UNIVERZITET U BEOGRADU
MATEMATIČKI FAKULTET

Vladana M. Đorđević

PREPOZNAVANJE BEZBEDNOSNIH NAPADA NA RAČUNARSKI SISTEM PRIMENOM MAŠINSKOG UČENJA

master rad

Beograd, 2023.

Mentor:

dr Mladen Nikolić, vanredni profesor
Univerzitet u Beogradu, Matematički fakultet

Članovi komisije:

dr Milena Vujošević Janičić, vanredni profesor
Univerzitet u Beogradu, Matematički fakultet

dr Jovana Kovačević, docent
Univerzitet u Beogradu, Matematički fakultet

Datum odbrane: _____

Zahvaljujem se mentoru profesoru Mladenu Nikoliću, profesorki Mileni Vujošević Janičić i profesorki Jovani Kovačević na savetima, podršci i strpljenju. Mojoj porodici i prijateljima

Naslov master rada: Prepoznavanje bezbednosnih napada na računarski sistem primenom mašinskog učenja

Rezime:

Bezbednost računarskih sistema je važna oblast informatike, jer istražuje slabe tačke sistema, ranjivosti i nedostatke, i predlaže alate i metode za zaštitu sistema i privatnosti. Mašinsko učenje je oblast veštačke inteligencije koja ima veliku primenu u rešavanju svakodnevnih problema. Koristeći bazu podataka koja u sebi sadrži konekcije razvrstane na benigne i maliciozne, mogu se primeniti modeli mašinskog učenja koji bi, nakon obučavanja nad tom bazom, bili u stanju da klasifikuju nove konekcije. Takav model mašinskog učenja se može koristiti za razvoj alata za prepoznavanje upada u sistem koji bi pravovremeno alarmirao da je došlo do napada na računarski sistem. Cilj ovog rada je da se formiranjem kvalitetnih modela mašinskog učenja koji su obučavani nad bazom podataka napravi način prepoznavanja zlonamernih konekcija na mreži. Baza podataka koja će biti korišćena u ovom radu je *CIC-IDS2017* i formirana je upravo za primenu klasifikacije nad njom. Autori baze su u propratnom naučnom radu predstavili svoja istraživanja i finalne rezultate čime su predložili određene modele mašinskog učenja kao najkvalitetnije. U radu će biti isprobani predloženi modeli i istraženi novi, radi bržeg i kvalitetnijeg uočavanja napada.

Ključne reči: Bezbednost računarskog sistema, Mašinsko učenje

Sadržaj

Glava 1

Uvod

Današnjica je nezamisliva bez korišćenja softvera i interneta. Veliki broj operacija je automatizovan, i skoro da nema osobe koja nema društvene mreže i ne koristi internet svakodnevno. Sve te olakšice dovode do toga da je svako potencijalna meta sajber napada, bilo da je u pitanju uništenje sistema, krađa podataka ili nešto treće. Neki softveri dostupni na internetu čuvaju veoma osetljive podatke i obavljaju ključne zadatke, stoga je njihovo neometano funkcionisanje i bezbednost od presudnog značaja. Jedna vrsta zaštite je sistem za prepoznavanje upada u računarski sistem koji je u stanju da prepozna malicioznu konekciju i da digne uzbunu da bi se na vreme zaustavio napad. Sistem zaštite bi se mogao sastojati od modela mašinskog učenja kroz koje prolaze konekcije na mreži i model ih klasifikuje na dobroćudne i zlonamerne. Za formiranje kvalitetnog modela potrebna je kvalitetna baza podataka. Takvih javno dostupnih baza nema mnogo zbog pitanja privatnosti, a za nekolicinu koja jeste javno dostupna je utvrđeno da postoje određeni nedostaci. Kanadski institut za sajber bezbednost je formirao novu bazu podataka pod imenom *CIC-IDS2017* i predložio je kao poboljšanje u odnosu na prethodne. Autori baze su primenili 7 modela mašinskog učenja nad njom i određene modele izdvojili kao najbolje. Cilj ovog rada je ispitati predložene modele, kao i isprobati neke druge u potrazi za potencijalno kvalitetnijim i bržim modelom.

U ovom radu je nad podacima iz baze *CIC-IDS2017* obučavan model slučajnih šuma, koji su autori baze predložili kao najbolje rešenje, u cilju provere kvaliteta modela. Obučavani su i modeli potpuno povezane neuronske mreže i kvadratne diskriminantne analize u cilju poboljšanja rezultata koje su dobili autori baze. Takođe, isprobani su dodatni modeli koji uključuju XGBoost i multinomijalnu logističku regresiju u cilju pronalaženja potencijalno kvalitetnijeg rešenja. Osim različitih mo-

dela isprobane su i različite metode balansiranja podataka i različit broj instanci po klasama nakon balansiranja.

Model slučajnih šuma se ispostavio kao zaista jedan od najboljih modela, model XGBoost je pokazao još bolje rezultate, dok su rezultati modela potpuno povezanih neuronskih mreža i kvadratne diskriminantne analize uspešno poboljšani u odnosu na rezultate autora baze kada se posmatraju težinski proseci preciznosti, odziva i F1 mere. Model multinomijalne logističke regresije se pokazao kao najslabiji od svih.

U poglavlju 2 su objašnjeni neki od ključnih pojmova bezbednosti sistema, sve vrste napada koje se pojavljuju u bazi i nešto više detalja o dostupnim bazama podataka. Poglavlje 3 obrađuje mašinsko učenje, osnovne koncepte, modele, tehnike i mere evaluacije koji su korišćeni u radu. Poglavlje 4 sadrži detaljan opis eksperimentalne evaluacije, koja uključuje pretprocesiranje, formiranje modela, tumačenje rezultata i odabir finalnih modela i njihovo ocenjivanje. Poslednje poglavlje, poglavlje 5, sadrži zaključak rada.

Glava 2

Bezbednost sistema

Softver danas kontroliše i automatizuje mnogo aspekata života i pruža mnogo prednosti — veću brzinu, veći otpor na greške, veću preciznost u odnosu na čoveka. Ne koristi se samo kod kuće ili u kancelariji, već kontroliše mnoštvo važnih i osetljivih operacija, kao što je kontrolisanje nuklearnih elektrana, medicinskih uređaja, upravljanje vozilima, protok podataka na internetu, dobavljanje električne struje gradovima, upravljanje satelitima itd. Međutim, korišćenje softvera može dovesti do raznih vrsta zloupotreba koje mogu imati dalekosežne posledice. Zato je bezbednost sistema od ključne važnosti u današnjem svetu [42].

„Bezbednost sistema je zaštita dodeljena automatizovanom informacionom sistemu u cilju da postigne primenljive ciljeve očuvanja integriteta, dostupnosti i poverljivosti resursa informacionog sistema (uključujući hardver, softver, firmver, informacije/podatke, i telekomunikacije) [21]". Ukoliko je bezbednost sistema narušena, to može dovesti do velikih problema kao što su pad sistema, narušavanje privatnosti, manipulacija i uništenje podataka, krađa osetljivih informacija, ogromni finansijski troškovi, naleti neželjenih reklama i spamova, itd. Mnogo je lakše narušiti bezbednosti sistema nego sagraditi bezbedan sistem, jer je napadaču dovoljno da pronađe jednu slabost koju može da iskoristi, dok se formiranjem bezbednog sistema mora razmišljati o svemu i uzeti u obzir sve što može da predstavlja ranjivost. Često je nemoguće pronaći sve slabe tačke [42].

Neki od najvažnijih elemenata softvera su integritet (eng. *integrity*), raspoloživost (eng. *availability*) i poverljivost (eng. *confidentiality*) podataka. Integritet podataka podrazumeva da se podaci i programi ne mogu menjati na neki način koji nije ovlašćen ili poželjan, čime se podaci održavaju tačnim, kompletnim i konzistentnim. Raspoloživost podataka podrazumeva da sistem izvršava funkcije na vreme, da

usluge nisu odbijene ovlašćenim korisnicima i da su podaci blagovremeno dostupni. Poverljivost podataka podrazumeva da se lične ili poverljive informacije ne odaju neovlašćenim pojedincima. Još jedan bitan softverski termin je integritet sistema (eng. *system integrity*) koji podrazumeva da funkcionalnost sistema nije narušena i da niko nije, namerno ili slučajno, manipulisao sistemom [21].

Postoji mnoštvo opasnosti koje prete sistemima uključujući greške koje narušavaju integritet podataka, požare koji uništavaju čitave računarske centre, krađe osetljivih i poverljivih podataka i narušavanje poverljivosti sistema, napade koji onesposobljavaju sistem i narušavaju raspoloživost sistema, radnici od poverenja koji su učinili prevaru itd. Samim tim nastaju različite vrste štete i nastaju značajni gubici. Napadači se obično nazivaju hakerima (eng. *hackers*). Termin haker se obično odnosi na maliciozne napadače koji pokušavaju da pristupe sistemima, mrežama, poverljivim informacijama a da pritom za to nemaju ovlašćenja. Međutim, postoje i etični hakeri (eng. *ethical hackers*) koji napadaju sisteme da pronađu ranjivosti u cilju zaštite sistema. Dobri hakeri se drugačije nazivaju hakeri sa belim šeširom (eng. *white-hat hackers*), a loši se nazivaju hakerima sa crnim šeširom (eng. *black-hat hackers*). Osim hakera, napadači mogu biti i zlonamerni korisnici u koje spadaju klijenti, poslovni partneri, ovlašćeni i pouzdani korisnici, zaposleni i bilo koji drugi korisnik koji zloupotrebljava dobijene privilegije da bi ostvario nepoštene ciljeve [21, 3].

„Prepoznavanje upada se odnosi na proces identifikovanja pokušaja prodiranja u sistem i dobijanja neovlašćenog pristupa [21]". To se postiže nadgledanjem sistema koje može da vrši odgovarajući hardver ili softver. Sistem za prepoznavanje upada (eng. *intrusion detection system*) je računarski sistem koji izvršava procese koji obezbeđuju sistem ili mrežu. Može se kategorisati u tri kategorije: sistem zasnovan na mreži, sistem zasnovan na domaćinu ili hibridni sistem. U sistemu za prepoznavanje upada koji je zasnovan na mreži, zlonamerne informacije su prepoznate iz različitih konekcija između računara, pa se ova vrsta sistema ugrađuje u rutere ili svičeve na mreži i glavni zadatak joj je da identifikuje maliciozne ili preteće informacije iz mrežnih evidencija i da izvesti rukovodioca mreže o tome. Ovaj sistem obično ne sprečava napad već samo alarmira u realnom vremenu [36].

S obzirom na kompleksnost softverskih sistema postoji veliki broj različitih vrsta napada. U daljem tekstu će biti izdvojeni i detaljnije opisani napadi koji su od značaja za razumevanje rada, odnosno napadi koji se pojavljuju u bazi podataka koja je korišćena u radu.

2.1 Napad uskraćivanjem usluga

Jedna od najvećih i najozbiljnijih pretnji po bezbednost na internetu je *napad uskraćivanjem usluga* (eng. *denial of service — DoS*). Mete napada se obično nazivaju „žrtvama". Glavni cilj ovog napada je da poremeti pružanje usluga i onemogući pristup legitimnim korisnicima tako što će ograničiti pristup mašini ili servisu. Cilj se postiže slanjem toka paketa žrtvi koji potom preplavi mrežu ili onesposobi obrađivanje podataka. Na ovaj način se pristup računaru ili mreži namerno smanjuje ili blokira. Ova vrsta napada ne mora nužno da ošteti podatke, bilo direktno ili trajno, ali namerno kompromituje dostupnost resursa. Teško ju je sprečiti jer je mrežni saobraćaj veoma gust i samim tim je teže razdvojiti legitiman saobraćaj od nelegitimnog saobraćaja. Nema očiglednih karakteristika napada koje mogu da pomognu u tome da se napad detektuje ili filtrira. Za napad se koriste ogromne količine paketa koje mogu imati različita polja što omogućava da se izbegnu karakterizacije i ostavljanje tragova [9, 27].

Posebna vrsta ovog napada — *distribuirani napad uskraćivanjem usluga* (eng. *distributed denial of service — DDoS*) — je najopasnija vrsta, jer posledice mogu biti teške. Distribuirani napad uskraćivanjem usluga podrazumeva napadača ili grupu napadača koji koriste veći broj mašina da izvedu napad uskraćivanjem usluga istovremeno, čime se povećava efikasnost i snaga napada. Veliki broj računara se koristi za koordinisani napad na jednu ili više meta, a ti računari su uglavnom nesvesni da se koriste za napad. Do napada može doći bez ikakvog upozorenja i, za kratak vremenski period, se mogu iscrpeti svi računski i komunikacioni resursi žrtve. Može se ciljati protok ili povezanost mreže. Napadi protoka preplavljuju mrežu velikim obimom saobraćaja što dovodi do toga da legitimni korisnički zahtevi ne mogu da dopru do usluge koja se nudi. Napadi na povezanost mreže preplavljuju računar velikim obimom zahteva za povezivanje tako da su svi dostupni resursi operativnog sistema iskorišćeni i računar više ne može da obrađuje zahteve legitimnih korisnika. Ako je obim napada veći od onoga što sistem može da podnese onda dolazi do pada sistema. Resursi žrtve su uvek ograničeni, za razliku od napadača koji koristi distribuirani sistem i samim tim može da povećava resurse za napad [9, 49].

Prvi distribuirani napad uskraćivanjem usluga je zabeležen 1999. godine i otad je većina ovakvih napada upravo distribuirane prirode. Prvi napadnuti računar nalazio se na Univerzitetu u Minesoti i napalo ga je 114 inficiranih računara. Računar nije bio funkcionalan cela dva dana. Ubrzo nakon toga napadnuti su mnogobrojni veb-

sajtovi uključujući Yahoo, Amazon i CNN [27, 35].

Prilikom dizajniranja arhitekture interneta prioritet je stavljen na funkcionalnost, a ne na bezbednost. Upravo to je njegova ogromna prednost, ali i ogromna mana. Sistem žrtve može imati veoma veliku bezbednost i jak sistem odbrane, ali u velikoj meri zavisi i od globalnog interneta, jer je na internetu bezbednost veoma međuzavisna. Ruteri u jezgru mreža ne mogu da pruže autentifikaciju dostavljenim IP paketima. Ovo dovodi do problema poznatog kao *„IP lažiranje"* (eng. *IP spoofing*) gde napadači pružaju lažne informacije kao što su lažne IP adrese izvora u IP paketima. Upravo to omogućava uspešan napad jer pruža napadačima mogućnost da ostanu skriveni tokom napada. Pored toga, ruteri nemaju mehanizam ulaženja u trag paketima zbog ogromne količine saobraćaja kojom barataju. Internet je decentralizovan, svim mrežama se upravlja lokalno i ne postoji centralna hijerarhija upravljanja ili neki autoritet. Samim tim, ne postoji implementacija centralnog mehanizma odbrane u slučaju napada. Iako ne postoji sveobuhvatno rešenje za ove vrste napada postoji nekoliko protivmera čiji je fokus ili na tome da učine napad težim ili da učine da napadač odgovara za svoje postupke [9, 27].

Distribuirani napad uskraćivanjem usluga se uobičajeno sastoji od 4 elementa [9]:

Napadač — osoba ili organizacija koja planira i izvodi napad.

Rukovaoci **(eng.** ***handlers*****)** ili ***masteri*** **(eng.** ***masters*****)** — oteti računari na kojima se izvršava poseban program za kontrolisanje više agenata.

Agenti demona **(eng.** ***daemon agents*****)** ili ***zombi domaćini*** **(eng.** ***zombie hosts*****)** — oteti računari na kojima se izvršava program za generisanje toka paketa koji se šalje žrtvi. Ovi računari se obično nalaze u spoljašnjoj mreži u odnosu na žrtvu da bi se izbegao efikasan odgovor žrtve, takođe je u spoljašnjoj mreži u odnosu na napadača da bi se izbeglo ulaženje u trag.

Žrtva — osoba ili organizacija koja je meta napada i koja je direktno pogođena napadom.

2.2 Skeniranje priključaka

„Kada aplikacija (tj. korisnički proces) poželi da uspostavi vezu s procesom neke udaljene aplikacije, ona mora da ga eksplicitno navede... Najčešće se primenjuje

sledeći postupak: definišu se adrese na kojima procesi mogu da osluškuju (čekaju) zahteve za uspostavljanje veze. Takve krajnje tačke na Internetu nazivaju se priključcima *(eng.* ports*) [46]".* Priključci su centralne tačke spajanja za protok informacija. IP adresa u kombinaciji sa brojem priključka predstavlja veoma bitnu informaciju koju čuva svaki pružalac internet usluga (eng. *internet service provider*) da bi mogao da ispunjava zahteve korisnika. Priključci se uglavnom nalaze u rasponu od 0 do 65 535. Neki od poznatijih priključaka su priključak 20 koji pruža uslugu za prenos podataka (eng. *file transfer protocol — FTP*), 53 koji prevodi domenska imena u IP adrese (eng. *domain name system — DNS*), 80 koji služi za komunikaciju na vebu (eng. *hypertext transfer protocol — HTTP*), itd. *Skeniranje priključaka* (eng. *port scan*) je metod za određivanje koji priključci na mreži su otvoreni i mogu da šaju ili primaju podatke, tj. da li je određena usluga dostupna na domaćinu ili mreži tako što se posmatraju odgovori na pokušaje povezivanja. Skeniranje priključaka ne izvode samo napadači, već mogu koristiti i zaštitnici mreže u cilju pronalaženja slabih tačaka koje mogu biti meta napada. Veoma je teško prepoznati maliciozno skeniranje priključaka [2, 4].

Prilikom skeniranja mreže prvo se identifikuje lista aktivnih domaćina i njihovih IP adresa, nakon toga se izvršava skeniranje priključaka. Cilj je identifikovati organizacije IP adresa, domaćina i priključaka da bi se utvrdile ranjivosti i nivo bezbednosti. Na taj način se može otkriti prisustvo bezbednosnih mera kao što je *zaštitni zid* (eng. *firewall*) koji je postavljane između servera i korisničkog uređaja. Osim toga napadači mogu prikupiti informacije i o operativnom sistemu i aplikacijama koje domaćin koristi. Sve se to postiže tako što napadač šalje posebne podatke na priključak domaćina na mreži i potom analizira odgovor. Primer jednostavnog skeniranja priključaka je *TCP skeniranje* (eng. *TCP scan*) koje pokušava da izvede *trostepeno usaglašavanje* (eng. *three-way handshake*) sa ciljanim priključkom i na taj način proverava da li je priključak u upotrebi. Problem sa tim je što se trostepeno usaglašavanje koristi prilikom standarnog TCP povezivanja tako da je veoma teško razlikovati napad i regularnu komunikaciju. Zbog toga su potrebne neke dodatne informacije koje se izvlače iz većeg broja paketa da bi se uspešno identifikovao napad. Primer bi bio broj pristupa priključcima u jedinici vremena sa istog domaćina ka različitim domaćinima ili broj saobraćaja po priključku [4, 8].

Rezultati skeniranja priključaka mogu otkriti napadaču da li su priključci otvoreni, zatvoreni ili filtrirani [4, 30].

Otvoreni priključak — ako je priključak otvoren to znači da aplikacija aktivno prihvata povezivanja, aplikacija odgovara paketom koji ukazuje na to da osluškuje. Uglavnom je pronalaženje otvorenih priključaka glavni cilj skeniranja priključaka. IT administratori su svesni da su otvoreni priključci prava meta za napad i na njima je da ih obezbede tako što će postaviti zaštitni zid koji će omogućiti pristup samo legitimnim korisnicima.

Zatvoreni priključak — ako je priključak zatvoren to znači da je priključak dostupan i server je primio zahtev, ali nema servisa koji osluškuje na tom priključku. Takođe, korisni su da pokažu da se domaćin nalazi na IP adresi i kao deo otkrivanja operativnog sistema. IT administratori bi trebalo da nadziru zatvorene priključke, jer postoji mogućnost da promene status i postanu otvoreni samim tim stvarajući ranjivosti. Takođe, treba razmisliti o tome da se zatvoreni priključci blokiraju zaštitnim zidom čime postaju „filtrirani" priključci.

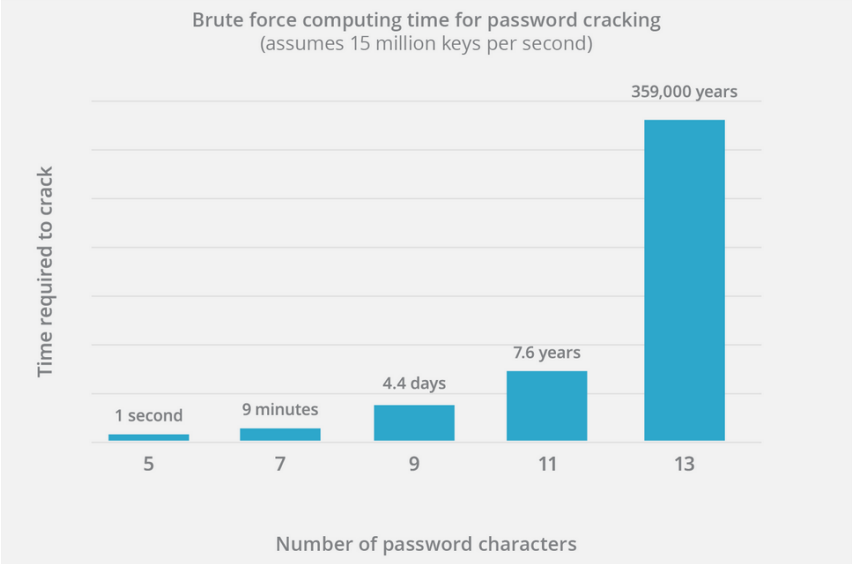
Filtrirani priključak — ako je priključak filtriran to znači da je zahtev poslat, ali domaćin nije odgovorio niti osluškuje zato što paket nije stigao do ciljne destinacije i samim tim napadač ne može da otkrije više informacija. To uglavnom znači da je zaštitni zid paket zahteva isfiltrirao ili blokirao ili oba, pa će filtrirani priključak odgovoriti porukom o grešci koja glasi „nedostupna destinacija" ili „zabranjena komunikacija" ili uopšte neće odgovoriti, što je najčešći slučaj. Napadači uglavnom probaju skeniranje priključaka na filtriranim priključcima više puta da bi proverili da li im je zahtev propao zbog filtriranja ili je možda u pitanju bilo preopterećenje mreže. Filtrirani priključci znatno usporavaju skeniranje.

2.3 Gruba sila

Napad grubom silom (eng. *brute force attack*) je napad koji iscrpno isprobava različite ulazne kombinacije sa ciljem da pogodi osetljive i skrivene informacije, prvenstveno *informacije za prijavljivanje* (eng. *login information*) uključujući provaljivanje šifara, ključeva za enkripciju, otkrivanje skrivenog sadržaja ili veb-stranica (čime se stiče neautorizovani pristup računaru). Osim toga, mete mogu biti i API ključevi i SSH prijavljivanja. Napadači konstruišu unapred određenu listu vrednosti koje će da isprobaju, šalju zahtev ka serveru koristeći te vrednosti i potom analiziraju odgovor. Lista unapred određenih vrednosti se uglavnom sastoji od svih mogućih

kombinacija karaktera [23, 6, 31]. *„Možemo napraviti paralelu sa lopovom koji pokušava da provali u kombinacijski sef tako što će isprobati svaku moguću kombinaciju brojeva dok se sef ne otvori [6]”.*

U zavisnosti od dužine i složenosti šifre provaljivanje može da traje od nekoliko sekundi do mnogo godina. Ako se šifra poveća čak i za jedan karakter vreme potrebno da se ta šifra provali značajno raste. Posle određenog broja karaktera postaje nerealistično da će šifra biti provaljena korišćenjem metode grube sile kao što se može videti na slici 2.1. Uviđa se da je sporost slabost ove vrste napada, što je rezultat toga što ovaj metod i nije preterano inteligentan. Sa druge strane, jednostavno se izvodi i uvek će uspeti ako ima dovoljno vremena i meta nema jaku bezbednost. Ako je napadač uspešno procenio efikasnost sistema koga napada onda može i da proceni vreme koje mu je potrebno da isproba sve vrednosti na listi. Vreme koje je potrebno da se grubom silom probije u sistem može biti mera sigurnosti sistema [23, 6, 31].



Slika 2.1: Vreme potrebno da se šifra provali u zavisnosti od broja karaktera šifre.

Tipovi napada grubom silom su sledeći [23, 44, 7]:

***Jednostavan napad* (eng. *simple attack*)** — Napadač mnogo puta pokušava da provali šifru koristeći nasumične vrednosti sve dok ne pogodi. Za ovaj metod je potrebno mnogo vremena i resursa. Metod je dobar za veoma jednostavne šifre i PIN-ove, kao što su „gost12345” ili „admin12345” na primer, koje u sebi ne mešaju velika i mala slova i koriste česte izraze i sekvence brojeva.

Tipični napadi grubom silom mogu da naprave nekoliko stotina pokušaja u jednoj sekundi.

***Rečnički napad* (eng. *dictionary attack*)** — Kao što se i u rečniku mogu pronaći skoro sve validne reči, tako i napadači mogu napraviti sopstvene kolekcije najpopularnijih rečenica i izraza. Osim popularnih izraza za šifre, napadači u rečnik dodaju i šifre koje su procurele ranije, one su dostupne za prodaju na dark vebu, a mogu se čak naći i na regularnom vebu i to besplatno. U više fokusiranim rečničkim napadima mogu čak i da pretraže individualnog korisnika na internetu (tražeći po njegovim blogovima, profilima na društvenim mrežama itd.) da bi se utvrdili njegovi hobiji, preferencije, i reči ili rečenice koje se ističu. Na kraju te reči mogu dodati u svoje rečnike.

***Hibridni napad* (eng. *hybrid attack*)** — Nastaje spajanjem dva različita načina napada grubom silom. Na primer, haker može da integriše rečnički napad sa tipičnim napadom grube sile. Ovi napadi se koriste da bi se pronašle kombinovane šifre koje mešaju učestale reči ili rečenice sa nasumičnim karakterima. Primer ovakvog napada bi bio isprobavanje šifara „Beograd1993" ili „Miloš1234".

***Obrnuti napad* (eng. *reverse attack*)** — Kao što i samo ime kaže, obrnuti napad grubom silom obrće strategiju napada tako što počinje poznatom šifrom i potom pretražuje milione korisničkih imena sve dok ne pronađe poklapanje. Ovakvi napadi mogu biti noćna mora za poslovanja čiji zaposleni ne koriste šifrovana prijavljivanja.

***Punjenje kredencijala* (eng. *credential stuffing*)** — Ako je napadač uspešno pronašao kombinaciju korisničkog imena i šifre za jedan veb-sajt, probaće ih i na mnoštvu drugih veb-sajtova. Ovo se pokazalo kao veoma uspešna tehnika s obzirom na to da korisnici često recikliraju svoje šifre. Takođe, tokom godina više od 8.5 milijardi korisničkih imena i šifri je procurelo, i oni se i danas prodaju na dark vebu.

***Mreža botova* (eng. *botnet*)** — Napad grubom silom zahteva mnogo računarske snage. Napadači mogu da ubrzaju napad ako zaposle više otetih računara da izvršavaju napad što ih, takođe, dodatno skriva od prepoznavanja. Mreža botova se može koristiti u bilo kom tipu napada grubom silom.

Napadi grubom silom se obično sastoje od tri faze: skeniranje, gruba sila i kompromitovanje. Tokom prve faze, napadači izvode horizontalno skeniranje mreže da bi pronašli mete. Horizontalno skeniranje mreže je skeniranje jednog priključka na više IP adresa. U ovom slučaju se traže aktivni demoni[1] na određenom priključku. Potom sledi napad grubom silom i, ukoliko se uspešno pronađe kombinacija, meta je kompromitovana čime se ulazi u treću fazu napada. Druga faza je najduža i najintenzivnija gde se šalje veliki broj paketa meti. Ukoliko meta postane kompromitovana, napadači mogu odmah da ih zloupotrebe ili da ih ostave po strani. Ne moraju sve faze da budu vidljive u određenom delu mrežnog saobraćaja. Napadači mogu da odlože izvršavanje određene faze napada, ili da različite faze izvršavaju sa različitih mašina da bi izbegli otkrivanje [22].

U daljem toku rada od interesa će biti dva tipa napada grubom silom — *„SSH Patator"* i *„FTP Patator"*. Patator je alat za izvođenje napada grubom silom. Napisan je u programskom jeziku *Python*, sa ciljem da bude fleksibilan, pouzdan i da podržava razne module za različite napade grubom silom, među kojima su i *ftp_login* i *ssh_login* [20].

„SSH ili secure shell je mrežni protokol za šifrovanu komunikaciju preko nebezbednih mreža. SSH se koristi za prijavljivanje na daljinu, izvršavanje naredbi, prenos datoteka, itd. SSH napadi grube sile se izvode tako što napadač pokušava uobičajena korisnička imena i šifre na hiljadama servera dok ne pronađe poklapanje [10]."

„FTP (file transfer protocol) je mrežni protokol za prenos datoteka između računara preko TCP/IP (eng. transmission control protocol/internet protocol*) konekcija [47]."* FTP napadi se izvode isto kao SSH napadi, samo na FTP serveru umesto na SSH.

2.4 Bot

Bot (skraćeno od robot) je softverska aplikacija koja automatizuje određene zadatke u cilju simuliranja ljudske aktivnosti. Obično je deo nekog programa i zadaci uključuju pretraživanje sadržaja sajtova na vebu, razgovore u četovima, nadziranje ispravnosti sistema ili mreže, izvršavanje transakcija ili raznih zadataka na društvenim mrežama. Automatizovani zadaci se izvršavaju brže i tačnije nego kad ih izvodi čovek, pogotovo ako se neki zadatak često ponavlja. Međutim, ponekad se botovi

[1]Demoni su pozadinski procesi koji se ne izvršavaju sve vreme već se aktiviraju kada stigne zahtev za određenom uslugom.

koriste za sprovođenje hakerskog napada, najčešće distribuiranog napada uskraćivanjem usluga, potom širenje spama, trojanske i pecaroške elektronske pošte, nezakonito distribuiranje piratizovanog softvera i sadržaja, krađu podataka i identiteta itd. Bot sam po sebi nije maliciozan, ali se može koristiti u maliciozne svrhe isto kao što se može koristiti u benigne svrhe. Prvobitno su botovi korišćeni za jednostavne operacije kao što je spamovanje, ali su vremenom prerasli u daleko ozbiljnije napade koje kriminalne organizacije koriste za različite kriminalne aktivnosti. Sofisticiraniji bot napadači ne koriste developerske alatke za pravljenje botova koji su besplatno dostupni na internetu, već pišu sopstveni kôd koji automatizuje napad tako da bude različite dužine i učestalosti čineći ga manje upadljivim za bezbednosno nadgledanje [38, 28, 11].

Mreža botova (eng. *botnet*) je mreža kompromitovanih i otetih mašina. Mašine se koriste bez znanja i dozvole vlasnika mašine. Što se više mašina nalazi u mreži botova napadač može brže i bolje da izvede svoj napad. Ukoliko je potrebno proširiti mrežu zarad još efikasnijeg napada, napadači to mogu postići brzo i jeftino. Velika mreža botova takođe pruža anonimnost napadaču, jer se botovi mogu nalaziti bilo gde na svetu, a međunarodni zakoni imaju ograničenja kada je u pitanje ulaženje u trag aktivnostima koje su se desile na prostoru druge države [24, 43, 11].

Napadač može biti jedna osoba ili čak čitava organizacija. Životni ciklus napada mreže botova se može podeliti u pet faza.

U prvoj fazi, koja se naziva inicijalna infekcija, cilj je pronaći žrtvinu ranjivost — na veb-sajtu, aplikaciji ili u ljudskom ponašanju — i inficirati je nekim načinom eksploatacije. Obično se to sprovodi kroz elektronsku poštu ili poruke na mreži.

U drugoj fazi, fazi sekundarne injekcije, inficirana žrtva izvršava skriptu pod imenom *shell code*. Uglavnom se korisnici nagovore preko socijalnog inženjeringa da preuzmu trojanski virus. Neki napadači su agresivniji pa koriste *drive by download* kada se poseti inficirani sajt. Koji god da je metod u pitanju, sajber kriminalci na kraju probiju bezbednost nekoliko korisničkih računara. *Shell code* preuzima kopiju pravog binarnog bota sa konkretne lokacije pomoću FTP-a, HTTP-a ili P2P-a, čime se binarni bot instalira na ciljanu mašinu i mašina postaje zombi koji izvršava maliciozan kôd. Bot aplikacija će se automatski pokrenuti svaki put kada se zombi mašina pokrene.

Treća faza je poznata kao faza konekcije gde je cilj da se zombi poveže na *naredi-i-kontroliši kanal* (eng. *command-and-control (C&C) channel*) kojim se povezuje sa *naredi-i-kontroliši (C&C)* serverom i postaje deo armije botova. C&C je server sa

koga potiču sve botnet instrukcije i vođstvo. Ovo je glavni server napadača i svaki zombi računar dobija naređenja odatle.

Četvrta faza je faza zlonamernih naredbi i kontrole žrtve, tu napadač počinje da izvršava nedozvoljene aktivnosti. Naređenja se šalju kroz C&C kanal, čime se omogućava da napadač sa daljine upravlja svojom armijom botova. Često sajber kriminalci imaju tendenciju da kontrolišu hiljade, desetine hiljada pa čak i milione računara. Sajber kriminalac se potom ponaša kao gazda ogromne mreže zombija. Nakon što je inficiran, zombi računar napadaču dozvoljava pristup upravljačkim (administratorskim) operacijama kao sto su čitanje i pisanje sistemskih podataka, prikupljanje ličnih podataka, slanje i primanje datoteka, nadgledanje korisnikovih aktivnosti, pretraživanje ranjivosti u drugim uređajima, instaliranje i pokretanje bilo koje aplikacije.

Poslednja faza je da se botovi održavaju aktivnim i ažurnim, botovi dobijaju naredbe da preuzimaju i ažuriraju binarni bot. Napadač će možda biti primoran da ažurira svoju mrežu botova iz više razloga, među kojima su izbegavanje detekcije ili dodavanje nove funkcionalnosti zombijima, pa čak i preseljavanje botova na drugi C&C server. Ovaj proces se naziva migracijom servera i veoma je korisno za napadače da održavaju svoje armije botova. To se postiže korišćenjem dinamičnog DNS-a — DDNS-a — što je usluga za razrešavanje koja olakšava česta ažuriranja i promene u lokacijama servera. Ako vlasti poremete rad C&C instance servera na određenoj IP adresi napadač može lako da postavi drugu C&C instancu servera sa istim imenom na drugoj IP adresi. Promene IP adresa na C&C serverima propagiraju se skoro odmah. Stoga, botovi će se migrirati na drugu lokaciju C&C servera i nastaviće sa radom. Ovakvom mrežom botova se može upravljati direktno ili indirektno korišćenjem sledećih modela — centralizovani klijent server model ili decentralizovani *peer to peer* model. Centralizovane modele vodi jedan server napadača. Mogu postojati dodatni serveri kao podređeni napadači ili proksiji. Međutim, sve komande dolaze od napadača. Obe ove strukture mogu dovesti do toga se napadač otkrije što čini ove zastarele metode manje nego idealnim. Decentralizovani modeli ugrađuju odgovornosti davanja instrukcija svim zombi računarima. Dokle god napadač može da kontaktira neki od zombi računara oni mogu da šire naređenja ostalima. Ova struktura dodatno skriva identitet napadača. Danas se mnogo više koristi *peer to peer* model zbog očiglednih prednosti [11, 24].

2.5 Veb napadi

U današnje vreme je teško zamisliti veću kompaniju koja nema svoj veb-sajt ili veb-aplikaciju. Umreženost omogućava daleko lakše poslovanje i za korisnike i za pružaoce usluga i uglavnom stoje na raspologanju 24 sata dnevno, 7 dana u nedelji. Otvoreni su i dostupni svima, što ih čini lakim metama za hakerske napade. Veb-sajt, kao i svaki softver, ima svoje ranjivosti i slabosti koje se mogu iskoristiti za neautorizovan pristup, ubacivanje zlonamernog sadržaja, izmenu sadržaja veb-sajta, kao i prikupljanje poverljivih informacija kao što su informacije o kreditnim karticama, medicinske informacije, itd [17, 50, 3].

Najčešći oblici veb napada su [17, 50, 3]:

- Injektovanje koda i injektovanje SQL-a
- Međulokacijsko skriptovanje
- URL manipulacija
- Manipulacija skrivenim poljima
- Uključivanje lokalne datoteke
- DDoS
- Napad otpremanjem datoteke
- Preopterećenje bafera
- Napad grubom silom

Za dalji rad će nam od interesa biti samo injektovanje SQL-a, međulokacijsko skriptovanje i napad grubom silom.

SQL injekcija (eng. *SQL injection*) je ubacivanje, odnosno „injekcija", kôda u SQL bazu podataka preko ulaznih polja veb obrasca sa klijenta na aplikaciju. Svaka aplikacija na vebu koja koristi podatke ima bazu podataka sa kojom komunicira. U slučaju da se, kao jezik za komunikaciju, koristi *SQL* (eng. *structured query language*) aplikacija šalje SQL upite bazi u cilju izvršavanja različitih operacija potrebnih za uspešan rad aplikacije. Zlonamerni napadači ubacuju SQL upite i, u slučaju uspeha, mogu da čitaju osetljive podatke iz baze podataka, koristeći izraz „SELECT", da menjaju ili obrišu podatke, koristeći „INSERT", „UPDATE" ili

„DELETE", kao i da ispituju oblik, verziju i strukturu baze podataka. Osim toga mogu i da izvršavaju administrativne operacije nad bazom podataka kao što je gašenje sistema za upravljanje bazom podataka, da pristupe sadržaju datoteke koja se nalaze u sistemu za upravljenje bazom podataka, u nekim slučajevima mogu da zadaju naredbe operativnom sistemu. Mogu se primenjivati i „UNION" izrazi, čime se vraćaju informacije iz različitih tabela u bazi podataka. Do ove vrste napada uglavnom dolazi kada aplikacija nije ispravno proverila unos ili kada server baze podataka i veb-server vraćaju informativnu poruku o grešci [3, 33, 34].

Dva osnovna tipa SQL injekcija su *standardna*, drugačija poznata i kao *„zasnovana na greškama"*, i *slepa*. Kada se SQL injekcijom u sistem ubace nevažeće informacije, sistem vraća grešku koja je vidljiva napadaču i daje mu važne informacije nakon kojih može uspešno da izvrši napad koji želi, ovo je poznato kao standardni napad. U slučaju da su poruke o greškama onemogućene, hakeru preostaje da pogađa šta vraća baza podataka i kako reaguje na napade [3].

Međulokacijsko (međuserversko) skriptovanje (eng. *cross-site scripting — XSS*) je vrsta ubacivanja — „injektovanja" — zlonamernog kôda u benigne veb-sajtove. Ukoliko aplikacija nema filtriranje ulaza korisnik može da unese šta god poželi, pa čak i maliciozan *JavaScript* kôd koji će se potom izvršiti. Moguće je i da napadač preusmeri žrtvu na svoj veb-sajt. Žrtva neće shvatiti da se ne nalazi više na prvobitnom veb-sajtu i uneće svoje podatke za prijavljivanje, uključujući korisničko ime i šifru, koje potom napadač beleži [3, 32].

U sekciji 2.3 je već obrađen napad grubom silom.

2.6 Infiltracija

Infiltracija (eng. *infiltration*) je proces neopaženog ulaska ili dobijanja pristupa nekom mestu ili organizaciji uglavnom u cilju pristupa tajnim informacijama i nanošenju štete. Tako i slučaju hakerskog napada se mogu koristiti različite metode, kao što je npr. prilog na elektronskoj pošti ili prenosivi mediji, da bi se infiltrirala mreža nakon čega se može postaviti zlonameran softver zarad preuzimanja kontrole nad mrežom ili sistemom, krađe poverljivih informacija, nanošenje štete. Infiltracija se može postići na različite načine kao što su prekoračenje bafera, pecanje, hakovanje šifara, preuzimanje besplatnog i piratizovanog softvera, fazing (eng. *fuzzing*), injekcija SQL-a i međulokacijsko skriptovanje, socijalni inženjering [5, 40].

2.7 Dostupni resursi

Bezbednost sistema je sve više ugrožena, nastaju nove vrste napada, a stare se usavršavaju. Cilj je napraviti sistem za prepoznavanje upada koji je zasnovan na mreži i koji će biti u stanju da razvrsta konekcije na napade i na normalne konekcije. Za formiranje takvog sistema ideja je koristiti tehnike mašinskog učenja koje će klasifikovati konekcije. Za potrebe mašinskog učenja neophodna je baza podataka na kojoj će model učiti kako da razvrsta konekcije.

Prilikom istraživanja pronađen je veći broj naučnih radova u kojima se spominje nekoliko baza podataka koje su napravljene upravo za svrhe konstruisanja sistema za prepoznavanje upada, međutim, njih nema mnogo zbog pitanja privatnosti. Izazov je pronaći odgovarajuću bazu, jer baza mora da sadrži različite savremene napade, dovoljno informacija, da su podaci prikupljani na različitom hardveru, softveru i mrežama, protokolima, saobraćaju itd...

Baze koje su godinama korišćene su DARPA, KDD'99, DEFCON, CAIDA, LBLN, CSX, Kyoto, Twente, UMASS, ISCX2012, ADFA. Međutim, pronađeno je mnogo nedostataka različite vrste u ovim bazama. [41].

Prilikom traženja baze za potrebe ovog master rada pronađene su neke od novijih baza koje su pravljene sa ciljem formiranja sistema za detekciju upada, a da pritom predstavljaju poboljšanje u odnosu na prethodno korišćene baze. U nastavku će biti predstavljene neke baze koje su bile razmatrane za ovaj rad:

DoS 2017 — Baza se nalazi u *PCAP* formatu. Nakon prevođenja u *CSV* format zaključuje se da ima 6 393 767 redova i 7 kolona, od atributa se nalaze redni broj, vreme, IP izvora, IP destinacije, protokol, dužina i informacije. Redni broj, IP izvora i destinacije nemaju mnogo koristi za klasifikaciju napada, ali najveći problem je što uopšte nema kolone sa klasom. Podaci nisu čitljivi da bi se iz njih mogle izvući neke informacije. Možda je došlo do greške prilikom pretvaranja u *CSV*. Takođe, nije bilo uspeha ni u pristupanju propratnom naučnom radu da bi se videlo u kom formatu i sa kojim ciljem je napravljena baza, stoga ona nije izabrana za dalji rad [13].

DDoS 2019 — Baza podataka je namenjena otkrivanju različitih vrsta DDoS napada. Sadrži benignu konekciju i većinu ažurnih DDoS napada među kojima su DNS, LDAP, MSSQL, NetBIOS, NTP, SNMP, SSDP, UDP, Syn, TFTP i UDP-Lag. Prilikom preuzimanja baze dobijaju se dva direktorijuma, u jednom se nalaze podaci prikupljeni na dan treninga, a u drugom na dan testiranja. Na

dan treninga ima 11 *CSV* fajlova, po jedan za različitu vrstu DDoS napada. Svaka od ovih baza ima nekoliko miliona redova i 88 kolona, sa izuzetkom NTP i Syn koje imaju preko 1 milion redova. TFTP čak ima preko 20 miliona redova. Ideja autora je bila da koristeći alat *CICFlowMeter-V3* beleže konekcije u realnom vremenu i da sprovode napade po danima. Najvažnije od svega je bilo da generišu realistični pozadinski saobraćaj koji se može naći u svakodnevnom korišćenju većine korisnika. Na taj način je dobijena baza podataka koja je veoma slična bazi sa podacima o stvarnim napadima. Zbog hardverskih ograničenja ova baza podataka nije izabrana za korišćenje u ovom radu, ali može biti veoma korisna za izgradnju sistema za detekciju upada specijalizovanu za DDoS napade [14].

URL 2016 — Baza podataka napravljena sa ciljem da detektuje i kategorizuje zlonamerne URL-ove po vrsti napada. Još jedan cilj je bio pokazati da je leksička analiza efikasna i takođe je proučavan efekat tehnike zamagljivanja na zlonamerne URL-ove da bi se utvrdilo koji tip tehnike zamagljivanja se koristi na kom tipu zlonamernog URL-a. Obrađeno je 5 različitih vrsta URL-ova: benigni, spam, pecanje, malver i zamrljavanje URL-ova, samim tim i ova baza ima 5 klasa. Sastoji se od 36 707 redova i 80 kolona. Poprilično dobro je balansirana. U ovom radu nije korišćena jer je previše mala za potrebe master rada [16].

Botnet 2014 — Autori su napravili ovu bazu sa ciljem da obuhvate što više botnet napada i da podaci budu sveobuhvatni, realistični, da reflektuju stvarni svet kao i da budu reprezentativni. Baza je nastala kombinovanjem podataka koji se ne preklapaju iz tri druge baze, a potom je podeljena na trening i test bazu koje uključuju 7 i 16 tipova botnet napada, tim redom. Trening baza je veličine 5.3 GB, pri čemu je 43.92% zlonamerna konekcija a ostatak benigna. Test baza je veličine 8.5 GB od čega je 44.97% zlonamerna konekcija. Došlo je do neuspeha prilikom pokušaja preuzimanja ove baze, stoga nije razmatrana za ovaj rad [12].

Na veb-sajtu Kanadskog instituta za sajber bezbednost se mogu naći baze vezane za IoT, Android OS, Dark web, Tor i VPN.

Za potrebe ovog rada izabrana je baza podataka *CIC-IDS2017* koju je napravio već pomenuti Kanadski institut za sajber bezbednost 2017. godine. Naime, na istom institutu je sprovedeno istraživanje nad već postojećim i dostupnim bazama da bi se

utvrdilo koji su njihovi nedostaci i koje kriterijume jedna baza treba da zadovolji da bi bila pouzdana. Oni su predložili 11 kriterijuma: potpuna mrežna konfiguracija (eng. *complete network configuration*), potpuni saobraćaj (eng. *complete traffic*), označena baza podataka (eng. *labelled dataset*), potpuna interakcija (eng. *complete interaction*), potpuno hvatanje (eng. *complete capture*), dostupni protokoli (eng. *available protocols*), raznovrsnost napada (eng. *attack diversity*), anonimnost (eng. *anonymity*), heterogenost (eng. *heterogeneity*), skup atributa (eng. *feature set*), metapodaci (eng. *metadata*). Nijedna od ranije dostupnih baza ne ispunjava sve kriterijume, pa je njihov cilj bio da generišu bazu koja će ih ispunjavati i tako je nastala baza *CIC-IDS2017*. Generisali su realističan pozadinski saobraćaj, izgradili su sistem koji simulira ponašanje 25 korisnika na osnovu HTTP, HTTPS, FTP, SSH i email protokola. Za potrebe generisanja ove baze napravljene su dve mreže — napadača i žrtve. Korišćeno je više mašina sa različitim operativnim sistemima, korišćeni su ruteri, svičevi. Prikupljanje podataka počelo je u ponedeljak, 3. jula 2017. u 9 ujutru i završilo se u petak 7. jula 2017. u 5 posle podne, što je ukupno pet dana prikupljanja podataka. Ponedeljak je normalan dan i uključuje samo benigni saobraćaj. Napadi uključuju Brute Force FTP, Brute Force SSH, DDoS, veb napade, skeniranje priključaka, infiltraciju, mrežu botova i sprovedeni su ujutru i posle podne u utorak, sredu, četvrtak i petak. Aktivnost korisnika je beležena pomoću programa *CICFlowMeter* koji iz toka izvlači potrebne atribute [15].

Baza ima skoro 2 700 000 redova i 79 kolona, u sebi sadrži 10 klasa — normalan saobraćaj i 9 napada. Najveći problem sa bazom je velika nebalansiranost klasa, nešto više od 2 300 000 instanci pripadaju normalnoj konekciji, a ostatak napadima, pri čemu i među napadima ima nebalansiranosti.

Glava 3

Mašinsko učenje

Mašinsko učenje (eng. *machine learning*) je oblast veštačke inteligencije i predstavlja moćan alat koji teži tome da se približi ljudskom učenju po efikasnosti. Danas se, u nekim domenima, njime postižu bolji rezultati od rezultata ljudskih eksperata. Mašinsko učenje pokušava da oponaša sposobnost učenja živih bića i da iz podataka izvuče šablone i algoritme, bez eksplicitnog programiranja, odnosno da od ograničenog broja uzoraka dođe do univerzalnih zaključaka [29].

Za različite vrste problema koriste se različite metode mašinskog učenja i osnovna podela tih metoda je upravo po problemu koji rešavaju, a to su problemi nadgledanog učenja (eng. *supervised learning*), problemi nenadgledanog učenja (eng. *unsupervised learning*) i problemi učenja potkrepljivanjem (eng. *reinforcement learning*). U ovom radu se koriste samo metode nadgledanog učenja, pa će o njima biti više reči u daljem tekstu [29].

3.1 Nadgledano učenje

Nadgledano učenje je vid učenja pri kome se modelu daju podaci iz kojih uči (ulaz) i ono što je potrebno naučiti (izlaz). Zove se nadgledano jer se posle učenja modelu daju tačna rešenja da proveri da li je dobro naučio ili ne. Potrebno je pronaći funkciju koja će ustanoviti vezu između ulaza i izlaza, tako da kasnije — kada mu bude dat samo ulaz — može da mu dodeli ispravan izlaz. Ove funkcije se u mašinskom učenju nazivaju modelima. Izlaz je obično jednodimenziona vrednost koja se naziva ciljna promenljiva. Dve osnovne vrste problema nadgledanog učenja su regresija (eng. *regression*) i klasifikacija (eng. *classification*). Regresija je problem predviđanja neprekidne ciljne promenljive, dok je klasifikacija problem predviđanja

kategoričke ciljne promenljive. S obzirom na to da se u ovom radu rukuje podacima koji imaju kategoričku ciljnu promenljivu, od interesa će nam nadalje biti samo klasifikacija [29].

Na prvi pogled može delovati da je potrebno naći savršen model — model koji uvek tačno klasifikuje raspoložive podatke i nikad ne greši, međutim, problem sa tim pristupom je što se ne može naći takav skup podataka koji je sveobuhvatan i obuhvata ceo prostor mogućnosti da bi se nad njim model obučavao. Obično se u stvarnosti dobije jedan deo mogućeg skupa podataka i na njemu obučava model. Ako se napravi savršen model, to vodi tome da se model potpuno prilagodio podacima koje ima i iz kojih je mogao da uči, što znači da kada u budućnosti naiđe na drugačiji oblik tih podataka neće moći da ih prepozna jer je izgubio moć generalizacije i preprilagodio se podacima. U mašinskom učenju ovo je jedan od najvećih i najčešćih problema, poznat kao preprilagođavanje (eng. *overfitting*) podacima. Ako se ode u drugu krajnost gde model previše greši i nije u stanju da dobro klasifikuje većinu instanci na kojima se trenira dolazimo do drugog problema u mašinskom učenju, poznatog kao potprilagođavanje (eng. *underfitting*) [29].

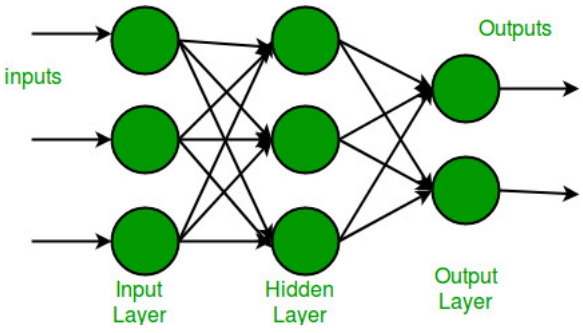
3.1.1 Slučajne šume

Pre zalaženja u metod slučajnih šuma (eng. *random forest*), neophodno je pojasniti metod stabala odlučivanja (eng. *decision tree*). Stabla odlučivanja predstavljaju jednostavan i interpretabilan model mašinskog učenja gde se problem klasifikacije može predstaviti stablom. Počevši od korena stabla za instancu iz test skupa se postavljaju pitanja o njenim atributima ili kombinaciji atributa i na osnovu odgovora, prateći odgovarajuću granu, postavljaju dalja pitanja dok se ne dođe do klase kojoj pripada. Pitanja, koja imaju više od jednog ishoda, su sadržana u čvorovima stabla — korenu i unutrašnjim čvorovima — a ne u listovima, jer se u listovima nalazi klasa koju model predviđa. Pitanja su uglavnom uslovna i odnose se na atribute instance, obično na jedan atribut, ali se mogu odnositi i na kombinaciju atributa. Svaki mogući ishod testa se vezuje za tačno jednu granu, odnosno tačno jedan dete-čvor. U slučaju kategoričkih atributa uglavnom se proverava jednakost sa nekom vrednošću atributa ili pripadnost nekom skupu vrednosti, dok se u slučaju neprekidnih atributa uglavnom proverava da li je vrednost atributa veća ili manja od neke konkretne vrednosti. Određivanje vrednosti koja će pripasti listovima u sličaju regresije je uglavnom prosek vrednosti, dok je u slučaju klasifikacije većinska klasa. [29, 45].

Ansambl je skup većeg broja modela koji zajednički donose odluke. Slučajne šume poboljšavaju performanse stabala odlučivanja tako što grade ansambl stabala koja nisu mnogo korelisana. Ansambl se konstruiše od m stabala, pri čemu je svako stablo trenirano na različitom podskupu za obučavanje ili na različitom skupu atributa. Obučavanje na različitim podskupovima instanci ili atributa se radi da bi greške stabala bile što slabije korelisane. U praksi se pokazalo da se za velike vrednosti parametra m, koji predstavlja broj stabala, model bolje ponaša, jer smanjuje preprilagođavanje. U obzir treba uzeti i to da više stabala za obučavanje podrazumeva i više utrošenog vremena za obučavanje modela. Za razliku od stabala odlučivanja, slučajne šume nisu interpretabilan model [29, 45].

3.1.2 Potpuno povezana neuronska mreža

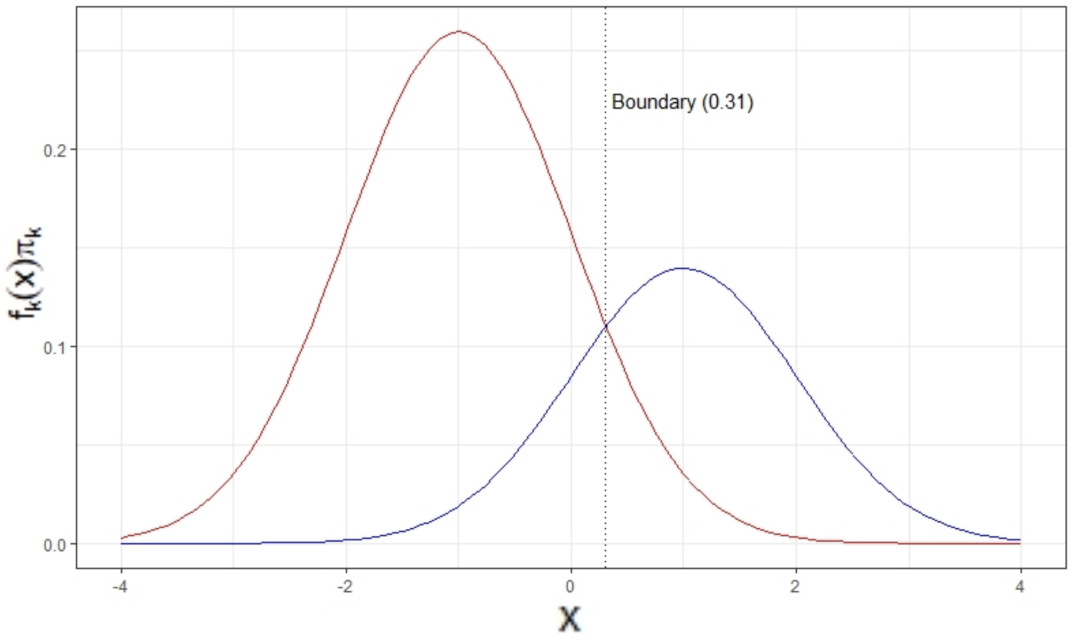
Veštačka neuronska mreža (eng. *artificial neural network — ANN*) je jedna od najpopularnijih metoda mašinskog učenja. Postoje razne vrste neuronskih mreža, u ovom radu će akcenat biti na potpuno povezanim mrežama (eng. *fully connected*). Mreže se sastoje od neurona, odnosno računskih jedinica koje predstavljaju parametrizovane funkcije. Svaki neuron računa linearnu kombinaciju svojih atributa a potom nad njom izvršava aktivacionu funkciju (eng. *activation function*) koja predstavlja nelinearnu transformaciju. Neuroni su postavljeni u slojeve, postoje ulazni i izlazni sloj kao i skriveni slojevi koji se nalaze između ulaznog i izlaznog. Neuroni ulaznog sloja primaju vrednosti, vrše tranformacije nad njima i potom ih prosleđuju sledećem sloju. Svaki sledeći sloj prosleđuje narednom dok se ne dođe do izlaznog sloja. Neuron svoj izlaz prosleđuje svakom od neurona narednog sloja, otud i naziv potpuno povezane neuronske mreže. Na slici 3.1 se može videti vizuelni prikaz neurona raspoređenih u slojeve i tok rezultata transformacija od jednog neurona ka drugima [29].



Slika 3.1: Potpuno povezana neuronska mreža

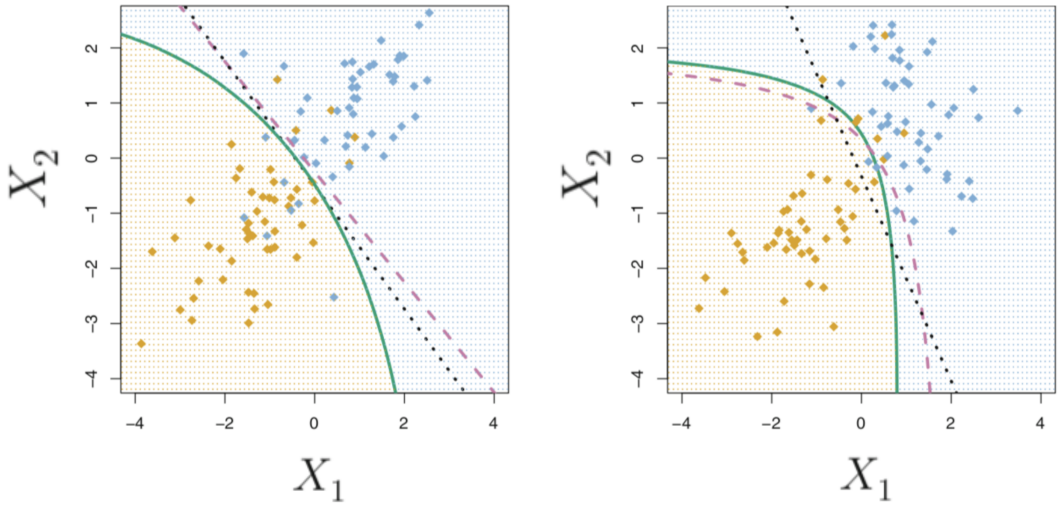
3.1.3 Kvadratna diskriminantna analiza

Kvadratna diskriminantna analiza (eng. *quadratic discriminant analysis — QDA*) je klasifikator koji polazi od pretpostavke da svaka klasa ima normalnu raspodelu i za svaku klasu računa očekivanje i matricu kovarijansi posebno. Očekivanje klase k se računa kao prosek podataka iz trening skupa za klasu k. Kvadratna funkcija predstavlja granicu odlučivanja koja razdvaja dve klase, tu kvadratnu funkciju definišu one instance koje imaju istu verovatnoću da pripadaju obema klasama. Na slici 3.2 se može videti grafik na kome se nalaze normalne raspodele dve različite klase.



Slika 3.2: Normalne raspodele dve klase i linearna granica odlučivanja

U tački $x = 0.31$ se nalazi granica odlučivanja jer je tu verovatnoća ista da ta instanca pripada crvenoj ili plavoj klasi. U ovom primeru granica odlučivanja je linearna funkcija što odgovara linearnoj diskriminantnoj analizi, a primer kvadratne granice se može videti na slici 3.3. Ovaj algoritam je pogodan jer se lako izračunava, u praksi se dobro pokazao i nema metaparametre [39, 48].



Slika 3.3: Normalne raspodele dve klase i granica odlučivanja

3.1.4 XGBoost

Kada se gradi ansambl modela prilikom formiranja modela moguće je uzeti u obzir ponašanje drugih modela, odnosno moguće je ansambl graditi dodajući model po model, tako da se svaki od modela obučava da nadomesti slabosti trenutnog skupa modela. Ova metoda se zove pojačavanje (eng. *boosting*), jer se svaki model gradi tako da pojača prethodne modele. XGBoost (eng. *extreme gradient boosting*) je biblioteka za gradijentno pojačavanje koja je namenjena za efikasno i skalabilno obučavanje modela. Gradijentno pojačavanje (eng. *gradient boosting*) se oslanja na gradijentne metode optimizacije koje popravljaju trenutno rešenje optimizacionog problema tako što dodaju vektor proporcionalan negativnoj vrednosti gradijenta funkcije koja se minimizuje. XGBoost je jedan od najpopularnijih i najprimenjenijih algoritama mašinskog učenja jer može da radi sa velikim bazama podataka i daje odlične rezultate za regresione i klasifikacione probleme [29, 19].

3.1.5 Multinomijalna logistička regresija

Logistička regresija (eng. *logistic regression*) je model binarne klasifikacije. Ukoliko imamo nezavisne promenljive X i zavisnu promenljivu y, uslovna verovatnoća se računa na sledeći način:

$$P(y = 1|X) = \frac{e^{\beta_0 X_0 + \beta_1 X_1 + \ldots \beta_p X_p}}{1 + e^{\beta_0 X_0 + \beta_1 X_1 + \ldots \beta_p X_p}}$$

$$P(y = 0|X) = \frac{1}{1 + e^{\beta_0 X_0 + \beta_1 X_1 + ... \beta_p X_p}}$$

Gde su $X_0, X_1, ..., X_p$ atributi podataka. U hipotezi se koristi sigmoidna funkcija koja je izabrana jer transformiše interval $[-\infty, \infty]$ u interval [0, 1]. Cilj logističke regresije je da izabere ocene $\hat{\beta}_1, \hat{\beta}_2, ..., \hat{\beta}_p$ parametara $\beta_1, \beta_2, ..., \beta_p$ koje maksimizuju verovatnoću

$$\prod_{i=1}^{n} P(Y = y_i|X = x_i)$$

Pronalaženje ocena se vrši numeričkim metodama [1].

Multinomijalna logistička regresija (eng. *multinomial logistic regression*) se gradi nad logističkom regresijom i predstavlja metod višeklasne klasifikacije. Ukoliko se u podacima nalazi K klasa, onda se pravi K-1 model logističke regresije u kome se jedna klasa fiksira a ostalih K-1 klasa se odvojeno upoređuje sa njom. Cilj multinomijalne logističke regresije je naći matricu ocena parametra β čije su dimenzije $p \times (K - 1)$ [1].

3.2 Pretprocesiranje podataka

Retko kad možemo direktno očitane, sirove podatke da ubacimo u modele mašinskog učenja. Modeli su osetljivi na određene osobine koje se mogu javiti u podacima, zato je neophodno razumeti kako model radi da bismo mogli ispravno da pripremimo podatke za korišćenje u modelu. Ako se podaci ne pripreme kako treba vrlo je verovatno da model neće raditi ispravno ili efikasno. Ovo je poznato kao pretprocesiranje podataka i odnosi se na transformaciju podataka na različite načine pre nego što se prosledi algoritmima mašinskog učenja [29].

3.2.1 Transformacija podataka

Prilikom izvršavanja algoritama u računskim operacijama figurišu vrednosti atributa. Ukoliko je opseg jednog atributa veći od opsega drugog, neki koeficijenti kod linearnih modela mogu biti veliki samo zbog opsega u kome se promenljiva nalazi, čime se gubi interpretabilnost i numerička stabilnost. Ako model koristi regularizaciju, može se desiti da ona umanji koeficijent važnog atributa samo zato što je predstavljen na manjoj skali pa mu je koeficijent veći od ostalih, a to nema veze sa

odnosom među promenljivama. Zato se koristi standardizacija podataka. Standardizacija je svođenje srednje vrednosti atributa na 0, a standardne devijacije na 1. Time se vrednosti atributa svode na isti red veličine [29].

Prilikom rada sa podacima posebnu pažnju treba obratiti na *NULL* vrednosti. *NULL* je zapravo odsustvo vrednosti. Ukoliko prilikom prikupljanja podataka neka vrednost nije zabeležena, na njeno polje staviće se *NULL* da označi njeno odsustvo. Postavlja se pitanje kako *NULL* vrednosti integrisati u matematički račun prilikom izvođenja algoritama. Rešenje za to je da se one u podacima ili popunjavaju određenom metodom (srednja vrednost, medijana, najčešća vrednost, ...) ili da se podaci izbrišu. Na sličan način se posmatraju $-\infty$ i $+\infty$ [29].

3.2.2 Smanjenje dimenzionalnosti podataka

Smanjenje dimenzionalnosti podataka se koristi da bi se smanjila složenost algoritma i povećala brzina izvršavanja. Smanjuje se prokletstvo dimenzionalnosti. Takođe, može se koristiti za uklanjanje korelisanih atributa.

Metod glavnih komponenti (eng. *principal component analysis — PCA*) je jedna od najprimenjenijih tehnika za smanjenje dimenzionalnosti. Koriste se metode linearne algebre da bi se podaci iz visoko dimenzionalnog prostora projektovali na manje dimenzionalni prostor. Visoka korelisanost atributa znači da nisu svi atributi važni i da neki od njih mogu približno izraziti kao funkcije drugih atributa. Metod glavnih komponenti pronalazi nove atribute koji su linearna kombinacija originalnih atributa, međusobno su ortogonalni i obuhvataju maksimalnu količinu varijacije u podacima. Na taj način ne samo da se smanjuje dimenzionalnost podataka i ubrzava obučavanje metoda mašinskog učenja, nego se i smanjuju negativni efekti koju visoka dimenzionalnost i korelisanost atributa imaju na statističke aspekte tih metoda [29, 45].

3.2.3 Uzorkovanje podataka

Ukoliko se radi sa klasama koje su nebalansirane, odnosno jedna klasa ima više instanci od druge, može doći do lošijeg predviđanja modela, jer veća klasa više utiče na formiranje modela što dovodi do toga da model češće predviđa veću klasu od manje. Postoji nekoliko metoda za balasiranje podataka. Jedan metod je smanjenje uzorka (eng. *undersampling*) i to je tehnika kojom se instance iz većinske klase izbacuju da bi se postigla balansiranost, mana ove metode je što se mogu izgubiti

značajne informacije. Drugi metod je povećanje uzorka (eng. *oversampling*) kojim se povećava broj instanci manjinske klase, mana ove metode je što se u podatke unose instance koje nisu stvarne tako da se potencijalno ubacuju lažne informacije. Neki od metoda smanjenja uzorka su algoritam *NearMiss* verzije 1, 2 i 3, koje će u daljem tekstu biti označene kao *NearMiss-1*, *NearMiss-2* i *NearMiss-3*, i algoritam *AllKNN* [29].

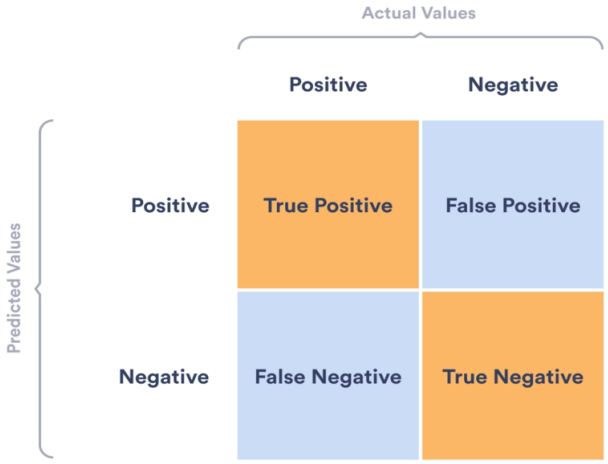
Algoritam *NearMiss-1* bira uzorke iz većinske klase za koje je prosečno odstojanje do najbližih suseda najmanje. *NearMiss-2* bira uzorke iz većinske klase za koje je prosečno odstojanje od najudaljenijih suseda najmanje. *NearMiss-3* se može podeliti u 2 koraka. Prvo, metod najbližih suseda se koristi da bi se smanjila lista uzoraka iz većinske klase, nakon toga se bira uzorak sa najvećim prosečnim odstojanjem od k najbližih suseda. *EditedNearestNeighbours* koristi algoritam k najbližih suseda da bi izbacio instance koje se nalaze na granici odlučivanja. Algoritam *AllKNN* pokreće *ENN* nekoliko puta pri čemu se varira broj najbližih suseda [26, 25].

SMOTE (eng. *synthetic minority over-sampling technique*) je algoritam koji uvećava broj podataka tako što formira sintetičke instance na osnovu originalnih instanci. Prednost algoritma *SMOTE* je ta što se instance ne dupliraju, nego se generišu novi podaci koji su malo drugačiji od originalnih. Algoritam *SMOTE* radi tako što izabere nasumični uzorak iz manjinske klase, identifikuje k najbližih suseda, izabere jednog od suseda i napravi vektor između trenutne instance i izabranog suseda, potom vektor pomnoži nasumičnim brojem između 0 i 1 i na kraju se taj vektor sabere sa originalnom instancom [37].

3.3 Mere kvaliteta modela

Mere kvaliteta modela se koriste da bi se utvrdilo koliko je model dobar u odnosu na konkretan skup podataka. Mogu se koristiti razne mere i mogu zavisiti od konkretnog problema, ali se za klasifikaciju najčešće koristi matrica konfuzije.

U binarnoj klasifikaciji imamo dve klase, najčešće se nazivaju pozitivna i negativna klasa. Cilj je napraviti model koji će korektno klasifikovati podatke, odnosno, podatke pozitivne klase dodeliti pozitivnoj klasi, a podatke negativne negativnoj klasi. Model greši ukoliko instance koje zapravo pripadaju pozitivnoj klasi dodeli negativnoj klasi i obrnuto. Da bi se video broj tačno i netačno klasifikovanih instanci koristi se matrica konfuzije (eng. *confusion matrix*) koja se može videti na slici 3.4 [29].



Slika 3.4: Matrica konfuzije

Polja u matrici se tumače na sledeći način [29]:

Stvarno pozitivne (eng. *true positive — TP*) instance su pozitivne instance koje je model klasifikovao kao pozitivne.

Stvarno negativne (eng. *true negative — TN*) instance su negativne instance koje je model klasifikovao kao negativne.

Lažno pozitivne (eng. *false positive — FP*) instance su negativne instance koje je model klasifikovao kao pozitivne.

Lažno negativne (eng. *false negative — FN*) instance su pozitivne instance koje je model klasifikovao kao negativne.

Na dijagonali matrice konfuzije se nalaze tačno klasifikovane vrednosti, pa se prilikom formiranja modela teži tome da te vrednosti budu što veće. Na osnovu vrednosti u matrici konfuzije izvode se mere kvaliteta klasifikacije koje se često koriste. Te mere su [29]:

Tačnost (eng. *accuracy*) je mera koja predstavlja udeo tačno klasifikovanih instanci u ukupnom broju instanci. Iako je veoma intuitivna mera i deluje kao odličan indikator kvaliteta modela, tačnost može da zavara kada je u pitanju model koji se trenira na nebalansiranom skupu. Naime, ako jedna klasu čini 99% instanci i ako model dodeli sve instance toj klasi, onda će model biti tačan u 99% slučajeva. U tom slučaju tačnost nam daje informaciju da je model izuzetno dobar, a on je u stvari

potpuno beskoristan, jer ne radi ono što je njegova osnovna uloga — da razlikuje dve klase.

$$\text{Tačnost (Acc)} = \frac{TP + TN}{TP + TN + FP + FN}$$

Preciznost (eng. *precision*) je mera koja predstavlja udeo tačno klasifikovanih pozitivnih instanci u instancama koje je model proglasio pozitivnim. Odnosno, koliko je model bio precizan kada je neku instancu označio pozitivnom.

$$\text{Preciznost (Prec)} = \frac{TP}{TP + FP}$$

Odziv (eng. *recall*) je mera koja predstavlja udeo tačno klasifikovanih pozitivnih instanci među svim zaista pozitivnim instancama.

$$\text{Odziv (Rec)} = \frac{TP}{TP + FN}$$

F1 mera (eng. *F1*) je mera koja predstavlja harmonijsku sredinu preciznosti i odziva. Preciznost i odziv ne treba posmatrati zasebno, jer, takođe, mogu da zavaraju. Ako model proglasi sve instance za pozitivne, znači da nijednu nije proglasio negativnom pa će FN (lažno negativni) biti 0, time se postiže maksimalan odziv. Ako model ne proglasi nijednu klasu za pozitivnu ne pravi se nijedna greška. Samim tim je potrebno posmatrati ove dve mere zajedno i u te svrhe se najčešće koristi F1 mera.

$$\text{F1} = 2 * \frac{Preciznost * Odziv}{Preciznost + Odziv}$$

3.4 Tehnike evaluacije

Postoji više različitih tehnika izbora i evaluacije modela mašinskog učenja koji se razlikuju po složenosti. Neke od tehnika uključuju evaluaciju pomoću skupa za testiranje, K-slojnu unakrsnu validaciju i evaluaciju pomoću skupova za validaciju i testiranje. Koja god tehnika da se koristi ne sme se narušiti glavno pravilo evaluacije a to je da se podaci korišćeni u evaluaciji modela nikako ne smeju koristiti u njegovom obučavanju. U daljem tekstu će biti obrađena samo tehnika evaluacije pomoću skupova za validaciju i testiranje [29].

Kao što je već pomenuto, klasifikacija pripada nadgledanom učenju, što znači da model uči na određenim podacima, potom testira svoje znanje na nepoznatim

podacima i na kraju dobije tačna rešenja da proveri koliko je dobro naučio. Ovo se postiže korišćenjem trening i test skupova. Naime, podaci se obično dele na skup za obučavanje (eng. *training*) i test (eng. *test*) skup i uglavnom se veći deo podataka koristi za obučavanje. Nakon obučavanja, modelu se prosleđuju test podaci bez oznake klase i model ih klasifikuje. Potom se klase koje je dodelio model upoređuju sa stvarnim klasama modela čime se dobija ocena modela. Ukoliko rezultati nisu zadovoljavajući mogu se promeniti neki metaparametri u modelu i početi proces ispočetka u nadi za boljim rezultatima [29].

Ovaj pristup nosi sa sobom jedan problem — nakon što se model testira koriste se ti rezultati da bi se donela odluka kako izmeniti metaparametre modela. Time se narušava osnovno pravilo evaluacije modela, a to je da se test skup ni na koji način ne može koristiti u treniranju modela. U tu svrhu se uvodi još jedan skup — validacioni (eng. *validation*). Validacioni skup preuzima dosadašnju ulogu test skupa i služi proveri učenja modela i donošenju odluka o metaparametrima modela. Nakon pronalaska metaparametara koji daju zadovoljavajuće rezultate se testira model skupom za testiranje i daje se finalna ocena o modelu [29].

Glava 4

Eksperimentalna evaluacija

U ovom poglavlju će biti detaljnije predstavljeni podaci nad kojima se radi, njihovo pretprocesiranje i eksperimentalna evaluacija koja uključuje formiranje modela mašinskog učenja i tumačenje njihovih rezultata. Ispitani su modeli predloženi od strane autora baze, odnosno, slučajne šume, potpuno povezana neuronska mreža (u daljem tekstu neuronska mreža), kvadratna diskriminantna analiza, i isprobani novi koji uključuju XGBoost i multinomijalnu logističku regresiju.

Za eksperimentalnu evaluaciju će biti korišćen programski jezik *Python*, interaktivna platforma *Jupyter Notebook* i biblioteke namenjene za rad sa podacima koje uključuju *pandas, numpy, matplotlib, sklearn, imblearn* i *xgboost*.

Osobine računara na kome se izvršava eksperimentalna evaluacija su:

1. Procesor: *Intel(R) Core(TM) i5-4300U CPU @ 1.90GHz, 2501 Mhz, 2 Cors, 4 Logical Processors*
2. RAM: *8GB*
3. OS: *Microsoft Windows 10 Pro*

4.1 Upoznavanje sa podacima i pretprocesiranje

Baza je dostupna u obliku direktorijuma sa 8 datoteka:

1. Monday-WorkingHours.pcap_ISCX.csv
2. Tuesday-WorkingHours.pcap_ISCX.csv
3. Wednesday-workingHours.pcap_ISCX.csv

4. Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv
5. Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX.csv
6. Friday-WorkingHours-Morning.pcap_ISCX.csv
7. Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv
8. Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv

Svaka od ovih *CSV* datoteka u sebi sadrži podatke klasifikovane na benignu klasu i jednu vrstu napada. Nakon učitavanja svih datoteka one se spajaju u jednu *CSV* datoteku pod imenom *data.csv* koja ima 2 667 958 redova i 79 kolona, od toga su 78 kolona atributi podataka, a preostala kolona, pod imenom *Label* predstavlja klasu.

U tabeli 4.1 se mogu videti nazivi klasa i broj instanci koje pripadaju tim klasama.

Tabela 4.1: Raspodela broja instaci po klasama

BENIGN	2 362 984
PortScan	158 930
DDoS	128 027
FTP-Patator	7 938
SSH-Patator	5 897
Bot	1 966
Web Attack Brute Force	1 507
Web Attack XSS	652
Infiltration	36
Web Attack Sql Injection	21

S obzirom na to da će se na podatke primenjivati numeričke operacije, potrebno je obratiti pažnju da li se u podacima nalaze *NULL* vrednosti ili beskonačne vrednosti. *NULL* predstavlja odsustvo vrednosti, to znači da nije zabeležena vrednost nekog atributa neke instance i na prazno polje je postavljen *NULL*. U podacima ima 414 *NULL* vrednosti i 3600 vrednosti $+/-\infty$.

Tabela 4.2: Raspodela broja instaci po klasama nakon spajanja nekih klasa

BENIGN	2 362 984
PortScan	158 930
DDoS	128 027
Brute Force	13 835
Web Attack	2 180
Bot	1 966
Infiltration	36

Pre same primene modela mašinskog učenja na podatke, neophodno je izvršiti određene transformacije podataka da bi modeli mogli biti, pre svega, ispravni, potom precizniji i efikasniji.

S obzirom na to da se formiraju modeli višeklasne klasifikacije, potrebno je ostaviti oznaku klase. Međutim, u podacima je prisutna velika nebalansiranost. Da bi se nebalansiranost umanjila manje klase su spojene u jednu veću. Naravno, klase moraju biti međusobno slične, odnosno, moraju pripadati nekoj većoj grupi sajber napada. *Web Attack Brute Force*, *Web Attack XSS* i *Web Attack Sql Injection* se spajaju u jednu klasu pod imenom *"Web Attack"*. *FTP-Patator* i *SSH-Patator* se spajaju u klasu pod imenom *"Brute Force"*. Ovim se dobija 7 klasa i raspodela broja instanci po klasama je prikazana u tabeli 4.2.

Kao što je već spomenuto, u podacima ima 414 *NULL* vrednosti i 3600 vrednosti $+/-\infty$. Potrebno je ukloniti te vrednosti iz podataka da ne bi dolazilo do problema sa numeričkim izračunavanjima kasnije u radu. Vrednosti $-\infty$ i $+\infty$ se zamenjuju *NULL* vrednostima, a potom se brišu sve instance koje u sebi imaju *NULL*, kojih je nakon zamene bilo ukupno 4014. Ciljna klasa se izdvaja u posebnu promenljivu y, dok se ostali atributi smeštaju u promenljivu X. Atributi su većinom tipa *int64*, ostatak je *float64* i oznaka klase *Label* je tipa *object*. Svi atributi, sem ciljne klase, se kastuju u tip *float64*, dok se ciljna klasa kastuje u tip *category*, a potom kodira u vrednosti od 0 do 6, koje odgovaraju klasama. Iz tabele 4.3 može se zaključiti koji kod odgovara kojoj klasi.

Skup podataka se prvo deli na trening i test skup u odnosu 80 : 20, pri čemu se vodi računa o stratifikaciji y. Potom se trening skup deli na trening i validacioni skup u odnosu 80 : 20, pri čemu se, takođe, vodi računa o stratifikaciji y. Nakon

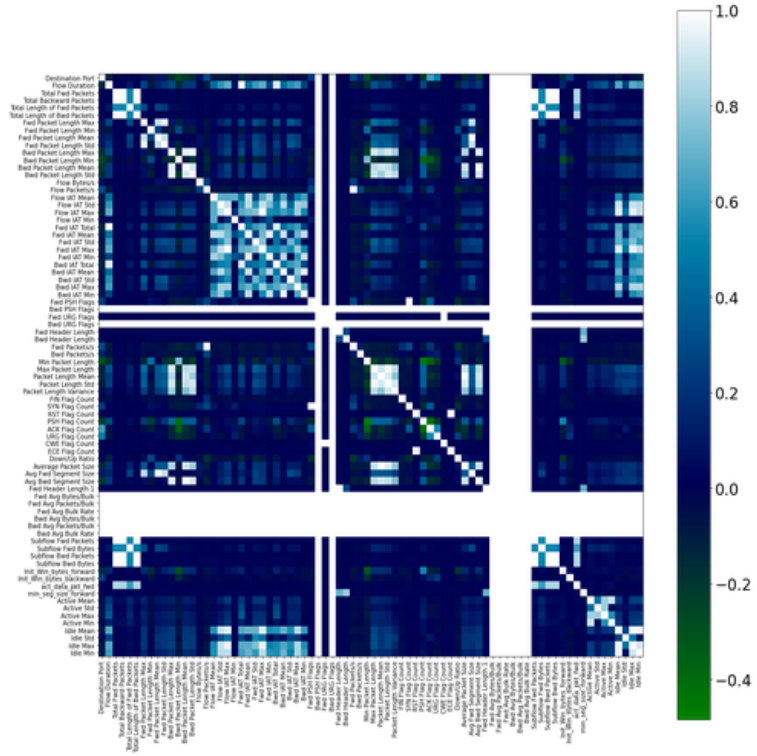
Tabela 4.3: Ime klase i odgovarajuća kodirana vrednost

BENIGN	0
PortScan	5
DDoS	3
Brute Force	2
Web Attack	6
Bot	1
Infiltration	4

ovoga su trening, validacioni i test skup u razmeri 64 : 16 : 20, tim redom. Ostala zaključivanja koja su nam od značaja u radu će biti donošena isključivo na trening skupu. Prvo se posmatra matrica korelacije na trening skupu. Na slici 4.1 se može videti vizuelni prikaz matrice korelacije. Belom bojom je označena korelacija u vrednosti 1, što je najveća vrednost. Uočava se da su neki atributi u maksimalnoj korelaciji sa neki drugim atributima. Ti atributi su: *' Bwd PSH Flags', ' Bwd URG Flags', 'Fwd Avg Bytes/Bulk', ' Fwd Avg Packets/Bulk', ' Fwd Avg Bulk Rate', ' Bwd Avg Bytes/Bulk', ' Bwd Avg Packets/Bulk'* i *'Bwd Avg Bulk Rate'* i oni se izbacuju iz trening, validacionog i test skupa. Time je dimenzionalnost podataka smanjena sa 78 kolona na 70.

Osim ovih atributa koji su izbačeni, postoji još 103 para atributa koji su međusobno u visokoj korelaciji, odnosno, korelacija im je veća od 0.8 ili manja od -0.8. Zbog toga će u nastavku rada biti korišćen algoritam *PCA*, kojim će se ne samo smanjiti dimenzionalnost i broj atributa, već će se takođe ukloniti i visoke korelacije.

Pre upotrebe *PCA*, neophodno je standardizovati podatke. Ne samo zbog već spomenute potrebe za standardizacijom podataka, nego i zbog toga što je *PCA* osetljiv na skale. Metodom *StandardScaler()* iz biblioteke *preprocessing*, koji se obučava isključivo na trening podacima, transformišu se trening, validacioni i test skup. Zbog velikog broja korelisanih atributa primenjuje se algoritam *PCA*. Potrebno je naći odgovarajuću vrednost broja komponenti na koje će se smanjiti dimenzionalnost da bi se uklonila korelisanost i ujedno očuvala količina informacija. U tu svrhu je isprobano smanjivanje dimenzionalnosti sa 70 kolona na 50, 40, 30, 20 i 10 kolona, pri čemu je očuvano 99.98%, 99.81%, 98.24%, 90% i 70.51% varijanse, tim redom.

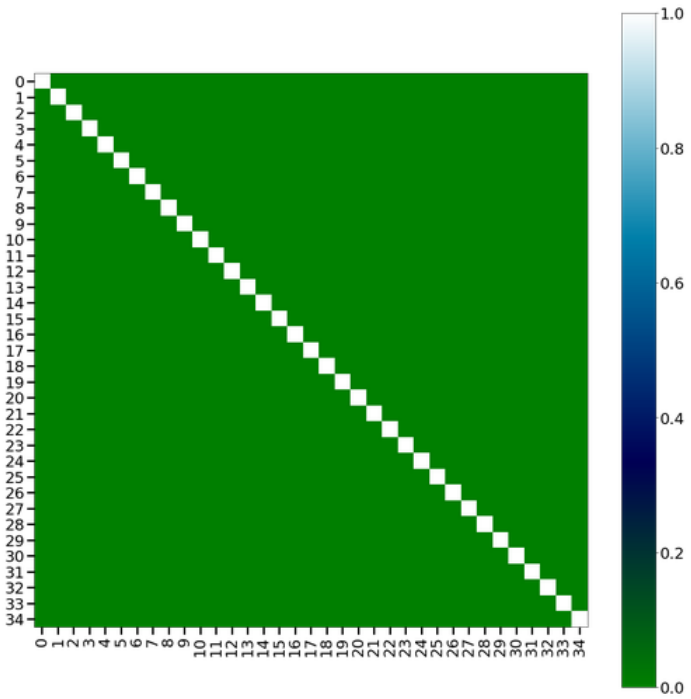


Slika 4.1: Matrica korelacije

Na osnovu ovih brojki zaključeno je da je optimalan broj negde između 30 i 40 atributa pa je isprobana vrednost 35 atributa koja je očuvala 99.4% informacija i ta vrednost je izabrana za transformaciju podataka. *PCA* je obučavan isključivo na trening skupu i onda je primenjen na trening, validacioni i test skup. Matrica korelacija nakon primene algoritma *PCA* se može videti na slici 4.2.

4.2 Višeklasna klasifikacija

Autori *CIC-IDS2017* baze podataka su primenili 7 različitih modela mašinskog učenja da bi pronašli koji je najbolji model po težinskom proseku preciznosti, odziva i F1 mere i brzine izvršavanja obučavanja i predviđanja. Modeli uključuju k najbližih suseda, slučajne šume, ID3, Adaboost, neuronsku mrežu, naivni Bajesov klasifikator i kvadratnu diskriminantnu analizu, od kojih su prema težinskom proseku preciznosti, odziva i F1 najbolje rezultate dali k najbližih suseda, slučajne šume i ID3. Nešto slabije rezultate su dali Adaboost, neuronska mreža i kvadratna diskriminantna analiza, dok se naivni Bajesov klasifikator nije pokazao kao dobar



Slika 4.2: Matrica korelacije nakon primene algoritma PCA(n_components=35)

izbor. Od prva tri najbolja slučajne šume imaju ubedljivo najkraće vreme izvršavanja tako da su proglašene najboljim modelom od strane autora baze. U ovom radu cilj je ispitati kvalitet modela slučajnih šuma nad ovom bazom, pokušati poboljšati rezultate neuronske mreže i kvadratne diskriminantne analize, kao i isprobati modele XGBoost i multinomijalnu logističku regresiju. Modeli k najbližih suseda, ID3 i Adaboost se neće ispitivati zbog velikog vremena izvršavanja i hardverskih ograničenja. Jedan od ciljeva, takođe, je da se pronađe što kvalitetniji model nad što manjim skupom podataka da bi izvršavanje bilo brže i efikasnije, ali da se ne izgubi previše na kvalitetu.

U cilju pronalaženja najkvalitetnijeg modela potrebno je naći najbolju kombinaciju balansiranja podataka i modela. U ovom radu su isprobane različite metode balansiranja podataka i različit broj instanci po klasama nakon balansiranja. Nad tim kombinacijama su potom primenjeni različiti modeli mašinskog učenja sa različitim metaparametrima, u potrazi za kombinacijom koja daje najbolje rezultate.

Pre primene samih modela, potrebno je balansirati podatke. Balansiranje podataka se vrši isključivo nad trening skupom u kome ima 1 706 208 instanci. Pre samog balansiranja, raspodela instanci po klasama u trening skupu se može videti u tabeli 4.4.

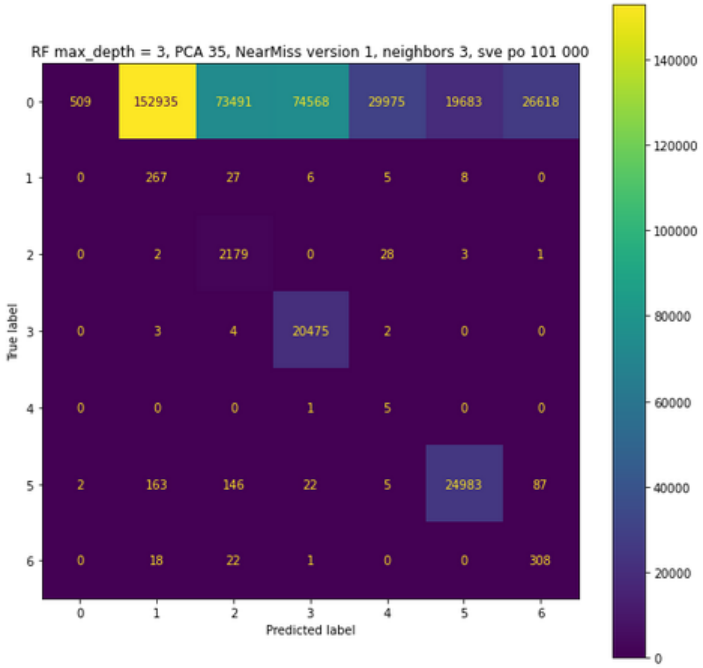
Tabela 4.4: Raspodela broja instaci po klasama u trening podskupu

0	1 511 115
5	101 635
3	81 936
2	8 852
6	1 395
1	1 252
4	23

Jedna od prvih kombinacija balansiranja podataka koja je isprobana uključuje primenu algoritma *NearMiss-1* za smanjenje broja instanci i algoritma *SMOTE* za povećavanje broja instanci, čime je dobijeno oko 101 000 instanci svake klase. Za *NearMiss-1* korišćeni su metaparametri *neighbors = 3* i *sampling_strategy = „majority"* čime je ciljana samo benigna klasa za smanjenje broja instanci. Algoritam *NearMiss-1* je smanjio broj instanci benigne klase sa 1 511 115 na svega 23 instance što je broj instanci najmanje klase, i time je 5. klasa postala najbrojnija sa 101 635 instanci. Vodeći se tim brojkama pokrenut je algoritam *SMOTE* kome je kao cilj zadato da svaka klasa ima oko 101 000 instanci. Nakon tako zadate strategije balansiranja isprobano je nekoliko modela mašinskog učenja:

- Slučajne šume sa metaparametrima *max_depth = 3* i *n_estimators = 100*
- Slučajne šume sa metaparametrima *max_depth = 5* i *n_estimators = 100*
- Slučajne šume sa metaparametrima *max_depth = 7* i *n_estimators = 100*
- Slučajne šume sa metaparametrima *max_depth = 9* i *n_estimators = 100*
- Slučajne šume sa metaparametrima *max_depth = 11* i *n_estimators = 100*
- Slučajne šume sa metaparametrima *max_depth = None* i *n_estimators = 200*
- Slučajne šume sa metaparametrima *max_depth = None* i *n_estimators = 400*
- XGBoost sa metaparametrom *n_estimators = 100*
- XGBoost sa metaparametrom *n_estimators = 400*

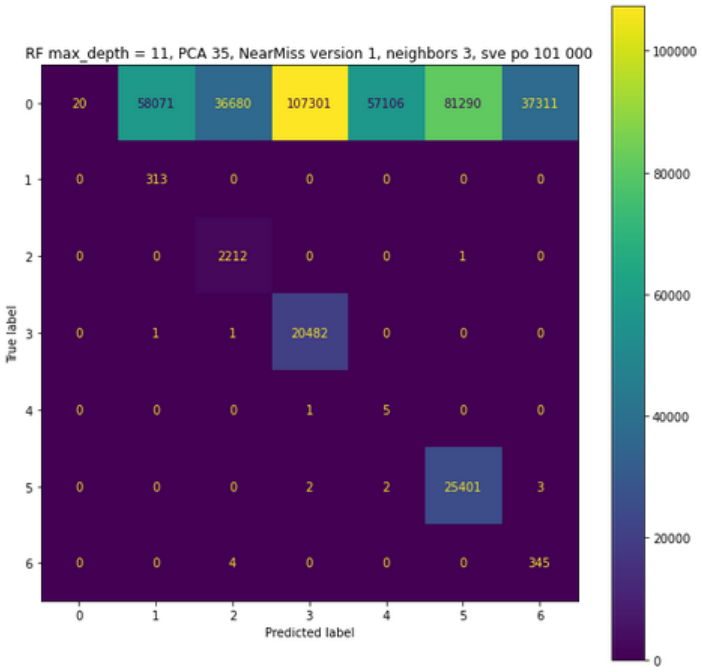
Ako se pogleda matrica konfuzije prvog isprobanog modela, odnosno, slučajne šume sa metaparametrima *max_ depth = 3* i *n_ estimators = 100*, slika 4.3, može se primetiti da je nulta, odnosno benigna, klasa veoma loše klasifikovana. Napadi su bolje klasifikovani, mada se vidi da ima određen broj instanci napada koji je pogrešno klasifikovan. Tačnost modela je 0.11 što je veoma loše.



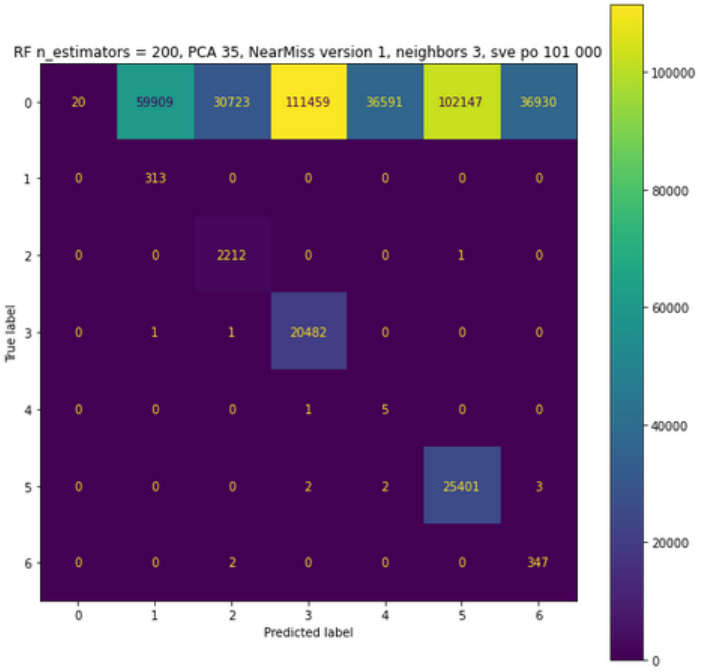
Slika 4.3: Matrica konfuzije za model slučajne šume, max_depth = 3, n_estimators = 100

Kako se povećava parametar *max_ depth* kod algoritma slučajnih šuma tako se napadi sve bolje klasifikuju, odnosno, matrica konfuzije ispod prvog reda postaje sve koncentrisanija oko dijagonale, ali benigna klasa je sve lošije klasifikovana. Matrica konfuzije modela slučajnih šuma sa metaparametrima *max_ depth = 11* i *n_ estimators = 100* se može videti na slici 4.4. Tačnost modela je i dalje 0.11.

Ukoliko bi parametar *max_ depth* bio *None*, a povećao se broj stabala na 200, onda se dobija jako slična matrica konfuzije kao i za *max_ depth = 11*, *n_ estimators = 100*, što se može videti na slici 4.5. I dalje je jako loše zbog benigne klase i tačnosti koja opet iznosi 0.11.



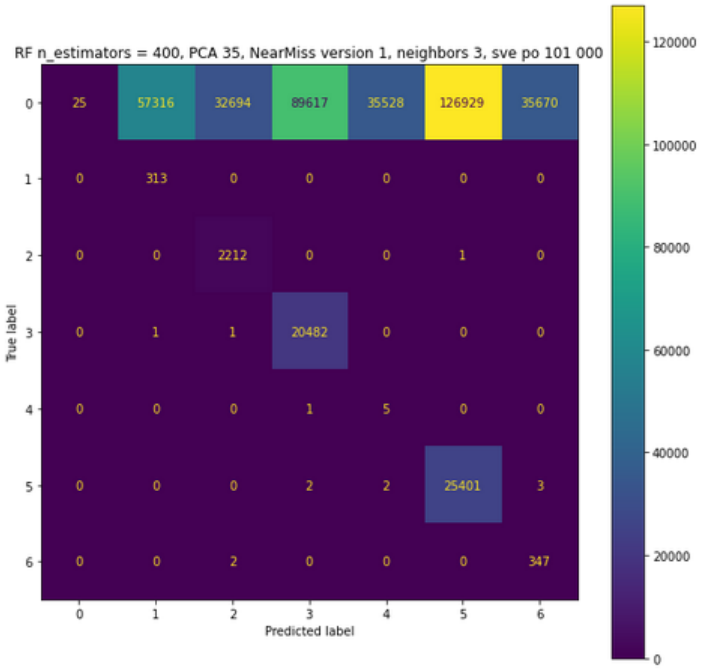
Slika 4.4: Matrica konfuzije za model slučajne šume, max_depth = 11, n_estimators = 100



Slika 4.5: Matrica konfuzije modela slučajnih šuma, max_depth = None, n_estimators = 200

Na slici 4.6 vidi se matrica konfuzije kada se broj stabala poveća na 400. Ne uočava se neka razlika u odnosu na prethodne modele i tačnost ostaje nepromenjena.

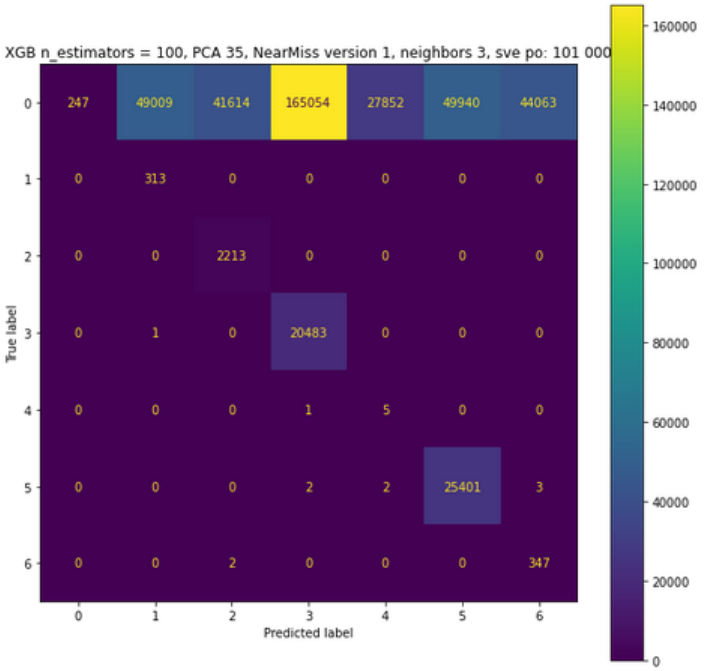
Zaključuje se da model slučajnih šuma sa ovakvim balansiranjem ne može da da dobre rezultate.



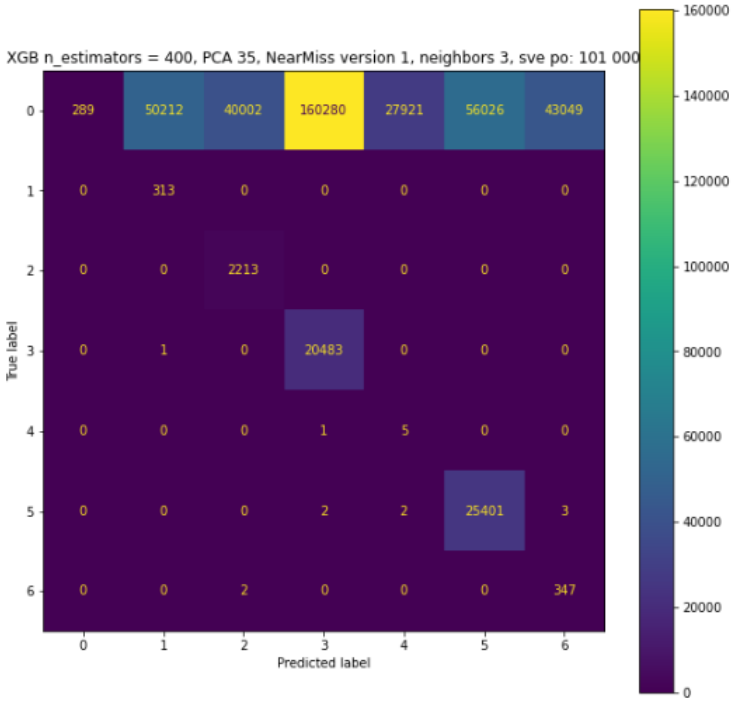
Slika 4.6: Matrica konfuzije modela slučajnih šuma, max_depth = None, n_estimators = 400

Sledeći model klasifikacije je XGBoost sa metaparametrima *n_ estimators = 100* na slici 4.7, tačnost modela je 0.11. Na slici 4.8 se vidi matrica konfuzije ukoliko se poveća broj stabala na 400. Može se primetiti veoma malo poboljšanje u predviđanju benigne klase, ali tačnost ostaje nepromenjena. Nijedan od ispitanih modela nije dao zadovoljavajuće rezultate. Napadi su dobro klasifikovani, ali najveća greška u predviđanju se uočava kod benigne klase, veoma mali broj instanci je ispravno klasifikovan, većina benignih konekcija je proglašena nekim napadom.

Od prethodnih 9 modela najbolje rezultate je dao XGBoost sa metaparametrom *n_ estimators = 400*. Na slici 4.9 se nalazi izveštaj klasifikacije. Velika većina instanci benigne klase je svrstana u neki od napada čime je odziv vrednosti 0, a pošto je benigna i najbrojnija onda to umnogome narušava preciznost kod drugih klasa. Problem proizilazi iz toga što je algoritam *NearMiss-1* smanjio broj instanci benigne klase sa otprilike 1.5 miliona na 23 instance, čime je velika količina informacija izgubljena i zato model nije mogao dobro da prepozna instance iz validacionog skupa. Iz tog razloga nema smisla isprobavati ostale modele nad ovakvom strategijom balansiranja.



Slika 4.7: Matrica konfuzije modela XGBoost, n_estimators = 100

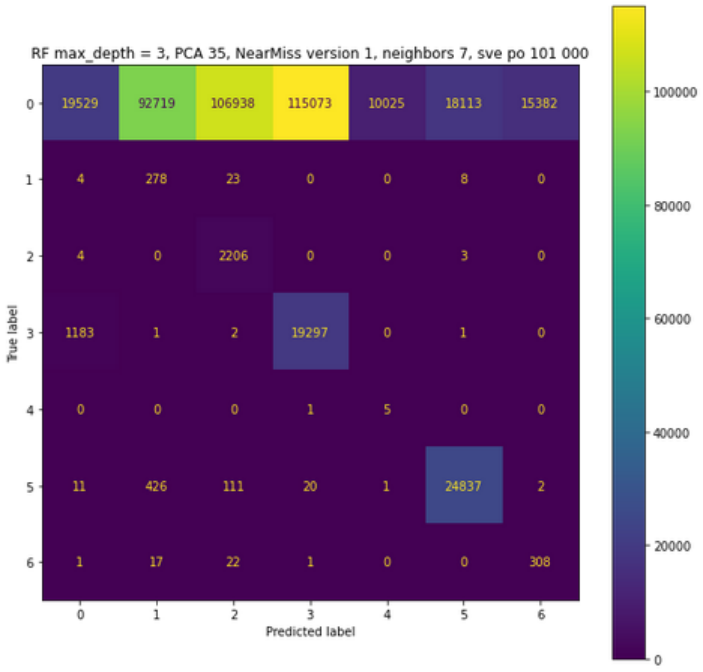


Slika 4.8: Matrica konfuzije za model XGBoost, n_estimators = 400

0	1.00	0.00	0.00	377779
1	0.01	1.00	0.01	313
2	0.05	1.00	0.10	2213
3	0.11	1.00	0.20	20484
4	0.00	0.83	0.00	6
5	0.31	1.00	0.48	25408
6	0.01	0.99	0.02	349
accuracy			0.11	426552
macro avg	0.21	0.83	0.12	426552
weighted avg	0.91	0.11	0.04	426552

Slika 4.9: Izveštaj klasifikacije za model XGBoost, n_estimators = 400

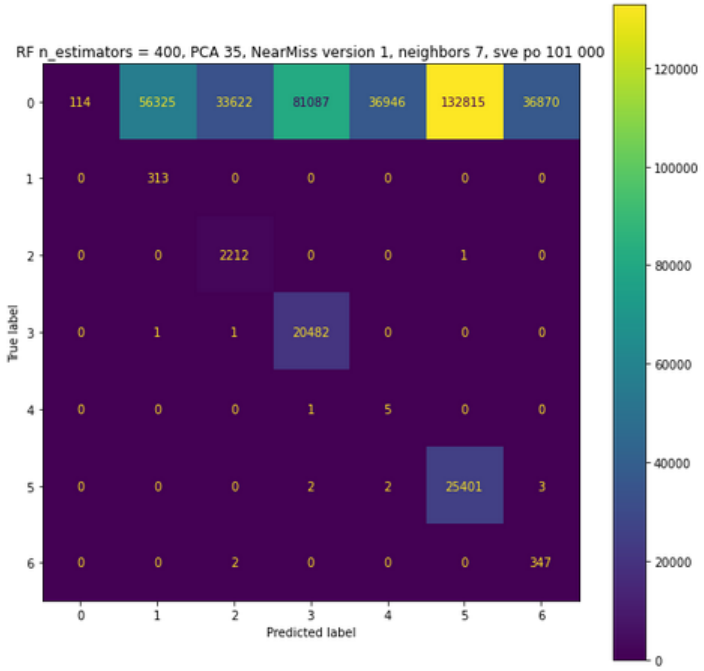
Sledeća kombinacija koja je isprobana je skoro identična prethodnoj, jedina razlika je što je za algoritam *NearMiss-1* korišćeno *neighbors = 7* umesto *neighbors = 3*. Ideja je bila videti da li će rezultati biti bolji ukoliko se poveća broj suseda prilikom smanjenja broja instanci. Primenjeno je istih 9 modela mašinskog učenja kao i za prethodnu kombinaciju. Već na prvom modelu — slučajne šume *max_depth = 3, n_estimators = 100* na slici 4.10 — može videti poboljšanje klasifikacije benigne klase, ranije su se tačno predviđene instance merile u stotinama a sada u desetinama hiljada i tačnost je malo bolja i iznosi 0.16.



Slika 4.10: Matrica konfuzije za model slučajnih šuma, max_depth = 3, n_estimators = 100, 2. kombinacija

Interesantno je da kako se povećava metaparametar *max_depth* tako broj dobro

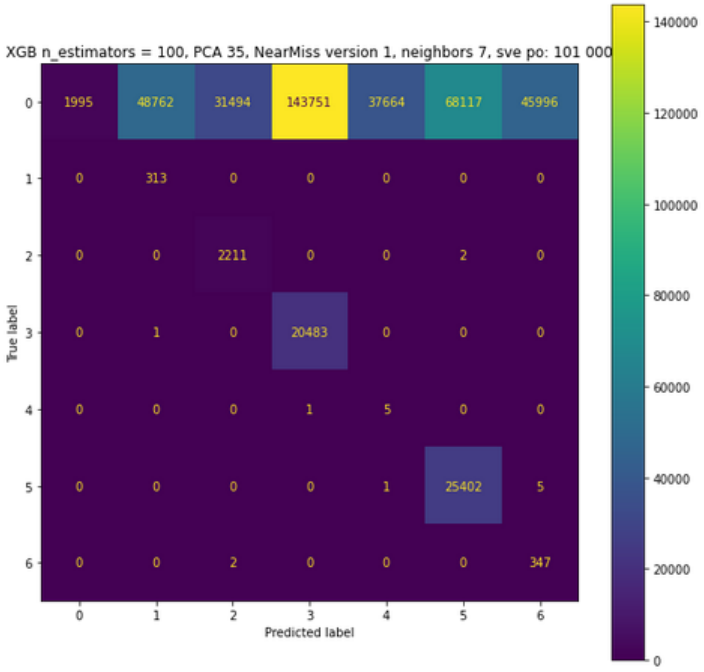
klasifikovanih instanci benigne klase opada, a napada raste. Isto to se dešava i za modele gde je *max_ depth* = *None* a broj stabala raste, pa se tako na slici 4.11 vidi koliko je drastično opala vrednost prvog polja u matrici, a naredni redovi su smanjili vrednosti van dijagonale, tačnost je opala na 0.11.



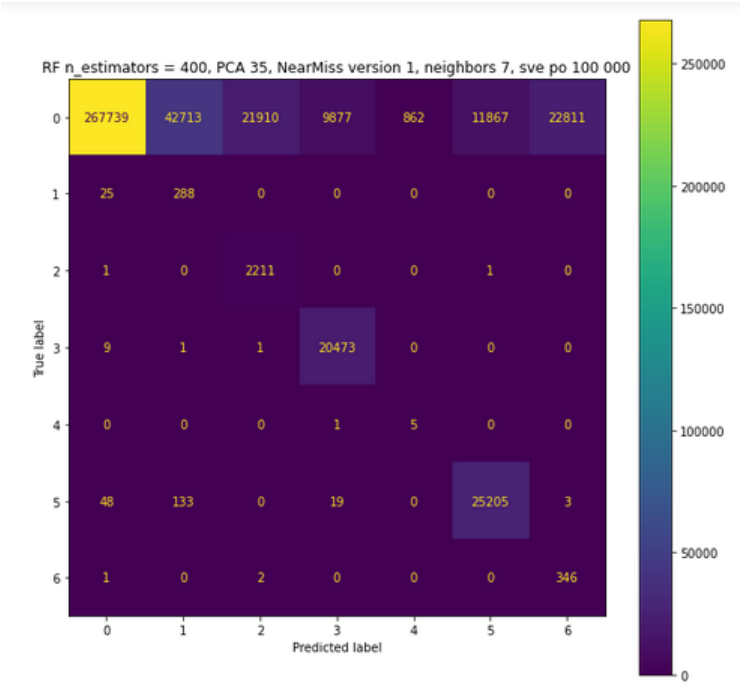
Slika 4.11: Matrica konfuzije modela slučajnih šuma, max_depth = None, n_estimators = 400, 2. kombinacija

Model XGBoost daje bolje rezultate, jer već sa 100 stabala dobro klasifikuje napade, a benignu klasu bolje nego kada je broj suseda bio 3, mada i dalje rezultati nisu zadovoljavajući jer je tačnost 0.12, slika 4.12. Model sa 400 stabala za nekoliko stotina instanci manje dobro klasifikuje benignu klasu, sa istom tačnošću kao i prethodni model. Najveći problem za sve prethodne modele u obe kombinacije je taj što je algoritam *NearMiss-1* smanjio broj instanci sa milion i nešto na 23 instance čime je velika količina informacija izgubljena.

Sledeća strategija smanjenja broja instanci je da se koristi algoritam *NearMiss-1* i 7 suseda, ali da se eksplicitno naglasi da se benigna klasa smanjuje na 100 000 instanci, što se postiže tako što se parametru *sampling_ strategy* prosledi rečnik gde se naglašava koliko instanci želimo da dobijemo za svaku klasu. Nakon primene algoritma *SMOTE* sve klase imaju po 100 000 instanci. Na takve podatke su primenjeni prethodno pomenutih 9 modela mašinskog učenja.



Slika 4.12: Matrica konfuzije modela XGBoost, n_estimators = 100, 2. kombinacija

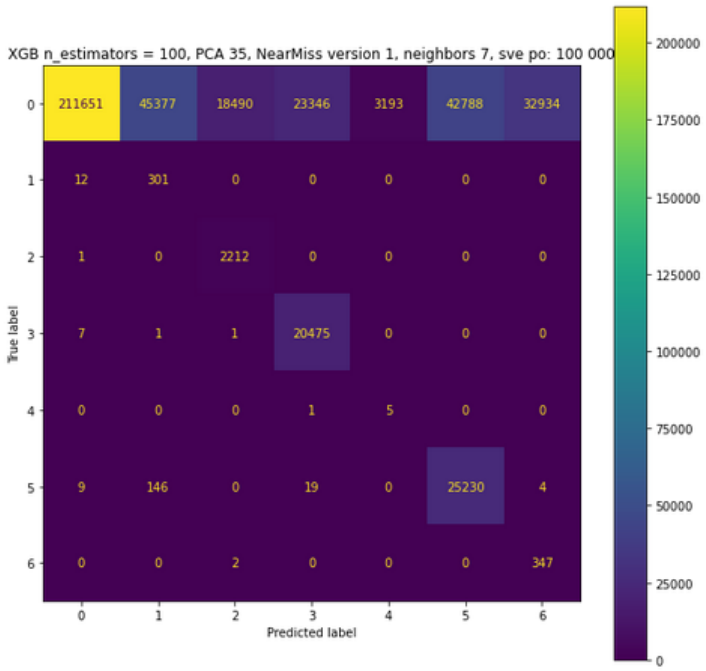


Slika 4.13: Matrica konfuzije modela slučajnih šuma, max_depth = None, n_estimators = 400, 3. kombinacija

Kod algoritma slučajnih šuma se vidi poboljšanje u klasifikaciji benigne klase,

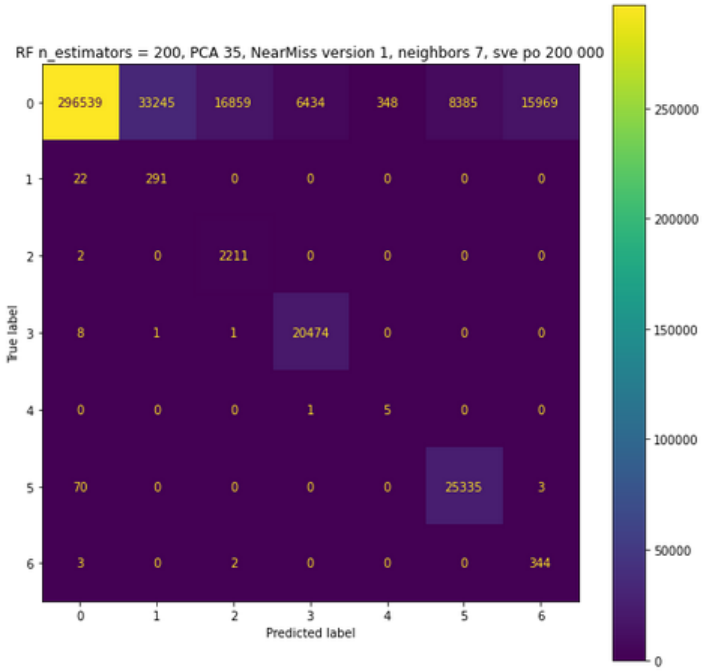
štaviše, kako se povećava metaparametar *max_depth* tako se bolje klasifikuje benigna klasa. Isto se dešava i sa povećanjem stabala, tako da je najbolji model slučajnih šuma dobijen gde je *max_depth* = *None* i *n_estimators* = *400*, čija je tačnost modela značajno unapređena u odnosu na prethodne modele i iznosi 0.74. Matrica konfuzije tog modela se može videti na slici 4.13.

Kod algoritma XGBoost bolje rezultate daje kada ima 100 stabala nego kada ih ima 400, sa tačnošću 0.61. Matrica konfuzije prikazana na slici 4.14 pokazuje da je lošija u tačnom predviđanju benigne klase u odnosu na prošli model, ali za nijansu bolje predviđa napade.

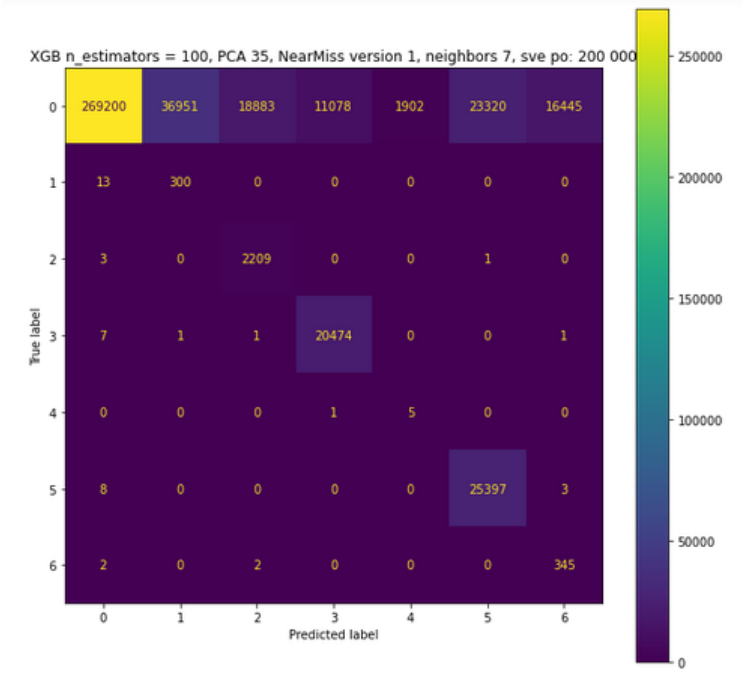


Slika 4.14: Matrica konfuzije modela XGBoost, n_estimators = 100, 3. kombinacija

Naredna kombinacije uključuje ponovo algoritam *NearMiss-1* sa 7 suseda, ali ovog puta sve klase imaju po 200 000 instanci. Ima poboljšanja kod predviđanja benigne klase. Kod modela slučajnih šuma kako raste parametar *max_depth* tako se i benigna klasa bolje predviđa. Kod XGBoost modela, bolje predviđanje benigne klase ima model sa 100 stabala. Najbolje rezultate su dali modeli slučajnih šuma sa parametrima *max_depth* = *None*, *n_estimators* = *200*, čija je tačnost 0.81, i XGBoost sa 100 stabala, čija je tačnost 0.75, koji se mogu videti na slikama 4.15 i 4.16, tim redom.



Slika 4.15: Matrica konfuzije modela slučajnih šuma, max_depth = None, n_estimators = 200, 4. kombinacija

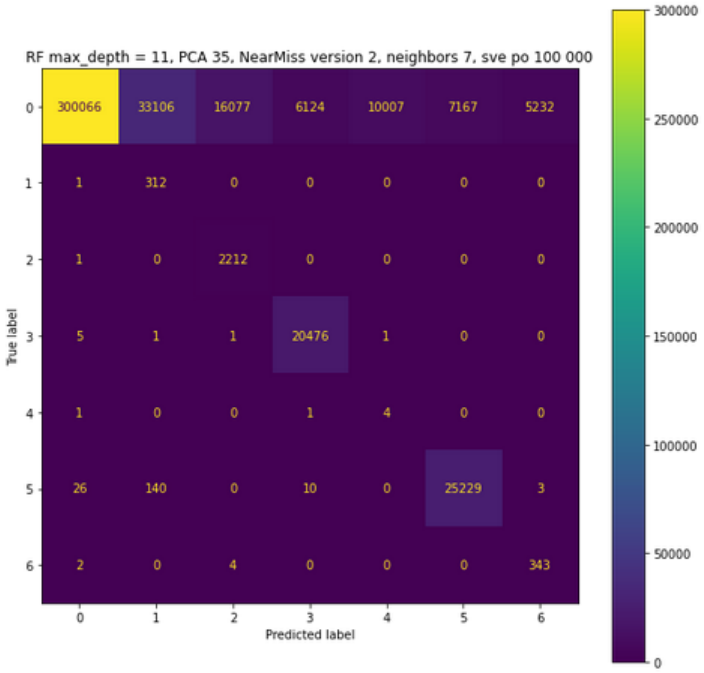


Slika 4.16: Matrica konfuzije modela XGBoost, n_estimators = 100, 4. kombinacija

Sledeća kombinacija uključuje algoritam *NearMiss-2* sa 7 suseda i algoritam *SMOTE* i svaka od klasa će imati po 100 000 instanci. Ideja je isprobati ponovo

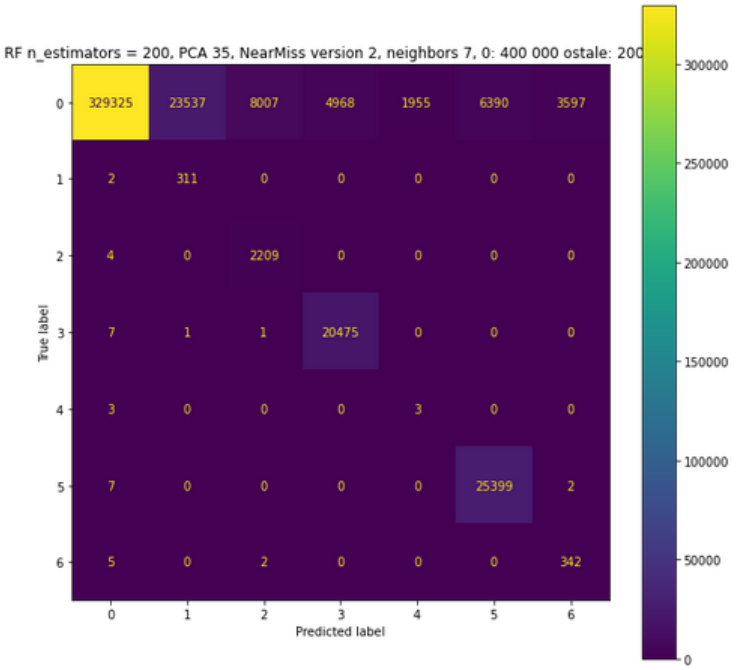
manje vrednosti broja instanci, ali ovog puta koristeći drugu strategiju smanjenja broja instanci. Većina modela je dala lošije rezultate u odnosu na prethodnu kombinaciju, izuzev slučajnih šuma sa parametrom *max_ depth = 11* koji je bio najbolji dosad, može se videti na slici 4.17, i tačnost mu je 0.82.

Algoritam *NearMiss-2* je dao bolje rezultate za duplo manji broj instanci. Sledeće kombinacije će se fokusirati na verziju 2 algoritma *NearMiss*.



Slika 4.17: Matrica konfuzije modela slučajnih šuma, max_depth = 11, 5. kombinacija

S obzirom na to da je najveći problem dosad bilo predviđanje benigne klase koje ima najviše, sledeća ideja je da broj instanci benigne klase bude duplo veći od svake od klasa napada. Tako da je sledeća kombinacija *NearMiss-2* sa 7 suseda, gde benigne klase ima 400 000, svih ostalih po 200 000. Nad ovom kombinacijom balansiranja podataka je primenjeno prethodnih 9 modela mašinskog učenja. Primećuje se poboljšanje u odnosu na prethodne kombinacije, najbolji rezultat daju slučajne šume sa 200 stabala, gde je tačnost čak 0.89. Matrica konfuzije tog modela se može videti na slici 4.18.



Slika 4.18: Matrica konfuzije modela slučajnih šuma, n_estimators = 200, 6. kombinacija

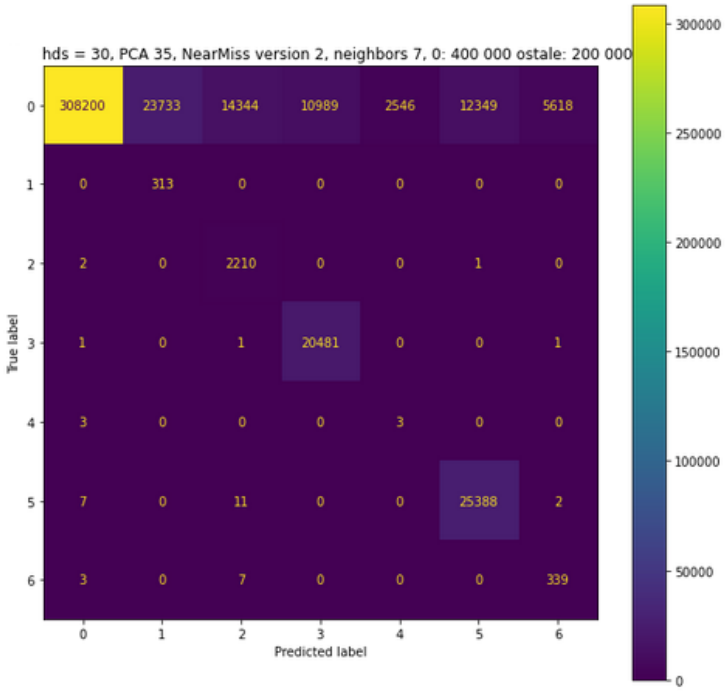
S obzirom na to da su skoro svi modeli dali preciznost iznad 0.80 (izuzev slučajnih šuma max depth = 3), isprobano je još modela mašinskog učenja među kojima su neuronska mreža, kvadratna diskriminantna analiza i multinomijalna logistička regresija. Dodatno isprobani modeli su:

- Neuronska mreža sa metaparametrom *hidden_layer_sizes = 100*
- Neuronska mreža sa metaparametrom *hidden_layer_sizes = 50*
- Neuronska mreža sa metaparametrom *hidden_layer_sizes = 20*
- Neuronska mreža sa metaparametrom *hidden_layer_sizes = 30*
- Neuronska mreža sa metaparametrom *hidden_layer_sizes = 40*
- Neuronska mreža sa metaparametrom *hidden_layer_sizes = 35*
- Kvadratna diskriminantna analiza
- Multinomijalna logistička regresija sa metaparametrima *max_iter = 4000* i *solver = 'lbfgs'*
- Multinomijalna logistička regresija sa metaparametrima *max_iter = 4000*, *solver = 'lbfgs'* i *class_weight = 'balanced'*

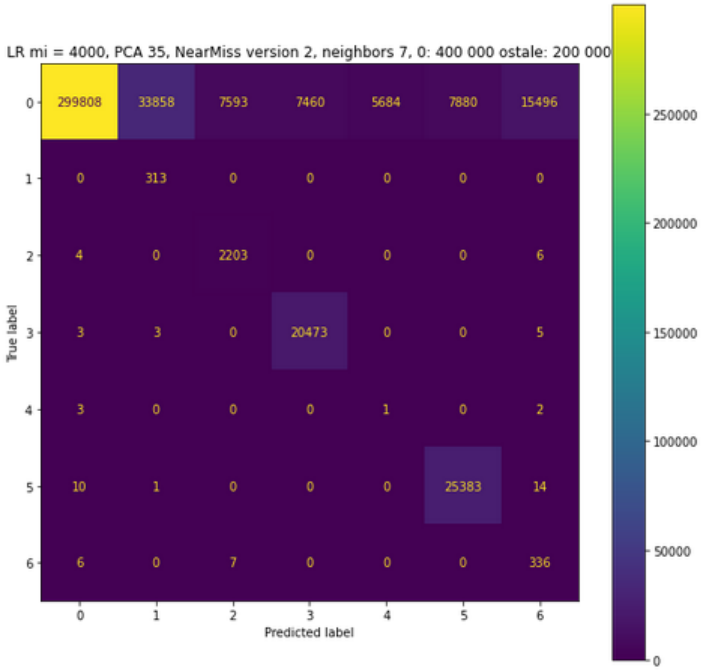
- Multinomijalna logistička regresija sa metaparametrima *max_iter = 4000* i *solver = 'sag'*

Kod multinomijalne logističke regresije je bilo neophodno povećati vrednost *max_iter* parametra jer model nije uspeo da iskonvergira. Uspešno je iskonvergirao tek sa maksimalnim brojem iteracija 4000, a pre toga je pokušano sa 100, 300, 1000 i 2000 maksimalnih iteracija. Neuronska mreža je najbolji rezultat dala sa metaparametrom *hidden_layer_sizes = 30* sa tačnošću 0.84 (slika 4.19), a multinomijalna logistička regresija sa metaparametrima *max_iter = 4000* i *solver = 'lbfgs'*, tačnosti 0.82 (slika 4.20). Model slučajnih šuma je ipak bolji od ova dva dodatna modela.

Model kvadratne diskriminantne analize je dao odlične rezultate (slika 4.21), tačnost modela je 0.96, a obučavanje i predviđanje modela je završeno za 19 sekundi. Ovaj model je ubedljivo najbolje od svih klasifikovao benignu klasu, mada se primećuje nešto lošije klasifikovanje napada, odnosno veći broj instanci klase napada je proglašeno benignom konekcijom nego što je to kod modela slučajnih šuma.



Slika 4.19: Matrica konfuzije modela neuronske mreže, *hidden_layer_sizes = 30*, 6. kombinacija

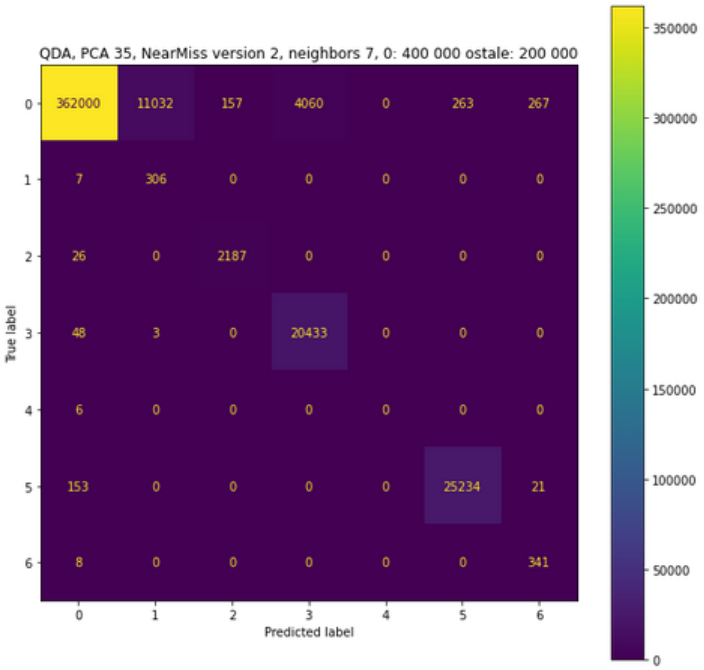


Slika 4.20: Matrica konfuzije modela multinomijalne logističke regresije, *max_ iter = 4000*, 6. kombinacija

Najveći problem koji se uočava kod ovog modela je taj što je u potpunosti loše klasifikovao klasu 4 koja predstavlja napad infiltracije. Sve konekcije koje su infiltracija je predvideo da je benigna konekcija, stoga su preciznost, odziv i F1 mera ove klase 0. Za razliku od kvadratne diskriminantne analize, model slučajnih šuma je od 6 instanci infiltracije u validacionom skupu za 3 od njih rekao da jesu infiltracija, a za ostale tri da su normalna konekcija, ali je slabije prepoznao benignu klasu i obučavanje i predviđanje modela je trajalo više od 44 minuta.

Sledeća kombinacija balansiranja podataka uključuje algoritam *NearMiss-3*, sa 7 suseda. Isprobano je da se algoritmu prosledi rečnik kao strategija uzorkovanja čime bi moglo da se kontroliše na koliko instanci će se smanjiti benigna klasa, međutim, ova verzija algoritma ne radi dobro kada se kao strategija uzorkovanja prosledi rečnik. Kao problem se navodi taj što je željeni broj instanci za uzorkovanje manji od broja instanci koje postoje. Razlog tome može biti taj što se *NearMiss-3* izvodi u 2 koraka i u prvom koraku sam algoritam već drastično smanji veličinu skupa tako da u drugom koraku ima manji broj instanci nego što je zadato u rečniku. Ukoliko se algoritmu zada da smanji benignu klasu sa preko 1 500 000 na 100 000, a da ostale klase zadrže postojeći broj instanci, algoritam izbacuje upozorenje i kao rezultat

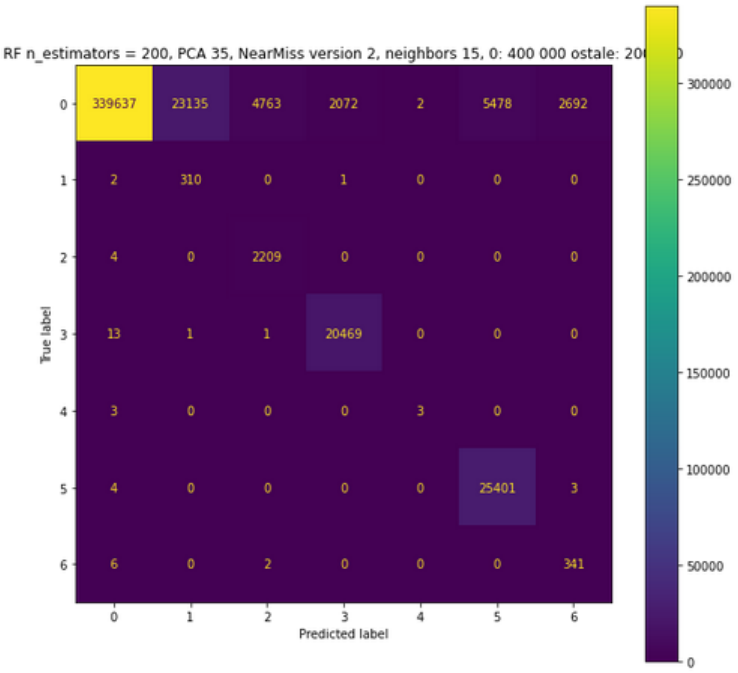
vraća 118 instanci benigne klase i manje od 100 instanci svake od ostalih klasa.



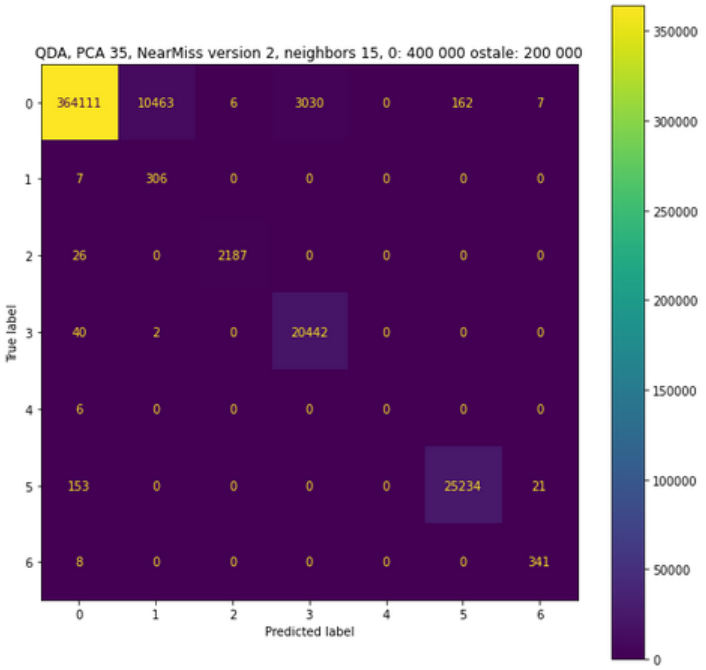
Slika 4.21: Matrica konfuzije modela kvadratne diskriminantne analize, 6. kombinacija

Zbog toga je jedina strategija za uzorkovanje koja se može koristiti *'majority'* kojom se cilja da se smanji isključivo najveća (benigna) klasa. Algoritam je potom smanjio instance benigne klase sa 1 500 000 na 23 instance, što je preveliki gubitak informacija, kao što je i ranije isprobano. Algoritmom SMOTE su potom sve klase dovedene na 101 000 instanci. Od modela je isprobano samo XGBoost sa metaparametrom $n\_estimators = 100$ i Random Forest sa metaparametrima $max\_depth = None$ i $n\_estimators = 400$, modeli su očekivano dali veoma loše rezultate. Stoga nema smisla nastavljati dalje sa algoritmom *NearMiss-3*.

Sledeća kombinacija balansiranja podataka koristi algoritam *NearMiss-2*, ali ovog puta sa 15 suseda, gde će benigna imati 400 000 instanci, a ostale po 200 000. Isprobano je prvobitnih 9 modela, kao i modeli neuronske mreže i multinomijalne logističke regresije koji su se pokazali kao najbolji u prethodnoj kombinaciji — neuronska mreža sa metaparametrom $hidden\_layer\_sizes = 30$, multinomijalna logistička regresija sa metaparametrima $max\_iter = 4000$ i $solver = 'lbfgs'$ — i model kvadratne diskriminantne analize. Kao najbolji modeli su se pokazali model slučajnih šuma sa 200 stabala (slika 4.22) i model kvadratne diskriminantne analize (slika 4.23).



Slika 4.22: Matrica konfuzije modela slučajnih šuma, n_estimators = 200, 8. kombinacija

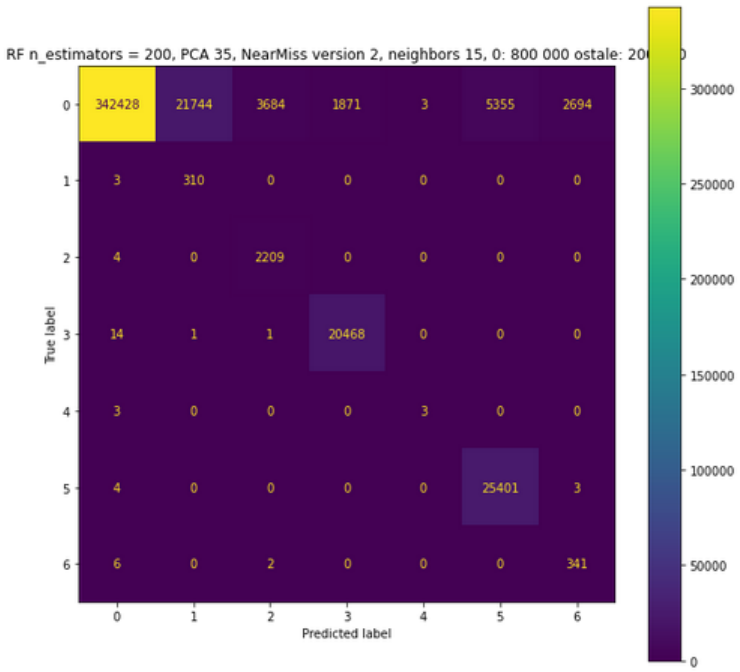


Slika 4.23: Matrica konfuzije modela kvadratne diskriminantne analize, 8. kombinacija

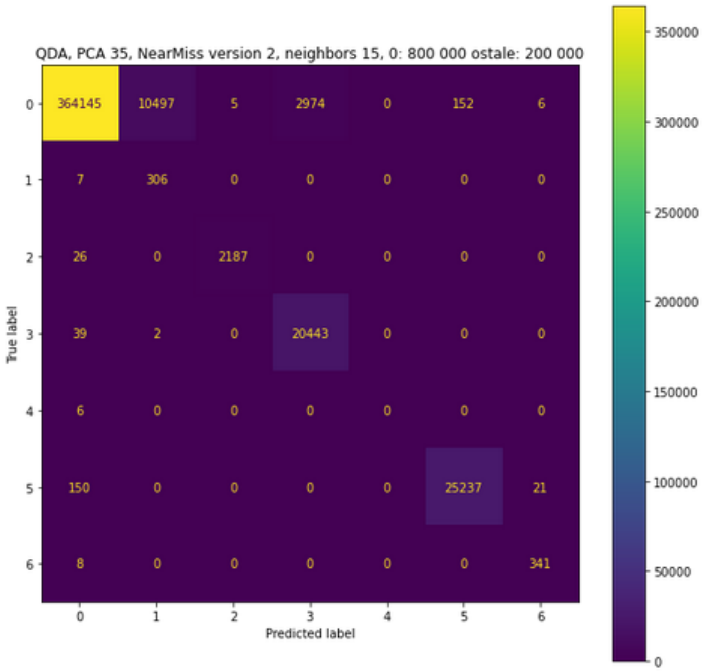
Situacija je slična kao i prošli put kada su ova dva modela upoređivana. Model slučajnih šuma bolje klasifikuje napade, tačnost modela je 0.91, a vreme izvršava-

nja i predviđanja skoro 44 minuta. Model kvadratne diskriminantne analize bolje klasifikuje benignu klasu, tačnost modela je 0.97 a vreme obučavanja i predviđanja 12.5 sekundi.

Naredna kombinacija balansiranja podataka je identična kao prethodna s tim što je broj instanci benigne klase dupliran, pa benigne ima 800 000, a ostalih po 200 000. U ovoj kombinaciji je isprobano samo XGBoost sa 100 i 400 stabala i slučajne šume sa 200 stabala, neuronska mreža sa metaparametrom $hidden\_layer\_sizes = 30$, multinomijalna logistička regresija sa metaparametrima $max\_iter = 4000$ i model kvadratne diskriminantne analize. U ranijim evaluacijama se ispostavilo da slučajne šume sa 200 i 400 stabala daju bolje rezultate nego kada se zadaje maksimalna dubina stabala, takođe, veoma je mala razlika između slučajnih šuma sa 200 i sa 400 stabala a model sa 400 se mnogo duže izvršava tako da se nadalje eksperimentiše samo sa modelom od 200 stabala. Među ovih 6 modela najbolje rezultate daju model slučajnih šuma sa 200 stabala čija se matrica konfuzije može videti na slici 4.24 i model kvadratne diskriminantne analize čija se matrica konfuzije može videti na slici 4.25.



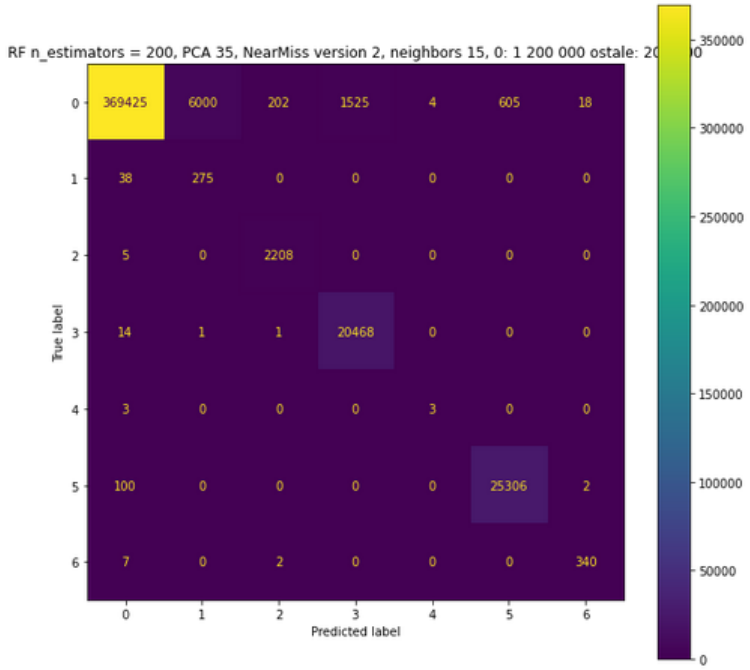
Slika 4.24: Matrica konfuzije modela slučajnih šuma, n_estimators = 200, 9. kombinacija



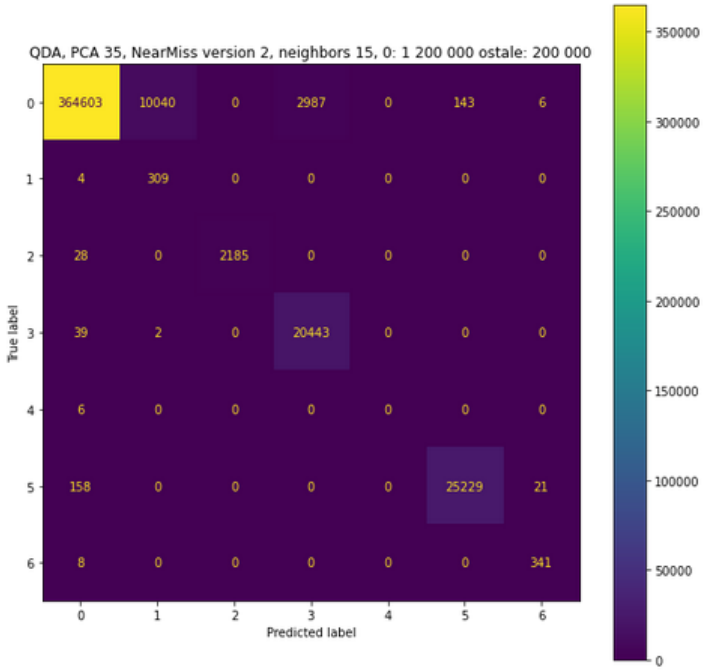
Slika 4.25: Matrica konfuzije modela slučajnih kvadratne diskriminantne analize, 9. kombinacija

Upoređivanje ova dva najbolja rezultata daje isti rezultat kao i u prethodnim kombinacijama. Model slučajnih šuma ima tačnost 0.92 i izvršavao se preko 55 minuta, napade bolje klasifikuje od benigne klase. S druge strane model kvadratne diskriminantne analize ima tačnost 0.97 i izvršavao se skoro 16 sekundi, benignu klasu bolje klasifikuje od napada kada se uporedi sa modelom slučajnih šuma.

Sledeća kombinacija balansiranja podataka je identična prethodnoj, ali je broj instanci benigne klase ponovo povećan i to ovog puta na 1 200 000 instanci. Korišćeno je istih 6 modela kao i u prethodnoj kombinaciji, od kojih su najbolje rezultate ponovo dali model slučajnih šuma sa 200 stabala (slika 4.26) i model kvadratne diskriminantne analize (slika 4.27). Ova dva modela daju veoma slične rezultate što se tiče matrice konfuzije, model slučajnih šuma ima tačnost 0.98 a vreme obučavanja i izvršavanja je 1 sat i 15 minuta, dok je kod modela kvadratne diskriminantne analize tačnost 0.97 a vreme obučavanja i izvršavanja 19 sekundi. Problem kod modela kvadratne diskriminantne analize je ponovo taj što klasu 4, odnosno klasu infiltracije, uopšte ne klasifikuje dobro.



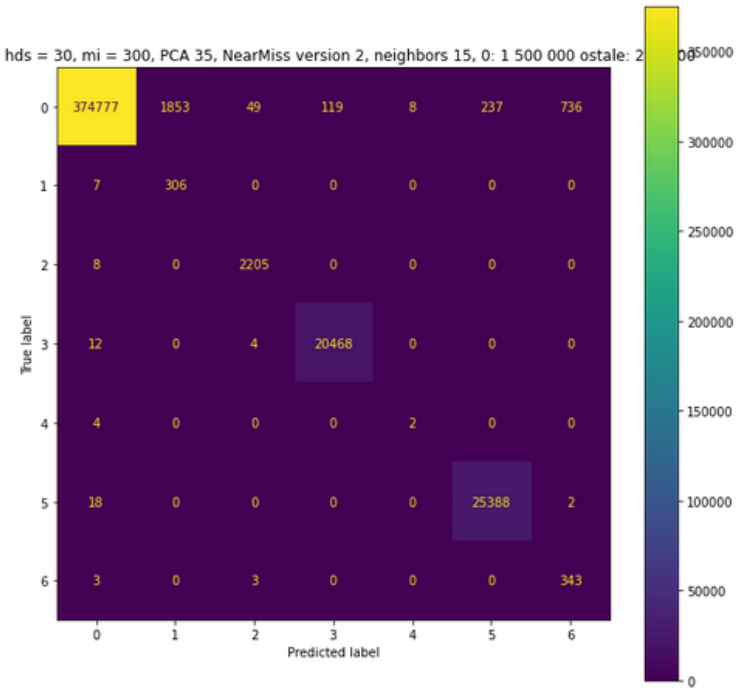
Slika 4.26: Matrica konfuzije modela slučajnih šuma, n_estimators = 200, 10. kombinacija



Slika 4.27: Matrica konfuzije modela kvadratne diskriminantne analize, 10. kombinacija

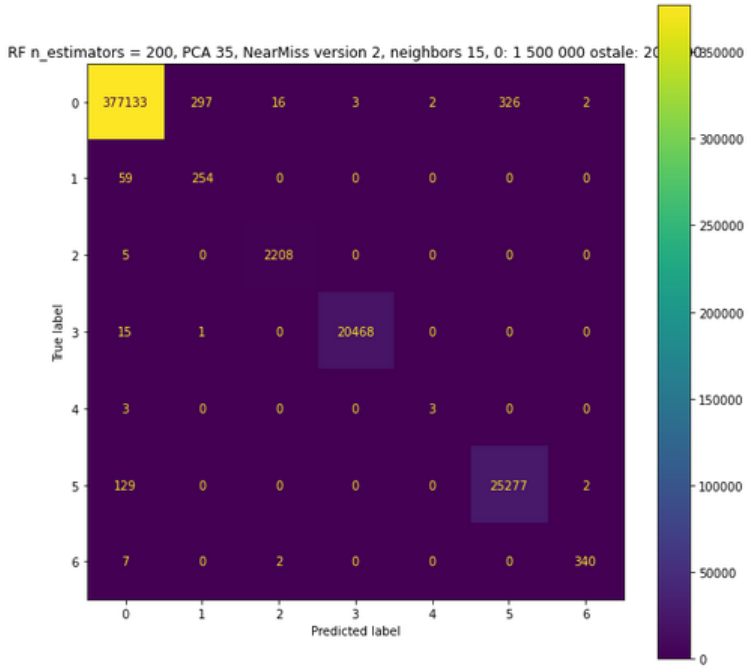
Naredna kombinacija balansiranja podataka je identična prethodnoj, osim što je broj instanci benigne klase povećan na 1 500 000 što je skoro ceo skup instanci

benigne klase u trening skupu. Korišćeno je istih 6 modela kao i u prethodnoj kombinaciji. Svi modeli su se pokazali veoma dobro, XGBoost sa 100 stabala ima tačnost 0.99 kao i sa 400 stabala, ali se prvi model obučavao i izvršavao nešto više od 4 sata, a drugi 17 sati. Model slučajnih šuma ima isto tačnost 0.99 ali je vreme obuke i predviđanja malo manje od sat i po, dok neuronska mreža ima istu tačnost a vreme je 35 minuta. Model kvadratne diskriminantne analize ima tačnost 0.97 i vreme obuke i predviđanja 19 sekundi, dok se logistička regresija pokazala najslabijom sa tačnošću 0.95 i vremenom obuke i previđanja 2 sata i 50 minuta, s tim što je broj maksimalnih iteracija povećan na 8000 jer sa 4000 nije model nije uspeo da iskonvergira. U ovoj strategiji balansiranja podataka najbolje se pokazao model neuronske mreže jer se najbrže izvršio, a imao je približnu tačnost kao i ostali modeli (slika 4.28).

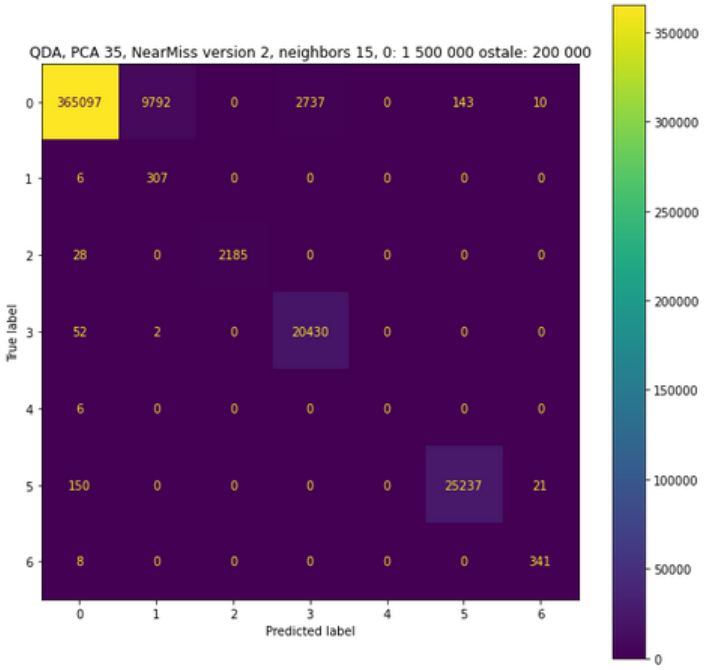


Slika 4.28: Matrica konfuzije modela neuronske mreže sa metaparametrima hidden_layer_sizes = 30 i max_iter = 300, 11. kombinacija

Model slučajnih šuma (slika 4.29) je bio sporiji od neuronskih mreža i slabije je klasifikovao napade, ali je nešto bolje klasifikovao benignu klasu. Kvadratna diskriminantna analiza već nekoliko kombinacija ne poboljšava tačnost. Takođe, ni u jednoj kombinaciji dosad nije ispravno klasifikovala napad infiltracije, pa ni u ovoj nije uspela nijednu instancu te klase da prepozna, što se može videti na slici 4.30.



Slika 4.29: Matrica konfuzije modela slučajnih šuma, n_estimators = 200, 11. kombinacija



Slika 4.30: Matrica konfuzije modela kvadratne diskriminantne analize, 11. kombinacija

Sledeća kombinacija balansiranja podataka uključuje korišćenje algoritma *AllKNN* za smanjenje broja instanci. Algoritmu je kao parametar dato da želi da se smanji

nulta, odnosno benigna klasa. Algoritam je smanjio broj instanci benigne klase sa 1 511 115 na 1 502 558 instanci, nakon čega je primenjen algoritam *SMOTE* i ostale klase su povećane da imaju po 200 000 instanci. Algoritmi mašinskog učenja su dali podjednako dobre rezultate kao i prethodna kombinacija. Problem sa ovim pristupom je taj što se algoritam *AllKNN* izvršavao puna 3 dana. Zbog nepraktičnosti ovaj algoritam se više neće koristiti i kao finalna strategija balansiranja podataka se bira *NearMiss-2* sa 15 suseda gde benigna ima 1 500 000, a ostale po 200 000 instanci.

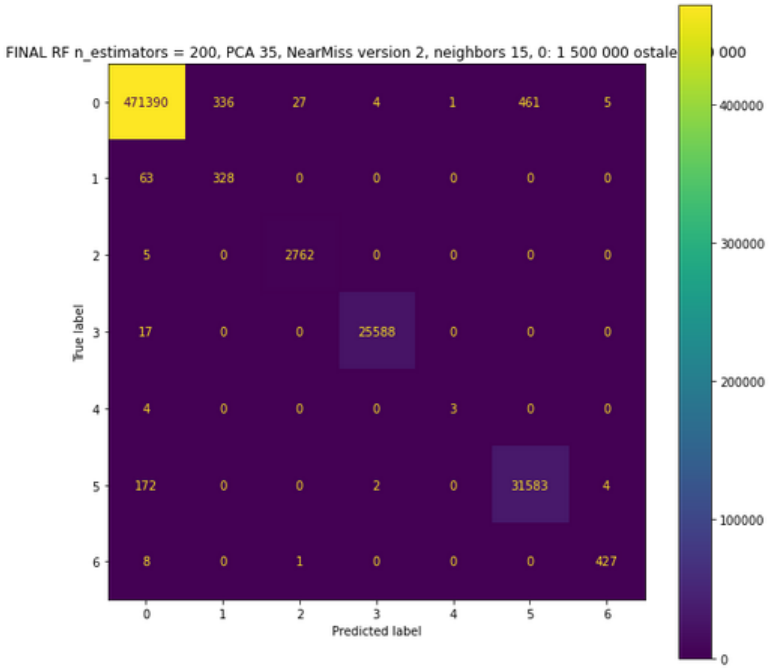
4.3 Finalni modeli i njihove ocene

Prilikom eksperimentalne evaluacije najbolje rezultate su dali modeli trenirani na podacima nad kojima je primenjen algoritam *NearMiss-2* sa 15 suseda gde je benigna klasa smanjena na 1 500 000 instanci, a ostale klase su povećane na po 200 000. Od pet algoritama mašinskog učenja najbolje rezultate su dali sledeći metaparametri:

- Slučajne šume sa metaparametrima *max_ depth = None* i *n_ estimators = 200*
- XGBoost sa metaparametrom *n_ estimators = 100*
- Neuronska mreža sa metaparametrima *hidden_ layer_ sizes = 30* i *max_ iter = 300*
- Multinomijalna logistička regresija sa metaparametrom *max_ iter = 8000*

Kvadratna diskriminantna analiza nema metaparametre koji se podešavaju.

Da bi se dale finalne ocene ovih modela, neophodno je testirati ih na test skupu. Nad trening skupom je izvršeno balansiranje podataka algoritmom *NearMiss-2* sa 15 suseda gde je benigna klasa smanjena na 1 500 000 instanci, a ostale na po 200 000, a potom je nad trening skupom obučavano 5 finalnih modela. Model slučajnih šuma je dao odlične rezultate. Matrica konfuzije modela se može videti na slici 4.31, dok se izveštaj klasifikacije može videti na slici 4.32. Najlošije je predvideo klasu 4, odnosno infiltraciju, koja je ujedno i najmanja klasa. Tačnost klasifikacije je 0.99, vreme obučavanja je 1 sat i 25 minuta, dok je vreme predviđanja 22 sekunde.



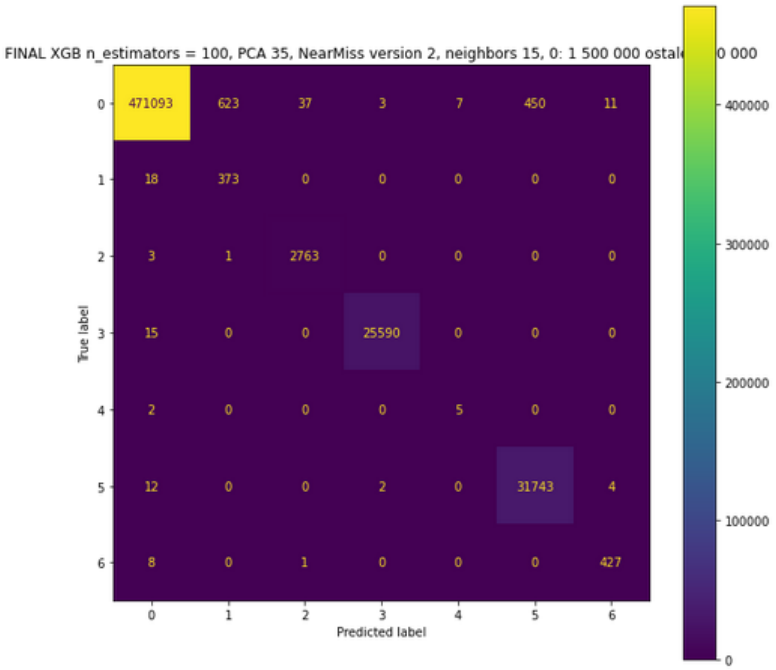
Slika 4.31: Matrica konfuzije finalnog modela slučajnih šuma

	precision	recall	f1-score	support
0	0.9994297	0.9982339	0.9988314	472224
1	0.4939759	0.8388747	0.6218009	391
2	0.9899642	0.9981930	0.9940615	2767
3	0.9997656	0.9993361	0.9995508	25605
4	0.7500000	0.4285714	0.5454545	7
5	0.9856135	0.9943956	0.9899851	31761
6	0.9793578	0.9793578	0.9793578	436
accuracy			0.9979182	533191
macro avg	0.8854438	0.8909946	0.8755774	533191
weighted avg	0.9981833	0.9979182	0.9980159	533191

CPU times: total: 21.9 s
Wall time: 22 s

Slika 4.32: Izveštaj klasifikacije finalnog modela slučajnih šuma

Model XGBoost sa 100 stabala je isto dao odlične rezultate, čak su i bolji od modela slučajnih šuma jer se manje grešilo prilikom klasifikacije napada, pa je ujedno i infiltracija bolje klasifikovana nego kod modela slučajnih šuma. Matrica konfuzije se može videti na slici 4.33, a izveštaj klasifikacije na slici 4.34. Obučavanje modela je trajalo 2 sata i 17 minuta, a predviđanje 6 sekundi. Tačnost modela je 0.99.



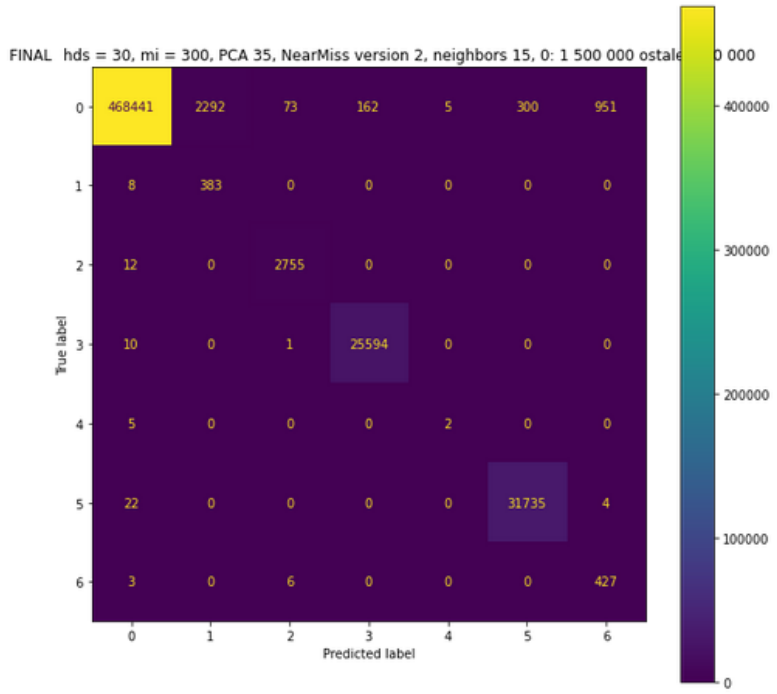
Slika 4.33: Matrica konfuzije finalnog modela XGBoost

	precision	recall	f1-score	support
0	0.9998769	0.9976050	0.9987396	472224
1	0.3741224	0.9539642	0.5374640	391
2	0.9864334	0.9985544	0.9924569	2767
3	0.9998046	0.9994142	0.9996094	25605
4	0.4166667	0.7142857	0.5263158	7
5	0.9860218	0.9994333	0.9926822	31761
6	0.9660633	0.9793578	0.9726651	436
accuracy			0.9977550	533191
macro avg	0.8184270	0.9489449	0.8599904	533191
weighted avg	0.9984842	0.9977550	0.9980222	533191

CPU times: total: 20.7 s
Wall time: 6.03 s

Slika 4.34: Izveštaj klasifikacije finalnog modela XGBoost

Model neuronske mreže je isto tačnosti 0.99 (slika 4.35). Rezultati su slični kao i prethodna dva modela. Ovaj model je najslabije prepoznao infiltraciju kao napad (slika 4.36), stoga je i dalje XGBoost dao najbolje rezultate. Obučavanje je trajalo 36 minuta, a predviđanje skoro 3 sekunde.



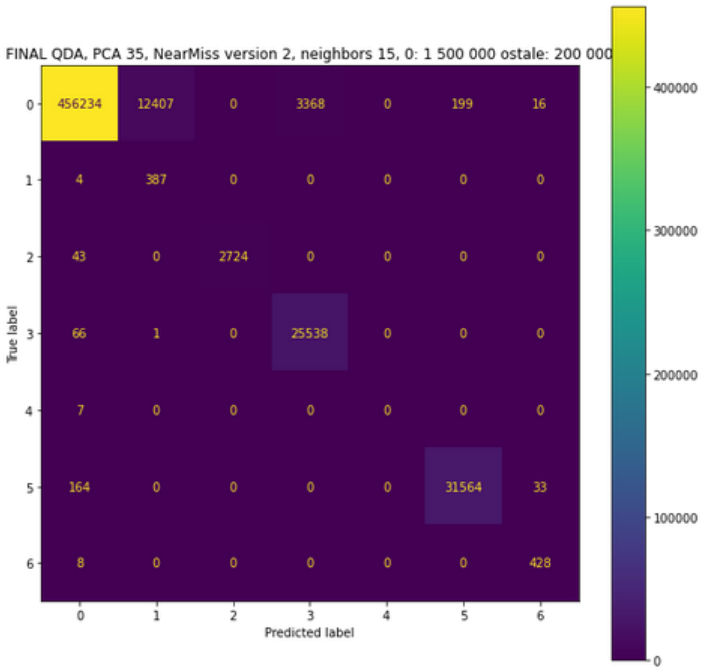
Slika 4.35: Matrica konfuzije finalnog modela neuronske mreže

	precision	recall	f1-score	support
0	0.9998719	0.9919890	0.9959149	472224
1	0.1431776	0.9795396	0.2498369	391
2	0.9717813	0.9956632	0.9835773	2767
3	0.9937102	0.9995704	0.9966317	25605
4	0.2857143	0.2857143	0.2857143	7
5	0.9906352	0.9991814	0.9948900	31761
6	0.3089725	0.9793578	0.4697470	436
accuracy			0.9927718	533191
macro avg	0.6705519	0.8901451	0.7109017	533191
weighted avg	0.9976775	0.9927718	0.9948375	533191

```
CPU times: total: 2.56 s
Wall time: 2.78 s
```

Slika 4.36: Izveštaj klasifikacije finalnog modela neuronske mreže

Model kvadratne diskriminantne analize je dao nešto slabiju tačnost od 0.97, rezultati se mogu videti na slikama 4.37 i 4.38. Međutim, iako se ubedljivo najbrže obučava od svih modela i odlično prepoznaje benignu i ostale napade, model nije u stanju da prepozna infiltraciju stoga se ovaj model ne bi mogao preporučiti kao dobro rešenje za predviđanje. Obučavanje je trajalo 17.5 sekundi, a predviđanje skoro 5 sekundi.



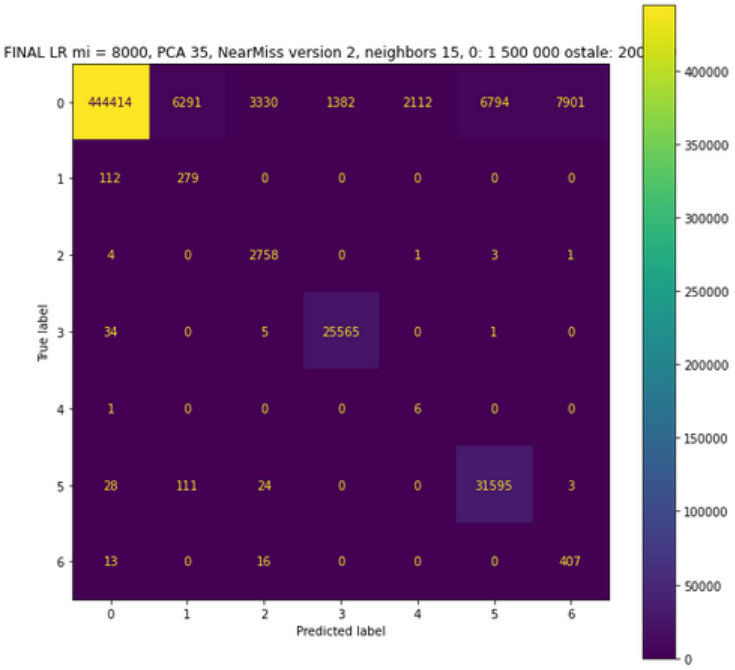
Slika 4.37: Matrica konfuzije finalnog modela kvadratne diskriminantne analize

	precision	recall	f1-score	support
0	0.9993604	0.9661390	0.9824689	472224
1	0.0302462	0.9897698	0.0586986	391
2	1.0000000	0.9844597	0.9921690	2767
3	0.8834844	0.9973833	0.9369852	25605
4	0.0000000	0.0000000	0.0000000	7
5	0.9937348	0.9937974	0.9937661	31761
6	0.8972746	0.9816514	0.9375685	436
accuracy			0.9693993	533191
macro avg	0.6863001	0.8447429	0.7002366	533191
weighted avg	0.9926567	0.9693993	0.9802809	533191

CPU times: total: 8.16 s
Wall time: 4.87 s

Slika 4.38: Izveštaj klasifikacije finalnog modela kvadratne diskriminantne analize

Multinomijalna logistička regresija je dala najlošiji model, sa tačnošću 0.95. Na slici 4.39 se vidi da je najlošije od svih modela klasifikovala benignu klasu i da je to veoma uticalo na preciznost klasifikacije klasa 1, 2, 4 i 6, što se može videti na slici 4.40. Sa druge strane, najbolje od svih modela je klasifikovala infiltraciju, koja u ovom modelu ima najveći odziv od čak 0.86. Vreme obučavanja modela je bilo 1 sat i 30 minuta, dok je vreme predviđanja bilo manje od jedne sekunde.



Slika 4.39: Matrica konfuzije finalnog modela multinomijalne logističke regresije

	precision	recall	f1-score	support
0	0.9995682	0.9411085	0.9694578	472224
1	0.0417602	0.7135550	0.0789027	391
2	0.4496984	0.9967474	0.6197753	2767
3	0.9487141	0.9984378	0.9729411	25605
4	0.0028315	0.8571429	0.0056444	7
5	0.8229365	0.9947735	0.9007327	31761
6	0.0489654	0.9334862	0.0930498	436
accuracy			0.9471728	533191
macro avg	0.4734963	0.9193216	0.5200720	533191
weighted avg	0.9822581	0.9471728	0.9623342	533191

CPU times: total: 1.12 s
Wall time: 937 ms

Slika 4.40: Izveštaj klasifikacije finalnog modela multinomijalne logističke regresije

Na početku ovog poglavlja je rečeno da su autori baze da bi testirali njene performanse primenili 7 najprimenjenijih algoritama mašinskog učenja. Kvalitet modela su procenjivali preko težinskog proseka preciznosti, odziva i F1 mere i brzine izvršavanja obučavanja i predviđanja. Rezultati njihovog istraživanja se mogu videti u tabeli 4.5. Jedan od ciljeva ovog rada je bio testirati model slučajnih šuma za koga su autori rekli da daje najbolje rezultate i koji se zaista pokazao kao jedan od najboljih. Modele neuronske mreže i kvadratne diskriminantne analize je bio cilj

poboljšati. Kada se uporede težinski proseci preciznosti, odziva i F1 mere rezultata ovog rada (tabela 4.6) i rezultata autora baze (tabela 4.5), može se smatrati da je cilj uspešno ostvaren. Brzina izvršavanja obučavanja i predviđanja je manja u ovom radu za sve algoritme. Može se pretpostaviti da je to do hardverskih ograničenja računara na kome su izvršavani algoritmi u ovom radu.

Tabela 4.5: Rezultati autora baze podataka

Algoritam	**Pr**	**Rc**	**F1**	**Izvršavanje (sekunde)**
Slučajne šume	0.98	0.97	0.97	74.39
Neuronske mreže	0.77	0.83	0.76	575.73
Kvadratna diskriminantna analiza	0.97	0.88	0.92	18.79

Međutim, iako model kvadratne diskriminantne analize ima odlične vrednosti težinskog proseka za sve tri vrednosti, ne bi se trebalo koristiti jer nema sposobnost prepoznavanja napada infiltracije. Dodatno isproban model XGBoost se odlično pokazao, pogotovo u klasifikaciji napada, mada je za njega vreme izvršavanja najveće. Model multinomijalne logističke regresije se pokazao kao najslabiji od svih.

Tabela 4.6: Rezultati iz ovog rada

Algoritam	**Pr**	**Rc**	**F1**	**Izvršavanje**
Slučajne šume	0.99	0.99	0.99	1 h 25 min
Neuronske mreže	0.99	0.99	0.99	36 min
Kvadratna diskriminantna analiza	0.99	0.97	0.98	22.5 sek
XGBoost	0.99	0.99	0.99	2 h 17 min
Multinomijalna logistička regresija	0.98	0.95	0.96	1 h 30 min

Gledajući testiranje na test skupu, kao najbolji modeli se izdvajaju model slučajnih šuma sa metaparametrima *max_ depth = None* i *n_ estimators = 200* i XGBoost sa metaparametrom *n_ estimators = 100*.

Glava 5

Zaključak

U ovom radu su napravljeni modeli mašinskog učenja koji se koriste za klasifikaciju mrežnih konekcija na normalnu konekciju ili na neku vrstu napada. Isprobani su modeli preporučeni od strane autora baze u cilju ispitivanja kvaliteta tih modela. Takođe, isprobani su novi algoritmi mašinskog učenja u cilju pronalaženja kvalitetnijeg modela kao i različiti načini balansiranja podataka.

Model slučajnih šuma, predložen od strane autora baze, se ispostavio kao zaista jedan od najboljih modela, a model XGBoost je pokazao još bolje rezultate. Za model slučajnih šuma je utvrđeno da najbolje rezultate daju metaparametri *max_ depth = None* i *n_ estimators = 200*, a za XGBoost metaparametar *n_ estimators = 100*. Ovi modeli mogu biti korišćeni za formiranje sistema za prepoznavanje upada. *CICFlowMeter*, alat korišćen za prikupljanje podataka, je dostupan na internetu i besplatan je. Može se preuzeti i koristiti za prikupljanje podataka o konekcijama na mreži, koje se, nakon prilagođavanja obliku baze podataka, mogu propustiti kroz formirane modele mašinskog učenja.

Autori su napravili veću i noviju verziju baze podataka pod imenom *CSE-CIC-IDS2018* koja ovde nije isprobana zbog hardverskih ograničenja, ali svakako može poslužiti u budućim istraživanjima modela mašinskog učenja.

Bibliografija

[1] Stanford Stats 202. Logistic regression, 2022. on-line at: https://web.stanford.edu/class/stats202/notes/Classification/Logistic-regression.html.

[2] Monowar H Bhuyan, D K Bhattacharyya, and J K Kalita. Surveying Port Scans and Their Detection Methodologies. *The Computer Journal*, 54(10):1565–1581, 2011.

[3] Kevin Biver. *Hakerske tajne za neupućene*. Mikro knjiga, 2019.

[4] Avast Business. What is port scanning?, 1988-2022. on-line at: https://www.avast.com/business/resources/what-is-port-scanning.

[5] Capec. CAPEC-511: Infiltration of Software Development Environment, 2007–2022. on-line at: https://capec.mitre.org/data/definitions/511.html.

[6] Cloudflare. What is a Brute Force attack?, 2022. on-line at: https://www.cloudflare.com/learning/bots/brute-force-attack/.

[7] CrowdStrike. Brute Force attacks, 2022. on-line at: https://www.crowdstrike.com/cybersecurity-101/brute-force-attacks/.

[8] Ono D, Guillen L, Izumi S, Abe T, and Suganuma T. A proposal of port scan detection method based on Packet-In Messages in OpenFlow networks and its evaluation. *International Journal of Network Management*, 31(6):e2174, 2021.

[9] Christos Douligeris and Aikaterini Mitrokotsa. DDoS attacks and defense mechanisms: classificationand state-of-the-art. *Computer Networks*, 2(44):643–666, 2003.

[10] ExtraHop. Brute Force Attack: Definition, Examples, and Prevention, 2022. on-line at: `https://www.extrahop.com/resources/attacks/brute-force/`.

[11] M. Feily, A. Shahrestani, and S. Ramadass. A survey of botnet and botnet detection. In *2009 Third International Conference on Emerging Security Information, Systems and Technologies*. IEEE, 2009.

[12] Canadian Institute for Cybersecurity University of New Brunswick. Botnet dataset, 2022. on-line at: `https://www.unb.ca/cic/datasets/botnet.html`.

[13] Canadian Institute for Cybersecurity University of New Brunswick. CIC DoS dataset (2017), 2022. on-line at: `https://www.unb.ca/cic/datasets/dos-dataset.html`.

[14] Canadian Institute for Cybersecurity University of New Brunswick. DDoS Evaluation Dataset (CIC-DDoS2019), 2022. on-line at: `https://www.unb.ca/cic/datasets/ddos-2019.html`.

[15] Canadian Institute for Cybersecurity University of New Brunswick. Intrusion Detection Evaluation Dataset (CIC-IDS2017), 2022. on-line at: `https://www.unb.ca/cic/datasets/ids-2017.html`.

[16] Canadian Institute for Cybersecurity University of New Brunswick. URL dataset (ISCX-URL2016), 2022. on-line at: `https://www.unb.ca/cic/datasets/url-2016.html`.

[17] Center for Internet Security. Election Security Spotlight – Web Attacks, 2022. on-line at: `https://www.cisecurity.org/insights/spotlight/ei-isac-cybersecurity-spotlight-web-attack`.

[18] GeeksforGeeks. Multi-Layer Perceptron Learning in Tensorflow, 2023. on-line at: `https://www.geeksforgeeks.org/multi-layer-perceptron-learning-in-tensorflow/`.

[19] GeeksforGeeks. XGBoost, 2023. on-line at: `https://www.geeksforgeeks.org/xgboost/`.

[20] GitHub. GitHub Patator, 2022. on-line at: `https://github.com/lanjelot/patator`.

[21] Barbara Guttman and Edward Roback. *An Introduction to Computer Security: The NIST Handbook Special Publication 800-12*. National Institute of Standards and Technology Technology Administration U.S. Department of Commerce, 1995.

[22] Rick Hofstede, Mattijs Jonker, Anna Sperotto, and Aiko Pras. Flow-Based Web Application Brute-Force Attack and Compromise Detection. *Journal of Network and Systems Management volume*, (25):735–758, 2017.

[23] Kaspersky. Brute Force attack, 2022. on-line at: `https://www.kaspersky.com/resource-center/definitions/brute-force-attack`.

[24] Kaspersky. What is a Botnet?, 2022. on-line at: `https://www.kaspersky.com/resource-center/threats/botnet-attacks`.

[25] Imbalanced Learn. AllKNN, 2022. on-line at: `https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.AllKNN.html`.

[26] Imbalanced Learn. Sample selection in NearMiss, 2022. on-line at: `https://imbalanced-learn.org/dev/auto_examples/under-sampling/plot_illustration_nearmiss.html#sphx-glr-auto-examples-under-sampling-plot-illustration-nearmiss-py`.

[27] Tasnuva Mahjabin, Yang Xiao, Guang Sun, and Wangdong Jiang. A survey of distributed denial-of-service attack, prevention, and mitigation techniques . *International Journal of Distributed Sensor Networks*, 13(3):246–256, 2017.

[28] NETSCOUT. What is a Bot?, 2022. on-line at: `https://www.netscout.com/what-is/bot`.

[29] Mladen Nikolić and Anđelka Zečević. *Mašinsko učenje*. 2019.

[30] NMAP. Port Scanning Basics. on-line at: `https://nmap.org/book/man-port-scanning-basics.html`.

[31] OWASP. Brute Force attack, 2022. on-line at: `https://owasp.org/www-community/attacks/Brute_force_attack`.

[32] OWASP. Cross Site Scripting (XSS), 2022. on-line at: `https://owasp.org/www-community/attacks/xss/`.

[33] OWASP. SQL Injection, 2022. on-line at: `https://owasp.org/www-community/attacks/SQL_Injection`.

[34] PortSwigger. SQL injection, 2022. on-line at: `https://portswigger.net/web-security/sql-injection`.

[35] MIT Technology Review. The first DDoS attack was 20 years ago. This is what we've learned since., 2022. on-line at: `https://www.technologyreview.com/2019/04/18/103186/the-first-ddos-attack-was-20-years-ago-this-is-what-weve-learned-since/`.

[36] S. Gupta S. Kumar and S. Arora. Research Trends in Network-Based Intrusion Detection Systems: A Review. *IEEE Access*, 9:157761–157779, 2021.

[37] Towards Data Science. SMOTE, 2023. on-line at: `https://towardsdatascience.com/smote-fdce2f605729`.

[38] Signal Sciences. What Are Bot Attacks?, 2021. on-line at: `https://www.signalsciences.com/glossary/bot-attack-protection/`.

[39] scikit learn. Linear and Quadratic Discriminant Analysis, 2023. on-line at: `https://scikit-learn.org/stable/modules/lda_qda.html#linear-and-quadratic-discriminant-analysis`.

[40] SecureReading. How a Hacker can Infiltrate your Network and What can be Done About it?, 2022. on-line at: `https://securereading.com/hacking-hacker-infiltrate-networks/`.

[41] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. Toward generating a new intrusion detection dataset and intrusiontraffic characterization. In *4th International Conference on Information Systems Security and Privacy (ICISSP)*, 2018.

[42] David Solomon. *Elements of computer secutity*. Springer, 2012.

[43] B. Soniya and M. Wilscy. Detection of randomized bot command and control traffic on an end-point host. *Alexandria Engineering Journal*, 55(3):2771–2781, 2016.

[44] Spiceworks. What Is a Brute Force Attack? Definition, Types, Examples, and Prevention Best Practices in 2022, 2006-2022. on-line

at: https://www.spiceworks.com/it-security/cyber-risk-management/articles/what-is-brute-force-attack/.

[45] PANG-NING TAN, MICHAEL STEINBACH, ANUJ KARPATNE, and VIPIN KUMAR. *Introduction to Data Mining*. Pearson, second edition, 2019.

[46] Endru Tanenbaum and Dejvid Vederol. *Računarske mreže*. Mikro knjiga, 2012.

[47] TechTarget. FTP (File Transfer Protocol) , 2000-2022. on-line at: https://www.techtarget.com/searchnetworking/definition/File-Transfer-Protocol-FTP.

[48] That Data Tho... Linear vs. Quadratic Discriminant Analysis – Comparison of Algorithms. on-line at: https://thatdatatho.com/linear-vs-quadratic-discriminant-analysis/.

[49] Karthikeyan Thyagarajan and Arunkumar Thangavelu. DDoS attacks and defense mechanisms: classificationand state-of-the-art. *European Journal of Scientific Research*, 139(3):246–256, 2016.

[50] TrustNet. Common Web Application Attacks, 2022. on-line at: https://www.trustnetinc.com/web-application-attacks/.

Biografija autora

Vladana Đorđević (*Beograd, 28. novembar 1996.*) je diplomirani informatičar. Pohađala je Osnovnu školu „Ujedinjene nacije" u Beogradu od 2003. do 2011. godine i završila kao nosilac Vukove diplome. Trinaestu beogradsku gimaziju društveno-jezički smer je upisala 2011. godine i završila 2015. godine kao nosilac Vukove diplome. 2019. godine je završila osnovne studije na Matematičkom fakultetu, Univerziteta u Beogradu, na smeru Informatika.