug. Taip pat dauguma esamų HIDS duomenų rinkinių yra orientuoti į „Linux“ OS ir turi mažai „Windows“ OS įrašų (Čeponis, 2021). AWSCTD duomenų rinkinys yra sudarytas iš sužymėtų sisteminių funkcijų kvietimų sekų, tinklo srauto duomenų, žurnalinių įrašų apie pasikeitimus „Windows“ registruose bei modifikuotus failu, kai kompiuteryje buvo paleistas kenkėjiškas programinis kodas. Šio duomenų rinkinio sisteminių funkcijų kvietimų sekos jau buvo analizuojamos ankstesniuose tyrimuose, tačiau žurnaliniai įrašai dar nebuvo analizuojami, todėl galima teigti, kad tema yra nauja.

Temos aktualumas – Kibernetinių incidentų skaičius nuolat auga, todėl yra būtina naudoti kuo pažangesnes įsibrovimo aptikimo sistemas, kurios gebėtų tiksliai ir greitai nustatyti įsibrovimus. Įprasti, parašais paremti, įsibrovimo aptikimo metodai negeba aptikti naujų atakų, todėl yra reikalinga taikyti anomalijomis pagrįstus įsibrovimo aptikimo metodus. Pastaruoju metu, šia tema, yra atliekama daug tyrimų, kurie pasiekia neblogų rezultatų, tačiau didžioji jų dalis yra orientuota į įsibrovimo aptikimą tinkle arba įsibrovimo aptikimą kompiuterio lygmenyje analizuojant sisteminių funkcijų kvietimus. Žurnalinių įrašų analizė įsibrovimams aptikti nėra taip plačiai nagrinėjama, bet šis duomenų šaltinis taip pat gali būti puiki priemonė identifikuojant įsibrovimus ir būti plačiai naudojamas kompiuterio lygmens įsibrovimo aptikimo sistemose.

Tyrimo metodika – Analitinėje darbo dalyje, atliekant įsilaužimo aptikimo sistemų, bei žurnalinių įrašų analizės metodų analizę, naudojami susijusių mokslinių straipsnių literatūros analizės ir lyginamosios analizės metodai. Tiriamojoje dalyje atliekamas empirinis tyrimas.

Mokslinė darbo vertė – papildytas duomenų rinkinys, pasiūlytas ir išbandytas naujas metodas įsilaužimui aptikti analizuojant kompiuterio lygmens žurnalinius įrašus.

Darbo rezultatai:

1. Atlikta literatūros šaltinių, susijusių su IDS veikimo metodais, bei žurnalinių įrašų analizės metodais, analizė.
2. AWSCTD HIDS žurnalinių įrašų rinkinys papildytas pakartotinai vykdant kenkėjiškas ir nekenkėjiškas programas ir renkant elgsenos duomenis su patobulinta HIDS konfigūracija.
3. Pasiūlytas žurnalinių įrašų atvaizdavimo būdas taikant NLP metodus, bei kenkėjiškų failų klasifikavimo metodas taikant ML ir DL modelius.
4. Atlikti įsibrovimo aptikimo ir kenkėjiškų programų tipo nustatymo bandymai, pateikti bandymų rezultatai.

Darbo struktūra:

1. Pirmame skyriuje apibūdinama nagrinėjama tema bei jos aktualumas. Taip pat aprašomi darbo tikslai ir uždaviniai, tyrimo objektas.
2. Antrame skyriuje analizuojamos įsibrovimo aptikimo sistemos – jų tipai, IDS naudojami metodai įsibrovimams aptikti. Taip pat gilinamasi į kompiuteryje veikiančias įsibrovimo aptikimo sistemas – analizuojama kokias technikas jos naudoja, bei kokius duomenis naudoja anomalijoms aptikti. Šiame skyriuje dar apžvelgiami duomenų rinkiniai naudojami kompiuterio lygmens įsibrovimams aptikti, bei metodai taikomi analizuojant žurnalinius įrašus.
3. Trečiame skyriuje apžvelgiamas AWSCTD duomenų rinkinys, bei aprašomas metodas taikomas duomenų rinkiniui papildyti. Taip pat apibūdinamas siūlomas metodas duomenims atvaizduoti ir vektorizuoti. Galiausiai aprašoma įsibrovimo aptikimo tyrimo eiga, bei apžvelgiami gauti rezultatai.

Darbo aprobacija:

Svarbiausieji darbo rezultatai pagal magistro tezių tematiką pateikti pranešime, skaitytame 2023 m. balandžio 9 d., Vilniaus Gedimino technikos universitete vykusioje 26-ojoje jaunųjų mokslininkų konferencijoje „Informacinių technologijų sauga ir informacinės sistemos“.

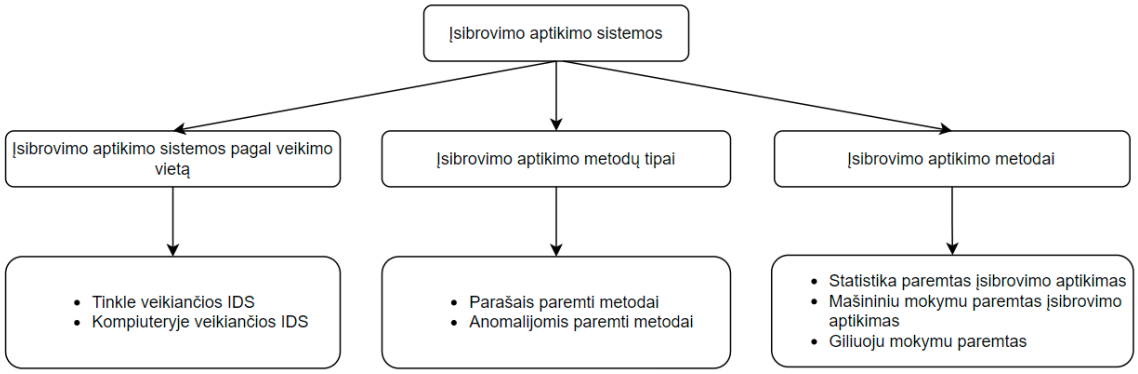
2. Susijusios literatūros analizė

Šiame skyriuje apžvelgiama su sprendžiama problema susijusi literatūra – apžvelgiamos skirtingos įsibrovimo aptikimo sistemos pagal jų veikimo vietą, nustatomi jų privalumai ir trūkumai. Taip pat apžvelgiamos metodologijos ir technikos, kurios yra naudojamos įsibrovimo aptikimo sistemose ir nustatomi jų privalumai ir trūkumai. Galiausiai yra apžvelgiami duomenų rinkiniai, kurie gali būti naudojami kompiuterio lygmens įsibrovimo aptikimo problemoms spręsti, bei apžvelgiami kitų tyrėjų anksčiau atlikti tyrimai, kuriuose buvo analizuojami kompiuterio lygmens žurnaliniai įrašai.

2.1. Įsibrovimo aptikimo sistemos

Įsibrovimų aptikimas – tai tinklo ar kompiuterio sistemoje vykstančių įvykių stebėjimas ir jų analizė, siekiant įspėti apie galimus pavojingus įvykius, tokius kaip atakas. IDS pagrindinė funkcija – aptikti įvykius, kurie galimai yra kenksmingi, apie juos surinkti informaciją ir apie tai pranešti saugos administratoriams. Taip pat IDS dar naudojamos kitiems tikslams, pavyzdžiui, aptikti saugumo politikos pažeidimus, pranešti apie esamas grėsmes ir atgrasyti asmenis nuo kibernetinių atakų.

Siekiant veiksmingo ir efektyvaus įsibrovimo aptikimo, IDS komponentai turi būti tinkamai apsaugoti. IDS sudaro įvairūs komponentai, įskaitant vartotojus, jutiklius, duomenų bazių serverius, valdymo serverius, tinklus ir kt.. Apsaugoti IDS komponentus yra labai svarbu, nes į juos nusitaiko užpuolikai, norintys sustabdyti IDS, kad įsibrovimas nebūtų aptiktas. Visų komponentų operacinėse sistemose ir programinėje įrangoje turi būti nuolat diegiami naujausi atnaujinimai, taip pat visi IDS programinės įrangos komponentai turi būti tinkamai apsaugoti nuo grėsmių. Dar būtų naudinga naudoti kelias skirtingas IDS technologijas, kad sistema būtų pilnavertiškesnė ir būtų pasiekiamas aukštesnis atakų aptikimo rodiklis. Egzistuoja įvairių IDS rūšių, pagrindinės dvi pagal veikimo vietą – veikiančios tinkle arba kompiuteryje. Kiekviena iš jų siūlo iš esmės skirtingas informacijos rinkimo, įrašymo, aptikimo ir prevencijos galimybes. Kompiuteryje veikiančio ir tinkle veikiančios IDS gali būti naudojamos ir kartu, taip būtų pasiekiamas didesnis įsibrovimo aptikimų efektyvumas. Renkantis IDS sprendimus, reikėtų atsižvelgti į skirtingas kiekvienos technologijos savybes ir privalumus. Dažniausiai pasitaikančios įsibrovimų aptikimo sistemų technologijos ir jose naudojami metodai pavaizduoti 2.1 pav..



2.1 pav. Įsibrovimo aptikimų sistemų klasifikavimas

Yra keletas svarbių sąlygų į kurias reikėtų atsižvelgti renkantis įsibrovimo aptikimo sistemas:

- turi būti minimalus klaidingai teigiamų aptikimų skaičius;
- aptikimo greitis;
- programinės ar techninės įrangos resursų naudojimas;
- gebėjimas tiksliai nustatyti įsibrovimo vietą;
- gebėjimas dirbti kartu su kita programine ar technine įranga;

Apibendrinant, galima teigti, kad IDS turi atlikti aukščiau paminėtas funkcijas, kad įsibrovimo aptikimas būtų efektyvus ir kad būtų galima tiksliai ir laiku aptikti atakas.

Bendrosios IDS funkcijos

Egzistuoja daug skirtingų IDS technologijų pagal atakų tipus, kuriuos jos gali atpažinti, ir metodą, kurį jos naudoja atakoms aptikti. Visų tipų IDS turi šias funkcijas:

- informacijos rinkimas – IDS renkama informacija, pavyzdžiui žurnaliniai įrašai ar informacija apie tinklo paketus, dažniausiai yra saugoma IDS įrenginyje, su tikslu duomenis panaudoti palyginimui ar sudaryti tam tikrus naudojimo profilius. Taip pat surinkti duomenys dažnai būna siunčiami į centrinį duomenų rinkimo serverį ar kitas saugumo valdymo sistemas.
- svarbių įvykiu aptikimas – yra būtina greitai ir tiksliai nustatyti įvykius, kurie vartotojui atrodo normalūs, tačiau pagal surinktą informaciją yra panašūs į anomalius.
- informavimas apie identifikuotus svarbius įvykius – įspėjamieji pranešimai yra perduodami įvairiais informacijos perdavimo būdais, pavyzdžiui el. paštu ar pranešimu saugumo valdymo sistemoje. Įspėjamuosiuose pranešimuose yra pagrindinė informacija apie įtartinus įvykius.

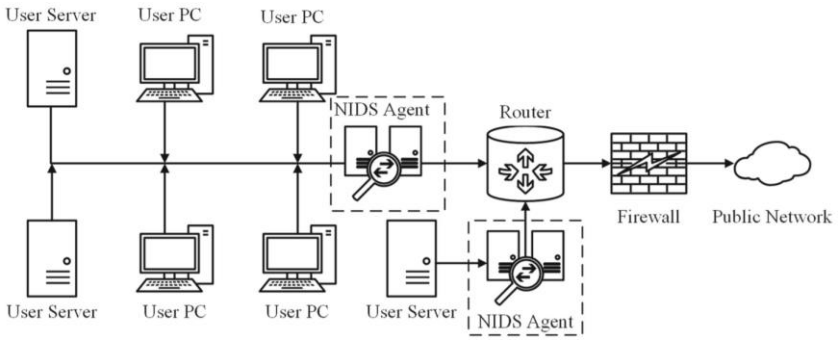
- ataskaitų generavimas – sugeneruotose ataskaitose apibendrinami pastebėti įvykiai arba pateikiama išsami informacija apie svarbius įvykius.

Pagrindinis bendras visų tipų IDS bruožas yra tas, kad jie negali užtikrinti visiškai tikslaus įsibrovimo aptikimo. Kai IDS įprastą veiklą nustato kaip anomaliją, tai yra vadinama klaidingai teigiamu (angl. *false positive*) aptikimu. Klaidingas neigiamas (angl. *false negative*) rezultatas gaunamas, kai kenkėjiška veikla nebuvo aptikta, bet iš tikro ji buvo. Neįmanoma visiškai pašalinti visų klaidingų teigiamų ir neigiamų rodiklių. Daugeliu atvejų sumažinus vieną, kitas rodiklis didėja. Daugelis IDS kūrėjų pasirenka sumažinti klaidingai neigiamą rodiklį, net jei klaidingai teigiamų rezultatų rodiklis didėja (Ozkan-Okay ir kt., 2021).

Tolimesniuose skyriuose yra giliau analizuojami 2 pagrindiniai įsibrovimo aptikimų sistemų tipai – tinklo lygmens ir kompiuterio lygmens.

2.1.1. Tinklo lygmens įsibrovimo aptikimo sistemos

Tinklo lygmens IDS (NIDS), stebi tinklo srautą ir analizuoja naudojamus protokolus, įtartinai veiklai aptikti. NIDS gali būti kaip kompiuterinis įrenginys arba kaip programinė įranga (Elrawy ir kt., 2018). Vienas iš programinės įrangos NIDS pavyzdžių yra „Snort“. Kadangi tinklo lygmens IDS veikia tinkle, ji gali stebėti visus įrenginius tinkle vienu metu. Įprasta NIDS architektūros schema pavaizduota 2.2 pav..



2.2 pav. NIDS architektūros schema (Elrawy ir kt., 2018)

NIDS didžioji duomenų analizės dalis paprastai atliekama aplikacijų lygmenyje, pagal TCP/IP modelį. Kai kurios NIDS, atlieka analizę ir fiziniame lygmenyje. Įprastą NIDS sudaro šie komponentai – jutikliai, valdymo serveriai, duomenų bazės serveris. Visi išvardinti komponentai, išskyrus jutiklius, yra naudojami ir kitose IDS technologijose. Tinklo lygmens įsibrovimo aptikimo sistemoje, jutikliai stebi ir analizuoja arba perduoda į centrinį serverį tinklo srautą.

Tinklo lygmens įsibrovimo aptikimo sistemos atlieka įvairias saugumą pagerinančias funkcijas. Pagrindinės funkcijos yra suskirstytos į tris kategorijas – informacijos rinkimas, žurnalizavimas, įvykiu aptikimas (Ozkan-Okay ir kt., 2021).

1. Informacijos rinkimas – NIDS turi ribotas galimybes rinkti duomenis. Dažniausiai renkami duomenys yra tinklo srauto paketai. Iš surinktų duomenų NIDS gali nustatyti tam tikrą informaciją:
    - įrenginių identifikavimas – NIDS gali sudaryti tinkle esančių įrenginių sąrašą;
    - operacinės sistemos identifikavimas – NIDS gali nustatyti kompiuterių operacines sistemas ir jų versijas. Žinoti naudojamos OS versiją yra naudinga nustatant pažeidžiamus kompiuterius;
    - aplikacijų identifikavimas – NIDS jutikliai gali identifikuoti veikiančių aplikacijų versijas stebint aplikacijos naudojamus prievadus ir programos komunikaciją tinkle. Ši informacija yra panaudojama nustatyti galimai pažeidžiamas aplikacijas ir jų neteisėtą panaudojimą;
    - tinklo charakteristikų nustatymas – renkama bendra informacija apie kai kuriuos NIDS jutiklius, tinklo konfigūraciją ir srautą. Šios informacijos dėka lengvai aptinkami bet kokie tinklo konfigūracijos pakeitimai.
2. Žurnalizavimas – NIDS saugo išsamius duomenis apie aptiktus įvykius. Šie duomenys naudojami įspėjimams patvirtinti, įvykiams tirti ir įvykių koreliacijai nustatyti. Duomenys apie įvykius, kurie dažniausiai registruojami į žurnalą yra šie:
    - data ir laikas;
    - prisijungimų skaičius;
    - įvykio tipas;
    - naudojamas protokolas;
    - IP adresai iš kurio, į kurį yra kreipiamasi;
    - siunčiamų paketų skaičius;
    - aplikacijų užklausos ir atsakymai.
3. Įvykiu aptikimas – IDS veikiančios tinkle naudoja parašais arba anomalijomis paremtus įsibrovimo aptikimo metodus. Parašais paremtų metodų atveju, surinkti tinklo duomenys yra lyginami su turimais parašais saugomais duomenų bazėje. Anomalijomis paremtų metodų atveju – yra įsisavinamas įprastas elgesys tinkle, o užfiksavus nukrypimus nuo įprasto elgesio, tai interpretuojama kaip įsibrovimas.

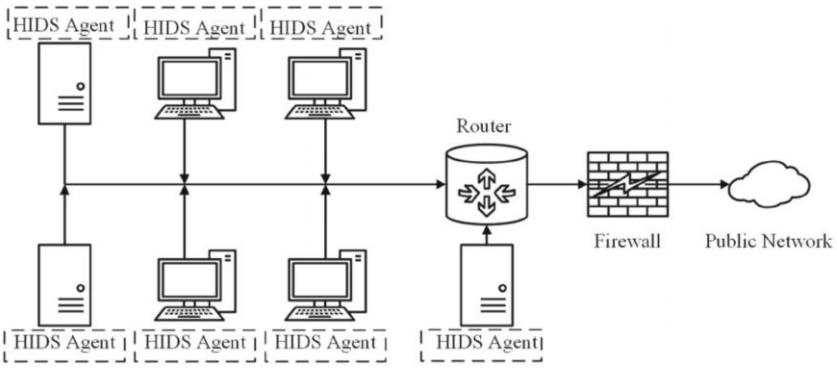
Tinklo lygmens įsibrovimo aptikimo sistemos pasižymi aukštu klaidingai teigiamų ir klaidingai neigiamu rezultatų rodikliu (Ozkan-Okay ir kt., 2021). Dauguma tradicinių NIDS naudoja parašais paremtus aptikimo metodus, kad aptiktų žinomas, paprastas atakas. Naujos kartos sistemos naudoja kelių metodų kombinaciją, tai padeda pasiekti aukštesnį tikslumą ir aptikti daugiau skirtingų atakų tipų. Taip pat tokiose sistemose klaidingų teigiamų ir neigiamų rodikliai yra mažesni. Nors NIDS gali neblogai aptikti atakas, tačiau jos taip pat turi ir trūkumų. Pagrindiniai jos trūkumai yra, kad NIDS negali analizuoti šifruoto srauto ir aptikti atakų jame, bei šioms sistemoms yra sunku veikti kai yra didelės tinklo apkrovos.

2.1.2. Kompiuterio lygmens įsibrovimo aptikimo sistemos

Kompiuterio lygmens IDS (HIDS) stebi kompiuterio parametrus ir veiklą, kad nustatytų galimas grėsmes. HIDS stebi tokius duomenis, kaip tinklo srauto informacija, žurnaliniai įrašai ar failų vientisumas (Deshpande ir kt., 2018). Dauguma HIDS kompiuteriuose turi įdiegtą tam tikrą programinę įrangą, vadinamą agentais. Kiekvienas agentas stebi veiklą viename kompiuteryje. Agentai persiunčia duomenis į valdymo serverius, kurie duomenis saugo atskiruose duomenų bazių serveriuose. Valdymo serveriai naudojami IDS valdymui ir stebėjimui. HIDS agentai gali veikti:

- serveriuose – agentas gali ne tik stebėti serverio operacinę sistemą, bet ir stebėti kai kurias programas;
- kompiuteriuose – agentai gali stebėti naudotojų kompiuterius. Jie dažnai stebi operacinę sistemą, įprastas programas, pvz., el. pašto programas, žiniatinklio naršykles ir pan.;
- programose – kai kurie agentai yra skirti tik konkrečių programų stebėjimui, pvz., žiniatinklio ar duomenų bazių serverio. Tokie agentai taip pat žinomi kaip aplikacijos lygmenyje veikiančios (angl. *application-based*) IDS.

Paprasta HIDS architektūros schema pavaizduota 2.3 pav..



2.3 pav. HIDS architektūros schema (Elrawy ir kt., 2018)

Kompiuterio lygmens įsibrovimo aptikimo sistemos pagrindinės funkcijos – žurnalizavimas ir įvykiu aptikimas.

1. Žurnalizavimas – HIDS renka išsamią informaciją apie aptiktus įvykius ir įrašinėja juos į žurnalą. Šie duomenys gali būti panaudojami įspėjimams įvertinti, įvykiams tirti ar nustatyti įvykių koreliacijas su įvykiais iš kitų šaltinių. Įprastai į žurnalą įrašomi šie duomenys:
    - įvykio data ir laikas;
    - įvykio ar įspėjimo tipas;
    - IP adresas;
    - prievado informacija;
    - aplikacijos informacija;
    - failų pavadinimai, failų keliai failų sistemoje, naudotojų informacija, kurie atliko veiksmus.
2. Įvykių aptikimas – dauguma HIDS turi galimybę aptikti kelių tipų kenkėjišką veiklą. Jose dažnai naudojami parašais pagrįsti aptikimo metodai, žinomoms atakoms aptikti. Taip pat yra naudojami politikomis ar taisyklių rinkiniais, ar anomalijomis pagrįsti metodai, kad būtų aptinkamos naujos, dar nematytos atakos.

Kaip ir kitos IDS technologijos, HIDS, taip pat turi aukštus klaidingai teigiamų ir klaidingai neigiamų aptikimų rodiklius. Įsibrovimo aptikimas kompiuterio lygmenyje yra sudėtingesnis, kadangi tam reikia ištirti daugiau skirtingų duomenų, pvz., išanalizuoti įvykių žurnalą, bei failų pasikeitimus ir juos susieti. Pavyzdžiui, kompiuteris gali būti perkraunamas ar į kompiuterį gali būti įdiegiama nauja programa, šie įvykiai, iš pirmos pažiūros neatrodo kenkėjiški, tačiau jie taip pat gali būti ir kenkėjiški. HIDS pačius įvykius aptinka tiksliai, tačiau be papildomos informacijos negalima nustatyti ar tai įprasti įvykiai ar tai yra ataka. HIDS, kurios naudoja kelių aptikimo metodų kombinaciją pasiekia tikslesnius rezultatus, nei tos kurios naudoja tik vieną metodą. Kadangi kiekvienas metodas gali stebėti kompiuterį skirtingais aspektais, todėl naudojant daugiau metodų galima surinkti daugiau informacijos apie kompiuteryje vykstančius įvykius. Tai suteikia galimybę sudaryti pilnesnius įvykių profilius ir tai gali palengvinti įvertinti ar veikla yra kenkėjiška ar ne (Ozkan-Okay ir kt., 2021).

2.1 lentelėje yra pateikiamas NIDS ir HIDS palyginimas. Palyginimui pasirinkti 6 kriterijai – naudojami duomenų šaltiniai, sistemos diegimo ypatumai, aptikimo efektyvumas, įsibrovimo atsekamumas, privalumai ir trūkumai.

2.1 lentelė. NIDS ir HIDS palyginimas

IDS tipas / Palyginimo kriterijus	**NIDS**	**HIDS**
Naudojami duomenys	Tinklo paketai, dažniausiai aplikacijų lygyje (pvz. HTTP, DNS, FTP, SNMP ir kt.).	Sistemos žurnaliniai įrašai, audito žurnalas, sisteminių funkcijų kvietimai, Windows registrų pakeitimų, failų pakeitimų žurnalas, sistemos apkrovos parametrai.
Diegimas	Paprasta įdiegti, diegiamas tik pagrindiniuose tinklo mazguose.	Sudėtinga įdiegti, kadangi reikia diegti į kiekvieną kompiuterį, yra priklausoma nuo įrenginio OS – kai kuriose OS gali neveikti tam tikri HIDS produktai.
Aptikimo efektyvumas	Aukštas aptikimo efektyvumas, gali aptikti atakas realiu laiku.	Žemesnis aptikimo efektyvumas, yra reikalinga išanalizuoti didelį kiekį duomenų iš skirtingų šaltinių.
Įsibrovimo atsekamumas	Įsibrovimas atsekamas pagal IP adresus, laiko žymas.	Įsibrovimo procesus galima atsekti pagal sisteminių funkcijų kvietimo kelius.
Privalumai	Aptinka atakas tinkle analizuojant tinklo paketus. Gali stebėti daug įrenginių vienu metu, nereikia atskirai konfigūruoti kiekvieno kompiuterio.	Gali aptikti atakas nors ir tinklo srautas yra šifruojamas. Aptinka įsibrovimus analizuojant kompiuterio žurnalus, tinklo įvykius, failų sistemą, sistemines funkcijas. Nėra reikalinga papildoma techninė įranga.
Trūkumai	Negali aptikti atakų šifruotame tinklo sraute. Sunkiai aptinka įsibrovimus, kai yra didelis tinklo srautas. Ribotai aptinka atakas – gali aptikti atakas tik tinklo lygyje. Gali būti reikalinga papildoma techninė įranga.	Reikalinga įdiegti kiekviename kompiuteryje atskirai. HIDS veikimui yra naudojami kompiuterio resursai. Įsibrovimui aptikti reikalinga išanalizuoti daugiau duomenų iš skirtingų šaltinių, o tai atsiliepia aptikimo greičiui.

2.2. Įsibrovimo aptikimo metodų tipai

Įsibrovimo aptikimo sistemose dažniausiai naudojami įsibrovimo aptikimo metodai skirstomi į šiuos pagrindinius tipus:

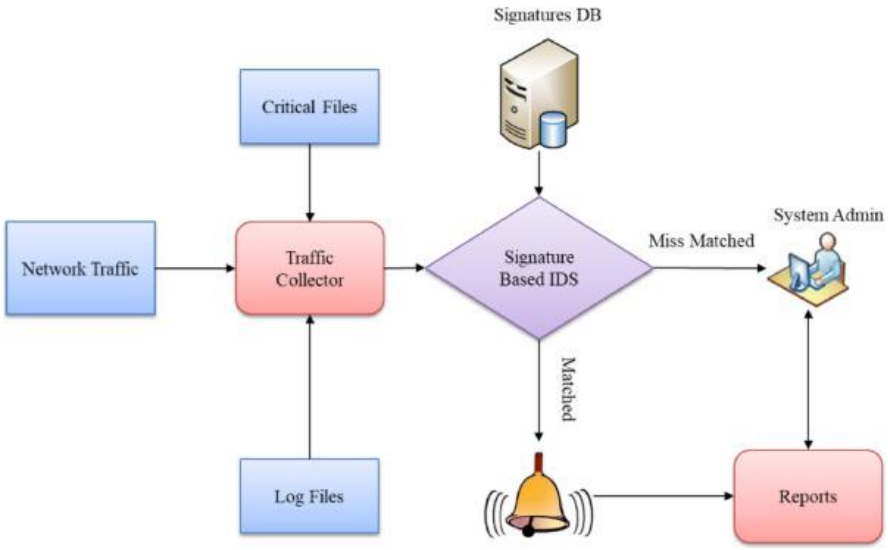
- parašais paremti (angl. *signature-based*), vadinamieji SIDS;
- anomalijomis paremti (angl. *anomaly-based*), vadinamieji AIDS.

Kiekvienas IDS metodas naudoja skirtingas technikas atakoms aptikti. Žinomiems atakų tipams, parašais pagrįstas metodas yra gana greitas ir efektyvus, tačiau jis negali atpažinti naujų, dar kitaip vadinamų nulinės dienos atakų. Anomalijomis pagrįstas metodas yra veiksmingas aptinkant naujas, anksčiau nematytas atakas, tačiau jį naudojant yra gaunama daug klaidingai teigiamų rezultatų, t.y. daug įprastų veiksmų yra palaikomi kenkėjiškais, nors jie tokie nėra. Šiame skyriuje plačiau nagrinėjamos SIDS ir AIDS įsibrovimo aptikimo metodologijos.

2.2.1. Parašais paremtas įsibrovimo aptikimas

Parašas – tai žinomą ataką atitinkantis šablonas. Parašais pagrįstame metode, įsibrovimas aptinkamas lyginant stebimus veiksmus su žinomais atakų parašais (Ozkan-Okay ir kt., 2021). Jei stebimas veiksmas turi įsibrovimo požymių, kurie yra apibrėžti parašų duomenų bazėje, tada IDS sistema įspėja apie įsibrovimą. Parašo pavyzdys galėtų būti – el. laiškas „Free software“ tema, kuri asocijuojasi su kenkėjišku programiniu kodu.

Parašais paremtas metodas yra paprastas aptikimo metodas, nes stebimi įvykiai, tokie kaip tinklo paketai ar žurnaliniai įrašai, yra tiesiog tikrinami su žinomais parašais esančiais parašų sąraše. Jei sąraše yra anksčiau apibrėžtos atakos sąlyga – sugeneruojamas įspėjimas apie galimai aptiktą įsibrovimą. Parašais pagrįstos IDS yra labai veiksmingos aptinkant žinomas grėsmes, tačiau yra nelabai veiksmingos aptinkant anksčiau nežinomas grėsmes. SIDS parašų duomenų bazės yra atnaujinamos vos tik yra sužinoma apie naujus pažeidžiamumus, tačiau jei ataka būtų šiek tiek modifikuota, jos aptikti gali nepavykti. Pavyzdžiui, jei įsilaužėlis kenkėjišką failą „system.exe“ pervadintų į „system2.exe“, lyginant su turimais parašais, IDS įsibrovimo gali nebeaptikti. Tam, kad parašais paremtas metodas būtų veiksmingas, reikalinga apibrėžti parašus kiekvienai įmanomai atakai, kurią galėtų atlikti įsilaužėlis. Taip pat parašų duomenų bazė turi būti nuolat atnaujinama. SIDS veikimas vizualizuojamas 2.4 pav..



2.4 pav. Parašais paremtos įsibrovimo aptikimo sistemos veikimo schema (Khan ir kt., 2020)

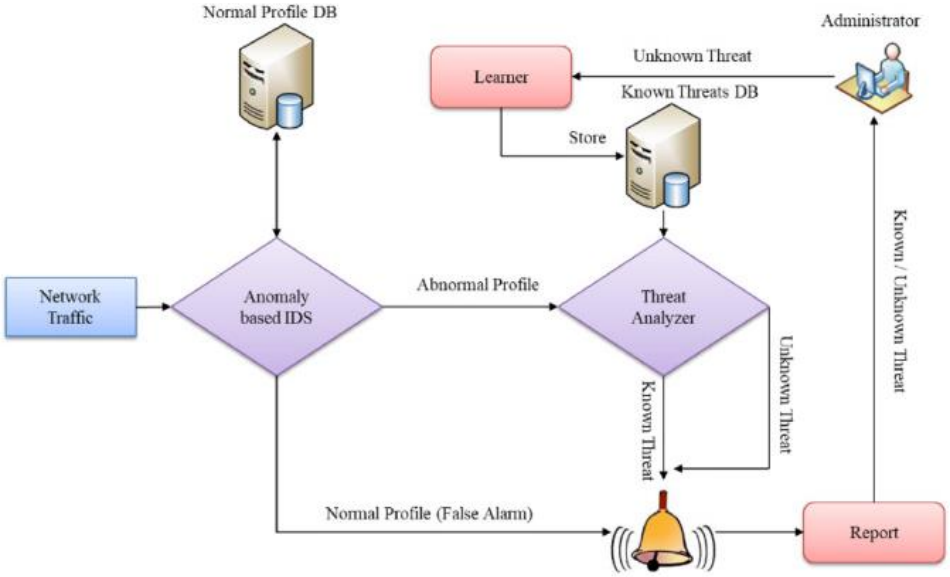
2.2.2. Anomalijomis paremtas įsibrovimo aptikimas

Anomalijomis paremtas įsibrovimo aptikimas yra procesas, kurio metu stebimų veiksmų duomenys yra lyginami su duomenimis, kurie yra apibrėžti kaip normali veikla tam, kad būtų identifikuojama anomali veikla. IDS, kurios naudoja anomalijomis pagrįstus aptikimo metodus turi apibrėžtas taisykles, kurios atspindi įprastą vartotojų, kompiuterių, tinklo ar programų elgesį. Šios taisyklės yra sudaromos tam tikrą laiką vykdant įprastą veiklą. Taisyklė, pavyzdžiui, gali būti vidutinis žiniatinklio naudojimo laikas darbo valandomis. Tada IDS lygina dabartinės veiklos charakteristikas su apibrėžtomis taisyklėmis, ir jeigu žiniatinklio naudojimas darbo valandomis yra didesnis nei įprastas, tada tokia veikla yra laikoma anomalia ir yra sugeneruojamas įspėjimas. Taisyklės gali būti sukuriamos daugeliui veiklos atributų, pvz., vartotojo išsiųstų el. laiškų skaičius, nepavykusių prisijungimo bandymų skaičius, per tam tikrą laiką perduotų tinklo paketų skaičius ir t.t.. Svarbiausias AIDS pranašumas yra tas, kad toks metodas yra veiksmingas aptinkant anksčiau nežinomas atakas.

Taisyklės, sukurtos anomalijoms aptikti, yra dviejų tipų: statinės arba dinaminės. Sukūrus statinių taisyklių rinkinį, rinkinys nesikeičia, nebent IDS yra nurodyta kurti naujas taisykles. Dinaminis taisyklių sąrašas yra nuolat atnaujinamas, kai įvyksta nauji įvykiai. Laikui bėgant sistemos ir tinklai kinta, todėl keičiasi ir naudojimosi jais įpročiai. Statinis sąrašas ilgainiui tampa nebegaliojantis, todėl jį reikia periodiškai atnaujinti. Dinaminiai profiliai neturi šios problemos, tačiau yra kitų spragų kuriomis gali pasinaudoti įsilaužėliai. Pavyzdžiui, užpuolikas gali atlikti nedidelį kiekį kenkėjiškos veiklos, tada po truputį didinti veiklos dažnumą ir kiekį. Jei pokyčių tempas yra pakankamai lėtas, AIDS kenkėjišką veiklą gali

laikyti normaliu elgesiu ir įtraukti ją į savo profilį. Netyčinis kenkėjiškos veiklos įtraukimas į normalaus elgesio profilį yra dažna anomalijomis pagristų IDS problema (Otoum & Nayak, 2021).

Kita AIDS problema yra ta, kad kai kuriais atvejais gali būti sudėtinga sukonfigūruoti normalaus elgesio profilius anomalijoms aptikti. Pavyzdžiui, kartą per mėnesį vyksta didelio failo perkėlimas, toks veiksmas gali būti palaikomas anomaliu, kadangi vyksta retai. Anomalijomis pagrįstose IDS dažnai gaunama daug klaidingai teigiamų rezultatų, dėl įprastos veiklos, kuri pagal savo charakteristikas labai nukrypsta nuo įprasto elgesio. Anomalijomis paremto metodo veikimas pavaizduotas 2.5 pav..



2.5 pav. Anomalijomis paremtos įsibrovimo aptikimo sistemos veikimo schema (Khan ir kt., 2020)

2.2 lentelėje yra pateikiamas parašais paremtų ir anomalijomis paremtų įsibrovimo aptikimo metodų tipų palyginimas. Metodologijos yra lyginamos pagal 4 kriterijus – įsibrovimo aptikimo rodiklius, aptikimo efektyvumą, priklausomybę srities žinioms, bei nežinomų atakų aptikimo efektyvumą.

2.2 lentelė. Įsibrovimo aptikimo metodų tipų palyginimas (Liu & Lang, 2019)

Metodologija / Vertinimo kriterijus	**SIDS**	**AIDS**
Aptikimo rodikliai	Žemas klaidingų aptikimų rodiklis, aukštas neaptiktų įsibrovimų rodiklis.	Žemas neaptiktų įsibrovimų rodiklis, aukštas klaidingų aptikimų rodiklis.
Aptikimo efektyvumas	Aukštas, priklauso nuo parašų kiekio duomenų bazėje.	Priklauso nuo modelio sudėtingumo.

Priklausomybė srities žinioms	Beveik visi aptikimai priklauso nuo srities pažinimo.	Maža priklausomybė.
Nežinomų atakų aptikimas	Aptinka tik žinomas atakas.	Aptinka žinomas ir nežinomas atakas.

2.3. Įsibrovimo aptikimo technikos

Įsibrovimo aptikimo sistemose labiausiai paplitusios įsibrovimo aptikimo technikos yra paremtos statistiniais metodais arba mašininiu mokymusi (Khraisat ir kt., 2019). Tobulėjant kompiuteriams, atsiranda galimybės efektyviau atlikti skaičiavimus, dėl to giliojo mokymosi metodai yra vis dažniau taikomi įvairiose srityse, tarp jų ir įsibrovimams aptikti. Tyrimų, susijusių su įsibrovimo aptikimu, taikant giliojo mokymo metodus pastaruoju metu yra atliekama vis daugiau ir dauguma jų pasiekia aukštus rezultatus. DL metodai, lyginant su ML, gali pasiekti ne tik aukštesnius rezultatus, bet taip pat gali dirbti su didesniais kiekiais duomenų (Čeponis, 2021). Šiame poskyryje apžvelgiamos 3 taikomos įsibrovimo aptikimo technikos – paremtos statistika, mašininiu mokymu ir giliuoju mokymu.

2.3.1. Statistiniais metodais paremta

Statistiniais metodais pagrįsti aptikimo metodai stebi įprastą veiklą, kad sukurtų įprastus profilius. Kai stebimi įvykiai skiriasi nuo įprasto profilio, tai parodo kad tai yra anomalija. Kiekvienai operacijai yra priskiriamas tam tikras balas, kad įsilaužimai būtų išskiriami iš įprasto srauto. Kai išmatuotas rezultatas yra didesnis nei slenkstinė reikšmė, sugeneruojamas įspėjimas. Slenkstinė vertė nustatoma atsižvelgiant į įvykių, įvykusių per nurodytą laikotarpį, skaičių. Sudarant įprastus profilius yra naudojami statistiniai rodikliai, tokie kaip vidurkis, mediana, moda, dispersija ir standartinis nuokrypis (Choudhary & Kesswani, 2021). Yra skirtingų tipų statistika paremtų technikų, jos gali būti suskirstytos į šias kategorijas: vienmačiai, daugiamačiai, laiko eilučių. Šios statistika paremtos technikos, taip pat gali būti suskirstomos į smulkesnes kategorijas – laiko eilučių modelis, operacinis modelis, Markovo proceso modelis ir kt..

Statistika paremtų IDS pagrindiniai privalumai:

- yra galimybė aptikti nulinės dienos atakas, nes nereikalingas atakos parašas;
- nesudėtinga prižiūrėti, kadangi nereikia atnaujinti;
- gali aptikti DDoS atakas;

Statistika paremtos technikos, taip pat turi ir trūkumų:

- įprastų profilių sukūrimas užtrunka;
- įprasti profiliai su laiku gali pasikeisti ir tapti nebeaktualūs;
- naudojami statistiniai skirstiniai turi būti tikslūs ir veiksmingi.

2.3.2. Mašininiu mokymu paremta

Mašininio mokymosi tikslas – automatizuoti duomenų analizės procesą be žmogaus įsikišimo. Mašininio mokymo technikos gali būti skirstomos į prižiūrimas ir neprižiūrimas. Yra daugybė metodų ir algoritmų, kurie yra naudojami anomalijoms aptikti, pvz., Bajeso, k-artimiausių kaimynų (KNN), atraminių vektorinių mašinų (SVM), sprendimo medžių (DT), dirbtinių neuroninių tinklų (ANN), genetinių algoritmų (GA) ir neraiškios logikos (angl. *fuzy logic*) (Ozkan-Okay ir kt., 2021). ML pagrįsti įsibrovimų aptikimo metodai gali aptikti naujas, nežinomas atakas. Nepaisant šios technikos pranašumų, siekiant pritaikyti šią techniką yra reikalinga turėti kokybiškus, pilnus duomenų rinkinius, kad būtų galima tinkamai apmokyti modelį ir įvertinti jo tinkamumą. Mokslininkai tiriantys šią techniką susiduria su problemomis gauti tinkamų duomenų (Khraisat ir kt., 2019). Kai kurie, anksčiau ištirti, duomenų rinkiniai nebuvo paviešinti, dėl privatumo problemų. Tačiau yra keletas viešai prieinamų duomenų rinkinių, kuriuos galima panaudoti tyrimo tikslams, bet jie neatspindi naujausių įsibrovimo tendencijų bei juose trūksta atakų tipų įvairovės, nes buvo sudaryti prieš kelis dešimtmečius. Kitas ML metodų trūkumas yra tas, kad daugumai ML algoritmų duomenų įvestyje yra reikalingi fiksuoto dydžio vektoriai. Bet žurnalinių įrašų duomenys, kurie naudojami ML algoritmo apmokymui, yra kintamo ilgio ir neturi tikrosios vertės. Siekdami išspręsti šią problemą, ankstesniuose darbuose, kiti tyrėjai naudojo dažniu pagrįstus duomenų atvaizdavimo metodus tokius kaip „Bag of Words“ ir TF-IDF. Tačiau šie metodai turi trūkumų, kad jie negali atspindėti nuoseklios sistemos žurnalo informacijos (Kim ir kt., 2021).

ML paremto įsibrovimo aptikimo privalumai:

- gali aptikti naujas, nulinės dienos atakas;
- metodų lankstumas;
- aukštas našumas;

ML metodų trūkumai:

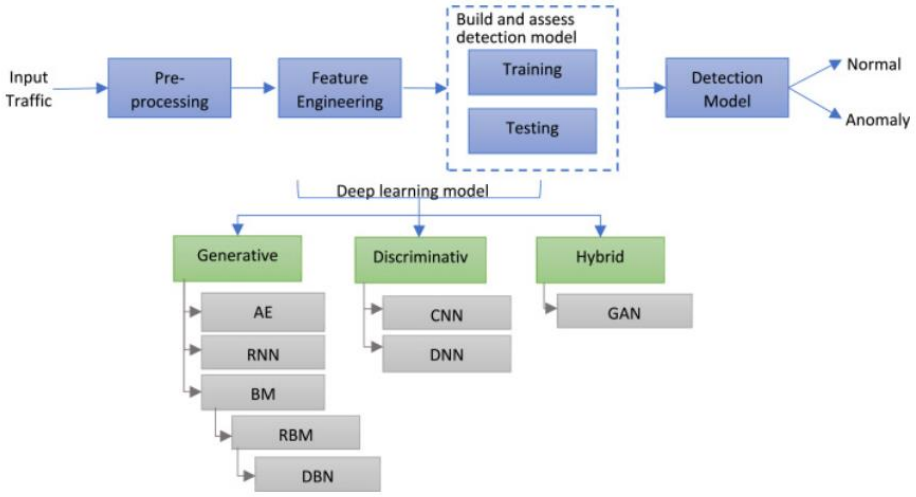
- reikalinga analizuoti didelius duomenų kiekius;
- prieš analizuojant duomenis, juos reikia tinkamai paruošti;
- aukštas klaidingai teigiamų rezultatų rodiklis;
- nėra daug atvirai prieinamų duomenų rinkinių su kuriais būtų galima eksperimentuoti.

2.3.3. Giliuoju mokymu paremta

Gilus mokymasis yra pažangesnė mašininio mokymosi šaka, kurioje neuroniniai tinklai susideda iš kelių sluoksnių, kurie atspindi mokymosi procesą. Gilus mokymasis gali susidoroti su dideliais duomenų kiekiais. Šis metodas parodė didelę sėkmę įvairiose srityse, tokiose kaip, vaizdų atpažinimas, garsų apdorojimas, natūralios kalbos apdorojimas, robotika bei autonominės sistemos. Dėl to mokslininkai plačiai taiko šį metodą ir kitose srityse, tarp jų ir įsibrovimų aptikimų tyrimuose.

Atsirandant naujoms technologijoms ir didėjant interneto srautui, gaunami didelio masto ir kelių dimensijų duomenys, o atakų scenarijai tampa vis sudėtingesni, todėl metodai, pagrįsti įprastu mašininiu mokymusi tampa neveiksmingi sprendžiant augančius saugumo iššūkius. Giliuoju mokymusi pagrįstas technikas naudojančiose IDS, gilieji tinklai mokosi iš istorinių srauto duomenų, kuriuos sudaro ir įprastas, ir anomalus srautas. Gilieji tinklai gali automatiškai sumažinti tinklo srauto sudėtingumą, kad būtų galima rasti duomenų koreliacijas be žmogaus įsikišimo. Be to, gilus mokymasis yra galingesnis aptinkant nulinės dienos atakas ir sudėtingus atakų modelius. Skirtingai nuo mašininio mokymosi algoritmų, gilaus mokymosi metodai gali būti sukurti taip, kad kartu atliktų duomenų požymių išgavimo ir klasifikavimo uždavinius (Aldweesh ir kt., 2020).

Įprasto mašininio mokymo algoritmuose, požymius duomenyse dažnai nustato ekspertas ir juos užkoduoja, o tai yra daug laiko reikalaujanti ir sudėtinga užduotis, kai yra dirbama su dideliu duomenų kiekiu. Pagrindinis skirtumas tarp įprasto mašininio mokymosi ir gilaus mokymosi yra giliųjų tinklų gebėjimas išmokti duomenų požymius skirtinguose apdorojimo sluoksniuose su skirtingais duomenų abstrakcijos lygiais, be žmogaus įsikišimo. Giliojo mokymo modeliai gali automatiškai surasti sudėtingas koreliacijas ir sąsajas tarp neapdorotų įvesties ir išvesties duomenų (Deng & Yu, 2013). Norint sukurti anomalijomis paremtą aptikimo modelį, gilaus mokymosi modelis gali būti apmokomas naudojant didelį kiekį istorinių duomenų. Modelis klasifikuoja naujus duomenis kaip įprastus arba anomalius. Jei naudojamas kelių klasių klasifikavimas, modelis toliau gali klasifikuoti anomalų srautą į dar smulkesnes atakos klases. Giliuoju mokymu paremtos IDS veikimas vaizduojama 2.6 pav..



2.6 pav. Giliuoju mokymu paremtos IDS veikimo schema (Aldweesh ir kt., 2020)

Gilaus mokymo metodai yra sudėtingesni, nes jie mokymo etape, per kelis paslėptus sluoksnius, taikant daugybę skirtingų parametrų, atlieka daug matematinių operacijų, kurios reikalauja daug laiko, bei yra intensyvios. Šioms operacijoms atlikti yra reikalingas didelis kiekis kompiuterio resursų, todėl DL metodai yra stipriai priklausomi nuo aparatinės įrangos.

DL metodų privalumai:

- automatinis duomenų požymių išskyrimas;
- geba susidoroti su labai dideliais duomenų kiekiais;
- galingas, efektyvus ir stipriai sumažina duomenų požymių erdvę;

DL metodų trūkumai:

- sudėtingiau realizuojami;
- reikalauja daug skaičiavimo resursų;
- paslėptų sluoksnių kūrimas užtrunka, bei pridėti papildomi sluoksniai retai padidina modelio našumą;
- nėra atsparūs vengimo atakoms.

2.4. HIDS duomenų rinkiniai

Kaip buvo rašoma ankstesniuose skyriuose, siekiant sėkmingai pritaikyti ML ar DL technikas įsibrovimams aptikti, reikalinga tinkamai apmokyti prognozavimo modelius. Modelio apmokymui yra reikalingi tam skirti duomenų rinkiniai. Įsibrovimams aptikti skirtų duomenų rinkinių, kurie būtų laisvai

prieinami nėra tiek daug. Kai kurie duomenų rinkiniai turi didelių trūkumų, pvz., yra pasenę ir neatitinką šių dienų atakų aktualijų. Populiariausi, mokslininkų tyrinėjami, duomenų rinkiniai skirti įsibrovimams aptikti yra „DARPA“, „KDD ’99“, „NSL-KDD“, „ADFA-LD“, „ADFWA-WD“ ir kiti. Kai kuriuos duomenų rinkinius sudaro duomenys skirti įsibrovimams aptikti tik tinklo lygmenyje. Šio darbo sprendžiama problema yra susijusi su įsibrovimo aptikimu kompiuterio lygmenyje, todėl yra apžvelgiami duomenų rinkiniai, kurie turi kompiuterio lygmens duomenų, tokių kaip, žurnaliniai įrašai, sisteminių funkcijų kvietimų pėdsakai, įrašai apie pasikeitimus registruose, failų sistemoje ir pan..

ADFA-LD

„ADFA-LD“ duomenų rinkinys yra skirtas kompiuterio lygmens įsibrovimams aptikti, „Linux“ operacinėje sistemoje. Duomenų rinkinį sudaro sisteminių funkcijų kvietimo pėdsakai, kurie buvo surinkti su „auditd“ įrankiu – programa, kuri yra skirta fiksuoti su saugumu susijusius įvykius „Linux“ OS. ”ADFA-LD“ duomenys buvo renkami kompiuteryje su pilnai atnaujintu „Ubuntu“ 11.04 OS su 2.6.38 branduolio versija, kuriame taip pat veikė kitos programos, pvz., žiniatinklio, duomenų bazių , „SSH“, „FTP“ serverių programinė įranga. Renkant duomenų rinkinį buvo vykdomos skirtingos atakos, atakų tipai aprašyti 2.3 lentelėje.

2.3 lentelė. Atakų tipai naudoti ADFA-LD duomenų rinkiniui generuoti (Borisaniya & Patel, 2015)

Atakos tipas	**Metodas**	**Atakos vektorius**	**Pėdsakų kiekis**
Hydra-FTP	Slaptažodžio „bruteforce“	FTP „bruteforce“ su Hydra	162
Hydra-SSH	Slaptažodžio „bruteforce“	SSH „bruteforce“ su Hydra	176
Adduser	Naujo vartotojo sukūrimas	Užkrėstas vykdomasis failas kliento pusėje	91
Java-Meterpreter	Java pagrindo Meterpreter	TikiWiki pažeidžiamumo išnaudojimas	124
Meterpreter	Linux Meterpreter Payload	Užkrėstas vykdomasis failas kliento pusėje	75
Webshell	C100 Webshell	Nutolusio PHP skripto įtraukimo pažeidžiamumas	118

ADFA-WD

Duomenų rinkinių, skirtų anomalijoms aptikti, kurie būtų pritaikyti „Windows“ OS yra labai mažai, dėl šios priežasties buvo sukurtas „ADFA-WD“ duomenų rinkinys (Haider ir kt., 2016). Rinkinį sudaro kompiuterio lygmens įrašai apie prieigos užklausas prie „Windows DLL“ bibliotekų ir sisteminių funkcijų kvietimai, kai buvo vykdomos įvairios atakos. Sudarant šį duomenų rinkinį buvo naudojama „Windows XP“ OS su „Service Pack“ 2 versija, kuri neturi ASLR (angl. *Address Space Layout Randomization*) ir DEP (angl. *Data Execution Prevention*) saugumo mechanizmų, o duomenims išgauti buvo naudojamas

„Procmon“ įrankis. Duomenims surinkti buvo vykdoma 12 skirtingo tipo, žinomų atakų, naudojant automatinius įsilaužimo įrankius, tokius kaip „Metasploit“. Atakos, kurios buvo vykdomos renkant šį duomenų rinkinį pateikiamos 2.4 lentelėje.

2.4 lentelė Atakos naudotos ADFA-WD duomenų rinkinio duomenims sugeneruoti (Borisaniya & Patel, 2015)

Pažeidžiamumas	Pažeidžiamas komponentas	Pažeidžiamumo išnaudojimo metodas	Pėdsakų kiekis
CVE: 2006-2961	CeasarFTP 0.99g	Reverse Orindal Payload injekcija	454
EDB-ID: 18367	XAMPP Lite v1.7.3	Kenkėjiškų skriptų įkėlimas ir leidimas per Xampp_webdav	470
CVE: 2004-1561	Icecast v2.0	Metasploit	382
CVE: 2009-3843	Tomcat v6.0.20	Metasploit	418
CVE: 2008-4250	OS SMB	Metasploit	355
CVE: 2010-2729	OS Print Spool	Metasploit	354
CVE: 2011-4453	PMWiki v2.2.30	Metasploit	430
CVE: 2012-0003	Wireless Karma	DNS klastojimas naudojant Pineapple maršrutizatorių	487
CVE: 2010-2883	Adobe Reader 9.3.0	Reverse Shell iškvietimas per kenkėjišką PDF failą	440
-	Backdoor vykdomasis failas	Reverse Inline Shell iškvietimas	536
CVE: 2010-0806	IE v6.0.2900.2180	Metasploit	495
-	Užkrėsti išoriniai įrenginiai	Blind Shell iškvietimas	621

Firefox-DS

„Firefox-DS“ duomenų rinkinys buvo sukurtas naudojant šiuolaikines įsilaužimo testavimo technologijas, tokias kaip „Metasploit“ (Murtaza ir kt., 2013). Duomenų rinkinys yra skirtas kompiuterio lygmens įsibrovimams aptikti „Linux“ operacinėse sistemose. Duomenų rinkinį sudaro įprastos ir anomalios veiklos sisteminių funkcijų kvietimų įrašai. Įprastos veiklos sisteminių funkcijų kvietimų pėdsakai yra renkami leidžiant septynis skirtingus testavimo rinkinius „Firefox“ 3.5 naršyklėje. Kiekvienas testų rinkinys yra atsakingas už skirtingas operacijas, kurios apima daugelį „Firefox“ funkcijų tam, kad būtų galima kiek įmanoma tiksliau nustatyti įprastą elgesį. Renkant anomalios veiklos duomenis, buvo vykdomos 5 modernios atakos, tokios kaip, „Memory Corruption“, „Integer Overflow“, „DOM Exploit“, „Dangling Pointer“, „Null Pointer“ atakos.

NGIDS-DS

„NGIDS-DS“ duomenų rinkinys yra dirbtinai sudarytas naudojant „IXIA Perfect Storm“ įrankį tam, kad sugeneruoti įprasto elgesio duomenis, taip pat septinių skirtingų tipų atakų duomenis. Atakos naudojamos šio rinkinio sudarymui yra šių tipų – „Exploit“, „DoS“, „Worms“, „Generic“, „Reconnaissance“, „Shellcode“ ir „Backdoors“. Duomenims surinkti buvo naudojami 2 kompiuteriai su

„Ubuntu“ 14.04 OS. Pirmame kompiuteryje buvo įdiegtas žurnalų rinkimo mechanizmas, taip pat veikė el. pašto, „FTP“, duomenų saugojimo programinė įranga. Antrame kompiuteryje buvo įdiegtas „tcpdump“ įrankis, kurio pagalba buvo surinkti tinklo srauto duomenys (Haider ir kt., 2017) . Šio duomenų rinkinio duomenys gali būti panaudojami tiek kompiuterio lygmens, tiek tinklo lygmens įsibrovimams aptikti. Duomenų rinkinį sudaro 99 failai, kuriuose yra sužymėti kompiuterių, su „Linux“ OS, žurnaliniai įrašai, bei tinklo srauto duomenys. Kompiuterio žurnaliniuose įrašuose yra tokie duomenys, kaip įvykių laikas, procesų identifikatoriai, programų vykdomųjų failų keliai, sisteminių funkcijų kvietimai ir kt. duomenys. „NGIDS-DS“ duomenų rinkinio atributai detalizuojami 2.5 lentelėje.

2.5 lentelė. NGIDS duomenų rinkinio atributų aprašymas

Atributas	Aprašymas
date	Įvykio data
time	Įvykio laikas
pro_id	Proceso, kuris veikė kompiuteryje, identifikatorius
path	Vykstančių procesų, vykdomųjų failų keliai failų sistemoje
sys_call	Sisteminės funkcijos, kuri buvo kviečiama vykstančių procesų, identifikatorius
event_id	Įvykio unikalus identifikatorius
attack_cat	Atakos tipas (pvz. DoS, Worms, Backdoors)
attack_subcat	Atakos tipo smulkesnė kategorija
label	Skaičius 0 arba 1, kuris parodo ar įvykis normalus ar anomalus

Apibendrinimui buvo sudaryta apžvelgtų duomenų rinkinių lentelė, kurioje palyginami kiekvieno duomenų rinkinio privalumai, trūkumai, duomenys ir operacinė sistema. Palyginimas yra pateikiamas 2.6 lentelėje.

2.6 lentelė. Kompiuterio lygmens įsibrovimo aptikimo duomenų rinkinių palyginimas

Duomenų rinkinys	Operacinė sistema	Duomenys	Privalumai	Trūkumai
ADFA-LD	Linux	Normalios ir anomalios veiklos sisteminių funkcijų kvietimų pėdsakai.	Realistiški, sužymėti duomenys, yra nulinės dienos atakų duomenų.	Sisteminių funkcijų kvietimų duomenis sudaro tik jų numeriai, anomalios veiklos duomenims generuoti naudojami tik 6 atakų tipai.

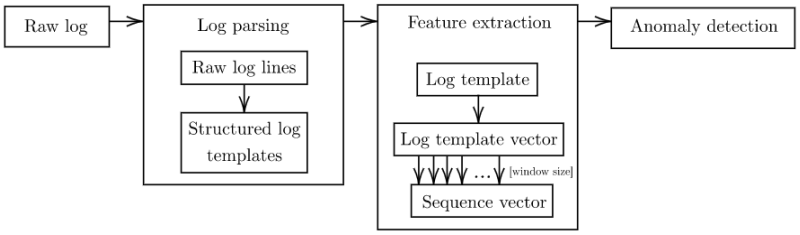
Duomenų rinkinys	**Operacinė sistema**	**Duomenys**	**Privalumai**	**Trūkumai**
ADFA-WD	Windows	Normalios ir anomalios veiklos sisteminių funkcijų kvietimų pėdsakai, prieigos prie DLL bibliotekų žurnalas.	Vienas iš nedaugelio duomenų rinkinių orientuotų į Windows OS.	Anomalios veiklos duomenims generuoti naudojamas mažas kiekis skirtingų atakų tipų.
FIREFOX-DS	Linux	Normalios ir anomalios veiklos sisteminių funkcijų kvietimų pėdsakai.	Didelis duomenų kiekis, pakankamai pilni normalios veiklos duomenys.	Duomenys surinkti tik vykdant atakas orientuotas į Firefox naršyklę, duomenų rinkinys nėra dažnai naudojamas atliekant įsibrovimo aptikimo tyrimus.
NGIDS-DS	Linux	Sužymėti kompiuterio žurnaliniai įrašai (įvykių laikas, procesų identifikatoriai, programų vykdomųjų failų keliai, sisteminių funkcijų kvietimai ir kt.), tinklo srauto duomenys.	Dirbtiniu būdu sudaryti duomenys, kurie yra realistiški.	Duomenų rinkinys nėra plačiai ištirtas kitų mokslininkų.

2.5. Žurnalinių įrašų analizės metodai

Modernios kompiuterinės sistemos tampa vis didesnės ir sudėtingesnės. Visos jų generuoja daugybę įvykių apie sistemų veikimą, kurie yra įrašomi į žurnalų failus. Nors dauguma įrašų praneša apie įprastus veiksmus, tačiau rečiau pasitaikantys neįprasti įrašai gali indikuoti tam tikrą neįprastą veiklą, pavyzdžiui, įvykusį gedimą, vykdomą ataką ar kompiuterio užsikrėtimą virusu. Sistemų žurnalai reikalauja kruopštaus stebėjimo, nes kiekviena įvykusi anomalija gali būti reikšminga sistemų pasiekiamumui, duomenų konfidencialumui ir integralumui (Ryciak ir kt., 2022). Tačiau didelės sistemos generuoja labai didelius

kiekius žurnalinių įrašų, todėl atlikti žurnalų stebėjimą žmogui yra sudėtinga ir brangi užduotis. Taip pat tai atliekant žmogui, gali pasitaikyti atvejų kada nėra pastebimi reikšmingi su saugumu susiję įvykiai. Žurnalinių įrašų analizė yra plačiai naudojama siekiant pagerinti sistemų pasiekiamumą, saugumą, palengvinti gedimo priežasties nustatymą, aptikti įsibrovimus ir anomalijas.

Pastaruoju metu, žurnalinių įrašų analizės metodai yra plačiai nagrinėjama ir vystomi tyrėjų (Le & Zhang, 2022). Dauguma tyrėjų, vystydami įsibrovimo aptikimo iš žurnalinių įrašų modelius, žurnalinių įrašų analizę išskiria į 3 žingsnius – žurnalinio įrašo struktūrizavimą (angl. *parsing*) iš neapdorotos žurnalo eilutės, žurnalinio įrašo ar jų sekos vektorizavimo (angl. *feature extracting*), bei anomalijos aptikimo, kaip pavaizduota 2.7 pav..

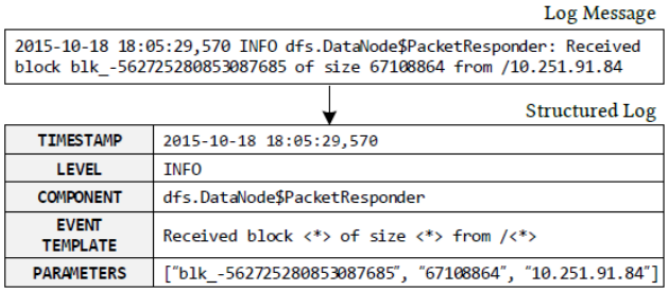


2.7 pav. Žurnalinių įrašų analizės pagrindiniai etapai (Wang ir kt., 2020)

Žurnaliniam įrašui paversti į struktūrizuotą formą yra naudojami automatizuoti struktūrizavimo metodai, kurie yra paremti euristiniais metodais, įvykių klasterizavimu, ilgiausios bendros sekos ir kitais metodais. Žurnaliniams įrašams vektorizuoti dažnai naudojami natūralios kalbos apdorojimo (angl. *Natural Language Processing*) žodžių įterpimo metodai, tokie kaip „word2vec“, „fastText“ ir panašūs. Anomalijoms aptikti yra naudojami mašininio mokymo ir giliojo mokymo metodai. Kiekvienas iš šių žingsnių plačiau aprašomas šiame poskyryje.

2.5.1. Žurnalinių įrašų struktūrizavimas

Žurnaliniai įrašai yra tekstiniai duomenys, kurie yra pusiau struktūrizuoti, bei turi statinę dalį, kuri parodo tam tikrą pranešimą ir kintamą dalį, kurioje yra tam tikri parametrai. Žurnalinių įrašų šablonų sudarymas iš žurnalinių įrašų yra vienas iš struktūrizavimo metodų. Šablonų sudarymas yra paruošiamasis darbas, kurio atlikimo tikslumas turi didelę reikšmę vėlesniuose etapuose. Dauguma struktūrizavimo įrankių išskaido žurnalinį įrašą į 2 dalis – kintamą ir pastovią, kaip pavaizduota 2.8 pav..

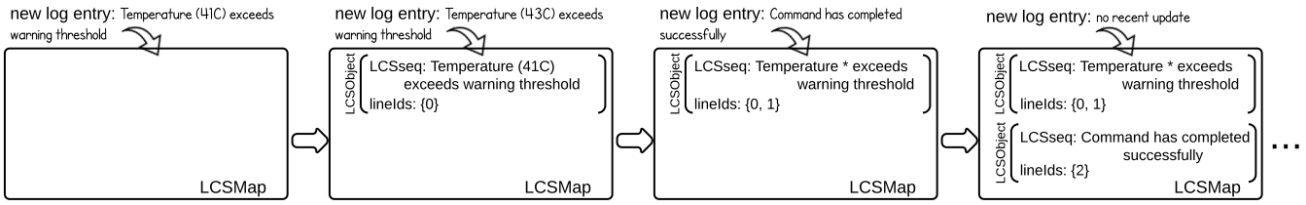


2.8 pav. Žurnalinio įrašo pavertimas į struktūrizuotą šabloną (Zhu ir kt., 2018)

Labiausiai paplitę žurnalų šablonų struktūrizavimo metodai yra šie – „iPLoM“ (Makanju ir kt., 2012), kuris yra paremtas euristiniais metodais, „Drain“ (He ir kt., 2017), kuris yra paremtas priešdėlio medžio struktūra, „Spell“ (Du & Li, 2016), kurio veikimas paremtas ilgiausio bendro posekio (angl. *Longest Common Subsequence, LCS*) metodu ir „LogSig“ (Tang ir kt., 2011), kuris naudoja srauto klasterizavimo metodus.

„iPLoM“ (angl. *Iterative Partitioning Log Mining*) veikimas yra paremtas euristiniais metodais, kurie apskaičiuoja kiek kartų kiekvienas žodis pasikartoja tam tikroje pozicijoje žurnaliniame įraše, tada dažniausiai pasitaikantys žodžiai yra parenkami kaip šablono įvykiu kandidatai. Galiausiai iš įvykių kandidatų yra sudaromi įvykių šablonai. „LogSig“ sisteminiams įvykiams generuoti iš žurnalo pranešimų, naudoja pranešimo parašais paremtą algoritmą, pagal kurį įvykius skirsto į klasterius. Ieškodamas būdingų pranešimų parašų, „LogSig“ suskirsto žurnalo pranešimus į įvykių tipų rinkinį. Šis metodas gali tvarkyti įvairių tipų žurnalų duomenis ir gali įtraukti žmogaus srities žinias, kad pasiektų aukštesnį našumą. „Drain“ yra vienas iš dažniausiai naudojamų žurnalinių įrašų analizės algoritmų. Jis gali analizuoti žurnalus srautiniu būdu ir laiku. Kad pagreitintų analizės procesą, „Drain“ naudoja fiksuoto gylio analizės medį, kuriame užkoduotos specialiai sukurtos analizės taisyklės.

„Spell“ yra realaus laiko srautinis žurnalinių įrašų struktūrizavimo metodas, kuriame naudojamas ilgiausiu bendru posekiu pagrįstas algoritmas. Metodo pagrindinė idėja yra ta, kad jei į žurnalo išvestį žiūrima kaip į seką, daugumoje žurnalo spausdinimo sakinių nekintanti dalis, žyminti pranešimo tipą, dažnai užima didžiąją sekos dalį, o parametrų reikšmės – tik nedidelę. Jei du žurnalo įrašai gaunami pagal tą patį žurnalinio įrašo išvedimo sakinį, bet skiriasi tik skirtingomis parametrų vertėmis, labai tikėtina, kad šių dviejų sekų ilgiausias bendras posekis yra žurnalinio įrašo sakinio išvedimo nekintama dalis, kuri reiškia pranešimo tipą. Metodo veikimas pavaizduotas 2.9 pav..



2.9 pav. Spell metodo veikimas (Du & Li, 2016)

Iš pradžių *LCSMap* sąrašas yra tuščias. Kai gaunamas naujas žurnalo įrašas *e_i*, jis pirmiausia išskaidomas į ženklų seką *s_i*, naudojant skirtukų rinkinį. Po to palyginami *s_i* su visais objektais *LCSObject*, esančiais dabartiniame *LCSMap*, kad nustatyti, ar *s_i* atitinka vieną iš esamų sekų *LCSseq* (todėl eilutės ID *i* yra pridedamas prie atitinkamo *LCSObjecti LineIds*). Kitu atveju sukuriamas naujas objektas tarp *LCSMap*.

2.5.2. Žurnalinių įrašų vektorizavimas

Kai kurie anomalijų aptikimų iš žurnalinių įrašų sprendimai yra pagrįsti tradiciniais duomenų gavybos metodais, pavyzdžiui, naudojant reguliarias išraiškas (angl. *Regular Expression*), kurios filtruoja tam tikrus įrašus, kuriuose yra tam tikri raktiniai žodžiai galintys indikuoti kad tai yra anomalija (pavyzdžiui, ieškoma raktažodžių „Error“, „Failure“ ir pan.) (Ryciak ir kt., 2022). Tokie sprendimai pasižymi ne tik dideliu klaidingai teigiamų rezultatų skaičiumi, bet ir negali aptikti kai kurių tipų anomalijų. Ar žurnalinis įrašas ar įrašų seka turėtų būti klasifikuojamas kaip anomalija, priklauso ne tik nuo paties įrašo turinio, bet ir nuo jį supančių kitų žurnalinių įrašų. Tradiciniai metodai leidžia aptikti tokias anomalijas, kaip „grubios jėgos“ (angl. *Brute force attack*) ar paslaugos trikdymo atakos (angl. *Denial of service*). Kai kuriuose metoduose daugiausia dėmesio skiriama atskirų žurnalinių įrašų statistinių požymių išskyrimui, pavyzdžiui, tam tikro šablono pasireiškimo dažnumui, laiko intervalui tarp įrašų arba įrašų, klasifikuojamų kaip tam tikras šablonas, skaičiui. Tačiau žurnalinių įrašų analizei naudojant NLP metodus, remiantis eksperimentų rezultatais, žurnalinių įrašų šablonų sekų požymiai yra išgaunami universaliau, todėl gaunami aukštesni anomalijų aptikimo vertinimo rodikliai (Ryciak ir kt., 2022).

Pastaruoju metu, tyrėjai vis dažniau taiko natūralios kalbos atpažinimo metodus žurnalinių įrašų analizei. Žurnaliniams įrašams vektorizuoti yra taikomi „bag-of-words“, TF-IDF, „N-gram“, „word2vec“, „FastText“ ir kt. metodai.

TF-IDF

TF-IDF (angl. *Term Frequency and Inverse Document Frequency*) yra statistinis dydis, kuris dažniausiai naudojamas kaip svorio tipas teksto gavybos srityje. Jis proporcingai įvertina kiek kartų žodis

pasikartoja dokumente, bet taip pat įtraukia įvertį pagal tai, kaip dažnai žodis vartojamas visuose dokumentuose. Vienas iš jo naudojimo atvejų yra nereikšmingų žodžių (angl. *stop words*) paieška. Žodžio terminų dažnis (TF) dokumente normalizuojamas atsižvelgiant į dokumento ilgį. Atvirkštinis dokumento dažnis (IDF) įvertina kaip dažnai žodžiai pasitaiko visuose dokumentuose, todėl visame turinyje pasitaikantys žodžiai turi mažesnę reikšmę, o žodžiai, būdingi tik keliems dokumentams turi didesnę reikšmę (Meng, Liu, Zhang, ir kt., 2019). Šiuos du skaičius padauginus gaunamas dokumento žodžio TF-IDF balas. Kuo aukštesnis balas, tuo svarbesnis žodis yra tame dokumente.

Analizuojant žurnalinius įrašus TF-IDF yra naudojamas žurnalų įvykiams paversti į vektorius, pavyzdžiui, anomalijų aptikimo tyrime, paruošiant duomenis, atskiriamos nekintamos žurnalinių įrašų dalys ir jos sugrupuojamos, tada kiekvienoje žurnalo duomenų grupėje suskaičiuojama kiek yra žodžių tipų ir jie įrašomi į žodžių rinkinį. Galiausiai kiekvienai žurnalo eilutei apskaičiuojama kiekvieno žodžio TF-IDF vertė (Ying ir kt., 2021). „LogClass“ (Meng, Liu, Zhang, ir kt., 2019) anomalijų klasifikavimo iš žurnalinių įrašų modelyje, taip pat išfiltruojama kintama žurnalinio įrašo dalis žodynui sumažinti, o nekintama dalis paverčiama į vektorių naudojant TF-IDF metodą. Kitame anomalijų aptikimo tyrime, žurnalinio įrašo žodžiai paverčiami į fiksuoto dydžio vektorius naudojant žodžių įterpimo metodus, o visų žodžių vektoriams apjungti ir galutiniam įvykio vektoriui išgauti, apskaičiuojama vektorių sandauga su atitinkamais TF-IDF svoriais (Zhang ir kt., 2019). TF-IDF panašiai išnaudojamas ir „LogEvent2Vec“ (Wang ir kt., 2020) modelyje, kuriame žurnalo įvykis yra traktuojamas kaip žodis. Įvykių vektoriams paversti į vieną sekos vektorių yra naudojamas TF-IDF.

N-gram

„N-gram“ NLP kontekste reiškia gretimą teksto *n* elementų seką. Vietoj to, kad vienas žodis būtų vektorius, *n* žodžių junginys yra vektorius. Tai naudojama siekiant iš kiekvieno žodžio gauti daugiau konteksto (Mikolov, Chen, ir kt., 2013). Žemiau pateikiamas paprastas 2 žodžių „n-gram” pavyzdys.

Tai yra paprastas pavyzdys → <Tai, yra>, <yra, paprastas>, <paprastas, pavyzdys>

„N-gram“ taip pat galima sudaryti iš žodžio raidžių ar simbolių. Pateikiamas paprastas 2 ženklų „n-gram“ pavyzdys:

Pavyzdys → <pa>,<av>,<vy>,<yz>,<zd>,<dy>,<ys>

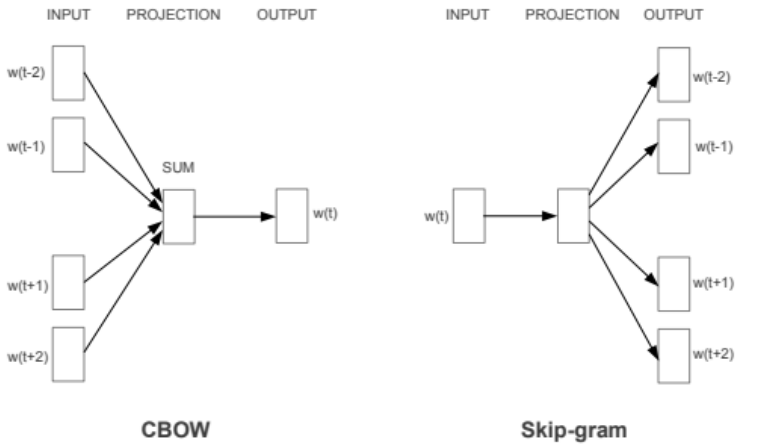
Žurnalinių įrašų analizės srityje „n-gram“ modelis naudojamas ne tik vektoriams sudaryti, bet ir kai anomalūs įvykiai yra nustatinėjami iš žurnalinių įrašų sekų, kad išmokti galimus kitus įvykius, pagal kuriuos yra prognozuojama ar kiti įvykiai bus anomalūs ar ne. „DeepLog“ modelyje (Du ir kt., 2017), kurio veikimas

paremtas LSTM algoritmu, autoriai tyrime naudoja „n-gram“ modelį palyginimui su naujai sukurtu „DeepLog“ modeliu, kad palygintų jo tikslumą. Kitame anomalijų aptikimo tyrime (Mäntylä ir kt., 2022), taip pat analizuojamos žurnalinių įrašų įvykių sekos bandant nustatyti sekantį įvykį naudojant „n-gram“ metodą, analizuojant viešai prieinamus žurnalinius įrašus iš „HDFS“ duomenų rinkinio. Palyginus pasiūlytą „n-gram“ metodu paremtą modelį su giliuoju mokymu paremtu modeliu, nustatyta, kad „n-gram“ modelis yra tik truputį mažiau tikslus, bet jo mokymo ir spėjimo laikas yra daug kartų mažesnis.

Word2vec

„Word2Vec“ – tai mašininiu mokymusi pagrįstas neprižiūrimas modelis. Jis yra negilus, nes jo neuroniniai tinklai naudoja tik 2 sluoksnius. Jis generuoja žodžių vektorinius pavidalus, kurie perteikia semantinį ir kontekstinį panašumą. „Word2Vec“, kurdamas žodžių įterpimus (angl. *Word Embeddings*), naudoja arba tęstinį žodžių maišo (angl. *Continuous Bag Of Words, CBOW*) modelį, arba „skip-gram“ modelį. Žodžiai, kurie kontekste yra panašesni, žodžių įterpimo erdvėje bus artimesni nei žodžiai iš kito konteksto. Originaliame „word2vec“ variante hierarchinis „SoftMax“ naudojamas kaip išvesties vienetas, o žodynas pateikiamas kaip Huffmano dvejetainis medis. Huffmano dvejetainis medis bendriems žodžiams priskiria trumpus dvejetainius kodus, kurie šiuo atveju sumažina neuroninio tinklo reikalingų išvesties vienetų skaičių (Mikolov, Chen, ir kt., 2013).

CBOW modelis yra neprižiūrimas neuroninio tinklo (NN) kalbos modelis, kuris prognozuoja einamąjį tikslinį žodį, remdamasis aplinkiniais žodžiais, kurie turi tą patį kontekstą. Modelis naudoja tekstų duomenų rinkinius kaip mokymo duomenis, išlaikydamas dabartinį tikslinį žodį ir prognozuoja bei lygina rezultatą su tekstų duomenimis. CBOW modeliui nėra svarbus žodžių eiliškumas, nes jis skaičiuoja aplinkinių žodžių įterpinių vidurkį (Mikolov, Chen, ir kt., 2013). „Skip-gram“ modelį galima apibūdinti kaip atvirkštinį CBOW modelį. Šis modelis – tai neprižiūrimas NN kalbos modelis, kuris paima žodį (įvesties žodį) sakinio viduryje ir nuspėja žodžius, kurie greičiausiai bus šalia šio žodžio (aplinkiniai žodžiai). Modelio išvestis bus visų žodyno žodžių tikimybė, o mokymo metu šios išvestys mokomos atvaizduoti artimus žodžius (Mikolov, Sutskever, ir kt., 2013). CBOW ir „skip-gram“ modelių įvesties ir išvesties pavyzdys yra pateikiamas 2.10 pav..



2.10 pav. CBOW ir skip-gram modeliai (Mikolov, Chen, ir kt., 2013)

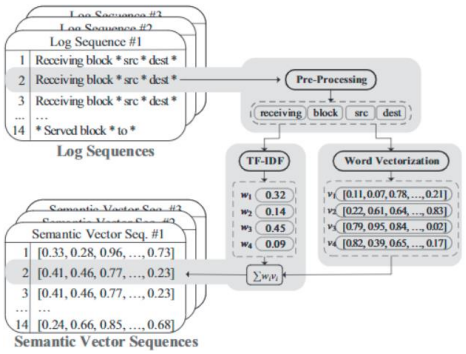
Kaip ir kiti NLP teksto apdorojimo metodai, „Word2vec“ yra naudojamas žurnaliniams įrašams perteikti į skaitinį pavidalą. „LogAnomaly“ anomalijų iš žurnalinių įrašų aptikimo modelyje žurnalo srautas modeliuojamas kaip natūralios kalbos seka (Meng, Liu, Zhu, ir kt., 2019). Tyrėjai pasiūlė paprastą, „word2vec“ modeliu paremtą, duomenų požymių išskyrimo metodą „template2vec“, kuris išgauna žurnalo šablonuose paslėptą semantinę informaciją, naudojant pasiskirstymo leksikos ir kontrasto įterpimo (angl. *distributional lexical contrast embedding , dLCE*) modelį. „Word2vec“ modelio apmokymui, analizuojant žurnalinius įrašus, reikalingas didelis kiekis skaičiavimo resursų, nes reikalinga vykdyti kelių tipų transformacijas (transformuoti žodžius, tada įvykį, galiausiai visų įvykių seką), dėl to buvo pasiūlytas patobulintas „LogEvent2vec“ (Wang ir kt., 2020) žurnalinių įrašų vektorizavimo metodas, kuris „word2vec“ įvesčiai ima ne vieną žodį, o struktūrizuotą žurnalinio įrašo eilutę, išlaikydamas konteksto sąsajas tarp žurnalinių įrašų. „LogEvent2vec“ transformuoja įvykių vektorių į žurnalo sekos vektorių. Kadangi žurnalo įvykių skaičius yra daug mažesnis už žodžių skaičių, „LogEvent2vec“ sumažina mokymo sąnaudas.

FastText

„FastText“ modelyje apdorojamas ne pilnas žodis, o jo simbolių „n-gram“ rinkinys, kai tuo tarpu „word2vec“ modelyje kiekvienas žodis nagrinėjamas kaip atskiras vienetas. Tai ypač naudinga dirbant su kalbomis, kuriose yra daug to paties žodžio variacijų. Naudojant šį modelį, retesni žodžiai turi daugiau galimybių gauti gerą atvaizdavimą, nes simbolių „n-gram“ rinkinys pasitaiko dažniau nei pats žodis (Bojanowski ir kt., 2016). FastText naudoja CBOW ir „skip-gram“ modelius, kurie sukuria skaitinį žodžių atvaizdavimą. Kuo arčiau vektoriai yra skaitinėje erdvėje, tuo artimesnis yra jų kontekstas ir reikšmė.

Atliekant teksto analizę paprastai naudojamas lematizavimas (angl. *lemmatization*) ir kamienų formavimas (angl. *stemming*), kad būtų sumažintas skirtingų žodžių skaičius. Lematizavimas naudojasi kalbos taisyklėmis, kad suderintų tos pačios reikšmės žodžius, o kamienų formavimas nukerpa žodžių galus, kad suderintų panašius žodžius. Vietoje to galima naudoti simbolių „n-gram“ rinkinius, ką ir naudoja „FastText“ metodas. Kalbant apie žurnalinius įrašus tai yra labai naudinga, nes taip galima geriau perteikti žurnalo įvykių prasmę, kadangi žurnalo įvykiuose yra kintamų reikšmių. Tai reiškia, kad į duomenis galima įtraukti ir niekada anksčiau nematytus kintamuosius (Ryciak ir kt., 2022).

„FastText“ metodas, žurnaliniams įrašams paversti į vektorius naudojamas „LogRobust“ anomalijų aptikimo modelyje (Zhang ir kt., 2019). „LogRobust“ modelyje išgaunama semantinė žurnalo įvykio informacija ir kiekvienas žurnalo įvykis paverčiamas fiksuoto dydžio vektoriumi (semantiniu vektoriumi), neatsižvelgiant į tai, ar toks įvykis egzistavo anksčiau. Semantinio vektorizavimo eigą sudaro trys etapai: žurnalo įvykio paruošimas, žodžių vektorizavimas naudojant „FastText“ ir žodžių vektorių agregavimas su TF-IDF svoriais. Paruošimo etape, struktūrizuotas žurnalinio įrašo šablonas yra apdorojamas – iš jo išimami nereikšmingi žodžiai, simboliai, o sujungti žodžiai išskaidomi. Po paruošimo, kiekvienas žurnalo įvykis yra paverčiamas į semantinį vektorių, kuris turi atitikti 2 reikalavimus – semantiniais vektoriais turėtų būti galima atvaizduoti skirtingus žurnalo įvykius su dideliu skiriamuoju požymiu, t.y. įvykių vektoriai, kurie yra labai skirtingi, turėtų stipriai skirtis vektorinėje erdvėje. Kitas reikalavimas – semantiniais vektoriais turėtų būti galima nustatyti nestabilius žurnalo įvykius su panašia prasme, t.y. įvykiu vektoriai, kurie yra skirtingi, bet turi tą pačia reikšmę, turėtų būti atvaizduojami panašiais vektoriais vektorinėje erdvėje. Kad įgyvendinti šiuos du reikalavimus, „LogRobust“ įtraukia semantinę žurnalo įvykių informaciją į vektorizavimą. Modelis panaudoja paruoštus žodžių vektorius, kurie buvo iš anksto apmokyti naudojant „FastText“ algoritmą, iš „Common Crawl Corpus“ duomenų rinkinio, kad iš anglų kalbos žodynų išgautų semantinę informaciją. „FastText“ gali pakankamai gerai perteikti natūralios kalbos žodžių vidinį ryšį (t. y. semantinį panašumą) ir atvaizduoti kiekvieną žodį į d-matmenų vektorių (kai „FastText“ žodžių vektoriuose $d = 300$). Žodžius pakeitus atitinkamais vektoriais, žurnalo įvykio sakinys yra paverčiamas žodžių vektorių sąrašu, kuris yra suagreguojamas su žodžių TF-IDF svoriais siekiant gauti fiksuoto ilgio vektorius, kadangi žodžių skaičius skirtinguose įvykiuose gali skirtis. „LogRobust“ semantinio vektorizavimo veikimo schema vaizduojama 2.11 pav..



2.11 pav. LogRobust semantinis vektorizavimas (Zhang ir kt., 2019)

2.5.3. Anomalijų aptikimas

Kad būtų galima patogiai ir greitai nustatyti sistemų žurnalų pranešimų nukrypimus, yra taikomi žurnalų sekų anomalijų aptikimo metodai, kurie gali savarankiškai ir tiksliai aptikti anomalijas analizuojant žurnalinių įrašų sekas. Dabartinius pagrindinius žurnalo sekų anomalijų aptikimo metodus galima suskirstyti į 2 tipus – mašininiu mokymu ir giliuoju mokymu paremtus metodus (Wang ir kt., 2022). Žurnalinių įrašų analizės sritis yra aktyviai tyrinėjama mokslininkų ir per paskutinį dešimtmetį buvo pasiūlyta įvairių anomalijų aptikimo modelių paremtų ML ir DL metodais. Ankstesniuose tyrimuose buvo taikomi ML paremti metodai, pavyzdžiui, SVM (Liang ir kt., 2007), klasterizavimo metodai (Lin ir kt., 2016) ir kt.. Šiems ML metodams būdingi tam tikri trūkumai: nelankstumas, neefektyvumas ir silpnas gebėjimas prisitaikyti. Siekiant išspręsti šias problemas, buvo pritaikyti DL metodai, kurie parodė gerus rezultatus (Le & Zhang, 2022). Naujausi tyrėjų pasiūlyti anomalijų aptikimo metodai yra paremti LSTM (Du ir kt., 2017; Meng, Liu, Zhu, ir kt., 2019), Bi-LSTM (Zhang ir kt., 2019), CNN (Lu ir kt., 2018) modeliais. Esami DL paremti žurnalų anomalijų aptikimo metodai demonstruoja daug žadančius rezultatus analizuojant viešus žurnalinių įrašų duomenų rinkinius (He ir kt., 2020) ir tvirtina, kad jie pranašesni už tradicinius ML paremtus metodus.

2.5.3.1. Mašininiu mokymu paremtas anomalijų aptikimas

Mašininiu mokymu paremtas anomalijų aptikimas yra skirstomas į 2 grupes – prižiūrimas ir neprižiūrimas (He ir kt., 2016). Prižiūrimas mokymasis (pvz. SVM) apibrėžiamas kaip mašininio mokymosi uždavinys, kurio tikslas – sukurti modelį iš suklasifikuotų mokymo duomenų. Sužymėti mokymo duomenys, kurie pagal klases nurodo normalią arba anomalią būsenas yra būtina sąlyga prižiūrimam anomalijų aptikimui. Skirtingai nuo prižiūrimų metodų, neprižiūrimas mokymasis yra kitas įprastas mašininio mokymosi uždavinys, tačiau jo mokymo duomenys yra nesuklasifikuoti. Neprižiūrimi metodai labiau pritaikomi realioje gamybinėje aplinkoje, nes duomenys įprastai nebūna sužymėti. Įprasti

neprižiūrimi metodai apima įvairius klasterizavimo metodus. Šiame poskyryje apžvelgiami keli anomalijų aptikimo, analizuojant žurnalinius įrašus, metodai, kurie yra paremti mašininiu mokymu.

SVM

SVM (angl. *Support Vector Machine*) yra prižiūrimas algoritmas klasifikavimo ir regresijos problemoms spręsti ir yra labai populiarus dėl savo gebėjimo klasifikuoti su ribomis tarp klasių. Tai vektorinės erdvės modelis, kuris randa sprendimo ribą tarp dviejų klasių, esančių kuo toliau nuo duomenų taškų. Duomenų taškai, esantys arti hiperplokštumos skiriančios klases, vadinami atraminiais vektoriais (Liang ir kt., 2007).

Šis algoritmas dažnai naudojamas atliekant anomalijų aptikimo tyrimus. Viename iš tyrimų SVM naudojamas anomalijoms žurnalo sekoje aptikti (Liang ir kt., 2007). Autorių pasiūlytame anomalijų aptikimų metode žurnaliniu įrašų vektoriai yra sudaromi skaičiuojant kiek kartų įvykiai, su tam tikra svarbos žyma (angl. *log level*), pasikartoja žurnalų sekoje ar slenkančiame lange. SVM algoritmas yra apmokomas sužymėtais žurnalų vektoriais. Atliekant anomalijų aptikimą, jei aptikta žurnalo seka yra virš hiperplokštumos, ji yra laikoma anomalija. Šis metodas turi tam tikrų trūkumų, pavyzdžiui, negali susidoroti su naujais žurnalinių įrašų šablonais.

Žurnalinių įrašų klasterizavimas

Klasterizavimo metodai naudojami anomalijų aptikimo modelyje „LogCluster“ (Lin ir kt., 2016). Pasiūlytame modelyje žurnaliniai įrašai skirstomi į 2 klasterius – normalių įvykių ir anomalių. Šis modelis yra neprižiūrimas. Klasterizavimui yra naudojamas aglomeracinio hierarchinio klasterizavimo (angl. *Agglomerative Hierarchical clustering*) algoritmas. Aglomeracinis klasterizavimas yra labiausiai paplitęs hierarchinio klasterizavimo tipas, naudojamas objektams grupuoti į klasterius pagal jų panašumą. Jis taip pat žinomas kaip „AGNES“ (angl. *Agglomerative Nesting*). Klasterizavimo pradžioje, kiekvienas objektas traktuojamas kaip pavienis klasteris. Paskui klasterių poros nuosekliai jungiamos, kol visi klasteriai sujungiami į vieną didelį klasterį, kuriame yra visi objektai. Rezultatas – medžiu pagrįstas objektų atvaizdavimas, vadinamas dendrograma (Kassambara, s.a.).

„LogCluster“ modelyje atliekami 2 mokymo etapai – žinių bazės inicijavimo ir mokymosi realiu laiku (angl. *online*) etapai. Žinių bazės inicijavimo etapą sudaro trys žingsniai: žurnalų vektorizavimas, žurnalų klasterizavimas ir reprezentatyvių vektorių išskyrimas. Pirmiausia žurnalinių įvykiu sekos vektorizuojamos kaip įvykių skaičiaus vektoriai, kurie toliau tikslinami taikant „IDF“ dažnius ir normalizavimą. Antra, „LogCluster“ klasterizuoja normalius ir anomalius įvykių skaičiaus vektorius atskirai, naudodamas aglomeracinio hierarchinio klasterizavimo metodą, kuris sukuria du vektorių klasterių

rinkinius (normalius ir anomalius) kaip žinių bazę. Galiausiai kiekvienam klasteriui parenkamas reprezentatyvus vektorius, apskaičiuojant jo centroidą.

Mokymosi realiu laiku etapas reikalingas tam, kad toliau pakoreguoti žinių bazės inicializavimo etape sudarytus klasterius. Mokymosi realiu laiku etape, į žinių bazę įvykių skaičiaus vektoriai įtraukiami po vieną. Turint įvykių skaičiaus vektorių, apskaičiuojami atstumai tarp jo ir esamų reprezentatyvių vektorių. Jei mažiausias atstumas yra mažesnis už ribą, įvykių skaičiaus vektorius pridedamas prie artimiausio klasterio ir atnaujinamas šio klasterio reprezentacinis vektorius. Priešingu atveju sukuriamas naujas klasteris, naudojant šį įvykių skaičiaus vektorių. Sukūrus žinių bazę ir užbaigus mokymosi realiu laiku procesą, „LogCluster“ yra paruoštas anomalijoms aptikti. Norint nustatyti naujos žurnalų sekos būseną, apskaičiuojamas jos atstumas iki žinių bazėje esančių reprezentatyvių vektorių. Jei mažiausias atstumas yra didesnis už slenkstinę ribą, žurnalo seka laikoma anomalia.

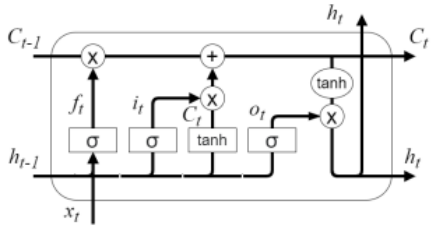
2.5.3.2. Giliuoju mokymu paremtas anomalijų aptikimas

Naujausiuose tyrėjų pasiūlytuose, anomalijų aptikimo iš žurnalinių įrašų, modeliuose yra naudojami giliojo mokymo metodai, tokie kaip, LSTM, Bi-LSTM ar CNN. Šiuose modeliuose, anomalijos žurnaluose aptinkamos analizuojant žurnalų sekas. Šie metodai, taip pat gali būti skirstomi 2 į grupes – prognozavimu pagrįsti metodai (anomalijos aptinkamos prognozuojant kitą žurnalo įvykio šabloną atsižvelgiant į ankstesnes žurnalo sekas), pavyzdžiui, „DeepLog“ (Du ir kt., 2017) ir „LogAnomaly“ (Meng, Liu, Zhu, ir kt., 2019). Kita grupė – klasifikavimu pagrįsti modeliai (anomalijos aptinkamos klasifikuojant žurnalų sekas), pavyzdžiui, „LogRobust“ (Zhang ir kt., 2019) ir CNN (Lu ir kt., 2018) metodai. Šiame poskyryje apžvelgiami keli populiarūs giliuoju mokymu paremti anomalijų aptikimo metodai.

LSTM

LSTM (angl. *Long Short-Term Memory*) yra gerai žinomas rekurentinis neuroninis tinklas (RNN), kuris dažnai naudojamas gilaus mokymosi modeliams kurti (Olah, 2015). Tinklo pasikartojanti dalis suteikia galimybę modeliui efektyviai apdoroti duomenų sekas. Tai labai naudinga apdorojant tekstą, vaizdo įrašus, prognozuojant laiko eilutes ir pan.. Paprastas RNN, kuris savo svoriams atnaujinti naudoja grįžtamąjį srautą (angl. *back propagation*), susiduria su nykstančių gradientų problema (angl. *vanishing gradient problem*), kaip ir įprasti tiesiniai gilieji neuroniniai tinklai. Sekančio žodžio nustatymo uždavinyje, RNN nustato sekantį žodį, kuris priklauso tik nuo ankstesnių duomenų taškų, tačiau praktikoje pastebima, kad jis nesugeba to padaryti, kai kontekstas pateikiamas ankščiau. LSTM sprendžia tai geriau, nes geriau įsimena ilgalaikes priklausomybes. Kiekvieną LSTM bloką sudaro kelios dalys, kurios apibrėžia kaip duomenys sklinda per LSTM ląstelę, kaip pavaizduota 2.12 pav.: įvesties vartai (i_t, C_t), išvesties vartai (o_t), užmiršties

vartai (f_t). Šios keturios ląstelės kartu sudaro ląstelės atmintį ir reguliuoja vidinę būseną. Užmiršties vartai kontroliuoja kokią informaciją reikia išmesti iš ląstelės būsenos. Įvesties vartai kontroliuoja kokias reikšmes reikia atnaujinti. Išėjimo vartai kontroliuoja kokias ląstelės būsenos dalys yra praleidžiamos.



2.12 pav. LSTM veikimas (Olah, 2015)

Anomalijoms aptikti, analizuojant žurnalinius įrašus, yra pasiūlyti keli LSTM rekurentiniu neuroniniu tinklu paremti modeliai – „DeepLog“ (Du ir kt., 2017) ir „LogAnomaly“ (Meng, Liu, Zhu, ir kt., 2019). Abu anomalijų aptikimo metodai yra paremti įvykių šablonų prognozavimu. Įvykių šablonų prognozavimo metoduose žurnalo įvykių seka yra traktuojama kaip natūralios kalbos seka, o neprižiūrimas anomalijų aptikimo uždavinys tampa įvykio šablono prognozavimo uždaviniu. Šie metodai paverčia žurnalinių įrašų sekas į žurnalo įvykių šablonų sekas ir tada prognozuoja kito žurnalo įvykio šablono klasę. Atrenkamas nustatytas kiekis žurnalo įvykių šablonų su didžiausia prognozavimo tikimybe ir iš jų sudaromas normalių žurnalo įvykių šablonų rinkinys, o iš likusių – anomalių šablonų rinkinys. Galiausiai galima nustatyti, ar žurnalo seka yra anomali, įvertinus rinkinio klasę, kuriai priklauso tikrasis žurnalinio įrašo šablonas (Zhang ir kt., 2021).

„DeepLog“ modelyje žurnalų šablonams išskirti naudojamas „Spell“ metodas. Šablonų statinė dalis yra prilyginama įvykiu raktams ir iš jų sudaromi sekų vektoriai, kurie yra naudojami sudėtinio LSTM (angl. *Stacked LSTM*) modelio apmokymui. Apmokius modelį normalių žurnalų raktų sekų požymiais, modelis geba prognozuoti sekantį žurnalo šabloną esamoje žurnalų sekoje, pagal tai nustatomas ar sekantis žurnalo šablonas yra anomalus ar ne. Kadangi naudojamas LSTM algoritmas, anomalijų aptikimas sekoje yra tikslesnis, nes modelio pagalba galima nustatyti semantinius ryšius tarp iš eilės einančių žurnalo įrašų. Tačiau naudojant vienkartinį kodavimą (angl. *one-hot encoding*) žurnalo šablonams paversti į skaitinę formą, iš dalies prarandama informacija apie labai svarbius semantinius ryšius tarp žurnalo šablonų (Ryciak ir kt., 2022).

„LogAnomaly“ modelis naudoja žurnalo įvykių skaičiaus vektorius kaip įvesties duomenis LSTM modeliui apmokyti. Autoriai, taip pat pasiūlė „template2vec“, sinonimais ir antonimais paremtą metodą, skirtą žurnalų šablonus paversti į semantinius vektorius, kad naujus žurnalinius įrašus būtų galima suderinti

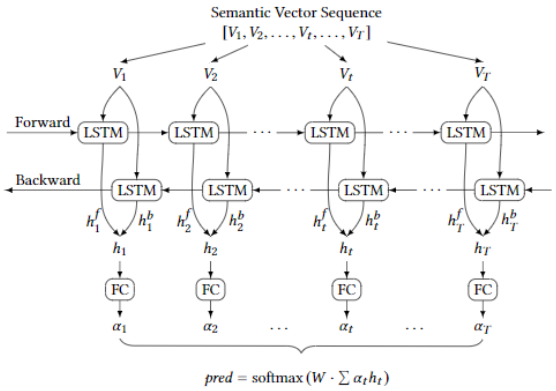
su esamais šablonais. Kaip ir „DeepLog“, „LogAnomaly“ yra prognozavimu pagrįstas anomalijų aptikimo modelis skirtas kitam žurnalo įvykiui prognozuoti. Jei analizuojamas žurnalo įvykis prieštarauja prognozavimo rezultatams, jis yra pažymimas kaip anomalija. Šio metodo pagrindiniai trūkumai – modelis prastai veikia su sudėtingais duomenų rinkiniais, bei yra jautrus žurnalinių įrašų struktūrizavimo klaidoms (Le & Zhang, 2022).

Bi-LSTM

Dvikryptis LSTM (angl. *Bidirectional LSTM*) yra LSTM modelio tipas, kuris apdoroja duomenis abejomis kryptimis – tiek pirmyn, tiek atgal (Verma, 2021). Dėl šios duomenų srauto abejomis kryptimis savybės, BI-LSTM išsiskiria iš kitų LSTM. Naudojant įprastą LSTM, įvesties srautą galima nukreipti tik viena kryptimi – pirmyn arba atgal. Tačiau dvikryptėje struktūroje įvesties srautas gali sklisti abejomis kryptimis, kad būtų išsaugota ir ateities, ir praeities informacija. Pavyzdžiui, sakinyje "berniukai eina į ....." negalime užpildyti paskutinio žodžio. Kai turime ateities sakinį "berniukai grįžta iš mokyklos", galime lengvai nuspėti praeities sakinio reikiamą žodį, dvikryptis LSTM leidžia neuroniniam tinklui tai atlikti. BI-LSTM paprastai naudojamas tais atvejais, kai reikia atlikti sekos į seką (angl. *sequence to sequence*) užduotis. Tokio tipo tinklas dažnai naudojamas teksto klasifikavimui, kalbos atpažinimui ir prognozavimo modeliams.

„LogRobust“ metode anomalijoms iš žurnalinių įrašų sekų aptikti naudojamas Bi-LSTM modelis su papildomu dėmesio sluoksniu (Zhang ir kt., 2019). Šis metodas naudoja iš anksto apmokytą „FastText“ modelį, kad konvertuotų žurnalų šablonus į semantinius vektorius, o tai užtikrina, kad atitinkami žurnalų šablonų atnaujinimai nesukels didelių semantinių vektorių pokyčių. Tai padidina anomalijų aptikimo patikimumą. Dėmesiu paremtas Bi-LSTM modelis turi galimybę fiksuoti kontekstinę informaciją žurnalo sekose ir automatiškai išmokti skirtingų žurnalo įvykių svarbą. Tokiu būdu „LogRobust“ gali nustatyti ir apdoroti nestabilius žurnalo įvykius ir sekas. Modelio architektūra pavaizduota 2.13 pav..

Kadangi skirtingi žurnalo įvykiai turi skirtingą poveikį klasifikavimo rezultatui, Bi-LSTM modelio dėmesio mechanizmas naudojamas siekiant priskirti skirtingus svorius žurnalo įvykiams. Taip galima sumažinti žurnalo įvykių triukšmo poveikį, nes triukšmingi žurnalo įvykiai paprastai yra mažiau svarbūs ir jiems dažniau skiriamas mažas dėmesys. „LogRobust“ anomalijų aptikimo modelyje svarba išmokstama automatiškai iš dėmesio sluoksnio – pilnai sujungtas (FC) sluoksnis pridedamas kaip dėmesio sluoksnis prie sujungtos paslėptos būsenos (h_t), o jo išvestis yra dėmesio svoris (α), kuris atspindi žurnalo įvykio svarbą. Kuo didesnė dėmesio reikšmė, tuo daugiau dėmesio modelis skiria šiam žurnalo įvykiui (Zhang ir kt., 2019).



2.13 pav. LogRobust anomalijų aptikimo modelis (Zhang ir kt., 2019)

LogRobust gali gerai veikti su žurnalo struktūrizavimo klaidų sukeltu triukšmu, nes naudoja dėmesiu paremtą Bi-LSTM modelį, kuris geba fiksuoti kontekstinę žurnalo sekų informaciją. Tačiau, modelio apmokymas yra prižiūrimas, dėl to mokymo etape reikia ir normalių, ir anomalių duomenų, o tai reikalauja daug rankinio darbo duomenims sužymėti. Kitas „LogRobust“ trūkumas yra tas, kad jį gali stipriai paveikti netinkamai sužymėtų žurnalų triukšmas (Le & Zhang, 2022).

CNN

Konvoliucinis neuroninis tinklas (CNN) priskiriamas prie prižiūrimo mokymo algoritmų ir yra vienas iš kompiuterinės vizijos srityje plačiai taikomų algoritmų. CNN yra tiesinis neuroninis tinklas, naudojamas tokiose užduotyse kaip vaizdų analizė, natūralios kalbos apdorojimas, bei kitoms sudėtingoms vaizdų klasifikavimo problemoms spręsti. Jis unikalus tuo, kad gali atrinkti ir aptikti vaizdų bei teksto šablonus ir suteikti jiems prasmę. CNN architektūra susideda iš trijų pagrindinių komponentų – konvoliucinio, sutelkiamojo ir pilnai sujungto sluoksnių. Konvoliucinis sluoksnis atlieka tiesinę operaciją, perkeldamas tam tikro dydžio filtrą (branduolį) per ankstesnio sluoksnio išvestį. Netiesinė „ReLU“ funkcija yra pagrindinė aktyvacijos funkcija, plačiai naudojama su CNN, siekiant padidinti netiesiškumo laipsnį ir visas neigiamas reikšmes duomenų požymių žemėlapyje pakeisti į 0. CNN gali būti daugiau nei vienas konvoliucinis sluoksnis, tačiau pirmasis konvoliucinis sluoksnis dažniausiai naudojamas paprastiems požymiams, pavyzdžiui, elemento kraštams ir kampams, išgauti, o kiti sluoksniai naudojami sudėtingesniems elementų požymiams išgauti. Sutelkimo sluoksnis naudojamas požymių matmenims sumažinti, todėl jo dėka yra sumažinamos skaičiavimo sąnaudos. Paskutinis, pilnai sujungtas sluoksnis naudojamas klasifikavimo tikslams (IBM Cloud Education, 2020).

Konvoliuciniai neuroniniai tinklai yra naudojami anomalijoms aptikti analizuojant žurnalinius įrašus. Pasiūlytame anomalijų aptikimo modelyje (Lu ir kt., 2018), neapdorotiems žurnalo duomenims taikomas dviejų etapų žurnalinių įrašų struktūrizavimas, kad būtų gautas žurnalinio įrašo raktas ir raktų sekų

vektorius pagal vykdymo tvarką. Žurnalų sekos paverčiamos semantiniais vektoriais naudojant iš anksto apmokytą „word2vec“ modelį, tada transformuojamos į mokymui tinkamą 29x128 dydžio matricą. CNN modelis mokomas naudojant šią matricą kaip įvesties duomenis, kad galėtų klasifikuoti žurnalo sekas į normalias ir anomalias. CNN modelis turi tris konvoliucinius sluoksnius su skirtingais filtrais ir maksimalaus sutelkimo sluoksnį požymiams išskirti. Filtro dydis yra lygus žurnalo įrašų grupės ilgiui, didžiausioms reikšmėms išrinkti taikomas „max-overtime“ sutelkimas. Skaičiavimams naudojami keli konvoliuciniai sluoksniai, tikimybių pasiskirstymo rezultatams gauti naudojamas visiškai sujungtas „softmax“ sluoksnis.

CNN modelis gali išgauti daugiau sąsajų žurnalo duomenų kontekste, kadangi yra naudojami keli filtrai. Konvoliucijos operacija leidžia modeliui užfiksuoti ryšį ne tik tarp žurnalo šablonų, bet ir ryšį žurnalo šablonų semantinio įterpimo kontekste. CNN modelis yra prižiūrimas ir gali pasiekti aukštą tikslumą, atlikus viešų žurnalinių įrašų duomenų rinkinių analizę. Tačiau, kaip ir „LogRobust“, CNN reikalingas didelis kiekis sužymėtų duomenų, kurių praktikoje dažniausiai nėra. CNN taip pat praranda tikslumą, kai susiduriama su labai nesubalansuotais duomenimis ir duomenų triukšmu (Le & Zhang, 2022).

2.6. Skyriaus apibendrinimas

Atlikus susijusios literatūros analizę, buvo apžvelgtos skirtingų tipų įsibrovimo aptikimo sistemos, pagal jų veikimo vietą. Buvo išsiaiškinta kad 2 pagrindiniai tipai yra tinklo pagrindu veikiančios ir kompiuterio pagrindu veikiančios IDS. Kiekvieno tipo IDS atliekamos funkcijos yra panašios ir tai yra puikus įrankis galintis padėti apsisaugoti nuo įsibrovimų, tačiau tiek vienos, tiek kitos turi tam tikrų trūkumų, ir nei viena iš jų neturi ryškaus pranašumo aptinkant visų tipų atakas. NIDS gali analizuoti tik tinklo duomenis, todėl įsibrovimo aptikimui, paremtam žurnalinių įrašų analize yra tinkamos tik HIDS.

Apžvelgus įsibrovimo aptikimo metodų tipus ir naudojamas technikas, nustatyta, kad pagrindiniai 2 tipai yra parašais paremti arba anomalijomis, o technikos – statistika, mašininiu mokymu, giliuoju mokymu paremtos. Parašais paremtas įsibrovimo aptikimas, nors ir atakas gali nustatyti labai tiksliai, bet turi didelių trūkumų – negali atpažinti naujų, modifikuotų ir slaptų atakų, todėl anomalijomis paremti metodai yra veiksmingesni, tačiau kad toks įsibrovimo aptikimo modelis būtų patikimas yra reikalinga taikyti tam tinkamas technikas, kurios visos stipriai priklauso nuo turimų duomenų kokybės.

Atlikus HIDS naudojamų duomenų rinkinių apžvalgą, nustatyta, kad kokybiškų duomenų rinkinių neprieinamumas yra tai kas stabdo kompiuterio lygmens įsibrovimo aptikimo sistemų vystymąsi. Duomenų rinkinius palyginus, nustatyta, kad duomenų rinkinių, kurie būtų orientuoti į „Windows“ OS ir būtų laisvai

prieinami beveik nėra, o dauguma duomenų rinkinių nėra pakankamai pilni – juose nėra įvykdytas pakankamas kiekis atakų ar juos sudaro tik siaura dalis kompiuterio informacijos, pvz. tik sisteminių funkcijų kvietimo pėdsakai. Taip pat nei viename duomenų rinkinyje nėra žurnalinių įrašų informacijos apie „Windows“ registrų, failų vientisumo pasikeitimus, todėl tokį duomenų rinkinį būtų pravartu sudaryti ir išanalizuoti.

Apžvelgus kitų mokslininkų atliktus tyrimus analizuojant kompiuterio lygmens žurnalinius įrašus įsibrovimams aptikti, nustatyta, kad dauguma mokslininkų sėkmingai taiko mašininiu mokymu paremtus įsibrovimo aptikimo metodus ir pasiekia aukštų rezultatų, tačiau rezultatai stipriai priklauso nuo naudojamų duomenų paruošimo būdų ir pačių duomenų kokybės. Gauti aukšti rezultatai parodo, kad žurnaliniai įrašai yra pakankamai geras duomenų šaltinis įsibrovimų nustatymui ir gali būti naudojamas įsibrovimo aptikimo sistemose. Taip pat pastebėta, kad aptinkant įsibrovimus geriausi rezultatai pasiekiami taikant DL metodus, todėl siūlant įsibrovimų aptikimo metodą paremtą žurnalinių įrašų analize, jame turėtų būti naudojami DL algoritmai.

Pasiremiant analitinėje dalyje gautomis išvadomis, suformuluojami uždaviniai, kurie toliau atliekami šiame darbe:

1. Atliekamas įsibrovimų aptikimo tyrimas analizuojant AWSCTD duomenų rinkinio OSSEC žurnalinius įrašus.
2. Tyrimo eigoje atliekamas žurnalinių įrašų struktūrizavimas, vektorizavimas ir anomalijoms aptikti apmokomi ML ir DL modeliai.
3. Duomenų vektorizavimo etape, tekstiniai žurnalinių įrašų duomenys paverčiami į skaitinę formą naudojant NLP metodus.
4. Atliekamas eksperimentas su apdorotais ir suvektorizuotais duomenimis, apmokant ML ir DL įsibrovimo aptikimo modelius, bei juos išbandant.
5. Pateikiamos analizės ir gautų rezultatų išvados.

3. Siūlomo metodo aprašymas

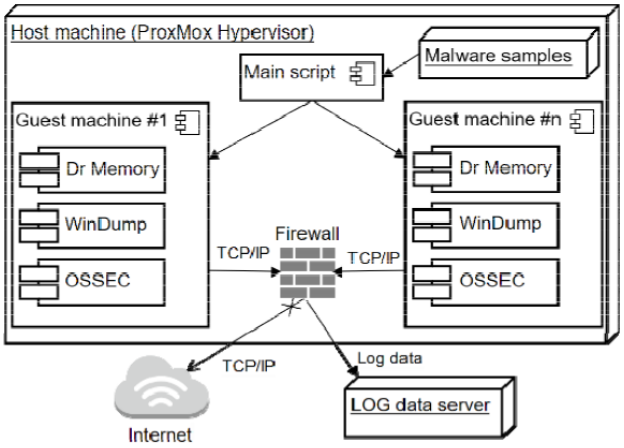
Siūlomame įsibrovimų aptikimų, analizuojant žurnalinius įrašus, metode, žurnalinių įrašų analizė yra išskiriama į 3 etapus – neapdorotų įrašų pavertimo į vieningą struktūrą, struktūrizuotų įrašų pavertimo į vektorius taikant NLP metodus, bei ML ir DL modelių, įsibrovimams aptikti, apmokymo ir testavimo. Visi šie etapai, bei naudojami duomenys detaliai aprašomi šiame skyriuje.

3.1. Duomenys

Šiame darbe atliekant žurnalinių įrašų analizę, su tikslu nustatyti įsibrovimus, naudojami HIDS žurnaliniai įrašai iš AWSCTD duomenų rinkinio. Šiame poskyryje plačiau aprašomas naudojamas duomenų rinkinys – naudojami metodai duomenų surinkimui, bei duomenų rinkinio turinys. Taip pat apžvelgiami pagrindiniai OSSEC HIDS komponentai, kurių pagalba buvo surinkti žurnalų duomenys.

3.1.1. AWSCTD duomenų rinkinys

AWSCTD (angl. *Attack-Caused Windows OS System Calls Traces Dataset*) yra kompiuterio lygmens įvykių, tokių kaip, sisteminių funkcijų kvietimo pėdsakų, failų vientisumo pasikeitimų, tinklo informacijos, kurie sistemoje užfiksuoti vykdant kenksmingo programinio kodo bandinius (KPK), duomenų rinkinys (Čeponis & Goranin, 2018). Renkant AWSCTD duomenis, KPK bandiniai yra vykdomi virtualioje mašinoje su „Windows“ 7 OS, „ProxMox“ hipervizoriuje. Kenkėjiškiems kompiuterio lygmens duomenims surinkti naudojamas duomenų rinkimo metodas, kurį sudaro 6 žingsniai – KPK bandinių paruošimas, hipervizoriaus paruošimas, virtualios mašinos paruošimas, duomenų surinkimo serverio paruošimas, KPK bandinių vykdymas ir duomenų surinkimas, surinktų duomenų paruošimas ir analizė. Virtualios mašinos paruošimo, KPK vykdymo ir duomenų surinkimo žingsniai yra pilnai automatizuojami „bash“ skripto pagalba. Duomenims surinkti yra naudojami tik viešai prieinama ir atviro kodo programinė įranga. „Windows“ sisteminių funkcijų kvietimų pėdsakams surinkti naudojamas „Dr. Memory“, failų ir registrų vientisumui stebėti, sistemos žurnaliniams įrašams – OSSEC (angl. *Open Source HIDS Security*), tinklo duomenims – „WinDump“. KPK vykdymo komponentų schema, renkant AWSCTD duomenis, vaizduojama 3.1 pav..



3.1 pav. KPK vykdymo ir duomenų rinkimo komponentų schema (Čeponis & Goranin, 2018)

Siekiant sumažinti priklausomybę nuo trečiųjų šalių programų (pvz. „office“ programų rinkinių, naršyklių ir kt.), buvo naudojamos tik vykdomojo tipo programos. Duomenims surinkti, kenkėjiškų programų bandiniai yra paimti iš viešai prieinamų KPK talpyklų – „VirusShare“ ir „theZoo“. Kiekvienas KPK bandinys įvykdomas naujoje virtualioje mašinoje, KPK paleidžiant automatiškai, kai VM užkraunama operacinė sistema. Paleidus KPK, vykdymo skriptas yra sustabdomas nustatytam laiko tarpui (AWSCTD duomenų surinkimo atveju – 30 minučių), kad KPK pilnai atskleistų savo kenkėjišką veiklą ir būtų surenkami atitinkami duomenys. Surinkti anomalijų duomenys (modifikuotų/atidarytų failų sąrašas, sisteminių funkcijų kvietimai ir susijusi informacija, tinklo duomenys) yra siunčiami į centralizuotą žurnalų duomenų serverį tolimesnei analizei atlikti. AWSCTD duomenimis surinkti buvo įvykdyta 10820 KPK bandinių, tačiau ne visi bandiniai, pagal „VirusTotal“ buvo kenkėjiški, todėl į anomalius duomenis buvo įtraukti tik tų bandinių duomenys, kurių „VirusTotal“ aptikimo rezultatas yra aukštesnis nei 15. AWSCTD duomenų rinkinyje yra didelis kiekis, skirtingų kategorijų KPK, sugeneruotų įvykių. KPK tipų, šeimų ir bandinių skaičius yra pateikiamas 3.2 pav..

Category	Count	Family	Count
AdWare	5139	AdWare.Win32	4655
Trojan	2353	Trojan.Win32	2326
Downloader	853	Downloader.Win32	830
WebToolbar	659	WebToolbar.Win32	654
DangerousObject	137	AdWare.MSIL	412
Trojan-Ransom	101	DangerousObject.Multi	137
Backdoor	79	Trojan-Ransom.Win32	96
RiskTool	55	Backdoor.Win32	75
Trojan-Downloader	45	AdWare.NSIS	71
Trojan-Spy	37	RiskTool.Win32	46
Packed	34	Trojan-Downloader.Win32	44
Virus	17	Packed.Win32	34
Trojan-PSW	16	Trojan-Spy.Win32	34
Trojan-Dropper	15	Downloader.NSIS	19
Trojan-Clicker	5	Trojan.MSIL	18

3.2 pav. Dažniausios kenkėjiškų programų kategorijos ir šeimos AWSCTD duomenų rinkinyje (Čeponis & Goranin, 2018)

KPK kategorijų ir šeimų informacija yra paremta „Kaspersky“ gamintojo duomenimis iš „VirusTotal“ duomenų bazės. Bandiniams, kuriems nepavyko priskirti tikslios kenkėjiškos programos kategorijos, „Kaspersky“ gamintojas priskiria „DangerousObject“ kategoriją.

Iš surinktų duomenų, ankstesniuose tyrimuose panaudoti buvo tik sisteminių funkcijų kvietimo pėdsakų duomenys. Kiekvieno bandinio sugeneruoti įvykiai, taip pat buvo papildyti „VirusTotal“ duomenimis, tokiais kaip, skenavimo rezultatai, KPK elgsenos informacija (veiksmai failų sistemoje, komunikacija tinkle, naudojamos bibliotekos) ir pan.. Tačiau surinkti OSSEC žurnalai, kuriuose užfiksuota informacija apie failų vientisumo pasikeitimus, „Windows“ registrų vientisumo pasikeitimus, bei įspėjimus, sugeneruotus iš reikšmingų „Windows“ sistemos žurnalinių įrašų, kenkėjiškos veiklos vykdymo metu, nebuvo panaudoti atliekant įsibrovimo aptikimo tyrimus.

3.1.2. OSSEC HIDS

Renkant duomenis AWSCTD duomenų rinkiniui, kompiuterio lygmens įvykių, tokių kaip žurnalinių įrašų, duomenims išgauti, kai buvo vykdoma anomali veikla, buvo naudojamas OSSEC HIDS įrankis. OSSEC yra nemokamas ir atviro kodo HIDS, kuris atlieka sistemos žurnalinių įrašų analizę, failų vientisumo patikrinimus, „Windows“ registrų stebėjimą, aptinka „rootkits“ pažeidžiamumus. Aptikus tam tikrus, įtartinus įvykius, OSSEC apie juos sugeneruoja įspėjimus administratoriui. OSSEC kenkėjiškos veiklos aptikimas yra paremtas taisyklėmis, kuriose yra apibrėžti kenkėjiškos veiklos šablonai. OSSEC veikia „Windows“, „Linux“, „OS X“, „FreeBSD“ ir kt. operacinėse sistemose ir turi centralizuotą valdymą, kuris suteikia galimybę lengvai stebėti ir valdyti daugybę sistemų (About OSSEC HIDS, s.a.). OSSEC pagrindiniai funkcionalumai, kurie buvo naudojami renkant AWSCTD duomenis, yra šie – žurnalinių įrašų analizė, failų vientisumo stebėjimas, „Windows“ registrų stebėjimas.

Failų vientisumo stebėsena

Nepakeitus sistemos failų sukompromituoti sistemą yra sudėtinga. Taip pat failų sistemoje yra konfidencialios ir privačios informacijos. Dėl šių priežasčių failai yra patrauklus kibernetinių nusikaltėlių taikinys. Šiuolaikinėse operacinėse sistemose naudojamos prieigos kontrolės politikos, kuriomis nustatoma kuris naudotojas gali pasiekti, skaityti ar redaguoti tam tikrą failą. Naudojant failų vientisumo stebėseną, HIDS gali stebėti kas ir kokius duomenis pasiekia, ir nustatyti, ar nėra netinkamos prieigos ar pažeidimų.

Failų vientisumo stebėjimas leidžia HIDS aptikti failų pakeitimus. OSSEC, tikrindamos failų vientisumą, saugo failų kriptografines maišos vertes duomenų bazėje. OSSEC periodiškai perskaičiuoja failų kontrolines sumas ir palygina jas su istorinėmis vertėmis. Aptikus maišos verčių neatitikimą,

sugeneruojamas įspėjimas. Kad vientisumo tikrinimas būtų veiksmingas, prieš pradinį failų vientisumo tikrinimą reikia įsitikinti, kad sistema yra nepažeista. Priešingu atveju maišos reikšmės bus apskaičiuotos pagal pažeistus failus, todėl įsilaužimo nebus galima aptikti (OSSEC Syscheck, s.a.).

„Windows“ registrų stebėsena

„Windows“ registrai yra sistemos duomenų bazė, kurioje saugoma informacija, reikalinga sistemai paleisti ir konfigūruoti, sistemos programinės įrangos nustatymai, kuriais valdomas „Windows“ veikimas, ir kiekvieno vartotojo konfigūracijos nustatymai. OSSEC gali gauti vertingos informacijos apie sistemos ir naudotojo elgseną stebint registrų įrašus.

Stebėdamas registrus, OSSEC gali aptikti sistemos konfigūracijų, naudotojų grupių ir įdiegtos programinės įrangos pakeitimus. Naudojant iš „Windows“ registro gautą informaciją, taip pat galima aptikti saugumo politikos pažeidimus, pavyzdžiui, neleistiną programinės įrangos naudojimą arba techninės įrangos, pavyzdžiui, išorinių laikmenų prijungimą prie kompiuterio. Tai aptinkama, kadangi atliekant šiuos veiksmus sistemoje yra pakeičiami ar sukuriami nauji, tam tikri registrų įrašai, o tuos pasikeitimus pastebi HIDS.

Žurnalinių įrašų analizė

Daugelyje šiuolaikinių programų yra naudojami tam tikri žurnalų registravimo mechanizmai, kurie suteikia galimybę administratoriams analizuoti programų veikimą. Dauguma įvykusių įsilaužimų palieka tam tikrus pėdsakus žurnalų failuose. HIDS žurnalinių įrašų analizei gali būti taikomi parašais paremti metodai. Parašai nustatomi iš žurnalo įrašų, kurie yra susiję su žinomais netinkamo sistemos naudojimo atvejais arba įsilaužimais. Todėl, kai yra vykdoma ataka prieš sistemą, OSSEC analizuojant žurnalą tikrina žurnalinių įrašu šablonus su žinomais kenkėjiškos veiklos šablonais, parašų duomenų bazėje ir jei atakos parašas sutampa, apie tai yra pranešama administratoriui (OSSEC Log monitoring/analysis, s.a.).

Renkant duomenis AWSCTD duomenų rinkiniui, OSSEC sugeneruoti įspėjamieji pranešimai buvo įrašomi į įspėjimų žurnalą. Įspėjimų žurnalas yra prieinamas keliais formatais – paprasto teksto ir JSON, kuriame įspėjimai yra struktūrizuoti objektai. OSSEC įspėjimų žurnalo įrašų pavyzdžiai, paprastu tekstu, yra pateikiami 3.1 lentelėje.

3.1 lentelė. OSSEC įspėjamųjų pranešimų pavyzdžiai

```
** Alert 1522444235.6900: mail  - ossec,syscheck,
2018 Mar 31 00:10:35 (303) 172.20.0.113->syscheck
Rule: 550 (level 7) -> 'Integrity checksum changed.'
Integrity checksum changed for: 'C:\Windows/Tasks/SCHEDLGU.TXT'
```

```
** Alert 1522446964.44237: - ossec,syscheck,
2018 Mar 31 00:56:04 (303) 172.20.0.113->syscheck-registry
Rule: 594 (level 5) -> 'Registry Integrity Checksum Changed'
Integrity checksum changed for:
'HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LanmanServer\Parameters'

** Alert 1522460858.293619: mail  - syslog,errors,
2018 Mar 31 04:47:38 (303) 172.20.0.113->WinEvtLog
Rule: 1002 (level 2) -> 'Unknown problem somewhere in the system.'
2018 Mar 31 04:47:50 WinEvtLog: System: INFORMATION(7036): Service Control Manager:
(no user): no domain: prod-PC: The Windows Error Reporting Service service entered
the stopped state.
```

3.1.3. AWSCTD OSSEC žurnaliniai įrašai

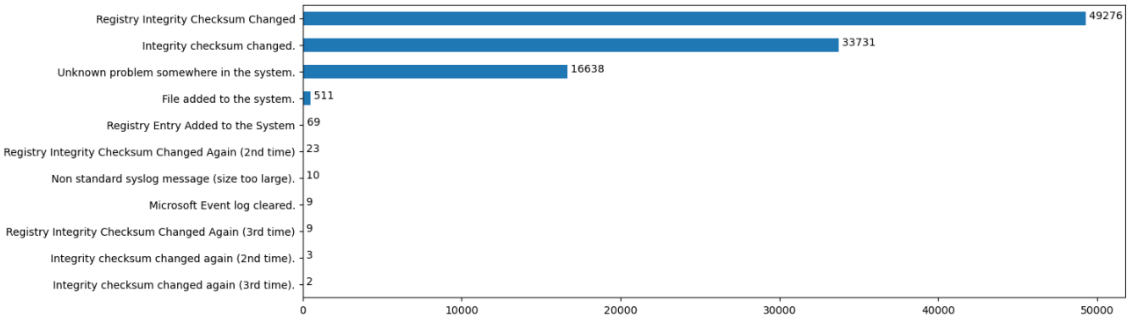
Šiame darbe aptinkant įsibrovimus naudojami OSSEC HIDS sugeneruotų pranešimų duomenys iš AWSCTD duomenų rinkinio. Tam, kad būtų galima tiksliai aptikti įsibrovimą naudojant tokius duomenis, reikalinga išsiaiškinti kokie įvykių duomenys (žurnaliniai įrašai) vyrauja duomenų rinkinyje ir kokią vertę pranešimai gali turėti nustatant ar paleistas vykdomasis failas yra kenkėjiškas. Analizuojant duomenis, pirmiausia pastebėta, kad duomenų rinkinyje egzistuoja pasikartojančių įrašų, kurie yra kiekviename bandinyje – tai pranešimai apie pridėtą vykdomojo failo bandinio failą, paleistą OSSEC agentą ir pan., iš viso yra 4 tokie pranešimai. Pasikartojantys pranešimai visuose bandiniuose vertės kenkėjiškų programų aptikimui neprideda, todėl yra pašalinami. Taip pat pastebėta, kad kiekvieno kompiuterio žurnalų faile pirmi bandinių įrašai turi labai didelį kiekį įspėjimų apie failų ir registrų vientisumo pasikeitimus – daugiau nei 50000, kai tuo tarpu kiti bandiniai vidutiniškai turi apie 10 pranešimų. OSSEC failų ir registrų vientisumo patikrinimas veikia tikrinant esamas failų ir registrų maišos reikšmes su reikšmėmis turimomis OSSEC vidinėje sistemos duomenų bazėje. Kadangi toks įrašų kiekis yra tik pirmuose bandiniuose, galima daryti prielaidą, kad tuo metu buvo inicializuojama OSSEC vidinė failų ir registrų maišų duomenų bazė, dėl to buvo sugeneruoti pranešimai. Tokie bandiniai, taip pat buvo pašalinti iš duomenų rinkinio. Išvalyto duomenų rinkinio įrašų statistika pateikta 3.2 lentelėje.

3.2 lentelė. Bandinių ir įvykių statistika

Bandiniai, įvykiai	Kiekis
Iš viso bandinių	9932
Kenkėjiškų bandinių	9780
Švarių bandinių	152
Iš viso įvykių	100281
Įvykiai kenkėjiškuose bandiniuose	98791
Įvykiai nekenkėjiškuose bandiniuose	1490
Iš viso skirtingų įvykių	11
Skirtingų įvykių kenkėjiškuose bandiniuose	11

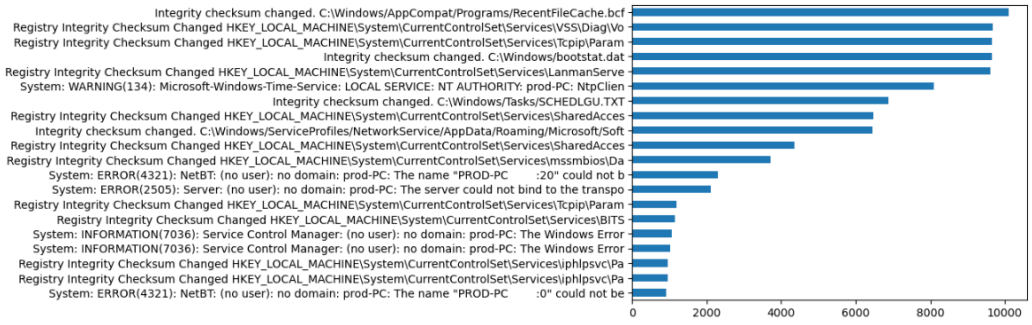
Skirtingų įvykių nekenkėjiškuose bandiniuose	6

Pagal duomenis bandinių ir įvykių statistikos lentelėje galima matyti, kad analizuojamuose AWSCTD duomenų rinkinio OSSEC žurnaluose yra mažas kiekis švarių bandinių duomenų – 152. Toks duomenų rinkinys yra labai nesubalansuotas 2 klasių (kenkėjiškų ir nekenkėjiškų) atžvilgiu, todėl apmokant ML modelį atpažinti įsibrovimus naudojant šiuos duomenis būtų būtina taikyti papildomus duomenų rinkinio subalansavimo metodus. Duomenų rinkinys turi didelį kiekį įvykių, kuriuos sugeneravo vykdomos programos, tačiau skirtingų įvykių kiekis yra gana mažas – 11. Diagramoje, 3.3 pav., vaizduojamas skirtingų įvykių duomenų rinkinyje pasiskirstymas.

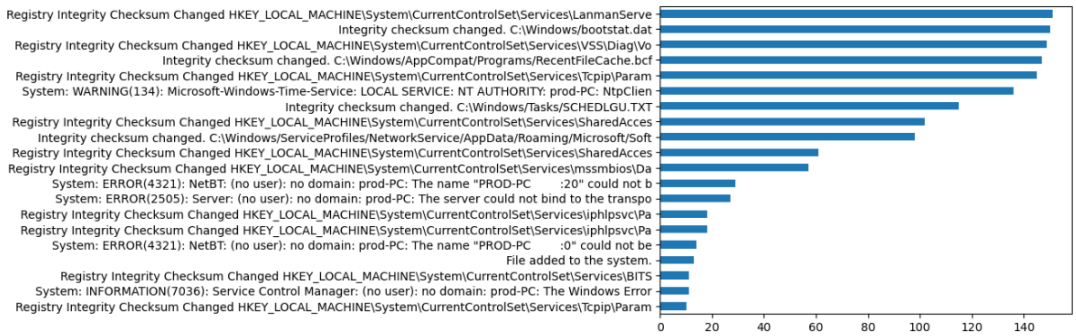


3.3 pav. Skirtingi įvykiai ir jų pasiskirstymas duomenų rinkinyje

Iš grafiko galima matyti, kad duomenų rinkinyje dominuoja 3 tipų žurnaliniai įrašai – įspėjimai apie registrų ir failų pasikeitimus, bei sistemoje užfiksuotas nežinomas problemas. „Unknown problem somewhere in the system“ įrašas reiškia, kad „Windows“ įvykių žurnale buvo sugeneruotas aukštos svarbos žurnalinis įrašas. Siekiant turėti didesnę įvykių įvairovę galima įtraukti įvykių kintamąsias dalis, tokias kaip, failo kelias failų sistemoje, registro kelias registrų medyje ar žurnalinis įrašas iš „Windows“ įvykių žurnalo. Tokiu atveju skirtingų įrašų kiekis yra didesnis. Diagramose, žemiau, vaizduojamas žurnalinių įrašų pasiskirstymas įtraukiant kintamąsias įspėjimų dalis, po 20 dažniausiai pasikartojančių pranešimų. 3.4 pav. vaizduojami pranešimai kenkėjiškiems bandiniams, 3.5 pav. – nekenkėjiškiems.



3.4 pav. Kenkėjiškų bandinių AWSCTD OSSEC dažniausi pranešimai su kintamosiomis žurnalo dalimis



3.5 pav. Švarių bandinių AWSCTD OSSEC dažniausi pranešimai su kintamosiomis žurnalo dalimis

Iš diagramų galima matyti, kad duomenų rinkinyje yra didelis kiekis pasikartojančių įvykių net įtraukus ir kintamąsias įvykių dalis. Taip pat matoma, kad dažnai pasikartojantys įvykiai nėra išskirtinai pasiskirstę kenkėjiškuose ar švariuose bandiniuose, jie yra tiek ten tiek ten, dėl to tai kiek komplikuoja įsibrovimų aptikimą naudojant tokius duomenis. Kita duomenų rinkinio problema yra ta, kad failų ir registrų pasikeitimai OSSEC „syscheck“ funkcijos yra fiksuojami kas tam tikrą laiko periodą (OSSEC Syscheck, s.a.), o ne realiu laiku, kada jie pasikeitė. Dėl to įvykių, kenkėjiškai veiklai aptikti, negalima analizuoti kaip įvykių sekos, nes tai neatspindi realios įvykių sekos.

Analizuojant AWSCTD duomenų rinkinio OSSEC pranešimų duomenis buvo įžvelgta, kad duomenų rinkinys turi tam tikrų trūkumų, kas apsunkina įsibrovimų aptikimą naudojant tokius duomenis – duomenų rinkinyje vyrauja pasikartojantys žurnaliniai įrašai, yra išskirtinių pranešimų trūkumas, įvykių eiliškumas neatspindi realios įvykių sekos, bei duomenų klasės yra nesubalansuotos. Dėl šių priežasčių buvo nuspręsta duomenų rinkinį papildyti pakartotinai simuliuojant tų pačių kenkėjiškų ir švarių programų elgseną, naudojant patobulintą HIDS konfigūraciją. Apie tai detaliai aprašoma sekančiame poskyryje.

3.2. Duomenų rinkinio papildymas

Duomenų rinkinio papildymo pagrindinis tikslas yra surinkti didesnę įvairovę turinčius duomenis, kurie galėtų geriau atspindėti kenkėjiškų ir švarių programų elgseną, kas leistų įsibrovimus aptikti tiksliau. Turint įvairesnius duomenis, taip pat galima tiksliau nustatyti ne tik įsibrovimą, bet ir atakos tipą. Pildant duomenų rinkinį siekiama tai atlikti kiek įmanoma panašiau, kaip tai buvo atliekama renkant duomenis originaliam duomenų rinkiniui. Programų elgsenai simuliuoti naudojamas tas pats kenkėjiškų ir švarių programų rinkinys, kuris buvo gautas iš originalaus duomenų rinkinio autorių. Iš viso buvo įvykdyta ir surinkta duomenų apie 12611 programų elgseną. Tai yra šiek tiek daugiau nei originalaus AWSCTD

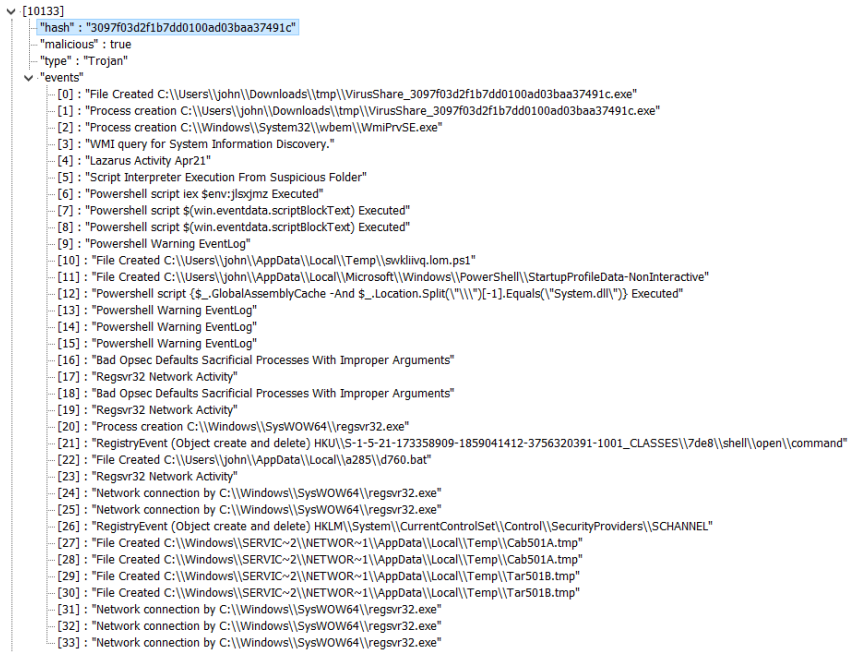
OSSEC žurnalinių įrašų rinkinyje, nes buvo įvykdytas didesnis kiekis nekenkėjiškų programų bandinių. Tokiu būdu sušvelninama pradinio duomenų rinkinio klasių nesubalansuotumo problema.

Programų veiklai stebėti ir įvykiams registruoti naudojama kita OSSEC HIDS variacija – „Wazuh“. Tai yra OSSEC projekto pagrindu išvystyta HIDS, kuri turi tam tikrų patobulinimų. „Wazuh“ yra nemokama ir atviro kodo saugumo stebėjimo, įvykių aptikimo ir atitikties saugumo reikalavimams valdymo platforma. Ji sukurta siekiant padėti organizacijoms aptikti kibernetines grėsmes ir reaguoti į jas realiu laiku. „Wazuh“ suteikia kibernetinių grėsmių aptikimo priemones apimančias įsibrovimo į kompiuterius aptikimą, žurnalinių įrašų analizę, pažeidžiamumų aptikimą ir elgsenos stebėjimą. Įrankis taip pat turi ir centralizuotą valdymo serverį, kuriame galima lengvai diegti, konfigūruoti ir stebėti agentus keliose sistemose (Getting started with Wazuh, s.a.). Šių įrankiu vienas iš pagrindinių skirtumų yra tas, kad „Wazuh“ turi papildomų funkcijų, kurių nėra OSSEC. Pavyzdžiui, „Wazuh“ turi atnaujintą žiniatinklio pagrindu veikiančią naudotojo sąsają, patikimesnę integraciją su kitais saugumo sprendimais, geriau išvystytą grėsmių informacijos kanalų palaikymą ir patobulintas duomenų dekodavimo priemones, bei saugumo grėsmių aptikimo taisykles, kurias aktyviai kuria ir palaiko bendruomenė. Išplėstas taisyklių rinkinys yra vienas pagrindinių privalumų siekiant surinkti didesnę duomenų įvairovę vykdant kenkėjiškas programas.

„Wazuh“ HIDS failų ir registrų vientisumo stebėjimo posistemė veikia tokiu pat būdu kaip ir OSSEC – kas tam tikrą laiko periodą tikrina esamas failų ir registrų maišos reikšmes su reikšmėmis turimomis vidinėje maišų duomenų bazėje. Jeigu užfiksuojami maišų pasikeitimai, apie tokį įvykį sugeneruojami įspėjamieji pranešimai. Renkant duomenis apie programų elgseną svarbu išlaikyti įvykių eiliškumą, nes tokius duomenis būtų galima pritaikyti DL algoritmams, kurie gali išmokti įvykiu sekas ir tiksliau aptikti įsibrovimus. Tačiau standartinis „Wazuh“ failų ir registrų vientisumo stebėjimas tam nėra tinkamas ir renkant duomenis nėra naudojamas. Vietoje to į HIDS integruojamas papildomas saugumo mechanizmas „Sysmon“ – tai yra „Microsoft“ sukurta „Windows“ OS stebėjimui skirta tarnyba, kuri suteikia pažangias stebėjimo galimybes. „Sysmon“ renka ir registruoja įvairius sistemos įvykius ir veiksmus, įskaitant procesų kūrimą, tinklo sujungimus, failų kūrimą ir šalinimą, registro pakeitimus ir kt.. Vienas iš pagrindinių „Sysmon“ privalumų yra tai, kad įrankis suteikia daug išsamesnį sistemos veikimo vaizdą nei tradiciniai „Windows“ įvykių žurnalai (Russinovich & Garnier, 2023). Taip pat šio įrankio užfiksuoti įvykiai, tokie kaip, failų, registrų sukūrimas, pakeitimas ar pašalinimas yra realiu laiku perduodami į „Wazuh“ ir iš jų sugeneruojami atitinkami pranešimai. Tokiu būdu yra išlaikoma reali įvykių seka susijusi su failų ir registrų pasikeitimų įvykiais.

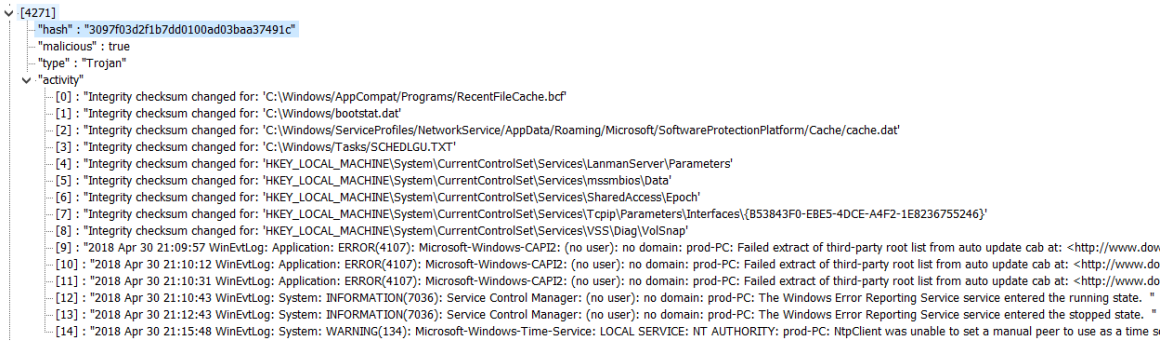
Kaip ir originaliame AWSCTD duomenų rinkinyje, papildant duomenų rinkinį kenkėjiškos ir švarios programos yra vykdomos virtualiose mašinose su „Windows“ 7 OS. Tam, kad kenkėjiškos programos pasileistų ir galėtų kiek įmanoma pilniau atskleisti savo elgseną, iš OS pusės yra atjungiami tam tikri saugumo mechanizmai, tokie kaip, antivirusinė sistema ir ugniasienė. Taip pat sistemose yra pakeičiami kai kurie „Windows“ žurnalinių įvykių registravimo nustatymai. Tam į pagalbą pasitelktas „LOG-MD“ įrankis, kuris suteikia galimybę atlikti žurnalų ir išplėstinės sistemos audito politikos nustatymų auditą ir padeda naudotojams sukonfigūruoti žurnalizavimo nustatymus pagal tam tikras gerąsias praktikas ar standartus, kad į „Windows“ žurnalą būtų registruojama daugiau sistemos saugumui reikšmingų įvykių (LOG-MD, s.a.). Konfigūruojant žurnalizavimo nustatymus atsižvelgiama į „Windows Logging Cheat Sheet“ konfigūravimo gaires (Windows Logging Cheat Sheet, s.a.). Tokiu būdu papildomai įjungiami „PowerShell“, „DNS“, „TaskScheduler“ ir kt. komponentų „Windows“ žurnalai, kas suteikia galimybę HIDS stebėti daugiau žurnalų ir atitinkamai sugeneruoti daugiau įvykių.

Renkant duomenis vykdomųjų programų bandiniai yra vykdomi ant 7 VM lygiagrečiai, kurios imituoja kompiuterius. Kiekvienoje mašinoje veikia „Wazuh“ agento programinė įranga, kuri atlieka kompiuterio žurnalinių įrašų stebėjimo funkciją ir tinklu perduoda duomenis į „Wazuh“ valdymo serverį, kuriame yra atliekama duomenų analizė ir generuojami įspėjamieji pranešimai. VM paleidimas, bandinių vykdymas ir VM atstatymas į pradinę būseną yra automatizuotas kelių tipų skriptų pagalba. VM paleidimui ir valdymui yra paleidžiamas „Shell“ skriptas, kuris kiekvienam bandiniui paleidžia naują, iš anksto paruoštą VM, „KVM“ hipervizoriuje, ir laukia tam tikrą laiko tarpą kol bandinys bus įvykdytas, o pasibaigus vykdymui VM atstato į pradinę būseną. Tuo tarpu VM OS pusėje automatiškai paleidžiamas „PowerShell“ skriptas, kuris į kompiuterį parsiunčia ir paleidžia dar nebandytą programos bandinį. Bandinio vykdymui laukiama 5 minutes. Praėjus nustatytam laikui mašina yra atstatoma į pradinę būseną ir vykdymas yra kartojamas su kitais bandiniais kol įvykdomi visi bandiniai. Taip iš viso įvykdyti 12611 programų bandiniai iš kurių 9633 kenkėjiški ir 2978 nekenkėjiški. Ar bandinys yra kenkėjiškas, tai nustatyta pagal aptikimų rezultatus „VirusTotal“ kenkėjiškų programų duomenų bazėje. Vieno įvykdyto kenkėjiško bandinio HIDS sugeneruoti įspėjamieji pranešimai su patobulinta HIDS konfigūracija pateikiami 3.6 pav..



3.6 pav. Programos bandinio duomenys papildytame duomenų rinkinyje

Vykdant šį bandinį sugeneruoti pranešimai apie sukurtus failus, procesus, registrų objektus, tinklo sujungimus, „PowerShell“ skriptų vykdymą ir kitus saugumui reikšmingus įvykius, kurie buvo įvykdyti veikiant kenkėjiškai programai. Palyginimui ar su nauja HIDS konfigūracija sugeneruoti pranešimai yra detalesni ir geriau atskleidžia programų elgseną galima paimti tam pačiam bandiniui sugeneruotus pranešimus iš originalaus AWSCTD duomenų rinkinio ir palyginti. To pačio bandinio pranešimai iš pradinio duomenų rinkinio vaizduojami 3.7 pav..



3.7 pav. Programos bandinio duomenys originaliame duomenų rinkinyje

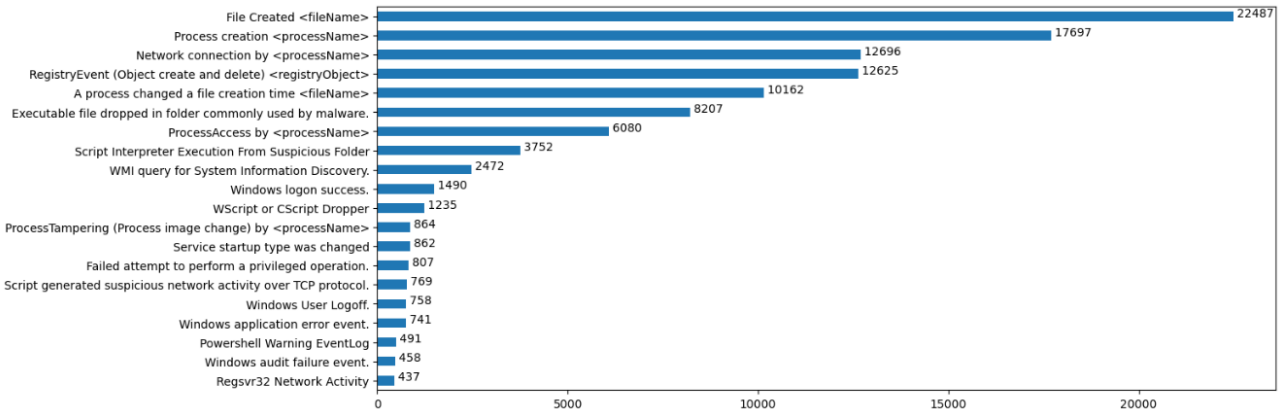
Lyginant tam pačiam bandiniui sugeneruotus pranešimus galima įžvelgti akivaizdžius skirtumus – pradiniuose duomenyse nebuvo užfiksuota, kad buvo sukurti tam tikri failai, procesai, buvo vykdomi skriptai, buvo atliekami veiksmai tinkle, vyksta kiti neįprasti veiksmai, kurie yra būdingi kenkėjiškoms programos. Tačiau pradiniuose duomenyse buvo užfiksuota daugiau įvykių susijusių su registrų pakeitimais,

bet pranešimai yra labai bendriniai ir neatskleidžia kenkėjiškos programos savybių. Patobulintos HIDS konfigūracijos atveju šių registrų pasikeitimai nebuvo užfiksuoti, nes būtent šie registrų objektai nėra stebimi, nes per dažnai kinta ir dėl to sukelia daug triukšmo, kas gali trukdyti pastebėti svarbius įvykius. Lyginant dviejų bandinių įvykius, taip pat galima matyti, kad įvykių įvairovė naujai surinktuose duomenyse yra didesnė, bet atsižvelgiant tik į vieną bandinį išvadų daryti negalima, todėl tai reikalinga patikrinti visų duomenų rinkinio bandinių atžvilgiu. 3.3 lentelėje pateikta papildyto duomenų rinkinio įvykių ir bandinių statistika.

3.3 lentelė. Papildyto duomenų rinkinio bandinių ir įvykių statistika

Bandiniai, įvykiai	Kiekis
Iš viso bandinių	12611
Kenkėjiškų bandinių	9633
Švarių bandinių	2978
Iš viso įvykių	108207
Įvykiai kenkėjiškuose bandiniuose	88741
Įvykiai nekenkėjiškuose bandiniuose	19466
Iš viso skirtingų įvykių	124
Skirtingų įvykių kenkėjiškuose bandiniuose	106
Skirtingų įvykių nekenkėjiškuose bandiniuose	60

Lyginant bandinių ir įvykių kiekius pradiniame ir papildytame duomenų rinkinyje galima matyti, kad surinktų įvykių kiekis abiejuose duomenų rinkiniuose yra gana panašus, bet skirtingų įvykių kiekis papildytame duomenų rinkinyje yra gerokai didesnis – daugiau nei 11 kartų, o tai buvo pagrindinis duomenų rinkinio papildymo tikslas. Dažniausiai 20 pasikartojančių skirtingų įvykių, kurie užfiksuoti vykdant kenkėjiškas ir nekenkėjiškas programas vaizduojami grafike, 3.8 pav..



3.8 pav. Dažniausiai pasikartojantys įvykiai papildytame duomenų rinkinyje

Iš grafiko galima matyti, kad duomenų rinkinyje dominuoja įvykiai apie failų, procesų sukūrimus, tinklo sujungimus, įvykius susijusius su registrų objektų kūrimu ir šalinimu. Papildytame duomenų rinkinyje, taip pat yra ne mažas kiekis kitų tipų HIDS sugeneruotų saugumo įvykių, kurie atskleidžia kenkėjiškų programų savybes. Tokie duomenys yra detalesni ir geriau atskleidžia vykdomų programų elgseną, todėl naudojant tokius duomenis, apmokius ML ar DL modelį atpažinti įsibrovimus, būtų galima tiksliau aptikti naujus, ankščiau nematytus įsibrovimus, bei tiksliau nustatyti atakos tipus.

3.3. Duomenų paruošimas

Pildant duomenų rinkinį, programų bandiniai buvo vykdomi keliose virtualiose mašinose lygiagrečiai, o duomenys siunčiami į centrinį serverį. „Wazuh“ žurnalų failą sudaro duomenys apie visų mašinų įvykius ir jie nėra sugrupuoti pagal bandinius. Tam, kad duomenis būtų galima panaudoti kituose tyrimo etapuose, yra reikalinga juos tinkamai paruošti. Duomenų paruošimas atliekamas keliais žingsniais:

1. Žurnalo failas išskaidomas į atskirus žurnalo failus, kuriuose yra tik vieno kompiuterio įvykiai.
2. Žurnaliniai įrašai sugrupuojami pagal vykdomojo failo bandinį – kiekvienam vykdomajam failui priskiriami įspėjamieji pranešimai, kuriuos sugeneravo HIDS, kai bandinys buvo paleistas ir veikė sistemoje.
3. Pagal vykdomojo failo bandinį sugrupuoti duomenys iš skirtingų kompiuterių apjungiami į vieną žurnalo failą.
4. Vykdomas įvykių apdorojimas:
    a. Iš eilės pasikartojantys įvykiai sutrumpinami;
    b. Įvykių skaičius kiekviename bandinyje sutrumpinamas;
    c. Pašalinami spec. simboliai ir žodžiai trumpesni nei 2 raidės.
5. Vykdomųjų failų bandiniams priskiriamos klasifikavimo žymos.
6. Iš bandinio sugeneruotų įvykių sukuriami tekstiniai dokumentai, kiekvienam bandiniui po dokumentą.

Pirmame žingsnyje, duomenys, pagal virtualios mašinos identifikatorių yra išskiriami į atskirus mašinos žurnalų failus. Kadangi renkant programų elgsenos duomenis bandiniai buvo vykdomi ant 7 mašinų lygiagrečiai, po žurnalų išskaidymo yra 7 atskiri kiekvienos mašinos žurnalo failai. Antrame žingsnyje, kiekvienos mašinos žurnalo failo bandiniai yra grupuojami į bandinio objektus, kuriuos sudaro bandinio maišos funkcijos reikšmė, sugeneruoti pranešimai, bandinio klasė (įprasta ar anomali), KPK tipas. Unikalus

bandinys identifikuojamas pagal failo maišos funkcijos reikšmę. Darbui su „Wazuh“ duomenimis naudojamas JSON žurnalinių įrašų formatas, o bandinių objektų sudarymui, taip pat naudojama JSON struktūra. Struktūrizuoto bandinio objekto pavyzdys pateikiamas 3.6 pav..

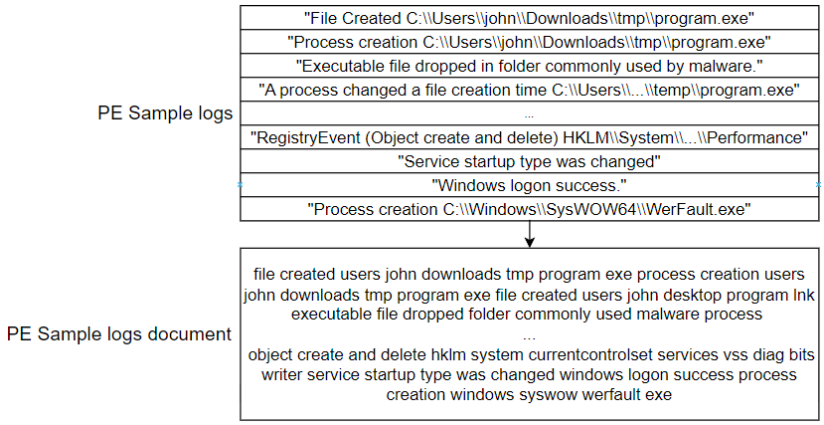
Duomenų paruošimas toliau tęsiamas apjungiant visus bandinius iš skirtingų kompiuterių į vieną failą, bei apdorojant įvykius. Įvykiai, kurie iš eilės pasikartoja daugiau nei 3 kartus yra sutrumpinami iki 3 įvykių, nes didelis pasikartojančių įrašų kiekis ML modeliui gali trukdyti įžvelgti kitas svarbias detales, bei didesnis duomenų kiekis prailgintų mokymo laiką. Toks duomenų optimizavimo būdas taikomas įsibrovimo aptikimo analizuojant sisteminių funkcijų kvietimus tyrimuose ir padeda pasiekti geresnių klasifikavimo rezultatų (Yesir & Soğukpinar, 2021). Toliau, sugeneruotų įvykių skaičius kiekviename bandinyje yra sutrumpinamas iki 50 įvykių, pašalinant įvykius, kurie yra už 50-ojo įvykio, siekiant sumažinti duomenų apimtis ir sutrumpinti dokumentų vektorių generavimo laiką. Galiausiai iš įvykių aprašymo pašalinami specialūs simboliai ir žodžiai trumpesni nei 2 raidės, kas yra dažnai naudojami tekstų paruošimo metodai NLP srityje. Duomenų paruošimas tęsiamas bandiniams priskiriant klases – švarus ar kenkėjiškas, bei jeigu failas yra kenkėjiškas, priskiriamas KPK tipas. Klasės priskyrimas automatizuojamas „VirusTotal“ API pagalba – kiekvieno bandinio maišos reikšmė yra patikrinama „VirusTotal“ kenkėjiškų programų duomenų bazėje. Jeigu aptikimo rezultatas yra 0, bandiniui priskiriama klasė „švarus“, jeigu rezultatas aukštesnis nei 30 – „kenkėjiškas“. KPK tipas priskiriamas kenkėjiškiems failams pagal „Kaspersky“ gamintojo failo skenavimo rezultatą – „AdWare“, „Downloader“, „Trojan“, „WebToolbar“, „DangerousObject“, „Backdoor“, „RiskTool“ arba „NoType“ jeigu „Kaspersky“ gamintojas KPK tipo nepateikė. 3.4 lentelėje pateiktas skirtingų bandinių tipų kiekis bei pasiskirstymas duomenų rinkinyje.

3.4 lentelė. KPK tipų pasiskirstymas duomenų rinkinyje

Klasė	Kiekis	Santykis (%)
AdWare	5369	42.57
Trojan	2724	21.60
Downloader	692	5.49
WebToolbar	581	4.61
NoType	141	1.12
Backdoor	67	0.53
RiskTool	31	0.25
DangerousObject	28	0.22
Benign	2978	23.61
Viso	12611	

Sužymėjus duomenis, toliau bandiniams sukuriami dokumentai, kuriuose yra atvaizduojami programos elgseną atspindintys HIDS įspėjamieji pranešimai ir kurie vėliau perteikti į skaitinę formą yra

naudojami klasifikavimui. Įsibrovimų aptikimo uždavinį, naudojant tokius duomenis, galima spręsti kaip prižiūrimo mokymo klasifikavimo uždavinį, kuriame ML klasifikavimo modelis yra apmokomas atpažinti NLP metodais užkoduotus tekstus ar dokumentus, kuriuos sudaro žodžiai ir sakiniai iš žurnalinių įrašų, taip pat įtraukiant ir kintamąsias žurnalo dalis, pavyzdžiui failų ar registrų kelius failų sistemoje ar registrų medyje, motyvuojant tuo, kad kenkėjiškos programos yra linkusios naudoti tam tikrus „Windows“ registrus ar įrašinėti failus tam tikrose direktorijose, skirtingai nei nekenkėjiškos programos. Šie duomenys gali būti perteikiami naudojant natūralios kalbos apdorojimo metodus.



3.9 pav. HIDS pranešimų atvaizdavimas dokumente

Vieno vykdomojo failo bandinio HIDS sugeneruotų pranešimų pavertimo į dokumentą pavyzdys vaizduojamas 3.9 pav.. Toliau šie dokumentai NLP metodais perteikiami į skaitinę formą, kad būtų galima apmokyti ML ir DL modelius klasifikuoti kenkėjiškus ir švarius dokumentus, bei nustatyti kenkėjiškos programos tipą, kurie pagal HIDS sugeneruotus pranešimus atspindi programų elgseną.

3.4. Duomenų vektorizavimas

Atliekant literatūros analizę buvo nustatyta, kad žurnaliniams įrašams vektorizuoti, įsibrovimo aptikimo tyrimuose, yra sėkmingai taikomi NLP metodai, tokie kaip „word2vec“ ar „FastText“, kadangi žurnaliniuose įrašuose dominuoja tekstiniai duomenys. Tačiau skirtingai nuo apžvelgtų tyrimų, kur vektorizuojant žurnalinius įrašus didžiausias dėmesys buvo skiriamas statinei žurnalinių įrašų daliai ir įrašų sekoms, šiame tyrime analizuojant žurnalinius įrašus dėmesys skiriamas žurnaliniams įrašams su kintamosiomis jų dalimis (failų, procesų, registrų keliais ir kt.), todėl vektorizuojant duomenis svarbu parinkti tinkamą NLP metodą. Kadangi žodžių sąrašas tokiuose įrašuose nėra baigtinis ir nauji žodžiai algoritmui gali būti nežinomi, todėl viena iš sąlygų, kurią vektorizavimo metodas turi tenkinti tai gebėti tvarkytis su nežinomais žodžiais. Kita svarbi sąlyga – perteikti dokumento sakinių semantinę informaciją,

kas turėtų leisti tiksliau aptikti kenkėjišką veiklą. Šiame poskyryje apžvelgiami „word2vec“, „FastText“, bei BERT NLP modeliai ir jie palyginami, su tikslu parinkti geriausiai tinkantį modelį sprendžiamai problemai.

„Word2vec“ – tai natūralios kalbos apdorojimo metodas, kuris sukuria vektorinį žodžio atvaizdavimą, pagrįstą kontekstu, kuriame žodis pasitaiko teksto rinkinyje. Šis metodas naudoja neuroninį tinklą, kad išmoktų žodžių vektorines reprezentacijas, kurias vėliau galima naudoti įvairioms natūralios kalbos užduotims, pavyzdžiui, žodžių ar frazių panašumui nustatyti ir tekstui klasifikuoti. „Word2vec“ modelis yra apmokomas naudojant didelį teksto duomenų rinkinį, o jo architektūra pagrįsta negiliu neuroniniu tinklu.

Yra dvi pagrindinės „word2vec“ architektūros: nepertraukiamo žodžių maišo (angl. *Continuous Bag-of-Words, CBOW*) ir „skip-gram“. CBOW modelis yra apmokytas nuspėti tikslinį žodį pagal konteksto žodžius, t. y. žodžius, esančius aplink tikslinį žodį sakinyje. Modelio įvestis yra užkoduotas konteksto žodžių atvaizdavimas, „one-hot“ kodavimo būdu, o išvestis – tikslinio žodžio tikimybė, atsižvelgiant į konteksto žodžius. Tuo tarpu „skip-gram“ modelis yra apmokytas nuspėti konteksto žodžius pagal tikslinį žodį. Modelio įvestis yra tikslinio žodžio „one-hot“ atvaizdas, o išvestis – konteksto žodžių tikimybė, atsižvelgiant į tikslinį žodį. Apmokytame „word2vec“ modelyje išvesties sluoksnio svoriai naudojami kaip žodžių vektoriniai atvaizdai (Mikolov, Chen, ir kt., 2013). ”Word2vec“ galima naudoti vektorizuojant ne tik žodžius, bet ir sakinius ar dokumentus. Dokumentui užkoduoti, pirmiausia dokumentas suskirstomas į žodžius ir tada kiekvienas žodis pateikiamas apmokytam „word2vec“ modeliui, žodžių vektoriniams atvaizdams gauti. Tuomet kiekvieno žodžio vektoriai gali būti sumuojami arba skaičiuojamas jų vidurkis, kad būtų gautas bendras viso dokumento vektorius.

„FastText“ – tai „Facebook AI Research“ sukurta nemokama, lengva ir atviro kodo teksto atvaizdavimo ir klasifikavimo biblioteka. Tai „word2vec“ „skip-gram“ architektūra pagrįstas modelis, kuris realizuotas negilaus neuroninio tinklo pagrindu. Modelis gali būti naudojamas kelioms natūralios kalbos apdorojimo užduotims, pavyzdžiui, teksto klasifikavimui, žodžių atvaizdavimui ir teksto panašumo nustatymui. Skirtingai nei „word2vec“, „FastText“ modelis apdoroja ne pilnus žodžius, bet žodžio dalis (angl. *subword*) arba simbolių „n-gram“ rinkinius ir skaičiuoja jų vektorių vidurkį arba sumą, taip sudarydamas vieno žodžio vektorinį atvaizdą. Tai leidžia apdoroti žodžius, kurių nėra žodyne, ir morfologines žodžių variacijas, kas yra dažna natūralios kalbos apdorojimo problema (Bojanowski ir kt., 2016). „Word2vec“ gali gerai perteikti žodžio reikšmę, tačiau gali sunkiai suprasti kontekstą ir ryšius tarp

žodžių ar tvarkytis su nežinomais žodžiais. „FastText“ sprendžia šias problemas apdorojant ne visą žodį, o jų dalis, kas leidžia tvarkytis su žodžiais, kurie nėra įtraukti į žodyną, kas taip pat yra ir šio darbo problema.

BERT – dvikryptis transformacinio kodavimo būdas (angl. *Bidirectional Encoder Representations from Transformers*) yra iš anksto apmokytas gilaus mokymosi modelis, skirtas NLP problemoms spręsti. Jį galima pritaikyti įvairioms NLP užduotims, pavyzdžiui, teksto klasifikavimui, klausimų atsakymui, įvardytų esybių atpažinimui ir kalbos supratimui. BERT sukuria kiekvieno sakinio žetono (angl. *token*) vektorinį atvaizdavimą, užkoduodamas žetono reikšmę, atsižvelgdamas į kontekstą ir ryšį tarp sakinio žodžių. Šie vektoriniai atvaizdai vadinami kontekstualizuotais žodžių įterpiniais, kurie skiriasi nuo tradicinių žodžių įterpinių, tokių kaip „word2vec“ ir „FastText“, kurie koduoja žodžio reikšmę tik pagal kontekstą, kuriame jis pasirodo mokomajame žodyne. Kontekstualizuoti žodžių įterpiniai, sukurti naudojant BERT, laikomi pranašesniais už tradicinius žodžių įterpinius, nes jie gali atspindėti žodžio reikšmę įvairiuose kontekstuose, kas leidžia efektyviau spręsti NLP uždavinius (Devlin ir kt., 2018).

BERT veikimas paremtas transformatoriaus (angl. *transformer*) architektūra. Šis neuroninio tinklo architektūros tipas dažnai naudojamas sprendžiant NLP uždavinius. Įvesties žetonų kontekstiniams vektoriams apskaičiuoti transformatorių architektūroje naudojami dėmesio į save (angl. *self-attention*) mechanizmai, vietoje tradicinėse neuroninių tinklų architektūrose naudojamų rekurentinių ar konvoliucinių sluoksnių. Įprastą transformatoriaus architektūrą apima 2 mechanizmai – enkoderiai, kurie skaito įvesties žetonus ir juos užkoduoja atsižvelgiant į kontekstą, bei dekoderiai, kurie iš enkoderių sugeneruotų vektorių sprendžiamam uždaviniui sukuria atitinkamus išvesties žetonus, pvz., sprendžiant teksto vertimo uždavinį išveda išverstą žodį. Kadangi BERT tikslas sudaryti tik kalbos modelį, dekoderio mechanizmas jame nėra naudojamas. Kontekstinių žodžių atvaizdų mokymuisi BERT modelyje naudojamas išankstinio apmokymo metodas, kuris susideda iš dviejų etapų:

1. Užmaskuotos kalbos modelio (angl. *Masked Language Model, MLM*) išankstinis apmokymas. MLM išankstinio apmokymo tikslas – išmokti kontekstines žodžių reprezentacijas nuspėjant užmaskuotus žodžius sakinyje. Atliekant išankstinį MLM mokymą, BERT įvestyje gauna žodžių seką, kurioje kai kurie iš jų atsitiktinai pakeičiami specialiu maskavimo žetonu, o modelis mokomas nuspėti originalų žodį pagal jo kontekstą. Atliekant šią užduotį, BERT išmokomas kurti kontekstinius žodžių įterpinius, kurie atspindi žodžio reikšmę pagal jį supantį kontekstą.
2. Sekančio sakinio nuspėjimo (angl. *Next Sentence Prediction, NSP*) išankstinis apmokymas. Šiame etape modelis mokomas atlikti sekančio sakinio prognozavimo užduotį, kurioje jis mokosi prognozuoti, ar du sakiniai dokumente eina vienas paskui kitą. Mokydamasis atlikti NSP užduotį,

BERT išmoksta suprasti dviejų sakinių ryšį ir gali kurti įterpinius, kurie atspindi bendrą dokumento kontekstą.

Iš anksto apmokytą BERT galima pritaikyti atlikti tolimesnius NLP uždavinius, pavyzdžiui, teksto klasifikavimą, klausimų atsakymą, įvardytų esybių atpažinimą ir kt., prie iš anksto apmokyto BERT modelio prijungiant konkrečiai užduočiai atlikti skirtą išvesties sluoksnį ir visą modelį galutinai apmokant atlikti atitinkamą užduotį. 3.5 lentelėje aprašomi apžvelgtų vektorizavimo modelių privalumai ir trūkumai.

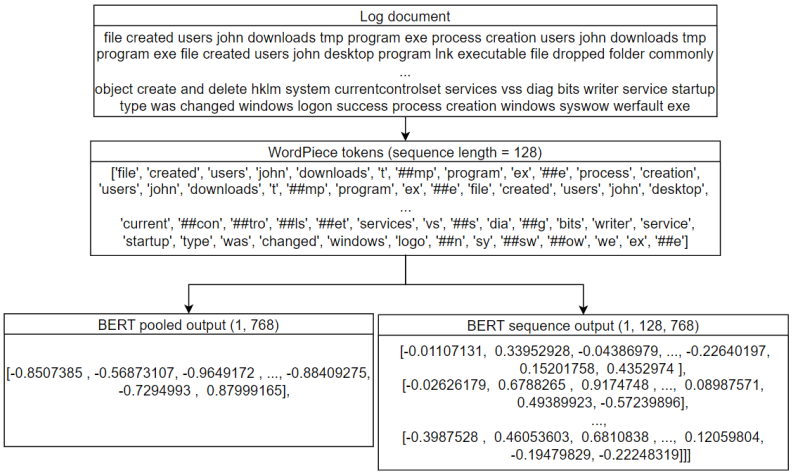
3.5 lentelė. NLP modelių privalumai ir trūkumai

Modelis	Privalumai	Trūkumai
Word2vec	Modelio paprastumas – yra gana paprastas modelis, kurį lengva suprasti ir įgyvendinti. Pagrįstas negilaus neuroninio tinklo architektūra, kuri skaičiavimo požiūriu yra pigesnė. Greičiau apmokomas. Turi mažiau parametrų. Vektorius galima aiškiau interpretuoti.	Modelis negeba tvarkytis su nežinomais žodžiais. Modelis negali atvaizduoti žodžių, jei skiriasi galūnė ar priesaga, jei tokių žodžių nėra žodyne. Negali perteikti konteksto.
FastText	Atvaizduoja nežinomus žodžius – modelis analizuoja žodžius išskaidant juos į dalis (simbolių „n-gram“ rinkinius), kad sukurtų žodžių vektorinius atvaizdus. Nereikalauja daug skaičiavimo resursų. Egzistuoja didelis kiekis paruoštų modelių tam tikroms dalykinėms sritims ar kalboms.	Negali perteikti konteksto. Ribotas modelio panaudojamumas – efektyvus vektorizuojant ir klesifikuojant tekstus, tačiau neefektyvus sprendžiant kitus NLP uždavinius. Sudėtingai optimizuojamas.
BERT	Gali perteikti žodžių kontekstą sakinyje. Veiksmingas atliekant užduotis, kai reikia suprasti sakinio prasmę ir kontekstą, kuriame jis pateikiamas. Gerai tvarkosi su nežinomais žodžiais nes apdorojant žodžius skaido juos į smulkesnius žodžius, net iki simbolių kol atitikmuo randamas žodyne. Egzistuoja didelis kiekis paruoštų modelių įvairioms kalboms ar dalykinėms sritims. Platus panaudojamumas – gali būti naudojamas įvairioms NLP problemoms spręsti.	Suderinti ir paleisti modelį reikalinga daug skaičiavimo resursų. Ne visada aiškiai interpretuojami jo sprendimai. Sudėtinga modelio architektūra. Didelė priklausomybė nuo iš anksto apmokyto modelio. Jei užduotis yra labai specifinė arba jei nėra iš anksto parengto modelio konkrečiai kalbai ar sričiai, modelį apmokyti yra nepraktiška.

Naudojant tradicinius žodžių įterpimus galima paprastai ir efektyviai gauti kiekvieno žurnalinio įrašo žodžio vektorius, bet tokiu būdu nėra atsižvelgiama nei į žodžių tarpusavio sąsajas žurnalo įvykyje, nei į žodžių tvarką, todėl negalima visiškai perteikti semantinės žurnalo įvykio informacijos. Taikant BERT modelį vektorizuojant tekstinius duomenis atsižvelgiama į semantinį ir žodžių eiliškumo ryšį su kiekvienu žodžiu žurnalo įraše ir pagal tai koduojamas visas dokumentas. Taip geriau perteikiamas kenkėjiško dokumento kontekstas, kas turėtų padėti padidinti įsibrovimų aptikimo tikslumą (Zhang ir kt., 2022). Taip pat, BERT modelis gali atvaizduoti nežinomus žodžius, todėl dėl šių privalumų pasirenkama duomenis vektorizuoti su BERT modeliu.

Duomenų paruošimo etape sudarytų vykdomųjų failų veiklos dokumentams vektorizuoti naudojamas iš anksto apmokytas, „TensorFlow Hub“ platformoje viešai prieinamas BERT modelis „bert-en-uncased“ (BERT, s.a.). Tai viena iš specifinių BERT versijų, turinti 12 sluoksnių (transformatorių blokų), 12 dėmesio galvų ir 110 milijonų parametrų. Modelis yra apmokytas tekstais iš anglų kalbos „Vikipedijos“ versijos ir „BooksCorpus“ duomenų rinkinio, bei nėra jautrus didžiosioms raidėms, t. y. didžiąsias ir mažąsias raides traktuoja vienodai (Devlin ir kt., 2018). BERT modelis išvestyje pateikia kelių tipų vektorinius tekstų atvaizdus – apjungtą vektorių (angl. *pooled output*) ar sekos vektorius (angl. *sequence output*). Modelis, taip pat gali pateikti ir kitų tipų išvestis, tačiau jos nėra naudojamos atliekant tyrimus, dėl to nėra apžvelgiamos. Apjungtas vektorius tai fiksuoto dydžio, 768 dimensijų vektorius, apibendrinantis visą įvesties seką. Jis sudaromas vykdant BERT modelio sekos išvesties sutelkimo (angl. *pooling*) operaciją (įprastai vidurkio arba maksimalų sutelkimą). Apjungtas vektorius naudingas atliekant užduotis, kurioms reikalingas vienas fiksuoto dydžio, visos įvesties sekos atvaizdavimas, pavyzdžiui, teksto klasifikavimą ar sentimentų analizę. Sekos vektoriai tai kintamo ilgio vektorių seka, kurioje kiekvienas vektorius yra kontekstinis įvesties sekos žetono (žodžio ar žodžio dalies) atvaizdavimas 768 dimensijų vektoriaus pavidalu. Sekos ilgis yra nustatomas BERT modelio parametras, kuris pagal nutylėjimą yra 128, taigi tokiu atveju modelio sekos vektorių išvestis yra 128 vektoriai, kurių kiekvieno dydis yra 768 dimensijų. Sekos vektoriai gali būti naudojami užduotims, kurioms reikalingas išsamus įvesties sekos supratimas, pavyzdžiui, įvardytoms esybėms atpažinti, tekstui generuoti ar klausimams atsakyti ir pan..

Vektorizuojant duomenis modelio įvesčiai yra paduodami kiekvieno bandinio dokumentai, atspindintys programų elgseną, kurie yra išskaidomi į žetonus naudojant „WordPiece“ žetonų generavimo algoritmą. Jis veikia skaidydamas žodžius į žodžių dalis ir kiekvieną žodžių dalį atvaizduodamas į unikalų žodyno identifikatorių (žetoną). Tai leidžia modeliui veiksmingai tvarkytis su žodžiais, kurie nėra įtraukti į žodyną, nežinomus žodžius susiejant su žodyne esančiais daliniais žodžiais. Modelio išvestis yra kelių tipų dokumento vektoriniai atvaizdai – apjungtas vektorius ir sekų vektoriai.



3.10 pav. Dokumento vektorizavimo pavyzdys

3.10 pav. vaizduojamas vieno dokumento vektorizavimo pavyzdys. Su dokumentų vektorių duomenimis, atpažinti įsibrovimus ir nustatyti jų tipą, toliau apmokomi prižiūrimo mokymo klasifikavimo modeliai.

3.5. Įsibrovimų aptikimų vertinimas

Įvertinti ar siūlomas dokumentų vektorizavimo metodas yra tinkamas ir ar naudojant tokius duomenis galima aptikti įsibrovimus, bei nustatyti jų tipus apmokomi ML ir DL algoritmai ir patikrinamas jų įsibrovimų aptikimo ir kenkėjiškų programų tipų nustatymo tikslumas. Atliekant literatūros analizę buvo nustatyta, kad įsibrovimams aptikti sėkmingai naudojami ML ir DL klasifikavimo algoritmai, tokie kaip atraminių vektorių mašinų (SVM), atsitiktinių miškų (RF), daugiasluoksniai perceptronai (MLP) ar konvoliuciniai neuroniniai tinklai (CNN), todėl šie metodai panaudojami atliekant siūlomo vektorizavimo būdo vertinimą.

SVM yra prižiūrimo mokymosi algoritmas naudojamas klasifikavimo ir regresijos užduotims spręsti. SVM klasifikatoriaus tikslas – rasti geriausią ribą (vadinamąją hiperplokštumą), pagal kurią duomenys būtų suskirstyti į skirtingas klases. Geriausia riba yra ta, kuri maksimizuoja maržą, t. y. atstumą tarp ribos ir artimiausių kiekvienos klasės duomenų taškų – atraminių vektorių. SVM gali atlikti tiek tiesinį, tiek netiesinį klasifikavimą, naudodamas įvairių tipų branduolius. Įsibrovimams aptikti algoritmas yra apmokomas naudojant sužymėtus duomenis, kurie atvaizduoja kenkėjiškos ir normalios veiklos šablonus. Tada prognozavimo modelis gali būti naudojamas naujiems, dar nematytiems duomenims klasifikuoti kaip normaliems arba kenkėjiškiems. SVM dažnai naudojamas įsibrovimų aptikimų tyrimuose dėl savo savybių apdoroti didelių dimensijų duomenis ir atsparumo triukšmui, bei nukrypimams (Liang ir kt., 2007).

RF yra ansamblinis klasifikavimo, regresijos ir kitoms užduotims spręsti skirtas mokymosi metodas, kuriame mokymo proceso metu sudaromi keli sprendimų medžiai ir išvedama klasė, kuri yra atskirų medžių klasių moda. Šis metodas padeda sumažinti per didelį prisitaikymą prie mokymo duomenų ir pagerinti modelio našumą. RF yra prižiūrimo mokymosi algoritmas, įsibrovimams aptikti jis yra apmokomas su sužymėtais duomenimis, kurie atspindi kenkėjišką ar normalią veiklas. Apmokytas modelis tada gali būti naudojamas atpažinti įsibrovimus iš naujų, dar nematytų duomenų. RF yra patikimas algoritmas, galintis apdoroti didelių matmenų duomenis, jis gali apdoroti trūkstamas reikšmes ir kategorinius kintamuosius. Jis taip pat atsparus triukšmui ir nukrypimams, todėl tinka įsilaužimams aptikti. RF algoritmas naudojamas daugelyje įsibrovimų aptikimų tyrimuose ir įrodė, kad yra veiksmingas nustatant įsilaužimus, bei pasižymi dideliu tikslumu ir mažu klaidingų teigiamų rezultatų skaičiumi (Kim ir kt., 2021).

MLP – tai dirbtinis neuroninis tinklas, sudarytas iš kelių tarpusavyje sujungtų neuronų sluoksnių, kurie apdoroja informaciją. Šie sluoksniai yra pilnai sujungti, t. y. kiekvienas vieno sluoksnio neuronas yra sujungtas su kiekvienu kito sluoksnio neuronu. Sprendžiant klasifikavimo uždavinius, MLP išvesties sluoksnyje paprastai yra tiek neuronų, kiek yra klasių, į kurias reikia suskirstyti, o kiekvienas neuronas atitinka vieną klasę. Išvesties sluoksnyje naudojama aktyvacijos funkcija paprastai yra „softmax“ funkcija, kuri leidžia tinklui išvesti visų klasių tikimybių pasiskirstymą. Įvesties duomenys perduodami per tinklą, o išvesties sluoksnio išvestis interpretuojama kaip tinklo prognozė apie įvesties duomenų klasę. MLP gerai apdoroja didelių dimensijų duomenis, todėl jis plačiai naudojamas vaizdams klasifikuoti, kalbai atpažinti ir natūraliai kalbai apdoroti. MLP, taip pat, gali būti naudojamas įsibrovimams aptikti atitinkamai apmokius modelį klasifikuoti kenkėjiškus ir normalius duomenis. Tačiau MLP įsibrovimo aptikimų tyrimuose nėra toks populiarus kaip kiti mašininio mokymosi algoritmai, tokie kaip SVM ar RF. Algoritmo našumas priklauso nuo konkretaus duomenų rinkinio ir problemos savybių bei kitų veiksnių, pavyzdžiui, duomenų kokybės, algoritmo parametrų ir taikomo vertinimo metodo (Ozkan-Okay ir kt., 2021).

CNN yra gilaus neuroninio tinklo tipas, kuris plačiai naudojamas vaizdų ir vaizdo įrašų atpažinimui, taip pat NLP ir kitoms užduotims spręsti. CNN modelis imituoja žmogaus smegenų vaizdinės informacijos apdorojimo procesą. Šiuos tinklus sudaro kelių tipų sluoksniai, tarp jų – konvoliuciniai, sutelkimo ir pilnai sujungti sluoksniai. Konvoliuciniai neuroniniai tinklai originaliai sukurti vaizdų apdorojimo užduotims atlikti, tačiau jų gebėjimas automatiškai išmokti hierarchines reprezentacijas iš neapdorotų įvesties duomenų leidžia juos sėkmingai taikyti sprendžiant įvairias kitas problemas, apdorojant ne tik vaizdų duomenis. Vienas iš būdų naudoti CNN su ne vaizdų duomenimis – duomenis apdoroti kaip vienmatį signalą, pvz., laiko eilutes arba matavimų seką. Tokiu atveju tinklo konvoliuciniai sluoksniai gali būti naudojami vietiniams signalo dėsningumams ir požymiams išmokti, o sutelkimo sluoksniai naudojami

požymių atvaizdavimo matmenims sumažinti. CNN yra svarbus tinklo gylis, nes tai leidžia tinklui geriau išmokti hierarchines įvesties duomenų reprezentacijas. Kiekviename konvoliuciniame sluoksnyje mokomasi iš ankstesnio sluoksnio išvesties išgauti vis abstraktesnius ir sudėtingesnius požymius. Tai ypač svarbu atliekant vaizdo apdorojimo užduotis, kai reikia identifikuoti tokius požymius kaip briaunos, kampai ir formos, kad būtų galima atpažinti aukštesnio lygio objektus. Be konvoliucinių sluoksnių, atsakingų už požymių išskyrimą, CNN taip pat apima sutelkimo sluoksnius, kurie sumažina požymių matmenis, ir visiškai sujungtus sluoksnius, kurie naudoja išmoktus požymius įvesties duomenims prognozuoti. Dėl gilios CNN architektūros jie gali išmokti sudėtingas įvesties duomenų reprezentacijas, o tai lėmė jų sėkmę įvairiose taikymo srityse, tarp jų ir įsibrovimų aptikime (Lu ir kt., 2018). Vienas iš privalumų naudojant CNN kompiuterio įsilaužimams aptikti yra jų gebėjimas išmokti sudėtingus duomenų dėsningumus, kuriuos gali būti sunku nustatyti taikant tradicinius, negilius ML algoritmus, todėl CNN gali geriau aptikti naujus ir anksčiau nematytus atakų tipus, ir gali geriau prisitaikyti prie besikeičiančių grėsmių.

Modelių apmokymams ir bandymams naudojama „Python“ programavimo kalba ir „TensorFlow“ 2.10 mašininio mokymo karkasas, bei „Keras“ 2.10 ir „scikit-learn“ 1.2.2 ML bibliotekos. „TensorFlow“ yra galingas ir lankstus mašininio mokymosi karkasas, kuris naudojamas giliesiems neuroniniams tinklams kurti ir mokyti. Įrankis suteikia žemo lygio skaičiavimo grafų API, leidžiantį vartotojams apibrėžti ir vykdyti sudėtingas matematines operacijas dideliuose duomenų rinkiniuose. Tuo tarpu „Keras“ yra aukštesnio lygio neuroninių tinklų API, kuriuo paprasta naudotis, bei kuris yra lankstus ir lengvai išplečiamas. Jis suteikia paprastą sąsają giliesiems neuroniniams tinklams kurti ir mokyti, ir gali būti naudojamas su įvairiomis žemesnio lygio sistemomis įskaitant „TensorFlow“. „Keras“ suteikia iš anksto apmokytus ML modelius, skirtus įprastoms užduotims, pvz., vaizdų atpažinimui ar NLP, taip pat įvairias neuroninių tinklų architektūras, įskaitant konvoliucinius, rekurentinius tinklus, autoenkoderius ir kt.. Atliekant bandymus, naudojantis „Keras“ biblioteka buvo realizuoti BERT, MLP ir CNN modeliai. „Scikit-learn“ yra bendros paskirties ML biblioteka, kurioje yra daugybė priemonių ir algoritmų, skirtų įvairių tipų mašininio mokymosi uždaviniams, tarp jų klasifikavimui, regresijai, klasterizavimui, matmenų mažinimui ir kt., atlikti. Naudojantis šia biblioteka realizuoti SVM ir RF klasifikavimo modeliai.

Duomenų vektorizavimo etape, kenkėjiškų ir normalių programų elgsena buvo perteikta sukuriant kiekvienam vykdomajam failui dokumentą, kuriuos sudaro HIDS sugeneruoti įspėjamieji pranešimai, programos vykdymo metu. Po to, kiekvienas dokumentas buvo paverstas 768 dimensijų vektoriumi ar vektorių seka. Šie duomenys naudojami modelių apmokymui ir bandymui. Klasifikavimo modeliai apmokomi atlikti klasifikavimą keliais būdais:

1. Aptinkant įsibrovimą. Programų elgsenos dokumentai klasifikuojami į 2 klases – kenksmingas ir nekenksmingas.
2. Nustatant KPK tipą. Dokumentai klasifikuojami į 5 klases – „AdWare“, „Trojan“, „Downloader“, „WebToolbar“, kurios nusako KPK tipą, bei klasę „Benign“, kas reiškia, kad bandinys nėra kenkėjiškas.

KPK tipo klasifikavimo mokymui nebuvo naudojami visi surinkti duomenys, kadangi kai kurių klasių bandinių kiekis yra per mažas. 3.6 lentelėje pateikiamas naudojamas klasių ir jų bandinių kiekis atliekant KPK tipo klasifikavimą.

3.6 lentelė. Bandyme naudojamas duomenų kiekis pagal klases

Klasė	Kiekis
AdWare	5369
Trojan	2724
Downloader	692
WebToolbar	581
Benign	2978
Viso	12344

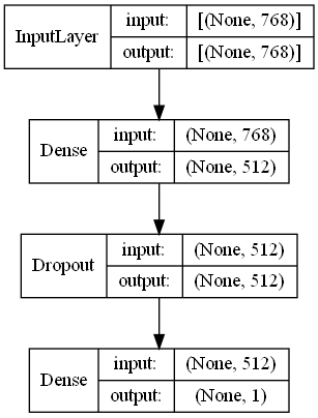
3.7 lentelėje aprašomi klasifikavimo modelių, kurie buvo naudojami atliekant bandymus, hiperparametrai. Neuroninių tinklų modeliams papildomai aprašyti parametrai kiekviename tinklo paslėptame, bei išvesties sluoksniuose.

3.7 lentelė. Klasifikavimo modelių hiperparametrai

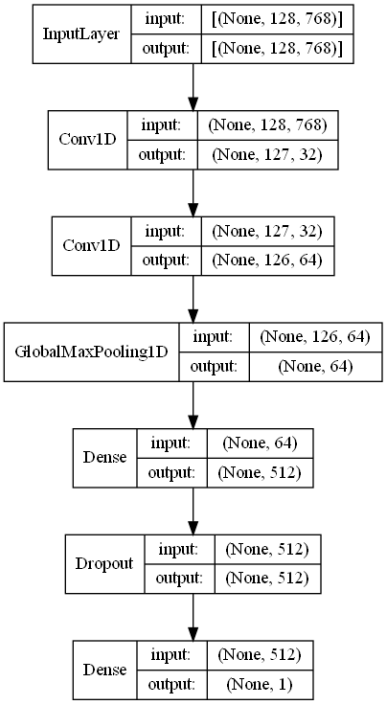
Modelis	Modelio parametrai	Sluoksnis	Parametrai
SVM	C = 100, kernel = rbf, gamma = 0.01	-	-
RF	n_estimators = 1800, min_samples_split = 5, min_samples_leaf = 2, max_depth = 150, criterion = entropy, bootstrap = False	-	-
MLP	optimizer = adam, loss = categorical_crossentropy / binary_crossentropy, epochs = 30, batch_size = 32	Dense	units = 512, activation = relu
		Dropout	rate = 0.2
		Dense	units = 1 / 5, activation = sigmoid / softmax
CNN	optimizer = adam,	Conv1D	filters = 32, kernel_size = 2, activation = relu

	loss = categorical_crossentropy / binary_crossentropy, epochs = 30, batch_size = 32	Conv1D	filters = 64, kernel_size = 2, activation = relu
		GlobalMaxPool1D	-
		Dense	units = 512, activation = relu
		Dropout	rate = 0.1
		Dense	units = 1 / 5, activation = sigmoid / softmax

3.11 pav. ir 3.12 pav. vaizduojamos MLP ir CNN modelių tinklų architektūros. Klasifikavimo modelių įvestyje pateikiami BERT sugeneruoti dokumentų vektoriai. MLP, SVM ir RF modelių įvesčiai pateikiami apjungti vektoriai, kurių kiekvieno dydis 768 dimensijos, tuo tarpu CNN – sekos vektoriai, kurių kiekvienam dokumentui buvo sugeneruota po 128 vektorius, iš kurių kiekvienas yra 768 dimensijų dydžio. Kiekvienas vektorius iš sekų vektorių atspindi vieną BERT įvesties sekos žetoną. Tokie vektoriai geriausiai perteikia viso dokumento kontekstą, o CNN modelis analizuojant šiuos duomenis turėtų galėti įžvelgti svarbius kenkėjiškos veiklos dėsningumus, kas turėtų padidinti įsibrovimo aptikimo tikslumą. CNN modelyje naudojami 2 iš eilės einantys konvoliuciniai 1 dimensijos sluoksniai tinklui pagilinti, kas turėtų leisti duomenyse įžvelgti daugiau svarbesnių programų elgsenos savybių. Naudojami 1 dimensijos sluoksniai nes jie yra pritaikyti darbui su tekstiniais duomenimis, kuriuos sudaro seka. „GlobalMaxPooling1D“ sluoksnyje atliekama sutelkimo operacija, kuri padeda užfiksuoti svarbiausią kiekvieno ankstesnio sluoksnio filtro požymį ir sumažinti duomenų dimensijas. Po pilnai sujungto sluoksnio naudojamas „Dropout“ sluoksnis, kurio pagalba sumažinama persimokymo riziką ir pagerinamos modelio apibendrinimo savybes.



3.11 pav. MLP modelio architektūra



3.12 pav. CNN modelio architektūra

Apmokytų mašininio mokymo modelių klasifikavimo našumas įvertinamas pagal metrikas, kurios dažniausiai naudojamos vertinant įsibrovimo aptikimo modelius – tikslumas (angl. *precision*), jautrumas (angl. *recall*) ir F1 rodiklis (angl. *F1-score*). Tikslumas tai rodiklis, rodantis, kiek teigiamų klasifikatoriaus prognozių iš tikrųjų yra teisingos. Aukštas tikslumas reiškia, kad modelis pateikia labai nedaug klaidingų teigiamų prognozių ir jam gerai sekasi nustatyti tikrus teigiamus rezultatus. Jautrumas – tai rodiklis, rodantis, kiek faktinių teigiamų atvejų nustatė klasifikatorius. Didelis jautrumo rodiklis reiškia, kad modelis gerai identifikuoja tikrai teigiamus atvejus, o klaidingų neigiamų atvejų yra nedaug. F1 rodiklis apjungia tikslumo ir jautrumo rodiklius, imant jų harmoninį vidurkį. Tai pusiausvyra tarp tikslumo ir jautrumo, jis paprastai naudojamas, kai duomenų rinkinys yra nesubalansuotas ir siekiama užtikrinti pusiausvyrą tarp teisingų teigiamų ir klaidingų teigiamų rezultatų rodiklių. Kiekvieno vertinimo rodiklio formulė pateikiama žemiau.

$$precision = \frac{TP}{TP + FP}$$

čia TP – teisingai teigiami aptikimai, FP – klaidingai teigiami aptikimai.

$$recall = \frac{TP}{TP + FN}$$

FN – klaidingai neigiami aptikimai.

$$F1 = \frac{2(precision * recall)}{precision + recall}$$

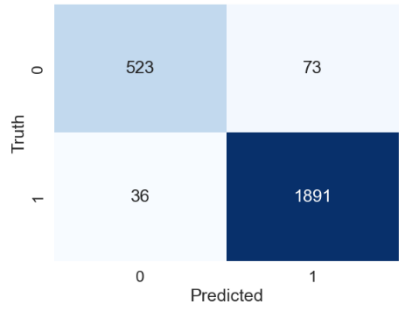
Toliau, modeliai buvo apmokyti ir ištestuoti duomenis išskaidžius į mokymo ir testavimo dalis, po 80% ir 20% atitinkamai. Mokymo rinkinys naudojamas modeliui mokyti, o testavimo rinkinys naudojamas apmokyto modelio efektyvumui įvertinti. Šio metodo idėja pagrįsta tuo, kad jei modelis gali tiksliai nuspėti testavimo rinkinio rezultatus, tikėtina, kad jis pasiteisins ir su naujais, nematytais duomenimis. Kiekvieno modelio klasifikavimo rezultatai pateikiami 3.8 lentelėje, geriausi rezultatai pateikti paryškintu šriftu. Tikslumo, jautrumo ir F-balo rodikliai gauti skaičiuojant visų klasių klasifikavimo rezultatų makro-vidurkį.

3.8 lentelė. Klasifikavimo rezultatai

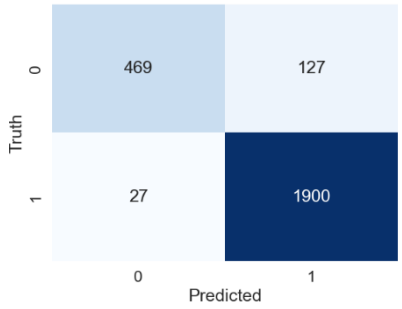
Modelis	**Įsibrovimo aptikimas**			**KPK tipo nuspėjimas**		
	Tikslumas	Jautrumas	F-balas	Tikslumas	Jautrumas	F-balas
RF	0.943	0.887	0.910	0.871	0.788	0.819
SVM	0.949	0.929	0.939	0.882	0.817	0.844
MLP	0.917	0.715	0.759	0.764	0.701	0.701
CNN	0.977	0.942	**0.957**	0.885	0.858	**0.871**

Iš klasifikavimo rezultatų lentelės galima matyti, kad 3 iš 4 išbandytų algoritmų geba gerai aptikti įsibrovimą, jų F-balo rodiklis yra aukštesnis nei 0.90, o geriausias F-balo rezultatas aptinkant įsibrovimą gautas su CNN modeliu – 0.957. Dėmesį vertėtų atkreipti ir į SVM įsibrovimo aptikimo rezultatą – 0.939, kas yra labai geras rezultatas atsižvelgiant į tai, kad lyginant modelį su CNN, modelis yra negilus ir yra labai paprastas, o jo apmokymui reikalingos gerokai mažesnės sąnaudos. Blogiausias rezultatas gautas su MLP – 0.759 F-balo rodiklis. Nors šio modelio tikslumas yra gana didelis – 0.917, tačiau jautrumo rodiklis yra gerokai mažesnis, kas parodo, kad modelis neblogai atpažįsta kenkėjiškus failus, bet dažnai klysta klasifikuojant švarius failus juos priskirdamas kenkėjiškiems. Nuspėjant KPK tipą, modelių F-balo rodikliai yra mažesni lyginant su įsibrovimo aptikimu, nes šiuo atveju yra klasifikuojama į daugiau klasių, o kai kurių kenkėjiškų programų užfiksuota elgsena yra panaši su bandiniais iš kitų klasių. Taip pat kai kurių klasių bandinių kiekis yra mažesnis, kas neleidžia modeliui taip gerai išmokti kai kurių KPK tipų elgsenos. Atliekant KPK tipo nuspėjimo užduotį geriausias F-balo rodiklis gautas su CNN modeliu – 0.871, o blogiausias su MLP – 0.701, taip pat kaip atliekant įsibrovimo aptikimo užduotį.

3.13 pav. – 3.16 pav. pateiktos kiekvieno modelio klasifikavimo painiavos matricos atliekant įsibrovimo aptikimo bandymą. Painiavos matrica – tai lentelė, naudojama ML modelių našumui įvertinti, lyginant prognozuotas klases su tikrosiomis klasėmis bandomuosiuose duomenyse.



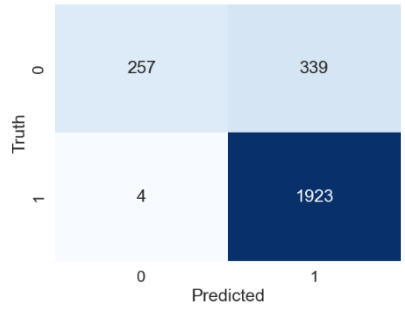
3.13 pav. SVM



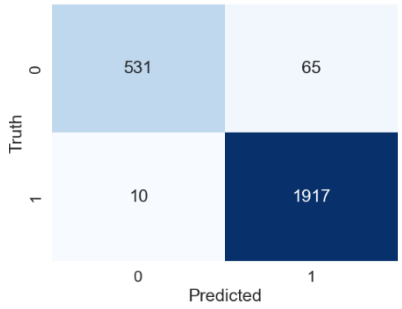
3.14 pav. RF

0 – nekensminga

1 – kenksminga



3.15 pav. MLP

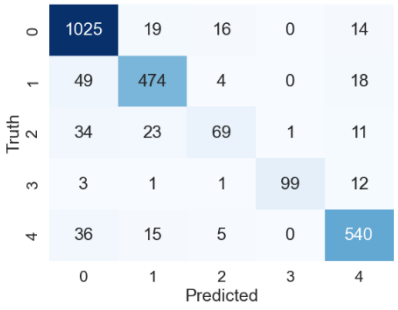


3.16 pav. CNN

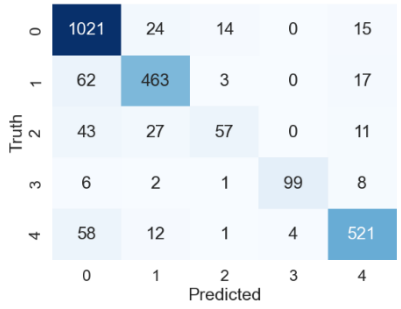
Painiavos matricose matomi modelių teisingų ir neteisingų spėjimų kiekiai. Geriausią rezultatą parodžiusio CNN modelio painiavos matricoje matoma, kad modelis gali labai gerai nuspėti kenksmingo failo klasę – vertinant modelį buvo tik 10 neteisingų spėjimų šiai klasei. Tai parodo, kad surinkti elgsenos duomenys, bei siūlomas vektorizavimo metodas gerai perteikia kenkėjiškų ir nekenkėjiškų programų savybes. Modelis kiek prasčiau nuspėja nekenksmingo failo klasę ir tokį failą priskiria kenksmingiems. Galima to priežastis yra ta, kad mokymui naudoti duomenys yra nesubalansuoti ir švarių failų bandinių yra tik 2978, kai tuo tarpų kenkėjiškų – 9633. Nesubalansuotas duomenų rinkinys gali turėti neigiamos įtakos mašininio mokymosi modelio našumui, nes modelis gali būti šališkas didumos klasės atžvilgiu. Taip yra todėl, nes modelis daugiau mokomas pagal didumos klasės pavyzdžius ir nepakankamai pagal mažumos klasės, todėl mažumos klasės tikslumas gali būti mažesnis. Atliekant bandymus nebuvo taikomi duomenų rinkinio klasių subalansavimo metodai, kuriuos pritaikius rezultatai galėtų būti dar geresni.

3.17 pav. – 3.20 pav. pateiktos painiavos matricos atliekant KPK tipo nustatymo bandymą.

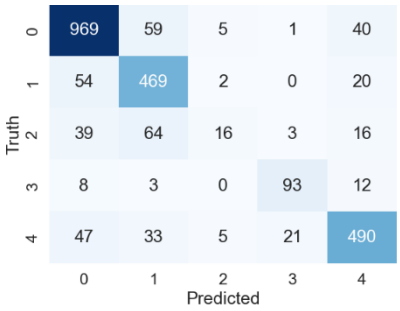
0 – AdWare
1 – Trojan
2 – Downloader
3 – WebToolbar
4 – Benign



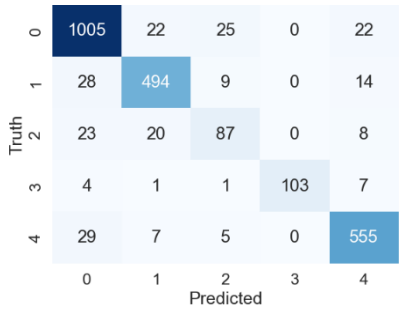
3.17 pav. SVM



3.18 pav. RF



3.19 pav. MLP



3.20 pav. CNN

KPK tipo nustatymo painiavos matricose galima matyti, kad dažnu neteisingo spėjimo atveju bandinys klasifikuojamas kaip „AdWare“ tipo KPK. Šios klasės bandinių duomenų rinkinyje buvo daugiausia, dėl to modeliai yra labiau linkę programos veiklą priskirti šiam tipui. Kaip ir įsibrovimo aptikimo bandymo atveju, jeigu klasės duomenų rinkinyje būtų subalansuotos, manoma, kad būtų galima gauti geresnius rezultatus. Kai kurių kenkėjiškų programų užfiksuota elgsena yra gana panaši į kitų tipų KPK elgseną, todėl modeliams jas atskirti yra sudėtinga. Optimizuojant geriausiai pasirodžiusį CNN modelį ir padidinant jo tinklo gyli pridedant daugiau konvoliucinių sluoksnių, tikėtina, kad modelis galėtų geriau įžvelgti skirtingų tipų programų elgseną ir klasifikuoti jas tiksliau.

3.5.1. Įsibrovimo aptikimas analizuojant pradinius AWSCTD HIDS duomenis

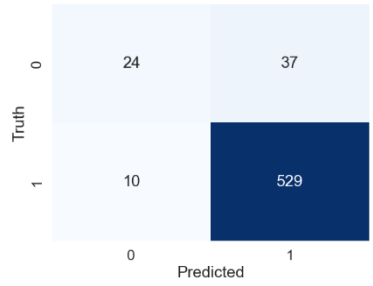
Siekiant palyginti ar papildytas duomenų rinkinys yra pranašesnis atliekamas bandymas su originalaus AWSCTD duomenų rinkinio OSSEC įspėjamaisiais pranešimais ir palyginami gauti rezultatai. Duomenims paruošti, vektorizuoti ir modeliams apmokyti atliekami tie patys žingsniai ir naudojami tie patys modelių parametrai. Bandymas atliekamas dvejais būdais – aptinkant įsibrovimą ir nustatant KPK tipą, taip pat kaip ir su papildytu duomenų rinkiniu. Klasifikavimas atliekamas tik su 1 DL modeliu – CNN, kadangi modelis parodė geriausius rezultatus ankstesniuose bandymuose. Atliekant AWSCTD OSSEC žurnalinių įrašų apžvalgą buvo nustatyta, kad nekenkėjiškų bandinių klasių kiekis duomenų rinkinyje yra per mažas. Netaikant papildomų priemonių klasėms subalansuoti, klasifikuojant į 2 klases – kenkėjiškus ir

nekenkėjiškus, geri rezultatai nebūtų gaunami. Šiai problemai sušvelninti, atliekant įsibrovimo aptikimo bandymą, modelis apmokomas tik su 2700 kenkėjiškų bandinių, o švarių bandinių kiekis padidinamas dvigubai, kad jų būtų apie 10% nuo kenkėjiškų bandinių skaičiaus. Mažumos klasės bandiniams padidinti naudojamas SMOTE (angl. *Synthetic Minority Over-sampling Technique*) metodas. Tai duomenų bandinių padidinimo metodas, naudojamas siekiant pašalinti klasių nesubalansuotumą duomenų rinkinyje. Šis metodas pagrįstas sintetinių mažumos klasės pavyzdžių kūrimu, o ne tiesiog esamų pavyzdžių dubliavimu. Tai atliekama atrenkant mažumos klasės pavyzdį ir surandant jo *k* artimiausių kaimynų. Tada sintetiniai bandiniai generuojami atsitiktinai pasirenkant vieną iš *k* artimiausių kaimynų, paimant skirtumą tarp to taško ir pradinio taško, padauginant šį skirtumą iš atsitiktinio skaičiaus nuo 0 iki 1 ir pridedant jį prie pradinio taško. Taip sukuriamas naujas taškas, kuris yra panašus į pradinį tašką, bet ne identiškas jam (Brownlee J, 2020). 3.9 lentelėje pateikiami klasifikavimo rezultatai.

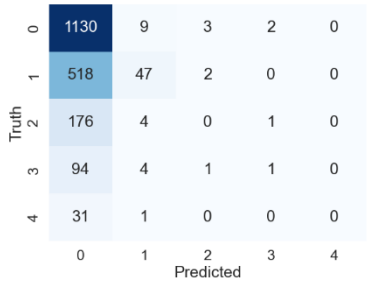
3.9 lentelė. Klasifikavimo rezultatai su originaliu duomenų rinkiniu

Modelis	**Įsibrovimo aptikimas**			**KPK tipo nuspėjimas**		
	Tikslumas	Jautrumas	F-balas	Tikslumas	Jautrumas	F-balas
CNN	0.820	0.687	0.731	0.311	0.216	0.180

Analizuojant HIDS įspėjamuosius pranešimus iš originalaus AWSCTD duomenų rinkinio ir atliekant įsibrovimo aptikimo bandymą pasiektas 0.731 F-balo rezultatas, kas yra mažiau nei su papildytu duomenų rinkiniu. Žemas jautrumo rodiklis aptinkant įsibrovimą parodo, kad modelis yra mažiau efektyvus atpažįstant nekenkėjiškus failus. Rezultatai rodo, kad originalūs duomenys, lyginant su papildytais, blogiau perteikia kenkėjiškų ir nekenkėjiškų programų elgseną. Atliekant KPK tipo nuspėjimo užduotį su CNN modeliu gautas 0.180 F-balo rezultatas, kas yra beveik 5 kartus blogesnis rezultatas nei atliekant bandymą su papildytais duomenimis. 3.21 pav. ir 3.22 pav. pateiktos modelio painiavos matricos, kurios detalizuoja modelių klasifikavimo našumą.



3.21 pav. CNN painiavos matrica. 0 – nekenksmingi, 1 – kenksmingi



3.22 pav. CNN painiavos matrica. 0 – AdWare, 1 – Trojan, 2 – Downloader, 3 – WebToolbar, 4 - Benign

Iš painiavos matricų galima matyti, kad atliekant įsibrovimo aptikimo uždavinį, nors modelis gali neblogai aptikti kenkėjiškus failus, tačiau jis nėra efektyvus klasifikuojant nekenkėjiškus failus ir dažniau klasę nuspėja neteisingai, nei teisingai. Dėl didelio klaidingai teigiamų aptikimų skaičiaus toks sprendimas su tokio tipo duomenimis negalėtų būti praktiškai panaudojamas. Atliekant KPK tipo nustatymo uždavinį, matricoje matoma, kad modelis šį uždavinį sprendžia labai blogai ir beveik visus bandinius klasifikuoja kaip „AdWare“ tipo KPK. Ankstesniame skyriuje apžvelgiant OSSEC įspėjamuosius pranešimus AWSCTD duomenų rinkinyje, buvo nustatyta, kad rinkinyje vyrauja pasikartojantys pranešimai ir tada buvo daroma prielaida, kad aptikti kenkėjiškas programas su mažą įvairovę turinčiais duomenimis bus sudėtinga. Šio bandymo rezultatai tai patvirtina – duomenų rinkinio pranešimai gerai neatspindi skirtingų tipų KPK savybių. Rezultatus galimai būtų galima pagerinti taikant kitus metodus atvaizduojant ir vektorizuojant programų elgseną.

3.6. Skyriaus apibendrinimas

Skyriuje buvo apžvelgiami AWSCTD duomenų rinkinio OSSEC žurnaliniai įrašai, kuriuos analizuojant buvo nustatyta, kad duomenys turi tam tikrų trūkumų, kas apsunkina įsibrovimų aptikimą naudojant tokius duomenis – duomenų rinkinyje vyrauja pasikartojantys žurnaliniai įrašai, yra išskirtinių pranešimų trūkumas, įvykių eiliškumas neatspindi realios įvykių sekos, bei duomenų klasės yra nesubalansuotos. Dėl šių priežasčių duomenų rinkinį nuspręsta papildyti pakartotinai vykdant kenkėjiškas ir nekenkėjiškas programas su patobulinta HIDS konfigūracija.

Atliekant tyrimą, surinkti programų elgsenos duomenys, kuriuos sugeneravo „Wazuh“ HIDS, buvo atitinkamai apdoroti – žurnaliniai įrašai buvo sugrupuoti pagal vykdomojo failo bandinį, kiekvienam bandiniui atvaizduojant jo vykdymo metu sugeneruotus HIDS įspėjamuosius pranešimus.

Tyrimo eigoje, duomenų paruošimo etape, buvo pasiūlytas žurnalinių įrašų atvaizdavimo būdas dokumentuose, kur kiekviename dokumente atvaizduojami atrinkti bandinio žurnaliniai įrašai. Taip pat, buvo atliktas tekstų vektorizavimo metodų palyginimas ir pasirinktas BERT modelis, dėl savo savybių apdoroti nežinomus duomenis ir perteikti kontekstą. Naudojant šį metodą programų veiklos dokumentams sugeneruoti vektoriniai atvaizdai.

Įvertinti ar žurnalinių įrašų atvaizdavimo būdas yra tinkamas, įsibrovimams aptikti, bei KPK tipams nustatyti buvo apmokyti ir išbandyti 4 ML modeliai – SVM, RF, MLP ir CNN. Klasifikavimo rezultatai parodė, kad dauguma modelių gali gerai atpažinti kenkėjiškus failus. Geriausią našumą parodė CNN

algoritmas su kuriuo gautas 0.957 F-balas atpažįstant ar failas yra kenkėjiškas ir 0.871 nustatant kenkėjiškos programos tipą.

Išvados

1. Atlikus įsibrovimų aptikimo temos literatūros analizę buvo nustatyta, kad tradiciniai įsibrovimo aptikimo metodai, paremti parašais, yra neveiksmingi aptinkant naujas, modifikuotas ar slaptas atakas, todėl IDS sistemose reikalinga naudoti pažangius anomalijomis paremtus įsibrovimo aptikimo metodus, kad būtų galima efektyviai aptikti dar nematytas, nulinės dienos atakas. Taip pat pastebėta, kad kokybiškų, laisvai prieinamų duomenų rinkinių orientuotų į įsibrovimo aptikimą kompiuterio lygmenyje, „Windows“ OS, beveik nėra, o tai stabdo HIDS anomalijomis paremtų metodų vystymąsi, todėl turėti tokio tipo duomenų rinkinių yra labai reikalinga.
2. Apžvelgus žurnalinių įrašų analizės metodus išsiaiškinta, kad mokslininkai atlikdami tyrimus sėkmingai taiko NLP metodus žurnaliniams įrašams paversti į skaitinę formą, kadangi žurnalinius įrašus pagrinde sudaro žodžiai, o įsibrovimams aptikti sėkmingai išnaudojami ML klasifikavimo metodai. Be to, naudojant NLP metodus, kuriuos taikant vektorizuojant tekstinius duomenis perteikiamas kontekstas, yra gaunami geresni rezultatai. Taip pat geresni įsibrovimo aptikimo rezultatai yra gaunami taikant gilius neuroninius tinklus, nei tradicinius ML klasifikavimo metodus. Todėl siekiant efektyviai aptikti įsibrovimus, analizuojant žurnalinius įrašus, reikalinga kombinuoti NLP, bei DL klasifikavimo metodus.
3. Išnagrinėjus AWSCTD duomenų rinkinio HIDS įspėjamuosius pranešimus buvo įžvelgti tam tikri duomenų trūkumai, kurie apsunkintų įsibrovimo aptikimą, dėl to duomenų rinkinys buvo papildytas pakartotinai vykdant tas pačias programas ir renkant jų elgsenos duomenis su „Wazuh“ HIDS. Taip buvo surinktas duomenų rinkinys, turintis daugiau įvairesnių pranešimų apie programų elgseną ir kurį galima panaudoti atliekant tolimesnius įsibrovimo aptikimo tyrimus, analizuojant kompiuterio lygmens žurnalinius įrašus.
4. Įsibrovimams aptikti pasiūlytas žurnalinių įrašų atvaizdavimo būdas dokumentuose ir jų vektorizavimas taikant BERT metodą. Taikant šiuos metodus kiekvienos programos elgseną atspindinčių HIDS įspėjamųjų pranešimų tekstiniai duomenys buvo atvaizduoti atitinkamame dokumente ir paversti į vektorius, su kuriais įsibrovimus aptikti apmokyti RF, SVM, MLP ir CNN klasifikavimo modeliai. Atlikus bandymus, su 3 iš 4 klasifikavimo modeliais buvo gauti aukštesni nei 0.90 F-balo rezultatai aptinkant įsibrovimą. Geriausi rezultatai gauti su CNN – 0.957 F-balas aptinkant įsibrovimą ir 0.871 nustatant kenkėjiškos programos tipą, kas parodo, kad siūlomas žurnalinių įrašų atvaizdavimo metodas gali gerai perteikti programų elgseną, o CNN klasifikavimo modelis gali gerai išmokti kenkėjiškų ir paprastų programų dėsningumus. Dėl aukšto tikslumo

aptinkant kenkėjiškus failus metodas gali būti naudojamas praktikoje aptinkant didžiąją dalį įsibrovimų be žmogaus įsiterpimo.

Literatūros sąrašas

About OSSEC HIDS. (s.a.). Gauta 2022 m. birželio 18 d., https://www.ossec.net/about/

Alboukadel Kassambara. (s.a.). *Agglomerative Hierarchical Clustering*. Gauta 2022 m. birželio 9 d., https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering/

Aldweesh, A., Derhab, A., & Emam, A. Z. (2020). *Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues*. *189*, 105124. https://doi.org/10.1016/j.knosys

BERT. (s.a.). Gauta 2023 m. gegužės 11 d., https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/4

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). *Enriching Word Vectors with Subword Information*. http://arxiv.org/abs/1607.04606

Borisaniya, B., & Patel, D. (2015). Evaluation of Modified Vector Space Representation Using ADFA-LD and ADFA-WD Datasets. *Journal of Information Security*, *06*(03), 250–264. https://doi.org/10.4236/jis.2015.63025

Brownlee J. (2020, sausio 17). *SMOTE for Imbalanced Classification with Python*.

Choudhary, S., & Kesswani, N. (2021). A Hybrid Classification Approach for Intrusion Detection in IoT Network. *Journal of Scientific & Industrial Research* (T. 80).

Christopher Olah. (2015, rugpjūčio 27). *Understanding LSTM Networks*. https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Čeponis, D. (2021). Research of machine and deep learning methods application for host-level intrusion detection and classification. *Research of machine and deep learning methods application for host-level intrusion detection and classification*. Vilnius Gediminas Technical University. https://doi.org/10.20334/2021-008-m

Čeponis, D., & Goranin, N. (2018). Towards a Robust Method of Dataset Generation of Malicious Activity for Anomaly-Based HIDS Training and Presentation of AWSCTD Dataset. *Baltic Journal of Modern Computing*, *6*(3). https://doi.org/10.22364/bjmc.2018.6.3.01

Deng, L., & Yu, D. (2013). Deep learning: Methods and applications. *Foundations and Trends in Signal Processing* (T. 7, Numeriai 3–4, p. 197–387). Now Publishers Inc. https://doi.org/10.1561/2000000039

Deshpande, P., Sharma, S. C., Peddoju, S. K., & Junaid, S. (2018). HIDS: A host based intrusion detection system for cloud computing environment. *International Journal of Systems Assurance Engineering and Management*, *9*(3), 567–576. https://doi.org/10.1007/s13198-014-0277-7

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, *abs/1810.04805*. http://arxiv.org/abs/1810.04805

Du, M., & Li, F. (2016). *Spell: Streaming Parsing of System Event Logs*.

Du, M., Li, F., Zheng, G., & Srikumar, V. (2017). DeepLog: Anomaly detection and diagnosis from system logs through deep learning. *Proceedings of the ACM Conference on Computer and Communications Security*, 1285–1298. https://doi.org/10.1145/3133956.3134015

Elrawy, M. F., Awad, A. I., & Hamed, H. F. A. (2018). Intrusion detection systems for IoT-based smart environments: a survey. *Journal of Cloud Computing* (T. 7, Numeris 1). Springer Verlag. https://doi.org/10.1186/s13677-018-0123-6

Ethala, K., Seshadri, R., Renganathan, N. G., & Saravanan, M. S. (2013). A role of intrusion detection system for wireless lan using various schemes and related issues. *American Journal of Applied Sciences*, *10*(9), 979–985. https://doi.org/10.3844/ajassp.2013.979.985

Getting started with Wazuh. (s.a.). Gauta 2023 m. gegužės 11 d., https://documentation.wazuh.com/current/getting-started/index.html

Haider, W., Creech, G., Xie, Y., & Hu, J. (2016). Windows based data sets for evaluation of robustness of Host based Intrusion Detection Systems (IDS) to zero-day and stealth attacks. *Future Internet*, *8*(3). https://doi.org/10.3390/fi8030029

Haider, W., Hu, J., Slay, J., Turnbull, B. P., & Xie, Y. (2017). Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling. *Journal of Network and Computer Applications*, *87*, 185–192. https://doi.org/10.1016/j.jnca.2017.03.018

He, P., Zhu, J., Zheng, Z., & Lyu, M. R. (2017). Drain: An Online Log Parsing Approach with Fixed Depth Tree. *Proceedings - 2017 IEEE 24th International Conference on Web Services, ICWS 2017*, 33–40. https://doi.org/10.1109/ICWS.2017.13

He, S., Zhu, J., He, P., & Lyu, M. R. (2016). Experience Report: System Log Analysis for Anomaly Detection. *Proceedings - International Symposium on Software Reliability Engineering, ISSRE*, 207–218. https://doi.org/10.1109/ISSRE.2016.21

He, S., Zhu, J., He, P., & Lyu, M. R. (2020). *Loghub: A Large Collection of System Log Datasets towards Automated Log Analytics*. http://arxiv.org/abs/2008.06448

IBM Cloud Education. (2020, spalio 20). *Convolutional Neural Networks*. https://www.ibm.com/cloud/learn/convolutional-neural-networks

Yesir, S., & Soğukpinar, İ. (2021). Malware Detection and Classification Using fastText and BERT. *2021 9th International Symposium on Digital Forensics and Security (ISDFS)*, 1–6. https://doi.org/10.1109/ISDFS52919.2021.9486377

Ying, S., Wang, B., Wang, L., Li, Q., Zhao, Y., Shang, J., Huang, H., Cheng, G., Yang, Z., & Geng, J. (2021). An Improved KNN-Based Efficient Log Anomaly Detection Method with Automatically Labeled Samples. *ACM Transactions on Knowledge Discovery from Data*, *15*(3). https://doi.org/10.1145/3441448

Yugesh Verma. (2021, liepos 17). *Complete Guide To Bidirectional LSTM*. https://analyticsindiamag.com/complete-guide-to-bidirectional-lstm-with-python-codes/

Khan, K., Mehmood, A., Khan, S., Khan, M. A., Iqbal, Z., & Mashwani, W. K. (2020). A survey on intrusion detection and prevention in wireless ad-hoc networks. *Journal of Systems Architecture*, *105*. https://doi.org/10.1016/j.sysarc.2019.101701

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, *2*(1). https://doi.org/10.1186/s42400-019-0038-7

Kim, C., Jang, M., Seo, S., Park, K., & Kang, P. (2021). Intrusion Detection Based on Sequential Information Preserving Log Embedding Methods and Anomaly Detection Algorithms. *IEEE Access*, *9*, 58088–58101. https://doi.org/10.1109/ACCESS.2021.3071763

Krašto apsaugos ministerija. (2021). *Pirmą kartą kibernetinio saugumo būklė apžvelgiama plačiau, ne tik iš Krašto apsaugos ministerijos perspektyvos*. http://kam.lt/lt/naujienos_874/aktualijos_875/pirma_karta_kibernetinio_saugumo_bukle_apzvelgiama_placiau_ne_tik_is_krasto_apsaugos_ministerijos_perspektyvos.html

Le, V. H., & Zhang, H. (2022). *Log-based Anomaly Detection with Deep Learning: How Far Are We?* https://doi.org/10.1145/3510003.3510155

Liang, Y., Zhang, Y., Xiong, H., & Sahoo, R. (2007). Failure prediction in IBM BlueGene/L event logs. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 583–588. https://doi.org/10.1109/ICDM.2007.46

Lin, Q., Zhang, H., Lou, J. G., Zhang, Y., & Chen, X. (2016). Log clustering based problem identification for online service systems. *Proceedings - International Conference on Software Engineering*, 102–111. https://doi.org/10.1145/2889160.2889232

Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences (Switzerland)* (T. 9, Numeris 20). MDPI AG. https://doi.org/10.3390/app9204396

LOG-MD. (s.a.). Gauta 2023 m. gegužės 11 d., https://www.imfsecurity.com/free

Lu, S., Wei, X., Li, Y., & Wang, L. (2018). Detecting Anomaly in Big Data System Logs Using Convolutional Neural Network. *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, 151–158. https://doi.org/10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00037

Makanju, A., Zincir-Heywood, A. N., & Milios, E. E. (2012). A lightweight algorithm for message type extraction in system application logs. *IEEE Transactions on Knowledge and Data Engineering*, *24*(11), 1921–1936. https://doi.org/10.1109/TKDE.2011.138

Mäntylä, M., Varela, M., & Hashemi, S. (2022). *Pinpointing Anomaly Events in Logs from Stability Testing -- N-Grams vs. Deep-Learning*. http://arxiv.org/abs/2202.09214

Meng, W., Liu, Y., Zhang, S., Pei, D., Dong, H., Song, L., & Luo, X. (2019, sausio 22). Device-Agnostic Log Anomaly Classification with Partial Labels. *2018 IEEE/ACM 26th International Symposium on Quality of Service, IWQoS 2018*. https://doi.org/10.1109/IWQoS.2018.8624141

Meng, W., Liu, Y., Zhu, Y., Zhang, S., Pei, D., Liu, Y., Chen, Y., Zhang, R., Tao, S., Sun, P., & Zhou, R. (2019). *LogAnomaly: Unsupervised Detection of Sequential and Quantitative Anomalies in Unstructured Logs*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. http://arxiv.org/abs/1301.3781

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). *Distributed Representations of Words and Phrases and their Compositionality*. http://arxiv.org/abs/1310.4546

Murtaza, S. S., Khreich, W., Hamou-Lhadj, A., & Couture, M. (2013). A host-based anomaly detection approach by representing system calls as states of kernel modules. *2013 IEEE 24th International Symposium on Software Reliability Engineering, ISSRE 2013*, 431–440. https://doi.org/10.1109/ISSRE.2013.6698896

OSSEC Log monitoring/analysis. (s.a.). Gauta 2022 m. birželio 18 d., https://www.ossec.net/docs/docs/manual/monitoring/index.html

OSSEC Syscheck. (s.a.). Gauta 2022 m. birželio 18 d., https://www.ossec.net/docs/docs/manual/syscheck/index.html

Otoum, Y., & Nayak, A. (2021). AS-IDS: Anomaly and Signature Based IDS for the Internet of Things. *Journal of Network and Systems Management*, *29*(3). https://doi.org/10.1007/s10922-021-09589-6

Ozkan-Okay, M., Samet, R., Aslan, O., & Gupta, D. (2021). A Comprehensive Systematic Literature Review on Intrusion Detection Systems. *IEEE Access* (T. 9, p. 157727–157760). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ACCESS.2021.3129336

Ryciak, P., Wasielewska, K., & Janicki, A. (2022). Anomaly Detection in Log Files Using Selected Natural Language Processing Methods. *Applied Sciences*, *12*(10), 5089. https://doi.org/10.3390/app12105089

Russinovich M, & Garnier T. (2023, balandžio). *Sysmon*. https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon

Tang, L., Li, T., & Perng, C.-S. (2011). *LogSig: Generating System Events from Raw Textual Logs*. Association for Computing Machinery.

Wang, J., Tang, Y., He, S., Zhao, C., Sharma, P. K., Alfarraj, O., & Tolba, A. (2020). LogEvent2vec: LogEvent-to-vector based anomaly detection for large-scale logs in internet of things. *Sensors (Switzerland)*, *20*(9). https://doi.org/10.3390/s20092451

Wang, Q., Zhang, X., Wang, X., & Cao, Z. (2022). Log Sequence Anomaly Detection Method Based on Contrastive Adversarial Training and Dual Feature Extraction. *Entropy*, *24*(1). https://doi.org/10.3390/e24010069

Windows Logging Cheat Sheet. (s.a.). Gauta 2023 m. gegužės 11 d., https://static1.squarespace.com/static/552092d5e4b0661088167e5c/t/5c586681f4e1fced3ce1308b/1549297281905/Windows+Logging+Cheat+Sheet_ver_Feb_2019.pdf

Zhang, C., Wang, X., Zhang, H., Zhang, H., & Han, P. (2021). Log Sequence Anomaly Detection Based on Local Information Extraction and Globally Sparse Transformer Model. *IEEE Transactions on Network and Service Management*, *18*(4), 4119–4133. https://doi.org/10.1109/TNSM.2021.3125967

Zhang, M., Chen, J., Liu, J., Wang, J., Shi, R., & Sheng, H. (2022). LogST: Log Semi-supervised Anomaly Detection Based on Sentence-BERT. *2022 7th International Conference on Signal and Image Processing, ICSIP 2022*, 356–361. https://doi.org/10.1109/ICSIP55141.2022.9886069

Zhang, X., Xu, Y., Lin, Q., Qiao, B., Zhang, H., Dang, Y., Xie, C., Yang, X., Cheng, Q., Li, Z., Chen, J., He, X., Yao, R., Lou, J. G., Chintalapati, M., Shen, F., & Zhang, D. (2019). Robust log-based anomaly detection on unstable log data. *ESEC/FSE 2019 - Proceedings of the 2019 27th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 807–817. https://doi.org/10.1145/3338906.3338931

Zhu, J., He, S., Liu, J., He, P., Xie, Q., Zheng, Z., & Lyu, M. R. (2018). *Tools and Benchmarks for Automated Log Parsing*. http://arxiv.org/abs/1811.03509