Vilniaus universitetas
Duomenų mokslo ir skaitmeninių technologijų institutas
L I E T U V A

INFORMATIKA (N009)

MAŠININIO MOKYMOSI METODŲ VYSTYMAS ĮSILAUŽIMAMS APTIKTI KOMPIUTERIŲ TINKLUOSE

Žydrūnas Vaišnoras

2020 m. spalis

Mokslinė ataskaita DMSTI-DS-N009-20-11

VU Duomenų mokslo ir skaitmeninių technologijų institutas, Akademijos g. 4,
Vilnius LT-08412
www.mii.lt

Santrauka

Įvairių anomalijų, ypač tų, kurios yra nežinomos kompiuterių tinklų programinėje ir aparatinėje įrangoje (angl. *zero-day attack*), aptikimas yra gana didelis iššūkis kompiuterių tinklų administratoriams, kibernetinio saugumo ekspertams ir mokslininkams. Bėgant metams kibernetiniai įsilaužimai tapo vis labiau veiksmingesni ir mažiau pastebimi naudojant įsilaužimo aptikimo sistemas. Veiksmingi realaus laiko anomalijų aptikimo metodai leistų pastariesiems ekspertams operatyviai reaguoti į įsilaužimus kompiuterių tinkluose ir padėtų išvengti ar sumažinti jų pasekmes. Dauguma įsilaužimo apsaugos sistemų aptinka anomalijas remiantis iš anksto aprašytomis taisyklėmis, tačiau atakos šiais laikais kuriamos vis sudėtingesnių struktūrų, kurios yra polimorfinės, daugiavektorinės, daugiapakopės ir itin tikslingos. Tad naujus įsilaužimus realaus laiko tinkluose itin tiksliai galima atpažinti taikant mašininio mokymosi metodus. Šioje mokslinėje ataskaitoje apžvelgiami didelių kompiuterių tinkle siunčiamų/gaunamų duomenų apdorojimo, mašininio mokymosi metodai, kurie automatiškai ir realiu laiku išskiria anomalijas iš normalaus kompiuterių tinklų duomenų srauto.

Reikšminiai žodžiai: kompiuterių tinklas, mašininis mokymasis, kibernetinis saugumas, įsilaužimai kompiuterių tinkle, įsilaužimų aptikimo sistemos.

Turinys

1 Įvadas

Šiuo metu beveik viso pasaulio gyventojai gali susisiekti su vienas kitu naudojant interneto ryšį. Kasdien vis didėja skaičius interneto naudotojų, kurių kiekvienas naudoja bent po vieną ar du įrenginius (pvz. nešiojamą kompiuterį, išmanųjį telefoną). Kartu su interneto naudotojų augančiu skaičiumi didėja duomenų perdavimo kiekiai , dėl to taip pat didėja ir kibernetinių atakų įvairovė, kurių dauguma nukreiptos į kompiuterių tinklus .

Eksponentiniu greičiu augantis kompiuterių tinklų dydis ir jų duomenų srautai vis dažniau atkreipia dėmesį į tokių tinklų saugumą. Neautorizuotas vartotojas gali netinkamai panaudoti ar net sutrikdyti kompiuterių tinklų resursus. Tačiau tam užkirsti kelią gali ugniasienė – pirmos eilės apsauginis mechanizmas, kuris geba aptikti įsilaužimus kompiuterių tinkluose. Vis dėlto, toks sprendimas nėra pakankamai veiksmingas būdas aptikti ir išvengti naujų sukuriamų įsilaužimų. Kaip antrasis tinklų apsaugos sluoksnis naudojamos antivirusinės programos, tačiau jos gali aptikti tik žinomas, iš anksto aprašytais modeliais, atakas.

Įsilaužimų aptikimo sistema (angl. *Intrusion Detection System,* IDS) aptinka kenkėjišką vartotojų elgesį, neautorizuotus prisijungimus ir kuo greičiau juos užkardo ir uždraudžia naudotis kompiuterių tinklo komunikacijomis. IDS yra stipri ir išmani įsilaužimų aptikimo sistema dėl to, nes ji kaupia įsilaužimų ar kitos saugumo taisykles pažeidžiančios veiklos informaciją. Ši informacija yra labai vertingas dalykas kompiuterių tinklų saugumo užtikrinime, nes asmuo, kuris bando įsilaužti į kompiuterių tinklus, vis bando nuslėpti duomenų paketus, kurių sukūrimo istorija ir prigimtis yra paslėpta. Egzistuoja du pagrindiniai IDS tipai – piktnaudžiavimo (angl. *misuse*) ir anomalijų (angl. *anomaly*) aptikimo sistemos. Šie apsauginiai mechanizmai gali būti apjungiami siekiant sukurti hibridinę detekcijos sistemą. Piktnaudžiavimo sistema aptinka tik tokias struktūras (angl. *signature*), kurios yra duomenų bazėje, tuo tarpu anomalijų aptikimo sistema apskaičiuoja elgesio nuokrypį nuo standartinio elgesio profilio [1]. Pagrindinis neprižiūrimos IDS tikslas yra padidinti detekcijos dažnį (angl. *detection rate*, DR) ir sumažinti netikro pavojaus dažnį (angl. *false alarm rate*, FAR). IDS aptinka įsilaužimą tinklo komunikacijoje, tuomet sukuria pavojaus signalą ir siunčia jį kompiuterių tinklų administratoriui, kad šis sustabdytų potencialių įsilaužimą. Pagrindinė mašininio mokymosi problema apsaugant kompiuterių tinklus yra ta, jog didėjant duomenų rinkinio dimensijoms ir dydžiui didėja skaičiavimo sudėtingumas. Tokiu atveju naudojamas tinkamiausių požymių atrinkimas siekiant sumažinti duomenų rinkinio dimensijas. Tai gali būti atliekama naudojant abiejų tipų – prižiūrimo ir neprižiūrimo – mokymosi modelių atveju [2].

1.1 Susiję darbai

Šiame skyriuje apžvelgiami darbai, kuriuose autoriai nagrinėja įvairius metodus įsilaužimams aptikti kompiuterių tinkluose. Atlikus literatūros apžvalgą nustatyti keli pagrindiniai tyrimai susiję su moksline disertacija. Kiekviename poskyryje aprašomas skirtingas metodas veiksmingesniam mašininio mokymosi metodų pritaikymui įsilaužimams aptikti. Šio skyrio pabaigoje aprašomi aktualiausių šios ataskaitos tema sukurtų mokslinių straipsnių įžvalgos.

1.1.1 Neprižiūrimo mokymosi požymių išrinkimas

Požymių išrinkimas (angl. *feature selection*) yra labai svarbus metodas siekiant parinkti geriausius rinkinio dalies požymius ir gauti gerus rezultatus [3]. Didelių matmenų duomenų rinkiniai sudaryti iš daugybės informacijos, kuriuose yra daug triukšmo ir dubliavimosi. Šiuo metu yra sukurta daugybė požymių išrinkimo metodų, tačiau daugelis jų taikomi naudojant prižiūrimą mokymąsi. Neprižiūrimų požymių išrinkimas yra sunki užduotis, nes yra naudojami nesužymėti duomenis (angl. *labeled data*), kurių naudojimas palengvina klasifikavimo procesą. Pastaruoju metu yra paskelbta darbų, kuriuose yra pateikiami neprižiūrimo mokymosi požymių parinkimo (angl. *Unsupervised Feature Selection,* UFS) [4] metodai su savybėmis: požymių taškų sistema [5]; požymių panašumas, klasterizavimas remiantis neneigiamų matricų faktorizavimu, duomenų lokalizacija, skirtingų klasterių atstumų maksimalizavimas, neneigiamoji spektrinė analizė, žemo rango struktūros išsaugojimas, savaiminės išraiškos modelis, dėsningumas vidinių elementų ir sąveika tarp požymių.

UFS išsprendžia klasterizavimo problemas, tokias kaip didelių skaičiavimo resursų poreikį ir padidina sistemos veiklos greitį. Pagrindinis UFS metodo tikslas yra išlaikyti kuo mažesnį mokymosi modelį, kad būtų sumažintas kuo didesnis skaičius požymių arba neturėtų jie svarios reikšmės. Prasad su savo kolegomis pritaikė UFS [6] siekiant atrinkti kuo daugiau reikšmingesnių požymių. Pastarieji mokslininkai taip pat sumažino duomenų rinkinio matmenis ir dydį.

Šis sukurtas būdas leido sukurti naują klasterizavimo metodą tikslesnių įsilaužimo ir įprastų duomenų klasifikavimui, įsilaužimų atakų nežymėtais duomenimis aptikti. Tai išsprendžia problemą siekiant atpažinti iki šiol nematytas kibernetines atakas klasifikuojant nežinomus nesužymėtus duomenų paketus.

1.1.2 *LO-shot* mokymasis

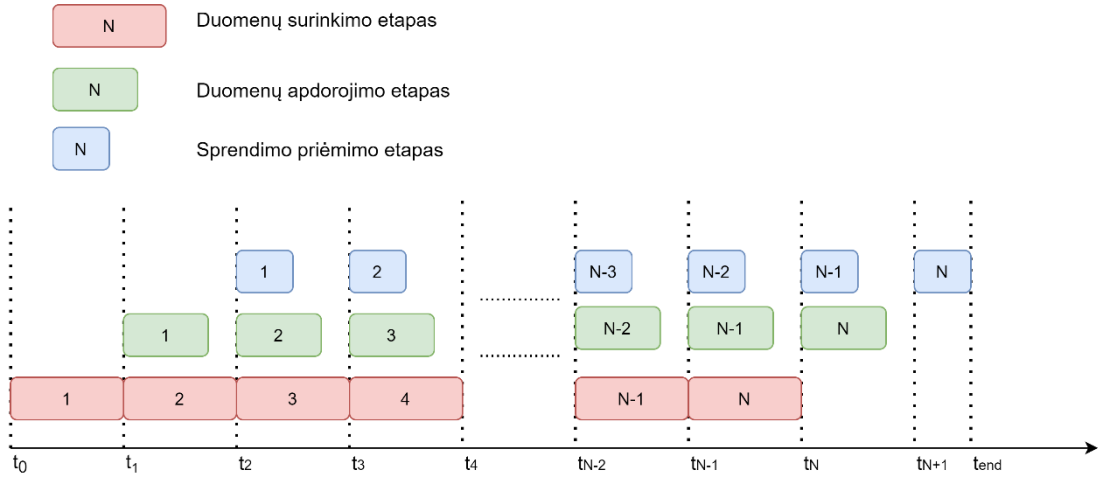
Gilaus prižiūrimo mokymosi modeliai reikalauja ypatingai daug duomenų jų apmokymui. Tuo tarpu žmonės gali lengvai ir greitai generalizuoti dalykus vien iš kelių pavyzdžių [7]. Siekiant mašinų sugebėjimus priartinti prie žmogaus yra svarbu priversti mašinas išmokti naudojant labai mažai duomenų. Kelių bandymų mokymasis (angl. *few-shot learning*, FSL) yra vienas iš būdų kurti mažai duomenų reikalaujančius

modelius. Naudojant tokį būdą modeliai turi išmokti atskirti klases mačius vos po kelis tų klasių pavyzdžius mokymosi metu [7], [8], [9]. Pastaroji pažanga šioje srityje leido sukurti dar ekstremalesnę FSL atmainą, pavadinimu vieno bandymo mokymasis (angl. *one-shot learning*, OSL). Tokio mokymo metu modeliai turi atskirti skirtingas klases būdami apmokyti naudojant tik po vieną skirtingos klasės pavyzdį [10], [11]. Kanados mokslininkai Waterloo ir Ontario taikydami SLaPkNN klasifikatorių savo straipsnyje [12] teoriškai įrodė, kad jų sukurtas *Less than one-shot* (LO-shot) prototipas geba suskirstyti duomenis į klases nepritaikant nei vieną apmokymą. Vystant prototipą galima pagreitinti pavyzdžiais paremtais (angl. *instance-based*) arba tingius, tokius kaip kNN algoritmus, apsimokymus sumažinant jų mokymosi duomenis.

LO-shot mokymosi metodo taikymas itin veiksmingas siekiant aptikti anomalijas kompiuterių tinkluose, kritinėje infrastruktūroje – atpažįstamos atakos prieš kompiuterių tinklo duomenims pasiekiant kompiuterių tinklo mazgus (angl. *hosts*). Tai yra pasiekiama pateikiant modeliui tik kelis naujos kibernetinės atakos duomenų paketus, o pastarasis geba juos tinkamai suskirstyti į duomenų klases.

1.1.3 *SwiftIDS* metodas

Kinijos mokslininkai atlikę analitinę IDS taikymo analizę pastebėjo, kad daugeliuose moksliniuose šaltiniuose aprašomi eksperimentai, tyrimai, kuriuose yra naudojami netiesioginiai kompiuterių tinklo srauto duomenis. T. y., dauguma sukurtų mašininio mokymosi metodų yra pritaikyti sukauptiems duomenims apmokyti modelio sukūrimui. Tačiau maža dalis geba dirbti su realaus laiko duomenimis – atpažinti naujas, dar kompiuterių tinkle, nematytas įsilaužimo atakas. Kad išspręstų šią problemą jie sukūrė metodą SwiftIDS [13]. Šis metodas geba aptikti įsilaužimus kompiuterių tinkle taikant paralelinį įsilaužimų aptikimo mechanizmą (1 pav.). Vadinasi, duomenys nėra kaupiami tam tikrą laiką ir tuomet apdorojami, o tiesioginio srauto duomenis yra apdorojami paraleliai ir su minimaliu užlaikymu. Šis metodas leido mokslininkams pasiekti 1 Gbps greitaveiką fiksuoto ryšio tinkle.



1 pav. SwiftIDS paralelinis sprendimas [13]

1.1.4 Mokslinių straipsnių apžvalga

Sydney Mambwe Kasongo ir kiti [14]pristatė IDS, paremtą giliuoju mokymusi, kur tiesioginio sklidimo gilieji neuroniniai tinklai (angl. *Feed Forward Deep Neural Networks*, FFDNN) buvo apjungti su filtravimu paremtu požymių parinkimo algoritmu. Sukurta FFDNN-IDS sistema įvertinta naudojant gerai žinomą NSL-KDD (angl. *NSL-knowledge discovery and data mining*) duomenų rinkinį ir palyginta su jau egzistuojančiais mašininio mokymosi algoritmais, tokiais kaip SVM, sprendimų medžiai, K-artimiausių kaimynų (angl. *K-Nearest Neighbor,* KNN) ir naivaus Bajeso (angl. *Naive Bayes*). Eksperimentiniai rezultatai parodė, jog FFDNN-IDS pasiekia aukščiausią tikslumą, lyginant su kitais pritaikytais modeliais

Pietų Korėjos mokslininkai Sana Ullah Janet su kolegomis sukūrė lengvą atakų detekcijos strategiją, kuri naudoja prižiūrimo mašininio mokymosi algoritmus (SVM), kurie aptinka bandymus įlieti kenksmingą srautą į kompiuterių tinklą. Simuliacijos rezultatai rodo, jog SVM paremtas klasifikatorius, naudojamas kartu su keliais nesudėtingais požymiais, gali pasiekti aukštesnį tikslumą ir trumpesnę detekcijos trukmę.

Hiral Vegdaet [15] publikuotame straipsnyje pasiūlo saugią ir lanksčią kompiuterių tinklo apsaugą. Ši apsauga sukonstruota taip, jog panaudojant elipsinės kreivės kriptografiją (angl. *Elliptic Curve Cryptography*, ECC) ir modifikuotą išplėstinį šifravimo standartinį algoritmą (angl. *Modify Advanced Encryption Standard Algorithm,* MAES) IDS sistemoje yra aptinkam ir padedama išvengti atakų belaidžiame *Ad hoc* tinkle. Pilnai sistemai sukurti straipsnyje siūloma naudoti *NS 2.35* programinę įrangą ir Ubuntu (Linux) operacinę sistemą.

Souparnika Jayaprakashet ir kiti [16] pateikė transakcijomis paremtą būdą, kuris sudarytas iš naivaus Bajeso klasifikatoriaus ir oktrapleto (tam tikra duomenų struktūra, talpinanti SQL užklausas). Lyginant su kitomis duomenų struktūromis, tokiomis kaip hexapletas ar tripletas, oktrapletas gali leisti pasiekti didesnį našumą ir greitesnę atakos detekciją. Mokymosi algoritmas lengvai gali aptikti rolių pasikeitimus. Naivaus Bajeso klasifikatorius yra paprasčiausias klasifikavimo algoritmas, kuris gali ištraukti visą informaciją, esančią žurnalo įrašuose (angl. *Log files*). Tokia sistema padeda greičiau aptikti kenkėjiškas užklausas.

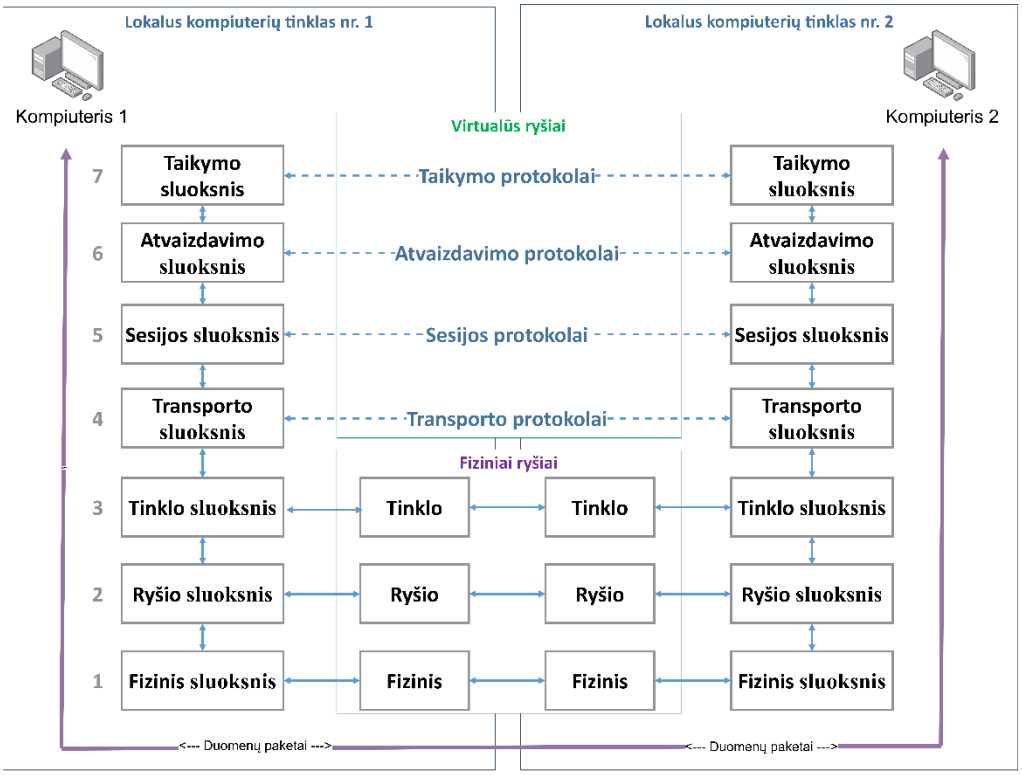
Pratik Satamet [17]pristatė anomalijomis paremtą įsilaužimų aptikimo sistemą *Bluetooth* tinklams – Bluetooth IDS (BIDS). BIDS naudoja n-gramomis paremtą metodą, kuris aprašo normalų Bluetooth protokolo elgesį. Išlyginimo (angl. *smoothing*) metodai, tokie kaip Jelinek-Mercer išlyginimas, buvo naudojami siekiant pagerinti mašininio mokymosi algoritmo našumą aptinkant neįprastas Bluetooth operacijas. Daugelis mašininio mokymosi algoritmų (pvz. C4.5, AdaBoostM1, SVM, naivaus Bajeso, RIPPER, Bagging) buvo pritaikyti Bluetooth protokolo elgesio modeliui kurti.

2 Įsilaužimai kompiuterių tinkluose

2.1 Kompiuterių tinklai ir jų atakos

1978-ais metais Tarptautinė Standartizacijos Organizacija (angl. *International Organization for Standartization*, ISO) sukūrė ir išleido specifikacijų rinkinį, kuris apibrėžė tinklo su nevienalyte įranga architektūrą. 1984-ais metais ta pati organizacija atnaujino projektą ir išleido naują versijos modelį, kurį pavadino *Atvirų sistemų savitarpio sąveikos* etalonu (angl. *Open System Interconnection (OSI) Reference Model*), kuris vėliau tapo tarptautiniu standartu, projektuojant tinklus ir tinklinius produktus. OSI modelis – tai daugiasluoksnė struktūra, kuri atspindi tinklo programinės ir techninės įrangos sąveiką darbo seanso metu.

OSI modelyje tinklo funkcijos paskirstytos į septynis sluoksnius. Septynių sluoksnių koncepciją pateikė Charles Bachman, kuris tuo metu dirbo *Honeywell* kompanijoje [18]. Kiekvienam sluoksniui priskirtos tam tikros tinklinės operacijos, įranga ir protokolai. Sluoksnių skiriamosios ribos – jų sąsajos (angl. *interface*). Kiekvienas sluoksnis teikia tik jam nustatytas paslaugas ir naudojasi žemesnių sluoksnių paslaugomis. Ryšiai tarp kompiuterių vykdomi taip, kad kiekvienas vieno kompiuterio sluoksnis veikia su tuo pačiu kito kompiuterio OSI sluoksniu, kuris yra vadinamas virtualiuoju ryšiu (2 pav.). Kompiuterių tinklo įrenginių sąveika vyksta fizinio sluoksnio lygyje. Kompiuterių tinklo įsilaužimai vyksta visuose OSI modelio lygmenyse, vadinasi, turi būti užtikrintas saugumas visuose sluoksniuose taikant skirtingus apsaugos metodus.



2 pav. OSI modelis

Nūdien žvelgiant į kompiuterių atakas, kurias IDS gali aptikti [19] pastebime, vis daugiau įsilaužimų į kompiuterių tinklų atakas galima apsaugoti naudojant hibridinę įsilaužimo sistemą. Tačiau tam, kad galėtume apsaugoti kompiuterių tinklą nuo įsibrovimų, turime žinoti nuo ko jas saugoti. Šiuo metu dažniausiai literatūroje aprašomų kompiuterių atakų tipai:

- Laikinosios talpyklos perpildymas (angl. *Buffer Overflow*) – nukreipta į laikinosios talpyklos atmintį, kuri įrašinėja į informaciją į atmintį iki tol, kol ji pilnai prisipildo. Prisipildžius yra įrašinėjama toliau – perrašoma esama informacija nauja ištrinant visą prieš tai turėtą informaciją;
- Kirminas – kopijuoja save vietiniame mazge arba per kompiuterių tinklą.
- Trojan – programos atrodo tikros ir taisyklingos, tačiau viduje talpina kenkėjišką kodą;
- Atsisakymas aptarnauti (angl. *Denial of Service*, DoS) – atakos metu programišius pažeidžia ar užima serverių skaičiavimo ar atminties išteklius padarydamas serverius nepasiekiamais vartotojams, kurie nori naudotis serverio teikiamomis paslaugomis. Literatūroje ir pastarųjų metų moksliniuose straipsniuose minimos šios dažniausios DoS atakos: *ping of death, back, mail bomb, UDP storm, apache, smurf, Neptune*;
- Įprastinių vartų sąsajos (angl. *Common gateway interface*, CGI) skriptai – įsilaužėjas naudoja CGI skriptus sukurti ataką išsiunčiant neteisėtas įvestis į saitynų serverį;
- Duomenų srauto užtvindymas – nukreipta į ribotą IDS gebėjimą apdoroti ir nustatyti galimus įsilaužimus, kai yra didžiulis tinklo srautas. Jeigu kibernetinis nusikaltėlis gali sukelti kompiuterių tinklų duomenų srautų perpildymą, tuomet IDS bus užimta šio srauto analizavimu;
- Fizinė ataka – nukreipta į fizinius kompiuterių sistemos mechanizmus;
- Slaptažodžio ataka – siekia atspėti slaptažodį per labai trumpą laiką ir tuo metu yra stebima serija nesėkmingų prisijungimų;
- Informacijos rinkimas – informacija renkama kompiuteriuose ar kompiuterių tinkluose stebint duomenų srautą;
- Aukštesnių teisių gavimo (angl. *User to Root*, U2R) ataka – naudodamasis sistemų pažeidžiamumais programišius įgauna aukštesnes, nei paprasto vartotojo teises. Įprastai pavyksta pasiekti aukščiausias teises – super vartotojo (angl. *Super-User*, SU) teises. *Perl, xterm* yra keletas šios atakos pavyzdžių;
- Nutolusio kompiuterio užvaldymo (angl. *Remote to Local*, R2L) ta tokio tipo ataka kurioje įsilaužėlis siunčia duomenų paketus į vartotojo kompiuterį nuotoliniu būdu tikėdamasis, kad pavyks pasinaudoti kompiuteryje esančiais pažeidžiamumais. Šiai atakų klasei priskiriamos atakos kaip *xlock, guest, xnsnoop, phf* ir kiti;
- Zondas (angl. *probe*) – tai tokios atakos, kurios metu programišius skanuoja kompiuterius ar kompiuterių tinklo įrenginius norint sužinoti sistemų silpniausias, pažeidžiamiausias vietas. Radus gali jas panaudoti įsilaužimams į sistemas. Šis būdas dažniausiai naudojama duomenų gavybos (angl. *data*

mining) tikslais. Įsilaužimai (atakos) yra klasifikuojamos į 4-ias pagrindines klases [20]. Įprasti naudojami įrankiai šiam tikslui pasiekti yra *saint, portsweep, mscan, nmap,* ir kiti [21].

Cisco NetFlow tapo savotišku de-facto naudojamu standartiniu srautu paremtu (angl. *flowbased*) duomenų protokolu įmonių kompiuterių tinkluose. Apskritai srauto duomenų anomalijų analizė gali būti išskaidoma į tris pagrindinius žingsnius [22]: 1) duomenų surinkimas, 2) duomenų paruošimas, 3) detekcijos algoritmo naudojimas. Anksčiau vykdytuose tyrimuose [23], [24], [25] mokslininkai pateikė kompiuterių tinklų anomalijų charakteristikas.

Pristatyta MINDS [26] sistema siūlo beveik realaus laiko *NetFlow* srauto duomenų analizę kas 10 minučių. MINDS sistema veikia kartu su duomenų analitikos komponentu duomenų paketų apdorojimui, dėl to tai nėra praktiškai pritaikoma realiam kompiuterių tinklų srauto stebėjimui ir valdymui.

Egzistuoja ne viena kompiuterių tinklų srautinių duomenų priežiūros įrankių, tokių kaip *Nfsen* [27], *Ntop* [28] ir *Scrutinizer* [29], kurie naudoja grupinių išrikiavimo (angl. *round-robin*) realiacines duomenų bazes. Tokie įrankiai tiko, kai dauguma vartotojų srauto duomenų buvo apdorojami trumpame laiko lange, tačiau jie gali nebesugebėti tinkamai apdoroti realaus laiko užklausų didelių duomenų analizėje.

2.2 IDS, jų tipai, klasifikacija

Informacinių sistemų saugumo samprata ISO/IEC 17799:2000 standarte [30] apibrėžiama trimis komponentėmis: a) vientisumas (angl. *integrity*) – informacijos bet jos apdorojimo būdai, patikimumo, autentiškumo užtikrinimas. b) konfidencialumas (angl. *confidentiality*) – garantuojama, kad informaciją gali matyti, gauti, turėti tik tie, kas turi teisę, įgaliojimus; c) prieinamumo (angl. *availability*) užtikrinimo, kad sankcionuoti vartotojai, kuomet reikia, turi prieigą prie informacijos. Siekiant užtikrinti informacijos sistemų saugumą kompiuterių tinkluose turime taikyti įsilaužimo apsaugos techninius įrankius, techninius sprendimus, sistemas.

Tinklų įsilaužimų aptikimo sistemos (angl. *Network Intrusion Detection System*, NIDS) stebi ir analizuoja kompiuterių tinklo duomenų srautą, kuris įeina ir/ar išeina iš kompiuterių tinklo įrenginių. NIDS skirstomas į du tipus – žymėmis paremtomis (angl. *Signature-based*) ir anomalijomis paremtomis (angl. *Anomaly-based*) metodai [31]. Žymėmis paremti metodai negali aptikti naujų ar anksčiau neidentifikuotų atakų, tuo tarpu anomalija paremti gali aptikti ir anksčiau nematytas atakas. Anomalija paremti metodai taip pat geba mokytis ir automatiškai adaptuotis prie specifinių kompiuterių tinklų srautų [32]. Vis dažniau šioje srityje pritaikomi ir mašininio mokymosi algoritmai.

Tradiciniai kompiuterių tinklo saugumo sprendimai nebuvo suprojektuoti apsisaugoti nuo šių laikų modernių kibernetinių atakų. Įprastinės ugniasienės (angl. *firewalls*) ir tinklo atakų prevencijos sistemos (angl. *Network Intrusion Prevention System*, NIPS) nebėra efektyvios norint apsaugoti kompiuterių tinklą nuo

nesankcionuotų vidinių vartotojų ar išorės vartotojų (internetu). Ugniasienių paskirtis yra leisti arba blokuoti duomenų srautą keliaujantį pro jį, o IPS paskirtis yra pritaikyti tam tikrą modelį (angl. *signature*).

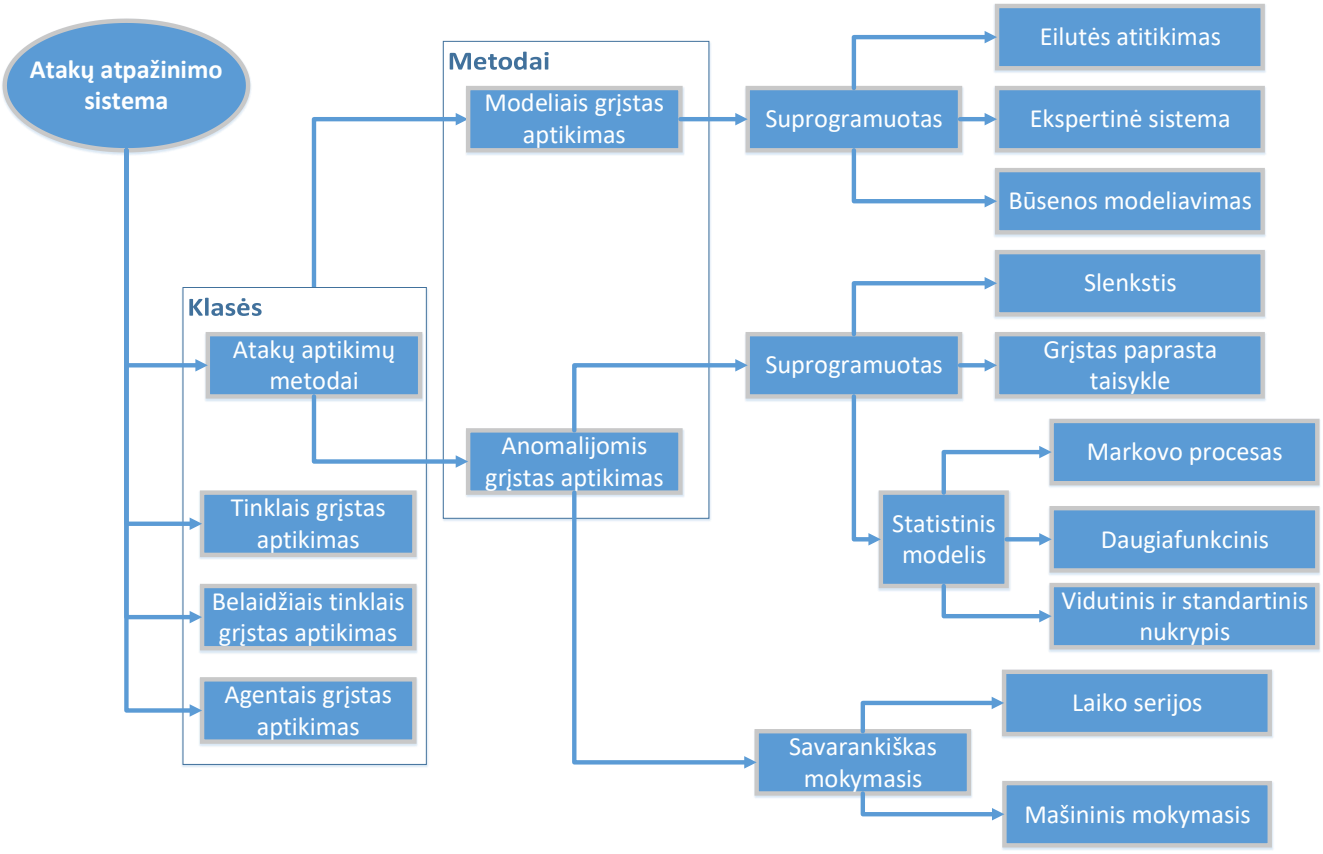
Dabartinių kibernetinių įsilaužimų į kompiuterių tinklus braižas, stilius, taktika ar modelis yra nuolatos besikeičiantis. Todėl daugelis organizacijų savo kompiuterių tinkluose naudoja kompleksinės priemones įsilaužimams aptikti: ugniasienes (angl. *firewalls*), IPS, IDS, tarpinius serverius (angl. *proxy servers*), saitynų turinio filtravimo serverius (angl. *web-content filtering server)*, antivirusinių šliužus (angl. *antivirus gataways*), saitynų aplikacijų apsaugą (angl. *Web Application Filter*, WAF). Tinklo atakų apsaugai paprastai naudojama atakų atpažinimo sistemos. Atakų atpažinimo sistema (angl. *Intrusion Detection System*, IDS) – tai programinė ar techninė įranga, skirta aptikti nepageidaujamus bandymus gauti, valdyti ir/arba išjungti kompiuterines sistemas, dažniausiai per kompiuterių tinklą (globalų ar municipalinį). IDS naudojama siekiant nustatyti kelių tipų kenksmingą elgesį, kuris gali pažeisti organizacijos kibernetinį saugumą. Tai apima kompiuterių tinklo atakas, kurių metu yra pažeidžiamos sistemų paslaugos. IDS gali būti sudarytas iš kelių komponentų: sensorių, kurie sukuria saugumo įvykius; konsolės – stebi įvykius ir valdo daviklius; procesoriaus (angl. *engine*) – kad įvykių įrašai pagal daviklių informaciją būtų registruojami duomenų bazėje ir naudojamos sistemos taisykles saugumo perspėjimams siųsti.

Atakų atpažinimo sistemos klasifikuojamos į keturias pagrindines klases (3 pav.), kurie naudojami skirtingų tipo tinkluose:

- Atakų aptikimų metodai – juose aprašomi metodai, kuriais atpažįstami įsilaužimai kompiuterių tinkle;
- Tinklais grįsta atakų aptikimo sistema (angl. *Network-based Intrusion Prevention System*, NIPS) – tai sistema, kuri stebi visą tinklą analizuodama protokolų aktyvumą;
- Belaidžiais tinklais grįsta atakų aptikimo sistema (angl. *Wireless Intrusion Prevention Systems*, WIPS) – stebi belaidžio tinklo įtartiną srautą analizuojant jų belaidžio tinklo protokolus;
- Agentais grįsta atakų aptikimo sistema (angl. *Host-based Intrusion Prevention System*, HIPS) – šioje sistemoje agentai yra įrašyta programinė įranga kompiuteriuose, kuri stebi visus duomenų srautus kurie iškeliauja ar atkeliauja į kompiuterį. Agentai neapsiriboja veikimu tik kompiuteriuose.

IDS taksonomiją kūrė daugybė autorių [33]–[39] iš skirtingų institucijų, mokslo įstaigų. Šioje mokslinėje ataskaitoje jų IDS taksonomijų elementai atvaizduojami 3 pav. Nagrinėjant paveikslą matome, kad metodai, kuriais remiantis IDS geba atpažinti atakas kompiuterių tinkluose, sudaryti iš : a) modeliais grįstais atakų aptikimo metodas (angl. *signature-based detection*), kuris sudarytas iš programuojamų eilučių atitikimo, ekspertinės sistemos ir būsenos modeliavimo algoritmų; b) anomalijomis ir statistika grįstų atakų aptikimo metodas (angl. *statistical anomaly-based detection*) sudarytas iš suprogramuotų sistemų ir savarankiško mokymosi metodų. Siekiant tiksliau, greičiau

ir veiksmingiau aptikti anomalija kompiuterių tinkluose yra naudojami dirbtinio intelekto, mašininio mokymosi metodai.



3 pav. Atakų atpažinimo sistema

2.1 IDS taikymas anomalijų aptikimui kompiuterių tinkluose

Anomalijomis grįstas aptikimas (3 pav.) yra vartotojų elgesiu paremta įsilaužimų aptikimo sistema. Ji aptinka normalaus aktyvumo sistemoje pokyčius sukonstruojant normalaus stebimos sistemos naudojimo profilį [40], [41]. Kadangi kompiuterių tinklo naudojimo profilis kuriamas tada, kai jis nėra puolamas atakų [42], atsiranda galimybė aptikti atakas, kurios yra naujos stebimoje sistemoje. Anomalijų aptikimas yra skirstomas į du tipus pagal tai, kokiu būdu yra aprašomas normalaus naudojimosi sistema profilis:

- **Savarankiško mokymosi** – tokia sistema veikia pagal tai, koks nustatytas standartinis normalaus naudojimosi profilis. Tai yra pasiekiama kuriant modelį su stebėtu sistemos srautu per tam tikrą nustatytą laiko periodą *[34]*. Tokios sistemos skirstomos ir dar smulkesnes kategorijas: laiko eilučių modelius ir mašininį mokymąsi:
    - **Laiko serijos modelis** naudoja nuoseklių stebėjimų, vykstančių tolygiais intervalais, seką. Jeigu naujo stebinio atsiradimo tikimybė tam tikru laiko momentu yra labai nedidelė, tuomet tai yra laikoma pokyčiu nuo normalaus elgesio. Laiko eilučių modelis turi privalumą

aptikti elgesio tendencijas tam tikrame laiko intervale bei pažymėti nuokrypius nuo normalaus elgesio. Šis modelis tiksliai veikia, kai atakos yra nuosekliai išsidėsčiusios laike *[43]*. Tačiau šis modelis reikalauja nemažai skaičiavimo resursų *[44]*. Automatinis slenkančio vidurkio regresijos (angl. *auto regressive moving average*, ARMA) modelis yra vienas iš laiko eilučių modelis, naudojamas IDS sistemose.

Elike Hodo *[45]* pritaikė generalizuotą autoregresinį slenkančių vidurkių (angl. *generalised autoregressive moving average*, GARMA) ir ARMA laiko eilučių modelius identifikuoti 4 atakų tipus tinkluose: DoS, zondo, U2R ir L2R. Modelių parametrai (ARMA(1,1) ir GARMA(1,2; ,1)) buvo nustatyti naudojant *Hannan-Rissanen* algoritmą, *Whittle* įvertį ir maksimalaus tikėtinumo įvertį. Prognozė gauta naudojant *Whittle* įvertį kartu su maksimalaus tikėtinumo įverčiu buvo arčiausiai originalios vertės. Laiko eilučių modeliai gebėjo prognozuoti atakas, tačiau GARMA parodė geresnius atakų aptikimo rezultatus.

- **Mašininio mokymosi modelis** taikomas norint leisti kompiuteriams „mąstyti“, t. y., kompiuterių sistemų mokymasis iš savo patirties, o ne taikant tiesioginio problemos sprendimą. Šis modelis remiasi statistika, nes mokymasis vyksta nagrinėjant jam pateiktus duomenų įrašus.

- **Suprogramuotas** – toks modelis yra apmokomas aptikti pakitimus vartotojų elgesyje paties vartotojo arba kito išorinio asmens. Vartotojas nusprendžia nenormalaus elgesio sistemoje apimtį ir pats pažymi įsilaužimo grėsmę [46]. Suprogramuoti modeliai grupuojami į tris kategorijas: slenkstiniai, paprasti taisyklėmis paremti ir statistiniai.
  - **Slenkstiniai modeliai** laikomi paprasčiausiais suprogramuotais aprašomaisiais statistiniais detektoriais [34]. Analizuojant statistinių duomenų koreliacijas vartotojas gali suprogramuoti sistemą siųsti pavojaus signalą tik pasiekus tam tikrą iš anksto nustatytą slenkstinę statistinio kintamojo vertę. Atsargus slenkstinės ribos parinkimas reikalingas siekiant sumažinti klaidingų pavojaus signalų dažnį. Per aukštos slenkstinės vertės parinkimas gali padidinti riziką praleisti įsilaužėlio kenkėjiškus veiksmus [47]. Dažnas pavyzdys yra pavojaus signalo išsiuntimas, kai vartotojas tris kartus iš eilės nesėkmingai bando prisijungti prie sistemos [39].

    Ke Wang *et al*. [48] pristatė kompiuterių tinklo apkrova paremtą anomalijų aptikimą ant 1999-ų metų DARPA duomenų rinkinio ir realaus laiko duomenų, surinktų iš JAV Kolumbijos Kibernetinio Saugumo departamento kompiuterių tinklo naudojimo. Apmokymo fazėje buvo apskaičiuoti bitų dažnio pasiskirstymas ir aplikacijos

apkrovos vienam mazgui ir prievadui standartinis nuokrypis. Per aptikimo fazę *Mahalanobio* atstumas naudotas apskaičiuoti naujų duomenų panašumą į prieš tai apskaičiuotą profilį. Detektorius palygina apskaičiuotą panašumą su nustatyta slenkstine verte ir sukuria pavojaus signalą, kai naujų duomenų skirtumas nuo normalių duomenų viršija slenkstinę ribą. Sukurtas modelis pademonstravo beveik 100 % tikslumą su 0,1 proc. klaidingai teigiamų spėjimų 80-o prievado (HTTP) srautui.

- **Grįsti paprastomis taisyklėmis paremti** modeliai tikrina, ar sistemoje vykstantys įvykiai atitinka taisykles, nurodančias, kas yra normalus vartotojo elgesys kompiuterių tinkle. Pagrindinis tokių sistemų trūkumas – jos neaptinka grėsmių, kurios nėra aprašytos taisyklėmis [49]. RIPPER (pakartotinis laipsniškas genėjimas, siekiant sumažinti klaidos dydį, angl. *repeated incremental pruning to produce error reduction*) yra vienas iš taisyklėmis paremtų modelių pavyzdžių, kuris kuria taisykles, padedančias aptikti normalų ir neįprastą elgesį kompiuterių tinkluose. Naidu *et al.* [50] panaudojo 1999-ų metų *KDDCup* duomenų rinkinį siekiant palyginti RIPPER, sprendimų medžių (C5) ir atraminių vektorių mašinų (angl. *support vector machine*, SVM) veikimą. Duomenų rinkinys buvo suskirstytas į 3 kategorijas: normalus srautas, zondo ir DoS atakos. RIPPER algoritmas testuotas per dvi eksperimento stadijas. Pirmoje stadijoje buvo inicijuojamos taisyklių sąlygos, o antroje stadijoje – taisyklių optimizavimas. Algoritmas kiekvienai taisyklei gavo sąlygas testavimo duomenų klasifikavimui. RIPPER algoritmas aptiko panašų kiekį anomalijų kaip ir kiti modeliai: RIPPER – 98,69 %, C5 - 98,75 % ir SVM - 98,63 %.
- **Statistiniai modeliai** renka duomenis ir sukuria vartojimo profilį. Analizuojant normalaus elgesio statistinį profilį yra pateikiama aprašomoji analizė ir dėsningumai, kurie padeda padaryti išvadas, ar elgesys tinkle yra normalus ar neįprastas. Tuomet sistema sukuria atstumo vektorių tarp stebimo srauto ir statistinio profilio. Pavojaus signalas sukuriamas, kai pasiekiamas iš anksto nustatytas atstumas [32]–[34], [51] . Šio tipo modeliai yra kategorizuojami į keturias kategorijas: vidurkio/standartinio nuokrypio, daugiamatis, Markovo proceso ir eksploatacinis (angl. *operational*). Dorothy Denning [52] aptarė modelį, paremtą hipoteze, jog saugumo pažeidimas gali būti aptinkamas stebint sistemos audito įrašus ir ieškant dėsningumų pokyčių. Toks modelis naudoja vartotojų elgseną atspindinčius profilius, kurie parodo metrikas ir statistinius modelius bei taisykles, aprašančias audito įrašuose esantį elgesį.
- **Vidutinio ir standartinio nuokrypio** ir kitos koreliaciją žyminčios metrikos statistikoje yra vadinamos momentais [46], [53]. Yra sakoma,

jog momentas yra anomalus, kai įvykiai patenka aukščiau ar žemiau nustatyto intervalo. Sprendimai yra daromi atkreipiant dėmesį į sistemos pokytį pakeičiant statistinių taisyklių grupę sistemoje [46]. Tokio modelio privalumas lyginant su eksploataciniu yra jo gebėjimas aptikti atakas neturint jokios išankstinės informacijos apie normalų tinklo elgesį. Jis mokosi iš stebėjimų ir pats nustato normalumo ribas. Tai yra sudėtingas modelis, tačiau tuo pačiu jis yra lankstesnis už slenkstinį modelį. Keičiant vidurkį ir standartinį nuokrypį nežymiai pakeičia skaičiavimus pridedant papildomų svorių naujausiems duomenims ;

- **Daugiafunkciniai modeliai** yra panašūs į vidurkio ir standartinio nuokrypio modelius [46], [54]. Šie modeliai yra paremti koreliacijomis tarp dviejų ar daugiau metrikų ir naudoja kelis kintamuosius galimų rezultatų spėjimui. Pavyzdžiui, CPU ciklų skaičius gali būti palyginamas su prisijungimo sesijos trukme. Teoriškai šis modelis galėtų gerai atskirti vieną kintamąjį [54]. Sha ir kiti [55] pasiūlė daugiamačių laiko eilučių ir aukštesnio laipsnio *Markovo* grandinių apmokymo ir testavimo algoritmus, kurie buvo įvertinti naudojant DARPA duomenų rinkinį. *Markovo* grandinių modelio rezultatai parodė, jog santykinės pozicijos tarp skirtingo laipsnio modelių rezultatų leidžia naują ir efektyvų anomalijų nustatymą. Siekiant padidinti jautrumą kelios sekos buvo apjungtos kaip daugiamatės į vieną paprastą modelį;
- **Markovo procesas** taikomas dviem metodais: *Markovo* grandinės ir paslėpti *Markovo* modeliai. *Markovo* modelis yra tarpusavyje susijusių baigtinių būsenų rinkinys, dalyvaujantis stochastiniame topologijos ir modelių galimybių nustatymo procese [32]. Kiekviena proceso stadija priklauso nuo praėjusios stadijos rezultatų. Anomalijos yra aptinkamos lyginant susijusią tikimybę, nustatytą procesui su nekintančia slenkstine verte. Tai suteikia privalumą aptikti neįprastus pasikartojančius įvykius [54]. Paslėptas *Markovo* modelis taria, jog sistema yra *Markovo* procesas, kuriame stochastiniai procesai su baigtinėmis galimų rezultatų būsenomis, yra paslėptas [32]. Ye Nong [56] pristatė anomalijų aptikimo metodą, kuri naudoja *Markovo* grandinių modelį įsilaužimų aptikimui. Šiame modelyje *Markovo* grandinės buvo naudojamos pavaizduoti laikiną normalaus elgesio profilį kompiuterių tinkle. Normalaus vartojimo profilio Markovo grandinės modelis yra išmokstamas iš istorinių normalaus sistemos elgesio duomenų. Stebimas sistemos elgesys yra analizuojamas ir žiūrima, ar normalaus elgesio *Markovo* grandinės modelis atitinka stebimą kompiuterių tinklo elgesį. Maža atitikimo tikimybė parodo, jog yra stebimas anomalus elgesys, reiškiantis įsilaužimą. Šis būdas buvo pritaikyta *Sun Solaris*

sistemoje ir tobulai atskyrė normalų kompiuterių tinklo elgesį nuo atakų.

Modeliais grįstų aptikimas nusako taisyklių rinkinį, kuris naudojamas dėsningumams aptikti kompiuterių tinklo sraute. Jeigu aptinkamas nesutapimas, tuomet yra sukuriamas pavojaus signalas [57]. Toks anomalijų aptikimo būdas turi privalumą prieš kitus, nes sugeba aptikti atakas su labai mažu klaidingai teigiamų atvejų dažniu [58]. Didžiausias trūkumas – aptinka tik tokias atakas, kurios yra įrašytos į duomenų bazę [59]. Modeliais paremta anomalijų aptikimo sistema yra suprogramuota naudojant atskiras sprendimų taisykles. Tokios taisyklės yra tiesmukai suprogramuotos aptikti įsilaužimą. Programavimas vyksta keturiais skirtingais būdais – būsenos modeliavimas, ekspertinė sistema, simbolių eilutės atitikimas.

2.2 IDS išvengimo metodai

Kibernetiniai nusikaltėliai tampa vis gudresni ir tam, kad išvengtų įsilaužimo aptikimo sistemas (IDS) naudoja tam tikrus metodus. Literatūros šaltiniuose randami keturi pagrindiniai IDS išvengimo metodai: fragmentacija, užtvindymas (angl. *flooding*), sumaišymas (angl. *obfuscation*) ir šifravimas.

- Fragmentavimo metodas – kompiuterių tinklo duomenų paketus suskaido į smulkesnius paketus. Šie smulkesni paketai yra iš naujo surenkami gavėjo mazge IP sluoksnyje prieš perduodant jį į aplikacijos sluoksnį. Kad fragmentuotas srautas būtų taisyklingai išanalizuotas, kompiuterių tinklo detektorius turi surinkti šiuos smulkius paketus kuo panašiau į tai, kokiu būdu jie buvo sufragmentuoti. Kad paketai būtų taisyklingai restruktūrizuoti, reikalinga daug duomenų laikyti atmintyje ir tikrinti srauto panašumą į modelius, esančius duomenų bazėje. Atakos yra paslepiamos ir atrodo kaip normalus srautas. Tam pasiekti įsilaužėliai naudoja fragmentavimo persidengimą, perrašymą ir laiko pasibaigimą (angl. *timeout*) [60], [61]. Fragmentavimo ataka pakeičia fragmentuotų paketų sudedamąsias dalis nauja informacija, kuri padeda sukurti kenkėjišką paketą. 4 pav. rodo fragmentų perrašymą. Tarkime 3-asis paketo fragmentas sukuriamas įsilaužėlio. Kompiuterių tinklo įsilaužimų detektorius turi išlaikyti tokią pačią visų stebimo tinklo paketų būseną. Tokio palaikymo trukmė gali būti trumpesnė už laiką, per kurį duomenys pasieks gavėją [62]. Atakos kūrėjas stengiasi pasinaudoti visais IDS trūkumais ir stengiasi siųsti fragmentuotus paketus, paskirstytus laike.

4 pav. Paketų fragmentų perrašymas

- Užtvindymas – įsilaužėjas pradeda ataką sutrikdant detektorių, kas pereina prie kontrolės mechanizmo neveikimo. Kai neveikia detektorius, visas srautas yra praleidžiamas [61]. Užtvindymui dažniausiai yra imituojami transporto lygmens (angl. *User Datagram Protocol*, UDP) ir tinklo valdymo (angl. *Internet*

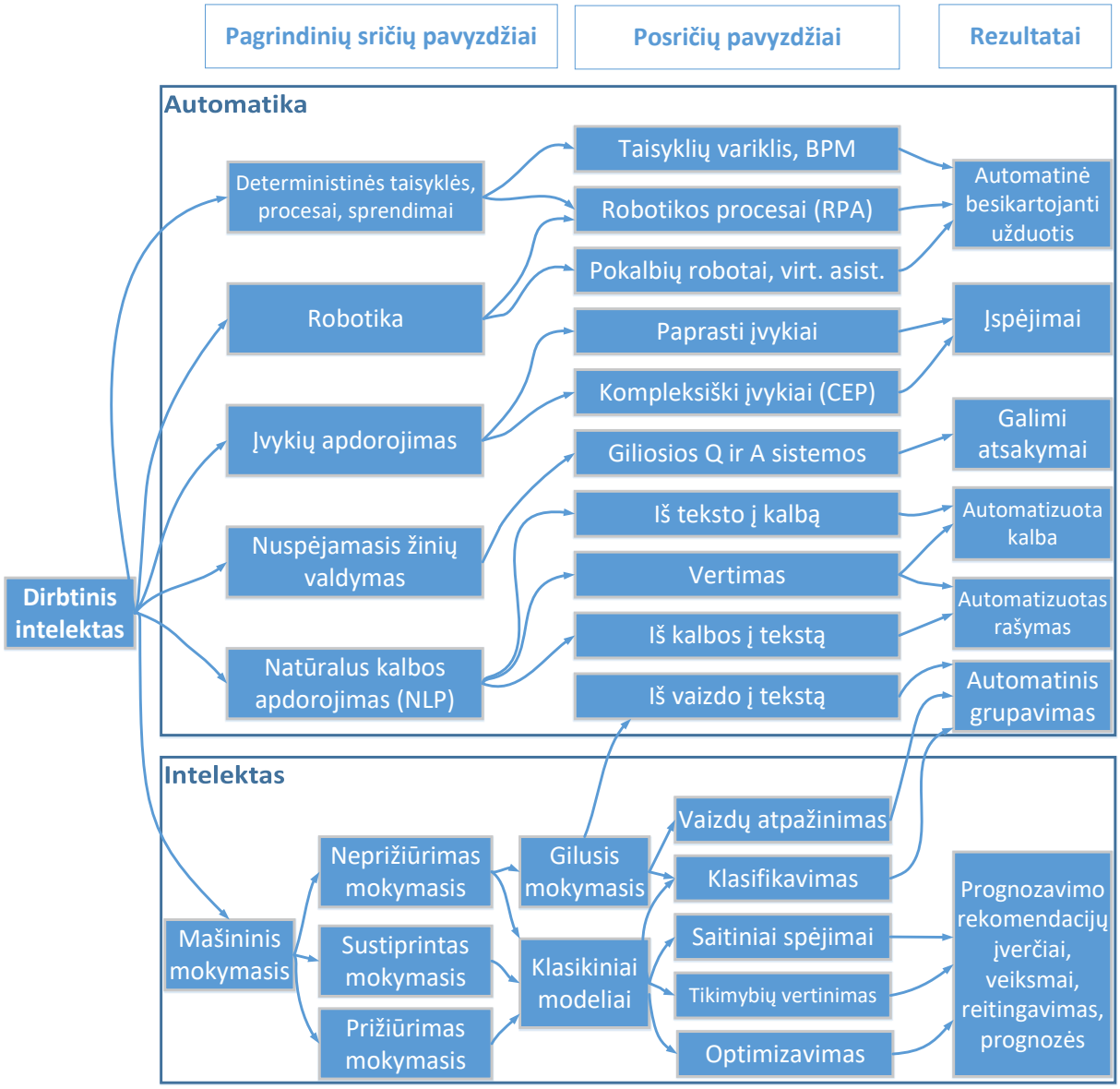
Control Message Protocol, ICMP) protokolai. Šis metodas yra taikomas maskuoti kenkėjišką kibernetinio nusikaltėlio veiklą. Dėl šios priežasties IDS sunkiai pastebi tokius neteisingus paketus dideliame sraute.

- Sumaišymo būdas paslepia ataką padarant perduodamą žinutę visiškai nesuprantama [63]. Šiuo atveju yra keičiamas programinis kodas paliekant tokį patį funkcinį veikimą, kad būtų sumažinamas aptikimas. Toks metodas leidžia išvengti IDS bandant išnaudoti modeliais paremtos įsilaužimų aptikimo sistemos trūkumus. Efektyvi IDS turėtų suprasti šešioliktainį atakų modelių užkodavimo formatą [64]. UTF-8 koduotė leidžia užkoduoti vieną simbolį keliais skirtingais formatais. Kibernetiniai nusikaltėliai dažnai naudoja dukart užkoduotus duomenis, eksponentiškai didinant modelių skaičių, reikalingą aptikti ataką. SIDS paremta modelių sutapimu kenkėjiškam elgesiui aptikti, kur modeliai yra sukuriami žmonių ekspertų, kurie išverčia kenkėjišką veiklą iš mašininio kodo į simbolinę kalbą. Tačiau šis sumaišymo metodas dažniausiai įsilaužėliams padeda išvengti IDS.
- Šifravimas užtikrina duomenų konfidencialumą, vientisumą ir privatumą. Kenkėjai naudoja šiuos saugumo atributus, kad išvengtų aptikimo ir paslėptų atakas, kurios gali būti nukreiptos į kompiuterines sistemas. Pavyzdžiui, šifravimo protokolas (pvz. HTTPS) negali būti nuskaitytas IDS – ji negali palyginti užšifruoto protokolo su duomenų bazėje esančiu modeliu [65]. Dėl to detektoriai sunkiai aptinka atakas, užslėptas tokiuose paketuose [66]. Statistiniai interneto srauto požymiai, kurie neparemti paketo vidine sudėtimi, leidžia mokslininkams pritaikyti juos atakų aptikimui [67]. Dėl to tokia kenkėjiška veikla gali būti atskirta nuo normalaus srauto.

3 Dirbtinis intelektas, klasifikacija, taikymas

Mokslininkas Jonas Makartis (angl. *John McCarthy*) sukūrė sąvoką „Dirbtinis intelektas – inžinerijos ir mokslo šaka kurianti intelektualias, mąstančias, protaujančias mašinas, ypatingai intelektualią programinę įrangą. Tai susiję su kompiuterių naudojimusi žmogaus intelektui suprasti, tačiau dirbtinis intelektas neturi apsiriboti tik biologiniais metodais“. Šis terminas pirmą kartą buvo pasakytas 1956-ais metais du mėnesius trukusioje vasaros stovykloje „Dartmouth College in Hanover“ Naujajame Hampšyre (angl. *New Hampshire*). Jos metu dešimt mokslininkų bandė spręsti klausimus: kokią tam tikrą kalbą turi naudoti mašinos?; kaip mašinos gali mąstyti sprendžiant uždavinius, kurie būdingi žmonėms?; kaip mašinos gali pačios save tobulinti ir apmokyti?

Kompiuterių tinklų saugumas gali būti užtikrintas naudojant dirbtinio intelekto metodus, nes įprastinės kompiuterių tinklo saugumo priemonės nebetinka nuolatos besikeičiančių atakų tipams, dėsningumams. Šiame skyriuje bus nagrinėjama kodėl ir kaip yra taikomi tam tikros dirbtinio intelekto sritys siekiant apsaugoti kompiuterių tinklus nuo įsilaužimų. Bus apžvelgti mašininio mokymosi metodai, kuriuos sukūrė, pritaikė tinklo duomenims ir aprašė mokslininkai savo tyrimuose.



5 pav. Dirbtinio intelekto taikymo sritys

Dirbtinis intelektas yra taikomas daugelyje sričių (5 pav.). Pagrindinės taikymo sritys yra dvi: automatikos ir intelekto. Automatikos srityje plačiausiai naudojami natūralios kalbos apdorojimo (angl. *Natural Language Processing*, NLP), robotikos ir nuspėjamų žinių valdyme. Automatikos srityje naudojami paprasti ar kompleksiniai įvykiai taikomi finansų ir rinkodaros srityse, robotikoje. Taip pat pabrėžtina, kad vertimai, kalbų vertimas į tekstus ir vaizdų apdorojimas į tekstus priskiriami dirbtinio intelekto automatikos taikymo sričiai. Pagrindiniai intelekto taikymo sričių rezultatai yra gauti prognozavimo rekomendacijų įverčius, reitingavimui, prognozavimui. Intelekto taikymo srityje naudojamas mašininis mokymasis su jam priskiriamomis trejomis paradigmomis. Gilusis mokymasis (angl. *Deep Learning*, DL) taikomas ir automatikai ir intelektui. Gilusis mokymasis priklauso dirbtinio intelekto grupės, mašininio mokymosi srities metodų grupei. Kai paprastas neuroninis tinklas imituoja vienpusį tiesioginį neuronų jungimąsi ir vadinamas sklidimo pirmyn neuroniniu tinklu, tai tinklai, sudaryti iš daugiau nei vieno sluoksnio dirbtinių neuronų, kuriuose galimos

tik tiesioginio sklidimo jungtys iš įėjimo į išėjimą, vadinami daugiasluoksniu tiesioginio sklidimo neuroniniu tinklu arba giliuoju neuroniniu tinklu. DL metodu modeliuojami sudėtingi netiesiniai ryšiai tarp duomenų objektų. Giliojo neuroninio tinklo architektūra leidžia duomenų objektus išreikšti sluoksnine neskaidomų dalių kompozicija. Gilesni sluoksniai naudoja požymius, gautus iš žemesnių lygių. Klaidos sklidimo atgal algoritmas daugiasluoksniam sklidimo pirmyn neuroniniam tinklui naudoja gradientinio nusileidimo mokymo strategiją. Algoritmą sudaro du žingsniai: 1) įėjimo reikšmių tiesioginis sklidimas tinkle į išėjimo sluoksnį; 2) klaidos vektoriaus sklidimas tinkle atgal iš išėjimo sluoksnio į įėjimo sluoksnį.

Pagrindinis dirbtinio intelekto tikslas yra sukurti technologiją, kuri leistų kompiuteriams ir mašinoms funkcionuoti kaip protingoms būtybėms. Didžiausias problemas kuriant intelektą būtų galima skirstyti į šias pagrindines problemų/kūrimo kategorijas:

- Problemų sprendimas, samprotavimas – siekiama atkartoti žmogaus samprotavimą bei problemų sprendimą, kurį žmonės naudoja, kuomet sprendžia galvosūkius arba daro logines išvadas;
- Žinių pateikimas – tai žinių/išminties inžinerijos sritis. Visapusiškas žinias (angl. comprehensive commonsense knowledge), tokias kaip objektai, savybės, kategorijos, objektų ryšiai, situacijos, įvykiai, būsenos ir laikas, priežastys ir efektai, žinių išmanymas (ką mes žinome apie tai, ką žino kiti) ir daug kitų žinių yra bandoma sutelkti į dirbtinį intelektą;
- Planavimas – intelektualios sistemos turi užsibrėžti tikslų ir juos pasiekti. Taip pat turi nuspėti ateityje vyksiančius dalykus ir jų rezultatą;
- Mokymasis – mašininis mokymasis (fundamentali DI sąvoka) yra mokslas apie kompiuterinius algoritmus, gebančius automatiškai tobulėti naudojant patirtį;
- Natūralios kalbos apdorojimas – ši savybe leidžia dirbtiniam intelektui perskaityti ir suprasti žmonių kalbą. Tokia savybė leidžia vartotojui sąveikauti su dirbtiniu intelektu, šiam suprantant žmonių kalba įvestas užduotis;
- Suvokimas – tai yra savybė panaudojant duomenis iš įvairių jutiklių (kamerų, mikrofonų, belaidžių signalų, lidarų, sonarų, radarų) daryti išvadas apie supančią aplinką. Dažniausiai pritaikymas – kalbos, veidų bei objektų atpažinime;
- Judėjimas ir manipuliacija – daugelis pramoninių robotų naudoja DI, leidžiantį tikslius judesius. Robotams vis dar sunku judėti dinamiškoje aplinkoje;
- Socialinis intelektas – kuo robotai daugiausiai skiriasi nuo žmonių? Emocijų nebuvimu, tad tai yra vienas iš aspektų, siekiant panaudoti DI robotų socialiniams įgūdžiams įdiegti;
- Bendrasis intelektas – tokia savybė leistų dirbtiniam intelektui veikti kuo įvairiausiose srityse, apjungiant gebėjimus veikti specifinėms, siauroms užduotims atlikti, galbūt net pralenkiant žmogaus savybes.

3.1 Mašininis mokymasis

Mašininis mokymasis atsirado norint nustatyti dėsningumą ir pagrinde remiasi prielaida, kad mašinos turėtų intelektą, kuris leistų joms samprotauti, mąstyti, mokytis iš savo patirčių, sukauptų žinių, gebėtų prisitaikyti prie juos supančios aplinkos. [68] Didėjant sugeneruotų duomenų perdavimo kiekiui kompiuterių tinkluose, vis labiau augančių išmaniųjų miestų, kritinės infrastruktūros, komunikacinių tinklų, atsiranda būtinybė analizuoti didelių duomenų kiekius pasitelkiant mašininį mokymąsi. Finansinės institucijos, valstybinės įstaigos, sveikatos priežiūros, technologijų, rinkodaros, aplinkosaugos ir pramogų sektoriai, ir daugelis kitų sektorių įsidiegia į savo turimas sistemas mašininio mokymosi modelius, algoritmus. Naudojant mašininio mokymosi algoritmus sistemos gali priimti sprendimus be žmogaus įsikišimo, prognozuoti dinaminės sistemos ar žmogaus elgsenas. Tarkime naudodami mašininį mokymąsi įgaliname sistemą suvokti visus socialinius santykius tarp individų ir galime atpažinti kiekvieno individo veidą, rašymo stilių, kalbą, judesius. Taip pat belaidžio ryšio sistemose, kaip daiktų internete (angl. *Internet of Things*, IoT) gali būti naudojamas mašininis mokymasis didelių duomenų analizėms atlikti. Mašininio mokymosi užduotys dažnai priklauso nuo pateikiamų duomenų tipo. Mokymasis mašininiame mokyme yra toks procesas, kuriame mašina apmokoma pasiekti tam tikrus specifinius tikslus kaip tarkime balso atpažinimą, objektų atradimą nuotraukose. Kitais žodžiais tariant, mokymasis leidžia mašininio mokymosi sistemai atpažinti potencialius sąryšius tarp įvesčių ir išvesčių duomenų be iš anksto apibrėžtų taisyklių. Bendrai egzistuoja keturios pagrindinės mokymo paradigmos:

a) prižiūrimo mokymosi algoritmai (angl. *supervised learning*) – šie algoritmai yra naudojami taikomosiose programose apmokant naudojantis žymėtus (angl. *labeled*) duomenis. Kai mokymuisi yra naudojami sužymėti duomenys, tuomet tiek įvesties, tiek išvesties duomenis yra žinomi sistemai. Prižiūrimo mokymosi algoritmai plačiausiai buvo naudojami sistemose, kuriuose neturime daug sukauptų duomenų. Šiais laikais yra metodų, kuriuos taikydami nebūtina turėti sužymėtus duomenis. Kaip pavyzdys yra pateiktas šioje ataskaitoje *LO-shot* mokymasis pirmame skyriuje.
b) neprižiūrimo mokymosi algoritmai (angl. *unsupervised learning*) – priešingai nei prižiūrimo mokymosi algoritmuose, šiuose algoritmuose mokymasis vyksta be sužymėtų duomenų. Neprižiūrimo mokymosi tikslas yra išnagrinėti visus įvesties duomenis ir rasti struktūrą kaip išvadą, kuri būtų susijusi su nežymėtais duomenimis.
c) hibridinis mokymasis (angl. *semi-supervised learning*) – tai mokymasis, kuris naudojamas taikomosioms programoms turint sužymėtus ir nesužymėtus duomenis. Šio mokymosi būdas gali būti taikomas su klasifikacijos, regresijos ar prognozių metodais. Hibridinis mokymasis naudingas tuomet, kuomet mokymuisi su duomenimis, kurie visi yra sužymėti, apskaičiavimo kaštas laiko atžvilgiu yra palygintinai per dideli.
d) sustiprintas mokymasis (angl. *reinforcement learning*) – priešingai nei visuose anksčiau aptartuose metoduose, sustiprintame mokymesi nėra naudojami

sukaupti duomenys. Sustiprinto mokymosi metode mokymasis vyksta iš duomenų, kurie yra gaunami vykstančiame mokymosi procese. Tai yra, sustiprinto mokymosi metodo tikslas yra mokytis iš jį supančios aplinkos veiksnių. Pagrindinis šio mokymosi tikslas yra rasti strategiją, pagal kurią programinės įrangos agentai elgdamiesi skirtingose aplinkose maksimaliai padidintų atlygio taškų (angl. *reward*) skaičių. Sustiprinto mokymosi algoritmai yra naudojami kompiuteriniuose žaidimuose, robotikoje ir navigacijoje *[69]*. Tam, kad būtų atlikti šie mokymosi uždaviniai buvo sukurtos kelios struktūros. Viena iš šių struktūrų, dirbtiniai neuroniniai tinklai *[70]*, yra svarbiausia mašininio mokymosi atrama. Nes jie geba atkartoti žmogaus intelektą, modeliuoti sudėtingiausius kompleksinius sąryšius tarp įvesčių ir išvesčių, rasti dėsningumus duomenyse, arba parodytų statistines struktūras iš nežinomų pasiskirstytų stebimų duomenų.

3.2 Mašininio mokymosi metodai ir jų taikymas

Požymių išrinkimo metodai padeda sumažinti požymių dimensijas, kas leidžia sumažinti sprendimų priėmimo fazės trukmę. Ambusaidi ir kiti [3] sukonstravo įsilaužimų aptikimo sistemą, kuri naudoja mažiausių kvadratų atraminių vektorių mašiną (angl. Least square support vector machine, LS-SVM) ir lankstų bendra informacija paremtą požymių atrinkimo metodą (angl. *Flexible mutual information based feature selection method*, FMIFS). LS-SVM buvo pritaikytas, nes jis yra generalizuotas klasifikavimo metodas ir reikalauja nedaug skaičiavimo resursų lyginant su kitais standartiniais SVM algoritmais [71] FMIFS metodas parenka klasifikavimui optimalius požymius, kurie gali būti tiesiškai arba netiesiškai priklausomi tarpusavyje. Šis metodas leido sumažinti klasifikatoriaus naudojamų požymių skaičių iki penkių naudojant *kyoto2006+* duomenų rinkinį tokiu būdu padedant sutaupyti mokymosi ir testavimo trukmę.

Bankovic ir kiti [72] pristatė piktnaudžiavimo (angl. *misuse*) aptikimo sistemą, paremtą genetiniu algoritmu (GA). Kad būtų galima apdoroti kompiuterių tinklo srauto duomenis realiu laiku, duomenų dimensijos turi būti sumažinamos pagrindinių komponenčių analize (angl. Principal component analysis, PCA).

Wattanapongsakorn N. [73] naudojo C4.5 sprendimų medžių algoritmą realaus laiko įsilaužimų aptikimui. Viso naudota tik 12 duomenų srauto požymių, tačiau požymių atrinkimo metodas nebuvo paminėtas.

Kang ir Kim [74] sukūrė vyniojimu paremtą (angl. *wrapper-based*) požymių atrinkimo metodą, kuris padeda atrinkti svarbiausius požymius, kuriuos naudos IDS. Šis metodas naudoja lokalios paieškos meta heuristinį algoritmą optimaliam požymių rinkiniui atrasti. Tuo tarpu daugiasluoksnis perceptronas (angl. *Multilayer perceptron*, MLP) pritaikytas siekiant įvertinti sukurto metodo veikimą naudojant NSL-KDD duomenų rinkinį.

Vis dėlto, požymių dimensijų sumažinimas naudojant požymių išrinkimo metodus nėra geriausias būdas realaus laiko IDS našumui padidinti. Jeigu yra drastiškai sumažinamas požymių skaičius, tuomet, logiška, jog nukenčia modelio tikslumas ir

preciziškumas. Jeigu požymių skaičius yra sumažinamas nežymiai, tuomet realaus laiko įsilaužimų aptikimo našumas taip pat nežymiai sumažėja.

Kiti mokslininkai adaptavo didelių duomenų analizės metodus IDS sistemoms, kad jos galėtų efektyviai analizuoti didelius realaus laiko duomenų kiekius. Fontugne ir kiti [75] sukūrė *Hashdoop* – programa, kuri išlaiko erdvinę ir temporalinę kompiuterių tinklo srauto struktūras išskaidant srautą su maišos funkcija. Šiuo atveju *MapReduce* modelis, sukurtas Dean ir Ghemawat [76] gali suskaidyti originalų srautą į mažesnes dalis nepakeičiant statistinės informacijos. Tyrimo rezultatai parodė, jog *Hashdoop* pagerino žemiausio detektoriaus greitį net 15 kartų. Lee ir Lee [77] pristatė lengvai pritaikomo įvairaus dydžio sistemoms interneto srauto matavimo ir analizės schemą panaudojant *Hadoop,* kuris gali apdoroti didelius kiekius *libpcap* failų. Rathore ir kiti [78] pristatė keturių sluoksnių realaus laiko IDS architektūrą, kuri sudaryta iš užfiksavimo sluoksnio, filtravimo ir apkrovos balansavimo sluoksnio, *Hadoop* sluoksnio ir sprendimų priėmimo sluoksnio. Eksperimentai neprisijungus prie realaus kompiuterių tinklo naudoti siekiant palyginti modeliavimo laiko ir sprendimų priėmimo pokyčius pritaikant skirtingus mašininio mokymosi algoritmus kuriamoje sistemoje. Vis dėlto, didelių duomenų metodai kaip *RAW* dažniausiai saugo paketus kaip nepadorotus srauto duomenis paskirstytoje failų sistemoje ir tik vėliau juos apdoroja [75].

2014-ais metais atliktas tyrimas [79] pasiūlė realaus laiko anomalijų detekcijos sistemą, paremtą *Apache Storm*, kuri naudoja mašininio mokymosi algoritmus, tokius kaip „k-NN“ [80] ir „dažnas algoritmas“ (angl. *frequent*) [81], Top-N anomalijų veiklai aptikti. Pastebėtina, jog ši sistema negeba atlikti realaus laiko analizės duomenims, atkeliaujantiems iš kelių duomenų šaltinių vienu metu. Tai paaiškinama šio įrankio duomenų analizės apdorojimo principu – jis vyksta neprisijungus prie kompiuterių tinklo, t. y., ne realiame laika. Paskirstytojo kompiuterių tinklo matavimo sistema [82] buvo pritaikyta naudojant paraleliuosius ir paskirstytuosius mašininio mokymosi algoritmus (*Apache Mahout* [83]). Tačiau *Mahout* sprendimas nepalaiko realaus laiko duomenų analizės dėl to, jog šis įrankis yra paremtas *Hadoop* failų sistema (HDFS).

Ankstesni tyrimai rodo, jog mašininio mokymosi algoritmai užima svarbią vietą aptinkant anomalijas kompiuterių tinkluose [84], [85]. Daugelis mašininio mokymosi metodų, tokie kaip *k*-artimiausio kaimyno algoritmai [86], neuroniniai tinklai [87], atraminių vektorių mašinų [88] ir k-vidurkio klasterizavimą [89] buvo pritaikyti anomalijoms atpažinti. Nepaisant to, daugelis atliktų tyrimų yra grįsti iš anksto sukauptų paketų apmokymu, o ne remiantis realiu laikų kompiuterių tinklo paketais.

Nepaisant to, jog visi prieš tai įvardinti sprendimai siūlo žymių mastelio keitimo patobulinimų, vis dėlto jie negeba realiu laiku fiksuoti įsilaužimų, nes duomenys pirmiausia yra sukaupiami ir tik tada apdorojami. Kadangi kompiuterių tinklo duomenys kaupiami nuolat, kai tinklas yra naudojamas, sunku užtikrinti tokių duomenų efektyvų apdorojimą.

4 Apibendrinimas

Išanalizavus aprašytų literatūros šaltinių, mokslinių straipsnių darbus pastebime, kad norint apsaugoti kompiuterių tinklus nuo įsilaužimų, visuose OSI modelio lygiuose, turime naudoti įsilaužimų apsaugos technikas, metodus, įrenginius. Įsilaužimų aptikimo sistemos (IDS) yra naudojamos kompiuterių tinklų saugume ir jų naudojimas tampa vienas iš pagrindinių priemonių kibernetinių atakų apsaugai. Daugelis IDS generuoja daugybę klaidingai teigiamų (angl. *false positive*) pranešimų, dėl galimai esančio įsilaužimo. Tad yra būtina taikyti geresnius metodus ir dėl klaidingų spėjimų, informavimo tinklo saugumo administratorius.

Kompiuterių tinklų saugumas gali būti užtikrintas naudojant dirbtinio intelekto metodus, nes įprastinės tinklų saugumo priemonės nebetinka nuolatos besikeičiančių atakų tipams, dėsningumams. Mašininio mokymosi algoritmai gali būti naudojami daugeliui kibernetinių atakų aptikimui ir sustabdymui, tačiau tam reikia didelių kompiuterinių technologijų resursų dideliems skaičiavimams atlikti. Mašininio mokymosi metodų vystymas yra aktualus ir prasmingas siekiant aptikti įsilaužimus kompiuterių tinkluose.

4.1 Problematika

Išnagrinėjus šioje mokslinėje ataskaitoje pateiktos literatūros sąrašą pastebima, kad susiduriama su didelių duomenų saugojimo, apdorojimo problemomis. *Hashdoop* geba dirbti su didžiulių failų bazėmis, tačiau tokiu atveju įsilaužimų aptikimas nėra realaus laiko. Taip pastebima problema, kad šiuo metu žinomi mašininio mokymosi metodai nėra pakankamai pritaikyti naujų įsilaužimų kompiuterių tinkluose aptikimui. T. y., apmokymas vyksta su sukauptais duomenimis, duomenų rinkiniais. Našumo problemos taip pat yra aktualios siekiant nesumažinti kompiuterių tinklo siunčiamų duomenų greitaveikos.

4.2 Tolesni darbai

Tolesniuose darbuose numatoma atlikti gilesnę literatūros analizę apimant, bet neapsiribojant, naujesnius mašininio mokymosi metodus. Taip pat pateikti literatūros analizėje eksperimentinių tyrimų išvadas ir jų taikymą geresniam mašininio mokymosi metodo vystymui. Vienas iš darbų yra techninių specifikacijų nagrinėjimo siekiant sukurti prototipą fizinėje kompiuterių tinklo laboratorijoje. Laboratorija bus naudojama siekiant taikyti ir plėtoti mašininio mokymosi metodą įsilaužimams kompiuterių tinkluose aptikti su realiai laiko duomenimis (įprasti paketai ir anomalijos).

Darbuose numatoma gilintis į kompiuterių tinklų paketų struktūrą, išsiaiškinti kaip paketų antraštės informacija, požymiai, turi būti pritaikomi mašininio mokymosi metodams. Tam bus atliekama esamų sukauptų įvairiuose moksliniuose laboratorijoje

duomenų rinkinių failai, aiškinimąsi kokius mašininio mokymosi metodus yra tikslingiausia pritaikyti.

5 Literatūra

[1] P. Mishra, V. Varadharajan, U. Tupakula, ir E. S. Pilli, „A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection“, *IEEE Commun. Surv. Tutor.*, t. 21, nr. 1, p. 686–728, Firstquarter 2019, doi: 10.1109/COMST.2018.2847722.

[2] T. Xie, P. Ren, T. Zhang, ir Y. Y. Tang, „Distribution preserving learning for unsupervised feature selection“, *Neurocomputing*, t. 289, p. 231–240, geg. 2018, doi: 10.1016/j.neucom.2018.02.032.

[3] M. A. Ambusaidi, X. He, P. Nanda, ir Z. Tan, „Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm“, *IEEE Trans. Comput.*, t. 65, nr. 10, p. 2986–2998, spal. 2016, doi: 10.1109/TC.2016.2519914.

[4] H. Jinlong, Q. Zhu, L. Yang, D. Cheng, ir Q. Wu, „QCC: a novel clustering algorithm based on Quasi-Cluster Centers“, *Mach. Learn.*, t. 106, kovo 2017, doi: 10.1007/s10994-016-5608-2.

[5] H. Wang, Y. Zhang, J. Zhang, T. Li, ir L. Peng, „A factor graph model for unsupervised feature selection“, *Inf. Sci.*, t. 480, p. 144–159, bal. 2019, doi: 10.1016/j.ins.2018.12.034.

[6] M. Prasad, S. Tripathi, ir K. Dahal, „Unsupervised feature selection and cluster center initialization based arbitrary shaped clusters for intrusion detection“, *Comput. Secur.*, t. 99, p. 102062, gruodž. 2020, doi: 10.1016/j.cose.2020.102062.

[7] B. M. Lake, R. Salakhutdinov, ir J. B. Tenenbaum, „Human-level concept learning through probabilistic program induction“, *Science*, t. 350, nr. 6266, p. 1332–1338, gruodž. 2015, doi: 10.1126/science.aab3050.

[8] J. Snell, K. Swersky, ir R. S. Zemel, „Prototypical Networks for Few-shot Learning“, *ArXiv170305175 Cs Stat*, birž. 2017, Žiūrėta: spal. 18, 2020. [Interaktyvus]. Adresas: http://arxiv.org/abs/1703.05175.

[9] Y. Wang, Q. Yao, J. Kwok, ir L. M. Ni, „Generalizing from a Few Examples: A Survey on Few-Shot Learning“, *ArXiv190405046 Cs*, kovo 2020, Žiūrėta: spal. 18, 2020. [Interaktyvus]. Adresas: http://arxiv.org/abs/1904.05046.

[10] Li Fei-Fei, R. Fergus, ir P. Perona, „One-shot learning of object categories“, *IEEE Trans. Pattern Anal. Mach. Intell.*, t. 28, nr. 4, p. 594–611, bal. 2006, doi: 10.1109/TPAMI.2006.79.

[11] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, ir D. Wierstra, „Matching networks for one shot learning“, *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, gruodž. 2016, p. 3637–3645, Žiūrėta: spal. 02, 2020. [Interaktyvus].

[12] I. Sucholutsky ir M. Schonlau, „’Less Than One’-Shot Learning: Learning N Classes From M<N Samples“, *ArXiv200908449 Cs Stat*, rugs. 2020, Žiūrėta: spal. 02, 2020. [Interaktyvus]. Adresas: http://arxiv.org/abs/2009.08449.

[13] D. Jin, Y. Lu, J. Qin, Z. Cheng, ir Z. Mao, „SwiftIDS: Real-time intrusion detection system based on LightGBM and parallel intrusion detection mechanism“, *Comput. Secur.*, t. 97, p. 101984, spal. 2020, doi: 10.1016/j.cose.2020.101984.

[14] S. M. Kasongo ir Y. Sun, „A Deep Learning Method With Filter Based Feature Engineering for Wireless Intrusion Detection System“, *IEEE Access*, t. 7, p. 38597–38607, 2019, doi: 10.1109/ACCESS.2019.2905633.
[15] H. Vegda ir N. Modi, „Secure and Efficient Approach to Prevent Ad Hoc Network Attacks Using Intrusion Detection System“, *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, birž. 2018, p. 129–133, doi: 10.1109/ICCONS.2018.8662890.
[16] S. Jayaprakash ir K. Kandasamy, „Database Intrusion Detection System Using Octraplet and Machine Learning“, *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, bal. 2018, p. 1413–1416, doi: 10.1109/ICICCT.2018.8473029.
[17] P. Satam, S. Satam, ir S. Hariri, „Bluetooth Intrusion Detection System (BIDS)“, *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, spal. 2018, p. 1–7, doi: 10.1109/AICCSA.2018.8612809.
[18] M. M. Alani, *Guide to OSI and TCP/IP Models*. Cham: Springer International Publishing, 2014.
[19] H. Hindy *ir kt.*, „A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems“, *IEEE Access*, t. 8, p. 104650–104675, 2020, doi: 10.1109/ACCESS.2020.3000179.
[20] S. Paliwal, „Denial-of-Service, Probing & Remote to User (R2L) Attack Detection using Genetic Algorithm“, *Int. J. Comput. Appl.*, t. 60, p. 6.
[21] M. Sazzadul Hoque, „An Implementation of Intrusion Detection System Using Genetic Algorithm“, *Int. J. Netw. Secur. Its Appl.*, t. 4, nr. 2, p. 109–120, kovo 2012, doi: 10.5121/ijnsa.2012.4208.
[22] G. Munz ir G. Carle, „Real-time Analysis of Flow Data for Network Attack Detection“, *2007 10th IFIP/IEEE International Symposium on Integrated Network Management*, geg. 2007, p. 100–108, doi: 10.1109/INM.2007.374774.
[23] P. Barford ir D. Plonka, „Characteristics of network traffic flow anomalies“, *Proceedings of the 1st ACM SIGCOMM Workshop on Internet measurement*, San Francisco, California, USA, lapkr. 2001, p. 69–73, doi: 10.1145/505202.505211.
[24] F. Dressler ir G. Munz, „Flexible Flow Aggregation for Adaptive Network Monitoring“, *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, lapkr. 2006, p. 702–709, doi: 10.1109/LCN.2006.322180.
[25] C. Gates ir D. Becknel, „Host anomalies from network data“, *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*, birž. 2005, p. 325–332, doi: 10.1109/IAW.2005.1495970.
[26] L. Oz, E. Eilertson, A. Lazarevic, P. Tan, V. Kumar, ir J. Srivastava, „MINDS - Minnesota Intrusion Detection System“, rugpj. 2003.
[27] „NfSen“. http://nfsen.sourceforge.net/ (žiūrėta birž. 22, 2020).
[28] „ntop“, *ntop*. https://www.ntop.org/ (žiūrėta birž. 22, 2020).
[29] „Homepage“, *Plixer*. https://www.plixer.com/ (žiūrėta birž. 22, 2020).
[30] International Organization for Standardization, „ISO/IEC 17799:2000“, *ISO*. http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/03/34/33441.html (žiūrėta lapkr. 19, 2019).
[31] V. Chandola, A. Banerjee, ir V. Kumar, „Anomaly detection: A survey“, *ACM Comput. Surv.*, t. 41, nr. 3, p. 1–58, liep. 2009, doi: 10.1145/1541880.1541882.
[32] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, ir E. Vázquez, „Anomaly-based network intrusion detection: Techniques, systems and challenges“, *Comput. Secur.*, t. 28, nr. 1–2, p. 18–28, vas. 2009, doi: 10.1016/j.cose.2008.08.003.

[33] H. Debar, M. Dacier, ir A. Wespi, „A revised taxonomy for intrusion-detection systems“, *Ann. Télécommunications*, t. 55, nr. 7, p. 361–378, liep. 2000, doi: 10.1007/BF02994844.
[34] S. Axelsson, „Intrusion Detection Systems: A Survey and Taxonomy“, bal. 2000.
[35] P. Tripathi, „A Brief Review on Intrusion Detection System with its Classification Types“, *Int. J. Res. Appl. Sci. Eng. Technol.*, t. 7, nr. 8, p. 963–969, rugpj. 2019, doi: 10.22214/ijraset.2019.8143.
[36] C. Xenakis, C. Panos, ir I. Stavrakakis, „A Comparative Evaluation of Intrusion Detection Architectures for Mobile Ad Hoc“, *Comput. Secur.*, t. 30, p. 63–80, bal. 2011, doi: 10.1016/j.cose.2010.10.008.
[37] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, ir K.-Y. Tung, „Intrusion detection system: A comprehensive review“, *J. Netw. Comput. Appl.*, t. 36, nr. 1, p. 16–24, saus. 2013, doi: 10.1016/j.jnca.2012.09.004.
[38] R. Bace ir P. Mell, „Intrusion Detection Systems (IDS)“, National Institute of Standards and Technology, NIST Special Publication (SP) 800-31 (Withdrawn), lapkr. 2001. doi: https://doi.org/10.6028/NIST.SP.800-31.
[39] F. Sabahi ir A. Movaghar, „Intrusion Detection: A Survey“, *Syst. Netw. Commun. Int. Conf. On*, t. 0, p. 23–26, saus. 2008, doi: 10.1109/ICSNC.2008.44.
[40] N. K. Mittal, „A survey on Wireless Sensor Network for Community Intrusion Detection Systems“, *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, kovo 2016, p. 107–111, doi: 10.1109/RAIT.2016.7507884.
[41] J. Shun ir H. A. Malki, „Network Intrusion Detection System Using Neural Networks“, *2008 Fourth International Conference on Natural Computation*, Jinan, Shandong, China, 2008, p. 242–246, doi: 10.1109/ICNC.2008.900.
[42] A. G. Tokhtabayev ir V. A. Skormin, „Non-Stationary Markov Models and Anomaly Propagation Analysis in IDS“, *Third International Symposium on Information Assurance and Security*, rugpj. 2007, p. 203–208, doi: 10.1109/IAS.2007.72.
[43] A. Karami, „An anomaly-based intrusion detection system in presence of benign outliers with visualization capabilities“, *Expert Syst. Appl.*, t. 108, p. 36–60, spal. 2018, doi: 10.1016/j.eswa.2018.04.038.
[44] S. K. Gautam ir H. Om, „Computational neural network regression model for Host based Intrusion Detection System“, *Perspect. Sci.*, t. 8, p. 93–95, rugs. 2016, doi: 10.1016/j.pisc.2016.04.005.
[45] E. Hodo, X. Bellekens, A. Hamilton, ir C. Tachtatzis, „Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey“, p. 43, 2017.
[46] A. Qayyum, M. H. Islam, ir M. Jamil, *Taxonomy of statistical based anomaly detection techniques for intrusion detection*. 2005, p. 276.
[47] H. Debar, M. Becker, ir D. Siboni, *A Neural Network Component for an Intrusion Detection System*. 1992, p. 250.
[48] K. Wang ir S. J. Stolfo, „Anomalous Payload-Based Network Intrusion Detection“, *Recent Advances in Intrusion Detection*, t. 3224, E. Jonsson, A. Valdes, ir M. Almgren, Sud. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, p. 203–222.
[49] T. F. Lunt, „A survey of intrusion detection techniques“, *Comput. Secur.*, t. 12, nr. 4, p. 405–418, birž. 1993, doi: 10.1016/0167-4048(93)90029-5.
[50] R. C. A. Naidu ir P. S. Avadhani, „A comparison of data mining techniques for intrusion detection“, *2012 IEEE International Conference on Advanced*

Communication Control and Computing Technologies (ICACCCT), rugpj. 2012, p. 41–44, doi: 10.1109/ICACCCT.2012.6320731.
[51] D. Anderson, T. Frivold, ir A. Valdes, „Next-generation Intrusion Detection Expert System (NIDES) A Summary“, saus. 1995.
[52] D. E. Denning, „An Intrusion-Detection Model“, p. 17.
[53] J. Veeramreddy, V. Prasad, ir K. Prasad, „A Review of Anomaly based Intrusion Detection Systems“, *Int. J. Comput. Appl.*, t. 28, p. 26–35, rugpj. 2011, doi: 10.5120/3399-4730.
[54] B. A. Kuperman, „A categorization of computer security monitoring systems and the impact on the design of audit sources“, phd, Purdue University, USA, 2004.
[55] W. Sha, Y. Zhu, T. Huang, M. Qiu, Y. Zhu, ir Q. Zhang, *A Multi-Order Markov Chain Based Scheme for Anomaly Detection*. 2013, p. 88.
[56] N. Ye, „A Markov Chain Model of Temporal Behavior for Anomaly Detection“, p. 4, 2000.
[57] H. Zhengbing, L. Zhitang, ir W. Junqi, „A novel Network Intrusion Detection System (NIDS) based on signatures search of data mining“, *Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop*, Brussels, BEL, saus. 2008, p. 1–7, Žiūrėta: spal. 02, 2020. [Interaktyvus].
[58] H. E. Poston, „A brief taxonomy of intrusion detection strategies“, *2012 IEEE National Aerospace and Electronics Conference (NAECON)*, liep. 2012, p. 255–263, doi: 10.1109/NAECON.2012.6531064.
[59] H. Debar, M. Dacier, ir A. Wespi, „Towards a taxonomy of intrusion-detection systems“, *Comput. Netw.*, t. 31, nr. 8, p. 805–822, bal. 1999, doi: 10.1016/S1389-1286(98)00017-6.
[60] T. H. Ptacek ir T. N. Newsham, „Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection“, p. 66.
[61] C. Kolias, G. Kambourakis, A. Stavrou, ir S. Gritzalis, „Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset“, *IEEE Commun. Surv. Tutor.*, t. 18, nr. 1, p. 184–208, Firstquarter 2016, doi: 10.1109/COMST.2015.2402161.
[62] Q. Xiong, Y. Xu, B. Zhang, ir F. Wang, „Overview of the Evasion Resilience Testing Technology for Network Based Intrusion Protecting Devices“, *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, saus. 2017, p. 146–152, doi: 10.1109/HASE.2017.37.
[63] D. Kim *ir kt.*, *DynODet: Detecting Dynamic Obfuscation in Malware*. 2017, p. 118.
[64] M. Cova, C. Kruegel, ir G. Vigna, „Detection and analysis of drive-by-download attacks and malicious JavaScript code“, *Proceedings of the 19th international conference on World wide web - WWW '10*, Raleigh, North Carolina, USA, 2010, p. 281, doi: 10.1145/1772690.1772720.
[65] A. R. Metke ir R. L. Ekl, „Security Technology for Smart Grid Networks“, *IEEE Trans. Smart Grid*, t. 1, nr. 1, p. 99–107, birž. 2010, doi: 10.1109/TSG.2010.2046347.
[66] I. Butun, S. D. Morgera, ir R. Sankar, „A Survey of Intrusion Detection Systems in Wireless Sensor Networks“, *IEEE Commun. Surv. Tutor.*, t. 16, nr. 1, p. 266–282, First 2014, doi: 10.1109/SURV.2013.050113.00191.
[67] J. Camacho, A. Pérez-Villegas, P. García-Teodoro, ir G. Maciá-Fernández, „PCA-based multivariate statistical network monitoring for anomaly detection“, *Comput. Secur.*, t. 59, p. 118–137, birž. 2016, doi: 10.1016/j.cose.2016.02.008.

[68] C. Andrieu ir C. Andrieu, „An Introduction to MCMC for Machine Learning“, p. 39.
[69] L. P. Kaelbling, M. L. Littman, ir A. W. Moore, „Reinforcement Learning: A Survey“, *J. Artif. Intell. Res.*, t. 4, p. 237–285, geg. 1996, doi: 10.1613/jair.301.
[70] M. T. Hagan, H. B. Demuth, M. H. Beale, ir O. De Jesus, *Neural network design*, 2nd edition. Wrocław: Amazon Fulfillment Poland Sp. z o.o.
[71] G. Huang, H. Zhou, X. Ding, ir R. Zhang, „Extreme Learning Machine for Regression and Multiclass Classification“, *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, t. 42, nr. 2, p. 513–529, bal. 2012, doi: 10.1109/TSMCB.2011.2168604.
[72] Z. Banković, D. Stepanović, S. Bojanić, ir O. Nieto-Taladriz, „Improving network security using genetic algorithm approach“, *Comput. Electr. Eng.*, t. 33, nr. 5, p. 438–451, rugs. 2007, doi: 10.1016/j.compeleceng.2007.05.010.
[73] N. Wattanapongsakorn *ir kt.*, „A Practical Network-Based Intrusion Detection and Prevention System“, *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, birž. 2012, p. 209–214, doi: 10.1109/TrustCom.2012.46.
[74] S.-H. Kang ir K. Kim, „A feature selection approach to find optimal feature subsets for the network intrusion detection system“, *Clust. Comput.*, t. 19, kovo 2016, doi: 10.1007/s10586-015-0527-8.
[75] R. Fontugne, J. Mazel, ir K. Fukuda, „Hashdoop: A MapReduce framework for network anomaly detection“, *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, bal. 2014, p. 494–499, doi: 10.1109/INFCOMW.2014.6849281.
[76] J. Dean ir S. Ghemawat, „MapReduce: simplified data processing on large clusters“, *Commun. ACM*, t. 51, nr. 1, p. 107–113, saus. 2008, doi: 10.1145/1327452.1327492.
[77] Y. Lee ir Y. Lee, „Toward scalable internet traffic measurement and analysis with Hadoop“, *ACM SIGCOMM Comput. Commun. Rev.*, t. 43, nr. 1, p. 5–13, saus. 2012, doi: 10.1145/2427036.2427038.
[78] M. M. Rathore, A. Ahmad, ir A. Paul, „Real time intrusion detection system for ultra-high-speed big data environments“, *J. Supercomput.*, t. 72, nr. 9, p. 3489–3510, rugs. 2016, doi: 10.1007/s11227-015-1615-5.
[79] Y. Du, J. Liu, F. Liu, ir L. Chen, „A Real-Time Anomalies Detection System Based on Streaming Technology“, *2014 Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics*, rugpj. 2014, t. 2, p. 275–279, doi: 10.1109/IHMSC.2014.168.
[80] C. Wang, I. A. Rayan, ir K. Schwan, „Faster, larger, easier: reining real-time big data processing in cloud“, *Proceedings of the Posters and Demo Track*, Montreal, Quebec, Canada, gruodž. 2012, p. 1–2, doi: 10.1145/2405153.2405157.
[81] R. M. Karp, S. Shenker, ir C. H. Papadimitriou, „A simple algorithm for finding frequent elements in streams and bags“, *ACM Trans. Database Syst.*, t. 28, nr. 1, p. 51–55, kovo 2003, doi: 10.1145/762471.762473.
[82] Q. Zhang, Y. Jin, Y. Cui, ir M. Song, „A Distributed Network Measurement System Based on Hadoop“, *2012 8th International Conference on Wireless Communications, Networking and Mobile Computing*, rugs. 2012, p. 1–4, doi: 10.1109/WiCOM.2012.6478284.
[83] „Apache Mahout“. http://mahout.apache.org/ (žiūrėta birž. 22, 2020).
[84] S. Jiang, X. Song, H. Wang, J.-J. Han, ir Q.-H. Li, „A clustering-based method for unsupervised intrusion detections“, *Pattern Recognit. Lett.*, t. 27, nr. 7, p. 802–810, geg. 2006, doi: 10.1016/j.patrec.2005.11.007.

[85] P. Laskov, P. Düssel, C. Schäfer, ir K. Rieck, „Learning Intrusion Detection: Supervised or Unsupervised?“, *Image Analysis and Processing – ICIAP 2005*, t. 3617, F. Roli ir S. Vitulano, Sud. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, p. 50–57.
[86] Y. Liao ir R. Vemuri, „Use of K-Nearest Neighbor classifier for intrusion detection“, *Comput. Secur.*, t. 21, p. 439–448, spal. 2002, doi: 10.1016/S0167-4048(02)00514-X.
[87] „Evolutionary neural networks for anomaly detection based on the behavior of a program“, *ResearchGate*. https://www.researchgate.net/publication/7021102_Evolutionary_neural_networks_for_anomaly_detection_based_on_the_behavior_of_a_program (žiūrėta birž. 22, 2020).
[88] R. Zhang, S. Zhang, ir S. Muthuraman, „One Class Support Vector Machine for Anomaly Detection in the Communication Network Performance Data“, p. 7.
[89] G. Munz, S. Li, ir G. Carle, „Traffic Anomaly Detection Using K-Means Clustering“, p. 8.