Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva
Zagreb

Toni Gržinić

Analiza dnevnika pristupa web poslužitelja metodama dubinske analize podataka

Pristupni rad iz kolegija
Otkrivanje znanja u skupovima podataka

Voditelj rada:
prof.dr.sc. Nikola Bogunović

Zagreb, 2013.

Sadržaj

1 Uvod

Pogledamo li izvještaje o otvorenim portovima na poslužiteljskim računalima koji su dostupni Internetu vidjet ćemo da vrlo često prednjače TCP portovi 80 i 443 [21]. Ovi portovi se najčešće povezuju s protokolima HTTP (Hypertext Transfer Protocol) i HTTPS (Hypertext Transfer Protocol Secure), a namjena im je opsluživanje web stranica putem web poslužitelja poput: Apache web servera, Microsoft Internet Information Services, ngix i drugih.

Prema Netcraftovom izvještaju [18] iz kolovoza 2013., Apache web server posjeduje dominanti udjel od 52.19% od ukupnog broja web poslužitelja koji opslužuju aktivna web sjedišta na Internetu.

Danas su na Internetu, putem gore navedenih web poslužitelja, izloženi i osjetljivi servisi poput: Internet bankarstva, praćenja zdravlja kroničnih bolesnika kroz različite bolničke informacijske sustave, države nude različite servise svojim građanima (npr. zemljišne knjige, usluge matičnih ureda, birački sustavi i sl.).

Važno je napomenuti da neke poslovne organizacije svoje poslovanje temelje samo na Internet servisima. Iz toga razloga sigurnost takvih sustava i uspješno prepoznavanje pokušaja napada i/ili kompromitacije se u današnje vrijeme postavlja kao izazov ne samo pred poslovne organizacije nego isto tako i pred znanstvenike i istraživače koji se bave informacijskom sigurnošću. Također, sve više zabrinjavaju napadi na takve sustave od strane profesionalnih cyber-kriminalaca ili samoproglašenih aktivista. Ovi napadi koriste ažurne metode koje se konstantno mijenjaju i prilagođavaju u svrhu skrivanja od sustava koji omogućuju njihovu detekciju.

U ovom radu bit će riječi o detekciji napada na sustav sa strane analize podataka koji sam sustav generira. Naime, web poslužitelji imaju mogućnost spremanja dnevnika pristupa (log zapisa) prilikom pristupa korisnika sustavu. Rad će prikazati izradu modela za detekciju anomalnog ponašanja korisnika koji pristupaju web poslužitelju, koji se ne oslanja na klasične metode već na metode dubinske analize podataka.

1.1 Struktura rada

Rad je podijeljen u pet cjelina:

- **Uvod**, u kojem dajemo motivaciju za razradu teme
- **Ciljevi rada**, u kojem opisujemo problem analize dnevnika pristupa i postavljamo ciljeve istraživanja
- **Sustavi za otkrivanje mrežnih napada**, u ovom poglavlju opisujemo pristupe detekcije napada na mrežnoj i aplikacijskoj razini koji se trenutno primjenjuju u praksi
- **Dnevnik pristupa Apache httpd web poslužitelja**, opisujemo korišteni izvor podataka tj. dnevnik pristupa web poslužitelju
- **Model klasifikacije anomalnog korištenja resursa web poslužiteljan**, gdje opisujemo izabrane značajke upotrebljene u izgradnji modela za klasifikaciju te evaluaciju isprobanih algoritama strojnog učenja

2 Ciljevi rada i sažeti prikaz poslovne domene

Na danom skupu podataka prikupljenom u razdoblju od jedne godine, od 12/2010. do 12/2011., isprobat ćemo algoritme dubinske analize podataka u svrhu detekcije anomalnog ponašanja korisnika koji su pristupali web poslužitelju. Na web poslužitelju se nalazi aplikacija koju je koristio odjel sa šestoro zaposlenika, aplikacija je služila u svrhu bilježenja statusa usluge koje je nudio odjel partnerima poduzeća.
Administrator web aplikacije je nehotice, ostavio otvoren pristup s Interneta, te je u promatranom razdoblju pokušano različitim načinima kompromitirati aplikaciju. Sama aplikacija je generirala vrlo malu količinu dio prometa, te je iz tog razloga prikupljeno samo 7500 događaja. Interesanti su automatizirani napadi na samu web aplikaciju i upiti koji su generirani prema web poslužitelju prilikom samih napada.

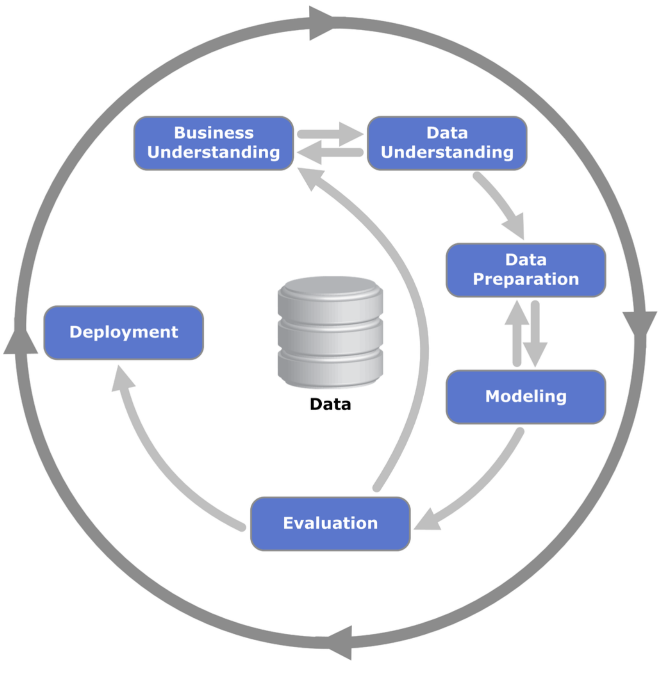
Kao što smo prethodno naveli, cilj našeg istraživanja je detekcija anomalnog ponašanja korisnika. Administrator koji nadzire sustav će prilikom svakog devijantnog ponašanja biti upozoren, ova informacija omogućava administratoru bolji nadzor nad samim sustavom.
Osnovna pretpostavka prilikom prepoznavanja anomalija jest da se anomalija u znatnoj mjeri razlikuje od normalnog ponašanja sustava. Prilikom opisivanja normalnog ponašanja korisnika posebnu pažnju valja posvetiti izboru značajki koje ćemo koristiti u procesu detekcije.
Detekcija anomalija može imati diskretne rezultate poput: anomalija, normalnog ponašenja ili sumnjivog ponašanja. Pošto događaje klasificiramo u diskretne klase, detekciju anomalija promatramo kao **klasifikacijski problem**. Kao što ćemo vidjeti u daljnjem tekstu, dnevnik pristupa ima određenu strukturu i ograničeni smo podacima koji se pišu u njega.
Također, u našem radu razlikovat ćemo dvije vrste anomalija - anomaliju na razini događaja i anomaliju na razini sesije korisnika. Anomalija na razina događaja je ona anomalija u kojoj se može ustanoviti zlonamjerno ponašanje korisnika prilikom jednog HTTP upita, dok ona na razini sesije sadrži korisnike koji su pokušali učiniti nešto zlonamjerno u svom pristupu sustavu obično kroz duži niz HTTP upita.

Kako bi postigli ovo potrebno je učiniti sljedeće:

- Označiti skup, odnosno označiti anomalane/normalne događaje
- Izabrati značajke koje će se koristiti kao ulaz u metode dubinske analize podataka
- Isprobati sljedeće metode dubinske analize podataka:

- Izabrati metodu koja postiže najbolji rezultat

Plan istraživanja i daljnji tekst slijedi korake Cross Industry Standard Process for Data Mining (CRISP/DM) metodologije.
Koraci su prikazani na slici 1, sama metoda se sastoji od šest faza. Proces počinje **(1)** *razumijevanjem problema* odnosno domene te postavljanjem ciljeva i mjera uspješnosti. U sljedećem koraku nastoje se **(2)** *razumijeti podaci* - u našem slučaju bavimo se analizom dnevnika pristupa (engl. access log), u koraku **(3)** *pripreme podataka* potrebno je podatke pripremiti u oblik koji je pogodan za analizu. Korak **(4)** *modeliranja* bavi se pripremom skupa na kojem će biti primjenjeni metode dubinske analize podataka, u našem slučaju je to podjela skupa na skup za učenje (trening skup) i skup za testiranje (testni skup). Korak **(5)** *evaluacije* bavi se testiranjem uspješnosti samih metoda, u ovoj fazi odabire se odgovarajući model koji će se koristiti u sljedećoj fazi - *primjene* **(6)**. Važno je primijetiti kako je proces ciklički, odnosno kako se teži konstantnom unaprijeđenju samih modela i pripremljenih skupova.



Slika 1: Faze CRISP/DM metodologije

3 Sustavi za detekciju neovlaštenog ulaska u informacijski sustav

Pojam neovlaštenog ulaska u informacijski sustav razlikuje se od organizacije do organizacije. Takve i slične nedozvoljene akcije trebale bi biti propisane pripadajućom sigurnosnom politikom, koja bi trebala sadržavati i procedure nakon samog proboja informacijskog sustava. U ovom poglavlju dati ćemo temeljne definicije: neovlaštenog ulaska u inf. sustav, detekcije napada i sustava za detekciju napada. Temeljne definicije nadopunit ćemo zahtjevima koje bi trebao zadovoljiti idealan sustav za detekciju napada.

Definicija 1. *Neovlašteni ulazak u informacijski sustav (engl. intrusion) je skup akcija koje počinitelj izvršava u cilju kompromitacije cjelovitosti, povjerljivosti ili dostupnosti nekog informacijskog resursa. [10]*

Definicija 2. *Detekcija neovlaštenog ulaska (engl. intrusion detection) je proces identificiranja i odgovora na maliciozni aktivnost koja za cilj ima kompromitirati računalne i mrežne resurse. [1]*

Definicija 3. *Sustav za detekciju neovlaštenog ulaska u informacijski sustav (engl. Intrusion Detection System - IDS) nadzire računalne sustave i/ili mrežnu aktivnost kako bi identificirao maliciozne ili sumnjive događaje. Prilikom svakog sumnjivog događaja IDS javlja upozorenje.*

IDS bi trebao biti u stanju detektirati, izvijestiti i spriječiti širok raspon sigurnosnih prijetnji uključujući penetracijske pokušaje, pokretanje zlonamjernih programa, neautorizirane mrežne konekcije i ostale slične sigurnosne prijetnje. Idealan IDS trebao bi poduzimati akcije u realnom vremenu, izdržati napade uskraćivanjem usluge, detektirati nepoznate i poznate metode napada uz što manji broj krivo detektiranih napada (lažnih pozitiva) [8].

Prema Crosbiju i Spaffordu [6] IDS bi trebao zadovoljiti sljedeće zahtjeve, neovisno o pristupu na kojemu je temeljen:

- U mogućnosti je samostalno raditi bez ljudskog nadzora. Sustav bi trebao pouzdano funkcionirati u pozadini i biti razumljiv svojim korisnicima. Interne funkcionalnosti bi trebale biti provjerljive izvana.
- Trebao bi biti otporan na greške, primjerice u slučaju isapada operacijskog sustava trebao bi obnoviti svoju bazu znanja prilikom ponovnog pokretanja.
- Optimalno bi trebao trošiti resurse koji su mu na raspolaganju.

- Dojavljivao bi devijacije u korištenju samih resursa
- Prilagodljiv je sustavu kojeg osigurava, svaki sustav koji se osigurava ima svoj uzorak korištenja dok mehanizmi obrane bi se trebali na jednostavan način prilagoditi samom sustavu koji osiguravaju.
- IDS bi se trebao prilagođavati promjenjivoj okolini kroz vrijeme.
- Naposljetku trebao bi biti otporan na različite oblike prijevara.

3.1 Vrste IDS-ova u odnosu na izvor podataka

IDS sustav na razini računala (engl. Host based Intrustion Detection System - HIDS) namijenjen je analizi interakcije između aplikacija na računalu i operacijskog sustava, HIDS sustav nadzire: datoteke, procese, zapise dnevnika (logove), memoriju i sl. Temeljnu ideju ovakvog pristupa detekciji napada opisala je Denning [7], njezin sustav je nadzirao zapise dnevnika u koje su sustavske aplikacije zapisivale događaje. Ukoliko bi sustav za nadzor uočio neki neuobičajen događaj javio bi upozorenje.
Popularno rješenje takvog sustava je *OSSEC* (`http://wwww.ossec.com`). On pomoću različitih zapisa dnevnika (engl. log) u realnom vremenu detektira moguće napade, pridjeljuje im razinu prijetnje te šalje upozorenja korisnicima. Korisnik može definirati vlastita pravila za detekciju putem *OSSEC*-ovog jezika.
U današnje vrijeme zbog porasti razine složenosti samih aplikacija ali i zbog resursa koji su potrebni HIDS-ovima njihova popularnost opada [8].

Mrežni IDS (engl. Network Intrusion Detection System - NIDS) namijenjen je analizi mrežnog prometa, NIDS sustav nadzire određeni mrežni segment i analizira promet koji prolazi kroz taj segment [3]. NIDS pretežno prisluškuje mrežu u pasivnom modu koji ne degradira performanse poput promiskuitetnog moda koji stvara kopije paketa na razini cijele mreže. Sustav prilikom inspekcije paketa traži uzorke svojstvene napadima te omogućuje korelaciju s drugim računalima na mreži kako bi detektirao prema kojim odredištima je napad usmjeren. U današnje vrijeme kriptirani promet kriptiran IPSecom, SSH-om, SSL/TLS sve više otežava rad ovakvih sustava [3]. Također, veliki problem predstavlja rekonstrukcija IP datagrama i TCP segmenata. Primjer popularnog NIDS sustava predstavlja *Snort*.

3.2 Vrste IDS-ova u odnosu na pristup otkrivanja napada

Inghman navodi dva razloga [12] zašto je potrebno testirati IDS-ove: (1) prvi je verifikacija efikasnosti algoritama za detekciju napada, (2) drugi je usporedba dva ili više algoritama kako bi utvrdili koji algoritam se bolje ponaša u odnosu na odabranu metriku. Testiranje IDS predstavlja opsežan posao zato što podaci za testiranje trebaju biti realistični i efikasni, ukoliko koristimo metode nadziranog učenja skup za učenje treba biti ispravno označen te se postavlja pitanje kako zaštititi privatnost korisnika [12]. Athanasiades i kolege [2] pesimistično konstatiraju kako problem detekcije napada nikada neće biti riješen na odgovarajući način.
U prethodnom dijelu naveli smo osnovnu podjelu IDS-ova s obzirom na izvore podataka. U ovom dijelu opisat ćemo različite pristupe koje koriste IDS sustavi za detekciju samih napada.

3.2.1 IDS-ovi temeljeni na pragovim tolerancije

Jedan od osnovnih načina detekcije napada je korištenje statističkih metoda. Detekcija korištenjem pragova tolerancije (engl. threshold detection) omogućuje IDS-u uočavanje događaja koji u određenim vremenskim intervalima premašuju vrijednosti koje su ustanovljene prilikom normalnog korištenja sustava [8]. Prema [8], detekcija korištenjem pragova tolerancije predstavlja loš detekcijski mehanizam te se koristi kao potkomponenta IDS-ova.

3.2.2 IDS-ovi temeljeni na pravilima

IDS temeljen na pravilima (engl. signature detecting ili pattern matching) predstavlja sustav koji sadrži određeni skup pravila koja javaljaju upozorenja ukoliko se zadovolji određeno pravilo [8]. Potpisi se najćešće pohranjuju u bazama pravila te se događaji uspoređuju sa samom bazom. Poznati primjer ovakvog sustava je Snort. Problem ovakvih sustava je statičnost samih potpisa te što male modifikacije nad potpisima mogu uzrokovati neefikasnost sustava detekcije. Isti problem s novim ranjivostima (popularno nazvanim zero-day), koje sustav za detekciju ne može otkriti jer ne posjeduje odgovarajuće potpise.

3.2.3 IDS-ovi temeljeni na anomalijama

Zbog ograničenja IDS-ova koji se temelje na potpisima (engl. signature based) koji nisu u stanju otkriti nepoznate i nove napade istraživači su započeli pronalaziti nove načine i pristupe sustavima za otkrivanje napada.

Jedan od načina su i sustavi bazirani na signaliziranju anomalija, sustavi bazirani na anomalijama grade statistički model normalnog ponašanja sustava, ovo je ujedno i faza učenja sustava za detekciju. Statistički model uspoređuje ulaze s normalnim stanjem sustava te ako uoči velika odstupanja javlja upozorenje. Pretpostavka sustava temeljenih na anomalijama je da se anomalije znatno razlikuju od normalnog ponašanja sustava.
Glavna prednost sustava sa signaliziranjem anomalija predstavlja a priori detekcija novih napada bez znanja i pravila o takvim napadima [8]. Probleme uzrokoju pretpostavke normalnog ponašanja sustava te odstupanja koja mogu predstavljati anomaliju u obliku napada ali i prihvatljivo korištenje sustava. Drugim riječima, česti su FP koje sustav detekcije javlja. Iscrpan pregled metoda koje se koriste u detekciji anomalija, ne samo u računalnoj sigurnosti već i u drugim granama, je rad Chandola i kolega [5].

3.3 Evaluacija IDS-ova

Prilikom nadzora mrežnog prometa IDS nastoji detektirati pokušaje napada. Pokušaje napada definiramo još kao pozitive. Ukoliko je tekući događaj pravi pokušaj napada njega definiramo kao **istiniti pozitiv(engl. true positiv - TP)**, odnoso IDS ispravno klasificira događaj koji je ujedno i napad. Ukoliko IDS označi događaj kao pokušaj napada, a taj napad ne postoji taj događaj se tretira kao **lažni pozitiv (engl. false postive - FP)**. Dakle, FP je lažno upozorenje za normalan (bezopasan) događaj.
Događaji koje IDS bilježi i koji nisu opasni nazivano negativima. Ukoliko IDS klasificira točno bezopasan događaj tada se taj događaj naziva **istinit negativ(engl. true negative -TN)**. Nasuprot toga, ako je IDS-ova klasifikacija netočna za bezopasan događaj, tada se događaj (koji je ujedno i pokušaj napada) naziva **lažni negativ (engl. false negative - FN)**.

Mjeru istinitih pozitiva (eng. True postive rate TPR) ili mjeru otkrivenih opasnih događaja definiramo kao:

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

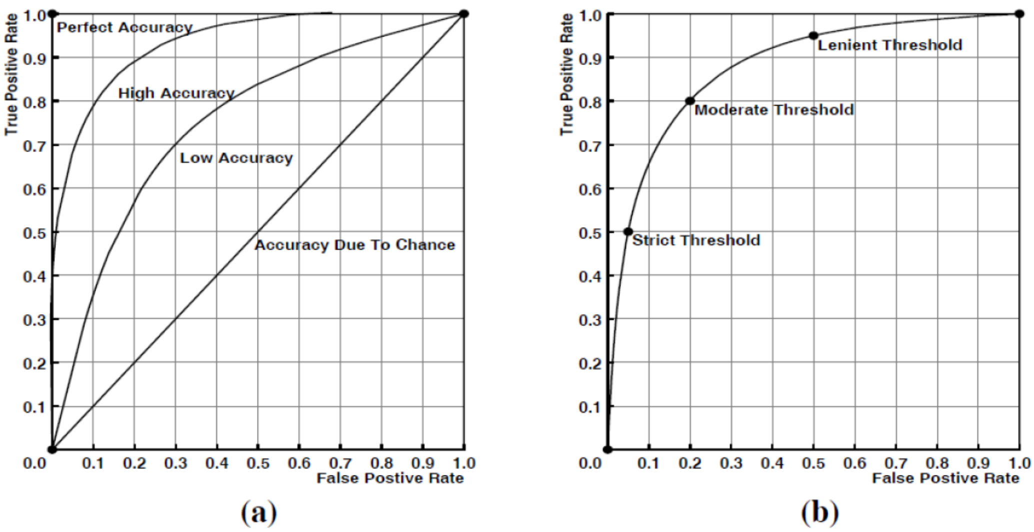
Gdje je TP broj istinitih pozitiva, a FN broj lažnih negativa. Idealan IDS ima TPR=1. TPR mjeri omjer između uspješno klasificiranih opasnih događaja i ukupnog broja opasnih događaja $(TP + FN)$.
Mjeru lažnih pozitiva (engl. False Positive Rate - FPR) definiramo kao:

$$FPR = \frac{FP}{FP + TN}$$

Gdje je FP broj lažnih pozitiva, a TN broj istinitih negativa. Drugim riječima, FPR mjeri omjer između normalnih događaja koji su označeni kao opasni i ukupnog broja normalnih događaja.

3.4 ROC krivulja

Receiver operating characteristic (ROC) krivulja se često koristi za procjenu uspješnosti IDS [16]. Ona pokazuje odnos između stvarnih napada (TPR) i pogrešno signaliziranih napada (FPR). ROC krivulja je dvodimenzionalni prikaz točnosti detekcije određenog signala, na y-osi se nalazi mjera detekcije stvarnih napada (TPR) dok na x-osi se nalazi mjera krivo klasificiranih napada (FPR). Idealan sustav bi se nalazio u točci (0,1), dok u zadovoljavajućem sustavu vrijednosti na y-osi moraju rasti brže nego one na x-osi.



Slika 2: Primjeri ROC krivulja (preuzeto iz [16])

Na slici 2a. prikazana je idealna krivulja koja prolazi točkom (0,1), također linearna krivulja (pod nazivom *Accuracy Due to Chance*) je ona koja je dobivena slučajnim izborom između pozitiva i negativa. Na slici 2b. prikazane su točke koje odgovaraju različitim pragovima tolerancije.

4 Analiza dnevnika pristupa web poslužitelja

4.1 Opis poslovnog sustava

Kao što smo naveli u poglavlju 2, naš sustav se sastoji od web poslužitelja (Apache httpd poslužitelj) koji je opsluživao interno napravljenu web aplikaciju čija je zadaća bila bilježenje stanja usluge koje je poduzeće nudilo svojim partnerima. Za sam opis poslovnog sustava opisivanje funkcionalnosti same aplikacije bilo bi nepotrebno. Važno je jedino napomenuti da je sustav služio za zapisivanje pojedinosti o partnerima s kojima je poduzeće surađivalo. O partnerima su se bilježili različiti meta atributi iz kojih je bilo moguće stvoriti izvještaje i dobiti uvid u stanje partnera.

Web aplikacija je komunicirala sa svojim korisnicima putem zaštićenog HTTP kanala, kanal je osiguravao SSL certifikat. Poduzeće je ovu uslugu koristilo interno, partneri su po potrebi mogli dobiti uvid u izvještaje same usluge. Web poslužitelj je generirao nisku količinu prometa, do trenutka kada je greškom administrator poslužitelja propustio pristup s dijela Interneta prema sučelju web aplikacije. Greška administratora je bila uzrok povećanju prometa prema web poslužitelju, unatoč tome prikupljen je zanimljivi skup podataka koji je korišten u ovom radu. Skup podataka je sadržavao pokušaje napade s različitih dijelova svijeta, na sreću web aplikacija nije bila kompromitirana u spomenutim napadima.

4.2 Dnevnik pristupa Apache httpd poslužitelja

Apache poslužitelj osim svoje primarne zadaće posluživanja statičkih i dinamičkih web stranica pohranjuje korisne informacije o greškama i pristupima korisnika. Greške se pohranjuju u datoteci *error.log* dok pristupi web serveru u datoteci *access.log*. Direktivnom *CustomLog* [11] definira se lokacija i struktura zapisa u logovima, svaki upit prema web serveru, bez obzira na HTTP statusni kôd, zapisuje se u dnevnik pristupa.

Analiza dnevnika pristupa korisna je administratorima web sjedišta, aplikacije poput Webalizera (`http://www.webalizer.org/`), Analoga (`http://www.analog.cx/`) i Awstatsa (`http://awstats.sourceforge.net/`) omogućuju sumarne statistike poput broja posjeta, vremenskog trajanja posjeta, klikova po stranici, prethodnih poveznica (engl. referer), zemalja iz kojih dolaze korisnici, korištenih web preglednika/operacijskih sustava, HTTP greški i sl. Navedene aplikacije daju dobru sliku o uspješnosti web sjedišta te pomažu prilikom planiranja oglašavanja sjedišta, donošenju odluka o nadogradnji hardvera web poslužitelja i sl.

Iz potonjeg možemo zaključiti da spremanjem podataka o korisnikovom pristupu tek započinje proces upravljanja dnevnikom pristupa. Zapise dnevnika pristupa možemo koristiti kao osnovni ulaz u sustav poput HIDS-a. Trenutno najpoznatiji HIDS *OSSEC* koristi potpise koji se uspoređuju s događajima na promatranim računalima ili segmentima mreže. Osnovi nedostatak takvih sustava je što potpisi brzo zastarijevaju, pojavljuju se novi napadi i teško je pratiti brze promjene u propustima i verzijama aplikacija.

Ono što olakšava analizu dnevnika pristupa je njegova struktura, dokumentacija Apache poslužitelja naglašava kako se struktura na jednostavan način može prilagoditi potrebama korisnika [11].

Dva glavna formata dnevnika pristupa u Apache poslužitelju su:

1. *Common Log* ili uobičajeni zapis
2. *Combined Log* ili kombinirani zapis

Uobičajeni ili *common log* se definira na sljedeći način:

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog logs/access_log common
```

Ovakav format zapisa generirao bi sljedeći zapis u dnevniku pristupa:

```
220.226.103.254 - - [13/Oct/2011:03:55:52 +0200] "GET /pma/
scripts/setup.php HTTP/1.1" 302 530
```

Značenje pojedinih slogova u zapisu bilo bi sljedeće:

Oznaka %h (220.226.103.254) označava IP adresu klijenta (udaljenog računala) s kojeg je upućen upit prema web serveru.

Oznaka %l (-) predstavlja identitet (oznaku) klijenta prema RFC 1413[1]. Ova informacija je nepouzdana i korisna je samo u internim mrežama. U našem slučaju crtica označava da oznaka nije dostupna.

Oznaka %u (-) predstavlja identifikator korisnika koji šalje upit temeljem HTTP autentikacije. U našem slučaju zatraženi dokument nije bio zaštićen HTTP autentikacijom te informacija nije dostupna.

[1] http://www.ietf.org/rfc/rfc1413.txt

Oznaka %t (13/Oct/2011:03:55:52 +0200) predstavlja vrijeme kada je zaprimljen zahtjev poslan web serveru.

Oznaka \"%r\" (GET /pma/scripts/setup.php HTTP/1.1) predstavlja zatraženu stranicu od klijenta. U našem primjeru klijent je zatražio putem GET metode putanju /pma/scripts/setup.php protokolom HTTP 1.1.

Oznaka %>s (302) predstavlja statusni kôd HTTP protokola koji je vratio web poslužitelj klijentu. Statusni kôd nam govori o uspjehu ili grešci prilikom zahtjeva. Specifikacija HTTP protokola (RFC 2616, poglavlje 10)[2] sadrži potpunu listu statusnih kôdova. U našem slučaju web server je vratio kôd 302 koji označuje redirekciju.

Oznaka %b (530) predstavlja veličinu objekta (u bajtovima) koji je vraćen klijentu, veličina objekta ne sadrži zaglavlja odgovora.

Kombinirani zapis ima sljedeću strukturu:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
agent}i\"" combined
CustomLog log/access_log combined
```

Kombinirani zapis pohranjuje sljedeći slog u dnevniku pristupa:

```
127.0.0.1 - - [05/Jan/2012:11:36:28 +0100]
"GET /svijet-o-sigurnosti/lotd/ HTTP/1.1" 200 7988
"https://sigurnost.carnet.hr/svijet-o-sigurnosti/lotd/post/"
"Mozilla/5.0 (Windows NT 6.1) AppleWebKit/535.7 (KHTML, like
Gecko)
Chrome/16.0.912.63 Safari/535.7"
```

Uz pomoć direktive *%{header}i.* svakom zahtjevu prema web serveru mogu se pridužiti dodatna polja koja se zapisuju u log datoteku. U našem primjeru dodana su polja:

\"%{Referer}i\" označuje s kojeg vanjskog sjedišta je korisnik kliknuo na poveznicu koja vodi prema našem sjedištu. Polje Referer je prisutno u HTTP zaglavlju zahtjeva.U našem primjeru zahtjev je upućen sa sjedišta `https://sigurnost.carnet.hr/svijet-o-sigurnosti/lotd/post/`.

[2] `http://www.ietf.org/rfc/rfc2616.txt`

\"%{User-agent}i\" je isto dio HTTP zaglavlja. U polje se zapisuje vrijednost HTTP zaglavlja koje sadrži klijentski web preglednik s kojime je korisnik pristupio web sjedištu. U našem primjeru korisnik se koristio web preglednikom Chrome 16.0.912.63 i operacijskim sustavom Windows NT 6.1.

Apache poslužitelj omogućuje stvaranje uvjetnih (engl. conditional) logova. Za zapisivanje u uvjetne logove koriste se *environment* varijable i uvjetni upiti koji nam omogućuju da zapisujemo ono što želimo u trenutku kada to želimo. Analizom dnevnika pristupa možemo saznati dodatne informacije, tako primjerice možemo saznati:

- Prema IP adresi zahtjeva možemo saznati iz koje države potječe korisnik koji je uputio zahtjev. Mapiranje se radi uz pomoć javno dostupnu geolokacijsku bazu GeoIP [3].
- Uz pomoć IP adrese može saznati pripada li klijent nekom od botneta. Također, IP adresa zahtjeva može poslužiti za identifikaciju anonimizacijskih servisa poput čvorova TOR mreže.
- Moguće je odrediti vrstu operacijskih sustava i web preglednik koji korisnici koriste. Isto tako, moguće je utvrditi vrstu uređaja s kojom je korisnik pristupao web sjedište, primjeri je li koristio stolno računalo, tablet ili mobitel.
- Moguće je iz agenta koji se koriste izdvojiti neuobičajene agente i robote koji služe za mapiranje web sjedišta. Najpoznatiji robot u današnje vrijeme na Internetu je zasigurno *GoogleBot* koji služi za indeksiranje sjedišta.
- Vrijeme pristupa i intervali između upita mogu poslužiti za identificiranje automatskih skenera koji koriste hakeri.
- Prema zatraženim putanja na web sjedištu moguće je odrediti radi li se o napadu koji šalje kombinacije vektora napada prema web sjedištu koji se nalaze pohranjene u tzv. rječniku.
- Korištenje zastarjelih operacijskih sustava može ukazivati na kompromitirana računala koja se nalaze u nekoj od botnet mreža i koje su upravljane od strane kriminalaca.

[3]Maxmind GeoIP, `http://www.maxmind.com/app/geoip_country`

4.3 Analiza prikupljenih dnevnika pristupa

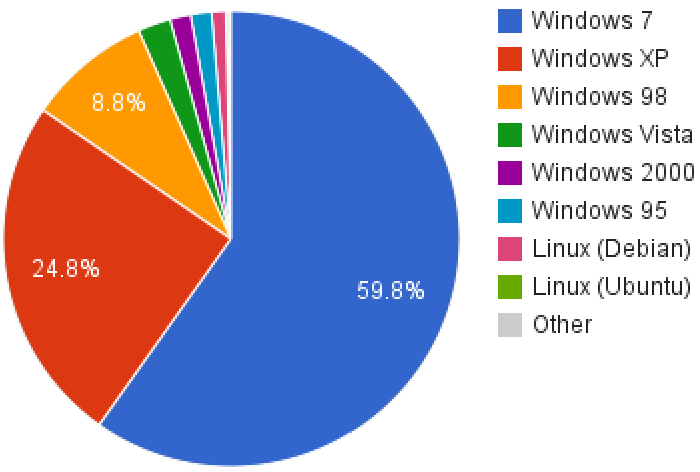
U svrhu analize dnevnika pristupa Apache poslužitelja koristili smo se zapisima prikupljenim tijekom slobodnog pristupa web aplikaciji, koju smo opisali u poglavlju 2.
U svrhu analize napisana je skripta koja služi čišćenju sakupljenih zapisa dnevnika pristupa i strukturiranje kako bi se olakšala njihova analiza. Skripta sprema u relacijsku bazu podataka ekstrahirane atribute iz samog dnevnika zapisa te ih nadopunjuje meta podacima koje je moguće dobiti iz samih atributa.

Tako smo uz pomoć GeoIP baze dobili mapiranja između IP adresa i zemalja, ASN (Autonomous System Number) oznake ISP operatera koji su vlasnici IP adrese, vrsti uređaja korisnika i dr. Na tablici 1 prikazani su pristupi po zemljama porijekla, možemo uočiti da su zemlje s većim brojem zahtjeva: Hrvatska, SAD, Kina, Rusija, Njemačka, Velika Britanija, Južna Korea i Rumunjska. Pošto se web poslužitelj i korisnici koji koriste web aplikaciju nalaze u Hrvatskoj razumljiv je velik broj zahtjeva.
Također, veliki broj upita potječe iz SAD-a tome su zasluži roboti za indeksiranje koji najčešće pripadaju kompanijama iz SAD-a. Sumnjivi su pristupi i zemalja gdje je računalni kriminal izražen, poput - Kine, Rusije, Rumunjske i dr.

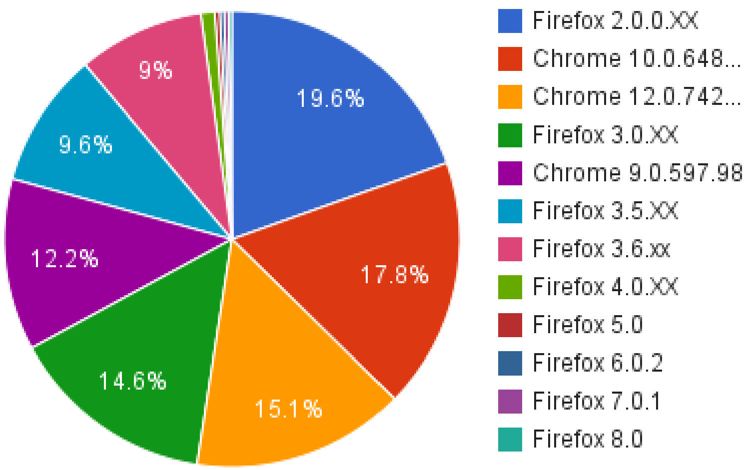
Tablica 1: Distribucija pristupa s obzirom na državu kojoj pripada korisnik

Država	**Broj pristupa**
Croatia	3214
United States	1373
China	619
Russian Federation	330
Germany	299
United Kingdom	271
Korea, Republic of	178
Romania	146
Italy	118
Thailand	118
Canada	103
France	101

Na slici 3 prikazana je distribucija najpopularnijih operacijskih sustava s kojima su korisnici pristupali web serveru. Najpopularniji operacijski sustavi su Windows 7 i Windows XP, zanimljivo je korištenje zastarjelih sustava poput Windows 98, Windows 95 i Windows 2000. Postoji mogućnost da su računala sa zastarjelim operacijskim sustavima kompromitirani i koriste se kao dijelovi botneta.



Slika 3: Distribucija operacijskih sustava klijenata



Slika 4: Distribucija web preglednika

Na slici 4 nalazi se distribucija web preglednika s kojima je pristupano web polsužitelju. Mozilla Firefox i Google Chrome su najpopularniji, zbog dužeg vremenskom perioda vidiljivo je korištenje različitih verzija ovih web preglednika.

Tablica 2: Distribucija nepoznatih agenata koji su indeksirali web sjedište

Nepoznati agent	**Udjel**
- (sakriven ili nepoznat)	38.99%
Morfeus F*** Scanner	22.66%
SSL Labs (https://www.ssllabs.com/)	17.12%
check_http/v1.4.15 (nagios-plugins 1.4.15)	8.56%
Comodo-Certificates-Spider	6.65%
Morfeus strikes again	1.08%
ZMEU LIBER! The Whitehat Anti-sec Group	0.65%
Ostali	4.29%

Na tablici 2 prikazana je distribucija automatiziranih i nepoznatih robota koji su pristupali stranicama. Većina ovih robota koristila se za nelegalne aktivnosti poput pokušaja kompomitacije web sjedišta. Izuzetak je Nagios agent (check_http) koji se koristio interno za provjeravanje dostupnosti web sjedišta.

5 Model klasifikacije anomalnog korištenja resursa web poslužitelja

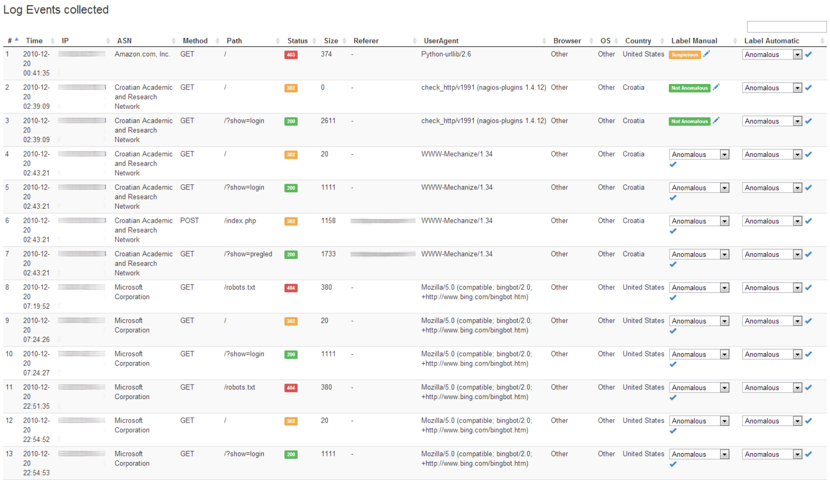
5.1 Priprema podataka

Za potrebe označavanja skupa podataka, kojeg ćemo koristiti u metodama nadziranog učenja za treniranje, razvijena je web aplikacija nazvana Logminer namijenjena označavanju samog skupa.
Sustav učitava datoteke pristupa i sprema ih u bazu podataka, sastoji se od dva dijela: svih upita i sesija korisnika. Predprocesirani zapisi iz dnevnika pristupa prikazani su na slici 5 i sadrže sljedeća polja:

- vremena pristupa,
- IP adrese,
- korištene HTTP metode,
- zahtijevanoj putanji na web poslužitelju,
- HTTP status koji je vraćen u odgovoru,
- korisnički agent odnosno preglednik s kojim je pristupano,
- veličina vraćenog objekta u odgovoru
- stranica s koje je korisnik došao na promatrano web sjedište.

Prilikom unosa zapisa iz dnevnika pristupa iz samih atributa se izvode i sljedeći atributi:

- vlasnik IP adrese (davatelj Internet usluge)
- korišteni preglednik i njegova verzija
- verzija operacijskog sustava
- zemlja porijekla korisnika



Log Events collected

#	Time	IP	ASN	Method	Path	Status	Size	Referer	UserAgent	Browser	OS	Country	Label Manual	Label Automatic
1	2010-12-20 00:41:35		Amazon.com, Inc.	GET	/	403	374	-	Python-urllib/2.6	Other	Other	United States	Suspicious	Anomalous
2	2010-12-20 02:39:09		Croatian Academic and Research Network	GET	/	302	0	-	check_http/v1991 (nagios-plugins 1.4.12)	Other	Other	Croatia	Not Anomalous	Anomalous
3	2010-12-20 02:39:09		Croatian Academic and Research Network	GET	/?show=login	200	2611	-	check_http/v1991 (nagios-plugins 1.4.12)	Other	Other	Croatia	Not Anomalous	Anomalous
4	2010-12-20 02:43:21		Croatian Academic and Research Network	GET	/	302	20	-	WWW-Mechanize/1.34	Other	Other	Croatia	Anomalous	Anomalous
5	2010-12-20 02:43:21		Croatian Academic and Research Network	GET	/?show=login	200	1111	-	WWW-Mechanize/1.34	Other	Other	Croatia	Anomalous	Anomalous
6	2010-12-20 02:43:21		Croatian Academic and Research Network	POST	/index.php	302	1158		WWW-Mechanize/1.34	Other	Other	Croatia	Anomalous	Anomalous
7	2010-12-20 02:43:21		Croatian Academic and Research Network	GET	/?show=pregled	200	1733		WWW-Mechanize/1.34	Other	Other	Croatia	Anomalous	Anomalous
8	2010-12-20 07:19:52		Microsoft Corporation	GET	/robots.txt	404	380	-	Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)	Other	Other	United States	Anomalous	Anomalous
9	2010-12-20 07:24:26		Microsoft Corporation	GET	/	302	20	-	Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)	Other	Other	United States	Anomalous	Anomalous
10	2010-12-20 07:24:27		Microsoft Corporation	GET	/?show=login	200	1111	-	Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)	Other	Other	United States	Anomalous	Anomalous
11	2010-12-20 22:51:35		Microsoft Corporation	GET	/robots.txt	404	380	-	Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)	Other	Other	United States	Anomalous	Anomalous
12	2010-12-20 22:54:52		Microsoft Corporation	GET	/	302	20	-	Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)	Other	Other	United States	Anomalous	Anomalous
13	2010-12-20 22:54:53		Microsoft Corporation	GET	/?show=login	200	1111	-	Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)	Other	Other	United States	Anomalous	Anomalous

Slika 5: Početni izgled Logminer web aplikacije, prikazani su prikupljeni događaji iz dnevnika pristupa

Za označavanje skupa postoje dvije vrste oznaka: za ručno (manual label) i automatsko (automatic label) označavanje. Atribut za automatsko označavanje namijenjen je metodi koja automatizira sam proces označavanja skupa (postupak je opisan u 5.1.2).
Iako su ovi atributi vidljivi u samoj Logminer aplikaciji, u bazi postoje i atributi vezani za vrste uređaja s kojima je korisnik pristupao i AS (engl. autonomous system) brojeve vezane za vlasnike IP adresa.

5.1.1 Prepoznavanje korisnika i njihovih sesija

Prilikom analize dnevnika pristupa anomalije možemo promatrati na dvije razine: razini događaja i razini sesije korisnika.

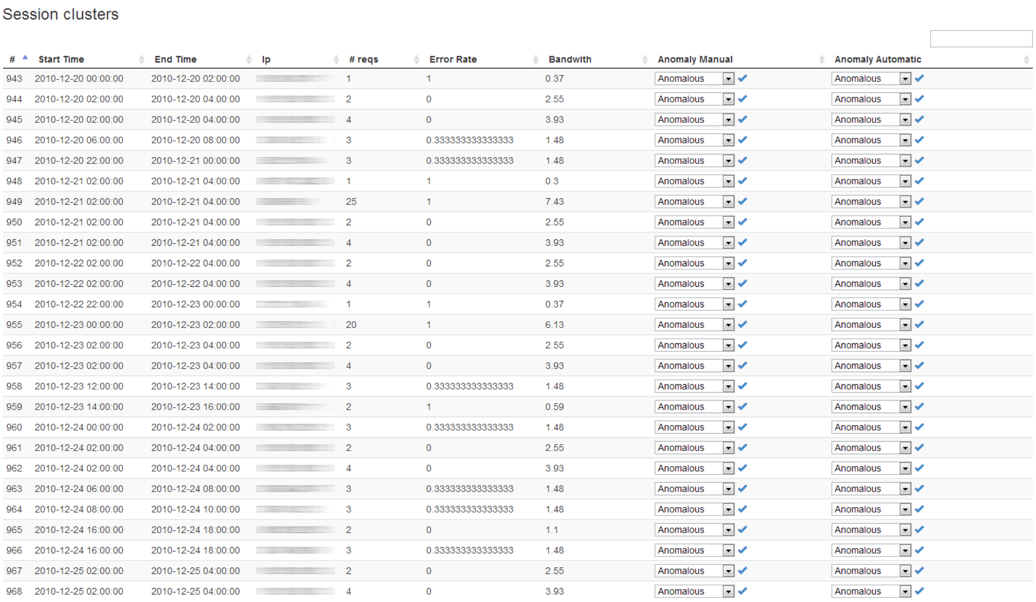
Anomalija na razini događaja očituje se u čudnim upitima prema web poslužitelju (npr. česte su duge putanje), nepoznati preglednici korišteni prilikom upita koji su često i lažno navedeni od strane klijenta, IP adrese koje spadaju u raspon različitih anonimizacijskih servisa ili potječu iz država u kojima je računalni kriminal uobičajen.

Anomalije na razini događaja često su vezane za napade na web aplikacije, pregled metoda korištenih u rješavanju ovog problema može se pronaći u [12] i [15].

U ovom radu analizirane su anomalije na razini sesije. Pošto sesiju možemo definirati na različite načine i njezina definicija se razlikuje od autora do autora [22], mi smo prilikom stvaranja sesija u ovom radu koristili sljedeću metodu:

1. Događaje u jednom danu podijelili smo po određenom vremenskom rasponu ($\Delta t = 2h$)

2. Svi oni događaji u tom vremenskom rasponu koji imaju jedanku IP adresu i preglednik smatrani su sesijom



Session clusters

#	Start Time	End Time	Ip	# reqs	Error Rate	Bandwith	Anomaly Manual	Anomaly Automatic
943	2010-12-20 00:00:00	2010-12-20 02:00:00		1	1	0.37	Anomalous	Anomalous
944	2010-12-20 02:00:00	2010-12-20 04:00:00		2	0	2.55	Anomalous	Anomalous
945	2010-12-20 02:00:00	2010-12-20 04:00:00		4	0	3.93	Anomalous	Anomalous
946	2010-12-20 06:00:00	2010-12-20 08:00:00		3	0.333333333333333	1.48	Anomalous	Anomalous
947	2010-12-20 22:00:00	2010-12-21 00:00:00		3	0.333333333333333	1.48	Anomalous	Anomalous
948	2010-12-21 02:00:00	2010-12-21 04:00:00		1	1	0.3	Anomalous	Anomalous
949	2010-12-21 02:00:00	2010-12-21 04:00:00		25	1	7.43	Anomalous	Anomalous
950	2010-12-21 02:00:00	2010-12-21 04:00:00		2	0	2.55	Anomalous	Anomalous
951	2010-12-21 02:00:00	2010-12-21 04:00:00		4	0	3.93	Anomalous	Anomalous
952	2010-12-22 02:00:00	2010-12-22 04:00:00		2	0	2.55	Anomalous	Anomalous
953	2010-12-22 02:00:00	2010-12-22 04:00:00		4	0	3.93	Anomalous	Anomalous
954	2010-12-22 22:00:00	2010-12-23 00:00:00		1	1	0.37	Anomalous	Anomalous
955	2010-12-23 00:00:00	2010-12-23 02:00:00		20	1	6.13	Anomalous	Anomalous
956	2010-12-23 02:00:00	2010-12-23 04:00:00		2	0	2.55	Anomalous	Anomalous
957	2010-12-23 02:00:00	2010-12-23 04:00:00		4	0	3.93	Anomalous	Anomalous
958	2010-12-23 12:00:00	2010-12-23 14:00:00		3	0.333333333333333	1.48	Anomalous	Anomalous
959	2010-12-23 14:00:00	2010-12-23 16:00:00		2	1	0.59	Anomalous	Anomalous
960	2010-12-24 00:00:00	2010-12-24 02:00:00		3	0.333333333333333	1.48	Anomalous	Anomalous
961	2010-12-24 02:00:00	2010-12-24 04:00:00		2	0	2.55	Anomalous	Anomalous
962	2010-12-24 02:00:00	2010-12-24 04:00:00		4	0	3.93	Anomalous	Anomalous
963	2010-12-24 06:00:00	2010-12-24 08:00:00		3	0.333333333333333	1.48	Anomalous	Anomalous
964	2010-12-24 08:00:00	2010-12-24 10:00:00		3	0.333333333333333	1.48	Anomalous	Anomalous
965	2010-12-24 16:00:00	2010-12-24 18:00:00		2	0	1.1	Anomalous	Anomalous
966	2010-12-24 16:00:00	2010-12-24 18:00:00		3	0.333333333333333	1.48	Anomalous	Anomalous
967	2010-12-25 02:00:00	2010-12-25 04:00:00		2	0	2.55	Anomalous	Anomalous
968	2010-12-25 02:00:00	2010-12-25 04:00:00		4	0	3.93	Anomalous	Anomalous

Slika 6: Prikaz klastera na razini sesija korisnika

Obavezna promjena IP adrese kod kućnih (broadband) korisnika na razini jednog dana (kod hrvatski ISP-ova) predstavlja problem prilikom detekcije korisnika. Vremenski raspon na razini pojedinog dana omogućuje nam razlikovati pristupe korisnika u različitim dijelovima dana te donekle umanjuje problem promjene IP adresa kod kućnih korisnika. Sesije korisnika u Logmineru prikazana su na slici 6.

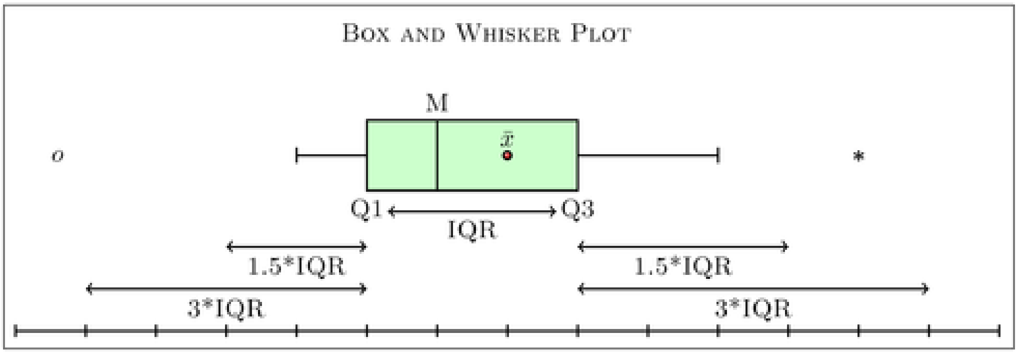
Svaka sesija opisana je određenim skupom značajki. Ove značajke kasnije koristimo u fazi izgradnje modela pomoću algoritama strojnog učenja.
Izvedene značajke na razini sesije su sljedeće (engleski nazivi odgovaraju strukturi u bazi podatka):

- **start_time** - početak sesije
- **end_time** - kraj sesije
- **ip** - IP adresa korištena od stane korisnika u danoj sesiji
- **agent** - zapis web preglednika korištenog od korisnika
- **num_requests** - broj upita u pojedinoj sesiji
- **error_rate** - omjer HTTP s vraćenom greškom
- **nonget_rate** - omjer upita koji ne koriste GET metodu za dohvaćanje sadržaja. Ostale metode se koriste za slanje, brisanje, provjeru dostupnosti ili ažuriranje sadržaja (npr. POST, PUT, DELETE, HEAD ...)
- **session_duration** - vremensko trajanje sesije u sekundama
- **stddev_btwn_reqs** - standardna devijacija između upita
- **anomaly_manual** - ručno posatvljena oznaka o klasi pojedine sesije, oznaka -1 je za normalne sesije dok je 1 za anomalne
- **anomaly_automatic** - oznaka o klasi sesije postavljena automatskom metodom

5.1.2 Označavanje skupa korisničkih sesija

Detekcija anomalija predstavlja klasifikacijski problem za čije rješavanje planiramo koristiti metode nadziranog učenja. Preduvjet za korištenje tih metoda je prethodno označen skup tj. svaka sesija mora biti označena - sesija može biti anomalna ili normalna. Zbog količine podataka (1998 sesije korisnika), ručno uređivanje oznaka postaje dugotrajan i zamoran posao. Iz ovog razloga odlučili smo se na označavanje danog skupa uz pomoć interkvartilnog razmaka.

Prema Šošiću [19], interkvartil predstavlja razliku između gornjeg i donjeg kvartila ($IQR = Q_3 - Q_1$), medijan ili drugi kvartil dijeli raspon vrijednosti u promatranom skupu na dva jednaka dijela. Donji kvartil (Q_1) dijeli promatrani skup na 25% vrijednosti koje su manje od Q_1, dok 75% vrijednosti su veće od vrijednosti Q_1. Gornji kvartil Q_3 dijeli skup tako da postoji 25% vrijednosti u promatranom skupu koje su veće od Q_3, dok 75% vrjednosti su manje od Q_3. Često se interkvartili raspon prikazuje u obliku *Box and Whisker* dijagrama, primjer takvog dijagram može se vidjeti na slici 7.



Slika 7: Box and Whiskers dijagram s označenim kvartilima i interkvartilnim rasponom

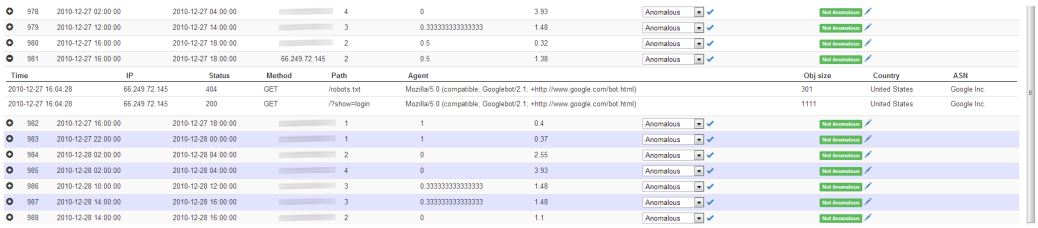
Prilikom crtanja *Box and Whisker* dijagrama definiramo unutrašnje i vanjske međe. Donja unutrašnja međa jednaka je $Q_1 - 1.5 \times IQR$, dok je gornja unutarnja međa jednaka $Q_3 + 1.5 \times IQR$. Vrijednosti koje se nalaze izvan unutrašnjih međa predstavaljaju predstavljaju netipične i sumnjive vrijednosti (engl. outlier).

U našem slučaju anomalni događaji se označavaju na temelju broja upita u pojedinoj sesiji, broj upita u sesiji da bi bio sumnjiv treba biti ili manji od donje unutarnje međe ili veći od gornje unutarnje međe.
Pošto anomalni događaji nisu vezani isključivo za broj upita kao dodatni

atribut prilikom označavanja korištena je i vrijednost omjera pogrešaka:

$$Omjer\_pogrešaka = \frac{Broj\_odgovora\_s\_vraćenom\_greškom}{Ukupan\_broj\_upita}$$

Odgovor s vraćenom greškom je onaj čiji se HTTP status između 400 i 500. U našem slučaju prag normalnih sesija ima maksimalni omjer pogrešaka od 0,3. Dakle sve neuobičajene vrijednosti broja upita izvan unutarnjih međa i sesije čiji je omjer pogrešaka veći od 0,3 smatrane su sumnjivim događajima.



978	2010-12-27 02:00:00	2010-12-27 04:00:00		4	0	3.93	Anomalous	Not Anomalous
979	2010-12-27 12:00:00	2010-12-27 14:00:00		3	0.333333333333333	1.48	Anomalous	Not Anomalous
980	2010-12-27 16:00:00	2010-12-27 18:00:00		2	0.5	0.32	Anomalous	Not Anomalous
981	2010-12-27 16:00:00	2010-12-27 18:00:00	66.249.72.145	2	0.5	1.38	Anomalous	Not Anomalous

Time	IP	Status	Method	Path	Agent	Obj size	Country	ASN
2010-12-27 16:04:28	66.249.72.145	404	GET	/robots.txt	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)	301	United States	Google Inc.
2010-12-27 16:04:28	66.249.72.145	200	GET	/?show=login	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)	1111	United States	Google Inc.

982	2010-12-27 16:00:00	2010-12-27 18:00:00		1	1	0.4	Anomalous	Not Anomalous
983	2010-12-27 22:00:00	2010-12-28 00:00:00		1	1	0.37	Anomalous	Not Anomalous
984	2010-12-28 02:00:00	2010-12-28 04:00:00		2	0	2.55	Anomalous	Not Anomalous
985	2010-12-28 02:00:00	2010-12-28 04:00:00		4	0	3.93	Anomalous	Not Anomalous
986	2010-12-28 10:00:00	2010-12-28 12:00:00		3	0.333333333333333	1.48	Anomalous	Not Anomalous
987	2010-12-28 14:00:00	2010-12-28 16:00:00		3	0.333333333333333	1.48	Anomalous	Not Anomalous
988	2010-12-28 14:00:00	2010-12-28 16:00:00		2	0	1.1	Anomalous	Not Anomalous

Slika 8: Pregled događaja koji pripadaju određenoj sesiji

Provjeru i eventualni ispravak označenih sesija dodatno je olakšao pregled putem Logminer aplikacije, tako za svaku sesiju postoji pregled događaja nastalih za vrijeme sesije. Na slici 8 prikazani su događaji koji pripadaju sesiji pod rednim brojem 981, događaji su generirani od strane Googleovog spidera koji indeksira web sjedišta. Nakon što označimo skup korištenjem interkvartilnog raspona, ekspert ručno provjera točnost oznaka. Preduvjet je da ekspert posjeduje vještinu da po različitim parametrima (primjerice putanja ili vrsta web preglednika) odluči o kojem tipu sesije se radi.

5.2 Korištene metode dubinske analize podataka

Na temelju izabranih značajki (iz p.5.1.1) prikupljeni skup smo testirali sljedećim algoritmima strojnog učenja:

- **Naivni Bayes** je jednostavni probabilistički klasifikator koji koristi Bayesov teorem i oslanja se na pretpostavku neovisnosti između značajki [17]. Učenje se provodi nad skupom koji se sastoji od značajki $\{a_1, a_2, \cdots, a_n\}$ i označen je ciljem klasifikacije koji poprima vrijednosti iz skupa V. Nakon faze učenja na skupu za učenje naučeni koncept predviđa klasifikaciju novih instanci. Nad danim značajkama $a_1, a_2, \cdots, a_n$ računa se najvjerojatnija vrijednost v_{MAP}:

  $$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j | a_1, a_2, \cdots, a_n)$$

  Možemo primijetiti kako se gornji izraz oslanja na Bayesov teorem:

  $$P(v_j | a_1, a_2, \cdots, a_n) = \frac{P(a_1, a_2, \ldots, a_n | v_j) P(v_j)}{P(a_1, a_2, \cdots, a_n)}$$

  Zato što se naivni Bayes oslanja na neovisnost između značajki u odnosu na klasifikacijsku vrijednost, izbacuje se nazivnik odnosno umnožak vjerojatnosti a_i. Izraz možemo zapisati i kao:

  $$v_{NaiveBayes} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i | v_i) \tag{1}$$

  Učenje naivnog Bayesa uključuje računanje vjerojatnosti pojavljivanja $P(v_j)$ u odnosu na dane značajke $P(a_i|v_j)$. Skup ovih procjena čine naučenu hipotezu, koja se kasnije koristi za klasificiranje novih instanci na temelju formule (1).

- **Logistička regresija** dodjeljuje određenu vjerojatnost pojedinim klasifikacijskim vrijednostima [13]. Kako bi procijenila vjerojatnosti koristi logističku funkciju:

  $$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}} \tag{2}$$

  Prethodni izraz možemo zapisati kao:

  $$\frac{p(X)}{1 - p(X)} = e^{\beta_0 + \beta_1 X}$$

  Lijeva strana $p(X)/1 - p(X)$ naziva se još i izglednost (engl. odds) i može poprimiti vrijednosti od 0 do ∞. Logaritmiranje dobivamo logaritamsku izglednost ili logit :

$$log(\frac{p(X)}{1-p(X)}) = e^{\beta_0+\beta_1 X}$$

Parametre β_0 i β_1 dobivamo na način da maksimiziramo izglednost klasifikacije. Ovaj postupak se zove maksimizacija izglednosti (engl. maximum likelihood estimation) i formaliziran je funkcijom izglednosti:

$$l(\beta_0, \beta 1) = \prod_{i:y_i=1} p(x) \prod_{i':y'=0} (1 - p(x_{i'}))$$

Procijenjeni parametri β_0 i β_1 trebaju maksimizirati funkciju izglednosti, parametri koji to zadovoljavaju uvrštavaju se u izraz (2). Na temelju naučene granice dodjeljuju se klase klasifikacijskog problema.

- **Stablo odlučivanja (algoritam C 4.5)** je nadogradnja klasičnog algoritma ID3, oba algoritma rezultat su istraživanja Rossa Quinlana [17]. C 4.5 prvo stvara od skupa za učenje prenaučeno (redundantno) stablo, u slučaju testiranju sa sličnim podacima (onim iz skupa za učenje) klasifikator ima dobre rezultate te loše rezultate ukoliko testiramo na neovisnom validacijskom skupu.

  Stablo stvara Ako/Onda pravila te računa koji uvjeti daju najbolju točnost pri klasifikaciji, na način da se AKO preduvjeti izbacuju ukoliko oni ne narušuju točnost klasifikacije.

  Podrezivanje se događa od listova prema korijenu stabla te je temeljno na pesimističnoj procjeni grešaka, greške se odnose na postotak krivo klasificiranih slučajeva u skupu za učenje. Na temelju razlike točnosti pravila i standardne devijacije uzete iz binomne distribucije uzima se određeni gornji limit povjerenja (engl. confidence) koji najčešće iznosi 0.25, na temelju kojeg se pod stabla podrezuju.
  Prednosti algoritma C4.5 za indukciju stabla odlučivanja naspram njegovog prethodnika ID3 jesu sljedeće [17]:

  – C4.5 može koristiti atribute s numeričkim vrijednostima (kontinuirane vrijednosti) osim diskretnih značajki. Kontinuirani atributi se pretvaraju u skup diskretnih vrijednost koji dijele cjelokupni skup na temelju pragova.
  – C4.5 stablo podrezuje nakon izgradnje;kako bi izbacio suvišna pod stabla. Također, ovom metodom se smanjuje pretreniranost klasifikatora.

  - C4.5 funkcionira s nedostajućim vrijednostima
  - C4.5 omogućuje normalizaciju značajki s različitim težinskim faktorima.

- **Neuronska mreža** je metoda koje se koristiti za procjenu diskretnih vrijednosti i kontinuiranih vrijednosti [17].
  Neuronska mreža se može sastojati od više slojeva, s kojima se za razliku od njihova prethodnika perceptora omogućuje raščlanjivanje i nelinearnih odluka. Ulazne vrijednosti (x_i) se uz pomoć određene funkcije (najčešće logistička ili tanh funkcija) pretvaraju u izlazne vrijednosti o_u. Funkcija tanges hiperbolni se primjenjuje na linearnu kombinaciju ulaznih vrijednosti i nasumično postavljenih težina w_i u svakom sakrivenom čvoru:

  $$o_u = tanH(\sum_{i=0}^{n} w_i x_i)$$

  Postupkom ulančavanjem unatrag (engl. backpropagation) za svaki izlaz k računamo grešku δ_k:

  $$\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$$

  Greška za izlazne slojeve δ_k služi za računanje greške za sakrivene slojeve δ_h:

  $$\delta_h \leftarrow o_h(1 - o_h) \sum_{k \in iz.čv.} w_{kh} \delta_k$$

  Greška za sakrive slojeve δh uz stopu učenja η služi da izračunamo korekcijsko pravilo težina prilikom gradijentalnog spusta:

  $$\Delta w_{ji} = \eta \delta_j x_{ji}$$

  Korekcijsko pravilo ažurira težinu sakrivenog čvora:

  $$w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$$

  Postupak se provodi iterativno i završava nakon fiksnog broja iteracija ili nakon što se greška manja od neke postavljene vrijednosti.

  Ovakav postupak ulančavanja koristi se metodom stohastičkog gradijentalnog spusta, i možemo ga promatrati kao minimizaciju greške između vrijednosti koji daju izlazni slojevi o_k izračunati unaprijed i stvarne vrijednosti korištene iz skupa za učenje t_k:

  $$min\frac{1}{2} \sum_{k \in iz.čv.} (t_k - o_k)^2$$

Nakon završetka ulančavanja, neuronska mreža je sprema za klasifikaciju novih instanci.

- **Metoda k-najbližih susjeda** smješta sve instance u n-dimenzionalni prostor $\mathbb{R}^n$. Udaljenost između instanci se računa pomoću klasične Euklidske udaljenost:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n}(a_r(x_i) - a_r(x_j))^2}$$

Gdje je a_r skup značajki $a_1(c), \cdots, a_n(x)$.

Iz skupa za učenje gledaju se k najbližih instanci u odnosu na izabrani x_q, za $k = 1$ dodjeljuje se ista klasifikacija kao i najbližoj instanci x_q koja se nalazi u skupu za učenje.
Za veći broj instanci npr. $k = 3$ dodijeljuje se klasifikacija koju ima većina izabranih k instanci.

- **Metoda potpornih vektora (engl. Support Vector Machine)** predstavlja algoritam koji pronalazi najveću marginu razdvajanja između klasa, pod marginom podrazumijevamo udaljenosti između kritičkih točaka koje su najbliže plohi razdvajanja.
Primjere najbliže plohi nazivamo potpornim vektorima, marginu M možemo gledati kao širinu između plohe razdvajanja (prikazano na slici 9).
Računanje potpornih vektora je optimizacijski problem [13]:

$$max_{\beta 0,..,\beta n,\xi 2,..,\xi n} M$$

u odnosu na

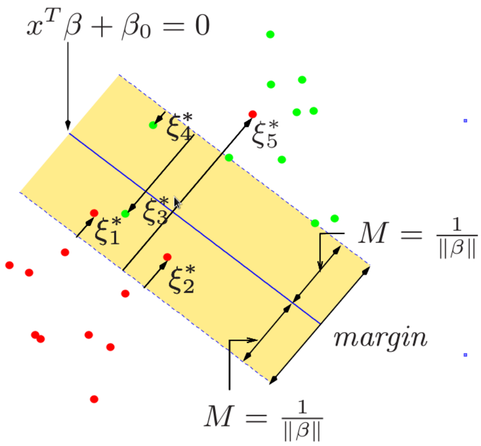
$$\sum_{j=1}^{n} \beta_j^2 = 1$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \xi_i)$$

uz uvjete

$$\xi_i \geq 0, \sum_{i=1}^{n} \xi_i \leq C$$

ξ_i nam pokazuje na kojoj strani margine je opservacija i, za $\xi_i = 0$ opservacija je na ispravnoj strani dok za $\xi i > 0$ je na krivoj strani. Vrijednost $\xi_i > 1$ na govori da se opservacija nalazi na krivoj strani hiperravnine. Parametar C nam govori o broju opservacija koje su na krivoj strani margine ili hiper ravnine, C se izabire u postupku unakrsne validacije i njime se kontrolira odnos između pristranosti i varijance (engl. bias-variance tradeoff).



Slika 9: Klasifikacija potpornim vektorima za slučaj preklapanja - margina je označena s M dok ξ_j^* označava instance na krivoj strani margine (preuzeto iz [9])

Trik koji se koristi kod SVM-a je korištenje različitih jezgrenih funkcija, koje neseparabilan ili neodgovarajući problem premještaju u višu dimenziju gdje je problem pogodniji za rješavanje. U našem slučaju korištena je linearna jezgrena funkcija oblika $K(x_i, x_{i'}) = \sum_{j=1}^{n} x_{ij}x_{i'j}$. Jezgrena funkcija je unutarnji produkt koji kvantificira sličnost između dvije opservacije (x_i i $x_{i'}$) [13]. Kod linearne jezgrene funkcije on kvantificira sličnost koristeći Pearsonovu korelaciju. Također, moguće je koristiti polinomnu jezgrenu funkciju oblika:

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{n} x_{ij}x_{i'j})^d, d > 1$$

ili radijalnu jezgrenu funkciju:

$$K(x_i, x_{i'}) = exp(-\gamma \sum_{j=1}^{n} (x_{ij}x_{i'j})^2), \gamma > 0$$

- **Nasumične šume (engl. Random Forest)** koriste veći broj modela stabla odlučivanja kako bi smanjili varijancu, metoda je slična metodi *bagging* od koje se razlikuje po postupku dekorelacije. Skup za učenje B se podijeli na nekoliko skupova, od svakog skupa se stvori model stabla odlučivanja. Pošto je broj podataka u skupu za učenje ponekad ograničen, koristi se metoda nasumičnog uzrokovanja prilikom izbora podskupova. Kako bi se smanjila varijanca konačni model je aritmetička sredina svih naučenih modela. U RF svako stablo se izgrađuje s različitim bootstrap uzorkom iz početnog skupa za učenje, trećina instanci se ostavljaja za testiranje izgrađenih modela.

Nasumične šume se razlikuju od *bagginga* u tome što algoritam vodi računa da koristi samo određeni broj nasumično izabranih značajki (najčešće $\sqrt{p}$) prilikom račvanja čvorova u stablu, izabrane značajke moraju se razlikovati u velikoj mjeri tj. imati veliku varijancu. Ovaj postupak se još naziva dekorelacija [13], koja izbjegavanjem koreliranih instanci smanjuje prenaučenost (engl. overfitting) modela i izabire optimale značajke koje se koriste u klasifikaciji. Iz podskupa se izabire značajka koja omogućuje najbolje račvanje trenutnog čvora u stablu, odnosno ima najveću vrijednost Gini indeksa.

5.3 Evaluacija isprobanih modela

Označeni skup sadržava trećinu anomalnih događaja, dok ostatak pripada normalnim događajima. Za testiranje uspješnosti klasifikacije korištena je metoda k-dijelne unakrsne validacije (engl. k-fold cross validation). Za naše potrebe korišteno je k=10 dijelova, Witten [23] navodi kako je k=10 dijelova optimalan broj za estimaciju greške.

Unakrsna validacija (k=10) dani skup dijeli na 10 jednakih dijelova ili particija, instance u dijelovima skupa biraju se nasumično. Postupak se ponavlja 10 puta, u svakom koraku jedan dio skupa izuzima se i koristi za testiranje naučenog koncepta dok ostali dijelovi ($k-1=9$) se koriste za učenje koncepta. Na izuzetom dijelu (eng. holdout) kojim se testira model računa se greška, nakon svih koraka (u našem slučaju 10 koraka) računa se srednja kvadratna greška (MSE) za regresijske probleme ili prosječna vrijednost krivo klasificiranih instanci za klasifikacijske probleme. Na ovaj način svaka instanca skupa se koristi za testiranje te za učenje koncepta.

Rezultati se nalaze u tablici 3. Prilikom evaluacije anomalnih događaja vrlo važno je obratiti pažnju na F-vrijednost. F-vrijednost jednaka je:

$$F - vrijednost = \frac{2TP}{2TP + FP + FN}$$

F-vrijednost uzima u obzir lažne pozitive i lažne negative, tj. krive klasifikacije. AUC (engl. Area Under the Curve) predstavlja površinu ispod ROC krivulje, kod ove mjere veća površina predstavlja bolju klasifikacijsku moć.

Tablica 3: Rezultati klasifikacije pomoću unakrsne validacije (k=10)

Algoritam	*Točnost %*	*TPR (N)*	*FPR (N)*	*F-mjera (N)*	*AUC (N)*	*TPR (A)*	*FPR (A)*	*F-mjera (A)*	*AUC (A)*
Naivni Bayes	77.68	0.70	0.06	0.81	0.85	0.94	0.30	0.73	0.85
C 4.5	84.13	0.89	0.25	0.88	0.89	0.75	0.11	0.75	0.89
Logistička regresija	81.33	0.75	0.06	0.84	0.86	0.94	0.25	0.77	0.86
Neuronska mreža	81.28	0.75	0.06	0.84	0.85	0.94	0.25	0.77	0.85
kNN	67.77	0.55	0.05	0.70	0.75	0.95	0.45	0.66	0.75
SVM	81.28	0.75	0.05	0.84	0.85	0.95	0.25	0.77	0.85
Random Forest	84.18	0.89	0.25	0.88	0.91	0.75	0.11	0.76	0.91

Napomena: N - normalne sesije, A - anomalne sesije

Za naš problem vrlo je važan broj ispravnih klasifikacija za anomalne događaje. Omjer TPR za anomalije treba biti što veći, dok visoki FPR za anomalije će generirati dodatni posao ljudima koji budu pratili sustav. Malo viši FPR kod anomalija nije poguban iz razloga što ekspert koji prati sustav moći će s lakoćom uočiti normalni događaj koji je u ovom slučaju krivo klasificiran. Razumljivo je promatramo li FPR u kontekstu normalnih događaja, važno je da taj omjer bude što manji.

Ukoliko obratimo pozornost na rezultate klasifikacije anomalnih događaja, gledajući F-vrijednosti, uočavamo da su svi algoritmi podjednaki izuzev metode k-najbližih susjeda (kNN, F-vrijednost=0,66).

Metode stabla odlučivanja (C 4.5 i RF) se izdvajaju po površini ispod krivulje 0,85 odnosno 0,91. Metode C 4.5 i RF imaju nisku razinu FPR (0,11) ali i dosta nisku razinu klasifikacije pozitiva TPR=0,75 za anomalije. Metode logističke regresije, neuronskih mreža i potpornih vektora imaju dobar TPR, ali veći FPR (0,25) za anomalne događaje.

Pogledamo li klasifikaciju normalnih događaja, vidimo da k-NN i Naivni Bayes imaju najlošije rezultate. U ovom slučaju situacija je obrnuta, u odnosu na klasifikaciju anomalnih događaja. C 4.5 i RF imaju dobru razinu razvrstavanja pravih normalnih događaja ali uz visoku razinu lažnih dojava.

Iz razloga što FPR kod normalnih događaja su zapravo u stvarnosti anomalije, preporučljivo bi bilo izabrati jednu od metoda: potpornih vektora (SVM), neuronske mreže ili logističke regresije.

5.4 Izbor značajki

Izbor značajki ili atributa je proces izbora relevantnog podskupa značajki za izgradnju modela. Izbor značajki se razlikuje od ekstrakcije (generiranja) značajki, koji predstavlja proces kojim se na temelju izvornih značajki stvaraju nove izvedene značajke.
Također, izbor značajki predstavlja heuristički postupak za smanjenje dimenzionalnosti problema, čiji je cilj izbor minimalnog podskupa značajki koji čuvaju dovoljno informacija za kvalitetnu predikciju. Postoji nekoliko metoda izbora značajki [20]:

Metode omotača izgrađene su na temelju nekog poznatog predikcijskog modela. Svaki podskup značajki se koristi za učenje danog modela, generalizacijska sposobnost daje rang podskupovima značajki. Iako su zahtjeve za računanje, metode omotača često daju najbolje rezultate.

Metode filtriranja koriste određenu mjeru umjesto greške klasifikacije za rangiranje značajki. Neki od češće korištenih mjera su uzajamna informacija (engl. mutual information) i korelacija. Za razliku od metoda omotača, filter metode najčešće rangiraju pojedinačno značajke.

Metode ugradnje (engl. Embedded) izabiru značajke prilikom izgradnje modela. Primjer ove metode je LASSO koja služi za izgradnju linearnog modela prilikom izgradnje penaliziraju se regresijski koeficijenti na način da se neke značajke mogu izbaciti iz modela.

Metode omotača i filtra možemo kombinirati i to na dva načina: odozgora prema dolje ili obrnuto od dolje prema gore. Načinom od dolje prema gore podskup značaki gradimo pohlepno, prvo umećemo visoko rangirane značajke pri čemu se kontrolira greška prilikom redukcije na validacijskom skupu. Heuristički postupak prestaje nakon što se greška na validacijskom skupu prestane smanjivati. Pristupom odozgora prema dolje započinje se punim skupom te se postepeno izbacuju značajke pri čemu se traži optimalna klasifikacija ili regresija uz kontroliranje pogreške na validacijskom skupu.

Prilikom korištenja metoda filtriranja moguće je narušiti uzajamnu informaciju između značajki. Iako značajke mogu biti neovisne, često se događa da dvije značajke daju dobre rezultate prilikom klasifikacije. Izbacivanjem takvih značajki narušavamo tu informaciju i samim time predikcijsku sposobnost modela.

Metoda Relief predložena je od Kira i Rendella [14], služi za izbor statistički relevantnih značajki, otporna je na šum u podacima i međuovisnost značajki. Značajke se vrednuju na način da se nasumično uzimaju uzorci instanci iz danog skupa te uzimaju najbliži susjedi koji pripadaju nekoj od klasa. Ukoliko su susjedi podudaraju s uzrokovanim instancama težinski faktor raste, suprotno tome u slučajnu najbližih različitih klasa težinski faktor pada.

Informacijski dobitak (engl InfoGain) ili metoda uzajamne informacije rangira značajke na temelju izračunatog informacijskog dobitka u odnosu na klasifikacijsku klasu, numeričke značajke se prvo diskretiziraju.

Računa se na način:

$$InfoGain(Klase, Značajka) = Entropija(Klase) - Entropija(Klase|Značajka)$$

gdje je entropija $H(klasa|značajka)$:

$$H(Klase|Značajka) = -\sum_{k \in Klasa} P(k|Značajka) log P(k|Značajka)$$

odnosno entropija klasa $H(Klasa)$:

$$H(Klase) = -\sum_{k \in Klase} P(k) log P(k)$$

Metoda RelieF najslabije rangira značajke *bandwith* i *stddev_btwn_reqs*, dok metoda informacijskog dobitka InfoGain najslabije rangira značajke: *stddev_btwn_reqs* i *session_duration*. Rezultati metoda prikazani su u poglavlju 7.

Metodom omotača za algoritme logističke regresije, SVM i neuronske mreže dobili smo da optimalni podskup sadrži značajke: *error_rate* i *session_duration*.

5.5 Evaluacija s reduciranim skupom značajki

Nakon što smo sve značajke u evaluirali različitim algoritmima strojnog učenja, u ovom djelu prikazat ćemo rezultate za evaluaciju algoritama s izabranim podskupovima značajki.
U tablici 4 prikazani su rezultati algoritmima koji su imali najbolju klasifikaciju s obzirom na cjeloviti skup.

Za algoritme logističke regresije, neuronske mreže i metode potpornih vektora korišten je sljedeći podskup značajki, koje su izabrane metodom omotača:

- **error_rate** - postotak upita koji je rezultirao greškom (HTTP status kod se nalazio u rasponu 400 do 500)
- **session_duration** - vremensko trajanje sesije korisnika (u sekundama

Za metode stabla odlučivanja (Random Forest i C 4.5) značajke smo izabrali metodom filtra, koristeći pritom postupak informacijskog dobitka (InfoGain):

- **nonget_rate** - postotak upita koji nije poslan HTTP-om GET metodom
- **error_rate** - postotak upita koji je rezultirao greškom (HTTP status kod se nalazio u rasponu 400 do 500)
- **num_requests** - ukupni broj upita u sesiji
- **bandwith** - ukupan promet u sesiji

Metode logističke regresije, neuronske mreže i metode potpornih vektora izabrane dvije značajke postižu slične rezultate kao i evaluacija s cijelim skupom značajki. FPR za predikciju anomalija ostao je isti 0.25.
Izabrane značajke metodom filtra, metoda C4.5 i Random Forest, nešto su bolje klasificirale normalne događaje (83.8 i 84.28). Anomalni događaji su klasificirani slabije nego s cijelim skupom značajki (0.75 i 0.76) sa sličnom količinom lažnih pozitiva 0.12 i 0.11.
Ono što nije dobro kod metoda koja koriste stabla odlučivanja je visoka razina lažnih događaja kod detekcije normalnih događaja, oko 0.25 za razliku od 0.06 kod metoda kojima smo značajke izabrali metodom omotača. Prema tome veća je mogućnost da nam promaknu anomalni događaja ukoliko koristimo metodu C 4.5 ili metodu Random Forest, pa bi bilo preporučljivije koristiti logističku regresiju, neuronsku mrežu ili SVM.

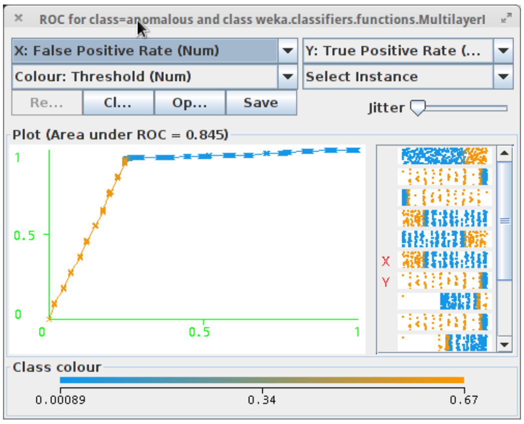
Iako naš broj atributa nije velik, metode poput neuronskih mređa i potpornih vektora zbog posjeduju veliku vremensku složenost. Pogotovo ako npr. kod SVM korisitmo neku od ne lineranih jezgrenih funkcija. Izborom atributa metodom omotača dobit ćemo optimalan skup atributa koji će biti reduciran i uštedjeti će vrijeme prilikom faze učenja odnosno stvaranja modela.

Na slici 10 možemo vidjeti ROC krivulje za evaluirane algoritme za klasu anomalnih događaja. Na slici su ispravne klasifikacije za anomalne događaje naznačene narančastom bojom, dok neispravne plavom bojom. Kod stabla odlučivanja (slika 10d vidimo da uz vidimo dobru klasifikaciju anomalnih događaja uz vrlo nizak broj lažnih pozitiva (iz tablice 0.12). Slično je i s metodom Random forest (slika 10e) koja ima niži broj lažnih pozitiva (iz tablice 0.11).
Neuronska mreža, logistička regresija i SVM (slike 10a do 10c) imaju vrlo dobru klasifikaciju anomalnih događaja (0.94) uz relativno visok broj lažnih pozitiva (iz tablice 0.25).

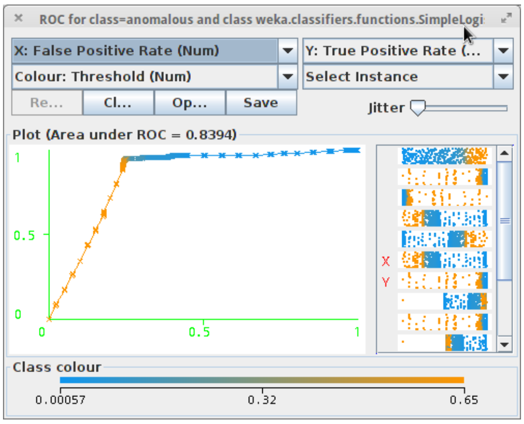
Tablica 4: Rezultati klasifikacije reduciranog skupa značajki, za logističku regresiju, SVM i neuronsku mrežu korišteni su atributi izabrani metodom omotača. Za metoda stabla odlučivanja korištene su značajke dobivene pomoću informacijskog dobitka. Za učenje korištena je metoda unakrsne validacije (k=10)

Algoritam	*Točnost %*	*TPR (N)*	*FPR (N)*	*F-mjera (N)*	*AUC (N)*	*TPR (A)*	*FPR (A)*	*F-mjera (A)*	*AUC (A)*
Logistička regresija	81.33	0.75	0.06	0.84	0.84	0.94	0.25	0.77	0.84
Neuronska mreža	81.38	0.75	0.06	0.84	0.84	0.94	0.25	0.77	0.84
SVM	81.28	0.75	0.05	0.84	0.85	0.95	0.25	0.77	0.85
C 4.5	83.8	0.88	0.24	0.88	0.89	0.76	0.12	0.75	0.89
Random Forest	84.28	0.89	0.25	0.88	0.91	0.75	0.11	0.76	0.91

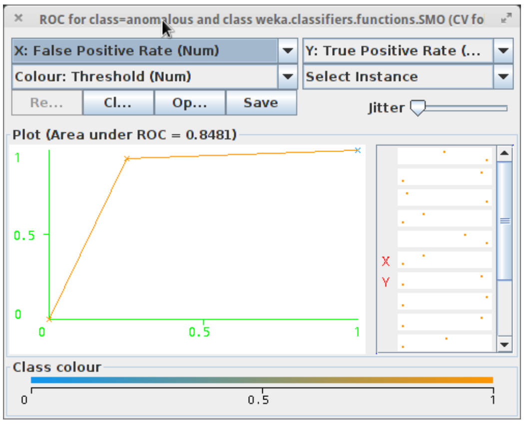
Napomena: N - normalne sesije, A - anomalne sesije



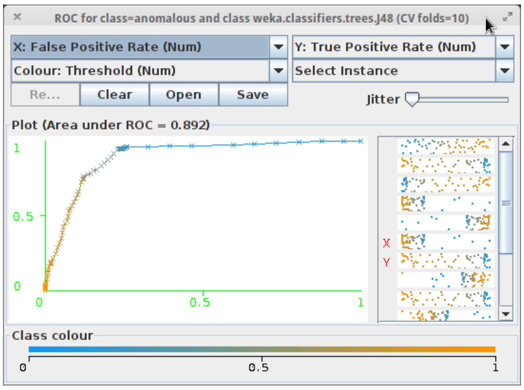
(a) Neuronska mreža



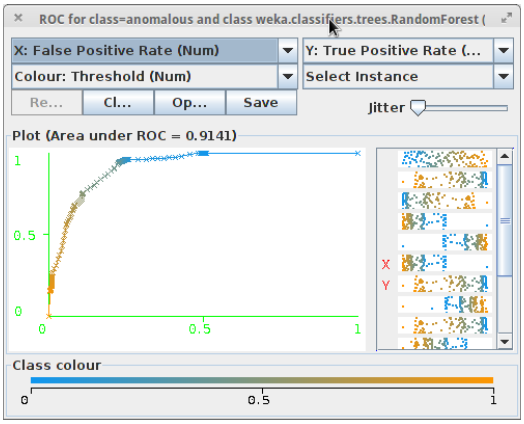
(b) Logistička regresija



(c) Metoda potpornih vektora (SVM)



(d) Stablo odlučivanja C4.5



(e) Random Forest

Slika 10: ROC krivulje za evaluirane algoritme

6 Zaključak

Metode strojnog učenja pokazale su se kao efikasno rješenje za detekciju anomalnog ponašanja u dnevniku pristupa promatranog poslužitelja. U našem slučaju SVM, neuronska mreža i logistička regresija imale su najbolju klasifikaciju. Važno je napomenuti da izbor značajki uvelike pomaže prilikom brzine izvršavanja neuronske mreže i SVM-a.

Glavni nedostatak ovakvog pristupa detekciji je taj što naučeni model odnosno koncept postaje nevaljan nakon nekog vremena. U literaturi ovaj problem je još poznat kao i *skretanje koncepta* (engl. concept drift). Svakako bi bilo dobro pronaći način kako detektirati nestabilnost modela i ponovno naučiti koncept nad podacima s novim karakteristikama.

Drugi problem je označavanje skupa, koje zbog količine podataka postaje komplicirano za izvesti ručno. Na ovom polju također postoje metode djelomično nadziranog učenja (engl. semi-supervised learning) koje uz male količine označenih podataka mogu za učenje koristiti i neoznačene podatke.

U radu je problem detekcije anomalija promatran kao klasifikacijski problem. U rješavanju ovog problema korisne su i metode koje koriste procjenu gustoće distribucije, koje promatraju gdje se pojavljuju neočekivane vrijednosti (engl. outliers). Kod takvih metoda promatraju se instance gdje je gustoća niska i koje u nekoj mjeri odskaču od same distribucije. Problem kod ovih metoda su podaci koji pripadaju raznim gustoćama.

Popularna metoda kod ovog pristupa je metoda lokalnog faktora neočekivane vrijednosti (engl. local outlier factor) [5]. Za svaku instancu se računa omjer prosječne lokalne gustoće K najbližih susjeda instance i lokalnu gustoće same instance. Lokalna gustoća se računa tako da se prvo nađe radius hiper sfere čiji je centar u promatranoj instanci i obuhvaća K instanci. Lokalna gustoća je tako rezultat volumna hiper sfere podijeljenog s K instanci. Razumljivo je da za normalne instance gustoća će biti veća i slična susjednim instancama, dok će za anomalne instance biti manja. Više o samom LOF algoritmu može se naći u [4].

Kako vidimo postoji mnogo prepreka ali i mogućnosti za rješavanje ovog problema. Napredak u ovom području predstavlja kombinacija nenadziranog i nadziranog učenja, s kojom se može kvalitetno i automatizirano označiti skup i koristiti ga prilikom nadziranog učenja. Nenadzirane tehnike, npr. temeljene na procjeni gustoće distribucije, mogu efikasno detektirati odskakanja od normalnog ponašanja. Također takvim tehnikama može se postići otpornost na problem nevaljanosti koncepta kroz vrijeme.

7 Dodatak 1 - Izbor značajki

```
=== Attribute Selection on all input data ===

Search Method:
Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 7 anomaly_
automatic):
ReliefF Ranking Filter
Instances sampled: all
Number of nearest neighbours (k): 10
Equal influence nearest neighbours
Ranked attributes:
0.03421 4 nonget_rate
0.01563 2 error_rate
0.00982 5 session_duration
0.00889 1 num_requests
0.00846 3 bandwith
0.00521 6 stddev_btwn_reqs

Selected attributes: 4,2,5,1,3,6 : 6
```

```
== Attribute Selection on all input data ===

Search Method:
Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 7 anomaly_
automatic):
Information Gain Ranking Filter

Ranked attributes:
0.3529 2 error_rate
0.1966 3 bandwith
0.1351 1 num_requests
0.0781 4 nonget_rate
```

```
0.0554 6 stddev_btwn_reqs
0.0419 5 session_duration

Selected attributes: 2,3,1,4,6,5 : 6
```

```
=== Attribute Selection on all input data ===

Search Method:
Best first.
Start set: no attributes
Search direction: forward
Stale search after 5 node expansions
Total number of subsets evaluated: 24
Merit of best subset found: 0.189

Attribute Subset Evaluator (supervised, Class (nominal): 7
anomaly_automatic):
Wrapper Subset Evaluator
Learning scheme: weka.classifiers.functions.SMO
Scheme options: -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K
weka.classifiers.functions.supportVector.PolyKernel -C 250007 -
E
1.0
Accuracy estimation: classification error
Number of folds for accuracy estimation: 5

Selected attributes: 2,5 : 2
error_rate
session_duration
```

```
=== Attribute Selection on all input data ===

Search Method:
Best first.
Start set: no attributes
Search direction: forward
Stale search after 5 node expansions
```

```
Total number of subsets evaluated: 22
Merit of best subset found: 0.186

Attribute Subset Evaluator (supervised, Class (nominal): 7
anomaly_automatic):
Wrapper Subset Evaluator
Learning scheme: weka.classifiers.functions.Logistic
Scheme options: -R 1.0E-8 -M -1
Accuracy estimation: classification error
Number of folds for accuracy estimation: 5

Selected attributes: 2,5 : 2
error_rate
session_duration
```

Literatura

[1] E. G. Amoroso. Intrusion detection. Technical report, 1999.

[2] N. Athanasiades, R. Abler, J. Levine, H. Owen, and G. Riley. Intrusion detection testing and benchmarking methodologies. In *Information Assurance, 2003. IWIAS 2003. Proceedings. First IEEE International Workshop on*, pages 63–72, 2003.

[3] D. Bolzoni. *Revisiting Anomaly-based Network Intrusion Detection Systems*. PhD thesis, CTIT PhD Thesis Series Number 09-147, Enschede, 2009.

[4] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *ACM Sigmod Record*, volume 29, pages 93–104. ACM, 2000.

[5] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.

[6] M. Crosbie and G. Spafford. *Active Defense of a Computer System Using Autonomous Agents*. 1995.

[7] Dorothy E. Denning. An intrusion-detection model. *IEEE Trans. Softw. Eng.*, 13(2):222–232, February 1987.

[8] Gustavo Miguel Barroso Assis do Nascimento. Anomaly detection of web-based attack, 2010.

[9] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., 2009.

[10] R. Heady, G. Luger, A. Maccabe, and M. Servilla. The architecture of a network level intrusion detection system. Technical report, LA-SUB–93-219, Los Alamos National Lab., NM (United States); New Mexico Univ., Albuquerque, NM (United States). Dept. of Computer Science, 1990.

[11] Apache httpd Documentation. Access logs, pristupano 30.10.2013.

[12] Kenneth L. Ingham and Hajime Inoue. Comparing anomaly detection techniques for http. In *Proceedings of the 10th international conference on Recent advances in intrusion detection*, RAID'07, pages 42–62, Berlin, Heidelberg, 2007. Springer-Verlag.

[13] G. James, T. Hastie, D. Witten, and R. Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Texts in Statistics. Springer London, Limited, 2013.

[14] Kenji Kira and Larry A Rendell. The feature selection problem: Traditional methods and a new algorithm. In *AAAI*, pages 129–134, 1992.

[15] Christopher Kruegel and Giovanni Vigna. Anomaly detection of web-based attacks. In *Proceedings of the 10th ACM conference on Computer and communications security*, CCS '03, pages 251–261, New York, NY, USA, 2003. ACM.

[16] R. A. Maxion and R. R. Roberts. Proper Use of ROC Curves in Intrusion/Anomaly Detection. Technical Report CS-TR-871, School of Computing Science, University of Newcastle upon Tyne, November 2004.

[17] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[18] Netcraft.com. Web server survey - august 2013, pristupano 10.9.2013.

[19] Vladimir Šošić, Ivan Serdar. *Uvod u statistiku*. 1994.

[20] Mauro Brunato Roberto Battiti. *The LION Way: Machine Learning plus Intelligent Optimization*. Lionsolver, Inc., 2013.

[21] Speedguide.net. Commonly open ports, pristupano 10.9.2013.

[22] Dusan Stevanovic, Natalija Vlajic, and Aijun An. Unsupervised clustering of web sessions to detect malicious and non-malicious website users. *Procedia Computer Science*, 5(0):123 – 131, 2011. The 2nd International Conference on Ambient Systems, Networks and Technologies (ANT-2011) / The 8th International Conference on Mobile Web Information Systems (MobiWIS 2011).

[23] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.