Analiza napada na mrežu KDD'99 Classifier Learning Contest

Leonora Gašpar, Filip Kiršek

Sažetak—Sa rastom prisutnosti računalnih mreža postaje sve važnije detektirati maliciozne veze. Dajemo analizu nekoliko metoda strojnog učenja primjenjenih na taj problem, koristeći podatke iz KDD'99 Classifier Learning Contest. Međutim, tokom istraživanja, i pregledom dosadašnjih radova, smo uočili manjkavosti dotičnih skupova podataka, pa iste analiziramo i ističemo.

Ključne riječi—KDDCup99, sigurnost, intrusion detection

1 Uvod

Natjecanje KDD CUP '99, naslovljeno Intrusion Detection Learning, se sastojalo od izrade modela sposobnog razlikovati normalne veze od napada. MIT Lincoln labs su 1998. pripremili i vodili DARPA Intrusion Detection Evaluation Program. Cilj programa je bilo analizirati i evalurati istraživanje u detekciji napada. Sa tim ciljem su pripremili skup podataka kojim bi ocjenili uspješnost. KDD CUP je koristio varijantu tog skupa podataka, prilagođenu njihovim potrebama. Rezultati natjecanja su dostupni, kao i neke dodatne informacije o samom natjecanju[2].
Cilj rada je ispitati kako bi se neke od osnovnih metoda strojnog učenja plasirale u tom natjecanju koristeći isto bodovanje, te pokušati nadmašiti pobjednike.
Međutim, mnogi algoritmi su se pokazali iznimno dobri na ovom skupu podataka, te se brzo uočavaju neki njegovi nedostatci.

2 Opis natjecanja

Skupljanje podataka sa obične mreže je problematično pošto je nepoznato koje su veze normalne, a koje maliciozne, a još teže ih svrstati u neki od tipova napada. Zato je skup podataka izrađeni simulirajući devetotjedni rad LAN mreže tipične baze Zrakoplovstva SAD-a, te simulirajuću napade na nju. Training set je izgrađen u prvih sedam tjedana, te se sastoji od 5 milijuna zapisa. Test set je izgrađen u zadnja dva, te se sastoji od 2 milijuna zapisa.
Spoj je niz TCP paketa koji počinju i završavaju u dobro definiranom vremenu, sa podatcima koji putuju između izvorne i odredišne IP adrese po nekom definiranom protokolu. Veza može biti normalna, ili napad, gdje svaki napad pripada nekoj od točno određenih, nepreklapajućih, tipova. Svaki zapis veze se sastoji od oko 100 bajtova.
Svaki tip napada se može svrstati u jednu od četiri glavne kategorije:

- DOS
- R2L
- U2R
- probing

Kako bi povećali vjerodostojnost podataka, training set i test set nemaju posve iste distribucije ove četiri kategorije. Cilj je simulirati razvoj novih vrsta napada s vremenom, pa se zato određeni tipovi napada pojavljuju isključivo u skupu za testiranje. Neki stručnjaci za sigurnost vjeruju da je većina novih napada samo varijanta prethodnih, pa je za njihovo prepoznavanje dovoljno dobro naučiti njih same. Skup za učenje sadrži 24 vrste napada, sa dodatnih 14 sadržanih isključivo u skupu za testiranje.

2.0.1 Opis podataka

Korištenjem prethodnih istraživanja, kao i znanja o vrstama napada iz osnovnih varijabli o vezi, poput trajanja, protokola i sličnog, su izvedene dodatne varijable. Popis svih varijabli sa njihovim opisima se nalazi u opisu natjecanja [1]. Ovdje navodimo njegovu podijelu i imena varijabli:

- Osnovni podatci individualnih TCP veza - trajanje, protokol, servis, duljina poruka, sl.
- Podatci o sadržaju veze zasnovani na znanju o domeni - broj neuspješnih *login* pokušaja, *root* pristup, broj pristupa datotekama, sl.
- Podatci o sličnom prometu - broj veza, broj veza sa SYN greškom, postotak neuspjelih veza, sl.

Prva grupa su varijable dobivene izravno iz simulacije te daju osnovne informacije o vezi. Nisu sadržane apsolutno sve informacije koje su dostupne, neke su ocijenjene nevažnima za detekciju napada, dok su druge agregirane u varijable drugih grupa.

Napadi tipa DOS i *probe* se oslanjaju na korištenju više uzastopnih veza. Kako bi se takvi napadi mogli detektirati, potrebno je uočiti sličnosti među većim brojem spajanja. Iako pojedinačno spajanje možda nije sumnjivo ili opasno, informacija da ih je bilo više unutar kraćeg vremenskog perioda indicira napad koji možda pripada jednoj od tih kategorija. Napadi tipa DOS se često oslanjaju na jednostavnom preopterećenju mreže velikim brojem spajanja u vrlo kratkom vremenu. Treća grupa varijabli zato sadrži one izvedene promatrajući ostale veze uspostavljene unutar kratkog vremenskog perioda, primjerice `count` (broj veza unutar 2 sekunde), `srv_serror_rate`(postotak veza unutar 2 sekunde koje se spajaju na isti servis i vraćaju grešku SYN) i slično. .
S druge strane, napadi tipa *probe* često koriste više veza koje se događaju rjeđe, primjerice jednomu minuti, no sa drugim sličnim svojstvima. Zato se u trećoj grupi nalaze i varijable koje opisuju postotak veza sa sličnim svojstvima, a koje se spajaju sa istog poslužitelja, primjerice `rerror_rate` (postotak veza istog poslužitelja sa greškom REJ), `serror_rate` (slično kao prethodno, ali sa greškom SYN) i slične.

No, napadi tipa R2L i U2R se oslanjaju na sadržaj veze, a ne njihov broj. Stoga druga grupa sadrži varijable bazirane na sadržaju veza, primjerice `logged_in` (uključuje li veza korisnika koji se spojio), `root_shell` (ima li veza *root* pristup, `num_access_files` (broj datoteka kojima je veza pristupila), i slično. Lako je vidjeti kako takvi podatci mogu indicirati potencijalno opasne veze.

Konačno, svaka veza je označena kao normalna, ili kao jedna od 36 vrsta napada, od kojih se 14 pojavljuje isključivo u skupu za testiranje.

2.1 Dodatna obrada podataka

Zbog veličine skupova podataka, za ocjenu algoritama je korišten podskup od 10% skupa za testiranje, koji ima istu distribuciju. Slično tome, mnogi natjecatelji su koristili podskup skupa za učenje. Kako je opis natjecanja rekao da ta dva skupa nisu posve iste distribucije, mnogi natjecatelji su koristili manji skup za učenje sa različitom distribucijom.
U našim primjerima smo kao početnu točku koristili 10% podataka skupa za učenje te 10% podataka skupa za testiranje. Oba podskupa su bili iste distribucije kao i početni skupovi.
Već smo spomenuli da su veze u skupu za učenje klasificirane po vrstama napada. U prvim pokušajima smo koristili te oznake kao osnova u algoritmima, pa smo ih kasnije pridružili kategorijama kojima pripadaju te ocjenili. Naravno, pošto je nemoguće naučiti oznake koje ne postoje u skupu za učenje, ocjenjivati tako naučene klasifikatore ima smisla tek kada vrste napada pridružimo kategorijama.
Drugi pristup je pretvoriti vrste napada u njihove kategorije prije korištenja metoda učenja. Kod nekih algoritama je to dovelo do boljih rezultata, no kod nekih ne.

Distribucije kategorija u danim skupovima su dane u tablici 1.

2.2 Bodovanje

Bodovanje tijekom natjecanja je koristilo konfuzijske matrice izlaza. Različite vrste krive klasifikacije su penalizirane vrijednostima između

Tablica 1
Distribucije tipova napada

	training	test
norm.	19.69%	19.48%
probe	0.83%	1.34%
DOS	79.24%	73.90%
U2R	0.01%	0.07%
R2L	0.23%	5.20%

1 i 4, kao što je pokazano u tablici 2. Konačna ocjena je dobivena zbrajanjem svih penala, i dijeljenjem sa ukupnim brojem primjera u skupu za testiranje. Skup za testiranje se, kao što je već spomenuto, sastoji od 10% svih podataka namjenjenih za testiranje, te je iste distribucije kao početni skup.
Razlog korištenja ovog načina bodovanja je taj što su U2R i R2L kategorije napada rjeđe, no i opasnije, od ostalih, pa je puno veća pogreška njih kategorizirati kao normalnu vezu od manje opasnih DOS i *probe* napada.

Tablica 2
Matrica bodovanja krive klasifikacije

	normal	probe	DOS	U2R	R2L
normal	0	1	2	2	2
probe	1	0	2	2	2
DOS	2	1	0	2	2
U2R	3	2	2	0	2
R2L	4	2	2	2	0

3 METODE RJEŠAVANJA PROBLEMA

Od algoritama strojnog učenja koristili smo:

- 1–najbližih susjeda (dalje u tekstu `1-nn`)
- C4.5 algoritam za generiranje stabla odlučivanja (dalje u tekstu `C4.5`)
- AdaBoost meta-algoritam primjenjen na `C4.5` (dalje u tekstu `ADB`)

Pobjednik natjecanja je koristio C5 algoritam za generiranje stabala odlučivanja, uz ansambl konstruiran koristeći *cost-sensitive bagged boosting*. Iz tog razloga smo se odlučili usredotočiti na metode slične pobjedničkima. Organizatori natjecanja su također istaknuli vrlo dobar uspjeh `1-nn` algoritma, pa smo i njega dodali na popis. Svaki od algoritama smo isprobali u nekoliko varijanti, te na varijantama ulaznih skupova, kako je opisano u 2.1. Za sve navedene algoritme smo koristili gotove implementacije koje dolaze sa Weka 3.7.12, dostupnoj na [8]. Ocjenjivanje i kategoriziranje tipova napada je obavljeno awk skriptama korištenim i za natjecanje, a dostupnima na [2].
Detaljniji opisi korištenja, te parametri, su dostupni na javnom github repozitoriju [9].

4 REZULTATI

Konačna tablica 3 je dobivena spajanjem rezultata natjecanja i rezultata isprobanih algoritama.

Tablica 3
Rezultati

rang	cijena	algoritam	rang	cijena	algoritam
1	0.2331	C5 csbb	**16**	**0.2532**	**1-nn vf**
2	0.2331	-	17	0.2545	**-**
3	0.2367	-	18	0.2552	-
4	0.2411	-	19	0.2575	-
5	0.2414	-	20	0.2588	-
6	**0.2417**	**1-nn lf**	**21**	**0.2592**	**ADB C4.5 cat**
7	0.2443	-	22	0.2644	-
8	**0.2452**	**C4.5 kat**	23	0.2684	-
9	**0.2457**	**ADB C4.5**	24	0.2952	-
10	**0.2467**	**C4.5**	25	0.3344	**-**
11	0.2474	-	26	0.3767	-
12	0.2479	-	27	0.3854	-
13	0.2523	1-nn	28	0.3899	-
14	0.2530	-	29	0.5053	-
15	0.2531	-	30	0.9414	-

Vidljivo je da, iako nijedan od algoritama nije plasiran među prvih 5 mjesta, većina ih se nalazi vrlo visoko u plasmanu, uz vrlo male razlike među najboljim algoritmima. Možemo također primjetiti da su najlošiji rezultati natjecanja drastično lošiji od ostalih, a najlošiji je drastično lošiji i od klasifikatora koji sve svrstava u drugu kategoriju (čime bi rezultat bio 0.5220).
Navedimo prvo pobjedničku konfuzijsku matricu, sa kojom ćemo uspoređivati naše rezultate.

Dalje opisujemo pojedine varijante i dajemo dodatne podatke o njihovom plasmanu.

4.1 C4.5 algoritam

C4.5 algoritam smo učili na dvije varijante skupa za učenje, jednoj sa tipovima napada (označenoj C4.5, danoj u 5), te jednoj sa kategoriziranim tipovima napada (označenoj C4.5 kat,u

Tablica 4
Konfuzijska matrica pobjednika

T \ P	normal	probe	DOS	U2R	R2L	%
norm.	60262	243	78	4	6	99.5%
probe	511	3471	184	0	0	83.3%
DOS	5299	1328	223226	0	0	97.1%
U2R	168	20	0	30	10	13.2%
R2L	14527	294	0	8	1360	8.4%
%	74.6%	64.8%	99.9%	71.4%	98.8%	

6). Druga varijanta se pokazala nešto uspješnijom sveukupno, no prva varijanta je prepoznala ispravno veći postotak kategorije R2L.

Tablica 5
C4.5 konfuzijska matrica. Prosječna cijena: 0.2467

T \ P	normal	probe	DOS	U2R	R2L	%
norm.	60247	236	89	3	18	99.4%
probe	656	3283	227	0	0	78.8%
DOS	6095	1050	222703	0	5	96.9%
U2R	134	65	1	12	16	5.3%
R2L	15329	12	0	5	843	5.2%
%	73.1%	70.7%	99.9%	60.0%	95.6%	

Tablica 6
C4.5 na kategorijama, konfuzijska matrica. Prosječna cijena: 0.2452

T \ P	normal	probe	DOS	U2R	R2L	%
norm.	60287	218	74	2	12	99.5%
probe	887	3112	167	0	0	74.7%
DOS	6167	11	223674	0	1	97.3%
U2R	200	4	1	21	2	9.2%
R2L	15197	340	2	118	532	3.3%
%	72.9%	84.5%	99.9%	14.9%	97.3%	

4.2 AdaBoost C4.5

Kao i C4.5, i AdaBoost algoritam na C4.5 metodi smo testirali na obje varijante skupa za testiranje. Za razliku od prethodnih primjera, prva varijanta (u tablici 7),u kojoj su korišteni upravo tipovi napada, je pobošljala rezultate odgovarajućeg C4.5 rezultata, dok je druga varijanta (u tablici 8) bila znatno lošija od ostalih metoda, kao i od njoj odgovarajuće C4.5.

Tablica 7
AdaBoost C4.5 konfuzijska matrica. Prosječna cijena: 0.2457

T \ P	normal	probe	DOS	U2R	R2L	%
norm.	60302	210	75	3	3	99.5%
probe	646	3327	193	0	0	79.9%
DOS	5345	161	224063	0	284	97.5%
U2R	178	21	5	13	11	5.7%
R2L	15741	10	0	7	431	2.7%
%	73.3%	89.2%	99.9%	54.2%	59.0%	

Tablica 8
AdaBoost C4.5 na kategorijama, konfuzijska matrica. Prosječna cijena: 0.2592

T \ P	normal	probe	DOS	U2R	R2L	%
norm.	59110	575	692	22	194	97.6%
probe	1284	2734	103	1	44	65.6%
DOS	6751	161	222923	0	18	97.0%
U2R	205	0	18	3	2	1.3%
R2L	15396	256	77	24	436	2.7%
%	71.4%	73.4%	99.6%	6.0%	62.8%	

4.3 1–nn

Najbolji rezultat među našim pokušajima je imala jedna od 1–nn varijanti. Pošto je metoda najbližih susjeda izrazito spora na velikim skupovima za učenje, a naš skup za učenje je imao preko 400 tisuća zapisa, bilo je potrebno reducirati njegovu veličinu. Kako duplikati za 1–nn ne utječu na rezultat klasifikacije, prvi korak u redukciji veličinu skupa za učenje je micanje duplikata, čime je veličina reducirana za 75%. Dodatna redukcija je bila korištenjem

`SpreadSubset`

filtera. U tom filteru, odabire se nasumičan podskup originalnih podataka, uz definirani maksimalni omjer među klasama. U prvoj varijanti, prikazanoj u tablici 9, je taj parametar postavljen na 100, a u drugoj, tablica 10, na 20. Takvo filtriranje je proizvelo skup od 2379, tj. 606, zapisa, koji su onda dalje korišteni za klasificiranje testnog skupa.
Ovdje su navedeni rezultati filtriranja primjenjenog samo na skup označen tipovima napada, ne njihovim kategorijama, kao što je to bio slučaj u prethodnim metodama. Primjena na kategorizirani skup je bila nešto uspješnija od druge varijante, no i dalje daleko lošija od

ostalih metoda.

Tablica 9
1–nn sa manje filtriranim skupom za učenje, konfuzijska matrica. Prosječna cijena: 0.2417

T \ P	normal	probe	DOS	U2R	R2L	%
norm.	57484	1376	1382	166	185	94.9%
probe	271	3340	272	24	259	80.2%
DOS	5392	495	223295	61	610	97.1%
U2R	17	149	0	52	10	22.8%
R2L	13788	40	211	161	1989	12.3%
%	74.7%	61.8%	99.2%	11.2%	65.1%	

Tablica 10
1–nn sa više filtriranim skupom za učenje, konfuzijska matrica. Prosječna cijena: 0.2532

T \ P	normal	probe	DOS	U2R	R2L	%
norm.	55643	433	3981	321	215	91.8%
probe	218	3382	165	51	350	81.2%
DOS	5326	334	223295	61	617	97.2%
U2R	17	135	0	66	10	28.9%
R2L	13408	34	596	181	1970	12.2%
%	74.6%	78.3%	97.9%	9.7%	62.3%	

Iz rezultata je vidljivo da se prva varijanta filtriranja za 1–nn pokazala puno uspješnijom od ostalih metoda, te uspješnijom od 1–nn klasifikatora koji se natjecao. Razlog tome je, prema mišljenju autora, upravo korištenje skupa sa tipovima napada, a ne njihovim klasama. Naime, već bi osnovna ideja te filtracije drastično promjenila distribuciju. Međutim, pošto postoje mnogi izrazito rijetki napadi koji pripadaju rijetkim kategorijama, to je dovelo do dodatnog iskrivljenja odnosa među kategorijama. To je također vidljivo i iz konfuzijskih matrica; ova dva klasifikatorima su češće radili greške koje se ne pojavljuju u drugim metodama. Primjerice, svrstavanje DOS u kategoriju U2R je greška koju niti jedna metoda, osim te dvije, nije napravila. Također, iako su točno svrstali veći postotak rijetke U2R kategorije, te R2L kategorije (koja ima različite distribucije) nego pobjednički unos, zato su puno veći postotak krivo svrstali u te iste kategorije. Razlog je, naravno, potpuno iskrivljenje učestalosti tih kategorija i točaka.

5 MANJKAVOSTI SKUPOVA PODATAKA

Među glavnim kritikama na skup podataka bile su te da nije provedena ikakva provjera da DARPA dataset uistinu pruža prikaz realnog mrežnog prometa. Čak i na letimičan pogled bilo je jasno da su stope u podacima bile daleko ispod onih koje bi bile u pravoj mreži srednje veličine. Usprkos tome, istraživači IDS-a nastavili su koristiti taj dataset jer nisu imali bolju alternativu.

Konkretno, 2003. god izgrađen je trivijalan sustav za detekciju napada na mrežu i pokrenut na DARPA podacima. Pronađene su brojne nelogičnosti, uključujući tu da su zbog načina generiranja podataka sve maliciozne veze imale TTL 126 ili 253, dok su svi normalni imali TTL 127 ili 254. (TTL, Time-to-live, je vrijednost u IP paketu koji govori mrežnom routeru jesu li paketi bili predugo u mreži i trebaju li biti odbačeni). To je poslužilo kao dokaz da je DARPA skup podataka u suštini loš i da se ne bi trebali izvoditi zaključci iz njih.

Testni skup podataka sadrži 311 029 primjera, ali oni nisu svi nezavisni. Gornja granica na broj nezavisnih testnih primjera je broj jedinstvenih testnih primjera, koji je 77291. Drugim riječima, skoro 75% i training i test skupa su duplikati, što je još jedna loša strana skupa podataka.

Dok se DARPA (i KDD Cup '99) skup podataka ne koristi kod proučavanja mrežne sigurnosti, i dalje se koristi u široj zajednici strojnog učenja. Dakle, iako se na tom skupu podataka i dalje vrše istraživanja, zbog problema sa njim ne možemo doći do relevantnih zaključke. Preporučuje se izbjeći njegovo korištenje. [6]

6 ZAKLJUČAK

Razlika u performansi između 3 najbolja sa natjecanja je statistički marginalna. Teško je procjeniti statistički značaj razlika između rezultata. Međutim, važno je da ne dajemo preveliku važnost razlikama koje su statistički neznačajne, zbog primjerice nasumičnosti u izboru training i test primjera.

Statistički značaj može se evaluirati na sljedeći način: Srednja vrijednost pobjedničkog rezultata, mjereno na određenom testnom skupu

je 0.2331, sa izmjerenom standardnom devijacijom 0.8834. Ako pretpostavimo da je srednja vrijednost izračunata iz N neovisnih opservacija, njena standardna greška je $0.8334/\sqrt{N}$. Uzmemo li u obzir da su skoro 75% podataka duplikati(što smo već objasnili u sekciji 5), standardna greška srednje vrijednosti pobjednika tada je barem $0.8334/\sqrt{77291} = 0.0030$.

Ako postavimo granicu za statističku značajnost na dvije standardne greške, tada je pobjednički algoritam značajno bolji od sviju osim drugog i trećeg po poretku. Koristeći istu granicu je jasno da su 4 najbolje plasirane metode među navedenima statistički vrlo slične, no ipak pripadaju boljim rezultatima.

Prema izvještaju o rezultatima KDD '99 Classifier Learning Contest, većina sudionika postigla je rezultate u rangu onih koji se mogu postići sa jako jednostavnim metodama. Jedan od sudionika koristio je najobičniji standardni 1-nn algoritam. Samo je 9 sudionika imalo bolji rezultat od 1-nn, od kojih je samo 6 bilo statistički bolje. U usporedbi sa 1-nn, glavno postignuće pobjednika bilo je prepoznavanje puno više "remote-to-local" napada : 1360 u odnosu na 95.
Kao što smo mogli vidjeti, taj algoritam je bilo moguće dodatno poboljšati. Dapače, najbolje plasirani algoritam je upravo naša verzija 1–nn algoritma, te spada u statistički znatno bolje algoritme. Razlog njenog uspjeha je manjak pretpostavki na distribuciju podataka, što ju čini robustnijom na njenu promjenu, te "pogođena" distribucija pri filtraciji. Pobjednički algoritam [4] je do svog uspjeha došao kombiniranjem sva 3 pristupa koja su se ovdje pokazali uspješnima - što je bio i razlog našeg odabira. Naime, pobjednik je također posve promjenio distribucije podataka, te višestruko učio na skupu za učenje.

Međutim, skoro sve metode koje smo koristili su bile iznimno uspješne. Većina rezultata leži u jako malom rasponu, statistički odudaraju samo iznimno dobri i iznimno loši. Jedan od mogućih razloga leži u samom skupu za testiranje. Kao što smo već spomenuli, on sadrži vrlo velik broj duplikata, i to duplikata među najčešćim kategorijama, što znači da će klasifikator koji točno klasificira jedan od tih unosa isto klasificirati i sve njegove duplikate, kojih vrlo vjerojatno ima još.
Nadalje, kao što smo već opisivali u početnom dijelu, grupe varijabli su konstruirane upravo tako da bi postojala mogućnost detektiranja nekih od kategorija, što dovodi do povećane preciznosti.

Kao što smo spomenuli, koristili smo dvije varijante skupa za učenje, jednu sa svim tipovima napada, a drugu sa njihovim kategorijama. Dodatno poboljšanje bi se možda moglo dobiti tako da se kombiniraju oba pristupa, primjerice u nekoj od ansambl metoda. Također, kao što je jasno i iz ovdje najboljeg, i iz pobjedničkog, veća preciznost se može dobiti dodatnim namještanjem distribucije skupa za učenje. Međutim, zbog problema sa samim podatcima, upitno je koliko bi bilo moguće detektirati ta poboljšanja.

Jedan od algoritama koji je razvije na temelju ovog skupa podataka je i PNrule [5]. Autori tvrde da se on pokazao iznimno dobar na ovim skupovima podataka. Međutim, zbog manjka dostupne implementacije, te teško dostupnog rada sa punim opisom algoritma, smo bili primorani odustati od toga.

LITERATURA

[1] KDD-CUP-99 Task Description
http://kdd.ics.uci.edu/databases/kddcup99/task.html

[2] Results of the KDD'99 Classifier Learning Contest
http://cseweb.ucsd.edu/~elkan/clresults.html

[3] KDD-CUP Data
http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[4] B. Pfahringer, Winning the KDD99 Classification Cup: Bagged Boosting
http://web.archive.org/web/20000120022008/http://www.ai.univie.ac.at/~bernhard/kddcup99.html

[5] R. Agarwal, M. V. Joshi, *PNrule: A New Framework for Learning Classifier Models in Data Mining (A Case-Study in Network Intrusion Detection)*
http://www.siam.org/meetings/sdm01/pdf/sdm01_30.pdf

[6] KDD CUP '99 dataset (Network Intrusion) considered harmful
http://www.kdnuggets.com/news/2007/n18/4i.html

[7] J. McHugh, *Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory*
https://www.cs.nmt.edu/~infosec/Critique%20on%20Testing%20IDS.pdf

[8] Weka 3: Data Mining Software in Java
http://www.cs.waikato.ac.nz/ml/weka/

[9] https://github.com/fkirsek/su-mreza