Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Pronalaženje skrivenog znanja (MS1PSZ)
Nastavnik: Prof. dr Veljko Milutinović
Asistent: Bojan Furlan
Školska godina: 2011/2012.

Projekat za domaći rad
- Projektni zadatak 1 –

Napomena

Pročitati tekst zadatka **u celini i pažljivo**, pre započinjanja realizacije ili traženja pomoći. Ukoliko u zadatku nešto nije dovoljno precizno definisano ili su postavljeni kontradiktorni zahtevi, student treba da uvede razumne pretpostavke, da ih temeljno obrazloži i da nastavi da izgrađuje preostali deo svog rešenja na temeljima uvedenih pretpostavki. Zahtevi su namerno nedovoljno detaljni, jer se od studenata očekuje kreativnost i profesionalni pristup u rešavanju praktičnih problema.

Tekst zadatka:

Sistem za detekciju upada (*Eng. Intrusion Detection System - IDS*) je softverski ili hardverski proizvod koji prati događaje i nadgleda stanje unutar računarskog sistema ili mreže. Glavni cilj ovakvih sistema je analiza i detekcija neodgovarajućih ili nepravilnih akcija od strane korisnika koji se mogu nalaziti kako izvan, tako i unutar samog sistema. Zadatak učenja/treniranja IDS-a ogleda se u izgradnji prediktivnog modela (klasifikatora) koji može napraviti razliku između „loših" konekcija (upada ili napada) i „dobrih" normalnih konekcija.

Opis podataka

Dostavljeni podaci za treniranje modela sastoje se od skupa različitih simuliranih napada unutar tipičnog vojnog mrežnog okruženja (U.S. Air Force LAN), prikupljanih u okviru DARPA programa za evaluaciju IDS sistema. Svi podaci su predstavljeni u formatu TCP otisaka (*Eng. TCP dump*) gde svaka konekcija predstavlja sekvencu TCP paketa koja počinje i završava se u tačno definisanom vremenskom intervalu. Unutar ovog intervala razmenjuju se podaci u smeru od izvorišne ka odredišnoj IP adresi i suprotno, pomoću nekog od unapred definisanih protokola. Pri tom, svaka konekcija je obeležena labelom ili kao normalna ili kao napad, sa tačno definisanim tipom napada.

Svi napadi spadaju u 4 glavne kategorije:

- DOS: denial-of-service, npr. syn flood;
- R2L: unauthorized access from a remote machine, npr. guessing password;
- U2R: unauthorized access to local superuser (root) privileges, npr. različiti „buffer overflow" napadi;
- probing: surveillance and other probing, npr. port scanning.

Pored osnovnih atributa opisanih u Tabeli 1. definisani su i izvedeni atributi ustanovljeni na osnovu ekspertskog znanja iz ovog domena, koji mogu doprineti većoj pouzdanosti pri razlikovanju normalnih konekcija u odnosu na napade. Atributi koji se odnose na isti odredišni uređaj (*Eng. "same host"*) razmatraju samo konekcije koje imaju istu odredišnu IP adresu kao i trenutna konekcija, i predstavljaju statistiku sračunatu u protekle 2 sekunde vezano za dati protokol, servis, itd. Na isti način, atributi iz grupe *"same service"* odnose se na konekcije ostvarene u protekle 2 sekunde koje koriste isti servis kao i trenutna konekcija. Obe vrste ovih atributa predstavljaju vremensku analizu saobraćaja.

Neke vrste napada (npr. *probing attacks*) koriste znatno duže vremenske intervale od 2 sekunde, tako da neki atributi su kreirani na osnovu prozora od 100 konekcija prema odredištnom uređaju (*destination host*) umesto vremenskog prozora od 2 sekunde.

Da bi validacija modela bila realistična, skup podataka za validaciju nema istu raspodelu verovatnoće kao trening podaci. Takođe, validacioni podaci sadrže neke specifične vrste napada koje se ne nalaze u trening setu. S obzirom da većina novih, neispitanih napada predstavlja varijantu već poznatih, „potpis" može biti dovoljan za otkrivanje nove vrste napada.

Opis svakog atributa dat je u sledećim tabelama:

feature name	*description*	*type*
duration	length (number of seconds) of the connection	continuous
protocol_type	type of the protocol, e.g. tcp, udp, etc.	discrete
service	network service on the destination, e.g., http, telnet, etc.	discrete
src_bytes	number of data bytes from source to destination	continuous
dst_bytes	number of data bytes from destination to source	continuous
flag	normal or error status of the connection	discrete
land	1 if connection is from/to the same host/port; 0 otherwise	discrete
wrong_fragment	number of “wrong” fragments	continuous
urgent	number of urgent packets	continuous

Table 1: Basic features of individual TCP connections.

feature name	*description*	*type*
hot	number of “hot” indicators	continuous
num_failed_logins	number of failed login attempts	continuous
logged_in	1 if successfully logged in; 0 otherwise	discrete
num_compromised	number of “compromised” conditions	continuous
root_shell	1 if root shell is obtained; 0 otherwise	discrete
su_attempted	1 if “su root” command attempted; 0 otherwise	discrete

num_root	number of “root” accesses	continuous
num_file_creations	number of file creation operations	continuous
num_shells	number of shell prompts	continuous
num_access_files	number of operations on access control files	continuous
num_outbound_cmds	number of outbound commands in an ftp session	continuous
is_hot_login	1 if the login belongs to the “hot” list; 0 otherwise	discrete
is_guest_login	1 if the login is a “guest”login; 0 otherwise	discrete

Table 2: Content features within a connection suggested by domain knowledge.

feature name	*description*	*type*
count	number of connections to the same host as the current connection in the past two seconds	continuous
	Note: The following features refer to these same-host connections.	
serror_rate	% of connections that have “SYN” errors	continuous
rerror_rate	% of connections that have “REJ” errors	continuous
same_srv_rate	% of connections to the same service	continuous
diff_srv_rate	% of connections to different services	continuous
srv_count	number of connections to the same service as the current connection in the past two seconds	continuous
	Note: The following features refer to these same-service connections.	

srv_serror_rate	% of connections that have “SYN” errors	continuous
srv_rerror_rate	% of connections that have “REJ” errors	continuous
srv_diff_host_rate	% of connections to different hosts	continuous

Table 3: Traffic features computed using a two-second time window.

dst_host_count	number of connections to the same host	continuous
	Note: The following features are constructed using a window of 100 connections to the same host.	
dst_host_srv_count	number of connections to services	continuous
dst_host_same_srv	number of connections to the same service	continuous
dst_host_same_srv_rate	% of connections to the same service	continuous
dst_host_diff_srv_rate	% of connections to different services	continuous
dst_host_same_src_port_rate	% of connections from the same source port	continuous
dst_host_srv_diff_host_rate	% of connections to different hosts	continuous
dst_host_serror_rate	% of connections that have “SYN” errors	continuous
dst_host_srv_serror_rate	% of connections that have “SYN” errors (same service)	continuous
dst_host_rerror_rate	% of connections that have “REJ” errors	continuous
dst_host_srv_rerror_rate	% of connections that have “REJ”	continuous

	errors (same service)	

Table 4: Traffic features computed using 100 connections window.

Zadaci

Projekat koji izrađuje svaki student sastoji se iz zadataka opisanih u nastavku. Štampane materijale pripremiti prema uputstvima datim u zadacima, a sve zajedno na sledeći način:

1. Na naslovnoj strani jasno napisati naziv predmeta, prezime i ime studenta, broj indeksa i adresu e-pošte.
2. Sve zajedno čvrsto povezati u jednu celinu, tako da se listovi ne mogu rasipati (najbolje spiralom).

Zadatak 1 – ETL procedura (10 poena)

Implementirati SQL Server ETL proceduru koja učitava podatke iz zadate datoteke i transformiše ih na sledeći na način:

1. Iz fajla sa zadate lokacije (proizvoljne putanje) učitava podatke u tabelu baze podataka tako što, u zavisnosti od tipa atributa (*continuous* ili *discrete* - datom u tabelama u tekstu zadatka), mapira svaki atribut u odgovarajući tip podataka podržan od strane baze podataka (*integer, float*, *nvarchar, itd*.)
2. Transformiše podatke na sledeći način:
    a. Za svaki red dodeljuje jedinstveni primarni ključ.
    b. Na osnovu poslednjeg atributa *attack_type* formira dve dodatne kolone – *is_attack* i *attack_group*, gde *is_attack* označava da li dati red predstavlja napad (uzima vrednosti 0 ili 1), a *attack_group* uzima jednu od 4 vrednosti iz kategorije napada u koju taj napad spada i to po sledećoj tabeli:

back dos	multihop r2l	satan probe
buffer_overflow u2r	neptune dos	smurf dos
ftp_write r2l	nmap probe	spy r2l
guess_passwd r2l	perl u2r	teardrop dos
imap r2l	phf r2l	warezclient r2l
ipsweep probe	pod dos	warezmaster r2l
land dos	portsweep probe	
loadmodule u2r	rootkit u2r	

Napomene:

I. Ukoliko nije nijedan od navedenih tipova svrstati u grupu *other.*
II. Labela *normal* označava normalnu konekciju.

3. Pomoću implementirane procedure u bazi podataka učitati dve tabele koje odgovaraju *Training* i *Validation set* fajlovima.

Zadatak 2 – *k-Nearest Neighbors* (10 poena)

Napisati SQL upit/skript koji realizuje *k-Nearest Neighbors* algoritam:

1. Sve numeričke podatke normalizovati pomoću *Min-Max* normalizacije i smestiti u zasebne tabele.
2. Kao ciljne atribute uzeti *is_attack* i *attack_group* kolone, a za treniranje modela koristiti sve ostale atribute osim atributa *attack_type*.
3. Koristiti težinsko glasanje za k=3, a kao meru distance uzeti :
   i. Euklidsko rastojanje

$$d_{Euclidean}(x, y) = \sqrt{\sum_{i}^{n}(x_i - y_i)^2}$$

n – broj atributa (dimenzija vektora)

Za diskretne vrednosti rastojanje računati pomoću funkcije sličnosti:

$$d_{Similarity}(x_i, y_i) = \begin{cases} 0, & if\ x_i = y_i \\ 1, & otherwise \end{cases}$$

4. Evaluirati trenirani model na datom validacionom skupu podataka poredeći prediktovanu vrednost svakog atributa sa stvarnom vrednošću . Rezultate prikazati na *Lift Chart* grafikonu (pomoću Excel ili nekog drugog alata) za 30, 50, 70 i 100% populacije validacionih podataka, gde se na x-osi nalazi procenat ukupne populacije, a na y-osi procenat korektno prediktovanih vrednosti za ciljne atribute. Posebno iscrtati krivu za korektno prediktovanu vrednosti 0, 1 i ukupno, tj. obe vrednosti za atribut *is_attack*. Takođe, isto uraditi i za atribut *attack_group.*

Zadatak 3 – *k-Nearest Neighbors* (10 poena)

Napisati SQL upit/skript koji realizuje *k-Nearest Neighbors* algoritam:

1. Koristiti normalizovane podatke iz Zadatka 2.
2. Kao i u Zadatku 2 za ciljne atribute uzeti *is_attack* i *attack_group* kolone, a za treniranje modela koristiti sve ostale atribute osim atributa *attack_type*.
3. Koristiti težinsko glasanje za k=3, a kao meru distance uzeti :
   ii. Kosinusno rastojanje :

U slučaju da se ne poklapaju vrednosti atributa *protocol_type*, *service* i *flag*

$$d_{Cosine}(x, y) = 1$$

U suprotnom postaviti vrednosti ovih atributa 1 i računati rastojanje kao:

$$d_{Cosine}(x, y) = 1 - \frac{\sum_{i}^{n}(x_i * y_i)}{\sqrt{\sum_{i}^{n}(x_i)^2} * \sqrt{\sum_{i}^{n}(y_i)^2}}$$

n – broj atributa (dimenzija vektora)

4. Evaluirati trenirane modele na datom validacionom skupu podataka tako što se poredi prediktovana vrednost svakog atributa sa stvarnom vrednošću . Rezultate prikazati na *Lift Chart* grafikonu (pomoću Excel ili nekog drugog alata) za 30, 50, 70 i 100% populacije validacionih podataka, gde se na x-osi nalazi procenat ukupne populacije, a na y-osi procenat korektno prediktovanih vrednosti za ciljne atribute. Posebno iscrtati krivu za korektno prediktovanu vrednosti 0, 1 i ukupno, tj. obe vrednosti za atribut *is_attack*. Takođe, isto uraditi i za atribut *attack_group.*
5. Dobijeni model uporediti sa modelom iz Zadatka 2 na sledeći način :
    a. Rezultate predikcije nad ciljnim atributima dobijenih od oba DM modela eksportovati u posebnu tabelu za poređenje, gde pored primarnog ključa po dve kolone predstavljaju dobijene rezultate za svaki model. Takođe, dopuniti tabelu stvarnim vrednostima iz validacionog skupa za ova dva atributa, kao i atributom *attack_type.*
    b. Uporediti dobijene vrednosti ciljnih atributa sa njihovim stvarnim vrednostima i dobijene rezultate prikazati na *Lift Chart* grafikonu (pomoću Excel ili nekog drugog alata) za 30, 50, 70 i 100% populacije validacionih podataka. Iscrtati rezultate za svaku vrstu napada ponaosob na osnovu kolone *attack_type* na sledeći način:
        i. Za sve poznate vrste napada (koje se nalaze u training skupu) dati po jednu zajednicku krivu za svaki DM model i označiti je labelom *known*.
        ii. *Z*a sve nepoznate napade (one vrste napada koje se nalaze u validacionom, ali ne i u training skupu) za svaki model dati zasebnu krivu i označiti je imenom napada, tj. vrednošću atributa *attack_type.*

    Rezultate poređenja iscrtati za svaku vrstu napada na zasebnom grafikonu, tako da se na x-osi nalazi procenat populacije, a na y-osi procenat korektno prediktovanih vrednosti. Više detalja i primer *Lift Chart* grafikona pogledati na adresi http://technet.microsoft.com/en-us/library/ms175428.aspx
6. Objasniti koji model daje bolje rezultate i zašto.

Proizvodi

Na prvi deo ispita potrebno je doneti kompletno urađen projektni zadatak.
Za usmenu odbranu uraditi i pripremiti sledeće:

- U elektronskoj formi (CD/DVD) :

    1. Implementirati softverski sistem upotrebom Microsoft SQL Server baze podataka. Potrebno je doneti kompletan projekat u elektronskoj formi.
    2. Na odbrani će biti dostavljen nezavistan validacioni skup podataka nad kojim treba demonstrirati rad realizovanog sistema. Demonstracija treba da omogući zadavanje proizvoljnih ulaznih podataka, tj. putanje do fajla čiji format odgovara formatu podataka dostavljenom za treniranje modela.

- U štampanoj formi predati kompletnu dokumentaciju. Detaljno dokumentovati dati softverski sistem opisujući svaki korak po uputstvima datim u zadatku. Takođe, uvrstiti sve tražene grafikone i naglasiti ključne delove sistema.

Zapisnik revizija

Ovaj zapisnik sadrži spisak izmena i dopuna ovog dokumenta po verzijama.

Verzija 1.1

Strana	**Izmena**
7,8	Poda tačkom 4. Zadataka 2 i 3 dodata rečenica koja se odnosi na atribut *attack_group*.