

**VILNIAUS UNIVERSITETAS
FIZIKOS FAKULTETAS
BENDROSIOS FIZIKOS IR SPEKTROSKOPIJOS KATEDRA**

Karolis Tarasaitis

**DDOS ATAKŲ APTIKIMO, NAUDOJANT STATISTINIUS METODUS,
EFEKTYVUMO TYRIMAS**

Pagrindinių studijų baigiamasis darbas

(studijų programa – KOMPIUTERINĖ FIZIKA)

Studentas

Karolis Tarasaitis

Darbo vadovas

dr. Feliksas Kuliešius

Recenzentas

dr. Alytis Gruodis

Katedros vedėjas

prof. Valdas Šablinskas

Vilnius 2016

Turinys

1	DDOS atakų aptikimo problematikos apžvalga	3
1.1	DDoS tipai	3
1.2	Sauga kompiuterių tinkluose	4
1.3	Apsauga nuo DDoS atakų.....	5
1.4	Atakų atpažinimo tipai.....	6
1.5	Informacijos entropija ir jos taikymai	7
1.5.1	Informacijos entropijos sumažėjimas	8
1.5.2	Informacijos entropijos padidėjimas	9
1.5.3	Informacijos entropijos lyginimas.....	10
1.5.4	Lango pločio parametras	10
1.6	Atstumo tarp tinklo tėkmių apskaičiavimas	10
1.7	Laikinės priklausomybės prognozavimas	12
1.8	Fazinės erdvės apskaičiavimas	13
1.9	Bendros informacijos apskaičiavimas	15
1.9.1	Cellucci algoritmas	18
1.9.2	Išvestiniai bendros informacijos apskaičiavimo algoritmai.....	20
1.9.3	Algoritmo įgyvendinimas.....	21
1.10	Prognozavimo uždavinys	22
2	DDOS atakų aptikimo tyrimo metodai	25
2.1	Duomenų parengimas.....	25
2.2	Trumpas programos aprašymas	27
2.3	Informacijos entropijos apskaičiavimas	28
2.4	Laikinės priklausomybės prognozavimas	33
3	Rezultatai ir jų aptarimas	41
	Pagrindiniai rezultatai ir išvados	43
	Literatūra	44
	Santrauka	45

Įvadas

Vis dažniau žiniuose išgirstame, jog internetinė svetainė tapo DDoS (angl. Distributed Denial of Service) atakų auka. Dėl DDoS atakų internetinių paslaugų kūrėjai patiria nuostolius, praranda ir vartotojus bei pelną. Sukurti DDoS ataką yra gana paprasta, o apsisaugojimas nuo jų yra vis dar neišspręsta problema. Pirmas žingsnis norint apsisaugoti nuo DDoS atakų yra šių atakų aptikimas. Šiame darbe yra tiriamas DDoS atakų aptikimo efektyvumas, naudojant du skirtingus statistinius metodus.

Pirmas metodas yra paremtas informacijos entropijos apskaičiavimu normaliaame tinklo darbo režime. Darbo režimo metu, apskaičiuojame ribinę - minimalią informacijos entropijos vertę. Tada galime stebėti ar informacijos entropijos vertės nusileidžia žemiau šios ribos. Užfiksavus tokį atvejį, galime pranešti apie tinklo anomaliją ir imtis tolimesnių veiksmų, kuriais būtų neleidžiama užkrėstų vartotojų siunčiamiems paketams pasiekti atakuojamą tinklo dalį.

Antras metodas paremtas laikinės priklausomybės verčių prognozavimu. Naudojant šį metodą taip pat yra tiriamas tinklas jo darbo režimu (kai žinome, jog nėra DDoS atakų). Jeigu sugebame prognozuoti tam tikro tinklo atributo reikšmės tinklo darbo režimo metu, galime pranešti apie tinklo anomalijas, kai realios vertės nukrypsta nuo prognozuojamų verčių. Tam, jog būtume tikri, kad tinklo anomalijos yra susijusios su DDoS atakomis, prognozuosime apskaičiuojamas informacijos entropijos vertes.

Darbo tikslas - ištirti statistinius metodus, paremtus informacijos entropijos sumažėjimu bei laikinės priklausomybės prognozavimu, naudojamus DDoS atakoms aptikti.

Darbo uždaviniai:

1. parodyti, kaip pasikeičia informacijos entropijos reikšmės, kai tinklą pradeda atakuoti DDoS atakos;
2. parodyti, kad galime prognozuoti periodinių laikinių priklausomybių vertes;
3. patikrinti ar skirtingų laiko intervalų realios informacijos entropijos turi deterministinį sąryšį, t. y. ar galime aptikti DDoS atakas, naudojant laikinės priklausomybės prognozavimu paremtą metodą.

1 DDOS atakų aptikimo problematikos apžvalga

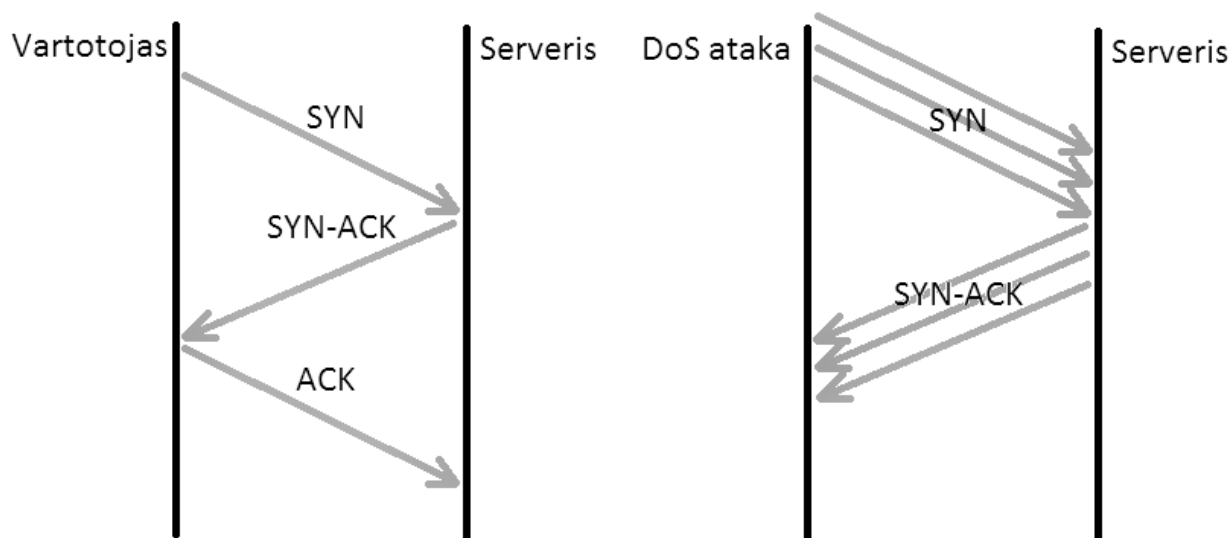
Atsisakymo aptarnauti (angl. Denial of Service - DoS) bei paskirstyto atsisakymo aptarnauti (angl. Distributed Denial of Service) atakos yra skirtos sutrukdyti legaliems vartotojams naudotis internetine paslauga. Atakų kūrėjai tikisi išnaudoti ribotą tinklo pralaidumą, serverio resursų kiekį arba serverio galimybę užmegzti tik ribotą kiekį sesijų tarp serverio ir kliento. Viso to tikslas yra neleisti legaliems klientams naudotis teikiamomis paslaugomis.

1.1 DDoS tipai

DoS atakos yra skirstomos į du pagrindinius tipus: atakos, priklausančios pirmajam tipui, išnaudoja įvairius programinės įrangos trūkumus, antro tipo atakos taikosi į tinklo pralaidos užpildymą dideliu kiekiu paketų. Vienas iš pirmojo tipo atakų pavyzdžių yra (S)SYN užtvindymo ataka (angl. (S)SYN flood), kurios metu yra išnaudojami TCP protokolo trigubo pasisveikinimo (angl. three-way-handshake) savybės. DoS atakos metu serveriui yra siunčiamas didelis kiekis SYN paketų, serveris atsako išsiųsdamas SYN-ACK paketus ir laukia grįžtančių ACK paketų. Tačiau paketai negrįžta, o dėl pusiau užmegztų seansų, skirtų TCP ryšiui užmegzti, kanalų skaičius yra išnaudojamas. Taigi legalūs vartotojai nebegali prisijungti prie serverio.

Apsiginti nuo tokio tipo atakų galima sutvarkant serverio nuostatas (pvz.: nustatant laiko limitą, per kurį serveris turi sulaukti atsakymo; atsakymui negrįžus kanalas atlaisvinamas) ar atnaujinant techninę įrangą.

Antro tipo atakos taikosi į tinklo pralaidos užpildymą dideliu kiekiu paketų, atrodančių kaip tikrų vartotojų siunčiami duomenys ar užklauskos. Užklauskų kiekiai, reikalingi, norint išnaudoti šiuolaikinių serverių ar kanalo resursus, yra labai dideli. Taigi sunku sukurti efektyvią DoS ataką, kai yra naudojamas tik vienas kompiuteris. Todėl buvo pradėtos naudoti DDoS atakos. Norint vykdyti DDoS ataką, reikia didelio kiekio kompiuterių, kurių kiekvienas vykdo individualią DoS ataką. Dažniausiai yra naudojami specialia, kenksminga programine įranga užkrėsti kompiuteriai. Šios programinės įrangos pagrindinis tikslas yra leisti išoriniam kompiuteriui pradėti bei sustabdyti DoS ataką. Valdant didelį kiekį taip užkrėstų kompiuterių (angl. botnet) galima atlikti galingas DDoS atakas. Toliau yra kalbama būtent apie tokias DDoS



1 pav. Kairėje pavaizduota legalaus vartotojo TCP ryšio užmezgimo schema. Siunčiamas SYN paketas, serveris atsako su SYN-ACK paketu ir laukia, kol iš vartotojo grįžta ACK paketas. Dešinėje pavaizduota, kas vyksta DoS atakos metu. Serveris nesulaukia grįžtančių ACK paketų.

atakas, kurias vykdant yra norima užimti visą aukos tinklo pralaidą arba sunaudoti visus aukos serverio turimus resursus.

1.2 Sauga kompiuterių tinkluose

Sauga kompiuterių tinkluose yra skirstoma į tris kategorijas: konfidencialumo, prieinamumo bei duomenų vientisumo. Akivaizdu, kad DDoS atakos taikosi į prieinamumo kategoriją. Pirmas žingsnis norint neutralizuoti DDoS atakas yra jų aptikimas. Tai darančios sistemos turėtų sugebėti aptikti atakas realiu laiku bei turėti galimybę atskirti atakų paketus nuo legalių vartotojų paketų.

Nepaisant industrijos atstovų bei mokslininkų pastangų, DDoS atakos yra vis dar neišspręsta problema. Atakų kaltininkus susekti yra sunku, o netgi ir žinant, kas praktiškai vykdo ataką, nubausti kaltininkus ne visada pavyksta, nes teoriškai ataką vykdyti gali ir visai kitas žmogus (dėl galimo IP adresų pakeitimo (angl. IP spoofing)). Yra keletas pagrindinių priežasčių, kodėl susiklostė tokia situacija: saugos dizainas, tinkluose egzistuojanti anarchija, skirtingi tarpvalstybiniai susitarimai.

Saugos dizainas kilo iš ARPANET (angl. Advanced Research Process Agency Network),

tai buvo pirmasis kompiuterinis tinklas naudojęs duomenų paketų maršrutizaciją. Šio tinklo vystymosi pradžioje jis buvo necentralizuotas, tai yra jungė tik du universitetus MIT (angl. Massachusetts Institute of Technology) bei UCLA (angl. University of California, Los Angeles), taigi saugumu rūpintis beveik nereikėjo. Vėliau, tinklui tapus viešam, sutvarkyti architektūrinius trūkumus tapo sunku. Nors buvo sukurta daug saugumo atnaujinimų, problemos, tokios kaip perduodamų paketų būsenos nebuvimas (kuri parodytų per kuriuos maršrutizatorius buvo siunčiamas paketas) arba galimas šaltinio IP adresų pakeitimas (angl. IP spoofing), liko neišspręstos. Be to Internetas yra didžiulė sistema, kuri iš dalies vis dar yra anarchijos būsenoje. Yra daugybė atskirų interneto tiekėjų. Skirtingi internetiniai tinklai naudoja skirtingą įrangą, kuri yra įvairaus senumo bei naudoja skirtingus standartus. Taigi visur sudiegti bendrą, saugumo reikalavimus atitinkančią, programinę įrangą yra praktiškai neįmanoma.

1.3 Apsauga nuo DDoS atakų

Apsisaugojimas nuo DDoS atakų vyksta trimis etapais: DDoS atakų aptikimas, atakų neutralizavimas bei jos vykdytojų susekimas. Neutralizavimas yra paketų, siunčiamų iš kompiuterio, vykdančio DDoS ataką, atmetimas (kad paketas nepasiektų atakuojamo tinklo), o susekimas—informacijos apie to kompiuterio IP adresą gavimas. Suprantama, kad pats svarbiausias žingsnis yra būtent atakų aptikimas, nes tik aptikus atakas, galimas nuo jų apsiginti. Visi metodai, skirti aptikti atakoms yra susiję su tinklo stebėjimu, tam tikrų loginių sekų ieškojimu, įvairių anomalijų apskaičiavimu. Kitaip sakant, ieškoma pėdsakų, kurie pasakytų, jog yra vykdoma DDoS ataka. Dauguma sistemų, aptinkančių DDoS atakas, yra sukurtos specifinėms atakoms (naudoja iš anksto paruoštus šablonus, su kuriais lygina registruojamų paketų srautą), dėl to nesugeba aptikti naujų atakų. Kitos sistemos, kurios veikia įvairių tinklo statistinių parametrų skaičiavimo principu, susiduria su sunkumais, kai reikia atskirti DDoS atakas nuo padidėjusio legalių vartotojų siunčiamų paketų srauto (angl. flash crowd). Legalių vartotojų srautas gali padidėti dėl daugelio priežasčių. Naujienų svetainė staiga gali susilaukti didelio vartotojų kiekio dėl neseniai įvykusios katastrofos, o internetinės bilietų į renginius parduotuvės dėl pradėto bilietų platinimo. Jeigu nesugebėsime atskirti DDoS atakų nuo legalių vartoto-

jų siunčiamų paketų srauto padidėjimo, DDoS atakų atpažinimo sistema sukels nereikalingus nuostolius, užkirsdama prieigą legaliems vartotojams.

Taigi, sistemos, aptinkančios DDoS atakas, privalo: **a)** neblokuoti padidėjusio legalių vartotojų paketų srauto (klaidingai teigiami (angl. false positive FP) atakų aptikimo atvejai), bet kartu ir **b)** aptikti visas DDoS atakas (klaidingai neigiami (angl. false negative FN) arba neaptiktų atakų atvejai). Atsižvelgiant į FP bei FN atakų skaičius bei lyginant juos su aptiktų bei tikrų atakų skaičiais galima kiekybiškai įvertinti atakų aptikimo sistemos efektyvumą.

DDoS atakų atpažinimo sistemos turi išspręsti kelis pagrindinius uždavinius. Tinklo įrenginiai turėtų a) sugebėti automatiškai (be žmogaus įsikišimo) aptikti, jog tinklas yra atakuojamas b) sureaguoti į vykdomą ataką ir neleisti DDoS atakai priklausantiems paketams pasiekti aukos, jog nebūtų bereikalingai naudojami paslaugą teikiančios aplikacijos serverio resursai ar tinklo pralaida. Atakų aptikimo algoritmai turėtų veikti plačiame internetinių sistemų spektre. Neturėtų reikėti žmogaus įsikišimo norint pritaikyti sistemas skirtinguose tinkluose.

1.4 Atakų atpažinimo tipai

Būdai, kaip gali būti atpažįstamos DDoS atakos yra labai įvairūs. Vieni jų yra paremti įvairiais siunčiamų paketų atributų pokyčiais. Šie metodai ieško pakitusių šaltinio IP adresų (angl. IP spoofing). Aptikimo principas pagrįstas tuo, kad šuolių skaičius (angl. hop count) tarp dviejų tinkle bendraujančių kompiuterių išlieka pastovus arba keičiasi nedaug [1]. Taigi saugojant IP adresą kartu su jį atitinkančiu šuolių skaičiumi ir stebint ar šuolių skaičius išlieka pastovus—galima aptikti anomalijas. Kiti būdai yra susiję su periodiniais siunčiamų paketų srauto padidėjimais. Norint juos įgyvendinti, reikia saugoti per tam tikrą laiką atėjusių paketų skaičius ir laiko žymas. Tada periodinius intervalus galima rasti panaudojus diskrečią Furjė transformaciją (angl. discrete Fourier transform—DFT) [2], kuri išryškina dažnio pasikartojimą. Trečia DDoS atakų aptikimo strategija yra ieškoti staigių tinklo apkrovos pasikeitimų. Normaliame darbo režime tinkle siuntinėjamų paketų šaltinių adresai, atsižvelgiant į išsiunčiamų paketų skaičius, turėtų būti pasiskirstę atsitiktinai. Jeigu yra pastebima, kad didžioji dalis gaunamų paketų yra išsiunčiami iš mažos visų šaltinių IP adresų dalies, galime manyti,

jog tinklas yra puolamas. Tokius apkrovos pasikeitimus galime tirti informacijos entropijos skaičiavimais pagrįstais metodais [3].

1.5 Informacijos entropija ir jos taikymai

Diskretaus atsitiktinio kintamojo X Šenono (angl. Shannon) entropija gali būti užrašoma tokia bendra išraiška:

$$H(X) = - \sum_i P(x_i) \log_2 P(x_i) \quad (1)$$

Priklausomai nuo logaritmo pagrindo b entropija yra vadinama skirtingais vardais. Dažniausiai vartojamos vertės yra 2, e , 10. Informacijos entropija, kurios logaritmo pagrindas yra 2, yra vadinama Šenono entropija. Šenono entropijos matavimo vienetai dažnai yra vadinami bitais. Informacijos entropijos vertės auga tiesiškai, kai didėja tiriamos sistemos dydis. To pavyzdys galėtų būti metama moneta, informacijos entropija metant monetą n kartų, turėtų būti lygi entropijai metant monetą vieną kartą, padauginant iš n . Monetą metant vieną kartą, kai tikimybės atvirsti skaičiui arba herbui yra lygios, Šenono entropija:

$$H = - \sum_{i=1}^2 \left(\frac{1}{2}\right)_i \log_2 \left(\frac{1}{2}\right)_i = 1 \quad (2)$$

Metant tokią pat monetą n kartų:

$$H = - \sum_{i=1}^{2^n} \left(\frac{1}{2^n}\right)_i \log_2 \left(\frac{1}{2^n}\right)_i = - \sum_{i=1}^{2^n} \left(\frac{1}{2^n}\right)_i n \log_2 \left(\frac{1}{2}\right)_i = n \left[- \sum_{i=1}^2 \left(\frac{1}{2}\right)_i \log_2 \left(\frac{1}{2}\right)_i \right] = n \quad (3)$$

Taip pat, Šenono entropija yra ypatinga, nes ji yra susijusi su informacija. Tai parodyti galima skaičiuojant $\log_2(1/p_i)$ vidurkį V :

$$V = \sum_i p_i \log_2 \frac{1}{p_i} \quad (4)$$

Dydį $\log_2(1/p_i)$ galima vadinti informacija, nes tuo atveju, kai visų įvykių tikimybės yra

lygios, būtų $(1/p_i)$ įvykių, tai reiškia, kad norint pasakyti, kuris įvykis įvyko, reikia naudoti $\log_2(1/p_i)$ bitų kiekį. Tačiau, pasirodo, jog vidurkis V turi prasmę ne tik tada, kai visų įvykių tikimybės yra lygios. Šenono glaudinimo teoremoje (angl. Shannon's source coding theorem) teigiama, kad neįmanoma suglaudinti (angl. compress) atsitiktinio n raidžių teksto taip, kad jo ilgis būtų mažesnis už nV .

Darbe, kai yra kalbama apie informacijos entropiją H , yra naudojama Šenono entropija logaritmo pagrindu 2:

$$H = - \sum_{i=1}^n (p_i \log_2 p_i) \quad (5)$$

, kur p_i yra tikimybė, o n - grupių skaičius. Tikimybė nusako, kuri dalis pasirinktos grupės paketų buvo registruota, tam tikro tipo intervale. Registruojami paketai gali būti skirstomi į grupes pagal šaltinio IP adresus, pagal paskirties IP adresus, taip pat pagal šaltinių bei paskirties adresų prievadus. Intervalą galima apriboti laiko srityje (pavyzdžiui lango plotis, kuriame apskaičiuojama entropija yra lygus vienai sekunde) arba paketų kiekio srityje (pvz.: lango plotis—1000 paketų). Apskaičiuojamos entropijos vertės pasiskirsto intervale nuo 0 iki $\log_2 n$, kur n yra grupių skaičius.

1.5.1 Informacijos entropijos sumažėjimas

Kai paslaugą teikianti aplikacija nėra atakuojama, informacijos entropija turėtų būti beveik pastovus, t. y. mažai svyruojantis dydis. Taip yra todėl, kad tinkle perduodamų paketų grupių skaičius $n \gg 1$, o visų grupių suminis paketų skaičius M yra daug didesnis už bet kurios iš grupių paketų skaičių N_i , t. y. $M \gg N_i$ visiems i , tai reiškia, kad vienos grupės įvykių tikimybių pokyčiai mažai įtakoja bendrą visų grupių entropiją H . Tačiau, prasidėjus DDoS atakai, kai kurie paketų srautai gali tapti dominuojantys ir $N_j \approx M$. Tai reiškia, jog entropija, skaičiuojama pasirinktame intervale (tai gali būti laiko intervalas, pvz.: sekundė, arba paketų skaičiaus intervalas, pvz.: 1000 paketų) ženkliai sumažės.

Tarkime, kad tinklui veikiant pastoviu režimu, yra $n = 100$ skirtingų grupių skaičius. Kiekvienoje grupėje, per pasirinktą laiko intervalą, registruojama $N = 100$ paketų, taigi bendras pa-

ketų skaičius $M = nN$, $M = 10000$. Tikimybė registruoti paketą tam tikroje grupėje $p_i = (N/M)$, $p_i = (1/100)$. Taigi Šenono entropija:

$$H = -100 \frac{1}{100} \log_2 \frac{1}{100} = -\log_2 \frac{1}{100} \cong 6,64 \quad (6)$$

Prasidėjus DDoS atakai, vienoje iš grupių kiekvieno laiko intervalo metu pradėta registruoti po 90100 paketų, o kitose grupėse registruojami paketų skaičiai nepakito. Tai reiškia bendras paketų skaičius $M = 10^5$. Tikimybė, kad paketas bus registruojamas pakitusioje grupėje $p_1 = 90100/10^5 \cong 9/10$, o tikimybė registruoti paketą bet kurioje vienoje iš nepakitusių grupių $p_i = 100/10^5 = 1/1000$. Tokios sistemos entropija:

$$H = -\frac{9}{10} \log_2 \frac{9}{10} - 99 \frac{1}{1000} \log_2 \frac{1}{1000} \cong 1,12 \quad (7)$$

Aiškiai matome, jog pateikto atakos pavyzdžio metu entropija sumažėja.

1.5.2 Informacijos entropijos padidėjimas

Tačiau informacijos entropija sumažėja ne visais atvejais. Pateiktame pavyzdyje atakos kaltininkė yra tik viena grupė. Tai nebūtinai visada yra tiesa. Gali būti, kad atakos metu pasikeičia daugelio grupių registruojamų paketų skaičiai arba net atsiranda naujų grupių. Dėl to entropija gali ir padidėti. Tarkime tinklo darbo režime yra $n = 100$ skirtingų grupių. Kiekvienoje iš grupių per nustatytą laiko intervalą vėl registruojama po $N = 100$ paketų. Sistemos entropija tokia pati, kaip ir pirmame pavyzdyje $H \cong 6,64$. Tačiau šiuo atveju prasidėjus DDoS atakai tinkle atsiranda $n_2 = 900$ papildomų grupių ir kiekviena iš jų taip pat siunčia po $N = 100$ paketų. Šiuo atveju bendras paketų skaičius $M = nN + n_2N$, $M = 10^5$, o tikimybė rasti paketą bet kurioje iš grupių $p_i = 100/10^5 = 1/1000$. Toje sistemoje apskaičiuota Šenono entropija bus lygi:

$$H = -1000 \frac{1}{1000} \log_2 \frac{1}{1000} = -\log_2 \frac{1}{1000} \cong 9,97 \quad (8)$$

Pateikto pavyzdžio metu matome, jog Šenono entropija išaugo ($9,97 > 6,64$). Matome, kad šiuo atveju entropija akivaizdžiai padidėja. Taigi priklausomai nuo DDoS atakų pobūdžio

entropija gali ir padidėti ir sumažėti.

1.5.3 Informacijos entropijos lyginimas

Kad būtų galima lyginti sistemų entropijas skirtinguose laiko intervaluose, kai skiriasi perduodamų paketų srautai, informacijos entropiją reikia normuoti. Informacijos entropija yra normuojama, dalinant apskaičiuotą vertę iš didžiausios galimos teorinės informacijos entropijos vertės, kuri gali būti su esamu grupių skaičiumi. Ši vertė bus didžiausia tada, kai visų grupių tikimybės bus lygios. Maksimali n grupių Šenono entropija:

$$H_{max} = \sum_{i=1}^n \frac{1}{n} \log_2 \frac{1}{n} = \log_2 \frac{1}{n} \quad (9)$$

1.5.4 Lango pločio parametras

Apskaičiuojama informacijos entropija yra labai nepastovus dydis ir apskaičiuojant ją mažuose intervaluose (pvz.: tokiuose, kad viename intervale esančios grupės yra visiškai pakeičiamos kitomis po vieno žingsnio) yra susiduriama su dideliais apskaičiuotų verčių svyravimais. Dėl to reikia įvesti lango pločio parametą (darbo metu jis žymimas P), kuris pasako kelių prieš taiėjusių intervalų paketų skaičiaus vertės yra įtraukiamos apskaičiuojant naują intervalo vertę. Šis parametras yra vienas iš esminių dydžių, kurio parinkimas lemia kaip efektyviai bus aptinkamos atakos.

1.6 Atstumo tarp tinklo tėkmių apskaičiavimas

Aptikus tinklo anomaliją, reikia metodo, kurį naudojant išskirtume DDoS atakas nuo legalių vartotojų srauto. Antras pavyzdys (kai tinkle atsirado daug naujų grupių) puikiai parodo, kad DDoS ataka gali būti visiškai neatskiriama nuo legalių vartotojų siunčiamų paketų srauto padidėjimo (angl. flash crowd). Pavyzdyje papildomai atsiradusios $n_2 = 900$ grupės gali būti tiesiog staiga prisijungę nauji vartotojai. Šios atakos dar yra vadinamos legalius vartotojus pamėgdžiojančiomis atakomis (angl. network behaviour mimicking DDoS attacks). Yra metodų, kuriuos galima naudoti bandant išskirti tokias DDoS atakas. Vienas iš jų yra paremtas

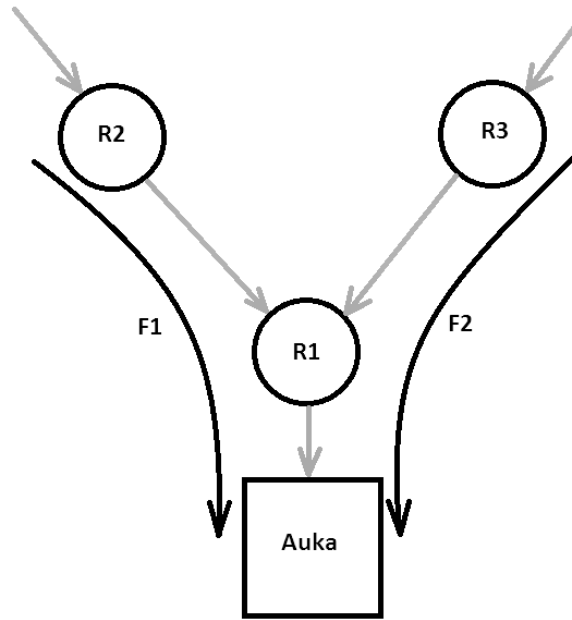
atstumo (angl. distance) (informacijos teorijoje dar vadinamo reliatyvia entropija) tarp dviejų tinklo t kmi  (2 pav.) apskai iavimu [4]. Tinklo t kme  iuo atveju yra vadinamas siun iam  paket , kurie turi bendr  paskirties adres  srautas. Atstumo tarp dviej  sraut  apskai iavimas yra id ji kai pana us į entropijos apskai iavim . Algoritm  galima išskirstyti į septynis pagrindinius  ingsnius [4].

1. Aptinkama įtartina t km .
2. Suskai iuojamas per tam tikr  laiko interval  t km je registruot  paket  skai ius.
3. Paketai suskirstomi į ma esnius intervalus.
4. Apskai iuojamos paket  pasiskirstymo ma esniuosiuose intervaluose tikimyb s.
5. Tikimyb s naudojame atstumui tarp tinklo t kmi  apskai iuoti.
6. Skirtinguose mar rutizatoriuose gauti atstumai lyginami tarpusavyje.
7. Jeigu aptinkama ataka, atakos t km s paketai atmetami.

Atstumas apskai iuojamas pagal formul :

$$D(p||q) = \sum_{x \in X} p(x) \log_b \frac{p(x)}{q(x)} \quad (10)$$

 ia x yra tam tikra grup , o X - aib  apimanti visas grupes. Idealiu atveju, t. y. jeigu abu srautai yra identi ki, $D(p||q) = 0$.  inoma, realiam tinkle to niekada nebus, bet atstumas D tarp dviej  atak  t kmi  tur t  b ti daug ma esnis u  atstum  tarp atakos bei legali  vartotoj  siun iam  paket  srauto t km s.  is DDoS atak  išskyrimo metodas remiasi keliomis s lygomis. Viena iš j  yra ta, kad dvi atakos t km s yra generuojamos tapa iai u kr st  kompiuteri , kas neb tinai turi b ti tiesa. Kita s lyga yra sistemos stabilumas ma uose laiko intervaluose (t. y. registruojam  paket  skai ius did ja arba ma  ja pakankamai ne ymiai), o tai da niausiai visada ir yra tiesa (nes atak  k r jai stengiasi m gd ioti legali  vartotoj  sraut , kuris da niausiai auga minu i , o ne milisekund i  intervaluose). Be to,  is algoritmas paremtas tuo, jog tinkle yra registruojami daugiau nei vienos t km s paketai, nes kitaip neb t  galima apskai iuoti atstumo.



2 pav. Dvi tinklo tėkmės (angl. network flows) $F1$ ir $F2$.

1.7 Laikinės priklausomybės prognozavimas

DDoS atakų aptikimo užduotį galime paversti į tam tikrų verčių kitimo laike prognozavimo užduotį. Analizuojamos vertės gali būti registruotų per pasirinktą laiko intervalą paketų skaičius ar tame intervale apskaičiuotos informacijos entropijos vertės. Sugebėjimas teisingai nuspėti, kaip tiriamos vertės elgsis ateityje gali padėti aptikti DDoS ataką. Jeigu pavyksta teisingai prognozuoti tinklo veiką, kai žinome, jog tinkle nėra jokių atakų, galime registruoti tinklo anomalijas tada, kai prognozuojamos vertės išsiskiria su realiai apskaičiuojamomis vertėmis.

Vystantis chaoso teorijai ir jos pritaikymui, buvo atrasta daug skirtingų metodų, kaip būtų galima nuspėti chaotiškas laikines priklausomybes. Visi metodai gali būti suskirstyti į dvi pagrindines grupes: globalius ir lokalius. Globalūs metodai naudoja visą laikinę skalę, kad rastų tikslią funkciją nusakančią kaip vertės išsidėsto laikinėje skalėje. Tačiau šie metodai turi ir savo trūkumų. Pavyzdžiui tai, kad jeigu įskaitome naujas vertes, visi funkciją nusakantys parametrai pasikeičia, to pasekoje naujų parametru verčių apskaičiavimas atima nemažai laiko. Lokalūs metodai išsprendžia šią problemą, tačiau turi trūkumą, jog funkcija, nusakanti verčių išsidėstymą, yra apskaičiuojama naudojant tik dalį prieš tai gautos informacijos. [5] Šiame darbe yra naudojamame metode naudojamos tik lokalias vertės laikinės priklausomybės prognozavimui.

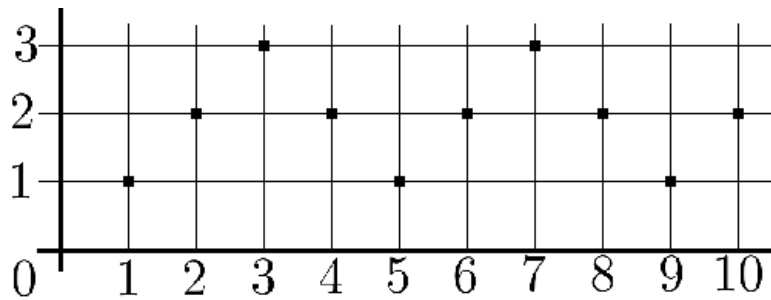
Taken teorema nurodo sąlygas, kurios turi būti įgyvendintos, jog būtų galima prognozuoti chaotišką laiko priklausomybę iš tam tikro skaičiaus šios priklausomybės stebėjimų [5]. Laikantis šių sąlygų, prognozuojama laikinė priklausomybė turėtų sutapti su realia, jeigu nėra staigių funkcijos verčių pokyčių. Viena iš sąlygų yra ta, jog dimensių, naudojamų fazinės erdvės atstatymui skaičius, yra pakankamas. Jeigu turime diskrečią laikinę priklausomybę $x(n)$, kur $n = 1, 2, 3, \dots, N$, sudarytas fazinės erdvės vektorius atrodytų taip:

$$X(n) = [x(n), x(n - \tau), \dots, x(n - (m_{opt} - 1)\tau)]^T \quad (11)$$

čia $n = (m - 1)\tau + 1, (m - 1)\tau + 2, \dots, (m - 1)\tau + N$, kur m yra dimensių numeris. Teisingam prognozavimui turi būti teisingai parenkami du parametrai: m_{opt} t. y. optimalus dimensių skaičius bei τ , kuris nurodo laiko skalės poslinkį tarp vektorių. Kai pavyksta parinkti teisingą dimensių skaičių m_{opt} bei užlaikymo laiką τ , prognozuojamos vertės turėtų sutapti su realiomis vertėmis.

1.8 Fazinės erdvės apskaičiavimas

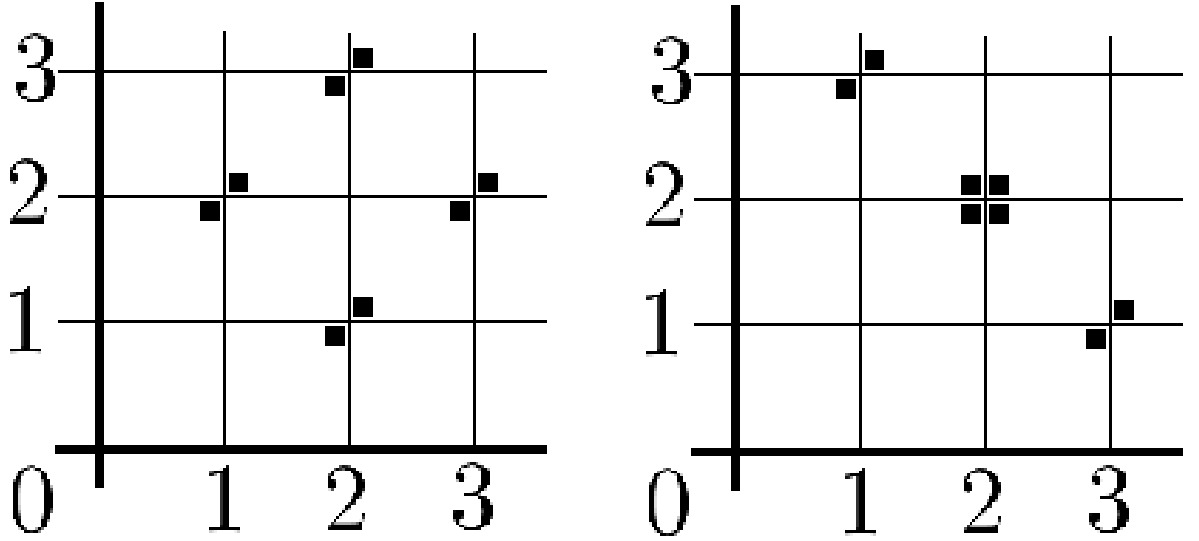
Pirmas žingsnis, norint rasti optimalų laiko poslinkį verčių prognozavimui, yra kaimyninių taškų fazinėje erdvėje, sudarytoje iš $X(n)$ vektorių, radimas. Sakykime, kad nagrinėjame paprastą laikinę priklausomybę, pavaizduotą 3 pav. Tiriant patį paprasčiausią atvejį, kai dimensių skaičius $m = 2$ išanalizuosime du atvejus, kai laiko poslinkis lygus vienetui ir kai laiko poslinkis lygus dviem abscisių ašies žingsniams.



3 pav. Laikinė priklausomybė, abscisių ašyje atidėta laikinė skalė, o ordinačių ašis atitinka užfiksuotas vertes.

Pirmu atveju fazinėje erdvėje bus atidedami vektoriai $X_1(10) = [2, 1]^T$, $X_2(9) = [1, 2]^T$,

$X_3[8] = [2, 3]^T$, $X_4[7] = [3, 2]^T$, $X_5[6] = [2, 1]^T$, $X_6[5] = [1, 2]^T$, $X_7[4] = [2, 3]^T$, $X_8[3] = [3, 2]^T$ (vektorių ilgiai turi sutapti bei būti lygūs dimensių skaičiui). Fazinė erdvė, sudaryta iš šių vektorių, yra pavaizduota 4 pav. kairėje pusėje. Fazinės erdvės koordinatę $[2, 1]$ atitinka du vektoriai X_1 bei X_5 . Šie vektoriai yra kaimyniniai taškai fazinėje erdvėje, taigi jie gali būti naudojami tolimesniam laikinės priklausomybės prognozavimui.



4 pav. Fazinė erdvė rekonstruota iš laikinės sekos 3 pav. pavaizduotos priklausomybės duomenis ir skirtingu laiko postūmius. Kairėje esanti fazinė erdvė atitinka vienetinį, o dešinėje - dviejų skalės verčių laikinę postūmį.

Antru atveju fazinėje erdvėje bus atidedami vektoriai $X_1(10) = [2, 2]^T$, $X_2(9) = [1, 3]^T$, $X_3(8) = [2, 2]^T$, $X_4(7) = [3, 1]^T$, $X_5(6) = [2, 2]^T$, $X_6(5) = [1, 3]^T$, $X_7(4) = [2, 2]^T$, $X_8(3) = [3, 1]^T$. Fazinė erdvė nubraižyta atidėjus šiuos vektorius atitinkamuose koordinatinių plokštumos taškuose yra pavaizduota 4 pav. dešinėje pusėje. Šiuo atveju fazinės erdvės koordinatę $[2, 1]$ atitinka keturi vektoriai X_1 , X_3 , X_5 bei X_7 . Visi šie vektoriai fazinėje erdvėje yra kaimyniniai taškai. Tačiau atstumas tarp jų nėra optimalus atstumas tolesniam verčių prognozavimui (nes dviejų laiko žingsnių postūmis yra lygus pusei 3 pav. pavaizduotos priklausomybės periodui ir atlikus šį postūmį kreivė netampa panaši, o invertuojasi), taip yra dėl to, kad kaimyninių taškų ieškojimas įtraukia tik tik pavienius vektorius (kurie gali pasikartoti dažniau nei nuspėjamos tiesinės priklausomybės periodas). Teisingai kaimyninių taškų paieškai reikėtų atsižvelgti ne tik į pavienio taško vertę. Jeigu ieškodami vektoriaus $X_1 = [x(10), x(8)]^T$ kaimyninių taškų, atsižvelgsime

ir į ankstesnius vektorius ($X_2, X_3, \dots$) taip būtų randami tik teisingi kaimyniniai taškai.

Fazinei erdvei atvaizduoti reikėtų pasirinkti tokį laiko postūmį, kad paslinkta bei dabartinė (t. y. seka prasidedanti nuo paskutinės registruotos vertės) sekos būtų kuo nepanašesnės. Tačiau kartu paslinkta seka su dabartine turėtų išlaikyti loginį deterministinį sąryšį (t. y. dabartinės sekos vertės turėtų priklausyti nuo paslinktos sekos verčių). Tokia fazinė erdvė teikia daugiausia informacijos.

1.9 Bendros informacijos apskaičiavimas

Jeigu fazinėje erdvėje atvaizduojami vektoriai yra sudaromi naudojant nedidelį laiko poslinkį, vertės, atitinkančios fazinės erdvės koordinates, yra labai artimos. Naudojant dvi dimensijas bei mažą laiko postūmį, fazinė erdvė bus tiesė [6]. Kad to išvengtume, reikia, kad rekonstruojamos fazinės erdvės taškų ašių vertės būtų kuo labiau nepriklausomos. Tuo tikslu, optimaliam laiko poslinkiui rasti, yra naudojamas bendros informacijos apskaičiavimo metodu pagrįstas algoritmas. Laiko skalė yra paslenkama per tam tikrą $\tau_1 = \delta t$ ir apskaičiuojama paslinktos sekos verčių bendrąją informaciją (angl. mutual information) su nepaslinktomis vertėmis. Skalė slenkama toliau į $\tau_i = i * \delta t$, kur $i = 1, 2, 3, \dots$ tol, kol bendra informacija pasieks pirmą minimumo tašką, t. y. rasime pirmą laiko poslinkį, kai paslinktos laiko priklausomybės vertės daugiausiai nutolsta nuo pradinių verčių, tačiau išlaiko deterministinį sąryšį [6].

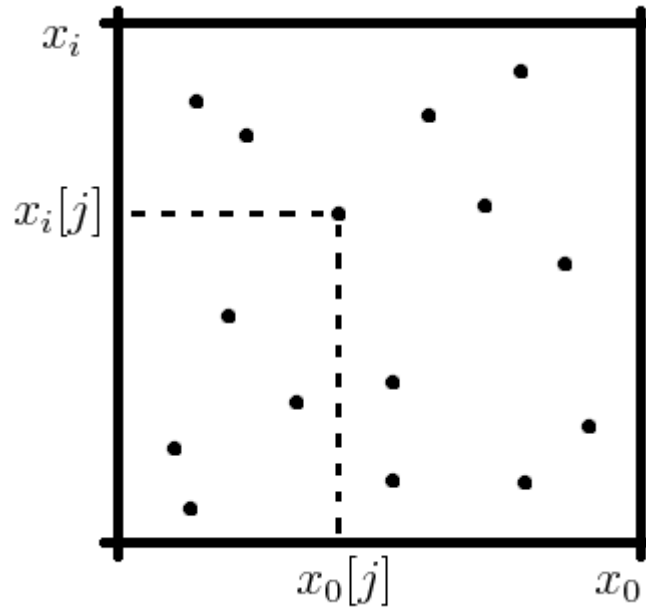
Bendra informacija yra apskaičiuojama pagal formulę (12):

$$I(x_0, x_i) = H(x_0) + H(x_i) - H(x_0, x_i) \quad (12)$$

kur $H(x_0)$ bei $H(x_i)$ yra sekų x_0 (nepaslinktos sekos) bei x_i (paslinktos per $\tau = i * \delta t$) entropijos, o $H(x_0, x_i)$ yra bendra sekų x_0 ir x_i informacijos entropija.

Bendrosios informacijos entropiją, abiejų laikinių priklausomybių x_0 bei x_i vertės yra naudojamos kaip dvi statmenos koordinačių ašys, o verčių poros $x_0[j]$ ir $x_i[j]$ atvaizduojamos kaip taškai koordinačių plokštumoje (žr. 5 pav.).

Koordinačių ašių skalės prasideda nuo mažiausių sekos verčių ir tęsiasi iki didžiausių sekos verčių. Kad galėtume apskaičiuoti informacijos entropijas, kurios naudojamos bendros infor-



5 pav. Koordinačių plokštuma bendrai informacijai tarp dviejų laikinių priklausomybių skaičiuoti

macijos formulėje (12), reikia suskaidyti koordinačių ašių skales segmentais. Segmentai gali būti parenkami pagal skirtingus kriterijus (žr. 6 pav.) ir nuo šių kriterijų parinkimo priklauso apskaičiuotos bendros informacijos vertės [7]. Blogai parinkus skalių į segmentus skirstymo kriterijus galime nustatyti blogą bendros informacijos minimumo tašką ir dėl to rasta optimalaus laiko poslinkio vertė nebus tiksli. Tikimybė vertėms pakliūti į tam tikrą segmentą x_{is} , $s = 0, 1, 2, \dots$ yra lygi verčių esančių segmente skaičiui padalintam iš viso verčių esančių koordinačių plokštumoje skaičiaus $P_{is} = N_s/N$. Taigi, atskirų sekų entropijos yra apskaičiuojamos pagal formulę (13):

$$H(x_i) = - \sum_{s=0} P(x_{is}) \log_2 P(x_{is}) \quad (13)$$

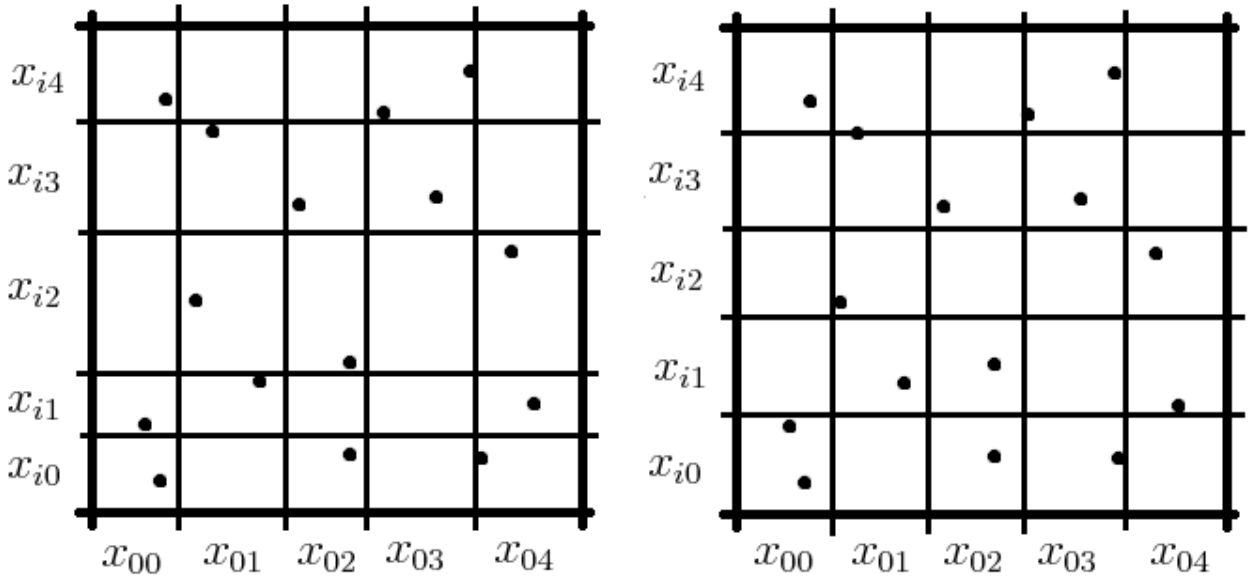
Norint apskaičiuoti dviejų sekų bendrą informaciją (angl. mutual information) papildomai reikia rasti tikimybes, kad tam tikra dviejų verčių seka pasikartos. Šią pasirinktų verčių tikimybę randame padaliję atitinkamame ploto segmente esančių verčių skaičių iš viso verčių

skaičiaus $P_{isq} = N_{sq}/N$. Bendra informacijos entropija yra apskaičiuojama pagal formulę:

$$H(x_0, x_i) = H(x_i, x_0) = - \sum_{sq} P(x_{0s}, x_{iq}) \log_2 P(x_{0s}, x_{iq}) \quad (14)$$

Ištačius atitinkamas (13) bei (14) lygčių išraiškas į (12) gauname:

$$I(x_0, x_i) = \sum_s \sum_q P(x_{0s}, x_{iq}) \log_2 \left(\frac{P(x_{0s}, x_{iq})}{P(x_{0s})P(x_{iq})} \right) \quad (15)$$



6 pav. Koordinačių ašių skalių skirstymas segmentais. Kairėje pavaizduota koordinačių plokštuma yra suskirstyta taip, kad tikimybė vertėms patekti į kiekvieną stulpelį ir eilutę būtų vienoda. Dešinėje koordinačių ašių skalės suskirstytos vienodais intervalais

Abu metodai pavaizduoti 6 pav. turi savo trūkumų [7]. Fraser algoritmas, naudojamas optimalaus laiko poslinkio nustatymui, naudoja pirmąjį skirstymo segmentais algoritmą (reikšmių skaičiai segmentuose sutampa). Šio algoritmo metu koordinačių ašių skalės yra suskirstomos į keturis segmentus, taip, kad kiekviename segmente x_{0s} būtų vienodi verčių skaičiai. Žinoma, kad susidarę stačiakampiai nebūtinai turės vienodus verčių skaičius (tai galime pamatyti ir 6 pav.). Toliau yra tikrinama ar stačiakampiuose taškai išsidėstę tankiai (t. y. ar stačiakampyje taškai turi savo struktūrą) ar retai. Jeigu taškai išsidėstę tankiai, šį stačiakampį vėl daliname į keturis segmentus. Taip algoritmas tęsiasi tol, kol stačiakampiai nebeturi vidinių struktūrų. Šio algoritmo minusas yra tas, kad skirstymas į segmentus užima daug laiko, o

struktūrų saugojimas eikvoja nemažą kiekį atminties.

Algoritmas, kurio metu koordinačių ašių skalės yra padalijamos vienodais atstumais, palyginus su Fraser algoritmu yra spartesnis ir lengviau įgyvendinamas. Tačiau gauti rezultatai ženkliai priklauso nuo pasirinkto segmentų skaičiaus [7]. Padalijus skales į per didelę kiekį segmentų, atsiranda per daug stačiakampių, kurie turi tik keletą taškų arba neturi nei vieno taško ir tai iškraipo tikimybinį pasiskirstymą. Tuo pačiu, jeigu parinksime per mažai segmentų, sudarę segmentai turės vidines struktūras dėl kurių vėlgi gausime netikslias bendros informacijos vertes.

1.9.1 Cellucci algoritmas

Cellucci algoritmas buvo sukurtas, naudojant prielaidą, kad dvi sekos x_0 bei x_i yra statistiškai nepriklausomos. Jeigu ši prielaida galioja, tai verčių (iš x_0 ir x_i sekų) poros tolygiai pasiskirsto sukonstruotoje koordinačių plokštumoje. Tokiu atveju, jeigu $N(x_{0s})$ yra taškų skaičius x_0 sekos s segmente ir $N(x_{iq})$ yra taškų skaičius x_0 sekos q segmente, taškų skaičius (s, q) stačiakampyje yra žinomas skaičius:

$$N(x_{0s}, x_{iq}) = NP(x_{0s})P(x_{iq}) = N\left(\frac{N(x_{0s})}{N}\right)\left(\frac{N(x_{iq})}{N}\right) = \frac{N(x_{0s})N(x_{iq})}{N} \quad (16)$$

Kai verčių sekas į segmentus skirstome naudodami vienodų tikimybių segmentus ir atskirų sekų elementų skaičiai sutampa, galime teigti, jog $P(x_{0s}) = P(x_{iq}) = \frac{1}{N_E}$, kur N_E yra segmentų skaičius. Tada (16) lygtis tampa:

$$N(x_{0s}, x_{iq}) = NP(x_{0s})P(x_{iq}) = \frac{N}{N_E^2} \quad (17)$$

Iš (17) lygties matome, kad jeigu taškų skaičius tam tikrame plokštumos stačiakampyje yra žinomas, tai segmentų skaičius N_E irgi yra žinomas. Cellucci algoritme naudojamas optimalus taškų plokštumos stačiakampyje skaičius $N(x_{0s}, x_{iq}) \geq 5$, nes tokiu atveju tikimybė, kad stačiakampyje egzistuos vidinės struktūros yra nedidelė. Jeigu naudosisime, $N(x_{0s}, x_{iq}) = 5$ (17) lygtis

tampa:

$$N_E \leq \left(\frac{N}{5}\right)^{1/2} \quad (18)$$

Paprastai yra naudojamas didžiausias N_E tenkinantis (18) lygtį. Tada tais atvejais, kai N_E dalina visą taškų skaičių N be liekanos (N yra N_E kartotinis), kiekvienas segmentas turi vienodą taškų skaičių ir bendros informacijos išraiška (15) tampa:

$$I(x_{0s}, x_{iq}) = \sum_{s=0}^{N_E-1} \sum_{q=0}^{N_E-1} P(x_{0s}, x_{iq}) \log_2(N_E^2 P(x_{0s}, x_{iq})) \quad (19)$$

Tačiau, jeigu N nėra N_E kartotinis, taškų skaičius segmentuose pasiskirsto ne vienodai, taigi reikia kito metodo N_E nustatymui. Šiuo atveju Cellucci iškėlė du kriterijus, kuriuos turėtų atitikti N_E . Vienas iš jų - 80% plokštumos stačiakampių privalo turėti daugiau nei penkis taškus ($N_{80\%}(x_{0s}, x_{iq}) \geq 5$). Antras - kiekvienas stačiakampis privalo turėti bent vieną tašką. Tam, kad būtų patikrinta ar segmentų skaičius patenkina pirmą reikalavimą, galime tikrinti, kaip tolygiai pasiskirsto taškai, naudodami:

$$\chi^2 = \sum_{s=0}^{N_E-1} \sum_{q=0}^{N_E-1} \frac{(N_{80\%}(x_{0s}, x_{iq}) - N(x_{0s}, x_{iq}))^2}{N(x_{0s}, x_{iq})} \quad (20)$$

Nepriklausomi lygties (20) kintamieji (angl. degrees of freedom) apskaičiuojami žinant, kad kiekviena suma turi vienetu mažiau, nei sumoje narių, nepriklausomų kintamųjų. Kadan- gi sumuojame per visus plokštumos stačiakampius, tai $\nu = (N_E - 1)^2$. Tada tikimybė, kiek plokštumos stačiakampių turi daugiau nei penkis taškus [7]:

$$P = G\left(\frac{\nu}{2}, \frac{\chi^2}{2}\right) = 1 - \frac{1}{\Gamma(\nu/2)} \int_0^{\chi^2/2} e^{-t} t^{\nu/2-1} dt = \frac{1}{\Gamma(\nu/2)} \int_{\chi^2/2}^{\infty} e^{-t} t^{\nu/2-1} dt = 0,8 \quad (21)$$

čia $\Gamma(\nu/2)$ yra gama funkcija, taigi N_E gali būti apskaičiuotas iš (20) ir (21).

Celluci algoritmas gali efektyviai apskaičiuoti dviejų sekų bendrą informaciją tais atvejais, kai bendras taškų plokštumoje skaičius yra segmentų skaičiaus kartotinis. Tačiau kitais atvejais,

sprendimas tampa griozdiškas bei sunkiai įgyvendinamas. Laikai reikalingi segmentų skaičiui apskaičiuoti taip pat stipriai išauga [7].

1.9.2 Išvestiniai bendros informacijos apskaičiavimo algoritmai

Fraser bei Cellucci algoritmų kombinacija eliminuoja silpnąsias šių algoritmų savybes. Visi šie algoritmai bendrai informacijai apskaičiuoti prasideda nuo to, jog segmentų skaičius yra apskaičiuojamas pagal (18) (naudojama didžiausia šių nelygybę tenkinant vertė). Tada apskaičiuojamas didžiausias sveikas skaičius $N_{\square}$, kuris gali būti lygus arba mažesnis už N/N_E . Šis skaičius atitinka taškų, esančių kiekviename iš segmentų (išskyrus paskutinį, jeigu N nėra N_E daugiklis), skaičių. Jeigu N yra N_E daugiklis, bendrą informaciją galime apskaičiuoti iškart, kitais atvejais galime naudoti vieną iš trijų algoritmų [7].

Pirmas algoritmas apskaičiuoja bendrą informaciją, atmetus paskutinę eilutę ir paskutinį stulpelį. Tai galima daryti, nes Cellucci algoritmas paremtas dideliu kiekiu taškų bendra informacija. Pašalinus paskutinį stulpelį bei paskutinę eilutę, maksimalus atmetamų taškų skaičius yra tik $N_{\square} - 1$, taigi šių taškų ignoravimas neženkliai įtakos sekų bendros informacijos vertę.

Antras algoritmas apskaičiuoja bendrą informaciją naudodamas originalią plokštumą neatsižvelgdamas į tai ar N yra N_E kartotinis ar ne. Optimalus taškų skaičius plokštumos stačiakampyje yra penki, nes esant tokiam taškų skaičiui galime laikyti, kad plokštumos stačiakampiai neturi vidinių struktūrų. Tada taškų skaičius paskutinės eilutės bei paskutinio stulpelio stačiakampiuose yra mažiau nei penki taigi nėra nusižengiama Cellucci kriterijui, kuris reikalauja, jog plokštumos stačiakampiai neturėtų vidinių struktūrų. Taigi, galime laikyti, jog šis algoritmas irgi yra pakankamai racionalus.

Trečias algoritmas dirbtinai pakeičia taškų esančių paskutinėje eilutėje bei paskutiniame stulpelyje skaičius naudodamas dviejų priešpaskutinių stulpelių ir dviejų priešpaskutinių eilučių taškų santykius kaip atitinkamus daugiklius. Kampinis plokštumos stačiakampis, kuris priklauso paskutinei eilutei bei paskutiniam stačiakampiui turi būti dauginamas tik vieną kartą.

1.9.3 Algoritmo įgyvendinimas

Bendros informacijos apskaičiavimo sparta priklauso ne tik nuo pasirinkto algoritmo, bet ir nuo jo įgyvendinimo. Pagrindinis žingsnis bendros informacijos apskaičiavimo algoritme yra koordinatinių plokštumos, sudarytos iš atvaizduojamų sekų verčių, sudarymas.

Galime pastebėti, jog jeigu skirstome koordinatinių ašis vienodų tikimybių segmentais, tai į kurį segmentą paklius tam tikra vertė priklauso tik nuo jos reliatyvaus dydžio (lyginant su kitomis sekos vertėmis) ir nepriklauso nuo absoliučios vertės (t. y. svarbu tik kelinta vertė pagal vertės dydį yra pasirinkta vertė). Taigi jeigu pakeisime absoliutų vertės dydį, bet išlaikydami vertės eilę sekoje, bendros informacijos apskaičiavimo atžvilgiu niekas nepasikeis.

Išnaudodami šią savybę, galime išrikiuoti sekos vertes pagal dydį išsaugant šių verčių originalios pozicijos indeksą. Tada, vietoje išrikiuotų sekos verčių įrašyti jas atitinkančius originalios pozicijos indeksus. Dėl to yra sudaromos naujos verčių sekos, kurių visos vertės yra intervale nuo 0 iki $N - 1$, kur N yra sekos verčių skaičius. Matome, jog nesvarbu kokio dydžio vertės buvo originaliose sekose, naujų sekų vertės bus sveiki skaičiai mažesni už sekos verčių skaičių. Taigi koordinatinių plokštumos sudarymas tampa daug paprastesnis. Naudosime dviejų lygmenų ciklą, jog pereitumėme visus naujai sudarytų sekų $x_{0_{nauj.}}$ bei $x_{i_{nauj.}}$ vertes $[x_{0_{nauj.}}(j_1), x_{i_{nauj.}}(j_2)]$. Kiekvienam taškui apskaičiuosime plokštumos stačiakampį, kuriam jis priklauso. Taškas priklausys segmentui, kurio koordinatės plokštumoje yra sveiki skaičiai mažesni už $x_{0_{nauj.}}/N_{\square}$ ir $x_{i_{nauj.}}/N_{\square}$. Apskaičiuotam stačiakampiui pridėsime vienetą prie kintamojo, kuriame saugojimas taškų patekusių į stačiakampį skaičius. Matricą kurioje bus saugojamos šių kintamųjų vertės pažymėkime C .

Sudarius tokią matricą, bendra informacija apskaičiuojama pagal (jeigu yra naudojamas pirmas išvestinis algoritmas):

$$I(x_0, x_i) = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} \frac{C(m, n)}{N^*} \log_2 \left[\frac{C(m, n)N^*}{N_{\square}^2} \right] \quad (22)$$

čia $N^* = N_1 k$, o k yra didžiausias sveikas skaičius, kuris yra mažesnis už ar lygus $N/N_{\square}$.

Jeigu naudojame antrą išvestinį algoritmą:

$$I(x_0, x_i) = \sum_{m=0}^k \sum_{n=0}^k \frac{C(m, n)}{N} \log_2 \left[\frac{C(m, n)N}{N(m, n)} \right] \quad (23)$$

kur

$$N(m, n) = \begin{cases} N_{\square} N_{[\dots]} & m = k + 1 \text{ arba } n = k + 1 \\ N_{[\dots]}^2 & m = n = k + 1 \\ N_{\square}^2 & \text{kitais atvejais} \end{cases}$$

kur, $N_{[\dots]}$ yra liekana, kurią gauname padalinę visą taškų skaičių N iš viename segmente esančio taškų skaičiaus $N_{\square}$ (t. y. $N_{[\dots]}$ yra paskutiniame koordinatinių plokštumos stulpelyje (arba eilutėje) esančių taškų skaičius). Jeigu naudosime trečią išvestinį algoritmą, tai bendra informacija bus apskaičiuojama pagal:

$$I(x_0, x_i) = \sum_{m=0}^k \sum_{n=0}^k \frac{C^*(m, n)}{(k+1)N_{\square}} \log_2 \left[\frac{C^*(m, n)(k+1)}{N_{\square}} \right] \quad (24)$$

kur

$$C^*(m, n) = \begin{cases} C(m, n) & 0 < m, n \leq k \\ \frac{N_{\square}}{N_{[\dots]}} C(m, n) & \text{kitais atvejais} \end{cases}$$

1.10 Prognozavimo uždavinys

Suradus kaimyninius taškus fazinėje erdvėje, žinome, kuriuos taškus galime naudoti laikinės priklausomybės prognozavimui. Rekonstruojant fazinę erdvę nagrinėtame pavyzdyje laikinė priklausomybė turėjo tikslų periodą, dėl to laiko poslinkiai tarp skirtingų kaimyninių taškų sutapo, t. y. kaimyninių vektorių indeksai (X_1, X_5, X_9) buvo atskirti vienodu žingsnių skaičiumi, bet tai nebūtinai turi būti tiesa nagrinėjant realų atvejį).

Prognozavimo uždavinys yra paremtas deterministinio sąryšio egzistavimu tarp dabartinių ir ateityje užregistruotų laikinės priklausomybės verčių. Jeigu šis sąryšis egzistuoja, tai turėtų egzistuoti ir šio sąryšio tęstinumas, t. y. anksčiau registruotos vertės turėtų išlaikyti tokį patį deterministinį sąryšį su dabartinėmis laikinės priklausomybės vertėmis [5]. Einamas laikinės

priklausomybės vertes galime nuspėti naudodami:

$$x(n+1) = a_0 + \sum_{i=1}^{m_{opt}} a_i(n - (i-1)\tau_{opt}) = AY(n_{pask}) \quad (25)$$

čia $A = [a_0, a_1, a_2, \dots, a_{m_{opt}}]$, o $Y(n_{pask}) = [1, X(n_{pask})^T]^T$, kur $X(n_{pask})^T$ yra vektorius iš (11) lygties, kur n_{pask} yra paskutinė užregistruota laikinės priklausomybės vertė. Norint nuspėti, kokios vertės bus registruojamos ateityje, reikia apskaičiuoti koeficientų matricą A . Šią koeficientų matricą galime apskaičiuoti iš lygties $AB = D$, kur B yra $(m+1) \times k$ matrica (čia k yra kaimyninių taškų skaičius). Matricos B i -tasis stulpelis sudarytas iš $Y(n_i)$, kur $i = 1, 2, 3, \dots, k$ (n_i yra i -tojo kaimyninio taško indeksas), o D yra $1 \times k$ vektorius, kurio i -tasis stulpelis sudarytas iš $x(n_i + 1)$:

$$Y(n_i) = [1, x(n_i), x(n_i - \tau_{opt}), \dots, x(n_i - (m-1)\tau_{opt})]^T \quad (26)$$

$$D = [x(n_1 + 1), x(n_2 + 1), x(n_3 + 1), \dots, x(n_k + 1)] \quad (27)$$

Naudodami (25, 26, 27) lygtis užrašysime paprastą prognozavimui reikalingų sudaryti matricų pavyzdį, kai tyrinėjama verčių seka $x = [a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s]$. Tarkime, jog optimalus dimensių skaičius $m_{opt} = 3$ ir optimalus laiko poslinkis $\tau_{opt} = 2$. Sakykime $X(n) = [s, q, o]^T$ kaimyninių taškų indeksai $n_i = [n_o, n_l, n_h, n_e]$, tada $D = [x(n_{o+1}), x(n_{l+1}), x(n_{h+1}), x(n_{e+1})] = [p, m, i, f]$. Matrica $Y(n_o) = [1, o, m, k]^T$ naudojama pirmam matricos B stulpeliui sudaryti. Atitinkamai sudarius likusius matricų stulpelius, gauname galutinę lygtį šio pavyzdžio prognozavimo koeficientams rasti:

$$\begin{bmatrix} a_0 & a_1 & a_2 & a_3 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ o & l & h & e \\ m & j & f & c \\ k & h & d & a \end{bmatrix} = \begin{bmatrix} p & m & i & d \end{bmatrix} \quad (28)$$

Radę koeficientus, galime prognozuoti sekančią sekos vertę * naudodami (25) lygtį:

$$* = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \end{bmatrix} \times \begin{bmatrix} 1 \\ s \\ q \\ o \end{bmatrix} \quad (29)$$

taip algoritmą galime tęsti (vietoj paskutinių verčių naudojant prognozuotas) ir rasti sekančias sekos vertes.

2 DDOS atakų aptikimo tyrimo metodai

Darbo metu yra tikimasi ištirti skirtingų DDOS atakų aptikimo metodų efektyvumą. Tyrimui atlikti buvo parašytas programavimo paketas. Tyrimo metu yra tiriami metodai, paremti informacijos entropijos sumažėjimu bei laikinės priklausomybės prognozavimu, DDOS atakoms aptikti, tikrinamas jų efektyvumas naudojant modeliuojamus teorinius bei realių sistemų duomenis. Taip pat yra tikrinama ar teoriniai modeliai veikia, kai yra naudojami realūs (realaus tinklo) duomenys.

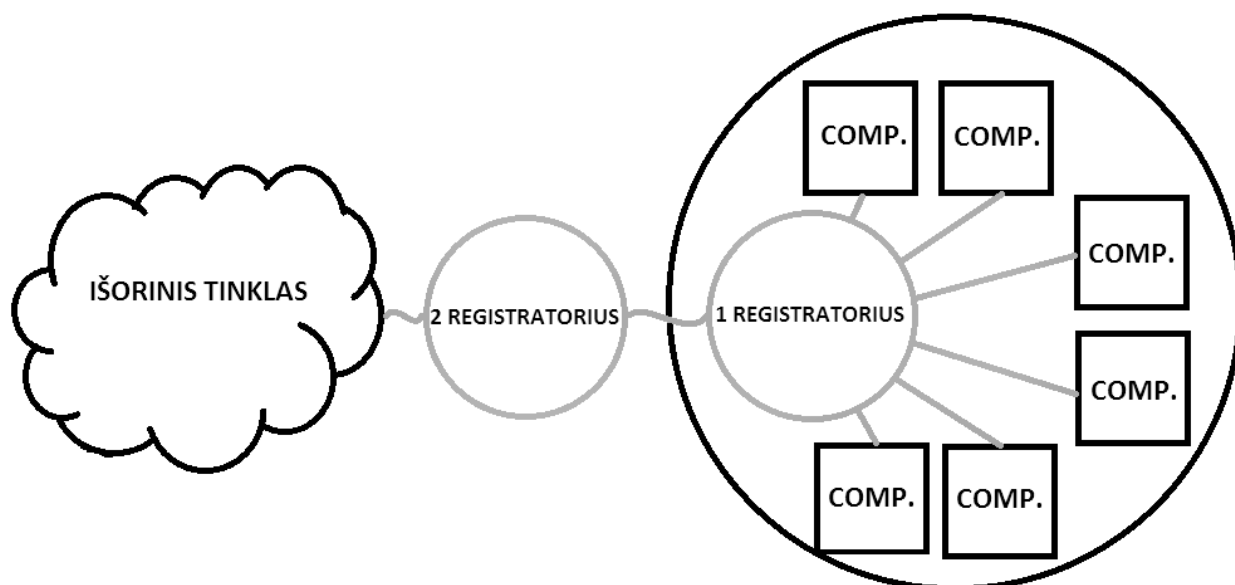
2.1 Duomenų parengimas

Norint, kad gautus rezultatus būtų galima panaudoti dirbant su realiomis sistemomis, reikia dirbti su realiais duomenimis. Iki šiol viešai yra prieinamos tik kelios DDOS atakoms tirti naudojamos duomenų bazės. Labiausiai tiriamos yra CAIDA (angl. Center for Applied Internet Data Analysis) bei MIT universiteto Lincoln laboratorijos publikuotos duomenų bazės.

Tyrime naudojami duomenys buvo surinkti MIT Linkolno laboratorijos kibernetinės apsaugos bei informacinių mokslų grupės. Ši grupė buvo remiama projektų agentūros DARPA ITO (angl. Defence Advanced Research Project Agency) bei AFRL/SNHS (angl. Air Force Research Laboratory). Duomenys yra surenkami registruojant tarp išorinio ir vidinio tinklo siunčiamus paketus dviem registratoriais. Šabloninė tyrinėjamo tinklo topologija pateikiama 7 paveiksle.

Pirmasis registratorius surenka visų vidiniame bei tarp vidinio ir išorinio tinklo siunčiamų paketų informaciją. O antrasis registruoja paketus siunčiamus tik tarp vidinio ir išorinio tinklų. Dėl to pirmo ir antro registratorių fiksuojama paketų informacija kartojasi (t. y. visi antro registratoriaus užfiksuoti paketai yra užfiksuojami ir pirmo registratoriaus). Kadangi tyrime nėra reikalinga išskirti išorinio bei vidinio tinklo paketus, yra naudojami tik pirmo registratoriaus užfiksuotų paketų informacija.

Ši informacija yra pateikiama „inside tcpdump data“ failuose. Šiame darbe yra naudojami sugretinti MIT universiteto Lincoln laboratorijos pateikiami duomenys. Trys 1999-ųjų metų kovo pirmos savaitės dienos buvo užrašytos viena ant kitos, taip padidinant registruojamų paketų per



7 pav. Šabloninė tyrinėjamo tinklo topologija. Pirmas registratorius surenka duomenis apie visus paketus siunčiamus vidiniame tinkle. Antras—registruoja siunčiamus paketus tarp išorinio ir vidinio tinklų.

tam tikrą laiką skaičių. Šis tankio padidėjimas leidžia tiksliau analizuoti registruotus paketus, neiškraipant duomenų realumo. Naudojami duomenų failai yra pateikiami [8–10]:

1. <https://www.ll.mit.edu/ideval/data/1999/training/week1/monday/inside.tcpdump.gz>
2. <https://www.ll.mit.edu/ideval/data/1999/training/week1/tuesday/inside.tcpdump.gz>
3. <https://www.ll.mit.edu/ideval/data/1999/training/week1/wednesday/inside.tcpdump.gz>

Duomenų failai apdorojami Wireshark (<https://www.wireshark.org/>) programa. Gaunama IP adresų (šaltinio bei paskirties), protokolo, ilgio (bitais), registracijos laiko, dienos eigoje registruoto paketo eilės numerio bei papildomo teksto informacija apie kiekvieną tinkle užregistruotą paketą. Tyrime nėra naudojama nei paketo eilės numerio, nei paketo ilgio, nei papildomo teksto informacija, nes ji nesuteikia papildomos informacijos nagrinėjant DDoS atakas. Visa kita informacija gali būti ir yra panaudota tyrimui atlikti. Duomenys yra importuojami į duomenų bazę ir indeksuojami (indeksavimas paspartina tyrimo metu naudojamas paketų paieškas). 8 pav. parodyta, kaip atrodo paketai duomenų bazėje (kiekviena eilutė saugo skirtingų paketų informaciją).

	id integer	timestamp timestamp without time zone	source character varying(255)	destination character varying(255)	protocol character varying(255)
1	1	1999-03-01 08:00:05.928772	Cisco 38:46:33	CDP/VTP/DTP/PagP/UDLD	CDP
2	2	1999-03-01 08:00:06.947983	172.16.0.1	224.0.0.9	RIPv2
3	3	1999-03-01 08:00:07.273717	Cisco 38:46:33	Cisco 38:46:33	LOOP
4	4	1999-03-01 08:00:07.616695	172.16.112.10	192.168.1.10	NTP
5	5	1999-03-01 08:00:07.618526	192.168.1.10	172.16.112.10	NTP
6	6	1999-03-01 08:00:11.967946	172.16.112.20	172.16.112.10	NTP
7	7	1999-03-01 08:00:11.96831	172.16.112.10	172.16.112.20	NTP
8	8	1999-03-01 08:00:17.27353	Cisco 38:46:33	Cisco 38:46:33	LOOP
9	9	1999-03-01 08:00:22.402159	172.16.113.50	172.16.112.10	NTP
10	10	1999-03-01 08:00:22.402585	172.16.112.10	172.16.113.50	NTP

8 pav. Paketų, saugomų duomenų bazėje, atvaizdavimas (iš pgAdmin III grafinės sąsajos).

2.2 Trumpas programos aprašymas

Programos kodas yra patalpintas: <https://github.com/VajetaItau/Bakalauras>. Pagrindinis programos kodas yra parašytas JAVA programavimo kalba. Aplikacijų serverio veikimui yra naudojama Spring Boot biblioteka. Hibernate biblioteka atlieka integraciją su duomenų baze. Taip pat yra naudojama PostgreSQL duomenų bazė.

Visas grafinis interfeisas yra aprašomas viename html faile (*index.html*). Vartotojo užklauskas aptarnauja klasė *MainController*, šioje klasėje yra apdorojami paduodami parametrai ir kviečiamas *DataService*, kur yra atliekama visa programos logika. Jeigu reikia bendrauti su duomenų baze, *DataService* kreipiasi į *PacketDao* klasę, kuri vykdo tiesiogines užklauskas į duomenų bazę.

DataService interfeisas yra pavaizduotas 9 pav. Yra pateikiamos visos pagrindinės funkcijos, kurios buvo naudojamos šiam darbui atlikti. Yra naudojamos šešios pagrindinės funkcijos.

1. Funkcija *uploadFileToDatabase* įkelia WireShark programa apdorotą failą į duomenų bazę. Galima paduoti kelis failų pavadinimus, dėl to, šiai funkcijai galima paduoti pavadinimų pasyvą.
2. *insertDDoSAttack* naudoja du parametrus įterpiamos DDoS atakos pozicijos laiko skalėje nustatymui. Parametras *destination* nurodo paskirties IP adresą.
3. Norint pašalinti DDoS atakas, reikia iškvieisti funkciją *removeDDoSAttacks*.
4. Entropijos apskaičiavimui yra naudojama funkcija *getEntropy*. Naudojant parametą *increment* yra nustatomas intervalo ilgis, o su parametru *windowWidth* - lango plotis.

5. Funkcija *getOptimalTimeDelay* apskaičiuoja optimalų laiko poslinkį. Parametras *type* nurodo, kurios kreivės poslinkis bus apskaičiuojamas (ar sinusoidės, ar realių duomenų kreivės), *pointCountList* nurodo kokius taškų skaičiai bus naudojami bendros informacijos apskaičiavime.
6. Funkcija *predict* apskaičiuoja prognozuojamus taškus. Taškai yra prognozuojami iki tol, kol apskaičiuotos vertės stipriai nutolsta nuo realių. Galime nurodyti dimensijų skaičiaus (*dimensionCount*) bei optimalaus laiko poslinkio (*optimalTimeDelay*) parametrus.

```
public interface DataService {  
  
    void uploadFileToDatabase(String[] fileNames)  
        throws GeneralException;  
  
    void insertDDoSAttack(Timestamp start, Timestamp end  
        , String destination)  
        throws GeneralException;  
  
    void removeDDoSAttacks();  
  
    String getEntropy(Timestamp start, Timestamp end  
        , Integer increment, Integer windowWidth)  
        throws GeneralException;  
  
    String getOptimalTimeDelay(Timestamp start, Timestamp end  
        , Integer increment, Integer windowWidth  
        , String type  
        , List<Integer> pointCountList)  
        throws GeneralException;  
  
    String predict(Timestamp start, Timestamp end  
        , Integer increment, Integer windowWidth  
        , String type  
        , Integer dimensionCount, Integer optimalTimeDelay)  
        throws GeneralException;  
}
```

9 pav. *DataService* sąsaja. Čia aprašomos visos pagrindinės programos funkcijos.

2.3 Informacijos entropijos apskaičiavimas

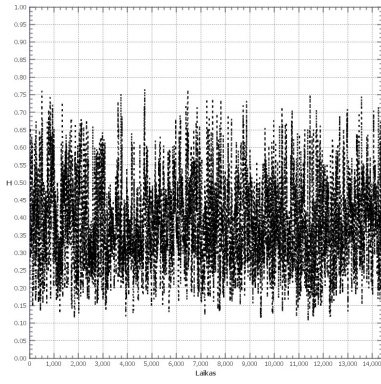
Pirmas žingsnis norint aptikti DDoS atakas, naudojant informacijos entropijos apskaičiavimu paremtais metodais, yra nustatyti apatinę informacijos entropijos ribą tinklo darbo režimo

metu (kai žinome, jog nėra DDoS atakų). Tai yra, reikia rasti mažiausią vertę, kurią gali pasiekti entropija, kai sistema nėra atakuojama. Žinant šią ribą, galima tikrinti ar apskaičiuotos tinklo (kai yra tikimasi DDoS atakų) entropijos vertės nusileidžia žemiau šios ribos. Apskaičiuotos entropijos vertėms esant mažesnėms nei nustatyta riba, galima manyti, jog yra vykdoma ataka ir imtis tolesnių veiksmų.

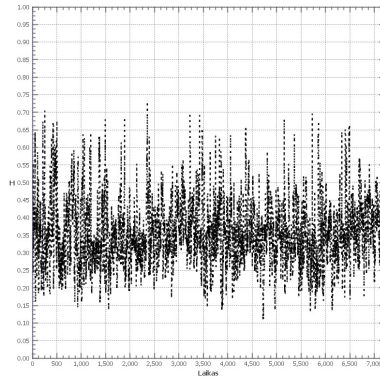
Žinoma, jog apskaičiuotos informacijos entropijos vertės priklauso nuo pasirinkto paketų grupavimo požymio, intervalo ilgio, kuriame yra grupuojami paketai bei lango pločio (kuris nurodo į kiek prieš dabartinį intervalą buvusių intervalų yra atsižvelgiama apskaičiuojant entropijos vertę) bei to intervalo tipo. Šiame darbe paketai yra grupuojami pagal jų paskirties IP adresus (t. y. grupavimo požymiu laikomas paskirties IP adresas), pasirinktas intervalo tipas yra laikinis (t. y. intervalai yra apribojami laiko intervalu). Intervalo ilgis bei lango plotis yra kintantys dydžiai.

DDoS atakų aptikimas naudojant informacijos entropijos sumažėjimo kriterijų gali teisingai aptikti atakas tik su optimaliu lango pločiu ir lango ilgiu. Intervalo ilgis bei lango plotis buvo parenkami taip, jog apskaičiuojant entropijos vertes būtų išvengta periodų, kai nėra registruojamas nei vienas paketas (pvz.: jeigu intervalo ilgis yra viena sekundė, o lango plotis 10, tai atsiradus dešimties sekundžių laiko periodams, kai nėra registruojamas nei vienas paketas, apskaičiuota Šenono entropija būtų lygi 0), nes egzistuojant tokiems periodams joks DDoS atakų aptikimas nėra galimas. Tokių periodų panaikinimui pasitarnauja ir aprašytas duomenų sutankinimas (tyrime nagrinėjamų savaitės dienų užrašymas viena ant kitos).

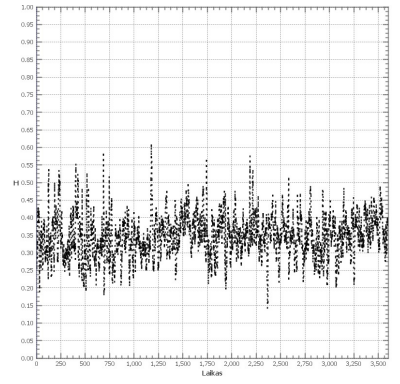
Sugrupavus paketus pagal paskirties adresą ir naudojant skirtingus lango pločius P bei skirtingus intervalo ilgius I gauname 10 pav. pavaizduotus grafikus. Galime pastebėti, jog didinant lango plotį informacijos entropijos kreivė yra glaudinama, o didinant intervalo ilgį skirtumai tarp gretimų entropijos verčių gali išaugti (t. y. atidedamų taškų skaičius per tą patį laiko intervalą sumažėja, dėl to verčių skirtumai tarp gretimų taškų gali padidėti). 10 pav. grafikuose yra atidėtos vienos valandos intervalo informacijos entropijos vertės. X ašies matavimo vienetai yra intervalo ilgiai, t. y. jeigu intervalo ilgis yra 1 sekundė, skaičius penki atitiks penktą sekundę, o 3600 - valandą.



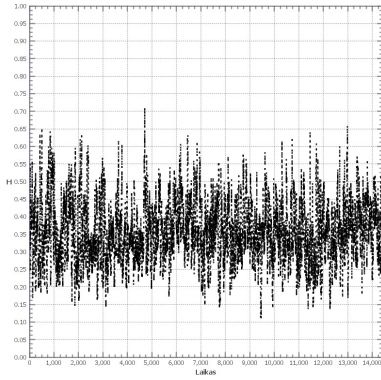
a $P = 10, I = 250$



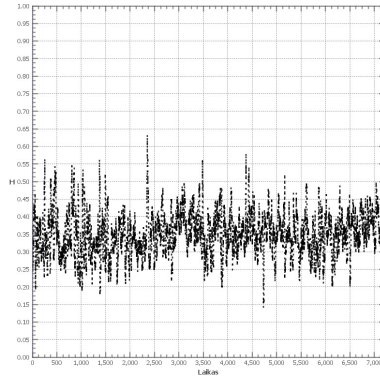
b $P = 10, I = 500$



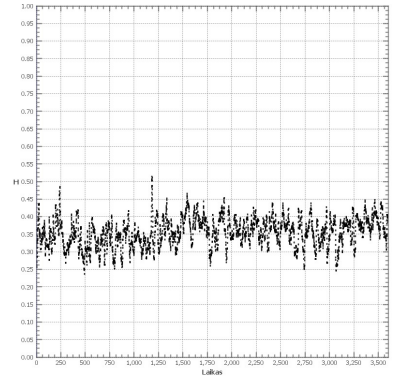
c $P = 10, I = 1000$



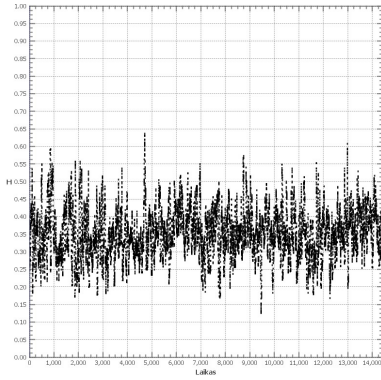
d $P = 20, I = 250$



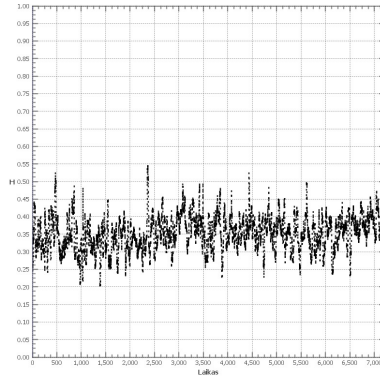
e $P = 20, I = 500$



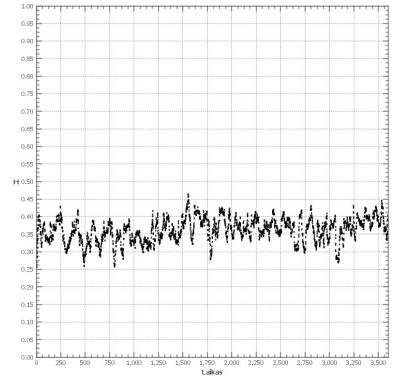
f $P = 20, I = 1000$



g $P = 30, I = 250$



h $P = 30, I = 500$



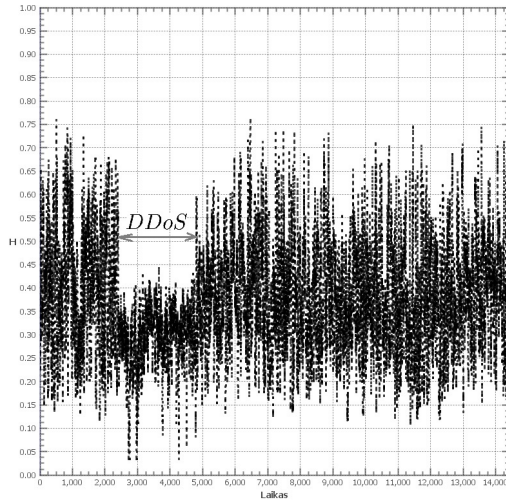
i $P = 30, I = 1000$

10 pav. Informacijos entropijos verčių priklausomybės nuo laiko su atitinkamais lango pločiais - P bei intervalo ilgiais - I .

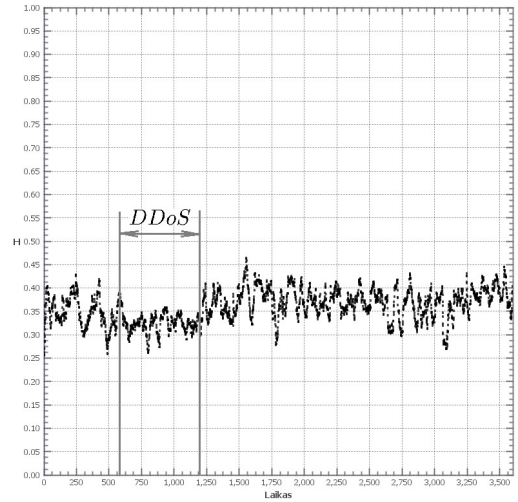
Tam, kad galėtume lyginti skirtingų parametrų daromą įtaką, kai yra norima aptikti DDoS atakas, sumodeliuosime DDoS atakas ir jas įterpsime prie darbo režime fiksuojamų paketų srauto. Simuliuojant DDoS atakas yra naudojamas šimtas skirtingų šaltinio adresų, kurių kiekvienas siunčia tam tikrą skaičių paketų per nustatytą laiko periodą į nurodytą paskirties IP adresą. Žinoma šios atakos yra gana paprastos ir realiais atvejais funkcijos generuojančios paketų srautus gali būti daug sudėtingesnės, tačiau naudojant paprastas atakas galime puikiai parodyti skirtumus tarp informacijos entropijos verčių, kai yra naudojami skirtingi parametrai jai apskaičiuoti. Yra simuliuojamos keturios skirtingo galingumo DDoS atakos, jos skiriasi tik paketų siunčiamų į paskirties adresą skaičiumi (pirma ataka siunčia po 1 paketą per 50 milisekundžių, antra du, trečia tris ir ketvirta keturis).

Iš 10 a pav. matome, kad lango pločiui bei intervalo ilgiui esant mažiems dydžiams apskaičiuotos informacijos entropijos vertės pasiskirsto dideliame intervale bei nuolat kinta nuo mažų iki didelių. Naudojant tokias lango pločio ir intervalo ilgio vertes pastebime, kad esant palyginus nedidelėms DDoS atakoms informacijos entropija nusileidžia žemiau mažiausios ribos. Tai galime matyti jeigu prie paketų srauto pridedame pirmą sumodeliuotos DDoS atakos srautą (11 a pav.). Matome, jog atsiranda atveju, kai būtų pranešama apie tinkle vykstančią anomaliją, bet pasitikėti pranešimu būtų sunku dėl tinkle vyraujančio informacijos entropijos nepastovumo. Tuo tarpu lango pločiui bei intervalo ilgiui esant pakankamai dideliems dydžiams (tokiems, kad kreivės triukšmas pakankamai sumažėja, kaip 10 i) apskaičiuotos informacijos entropijos vertės pasiskirsto mažesniame intervale. Taigi užfiksavę apskaičiuotos informacijos entropijos sumažėjimą, galėtume gana tiksliai sakyti, jog yra vykdoma DDoS ataka. Tačiau silpnų atakų atveju, DDoS atakos gali būti nepastebimos, tai yra matoma 11 b paveiksle. Taigi galime padaryti išvadą, jog mažų atakų atveju efektyviau yra naudoti mažesnes lango pločio bei intervalo ilgio vertes apskaičiuojant informacijos entropijos vertes.

Jeigu didinsime DDoS atakos siunčiamų paketų skaičių, informacijos entropijos vertės sumažės didesniu dydžiu. Informacijos entropijos vertes, naudojant skirtingo stiprumo DDoS atakas galime matyti 12 paveiksle. Matome, jog didėjant atakų aptikimo tikslumas irgi didėja nepriklausomai nuo to, kokie parametrai yra naudojami informacijos entropijai apskaičiuoti.



a $Nr_{DDoS} = 1, P = 10, I = 250$

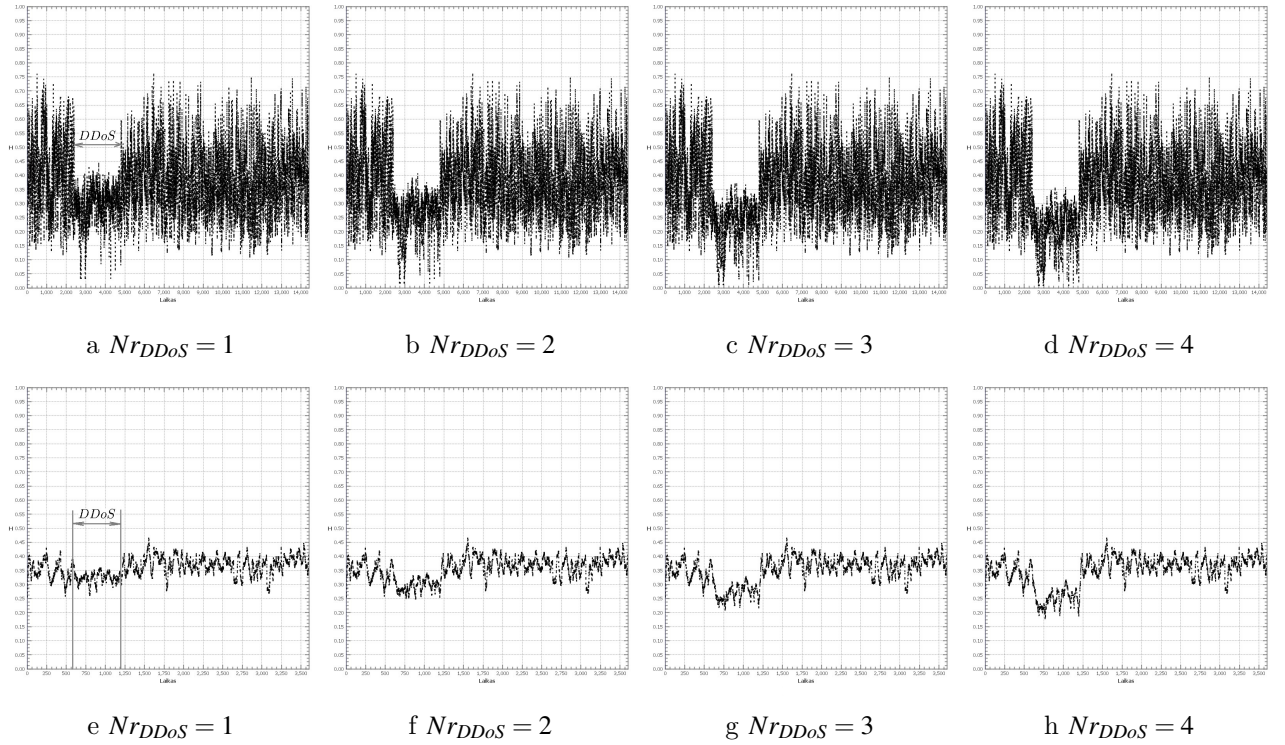


b $Nr_{DDoS} = 1, P = 30, I = 1000$

11 pav. Informacijos entropijos verčių priklausomybės nuo laiko pridėjus DDoS atakų paketų srautą. Paketų srautas yra pridamas tarp dešimos ir dvidešimos minučių, šį laiko intervalą atitinka 2400 – 4800 laiko skalės žingsniai 11 a pav. bei 600 – 1200 laiko skalės žingsniai 11 b pav. Prie grafikų yra pateikiamos parametrų (lango pločio P , intervalo ilgio I) vertės bei modeliuotų DDoS atakų numeriai (Nr_{DDoS})

Egzistuojant stiprioms atakoms, patogiau naudoti didesnes lango pločio bei intervalo ilgio vertes, nes išvengsime atvejų, kai bus pranešama apie ataką, kai ataka neegzistuoja. Tai galime matyti ir iš 12 h pav. - atakos metu informacijos entropija beveik visą laikotarpį turi tokias vertes, kurių nepasiekdavo nei prieš nei po atakos, kai tuo tarpu 12 d pav. atveju informacijos entropijos vertės tik dalyje DDoS atakos intervalo nusileidžia iki mažų (tiek, kad sistema registruotų tinklo anomaliją) informacijos entropijos verčių (žiūrėti 12 pav.).

Taigi galime padaryti išvadas, kad tyrinėjant DDoS atakas pasitelkiant informacijos entropijos sumažėjimu pagrįstais metodais, DDoS atakas aptikti galime parinkus teisingus informacijos entropijos apskaičiavimo parametrus. Deja atakos gali būti be galo skirtingos, dėl to informacijos entropijos parametrus turėtume parinkti pagal tai, kokio stiprumo DDoS atakas norime aptikti. Parinkus per mažas parametrų vertes, rizikuosime sulaukti didelio kiekio pranešimų apie anomalijas, kai atakos nevyksta, o parinkus per dideles parametrų vertes, mažesnės atakos liks neužfiksuotos.



12 pav. Informacijos entropijos verčių modeliuojant skirtingas DDoS atakas. Viršutinės eilės grafikuose (12 a, 12 b, 12 c, 12 d) parametrų vertės: $P = 10$, $I = 250$. Apatinės (12 e, 12 f, 12 g, 12 h): $P = 30$, $I = 1000$.

2.4 Laikinės priklausomybės prognozavimas

Kitas DDoS atakų aptikimo metodas, paremtas laikinės priklausomybės prognozavimu yra išbandomas. Šiame darbe yra prognozuojamos trys skirtingos laikinės priklausomybės. Dvi iš jų yra sumodeliuojamos (sinusoidė bei sinusoidė su triukšmu), o trečia yra realus paketų srautas. Ištirsime, kaip priklauso prognozuojamų verčių vertės nuo vertės prognozuoti pasirinkto dimensijų skaičiaus, optimalaus laiko poslinkio bei tiriamos laikinės priklausomybės savybių.

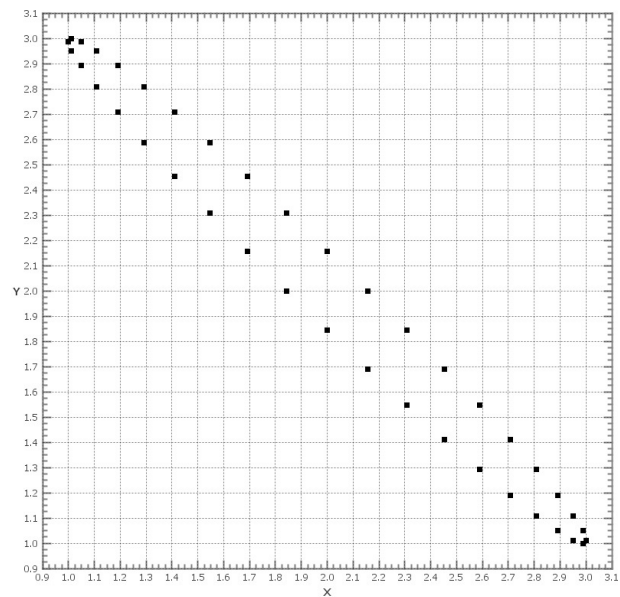
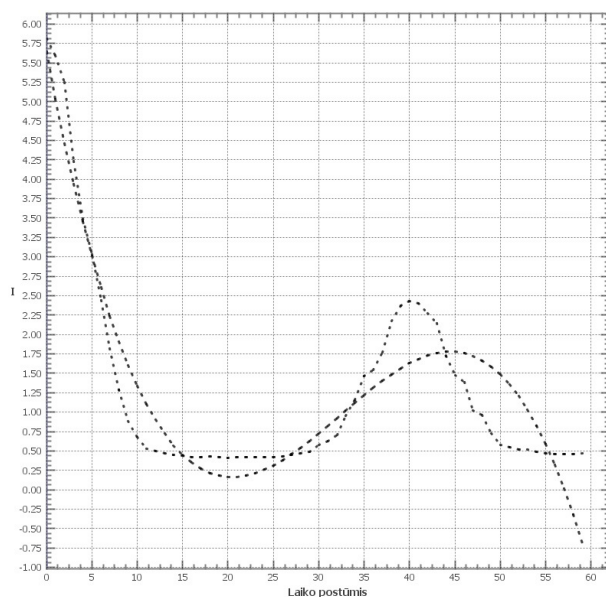
Sudarant realaus paketų srauto laikinę priklausomybę, ordinačių ašyje galime atidėti įvairias vertes. Gali būti atidedamos paketų (registruotų per tam tikrą laiką) skaičius, skirtingų paskirties (ar šaltinio) IP adresų skaičiai, apskaičiuotos informacijos entropijos vertės bei daugelis kitų apskaičiuojamų parametrų. Visi šie parametrai turi skirtingas prasmes. Naudojant paketų skaičiaus vertes, yra daroma prielaida, jog registruotų paketų skaičius tam tikrame laiko intervale turi deterministinį sąryšį su paketų skaičiumi kitame laiko intervale. Jeigu toks sąryšis egzistuoja, prognozuojamoms vertės stipriai išsiskiriant nuo realių verčių, galėsime teigti, kad

vyksta tinklo anomalija. Kita vertus, ši tinklo anomalija gali būti visiškai nesusijusi su DDoS ataka, nes nėra kreipiamas dėmesys nei į paketų šaltinio IP adresus, nei į paketų paskirties IP adresus (t. y. nežinoma ar yra atakuojama kuri nors tinklo dalis ar tiesiog pasikeitė tinklą naudojančių legalių vartotojų skaičius). Tas pats trūkumas išlieka tiriant skirtingų paskirties (ar šaltinio) IP adresų skaičius - negalime būti tikri, kad šių verčių pokyčiai yra susiję su DDoS atakomis. Tuo tarpu naudojant informacijos entropijos vertes, žinome, jog įvertiname daugelį skirtingų tinklo veiksmų, todėl informacijos entropijos verčių nukrypimai yra stipriai susiję su dėl DDoS atakų atsiradusiomis tinklo anomalijomis.

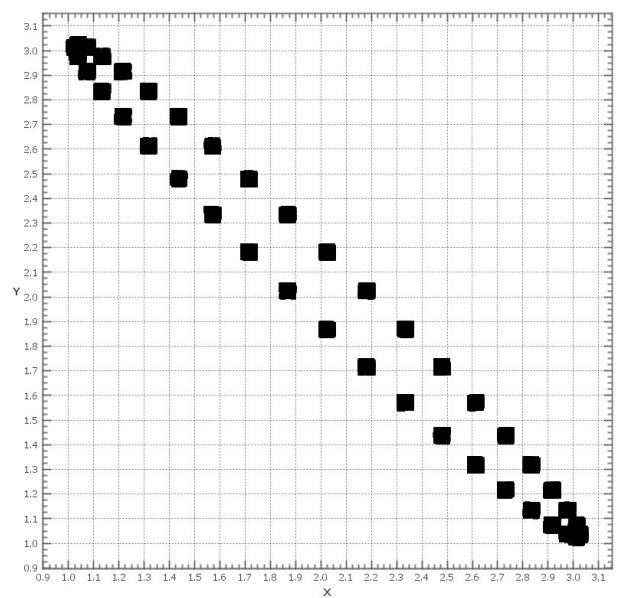
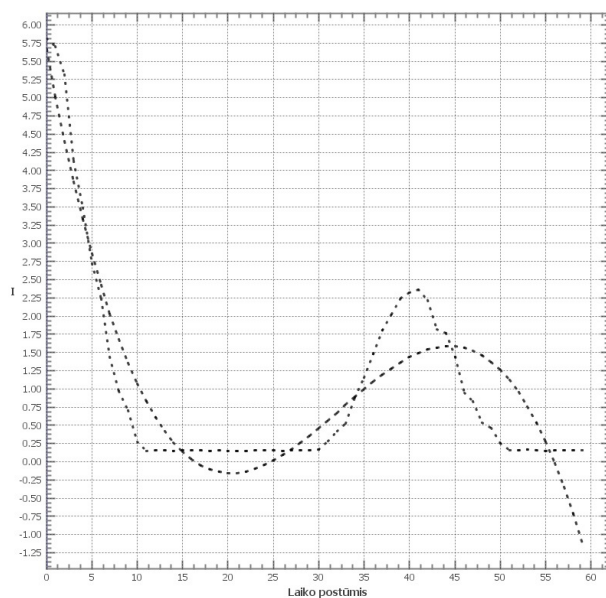
Prieš bandant prognozuoti realias laikines priklausomybes, ištirsime dvi teorines kreives. Prognozuojant laikines priklausomybes, turime rasti optimalų laiko poslinkį. Pirma apskaičiuosime laiko poslinkį remdamiesi bendros informacijos apskaičiavimu ir optimalia laiko poslinkio vertę laikysime pirmą bendros informacijos minimumo tašką. Kadangi bendros informacijos vertės priklauso nuo jos apskaičiavimo algoritmo bei fazinės erdvės koordinačių ašių skirstymo į segmentus kriterijų, optimalaus laiko poslinkio vertės nėra tiksliai nustatomas dydis. Varijuojant laiko poslinkį pabandydysime rasti optimalią vertę, kuri tikėtų laikinės priklausomybės prognozavimui.

Iš 13 pav. matome, kad apskaičiuojamas minimalus laiko poslinkis yra tada, kai sinusoidė yra paslenkama per pusę sinusoidės periodo. Matome, jog apskaičiuoto optimalaus laiko poslinkio vertės gaunamos tokios pat, nors absoliutinės bendros informacijos vertės skiriasi. Tai pat galime atkreipti dėmesį į tai, kad bendros informacijos vertė negrįžta į pradinę vertę, kai sinusoidė yra pastumiama per vieną periodą. Taip yra todėl, kad pastūmus vertes, sinusoidės vertės nežymiai pasikeičia, o apskaičiuojant bendrą informaciją labai svarbu į kokius segmentus vertės pakliūs. Sinusoidė nepasiskirsto fazinėje erdvėje tolygiai, dėl to nors ir nežymiai pakeitus vertes, fazinės erdvės taškai persiskirsto iš vieno segmentų į kitus.

Naudojant realius duomenis to neturėtų būti, nes realios laikinės priklausomybės vertės pasiskirsto visoje fazinėje erdvėje. Tai galime pamatyti iš 14 pav. Iš bendros informacijos priklausomybės nuo laiko postūmio grafikų, gautų tiriant realias informacijos entropijos kreives, matome, jog susidaro laiptuota kreivė. Kartais bendra informacija net vėl nežymiai pakyla,

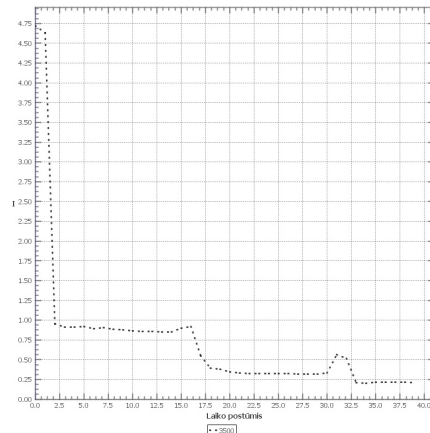


a Bendros informacijos priklausomybė nuo laiko b Sinusoidės, neturinčios triukšmo, fazinė erdvė, kai postūmio, kai tiriama neturinti triukšmo sinusoidė. laiko postūmis yra lygus 20 laiko vienetų.

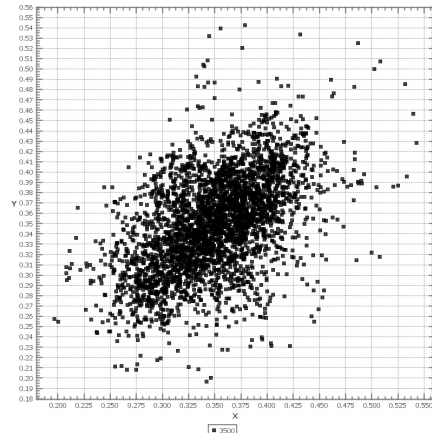


c Bendros informacijos priklausomybė nuo laiko d Sinusoidės, turinčios triukšmą, fazinė erdvė, kai laiko postūmio, kai tiriama turinti triukšmą sinusoidė. ko postūmis yra lygus 20 laiko vienetų

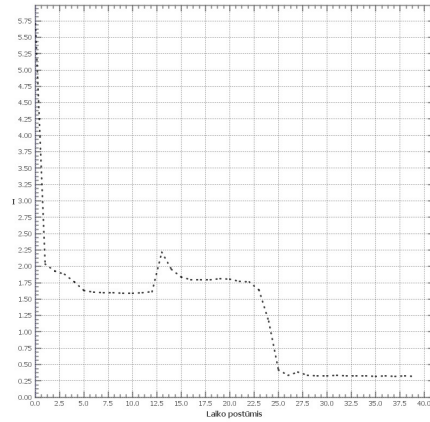
13 pav. Dviejų sinusoidžių (neturinčios triukšmo ir turinčios triukšmą) bendrą informaciją priklausomybės nuo laiko postūmio bei fazinės erdvės, gautos naudojant apskaičiuotą optimalų laiko poslinkį. Abiejų sinusoidžių periodai yra lygūs 40 laiko vienetų.



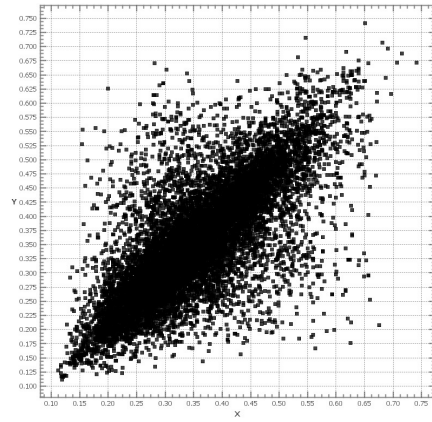
a $I = 1000, P = 15$



b $I = 1000, P = 15$



c $I = 250, P = 15$



d $I = 250, P = 15$

14 pav. Bendros informacijos priklausomybės nuo laiko postūmio bei fazinės erdvės apskaičiuotos naudojant optimalių laiko poslinkių kreivės, kai yra tiriama realių informacijos entropijos verčių laikinė priklausomybė. Parametrai informacijos entropijos vertėms apskaičiuoti yra pateikiami po grafikais.

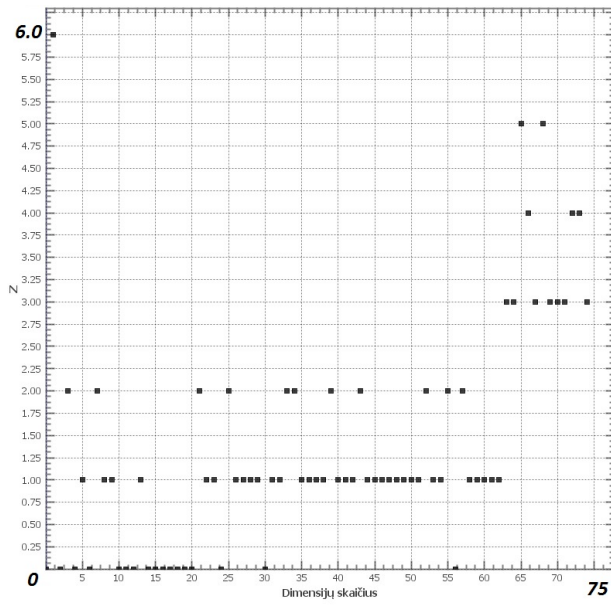
iš ko galėtumėm spręsti, jog egzistuoja deterministinis sąryšis tarp tam tikrų apskaičiuotos informacijos verčių. Tačiau kartu galime sakyti, jog šis sąryšis yra labai silpnas, nes palyginus su sinusoidžių bendros informacijos priklausomybės, vertės pakyla labai mažai. Galime atkreipti dėmesį ir į tai, kad laiptelio ilgis yra didesnis už informacijos entropijai apskaičiuoti naudojamą lango plotį, tai rodo, kad vertės yra deterministinės ne vien dėl to, kad apskaičiuojant entropijos vertes yra naudojami ir anksčiau registruoti paketai.

Realių informacijos entropijų verčių kreivėse optimaliu lango pločiu laikysime laiko postūmį tarp $\tau_{opt} = 15$ bei $\tau_{opt} = 25$. Tikslios vertės negalime pasakyti, nes nėra aiškaus minimumo taško. Tačiau savaime suprantama, jog tikėtis aiškaus minimumo taško būtų gana naivu, žinant kaip

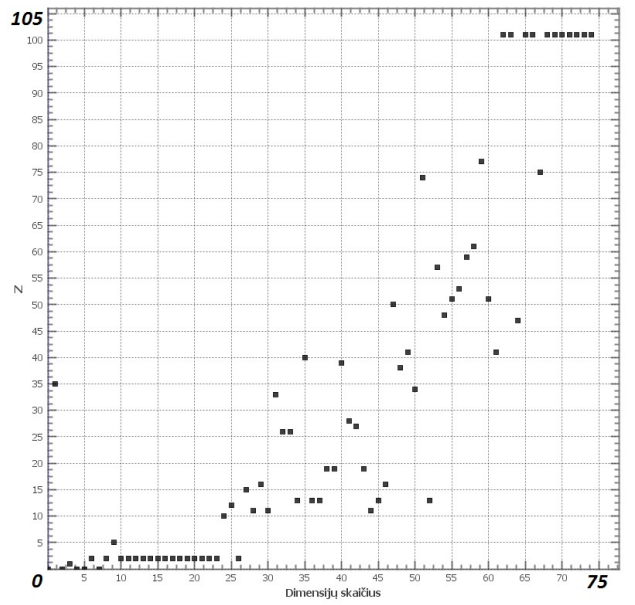
sunkiai aprašoma yra realių paketų informacijos entropijos laikinė priklausomybė.

Prognozuojant sinusoidės, neturinčios triukšmo, vertes, teisingai prognozuojamų taškų skaičius priklauso nuo sinusoidės periodo, dimensių (naudojamų prognozuojant vertes) skaičiaus bei naudojamo optimalaus laiko poslinkio. Priklausomybė nuo sinusoidės periodo, parodo tik tai, kaip elgiasi prognozavimo algoritmas, kai atstumai tarp gretimų taškų, laiko skalėje, sumažėja ar padidėja. Prognozuojant sinusoidę su didesniu periodu ($T = 400$) (15 b pav.) yra sugebama teisingai prognozuoti daugiau taškų, negu prognozuojant mažesnio periodo sinusoidę ($T = 40$) (15 a pav.). Taip yra dėl to, kad sutrumpėjus sinusoidės periodui, verčių skirtumas tarp gretimų sinusoidės taškų padidėja. Esant didesniai skirtumui tarp gretimų taškų, maži periodinių verčių netikslumai daro gana didelę įtaką prognozavimo rezultatui.

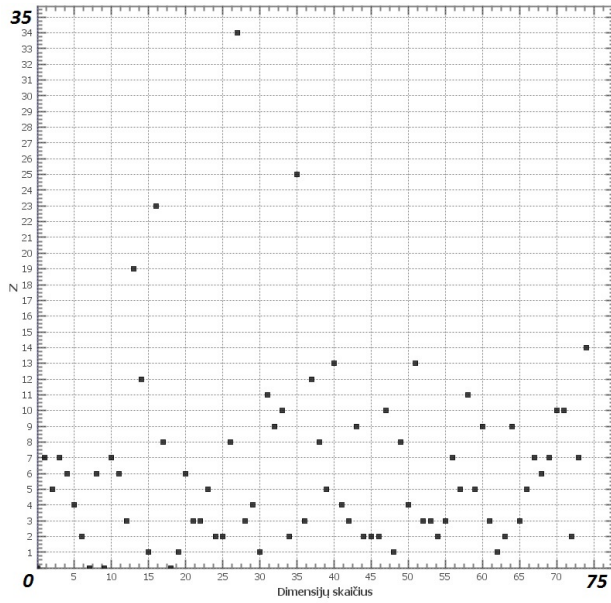
15 pav. teisingai nuspėjamų taškų skaičiaus grafikai yra tik tų optimalių laiko poslinkių, kai nuspėjamų verčių skaičiai yra didžiausi (išskyrus 15 d pav., nes visų verčių patikrinti negalime, taigi pateiktas grafikas su $\tau_{opt} = 25$). Kai tiriamo sinusoidę su periodu $T = 40$ (15 a pav.) teisingai nuspėti sugebame tik penkias vertes, naudojant kitus laiko poslinkius, nepavyksta nuspėti net tiek verčių. Tuo tarpu, tiriant didesnio periodo $T = 400$ sinusoidę, nuspėjamų verčių skaičius žymiai padidėja (sugebame nuspėti net iki $N = 102$ verčių). Taip pat, galime pastebėti tendenciją, jog didinant dimensių skaičių, teisingai nuspėjamų verčių skaičius auga. Šiuo atveju teisingai apskaičiuotų verčių skaičių riboja ne tai, kad prognozuojamos vertės nukrypsta nuo realių, o tai, kad negalime invertuoti B matricos, t. y. apskaičiuoti B^{-1} matricos, kurios reikia, kad galėtume rasti koeficientų matricą A . Jeigu nesugebame rasti koeficientų matricos A , tai negalime prognozuoti tolesnių verčių, t. y. negalime rasti $x(n+1)$ (iš 25 lygties). Pridėjus prie sinusoidės triukšmą (15 c pav.) teisingai prognozuojamų verčių skaičius žymiai sumažėjo, pavyksta prognozuoti iki $N = 34$ ateinančių verčių, iki tol, kol prognozuojamos vertės nukrypsta nuo realių verčių. Taip pat, su daugeliu dimensių pavyksta nuspėti iki $N = 11$ verčių. Šiuo atveju, visada pavyksta apskaičiuoti koeficientų matricą A , bet apskaičiuojamos sekančios vertės būna per toli nuo realių verčių. Kai tiriamo realių informacijos verčių kreivę, nuspėjamų verčių skaičiai siekia $N = 17$ verčių (informacijos entropija buvo apskaičiuojama naudojant lango plotį $P = 15$ bei intervalo ilgį $I = 250$ milisekundžių).



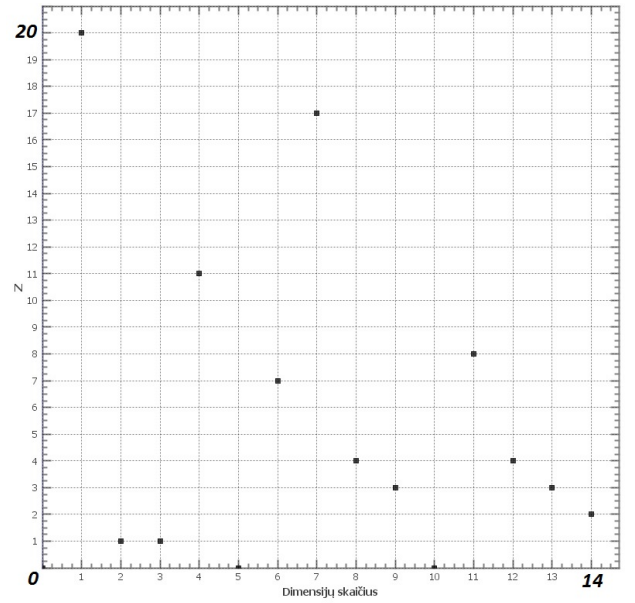
a Tiriama sinusoidė su periodu $T = 40$, $\tau_{opt} = 40$



b Tiriama sinusoidė su periodu $T = 400$, $\tau_{opt} = 400$



c Tiriama turinti triukšmą sinusoidė, kurios periodas d Tiriamos realios informacijos entropijos vertės, $T = 400$, $\tau_{opt} = 200$



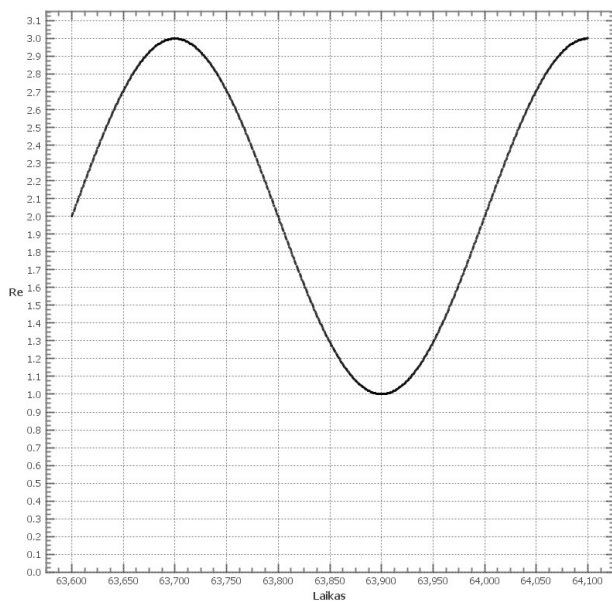
$\tau_{opt} = 25$

15 pav. Teisingai nuspėjamų taškų skaičiaus priklausomybės nuo dimensijų, naudojamų prognozuojant taškus, skaičiaus grafikai. Pateikiamos skirtingų prognozuojamų kreivių rezultatai: 15 a pav. trumpo periodo sinusoidė, 15 b pav. ilgo ($T = 400$ laiko vienetų) periodo sinusoidė, 15 c pav. yra tiriama sinusoidė su pridėtu triukšmu, 15 d pav. tiriama realių informacijos entropijos verčių kreivė.

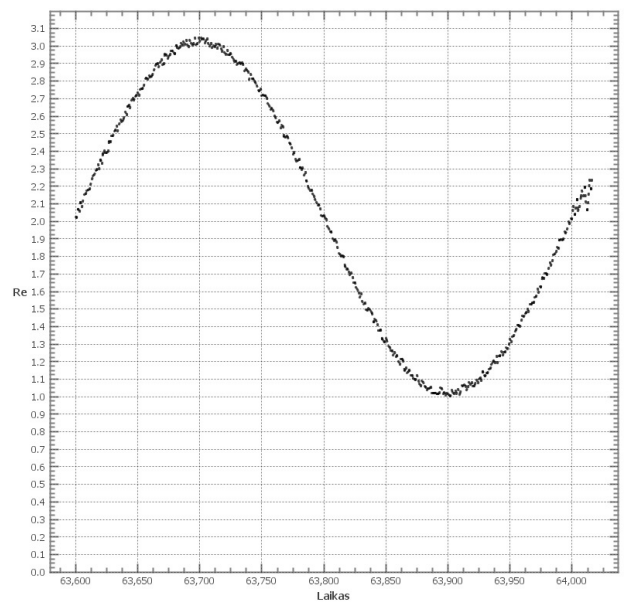
16 pav. parodyta, kaip atrodo prognozuojamos vertės iki tol, kol vertės yra prognozuojamos teisingai. 16 a pav. atvaizduotos sinusoidės neturinčios triukšmo realios bei prognozuojamos (nuo 64000-ojo laiko skalės žingsnio) vertės. Matome, kad vertės nenukrypsta, bet tiesiog matrica B tampa neinvertuojama. Tuo tarpu, kai tiriamo sinusoidę, kuri turi triukšmą, prognozuojamos vertės staiga nukrypsta nuo sinusoidės pavidalo. Teisingai sugebame prognozuoti daug mažiau verčių, kai yra tiriama sinusoidė, turinti triukšmą, nei sinusoidė be triukšmo.

Tuo tarpu, kai tiriamo realių informacijos entropijos verčių laikinę priklausomybę 17 pav. matome, jog apskaičiuotos prognozuojamos vertės labai greitai nukrypsta nuo realių verčių. Šiuo atveju kriterijus, kuris nutraukia prognozuojamų verčių apskaičiavimą yra verčių neatitiktumas. Šis kriterijus nėra visiškai tikslus, ir nutraukia sekančių verčių prognozavimą tik prognozuojamoms vertėms ryškiai nukrypus nuo realių. Taigi teisingai apskaičiuotų prognozuotų verčių skaičius yra gerokai mažesnis, nei matomas 15 d. Priklausomai nuo kreivės pobūdžio, pavyksta prognozuoti iki 3 realių kreivę atitinkančių taškų. Tai atitiktų laikinį intervalą 750 milisekundžių, kas yra labai trumpas laiko tarpas, kai informacijos entropijos vertės apskaičiuojame su intervalo ilgio $I = 250$ milisekundžių.

Iš to, kad nepavyko rasti tokio optimalaus laiko poslinkio, bei dimensių skaičiaus, kad prognozuojamų verčių skaičius būtų panaudojamas praktikoje DDoS atakoms aptikti, galime spręsti, jog naudoti laikinės priklausomybės prognozavimą DDoS atakų aptikimui yra neefektyvu.

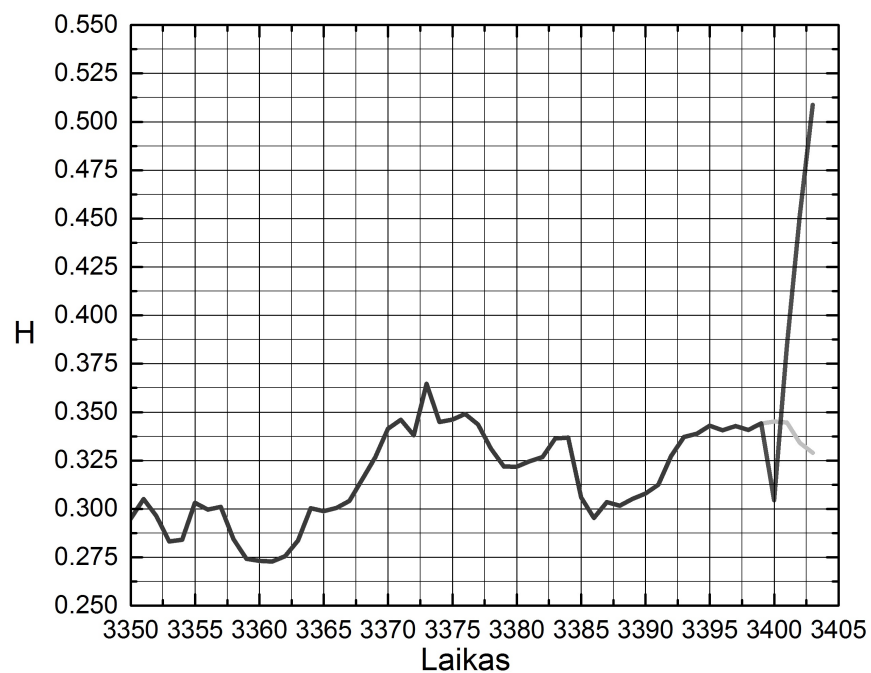


a $\tau_{opt} = 400$



b $\tau_{opt} = 400$

16 pav. Prognozuojamos vertės, kai naudojamos dvi periodo $T = 400$ sinusoidės. 16 a pav. sinusoidė neturi triukšmo, o 16 b sinusoidė turi nedidelį triukšmą. Abiem atvejais prognozuojamos vertės prasideda nuo 64000-ojo laiko skalės žingsnio.



17 pav. Prognozuojamos vertės, kai yra naudojama reali informacijos entropijos verčių kreivė. Vertės pradedamos prognozuoti nuo 3400-os vertės. Pilkai yra žymimos sekančios - realios vertės, o juodai - prognozuojamos vertės.

3 Rezultatai ir jų aptarimas

Iš 10 pav. matome kaip priklauso apskaičiuojamos informacijos entropijos vertės nuo pasirinktų apskaičiavimo parametrų. Parinkus didesnes lango pločio P bei intervalo ilgio I vertes, informacijos entropijos vertės pasiskirsto mažesniame intervale. Didinant lango pločio vertę P informacijos entropijos kreivė glaudėja, didinant intervalo ilgį I gali didėti atstumai tarp gretimų laikinės priklausomybės taškų, nes apskaičiuojant vertes kas $I = 1000$ milisekundžių (1 sekundę) galime tikėtis didesnių registruojamų paketų skaičiaus tinkle pasikeitimų, nei apskaičiuojant informacijos entropiją, kas $I = 250$ milisekundžių.

Prie realaus paketų srauto pridėjus modeliuotą DDoS ataką, informacijos entropijos vertės sumažėja. Nuo pasirinktų apskaičiavimo parametrų I bei P priklauso, ar pastebėsime, kad vyksta ataka, ar ne. Tai pavaizduota 11 pav., jeigu apskaičiuojame informacijos entropijos vertes naudodami lango pločio $P = 30$ bei intervalo ilgio $I = 1000$ vertes, nesugebame užfiksuoti DDoS atakos. Jeigu parenkame mažesnes šių parametrų vertes, DDoS ataką užfiksuojame, bet esant nepastoviam srautui, rizikuojame, jog pranešime apie anomalijas tada, kai DDoS atakų nevyksta. Iš 12 pav. galime suprasti, kaip informacijos entropijos vertės priklauso nuo modeliuotos DDoS atakos stiprumo. Matome, jog sustiprinus DDoS ataką, optimalūs parametrai informacijos entropijos apskaičiavimui tampa $P = 30$, $I = 1000$.

Apskaičiavus optimalų lango poslinkį fazinės erdvės atvaizdavimui (13, 14 pav.) bei iš geriausių teisingai nuspėjamų taškų skaičių grafikų (15 pav.) matome, kad optimalus laiko poslinkis fazinėms erdvėms apskaičiuoti nebūtinai sutampa su optimaliu laiko poslinkiu vertės prognozuoti. Prognozuojamų verčių skaičiai išauga, kai atstumai tarp gretimų taškų sumažėja (ilgesnio periodo sinusoidę sugebame prognozuoti tiksliau, nei trumpo periodo). Dėl to galėtume spręsti, jog sutankinus duomenis, turėtume pasiekti geresnius prognozavimo rezultatus, kai yra naudojami realūs duomenys. Tačiau net sugretinus kelių dienų duomenis, nebuvo pasiektas prognozuojamų taškų skaičius, kurį galėtume naudoti praktikoje DDoS atakoms aptikti.

Prognozuojant sekančias vertes, kai yra naudojami realūs duomenys (17 pav.) matome, kad prognozuojamos vertės iškart nukrypsta nuo realių verčių. Iš to, kad nepavyko rasti tokių optimalių laiko poslinkio verčių ar dimensijų skaičiaus, kuriuos naudojant ryškiai padidėtų

prognozuojamų verčių skaičius, galime spręsti, jog laikinė priklausomybė susidedanti iš realių, apskaičiuotų informacijos entropijos verčių, nepasižymi aiškiais deterministiniais sąryšiais. Kadangi nesugebėjome rasti skirtingų intervalų deterministinio sąryšio, galime teigti, jog laikinės priklausomybės prognozavimu paremti metodai nėra efektyvūs DDoS atakoms aptikti.

Pagrindiniai rezultatai ir išvados

Darbo metu išanalizuota DDoS atakų efektyvumo metodika paremta statistiniais metodais.

Sukurtas programavimo paketas DDoS atakų nustatymo tyrimui atlikti.

1. Tiriant sugretintus realius duomenis, įsitikinome, kad galime aptikti DDoS atakas informacijos entropijos sumažėjimu pagrįstais metodais. Atakos aptikimui reikia parinkti teisingus informacijos entropijos apskaičiavimo parametrus (t. y. lango pločio bei intervalo ilgio vertės turi būti pritaikytos prie norimų aptikti atakų stiprumo).
2. Naudojant laikinės priklausomybės prognozavimu paremtus metodus, sugebėjome teisingai prognozuoti iki 100 verčių (kai yra naudojama periodinė kreivė).
3. Tiriant realių duomenų (susidedančių iš informacijos entropijos verčių) laikinę priklausomybę, nustatyta, kad prognozuojamos vertės nesutampa su realiomis, nepriklausomai nuo to, koks laiko poslinkis bei kiek dimensių buvo naudojama prognozuojant ateinančias vertes. To priežastis - aiškaus deterministinio sąryšio tarp skirtingų laiko periodų nebuvimas, taigi naudoti laikinės priklausomybės prognozavimu pagrįstus metodus DDoS atakoms aptikti nėra efektyvu.

Literatūra

- [1] C. Jin, H. Wang, K. G. Shin, Hop-count filtering: an effective defense against spoofed ddos traffic, in *Proceedings of the 10th ACM conference on Computer and communications security* (ACM, 2003), 30–41.
- [2] Y. Chen, K. Hwang, Y. K. Kwok, Collaborative defense against periodic shrew ddos attacks in frequency domain, *ACM Trans. on Information and System Security (TISSEC)* (2005).
- [3] L. Feinstein, D. Schnackenberg, R. Balupari, D. Kindred, Statistical approaches to ddos attack detection and response, in *DARPA Information Survivability Conference and Exposition, 2003. Proceedings* (IEEE, 2003), volume 1, 303–314.
- [4] S. Yu, W. Zhou, R. Doss, Information theory based detection against network behavior mimicking ddos attacks, *Communications Letters, IEEE* **12**(4), 318–321 (2008).
- [5] Q. Meng, Y. Peng, A new local linear prediction model for chaotic time series, *Physics Letters A* **370**(5), 465–470 (2007).
- [6] A. M. Fraser, H. L. Swinney, Independent coordinates for strange attractors from mutual information, *Physical review A* **33**(2), 1134 (1986).
- [7] A.-H. Jiang, X.-C. Huang, Z.-H. Zhang, J. Li, Z.-Y. Zhang, H.-X. Hua, Mutual information algorithms, *Mechanical Systems and Signal Processing* **24**(8), 2947–2960 (2010).
- [8] <https://www.ll.mit.edu/ideval/data/1999/training/week1/monday/inside.tcpdump.gz>, (2016).
- [9] <https://www.ll.mit.edu/ideval/data/1999/training/week1/tuesday/inside.tcpdump.gz>, (2016).
- [10] <https://www.ll.mit.edu/ideval/data/1999/training/week1/wednesday/inside.tcpdump.gz>, (2016).

DDOS ATTACK DETECTION, USING STATISTICAL METHODS,
EFFICIENCY RESEARCH

Summary

Every now and then we hear in the news that DDoS attack has been used against one of the companies. Because of these attacks, companies lose customers and profit, that could be made in the time, when particular website or server, was down. There isn't much needed to create a DDoS attack, so it's much easier to create a DDoS attack, than defend against them. First line of defense against DDoS attacks is their detection, in this research we are using two different statistical methods to detect DDoS attacks.

First method is based on calculation of information entropy. We calculate the limit of minimal information entropy, when network is working normally (we are sure, that there are no on going attacks). Then, using this limit, we detect network anomalies, when calculated information entropy becomes lower than the limit. After the detection of an anomaly, we can take other actions to prevent packets, sent from infected computers, from reaching the application server.

Second method is based on prediction of time series. Once again we use network, which we are sure has no ongoing attacks. Then we check, if we are able to predict correct future values of the current time series. If predictions match current values, we can detect network anomalies, when DDoS attacks are introduced and real values do not match predicted ones. To increase the probability that anomalies happened because of DDoS attacks, we are going to use information entropy as range values of our time series.

The aim - to research methods based on decrease of information entropy and time series prediction for DDoS detection.

The tasks - show how information entropy changes, when DDoS attacks are introduced; show that we able to predict periodic functions; check if real information entropy value intervals have deterministic relationship.

Results of the first method showed that DDoS attacks can be detected, when we choose right parameters for the calculation of information entropy. Parameters should be chosen according to the strengths of attacks that we want to detect. The smaller the attacks whom we aim to, the bigger is the risk, that we will get false positive detection.

When using prediction of time series, we showed, that we can predict periodic curves. Unfortunately, we could not find deterministic relationship between different intervals of real information entropy values. That means it is not efficient to use prediction of the time series to detect DDoS attacks.