Primjena strojnog učenja za kibernetičku sigurnost

Kozlov, Marko

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Prijediplomski studij računarstva

Završni rad

Primjena strojnog učenja za kibernetičku sigurnost

Rijeka, rujan 2023.

Marko Kozlov
0069089061

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Prijediplomski studij računarstva

Završni rad

Primjena strojnog učenja za kibernetičku sigurnost

Mentor: doc. dr. sc. Goran Mauša

Rijeka, rujan 2023.

Marko Kozlov
0069089061

Umjesto ove stranice umetnuti zadatak za završni ili diplomski rad

Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad.

Rijeka, rujan 2023.

_____

Marko Kozlov

Sadržaj

Popis slika

Popis tablica

Poglavlje 1

Uvod

Zbog sve brže evolucije malicioznog softvera, u području kibernetičke sigurnosti jedan od većih izazova postaje dizajniranje sustava za detekciju neovlaštenog pristupa *(engl. Intrusion detection system (IDS))*. Maliciozni napadi postaju sve složeniji te je njihova detekcija znatno otežana radi primjene tehnika izbjegavanja detekcije. Također, sve je veći broj novih, prethodno nepoznatih vrsta napada *(engl. zero-day attacks)* [6]. Takvi napadi predstavljaju iznimno veliku sigurnosnu prijetnju, kako običnim korisnicima na internetu, tako i raznim firmama, velikim korporacijama te bankama. Po pitanju bankarskog sektora, kibernetičke kriminalne radnje su ponajprije bile fokusirane na klijente banaka, čime je fokus bio na krađi bankovnih računa ili kreditnih kartica. Međutim, razvojem malicioznog softvera te istovremenim povećanjem broja zero-day napada, primarnom metom danas postaju same banke čime napadači mogu ukrasti veliku količinu novca u samo jednom kibernetičkom napadu [6]. Također, u svrhu ostvarenja financijskih sredstava te nekih političkih ili aktivističkih ciljeva, u današnjem svijetu kibernetičkim kriminalcima sve veću vrijednost imaju same informacije. Drugim riječima, neovlašteni pristup osjetljivim informacijama od strane napadača, može imati vrlo snažan negativan utjecaj na korporacije te ljude na visokim organizacijskim ili političkim pozicijama. Upravo radi tih razloga, detekcija zero-day napada je postala jednom od glavnih prioriteta u polju kibernetičke sigurnosti.

Sustavi za detekciju neovlaštenog pristupa se mogu podijeliti na dvije skupine, one temeljene na raspoznavanju uzoraka *(engl. signature-based (SIDS))* te one te-

meljene na detekciji anomalija *(engl. anomaly-based (AIDS))*. Sustavi za detekciju neovlaštenog pristupa temeljenog na raspoznavanju uzoraka funkcioniraju na temelju tehnika uočavanja uzoraka kako bi detektirali neki prethodno poznati napad. Drugim riječima, računalne aktivnosti u nekom trenutku se uspoređuju sa aktivnostima koje su prethodno detektirane kao maliciozne. Zbog sve većeg broja zero-day napada, SIDS tehnike su progresivno postale sve manje djelotvorne te se zbog tog razloga sve veći fokus pridaje AIDS tehnikama. AIDS se temelji na razlikovanju prihvatljivog ponašanja računalnog sustava od onog abnormalnog. Glavna prednost AIDS-a je njegova sposobnost detekcije zero-day napada, pri čemu se raspoznavanje abnormalnih aktivnosti korisnika ne temelji na bazi prethodno poznatih uzoraka, već na temelju akcija, odnosno ponašanja korisnika [6].

Jedna od tehnika implementacije AIDS-a je ona temeljena na strojnom učenju. U sklopu ovoga rada, za realizaciju AIDS-a temeljenog na strojnom učenju koristit će se tehnika nadziranog učenja *(engl. supervised learning)*, pri kojoj su uzorci u skupu podataka za treniranje već klasificirani odgovarajućom oznakom. Za provođenje tehnike nadziranog učenja koristit će se NIDS *(engl. Network Intrusion Detection Systems)* skupovi podataka preuzeti iz radova [5] i [1]. Također, za treniranje i testiranje modela pomoću navedenih podataka koristit će se algoritmi naivni Bayesov klasifikator, slučajna šuma, stablo odluke i KNN. Na temelju takvog pristupa, glavna svrha ovoga rada je analizirati značajke skupova podataka te putem primjene više algoritama strojnog učenja ustanoviti koji model za detekciju anomalija u mrežnom prometu ima najbolje performanse.

Poglavlje 2

Skupovi podataka

Mrežni sustavi za otkrivanje neovlaštenog pristupa *(Network Intrusion Detection Systems, NIDS)* temeljeni na strojnom učenju postali su obećavajući alat za zaštitu mreža od kibernetičkih napada. U svrhu istraživanja i razvijanja mrežnih sustava za otkrivanje neovlaštenog upada javno je dostupna velika količina skupova podataka. To je korisno radi mogućnosti razvoja više različitih modela strojnog učenja, koji na temelju podataka za treniranje mogu detektirati ciljane vrste kibernetičkih napada [5].

Problem se javlja kod detaljnijeg analiziranja tih skupova podataka. Uspostavilo se da ovakve vrste skupova podataka imaju vrlo različite skupove značajki, što onemogućava pouzdano uspoređivanje više modela strojnog učenja preko različitih skupova podataka. Zbog toga se ujedno javlja problem primjene modela kod različitih tipova mreža i napadačkih scenarija. Ograničene mogućnosti za procjenu rada mrežnih sustava za otkrivanje neovlaštenog upada temeljenih na strojnom učenju su dovele do sraza između akademskog istraživanja i praktične primjene u stvarnim mrežama. U svrhu rješavanja ovog problema kreirano je pet NIDS skupova podataka sa zajedničkim skupom značajki temeljenim na *NetFlow-u*[1]. Tih pet skupova podataka je generirano iz četiri referentna skupa podataka: UNSW-NB15, BoT-IoT, ToN-IoT, CSE-CIC-IDS2018. Izdvajanje značajki NetFlow formata iz već postojećih NIDS skupova podataka istraživačima je omogućena procjena rada modela strojnog

[1]NetFlow je industrijski standardni protokol za prikupljanje mrežnog prometa.

učenja preko različitih skupova podataka uz korištenje istih skupova značajki [5].

U nastavku slijedi opis navedenih skupova podataka, odnosno prikaz njihova sadržaja pod što se podrazumijeva broj uzoraka te podjela tih uzoraka između različitih vrsta napada i normalnog mrežnog prometa:

- **NF-UNSW-NB15** skup podataka sadrži ukupno 1,623,118 uzoraka od kojih je 4.46% uzoraka definirano kao napad, a 95.54% definirano kao normalan promet. Napadi su još dodatno podijeljeni u više podkategorija, odnosno na više vrsta [5].
- **NF-BoT-IoT** skup podataka sadrži ukupno 600,100 uzoraka od kojih je 97.69% uzoraka definirano kao napad, a 2.31% definirano kao normalan promet. Napadi ovog skupa podataka su dodatno podijeljeni u četiri kategorije [5].
- **NF-ToN-IoT** skup podataka sadrži ukupno 1,379,274 uzoraka od kojih je 80.4% uzoraka definirano kao napad, a 19.6% definirano kao normalan promet [5].
- **NF-CSE-CIC-IDS2018** skup podataka sadrži ukupno 8,392,401 uzoraka od kojih je 12.14% uzoraka definirano kao napad, a 87.86% uzoraka definirano kao normalan promet [5].
- **NF-UQ-NIDS** skup podataka spaja sve prethodno navedene skupove podataka u značajno veći skup podataka, koji sadrži uzorke tokova dobivenih iz više mreža s različitim parametrima. Ovaj skup podataka ujedno sadrži dodatnu klasifikacijsku značajku, koja identificira originalni skup podataka svakog mrežnog toka. Ova značajka se može iskoristiti za usporedbu identičnih napadačkih scenarija provedenih nad dvije ili više testnih mreža. Napadačke kategorije iz ostalih skupova su spojene u poopćene roditeljske kategorije, odnosno više principno sličnih napada je definirano kao jedna kategorija[2]. Ovaj skup podataka sadrži ukupno 11,994,893 uzoraka od kojih je 23.23% uzoraka definirano kao napad, a 76.77% definirano kao normalan promet [5].
- **NF-UNSW-NB15-v2** skup podataka sadrži ukupno 2,390,275 uzoraka od kojih je 3.98% uzoraka definirano kao napad, a 96.02% definirano kao normalan

[2]Više vrsta DDos napada je klasificirano samo kao DDos.

promet [1].

- **NF-BoT-IoT-v2** skup podataka sadrži ukupno 37,763,497 uzoraka od kojih je 99.64% uzoraka definirano kao napad, a 0.36% definirano kao normalan promet [1].
- **NF-ToN-IoT-v2** skup podataka sadrži ukupno 16,940,496 uzoraka od kojih je 63.99% uzoraka definirano kao napad, a 36.01% definirano kao normalan promet [1].
- **NF-CSE-CIC-IDS2018-v2** skup podataka sadrži ukupno 18,893,708 uzoraka od kojih je 11.95% uzoraka definirano kao napad, a 88.05% definirano kao normalan promet [1].
- **NF-UQ-NIDS-v2** skup podataka sadrži ukupno 75,987,976 uzoraka od kojih je 66.88% uzoraka definirano kao napad, a 33.12% definirano kao normalan promet. Ovaj skup podataka se sa NF-UQ-NIDS skupom podataka razlikuje jedino u broju značajki i uzoraka [1].

2.1 Izrada skupova podataka u NetFlow formatu

Da bi se uopće kreirali skupovi podataka u NetFlow formatu potrebno je prvo snimiti i prikupiti dovoljnu količinu mrežnog prometa. Za to postoje dvije metode, prikupljanje kompletnih mrežnih paketa i izvlačenje sadržaja u obliku toka. Iako metoda prikupljanja mrežnih paketa omogućava dostupnost veće količine mrežnih podataka, ta metoda nije skalabilna jer može zahtijevati velike količine podatkovnog prostora kako bi se snimio kratak period mrežnog prometa. Takve količine podataka je ujedno i iznimno teško za analizirati. Alternativna metoda je izvlačenje sadržaja mrežnog prometa u obliku toka. Ova metoda identificira pakete koji sadrže neke zajedničke atribute, primjerice, izvorišna i odredišna IP adresa, protokol transportnog sloja itd. Ti paketi čine jedan tok. Kada se atributi u paketima promjene, stvara se novi tok [5].

Za konverziju datoteka u formatu *pcap* koje sadrže snimljeni mrežni promet te

izdvajanje 12 značajki navedenih u tablici 2.1, iskorišten je alat *nProbe*[3]. U tablici 2.1 su navedene značajke skupova podataka u NetFlowV1 formatu. Naknadno su kreirani i skupovi podataka u NetFlowV2 formatu. Ti skupovi podataka su nadogradnja na skupove podataka u NetFlowV1 formatu, odnosno uz postojećih 12 značajki sadrže još dodatnu 31 značajku, odnosno sve zajedno 43 značajke koje su vidljive na slici 2.1. Uz dodatne značajke, skupovi podataka u NetFlowV2 formatu sadrže i znatno veći broj uzoraka.

U skupove podataka se dodaju još dvije značajke "Label" i "Attack", koje će predstavljati klasifikaciju uzoraka. Ako određeni tok podataka sadrži vrijednosti značajki koje se poklapaju sa određenim napadačkim vrijednostima, taj tok podataka će biti označen kao napad, odnosno sa klasom 1. Pored značajke "Label", pod značajku "Attack" će biti definirana vrsta specifičnog napada. Inače ako to nije slučaj tok podataka će biti označen kao dobroćudni, odnosno klasom 0 [5].

Tablica 2.1 *Lista značajki skupova podataka u NetFlowV1 formatu [5]*

Feature	**Description**	**Feature Type**
IPV4_SRC_ADDR	IPv4 source address	Nominal
IPV4_DST_ADDR	IPv4 destination address	Nominal
L4_SRC_PORT	IPv4 source port number	Discrete
L4_DST_PORT	IPv4 destination port number	Discrete
PROTOCOL	IP protocol identifier byte	Discrete
TCP_FLAGS	Cumulative of all TCP flags	Discrete
L7_PROTO	Layer 7 protocol (numeric)	Numeric
IN_BYTES	Incoming number of bytes	Discrete
OUT_BYTES	Outgoing number of bytes	Discrete
IN_PKTS	Incoming number of packets	Discrete
OUT_PKTS	Outgoing number of packets	Discrete
FLOW_DURATION_MILLISECONDS	Flow duration in milliseconds	Discrete

[3]Detaljnije informacije o navedenim značajkama su dostupne na web stranici [7]

2.1.1 Opis značajki skupova podataka

U tablici 2.1 se nalaze značajke dobivene pomoću alata *nProbe*. Značajka *PROTOCOL* predstavlja identifikacijsku oznaku korištenog IP protokola, koja se nalazi u zaglavlju IPv4 paketa. Značajka *TCP_FLAGS* definira ukupan broj TCP zastavica (SYN, ACK...) korištenih za prijenos određenog paketa, pod uvjetom da se uopće koristio TCP protokol za prijenos tog paketa. Značajka *L7_PROTO* je broj koji definira korišteni protokol na aplikacijskom sloju, odnosno definira koja aplikacija ili protokol aplikacijskog sloja je korišten prilikom prijenosa paketa (npr. Facebook, IMAP, Discord, Steam, itd.). Ostale značajke predstavljaju informacije koje su standardne u mrežnim paketima, odnosno odredišnu i izvorišnu IPv4 adresu te odredišni i izvorišni broj porta, broj primljenih i poslanih paketa i količina podataka u bajtovima te vrijeme prijenosa podataka u milisekundama.

Nadovezivanjem na značajke skupova podataka u NetFlowV1 formatu, na slici 2.1 su prikazane značajke skupova podataka u NetFlowV2 formatu. S obzirom da su značajke NetFlowV2 formata nadogradnja nad značajkama NetFlowV1 formata, neke značajke NetFlowV1 formata su dodatno proširene detaljnijim značajkama NetFlowV2 formata. Primjerice, *TCP_FLAGS* značajka koja spada pod skupinu značajki NetFlowV1 formata je proširena s dodatne dvije značajke, *CLIENT_TCP_FLAGS* i *SERVER_TCP_FLAGS*. Dodatni primjeri se također mogu vidjeti u novim značajkama *RETRANSMITTED_IN_BYTES, RETRANSMITTED_IN_PKTS* te kod značajki koje predstavljaju broj paketa određene veličine, primjerice, *NUM_PKTS_UP_TO_128_BYTES i NUM_PKTS_128_TO_256_BYTES*. Još jedna nova uvedena značajka je *TCP_WIN_MAX*, kojom je određena maksimalna veličina TCP prozora primatelja podataka *(engl. TCP receiver window)*. TCP prozorom se određuje koliku količinu podataka u byte-ovima pošiljatelj može poslati primatelju, bez da dođe do odbacivanja paketa od strane primatelja. Bitno je također naglasiti da je *sliding window protocol* jedan od načina na koji TCP smanjuje utjecaj DDoS napada. Značajka *ICMP_IPV4_TYPE* određuje vrstu ICMP poruke koja se vraća u svrhu javljanja greške ili operacijskih informacija prilikom komunikacije sa nekim uređajem. Pored te značajke, dodana je također značajka *ICMP_TYPE* koja je specifična za NetFlow protokol, iz razloga jer se u NetFlow

paketu, vrsta i kod[4] ICMP poruke kodira u polje odredišnog port-a prema izrazu: $ICMP\_Type * 256 + ICMP\_Code$. Analiziranjem ICMP paketa, mogu se detektirati računalni crvi *(engl. Computer worms)*, maliciozni programi koji se samostalno repliciraju te se tako šire na sva računala u mreži. Računalni crvi, ako se šire UDP protokolom, mogu aktivirati ICMP poruku *port unreachable* u povratnim paketima te se s učestalim takvim porukama paket može klasificirati kao napadački. Za detekciju računalnog crva također se može iskoristiti *TCP_FLAGS* značajka. Primjerice, ako računalni crv skeniranjem različitih port-ova na više računala u mreži pokušava pronaći neku ranjivost, ta računala sa zatvorenim port-ovima će vratiti RST/ACK TCP zastavicu. Uočavanjem većeg broja RST/ACK zastavica u mrežnom prometu može se pretpostaviti da se radi o računalnom crvu [8]. *DNS_QUERY_TYPE* kod NetFlow protokola predstavlja vrstu DNS zapisa *(engl. DNS Record)*, a *DNS_TTL_ANSWER* predstavlja vrijeme tijekom kojeg drugi DNS serveri i aplikacije smiju držati pohranjen određeni DNS zapis, nakon čega ga moraju odbaciti te ako je potrebno tražiti novu kopiju zapisa. Vrijednost DNS TTL-a može varirati za različite web servere. Prema tome, web serveri sa velikim TTL vrijednostima mogu biti izloženi težim DDoS napadima, odnosno zbog dužeg perioda za odbacivanje DNS zapisa može doći do dužeg trajanja te većeg utjecaja DDoS napada na određenu IP adresu, odnosno web server [9]. *FTP_COMMAND_RET_CODE* predstavlja informacijski kod koji server vraća nakon što klijent izvrši određenu naredbu prema njemu. FTP je protokol koji služi za komunikaciju i prijenos podataka između klijenta i servera. Zbog toga što FTP protokol prenosi podatke u čistom tekstualnom obliku, napadač te podatke može presresti te modificirati *(engl. Man in the Middle attack)*. Također, napadač može zbog potencijalno loše konfiguriranog FTP servera, na njemu izvršiti maliciozni kod *(engl. Remote Code Execution)* te tako preuzeti kontrolu nad sustavom.

Način na koji mrežni sustavi za detekciju neovlaštenog upada (NIDS) temeljeni na strojnom učenju treniraju, predviđaju i klasificiraju mrežni promet kao maliciozni često nije intuitivno objašnjiv. Drugim riječima, strojno učenje se općenito gleda kao "crna kutija", gdje nema jasnog razumijevanja kakvi obrasci su naučeni i kakva pre-

[4]Kod ICMP *(engl. ICMP code)* poruke se koristi za detaljniji opis vrste ICMP *(engl. ICMP type)* poruke.

Feature	Description
IPV4_SRC_ADDR	IPv4 source address
IPV4_DST_ADDR	IPv4 destination address
L4_SRC_PORT	IPv4 source port number
L4_DST_PORT	IPv4 destination port number
PROTOCOL	IP protocol identifier byte
L7_PROTO	Layer 7 protocol (numeric)
IN_BYTES	Incoming number of bytes
OUT_BYTES	Outgoing number of bytes
IN_PKTS	Incoming number of packets
OUT_PKTS	Outgoing number of packets
FLOW_DURATION_MILLISECONDS	Flow duration in milliseconds
TCP_FLAGS	Cumulative of all TCP flags
CLIENT_TCP_FLAGS	Cumulative of all client TCP flags
SERVER_TCP_FLAGS	Cumulative of all server TCP flags
DURATION_IN	Client to Server stream duration (msec)
DURATION_OUT	Client to Server stream duration (msec)
MIN_TTL	Min flow TTL
MAX_TTL	Max flow TTL
LONGEST_FLOW_PKT	Longest packet (bytes) of the flow
SHORTEST_FLOW_PKT	Shortest packet (bytes) of the flow
MIN_IP_PKT_LEN	Len of the smallest flow IP packet observed
MAX_IP_PKT_LEN	Len of the largest flow IP packet observed
SRC_TO_DST_SECOND_BYTES	Src to dst Bytes/sec
DST_TO_SRC_SECOND_BYTES	Dst to src Bytes/sec
RETRANSMITTED_IN_BYTES	Number of retransmitted TCP flow bytes (src->dst)
RETRANSMITTED_IN_PKTS	Number of retransmitted TCP flow packets (src->dst)
RETRANSMITTED_OUT_BYTES	Number of retransmitted TCP flow bytes (dst->src)
RETRANSMITTED_OUT_PKTS	Number of retransmitted TCP flow packets (dst->src)
SRC_TO_DST_AVG_THROUGHPUT	Src to dst average thpt (bps)
DST_TO_SRC_AVG_THROUGHPUT	Dst to src average thpt (bps)
NUM_PKTS_UP_TO_128_BYTES	Packets whose IP size <= 128
NUM_PKTS_128_TO_256_BYTES	Packets whose IP size > 128 and <= 256
NUM_PKTS_256_TO_512_BYTES	Packets whose IP size > 256 and <= 512
NUM_PKTS_512_TO_1024_BYTES	Packets whose IP size > 512 and <= 1024
NUM_PKTS_1024_TO_1514_BYTES	Packets whose IP size > 1024 and <= 1514
TCP_WIN_MAX_IN	Max TCP Window (src->dst)
TCP_WIN_MAX_OUT	Max TCP Window (dst->src)
ICMP_TYPE	ICMP Type * 256 + ICMP code
ICMP_IPV4_TYPE	ICMP Type
DNS_QUERY_ID	DNS query transaction Id
DNS_QUERY_TYPE	DNS query type (e.g., 1=A, 2=NS..)
DNS_TTL_ANSWER	TTL of the first A record (if any)
FTP_COMMAND_RET_CODE	FTP client command return code

Slika 2.1 *Lista značajki skupova podataka u NetFlowV2 formatu (preuzeto iz [1])*

dviđanja su napravljena. Zbog ovakvih izazova organizacije često nerado pristupaju implementaciji alata temeljenih na strojnom učenju [10].

Jedan način na koji se mogu interpretirati rezultati modela strojnog učenja je određivanjem koje značajke mrežnih podataka doprinose konačnoj odluci klasifikatora (algoritma strojnog učenja). Određivanje tih značajki je kritično radi identifikacije značajki koje sadrže sigurnosne događaje, koji bi se mogli iskoristiti za dizajniranje modela ili boljeg skupa značajki [10]. Interpretacija odluka modela strojnog učenja se može objasniti računanjem doprinosa svake značajke. Doprinos svake značajke se može dobiti pomoću *Shapley vrijednosti*, koja se može iskoristiti za objašnjavanje važnosti značajki mrežnih podataka kod detekcije mrežnih napada.

Implementiranjem *SHAP*[5] metode, može se objasniti predviđanje svakog uzorka, tako što se izračuna važnost svake značajke prema doprinosu procesu predviđanja strojnog učenja [10].

[5]SHapley Additive exPlanation (SHAP) je XAI (eXplainable Artificial Inteligence) tehnika koja se temelji na metodi aditivne važnosti značajki za računanje Shapley vrijednosti [10]

Poglavlje 3

Proces strojnog učenja

Alat korišten u okviru ovoga rada za provođenje procesa strojnog učenja je programski jezik *Python*. Za provođenje procesa preliminarne obrade podataka te treniranje, testiranje i prikaz rezultata modela strojnog učenja, potrebno je uključiti sljedeće knjižnice koje sadrže potrebne gotove funkcionalnosti: *numpy, pandas, scikit-learn, matplotlib*. Za analizu i manipuliranje podacima iz različitih skupova podataka koriste se knjižnice numpy i pandas, dok se za treniranje i testiranje modela strojnog učenja koristi scikit-learn knjižnica. Potrebno je napomenuti kako se funkcije tih knjižnica izvršavaju isključivo preko procesora računala. Prema tome, treniranje i testiranje modela pomoću procesora računala, posebno nad velikim skupovima podataka i pomoću algoritama koji zahtjevaju veće količine računalnih resursa, može za posljedicu imati duže vrijeme izvršavanja. Iz tog razloga, za bržu obradu podataka te treniranje i testiranje modela koristila se kolekcija knjižnica *RAPIDS*, koja omogućuje izvršavanje procesa strojnog učenja pomoću NVIDIA grafičke kartice. Drugim riječima, zamjenom scikit-learn sa *cuml*, pandas sa *cudf* te numpy sa *cupy* knjižnicama, treniranje i testiranje modela u nekim situacijama traje umjesto 30 do 40 minuta, samo 2 do 3 minute.

Nakon uključivanja potrebnih knjižnica, potrebno je iz ciljanog skupa podataka izdvojiti značajke i oznake. Značajke (*features*) u strojnom učenju predstavljaju ulazne podatke, pomoću kojih se može realizirati model za predviđanje određenih ishoda. S druge strane, oznake (*labels*) predstavljaju izlazne vrijednosti, odnosno neki ishod koji je potrebno predvidjeti. Značajke izdvojene iz *NF-BoT-*

IoT skupa podataka su prikazane u tablici 2.1. Međutim, važno je naglasiti da su se od navedenih značajki, dodatno izbacile 4 značajke: *IPV4_SRC_ADDR, IPV4_DST_ADDR, L4_SRC_PORT i L4_DST_PORT*. Razlog za to je definiran u radu [5], gdje je navedeno da se identifikatori toka, kao što su izvorišna i odredišna IP adresa te portovi, izbacuju iz skupa značajki radi izbjegavanja pristranosti prema čvorovima napadača ili čvorovima žrtava. Nakon izdvajanja značajki i oznaka iz skupa podataka, potrebno je skalirati podatke značajki i kodirati oznake. Originalni podaci značajka su obično predstavljeni u različitim veličinama i mjerama, što može dovesti do nepouzdanog modeliranja skupa podataka. Iz tog razloga je podatke potrebno skalirati prije modeliranja, a to se može postići pomoću funkcije `StandardScaler().fit_transform(feature)`. Kodiranje oznaka je zapravo mapiranje para ključ-vrijednost, odnosno pridjeljivanje broja određenoj kategoriji. Primjerice, za trenutno korišteni skup podataka postoji stupac koji definira razne vrste napada, koji se prilikom kodiranja oznake definiraju brojem. Kodiranje oznake se može postići funkcijom `LabelEncoder().fit_transform(target)`. Sada kada su značajke skalirane i oznake kodirane, sljedeći korak je napraviti podjelu filtriranog skupa podataka na skup za treniranje i skup za testiranje. Primjerice, može se staviti da se 66% skupa podataka koristi za treniranje modela, a 34% za testiranje, odnosno vrednovanje modela. Naredba `train_test_split(feature_std, labels, test_size=0.34, random_state=42)` vraća skup za testiranje i skup za treniranje za x (značajke) i y (oznake). Na kraju, potrebno je definirati klasifikator, odnosno algoritam pomoću kojeg će se model strojnog učenja trenirati da bi mogao što točnije kategorizirati podatke. Od mnoštva algoritama koji se mogu primijeniti, svaki od njih će varirati u rezultatima s obzirom na korištene značajke u skupu podataka, te je stoga potrebno isprobati više klasifikatora te izmjenjivati značajke s ciljem postizanja boljih performansi. Dobiveni model se može vrednovati tako što na nepoznatom skupu podataka za testiranje pokušava točno klasificirati uzorke, odnosno predvidjeti određeni ishod.

3.1 Mjerenje performansi modela strojnog učenja

Za vrednovanje performansi modela strojnog učenja mogu se koristiti određene klasifikacijske metrike. Određene klasifikacijske metrike kao što su točnost, preciznost i odziv su izvedene iz matrice zabune *(engl. Confusion Matrix)* koja se sastoji od sljedećih kategorija:

- TP *(engl. True Positive)* - broj pozitivnih uzoraka koji su točno predviđeni od strane modela kao pozitivni.
- TN *(engl. True Negative)* - broj negativnih uzoraka koji su točno predviđeni od strane modela kao negativni.
- FP *(engl. False Positive)* - broj negativnih uzoraka koji su pogrešno predviđeni od strane modela kao pozitivni.
- FN *(engl. False Negative)* - broj pozitivnih uzoraka koji su pogrešno predviđeni od strane modela kao negativni.

Na temelju navedenih kategorija matrice zabune mogu se definirati sljedeće klasifikacijske metrike:

- Točnost *(engl. Accuracy)* - predstavlja broj točno predviđenih i pozitivnih i negativnih uzoraka u odnosu na ukupan broj predviđanja [11].

$$\text{Točnost} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

- Preciznost *(engl. Precision)* - predstavlja broj točno predviđenih pozitivnih uzoraka u odnosu na ukupan broj uzoraka koji su predviđeni kao pozitivni. Drugim riječima, od svih uzoraka koji su klasificirani kao pozitivni, koliko ih je uistinu pozitivno [11].

$$\text{Preciznost} = \frac{TP}{TP + FP} \tag{3.2}$$

- Odziv *(engl. Recall)* - predstavlja broj točno predviđenih pozitivnih uzoraka u odnosu na ukupan broj uzoraka koji su uistinu pozitivni. Drugim riječima, od svih uzoraka koji su uistinu pozitivni, koliko ih je klasificirano kao pozitivno

[11]. Drugi nazivi za odziv su još TPR *(engl. True Positive Rate)* i osjetljivost.

$$\text{Odziv} = \frac{TP}{TP + FN} \tag{3.3}$$

- F1-mjera *(engl. F1-score)* - mjera koja kombinira preciznost i odziv, odnosno računa njihovu harmonijsku sredinu.

$$\text{F1-mjera} = \frac{2 \times Preciznost \times Recall}{Preciznost + Recall} \tag{3.4}$$

- TNR *(engl. True Negative Rate)* - predstavlja broj točno predviđenih negativnih uzoraka u odnosu na ukupan broj uzoraka koji su uistinu negativni. Drugim riječima, TNR procjenjuje koliko uspješno model predviđa negativne uzorke. Drugi naziv za TNR je specifičnost.

$$\text{TNR} = \frac{TN}{TN + FP} \tag{3.5}$$

- Geometrijska sredina *(G-Mean)* - prema definiciji, geometrijska sredina je drugi korijen od umnoška *osjetljivosti* i *specifičnosti*. Drugim riječima, ona služi za usrednjavanje vrijednosti osjetljivosti i specifičnosti. Važno je napomenuti da u slučaju neujednačenosti vrijednosti osjetljivosti i specifičnosti geometrijska sredina se smanjuje.

$$\text{G-Mean} = \sqrt{TPR * TNR} \tag{3.6}$$

- MCC *(Matthewsov korelacijski koeficijent)* - u strojnom učenju se koristi primarno kao mjera za određivanje kvalitete binarne klasifikacije. Drugim riječima, MCC uzima u obzir sve četiri vrijednosti matrice zabune, pri čemu veća vrijednost MCC-a prikazuje bolju sposobnost modela za predviđanje uzoraka i pozitivnih i negativnih klasa, neovisno o neujednačenosti skupa podataka. Konceptualno, MCC predstavlja korelacijski koeficijent između promatranih i predviđenih binarnih klasifikacija te može sadržavati vrijednosti u rasponu od -1 do +1, gdje +1 predstavlja savršeno predviđanje, 0 nasumično predviđanje te -1 potpunu razliku između predviđenih i promatranih klasifikacija.

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3.7}$$

- ROC-AUC mjera - služi za mjerenje performansi modela strojnog učenja na različitim pragovima. U ovom kontekstu, prag (*engl. threshold*) predstavlja granicu postavljenu na intervalu između 0 (normalan promet) i 1 (napad). Prema toj granici, sve vrijednosti iznad će biti klasificirane kao napad, a sve vrijednosti ispod kao normalan promet. ROC *(engl. Receiver operating characteristic curve)* graf je definiran preko False Positive Rate-a na X osi i True Positive Rate-a na Y osi. Taj graf je koristan pri izdvajanju optimalnog praga. S druge strane, AUC *(engl. Area Under the Curve)* služi za određivanje sposobnosti modela za ispravno predviđanje normalnog prometa i napada. Drugim riječima, što je vrijednost AUC-a veća, to je model bolji u razlikovanju između normalnog prometa i napada.

Određene metrike se najbolje mogu objasniti primjerom, odnosno primjenom naivnog Bayesovog klasifikatora nad NF-BoT-IoT skupom podataka. Model dobiven pomoću skupa podataka sa podacima o računalnim čvorovima (slika 3.1a) ima točnost od 0.97. Ovako visoka vrijednost točnosti je posljedica TP vrijednosti, gdje se može vidjeti da je model predvidio 95.33% napada iz testnog dijela skupa podataka. Jedna neizbježna mana kod bilo kojeg pa tako i ovog modela je uočljiva kod FP vrijednosti, gdje je 1.03% normalnog prometa detektirano kao napad. Iako je model uspio detektirati iznimno velik broj potencijalno opasnih napada, to ne znači da je idealan i da ta zaštita od napada nema svoju cijenu. Pored toga, može se uočiti visoka vrijednost metrike preciznosti od 0.99. *Preciznost* se u ovom kontekstu može interpretirati kao omjer između broja stvarnih napada (TP) te zbroja stvarnih napada (TP) i normalnog prometa koji je krivo detektiran kao napad (FP). Uz preciznost je također vidljiva visoka vrijednost metrike odziva od 0.98. Generalno gledano za modele koji ne donose nasumične odluke, za *odziv* se može reći da je često obrnuto proporcionalan preciznosti, odnosno povećanjem odziva se smanjuje preciznost i obrnuto. Prema značenju, odziv je isti kao preciznost, jedina razlika je u tome što se mijenja nazivnik omjera iz $TP + FP$, u $TP + FN$, gdje FN predstavlja napade koji su krivo interpretirani kao normalan promet. Na odziv se također može gledati kao sposobnost modela da pronađe sve točke interesa (napade) u zadanom skupu podataka.

3.1.1 Predobrada značajki podataka

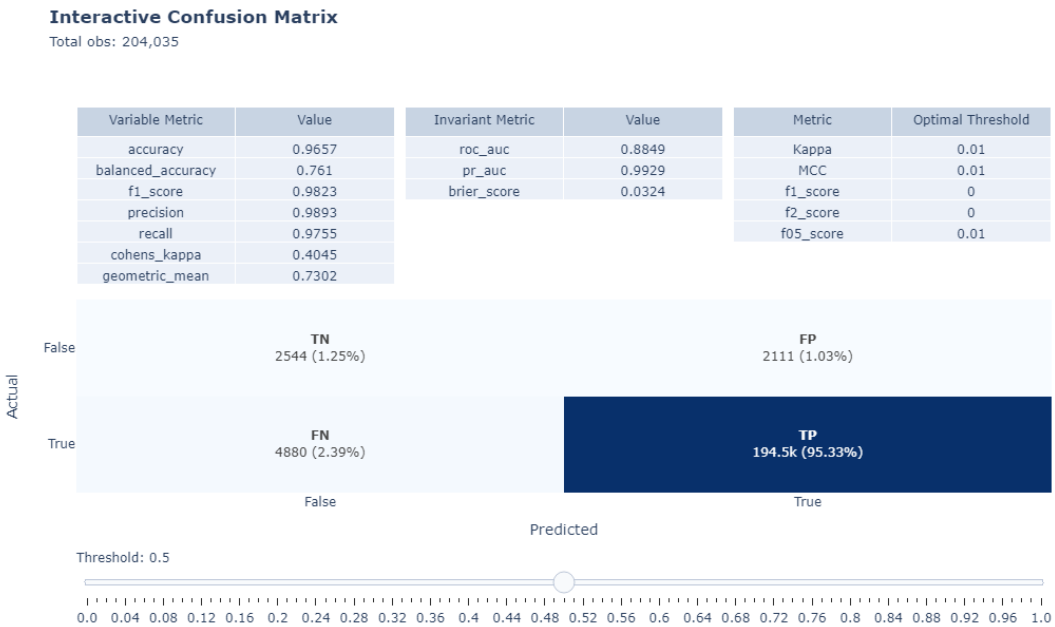
Kao što je prethodno navedeno, kako bi se potencijalno poboljšao model strojnog učenja, potrebno je odabrati i modificirati značajke iz skupa podataka. Jedan primjer ovog postupka je uočavanje razlike performansi modela strojnog učenja prilikom dodavanja ili uklanjanja značajki izvorišne i odredišne IP adrese te izvorišnog i odredišnog porta (*IPV4_SRC_ADDR, IPV4_DST_ADDR, L4_SRC_PORT i L4_DST_PORT*). Potrebno je isto naglasiti da je prilikom korištenja IP adresa kao značajki potrebno napraviti dodatne modifikacije na njima. IP adrese su prilikom izdvajanja značajki prikazane kao tekstualni tip podataka, odnosno kao string. Kako bi se klasifikacijski algoritam uspješno izvršio nad filtriranim skupom podataka, potrebno je kodirati tekstualni tip podataka pomoću funkcije `LabelEncoder().fit_transform()`. Međutim, u ovom slučaju kodiranjem tih oznaka se neće dobiti očekivani rezultat, odnosno za iste vrijednosti nekodiranih IP adresa u različitim stupcima, neće vrijediti iste kodirane vrijednosti IP adresa za različite stupce. Ovakva neujednačenost može za posljedicu imati utjecaj na rezultate modela strojnog učenja. S obzirom da se u ovom skupu podataka sve IP adrese nalaze unutar iste podmreže, može se kao značajku iz originalne IPv4 adrese izdvojiti četvrti oktet, koji zapravo čini jedinu razliku između svih adresa.

Primjenom naivnog Bayesovog klasifikatora nad istim skupom podataka, uz navedene promjene, mogu se dobiti statistički podaci, koji se zatim mogu analizirati (slika 3.1). Model dobiven pomoću skupa podataka sa podacima o čvorovima ima točnost od 0.966, dok model bez tih značajki ima točnost od 0.940. Ovakva razlika u rezultatima se može i pretpostaviti, jer model sa identifikatorima toka može razlikovati napadački promet od običnog prometa na temelju čvorova između kojih se podaci izmjenjuju. Drugim riječima, dolazi do prethodno navedene pristranosti modela prema napadačkim čvorovima, odnosno čvorovima žrtava. U ovom skupu podataka, pristranost prema određenim čvorovima je uzrokovana učestalim ponavljanjem tih istih čvorova.

Kako bi se bolje prikazala razlika između točnosti navedenih skupova podataka, može se uvesti metrika geometrijske sredine. Trenutni skup podataka nad kojim se provode izmjene značajki i testiranje modela, sadrži 97.69% napadačkih uzoraka i

2.31% uzoraka definiranih kao normalan promet. Ova neujednačenost između napadačkih i normalnih uzoraka se može povezati sa prethodno navedenim rezultatima točnosti. Model na slici 3.1b ima geometrijsku sredinu od 0.867. S obzirom na iznimno veliku neujednačenost klasificiranih uzoraka skupa podataka, može se reći da je prema geometrijskoj sredini, uspješnost predviđanja napada i normalnog prometa relativno ujednačena. S druge strane, kod modela na slici 3.1a se može uočiti da je geometrijska sredina jednaka 0.730, što je značajno manje od prethodnog modela. Dakle, kod modela 3.1a je veća točnost, a kod modela 3.1b je veća geometrijska sredina. Na temelju izdvojenih podataka, može se zaključiti da je model s identifikatorima toka, radi pristranosti prema određenim čvorovima, više uzoraka predvidio kao napad te je zbog iznimno neujednačenog omjera napada i normalnog prometa statistički više puta imao točno predviđanje. Drugim riječima, taj model je imao veću stopu osjetljivosti nego specifičnosti.

S obzirom da prvi model ima manju preciznost, ali veći odziv od drugog modela, za mjerenje performansi se može gledati F1-mjera. Prema rezultatima, prvi model ima F1-mjeru od 0.982, a drugi od 0.969, što znači da je očekivano prvi model bolji za predviđanje napada. Međutim, isto tako se može iznijeti pretpostavka da u nekim stvarnim scenarijima, gdje bi došlo do veće varijacije kod identifikatora toka, ne bi bilo značajne razlike u performansama između ta dva modela.



(a) *Dodavanje identifikatora toka*

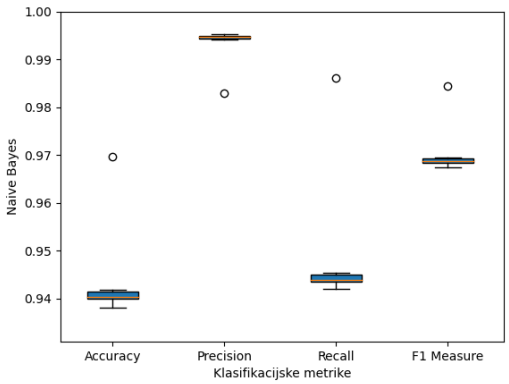
(b) *Uklanjanje identifikatora toka*

Slika 3.1 *Usporedba performansi modela promjenom značajki identifikatora toka*

3.1.2 Unakrsna validacija u k-preklopa

Unakrsna validacija u k-preklopa *engl. K-Fold Cross-validation* je metoda koja se koristi za procjenu performansi modela strojnog učenja. Ta metoda sadrži parametar k koji se odnosi na broj grupa na koje se određeni skup podataka može podijeliti. Primjerice, ako se za vrijednost parametra k uzme broj 10, model strojnog učenja će se trenirati i testirati 10 puta, svaki put sa drukčijim skupom za treniranje i skupom za testiranje podataka. Drugim riječima, od 10 izrađenih grupa podataka, kod svake iteracije će se izabrati jedna grupa za testiranje modela, a ostale grupe će se zajednički koristiti za treniranje modela.

U Pythonu se unakrsna validacija u k-preklopa može primijeniti pomoću funkcije `KFold(n_splits, random_state, shuffle)`. Za analizu performansi modela, koji je treniran i testiran nad raznim grupama podataka, mogu se kao i prije izdvojiti mjere klasifikacijskih metrika. Na slici 3.2 se mogu vidjeti dobiveni rezultati pomoću naivnog Bayesovog klasifikatora za sve iteracije metode. Iz dobivenog grafa se može uočiti veća vrijednost mjere preciznosti od odziva. Prema tome, za procjenu omjera rezultata između preciznosti i odziva može se uzeti F1-mjera koja iznosi oko 0.97. Pored tih rezultata, može se očitati i vrijednost točnosti koja iznosi oko 0.94. Pozitivna stvar kod ovih rezultata je nisko odstupanje između vrijednosti metrika svake iteracije unakrsne validacije u k-preklopa, odnosno kod svake iteracije sve vrijednosti su približno iste osim jedne iznimke koja se na grafu može vidjeti kao prazan kružić. Uz ovakve rezultate za model treniran naivnim Bayesovim klasifikatorom, potrebno je dodatno izvršiti unakrsnu validaciju u k-preklopa nad drugim algoritmima te usporediti rezultate s ciljem pronalaska algoritma koji će dati najbolje rezultate, bez prevelikih oscilacija.



Slika 3.2 *Shema kutijastog dijagrama*

3.2 Opis korištenih klasifikatora

3.2.1 Naivni Bayesov klasifikator

U strojnom učenju klasifikacijski problem se može prikazati kao klasu, koju je potrebno dodijeliti određenom podatkovnom uzorku. Drugim riječima, potrebno je odrediti najbolju hipotezu h za dani skup vrijednosti značajki d. Jedan način na koji se zadani problem može riješiti je pomoću *Bayesovog Teorema*, koji se temelji na računanju vjerojatnosti hipoteze prema određenom predznanju o podacima. Bayesov Teorem se može prikazati preko jednadžbe (3.8).

$$P(h|d) = \frac{P(d|h) * P(h)}{P(d)} \tag{3.8}$$

U toj jednadžbi navedeni izrazi se mogu objasniti na sljedeći način:

- **P(h|d)** predstavlja vjerojatnost da je hipoteza h istinita, pod uvjetom da je zadan određeni podatkovni uzorak d. To se ujedno naziva i *posteriorna vjerojatnost.*
- **P(d|h)** predstavlja vjerojatnost pojave uzorka d pod uvjetom da je hipoteza h istinita.
- **P(h)** je vjerojatnost da je hipoteza h istinita neovisno o podatkovnom uzorku. Ovo se naziva *prethodnom vjerojatnošću* hipoteze h.

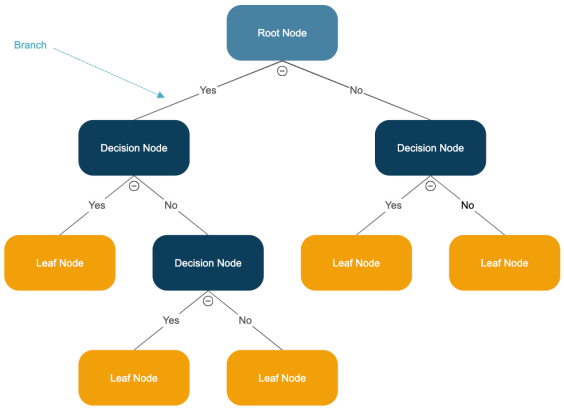
- **P(d)** je vjerojatnost pojave uzorka neovisno o hipotezi.

Prilikom provođenja procesa klasifikacije nad nekim uzorkom, Bayesov Teorem se može primijeniti tako da se izračuna posteriorna vjerojatnost za sve klase te se konačna odluka o klasifikaciji uzorka donosi na temelju maksimalne dobivene vjerojatnosti od svih klasa. Na navedenom principu funkcionira algoritam naivni Bayesov klasifikator *(engl. Naive Bayes)*, s time da se on temelji na pretpostavci da su sve značajke nekog uzorka uvjetno nezavisne te se iz tog razloga i naziva "naivni" Bayes. Ta pretpostavka često nije u skladu sa podacima iz stvarnog svijeta s obzirom da najčešće postoji neka povezanost među različitim značajkama. U kontekstu ovoga rada, prilikom treniranja i testiranja modela strojnog učenja, korišten je *Gaussian Naive Bayes* klasifikator, koji se razlikuje od običnog naivnog Bayes-a po tome što uz kategoričke podatke podržava i kontinuirane vrijednosti.

3.2.2 Stablo odluke

Algoritam stablo odluke *(engl. Decision Tree)* se u strojnom učenju primjenjuje za klasifikacijske i regresijske probleme. Kako samo ime kaže, algoritam stabla odluke koristi strukturu stabla (slika 3.3) kako bi prikazao predviđanja nastala prema podjeli temeljenoj na značajkama. Drugim riječima, unutarnji čvorovi stabla *(engl. Decision Nodes)* predstavljaju određene značajke, a grane ispod svakog čvora *(engl. Branches)* predstavljaju vrijednosti tih značajki, za koje se provjerava neki uvjet. O ispunjenosti ili neispunjenosti određenog uvjeta, ovisi koja će se sljedeća značajka, odnosno čvor provjeravati. Nakon provjere uvjeta određenog broja čvorova, odnosno značajki u konačnici se dolazi do određenog krajnjeg čvora, lista *(engl. Leaf Node)* koji predstavlja klasifikacijsku odluku za neki uzorak.

Algoritam stabla odluke definira najbolje značajke za određene čvorove stabla na temelju metrike entropije, koja predstavlja količinu nesigurnosti, odnosno mjeru poremećaja u nekom skupu podataka. Koncept entropije je u algoritmu stabla odluke implementiran na principu određivanja stupnja nasumičnosti u određenom čvoru. Za primjer se može uzeti neki čvor, koji predstavlja određenu značajku te sadrži određeni broj uzoraka različitih klasa. Ti uzorci se na temelju uvjeta koji se provjerava u zadanom čvoru, raspoređuju u lijevi ili desni čvor. Ako se u tim novonastalim



Slika 3.3 *Struktura stabla odluke (preuzeto iz [2])*

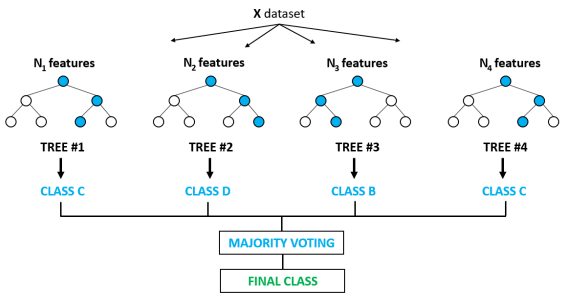
čvorovima i dalje nalaze uzorci različitih klasa, ti čvorovi sadrže određenu razinu nasumičnosti te se rekurzivno dalje dijele na nove čvorove. Rekurzivno dijeljenje čvorova prestaje kada lijevo dijete čvora roditelja sadrži uzorke jedne klase, a desno dijete sadrži uzorke druge klase. U tom slučaju, čvorovi koji sadrže uzorke jedinstvene klase se definiraju kao krajnji čvorovi, odnosno listovi i njihova vrijednost entropije je jednaka nuli. Još jedna metrika koju je potrebno spomenuti je informacijska dobit *(engl. information gain)*. Ta metrika služi za mjerenje razine smanjenja entropije nakon podjele prema nekoj značajci. Drugim riječima, informacijska dobit je razlika između entropije cijelog skupa podataka i entropije skupa podataka prema određenoj značajci. Informacijska dobit se u algoritmu stabla odluke koristi za odabir najbolje značajke, koja se zatim dodjeljuje korijenskom čvoru *(engl. Root Node)*, odnosno nekom unutarnjem čvoru prema kriteriju maksimalnog smanjenja entropije.

3.2.3 Slučajna šuma

Slučajna šuma *(engl. Random Forest)* je klasifikacijski algoritam, koji se temelji na stvaranju velikog broja pojedinačnih stabala odluke primjenom metode ansambla *(engl. ensemble)*. Metoda ansambla je tehnika u strojnom učenju koja povezuje više modela, kako bi se dobio jedan bolji klasifikacijski model. Prema toj metodi, svako izgrađeno stablo odluke vraća predviđenu klasu te se zatim konačna klasifikacijska odluka donosi prema tome koja je klasa najzastupljenija među svim stablima od-

luke (slika 3.4). Jedno bitno svojstvo na kojem se temelji algoritam slučajne šume je niska razina korelacije između izgrađenih stabala odluke, čime se ona međusobno izoliraju od vlastitih pojedinačnih grešaka. Način na koji algoritam slučajne šume ostvaruje nisku razinu korelacije među izgrađenim stablima odluke je primjenom *bagging (bootstrap aggregation) metode*. Bagging metoda funkcionira na način da svako stablo odluke nasumično izabire određeni broj uzoraka iz skupa podataka, s tim da je moguća pojava duplih uzoraka. Primjenom ove metode se kod stabala odluke iskorištava svojstvo osjetljivosti na podatke za treniranje, prema kojem manje promjene u skupu podataka za treniranje mogu rezultirati potpuno drukčijom strukturom stabla odluke. Uz nasumičan odabir uzoraka za treniranje, svakom stablu odluke se nasumično dodjeljuje i skup značajki, na temelju kojih ono donosi klasifikacijske odluke. Takva nasumična dodjela značajki nije uvijek pogodna za svako stablo odluke, odnosno neki modeli mogu potencijalno donijeti lošije klasifikacijske odluke na temelju izuzetka određenih značajki. Međutim, takav pristup rezultira većom varijacijom među stablima odluke, čime se smanjuje njihova međusobna korelacija. Također je bitno spomenuti, da je kod algoritma slučajne šume znatno manja vjerojatnost pojave *prenaučenosti (engl. overfitting)*[1] modela, što je rezultat smanjene korelacije među stablima odluke te nasumičnog odabira uzoraka i značajki za treniranje. U ovome radu za parametre algoritma slučajne šume su se koristile zadane vrijednosti. Odnosno, algoritam slučajne šume sadrži 100 stabala u šumi, maksimalna dubina svakog stabla je 16, omjer broja značajki koje treba uzeti u obzir po podjeli čvora je postavljen na recipročnu vrijednost drugog korijena od ukupnog broja značajki te kriterij za dijeljenje čvorova je postavljen na vrijednost *gini*. Gini nasumičnost skupa podataka se može objasniti kao vrijednost između 0 i 0.5, koja ukazuje na vjerojatnost da su novi nasumični uzorci pogrešno klasificirani, ako im je oznaka dodijeljena nasumično. Prema tome, za odabir sljedeće najbolje značajke, odnosno čvora u stablu odluke, za kriterij se uzima najmanja vrijednost gini nasumičnosti.

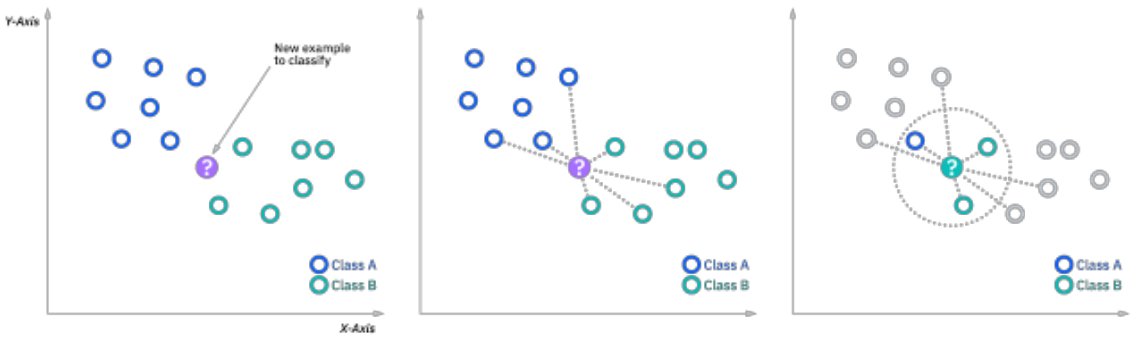
[1]Do prenaučenosti dolazi kada model nakon treniranja na jednom skupu podataka, ne može raditi uspješna predviđanja na drugom, nepoznatom skupu podataka. Drugim riječima, ta pojava govori da model nije uspio naučiti obrasce potrebne za klasifikaciju nepoznatih uzoraka.



Slika 3.4 *Struktura algoritma slučajne šume (preuzeto iz [3])*

3.2.4 Algoritam k-najbližih susjeda

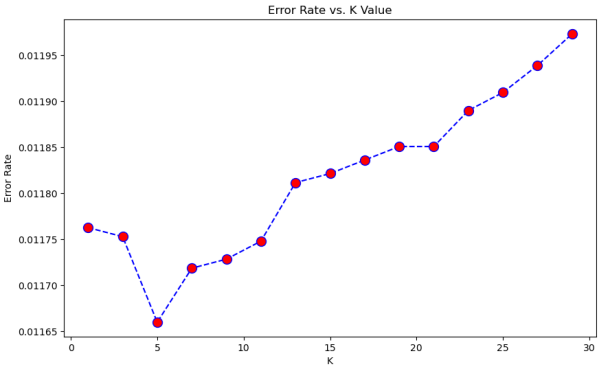
Algoritam k-najbližih susjeda *(engl. K-Nearest Neighbours, KNN)* je algoritam koji radi na principu lociranja određenog broja susjednih uzoraka oko novog, nepoznatog uzorka, kako bi mogao donijeti odluku kojoj klasi taj uzorak pripada (slika 3.5). Za odabir susjednih uzoraka, KNN algoritam računa euklidsku udaljenost od novog podatkovnog uzorka do svih uzoraka koji se nalaze u neposrednoj blizini. Na temelju skupa izračunatih udaljenosti, KNN izabire predodređeni broj uzoraka K, koji su najbliži neklasificiranom uzorku. Iz odabranih uzoraka, klasifikacijska vrijednost novog uzorka se dobiva tako da se uzme ona klasa koja je većinska od K uzoraka.



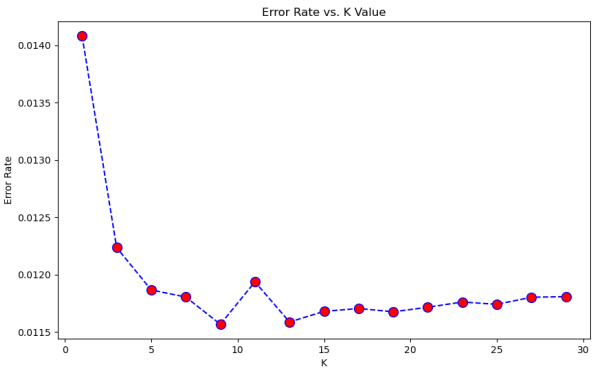
Slika 3.5 *KNN algoritam (preuzeto iz [4])*

Bitno je također napomenuti da je za K preporučeno uzimati neparne vrijednosti, kako ne bi došlo do situacije u kojoj skup od K uzoraka ima jednak broj uzoraka

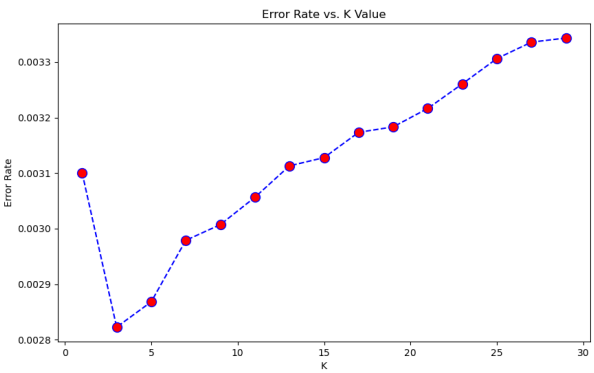
jedne i druge klase, pri čemu se ciljani uzorak klasificira nasumično. Prilikom odabira K vrijednosti, u obzir je potrebno uzeti utjecaj većih i manjih vrijednosti na uspješnost klasifikacije. Ako se uzima veća vrijednost K, može doći do pojave *nedovoljne naučenosti (engl. underfitting)*, pri čemu model ne može naučiti obrasce prilikom treniranja. S druge strane, manja vrijednost K može uzrokovati pojavu prenaučenosti, što može smanjiti performanse modela prilikom testiranja. Optimalna vrijednost K se može odabrati primjenom unakrsne validacije u k-preklopa i uspoređivanjem performansi KNN klasifikatora za različite K vrijednosti. Drugim riječima, za optimalni K se uzima vrijednost kod koje je greška klasifikacije uzoraka *(engl. Error Rate)* minimalna. Navedeni postupak je proveden za sve korištene skupove podataka te su rezultati prikazani grafovima 3.6. Iz prikazanih grafova mogu se odrediti optimalne K vrijednosti za svaki skup podataka. Za NF-BoT-IoT optimalni K iznosi 5, za NF-UNSW-NB15 iznosi 9 te za NF-UNSW-NB15-V2 iznosi 3. Jedna mana KNN algoritma je da model treniran na neujednačenom skupu podataka donosi klasifikacijske odluke na temelju pristranosti, odnosno prema većinskoj klasi koja prevladava u čitavom skupu podataka. Iz tog razloga, pouzdanost KNN klasifikatora može biti upitna prilikom korištenja neujednačenog skupa podataka te se stoga za procjenu modela strojnog učenja trebaju analizirati metrike na koje neujednačeni podaci nemaju velikog utjecaja.



(a) *Optimalni K - NF-BoT-IoT*



(b) *Optimalni K - NF-UNSW-NB15*



(c) *Optimalni K - NF-UNSW-NB15-V2*

Slika 3.6 *KNN - optimalne K vrijednosti*

Poglavlje 4

Rezultati

4.1 Procjena modela strojnog učenja - NF-BoT-IoT

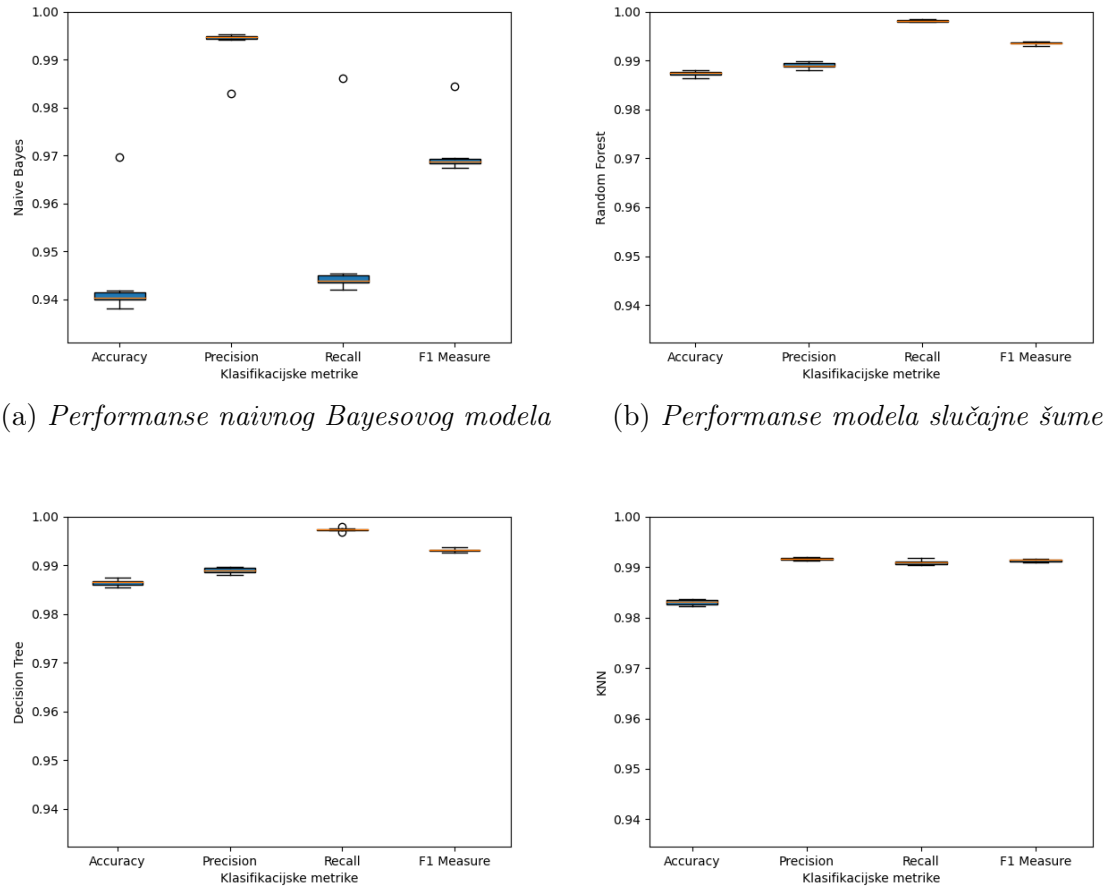
Nadovezivanjem na prethodne rezultate dobivene naivnim Bayesovim klasifikatorom i unakrsnom validacijom u k-preklopa, provedeno je dodatno treniranje i testiranje modela pomoću još tri algoritma: slučajna šuma, stablo odluke i k-najbližih susjeda (KNN). Performanse modela dobivenih pomoću navedenih algoritama su vidljive na slici 4.1.

Iz navedenih rezultata se može uočiti, da u usporedbi sa naivnim Bayesovim klasifikatorom ostala tri imaju znatno bolje rezultate po pitanju točnosti, odziva i F1-mjere. Međutim te rezultate je potrebno dodatno analizirati, odnosno staviti ih u kontekst detekcije napada i normalnog prometa. Na slici 4.1a se može vidjeti da model treniran naivnim Bayesovim klasifikatorom ima veću vrijednost preciznosti u odnosu na odziv. Kod takvih rezultata, može se reći da će model sa određenom sigurnošću klasificirati neki uzorak kao napad, iako će kao posljedica toga određeni broj napadačkih uzoraka biti krivo klasificiran kao normalan promet. Pored toga, na slikama 4.1b i 4.1c se može uočiti obrnuta situacija, odnosno kod modela treniranih algoritmima slučajne šume i stabla odluke, vrijednost odziva je veća od preciznosti. U toj situaciji, broj ispravno klasificiranih napadačkih uzoraka će se povećati, ali će također među njima biti određen broj pogrešno klasificiranih uzoraka koji predstavljaju normalan promet. Drugim riječima, povećanjem odziva posljedično će se

smanjiti točnost klasifikacije uzoraka. Također, potrebno je uočiti na slici 4.1d da u usporedbi sa ostalim algoritmima, KNN algoritam ima relativno ujednačene rezultate metrika preciznosti i odziva, što nije uobičajeno jer su te dvije klasifikacijske metrike međusobno komplementarne, što potvrđuju i prethodni rezultati. Na temelju izvedenih zaključaka, ako bi se kao mjerodavan kriterij gledala ujednačenost vrijednosti svake od navedenih metrika, logičan odabir za optimalan model bi bio onaj treniran KNN algoritmom. Međutim, za donošenje takvih odluka, najčešće se koristi metrika koja sažima utjecaj preciznosti i odziva, a to je F1-mjera. Prema tome, uspoređujući vrijednosti F1-mjere za prethodno navedene modele, može se zaključiti da su prema tom kriteriju optimalni modeli oni trenirani pomoću algoritama slučajne šume i stabla odluke.

Uz sve prethodne rezultate, navedene metrike se ne mogu u ovom slučaju uzeti kao vjerodostojni ocjenjivači performansi modela. Razlog za to je niska vrijednost geometrijske sredine, koja iznosi oko 0.73 za algoritme slučajne šume i stabla odluke te oko 0.8 za KNN algoritam (slika 4.2b). Iz tih vrijednosti se može zaključiti da postoji neujednačenost modela u uspješnosti predviđanja između normalnog prometa i napada. Vrijednosti geometrijske sredine ujedno potvrđuje i korišteni skup podataka koji sadrži neujednačen broj uzoraka napada (97.69%) i normalnog prometa (2.31%). Iznimka je u ovom slučaju model dobiven naivnim Bayesovim klasifikatorom čija geometrijska sredina iznosi oko 0.87. Iz toga se može zaključiti da je naivni Bayesov klasifikator potencijalno bolji algoritam za klasifikaciju, ako se koristi neujednačeni skup podataka.

Uz geometrijsku sredinu, za određivanje uspješnosti modela prilikom korištenja neujednačenog skupa podataka, može se koristiti i *Matthewsov korelacijski koeficijent (MCC)*. Primjena MCC-a i razlika u rezultatima za modele trenirane različitim algoritmima vidljiva je na slici 4.2a. Ako se usporede vrijednosti geometrijske sredine i MCC-a za sve korištene algoritme (slika 4.2), može se uočiti da za algoritme slučajne šume, stabla odluke i KNN, vrijednosti geometrijske sredine odgovaraju vrijednostima MCC-a. Iako se na prikazanim grafovima vide određena odstupanja u vrijednostima za navedene algoritme, potrebno je također uzeti u obzir da je raspon vrijednosti MCC-a od -1 do +1, dok je raspon geometrijske sredine od 0 do +1. Iz prikazanih vrijednosti se također može vidjeti da model treniran naivnim Baye-



(a) *Performanse naivnog Bayesovog modela*

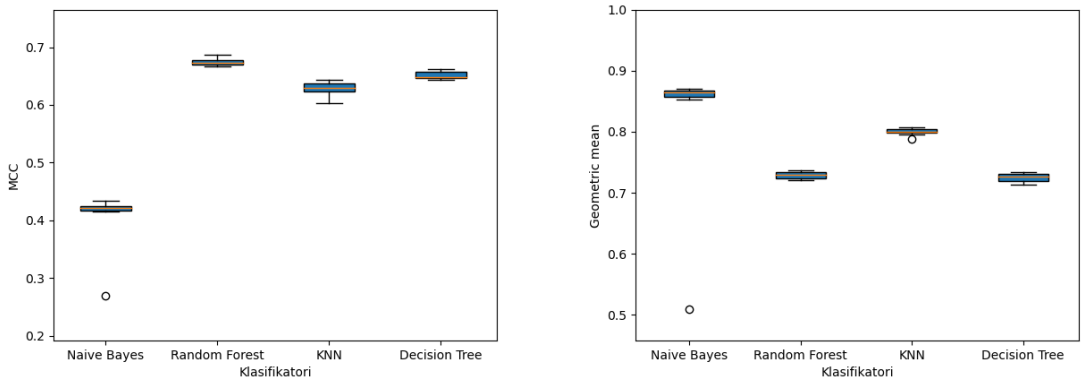
(b) *Performanse modela slučajne šume*

(c) *Performanse modela stabla odluke*

(d) *KNN performanse*

Slika 4.1 *Usporedba performansi algoritama*

sovim klasifikatorom, u usporedbi sa vrijednosti geometrijske sredine od 0.87 ima neočekivano nisku vrijednost MCC-a od oko 0.42. Iz toga se može zaključiti, da model treniran naivnim Bayesovim klasifikatorom potencijalno nije toliko dobar odabir prilikom korištenja neujednačenog skupa podataka kako je prije bilo navedeno.



(a) *Matthewsov korelacijski koeficijent algoritama*

(b) *Geometrijska sredina algoritama*

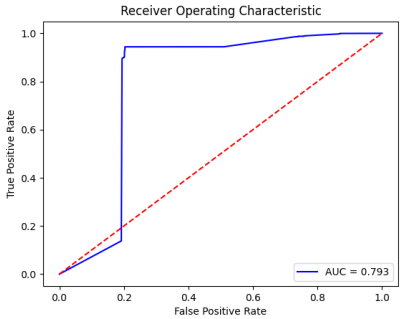
Slika 4.2 *Usporedba MCC i G-Mean vrijednosti algoritama*

4.1.1 ROC-AUC krivulja

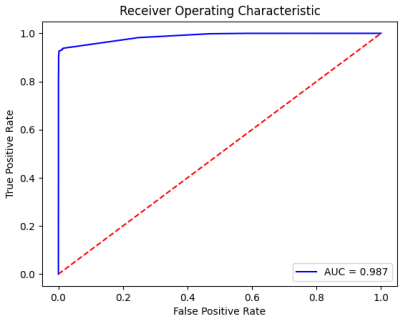
S druge strane, pored svih prethodno navedenih klasifikacijskih metrika, bolje vrednovanje performansi modela nad neujednačenim skupom podataka se može postići pomoću ROC krivulje i površine ispod te krivulje (AUC). Na slici 4.3 su prikazane ROC-AUC krivulje prethodno navedenih modela. Prema prikazanim AUC vrijednostima se može zaključiti da je model, koji je treniran pomoću algoritma slučajne šume, najbolji po pitanju razlikovanja normalnog prometa i napada. Odnosno model treniran algoritmom slučajne šume ima AUC vrijednost od 0.987. Model treniran algoritmom stabla odluke ima AUC vrijednost od 0.983, što je očekivano blizu modela slučajne šume. Nešto nižu AUC vrijednost od 0.828 ima model treniran pomoću KNN algoritma te najnižu AUC vrijednost od 0.793 ima model treniran naivnim Bayesovim klasifikatorom. Te vrijednosti se mogu provjeriti primjenom unakrsne validacije u k-preklopa. Na slici 4.4 se može uočiti kutijasti dijagram, izrađen pomoću 10 iteracija unakrsne validacije u k-preklopa, na kojem se vidi da AUC vrijednosti modela treniranih algoritmima KNN i naivnim Bayesovim klasifikatorom odstupaju od prethodno dobivenih vrijednosti. Drugim riječima, prethodno navedeni rezultati za ta dva modela predstavljaju nekakve iznimke *(eng. outliers)* u rezultatima. Prema tome, iz kutijastog dijagrama se može zaključiti da se skoro sve AUC vrijednosti na-

ivnog Bayesovog modela kreću oko 0.85, a kod KNN modela oko 0.93.

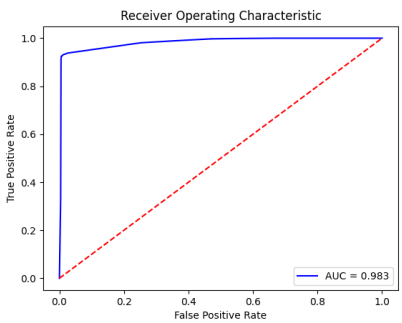
Ove AUC vrijednosti se mogu usporediti sa prethodno dobivenim vrijednostima geometrijske sredine. Može se reći da su prema dobivenim AUC vrijednostima, modeli trenirani pomoću algoritama slučajne šume i stabla odluke iznimno dobri prilikom razlikovanja između napada i normalnog prometa. Međutim, iz perspektive geometrijske sredine ti algoritmi imaju niske rezultate od oko 0.73 te bi na neujednačenom skupu podataka potencijalno bolji odabir bio model treniran naivnim Bayesovim klasifikatorom, koji ima vrijednost od 0.87. Prema tome, u ovoj situaciji prilikom korištenja neujednačenog skupa podataka nije moguće trenirati model bez određenih mana. Drugim riječima, iako algoritmi slučajne šume i stabla odluke imaju bolje performanse od naivnog Bayesovog klasifikatora, vjerojatnost je da su te performanse ostvarene radi neujednačenog omjera između napadačkih uzoraka i uzoraka normalnog prometa. Upravo u toj situaciji je naivni Bayesov klasifikator potencijalno bolji odabir jer je na temelju vrijednosti geometrijske sredine uspješnost klasifikacije napada i normalnog prometa više ujednačena. Po pitanju KNN algoritma, vrijednost geometrijske sredine je veća od algoritama slučajne šume i stabla odluke te iznosi oko 0.80. Međutim, AUC vrijednost je nešto niža, odnosno iznosi oko 0.93 prema slici 4.4. Prema tim rezultatima, model treniran KNN algoritmom se može potencijalno uzeti kao optimalan odabir između naivnog Bayesovog modela s višom vrijednosti geometrijske sredine te modela slučajne šume i stabla odluke s višom AUC vrijednosti.



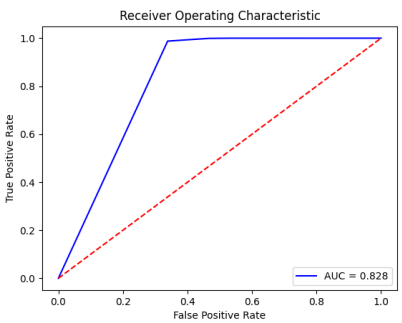
(a) *ROC-AUC naivnog Bayesovog modela*



(b) *ROC-AUC modela slučajne šume*

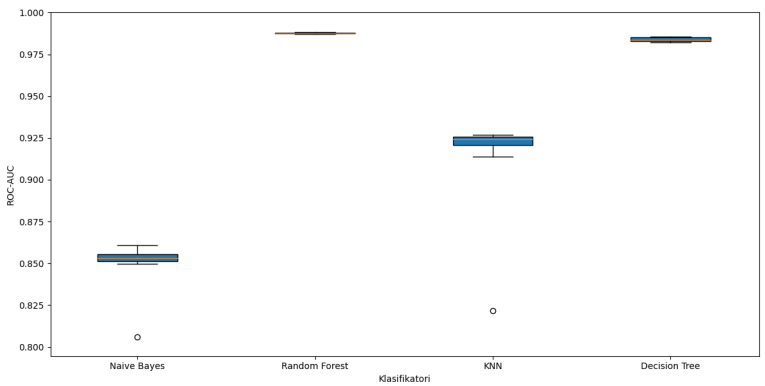


(c) *ROC-AUC modela stabla odluke*



(d) *KNN ROC-AUC*

Slika 4.3 *Usporedba ROC-AUC krivulja algoritama*



Slika 4.4 *Usporedba AUC vrijednosti algoritama*

4.1.2 Distribucija vjerojatnosti testnog skupa podataka

Pored ROC-AUC krivulje, za određivanje optimalnog praga se može iskoristiti i krivulja distribucije vjerojatnosti. Krivulja distribucije vjerojatnosti u ovom kontekstu služi za prikaz koliko je uzoraka i s kojom vjerojatnošću predviđeno kao napad, odnosno normalan promet. Ako se uzme primjerice, model treniran algoritmom slučajne šume, za njega se može prikazati krivulja distribucije vjerojatnosti testnog skupa podataka. Krivulja distribucije je vidljiva na slici 4.5a i ona zapravo daje detaljniji grafički prikaz matrice zabune (slika 4.5b).

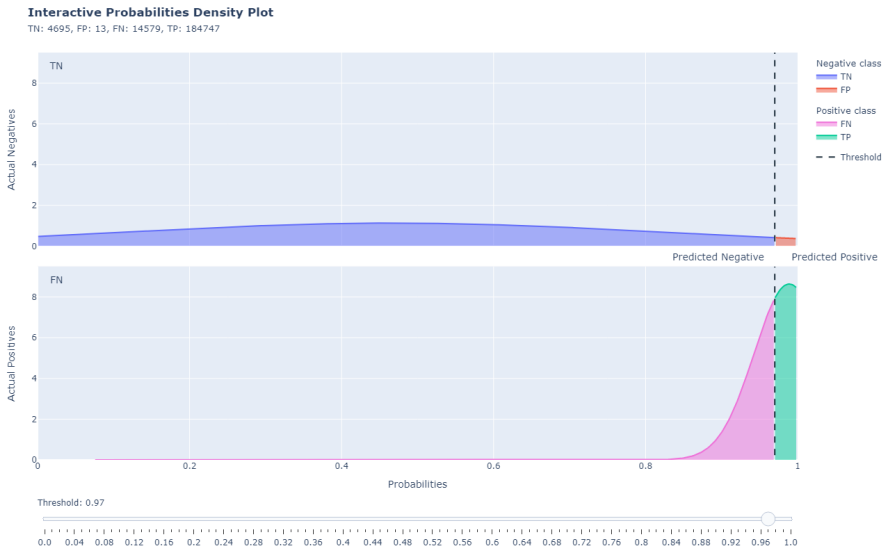
Iz prikazane distribucije vjerojatnosti se može uočiti da zadani prag prilikom testiranja modela iznosi 0.5. To znači da će se prilikom testiranja modela sve vjerojatnosti predviđanja veće od 0.5 klasificirati kao napad, a sve manje od 0.5 kao normalan promet. Da bi se postigla ujednačena uspješnost između predviđanja napada i normalnog prometa potrebno je navedeni prag pomaknuti tako da omjer između *False Positive Rate-a* i *True Positive Rate-a* bude optimalan. Takav optimalan prag se može dobiti pomoću ROC krivulje i geometrijske sredine na način da se uzmu sve vrijednosti *TPR-a* i *FPR-a* koje se koriste za izradu ROC krivulje te se za svaku od tih vrijednosti izračuna geometrijska sredina. Od svih dobivenih geometrijskih sredina, uzima se ona s najvećom vrijednošću, odnosno uzimaju se njezini *TPR* i *FPR* pomoću kojih se može odrediti traženi prag. Prema dobivenom optimalnom pragu (0.97) može se izraditi nova distribucija vjerojatnosti za model slučajne šume (slika 4.6a). Postupak dobivanja praga je u opisanom slučaju optimalan za ROC krivulju. Na slici 4.6b se može uočiti prilično visoka vrijednost geometrijske sredine. Uspoređivanjem te vrijednosti sa vrijednošću geometrijske sredine na slici 4.5b, može se uočiti da se pomicanjem praga, ujednačenost uspješnosti predviđanja i napada i normalnog prometa znatno poboljšala. Međutim, kao posljedica pomicanja praga s ciljem povećanja vrijednosti geometrijske sredine, na slikama 4.5b i 4.6b se može uočiti da se smanjila uspješnost modela za predviđanje napada, odnosno smanjila se vrijednost F1-mjere sa 0.99 na 0.96. To je ujedno direktna posljedica promjene vrijednosti odziva i preciznosti. Na prethodno navedenim slikama, se može uočiti promjena tih vrijednosti, odnosno može se vidjeti da se s povećanjem preciznosti do praktički savršene vrijednosti od 0.99, vrijednost odziva smanjila sa 0.99 na 0.93. Iz navedenih promjena vrijednosti klasifikacijskih metrika, može se zaključiti da se s

promjenom praga s ciljem povećanja geometrijske sredine, dodatni naglasak stavlja na povećanje preciznosti modela s ciljem smanjenja broja FP vrijednosti.

(a) *Distribucija vjerojatnosti modela slučajne šume*

(b) *Matrica zabune modela slučajne šume*

Slika 4.5 *Distribucija vjerojatnosti i matrica zabune za model slučajne šume*



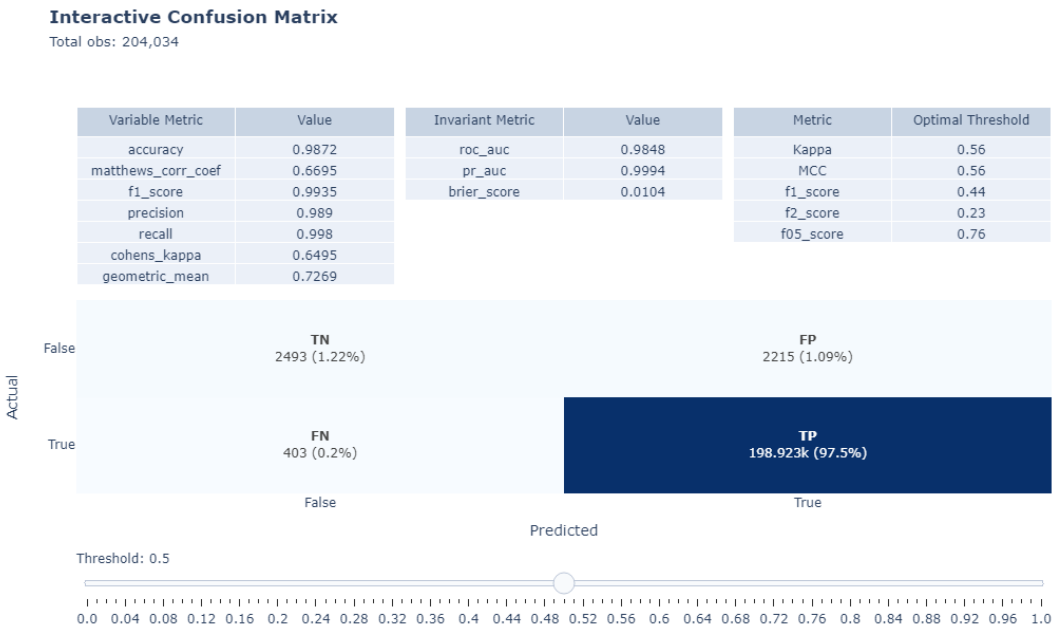
(a) *Distribucija vjerojatnosti modela slučajne šume*

(b) *Matrica zabune modela slučajne šume*

Slika 4.6 *Model slučajne šume - optimalni prag za ROC krivulju*

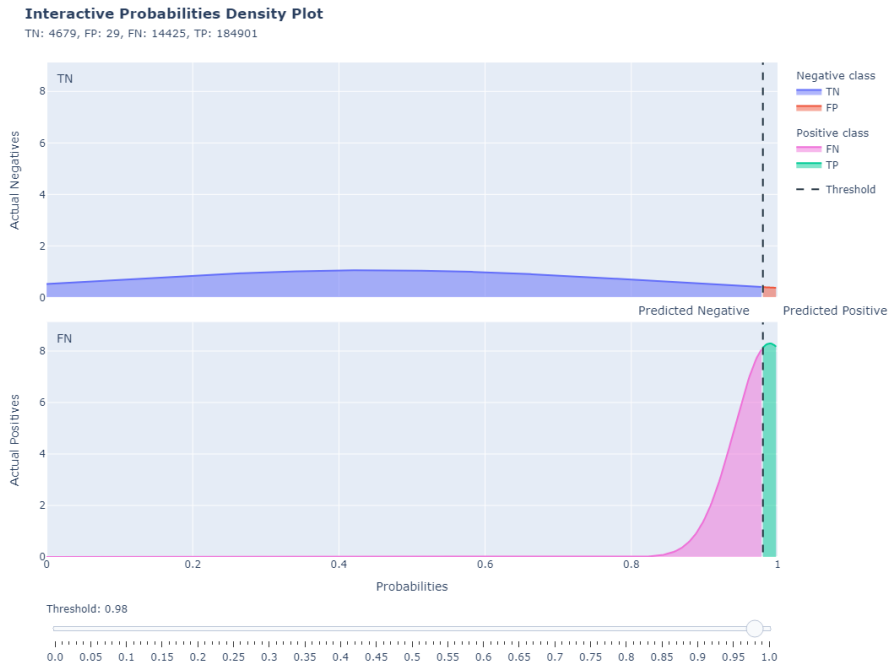
Kao što je napravljena distribucija vjerojatnosti za model treniran algoritmom slučajne šume, tako se može napraviti i za ostale korištene klasifikatore. Krivulja distribucije vjerojatnosti za model treniran algoritmom stabla odluke je vidljiva na slici 4.7a. Navedena krivulja distribucije za zadani prag od 0.5 se može usporediti sa krivuljom distribucije modela slučajne šume sa slike 4.5a. Dvije prikazane krivulje izgledaju gotovo identično te se prema tome mogu pretpostaviti i slične vrijednosti klasifikacijskih metrika, koje su vidljive za model stabla odluke na slici 4.7b te za model slučajne šume na slici 4.5b. Računajući prag optimalan za ROC krivulju za model stabla odluke, dobije se vrijednost od 0.98, što je približno jednako optimalnom pragu modela slučajne šume od 0.97, što je i očekivano s obzirom na sličnosti u krivuljama distribucije vjerojatnosti. Na temelju približno istih vrijednosti optimalnog praga te gotovo identičnih krivulja distribucije vjerojatnosti, za modele stabla odluke i slučajne šume mogu se potvrditi slične vrijednosti klasifikacijskih metrika. Sličnosti klasifikacijskih metrika i krivulja distribucije za modele slučajne šume i stabla odluke mogu se uočiti na slikama 4.6 i 4.8 respektivno.

(a) *Distribucija vjerojatnosti modela stabla odluke*



(b) *Matrica zabune modela stabla odluke*

Slika 4.7 *Distribucija vjerojatnosti i matrica zabune za model stabla odluke*

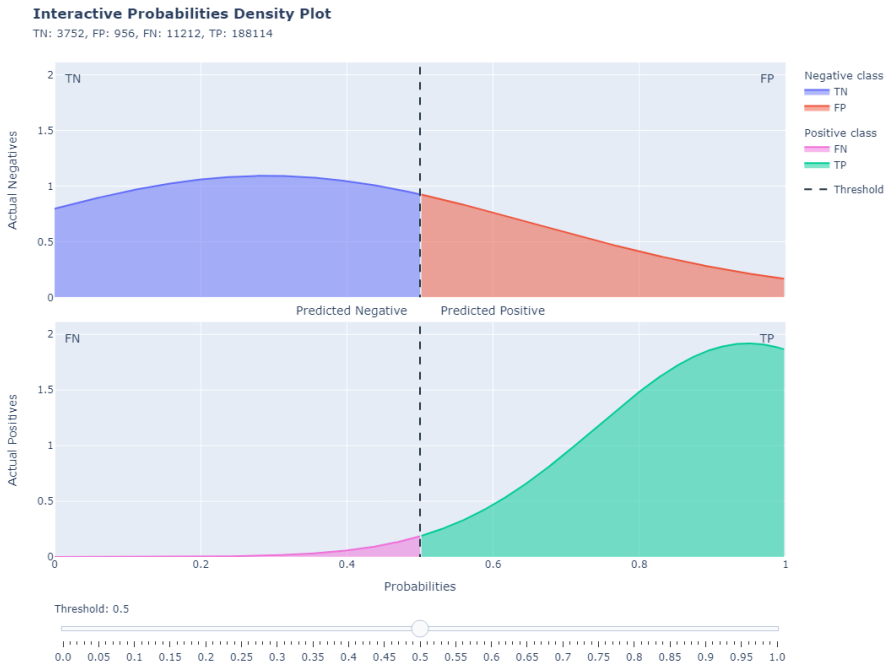


(a) *Distribucija vjerojatnosti modela stabla odluke*

(b) *Matrica zabune modela stabla odluke*

Slika 4.8 *Model stabla odluke - optimalni prag za ROC krivulju*

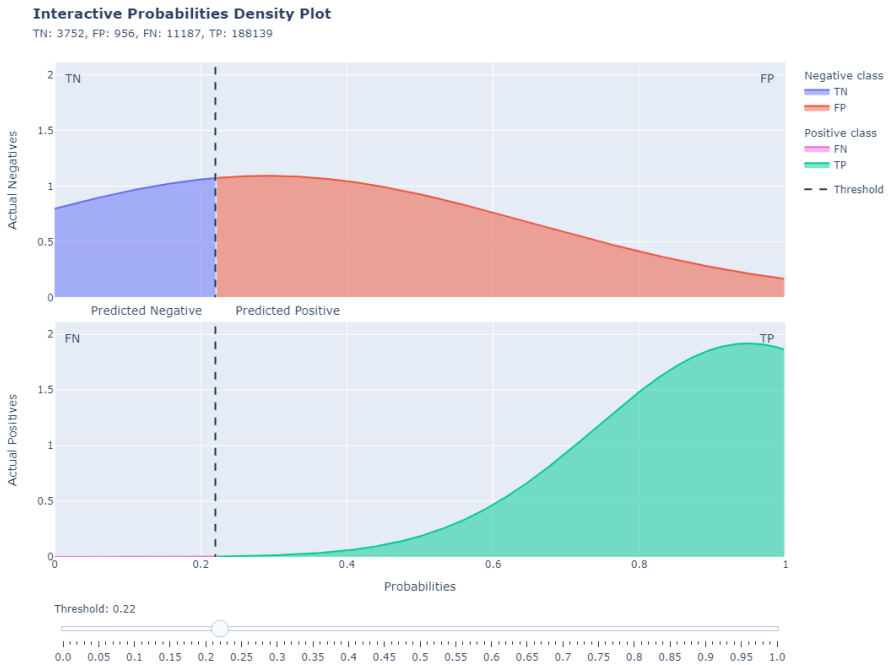
Za model treniran naivnim Bayesovim klasifikatorom, distribucija vjerojatnosti za zadani prag od 0.5 je vidljiva na slici 4.9a. Iz prikazane krivulje se može očitati znatno šira distribucija vjerojatnosti, što podrazumijeva potencijalno lošiji klasifikacijski model u usporedbi sa primjerice, modelom slučajne šume. Na temelju toga, na slici 4.9b se može uočiti niža vrijednost odziva koja iznosi oko 0.94, što implicira na veći broj FN-a. Međutim, uz nižu vrijednost odziva, može se uočiti visoka vrijednost preciznosti od 0.99, što implicira na niži broj FP-a. S druge strane, na slici 4.10a se može vidjeti distribucija vjerojatnosti s optimalnim pragom od 0.22. Usporedbom distribucije s optimalnim te sa zadanim pragom, može se uočiti povećanje broja TP-a. Međutim, ako se usporede vrijednosti klasifikacijskih metrika na slikama 4.9b i 4.10b može se uočiti da nema nikakvih značajnih promjena. Iz toga se može zaključiti da su za pragove od 0.5 i 0.22 vrijednosti FPR-a i TPR-a na ROC krivulji približno jednake.



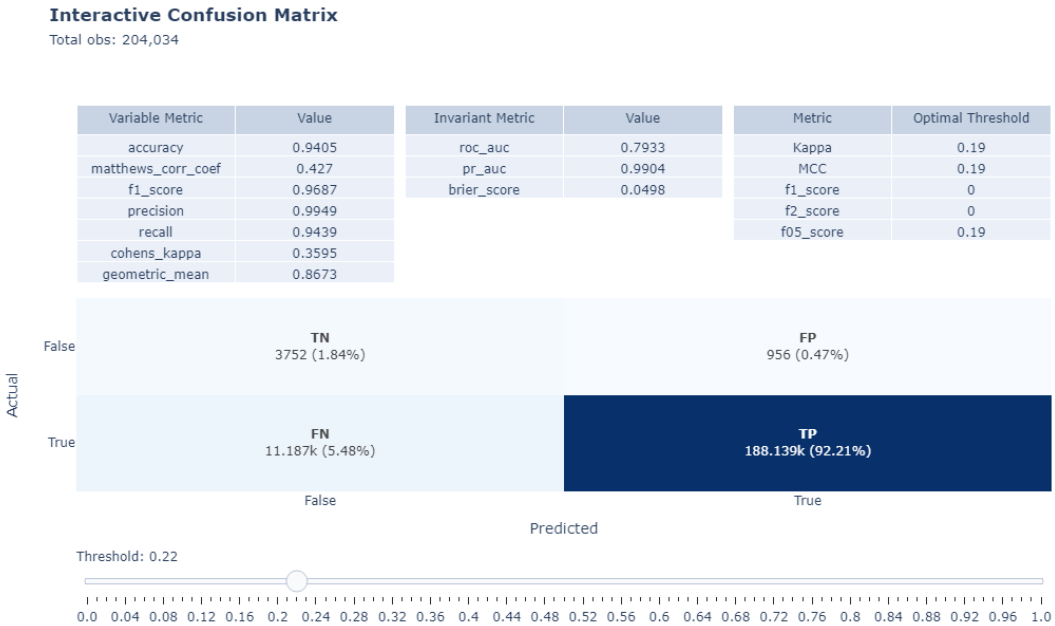
(a) *Distribucija vjerojatnosti naivnog Bayesovog modela*

(b) *Matrica zabune naivnog Bayesovog modela*

Slika 4.9 *Distribucija vjerojatnosti i matrica zabune za naivni Bayesov model*



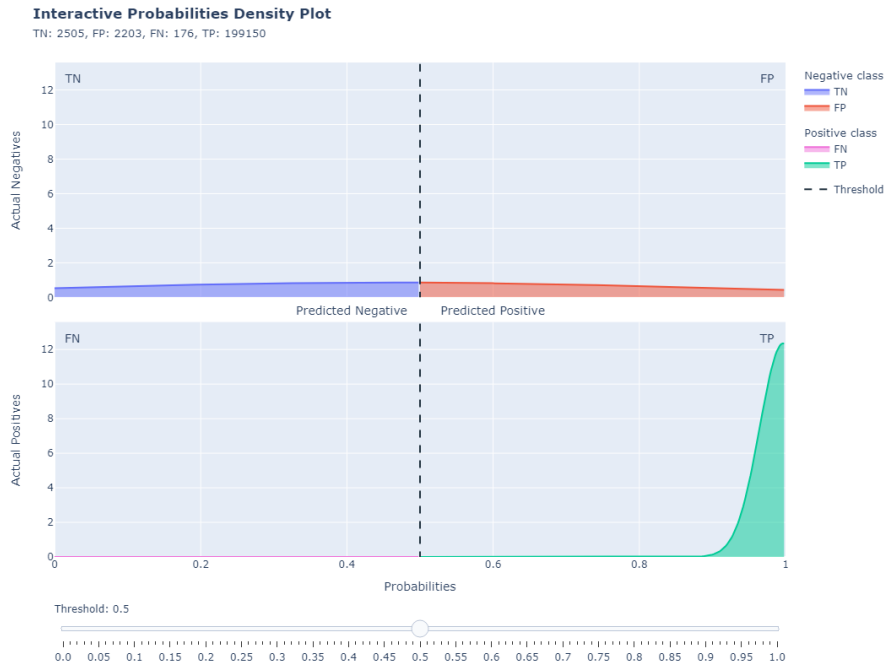
(a) *Distribucija vjerojatnosti naivnog Bayesovog modela*



(b) *Matrica zabune naivnog Bayesovog modela*

Slika 4.10 *Naivni Bayesov model - optimalni prag za ROC krivulju*

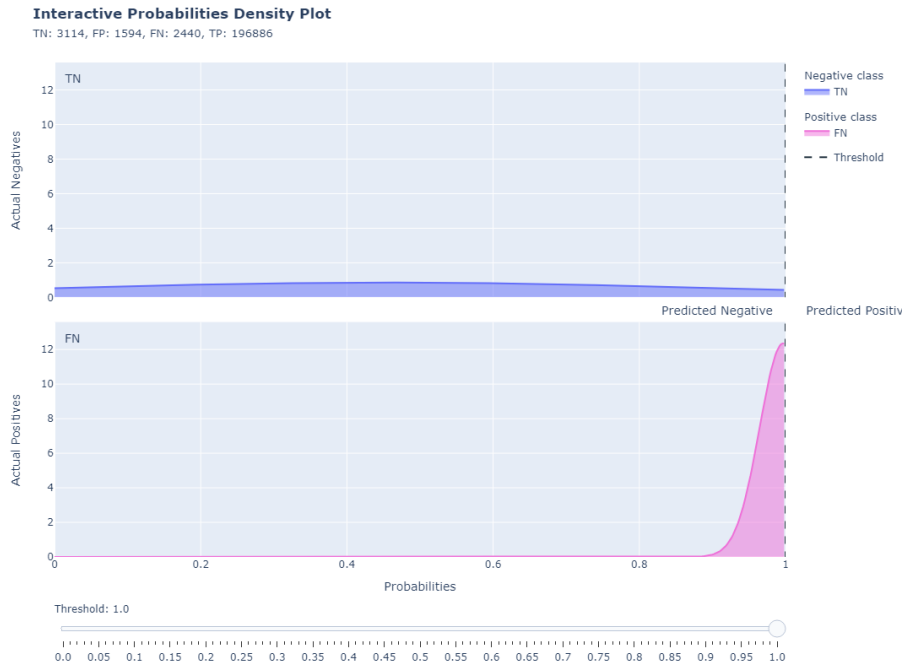
Za model treniran KNN algoritmom, na slici 4.11a je prikazana distribucija vjerojatnosti za koju se može reći da je dosta slična onima od modela stabla odluke (4.7a) i slučajne šume (4.5a). Takva sličnost u distribucijama vjerojatnosti, ujedno podrazumijeva i slične vrijednosti klasifikacijskih metrika, što je vidljivo na slikama: 4.11b (KNN), 4.7b (stablo odluke) i 4.5b (slučajna šuma). Na temelju iznimno dobrih klasifikacijskih vrijednosti, za KNN model bi se moglo reći da je čak najbolji od tri prethodno navedena modela. Međutim, prilikom računanja optimalnog praga za KNN model, dobivena je vrijednost 1.0. Na slici 4.12 su prikazane krivulja distribucije vjerojatnosti i matrica zabune za vrijednost praga od 1.0, pri čemu KNN model uzorke klasificira kao napadačke samo kad je potpuno siguran da se radi o napdačkom uzorku, dok za sve vjerojatnosti manje od 1.0 uzorke klasificira kao normalan promet. Ovakvi rezultati se mogu objasniti činjenicom da KNN algoritam donosi odluke na temelju najzastupljenije klase uzoraka koji okružuju neki ciljani uzorak. Na temelju toga i iznimno neujednačenog skupa podataka kod kojeg prevladavaju napadački uzorci, težnja KNN algoritma prema pragu od 1.0 nije u potpunosti neočekivana.



(a) *KNN distribucija vjerojatnosti*

(b) *KNN matrica zabune*

Slika 4.11 *Distribucija vjerojatnosti i matrica zabune za KNN model*



(a) *KNN distribucija vjerojatnosti*

(b) *KNN matrica zabune*

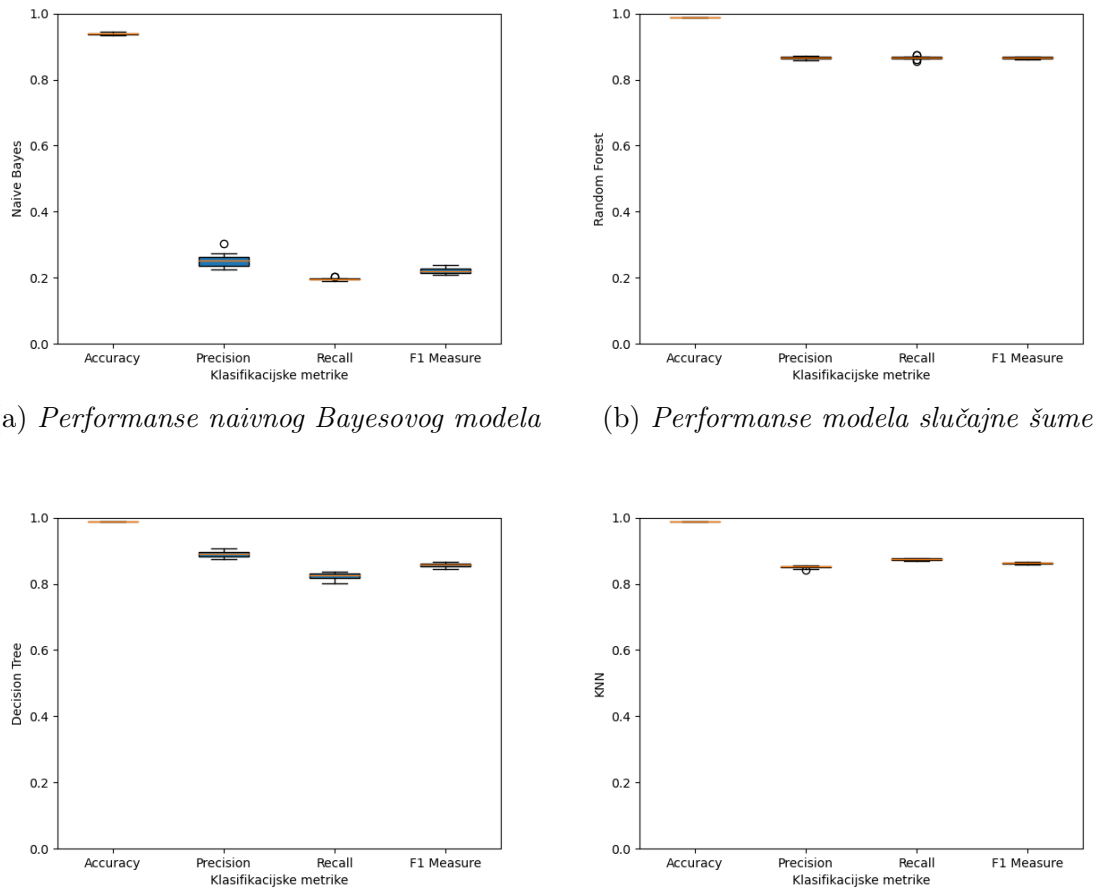
Slika 4.12 *KNN model - optimalni prag za ROC krivulju*

4.2 Procjena modela strojnog učenja - NF-UNSW-NB15

Uz proces treniranja i testiranja različitih modela strojnog učenja primjenom NF-BoT-IoT skupa podataka, u ovom poglavlju isti proces se primjenjuje nad NF-UNSW-NB15 skupom podataka. NF-UNSW-NB15, kao i NF-BoT-IoT spada pod skupove podataka NetFlowV1 formata te prema tome sadrži značajke prikazane u tablici 2.1. Bitno je naglasiti da pored NF-BoT-IoT skupa podataka, koji sadrži 97.69% napadačkih uzoraka i 2.31% uzoraka definiranih kao normalan promet, skup podataka NF-UNSW-NB15 sadrži 4.46% napadačkih uzoraka te 95.54% uzoraka definiranih kao normalan promet. Također, NF-UNSW-NB15 sadrži 1,623,118 uzoraka, znatno više od NF-BoT-IoT skupa podataka koji sadrži 600,100 uzoraka. Prema navedenim postocima raspodjele uzoraka napada i normalnog prometa, može se reći da navedeni skupovi podataka definiraju različite, odnosno suprotne okoline za treniranje i testiranje modela strojnog učenja. Upravo na temelju takve suprotnosti, uz dobivene klasifikacijske rezultate se mogu izvući određeni zaključci o uspješnosti različitih modela prilikom korištenja neujednačenih skupova podataka.

Primjenom prethodno navedenih algoritama strojnog učenja i unakrsne validacije u k-preklopa nad skupom podataka NF-UNSW-NB15, dobiju se vrijednosti klasifikacijskih metrika prikazanih na slici 4.13. Jedna metrika koja se iz svih prikazanih modela može izdvojiti je metrika točnosti. Iz prikazanih grafova različitih modela, može se vidjeti da je vrijednost točnosti dosta visoka (oko 0.99), što je vrlo vjerojatno posljedica neujednačenosti skupa podataka. Prije detaljnije analize rezultata prikazanih modela, potrebno je eliminirati model treniran naivnim Bayesovim klasifikatorom. Razlog za to su iznimno niske vrijednosti klasifikacijskih metrika prikazanih na slici 4.13a. Prikazani rezultati su savršen primjer utjecaja neujednačenog skupa podataka na uspješnost modela. Drugim riječima, može se vidjeti da metrika točnosti iznosi 0.99, a ostale klasifikacijske metrike imaju vrijednosti ispod 0.5, što znači da model klasificira uzorke iz skupa za testiranje na temelju nasumičnog pogađanja. Što se tiče modela slučajne šume, mogu se uočiti približno iste vrijednosti metrika preciznosti, odziva i F1-mjere, koje iznose oko 0.87. Za model treniran algoritmom stabla odluke (slika 4.13c), može se uočiti da vrijednost metrike F1-mjere iznosi oko

0.86, koja je izravna posljedica vrijednosti preciznosti (oko 0.90) i odziva (oko 0.84). S druge strane, F1-mjera modela treniranog KNN algoritmom (slika 4.13d) iznosi 0.87, pri čemu je u ovom slučaju vrijednost odziva (oko 0.88) veća od vrijednosti preciznosti (oko 0.86). Vrijednosti metrika odziva, preciznosti i F1-mjere modela slučajne šume, stabla odluke i KNN, mogu se procijeniti kao prilično visoke s obzirom da one procjenjuju sposobnost modela prilikom predviđanja napada, a korišteni skup podataka sadrži iznimno malen broj napadačkih uzoraka.



(a) *Performanse naivnog Bayesovog modela*

(b) *Performanse modela slučajne šume*

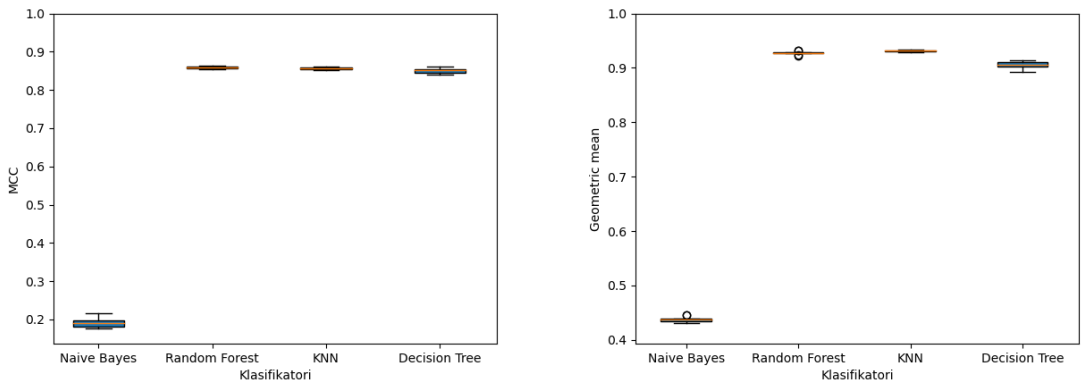
(c) *Performanse modela stabla odluke*

(d) *KNN performanse*

Slika 4.13 *Usporedba performansi algoritama*

Dobre performanse navedenih modela, uz izuzetak naivnog Bayesovog modela, prilikom klasificiranja i napada i normalnog prometa, dodatno potvrđuje visoka vri-

jednost geometrijske sredine, koja iznosi preko 0.9 (slika 4.14b). Također, za iste modele se može uočiti visoka vrijednost MCC-a (slika 4.14a), čime se uz geometrijsku sredinu dodatno potvrđuje uspješnost modela, koji su trenirani i testirani pomoću neujednačenog skupa podataka.

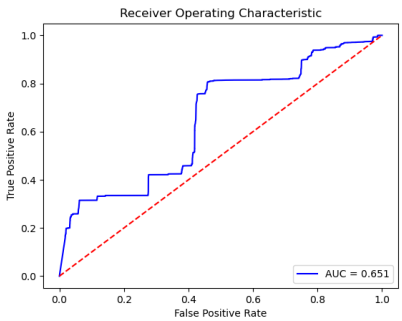


(a) *Matthewsov korelacijski koeficijent algoritama*

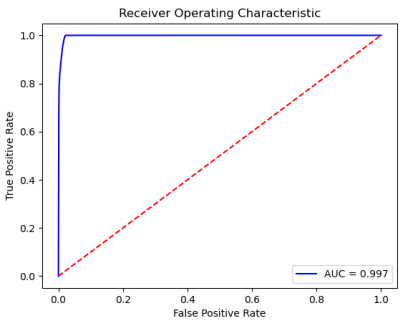
(b) *Geometrijska sredina algoritama*

Slika 4.14 *Usporedba MCC i G-Mean vrijednosti algoritama*

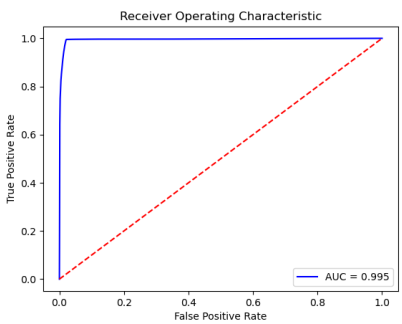
Nadovezivanjem na vrijednosti MCC-a i geometrijske sredine, na slici 4.15 se mogu uočiti ROC krivulje modela. Za prikazane ROC krivulje kod modela slučajne šume, stabla odluke i KNN, može se reći da imaju gotovo savršen oblik, što se dodatno očituje kod AUC vrijednosti, koje iznose između 0.99 i 1.0 (slika 4.15). S druge strane, kod naivnog Bayesovog modela ROC krivulja ima poprilično loš oblik (slika 4.15a), što se dodatno može očitati iz AUC vrijednosti koja iznosi 0.651.



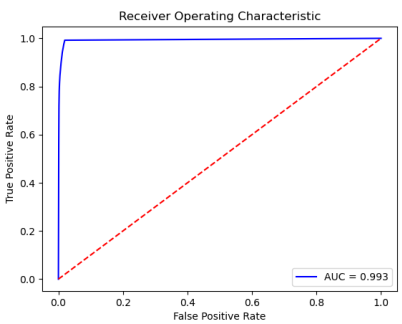
(a) *ROC-AUC naivnog Bayesovog*



(b) *ROC-AUC modela slučajne šume modela*



(c) *ROC-AUC modela stabla odluke*



(d) *KNN ROC-AUC*

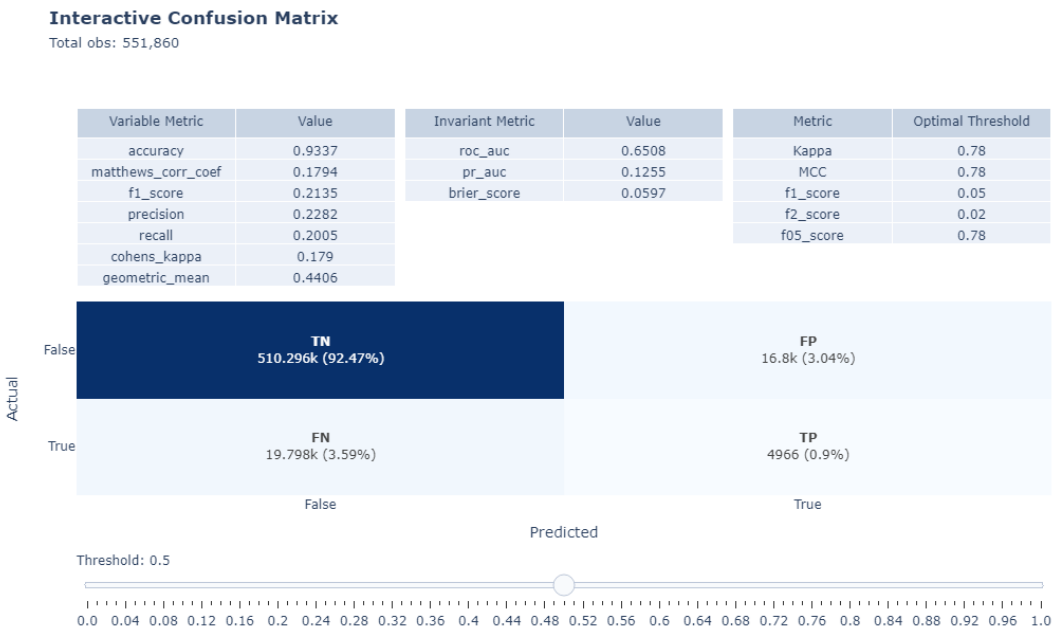
Slika 4.15 *Usporedba ROC-AUC krivulja algoritama*

4.2.1 Distribucija vjerojatnosti testnog skupa podataka

Uz pomoć ROC krivulje i geometrijske sredine modela, može se dodatno odrediti optimalni prag modela za uspješno i ujednačeno predviđanje uzoraka napada i normalnog prometa. Promjena praga modela se može prikazati pomoću distribucije vjerojatnosti te se utjecaj promjene praga dodatno može objasniti analizom promjene matrice zabune te klasifikacijskih metrika.

Za model treniran naivnim Bayesovim klasifikatorom, distribucija vjerojatnosti i matrica zabune za zadani prag od 0.5 su prikazani na slici 4.16. Iz distribucije vjerojatnosti se može odmah uočiti da su vjerojatnosti pozitivnih, odnosno napadačkih uzoraka dosta raspršene. Iz toga se dodatno može vidjeti da krivulja distribucije vjerojatnosti za pozitivne uzorke pada prema vrijednosti jedan, odnosno raste prema vrijednosti 0. Za pozitivne uzorke, poželjan je suprotan rast krivulje u odnosu na ovaj slučaj. Detaljnija interpretacija ovakve distribucije vjerojatnosti se može vidjeti u lošim rezultatima klasifikacijskih metrika prikazanim na slici 4.16b. Vrijednosti prikazanih klasifikacijskih metrika odgovaraju prethodno opisanim vrijednostima dobivenih unakrsnom validacijom u k-preklopa (slika 4.13a). Izračunati optimalni prag za naivni Bayesov model iznosi 0.01 te je distribucija vjerojatnosti za taj prag prikazana na slici 4.17a. Smanjenje vrijednosti praga se može gledati kao posljedica nastojanja povećanja vrijednosti geometrijske sredine te time povećanja uspješnosti klasifikacije i napada i normalnog prometa. Međutim, uz dosta loše generalne performanse modela, pomicanje praga u ovom slučaju ne donosi dodatna poboljšanja što je vidljivo na slici 4.17b.

(a) *Distribucija vjerojatnosti naivnog Bayesovog modela*



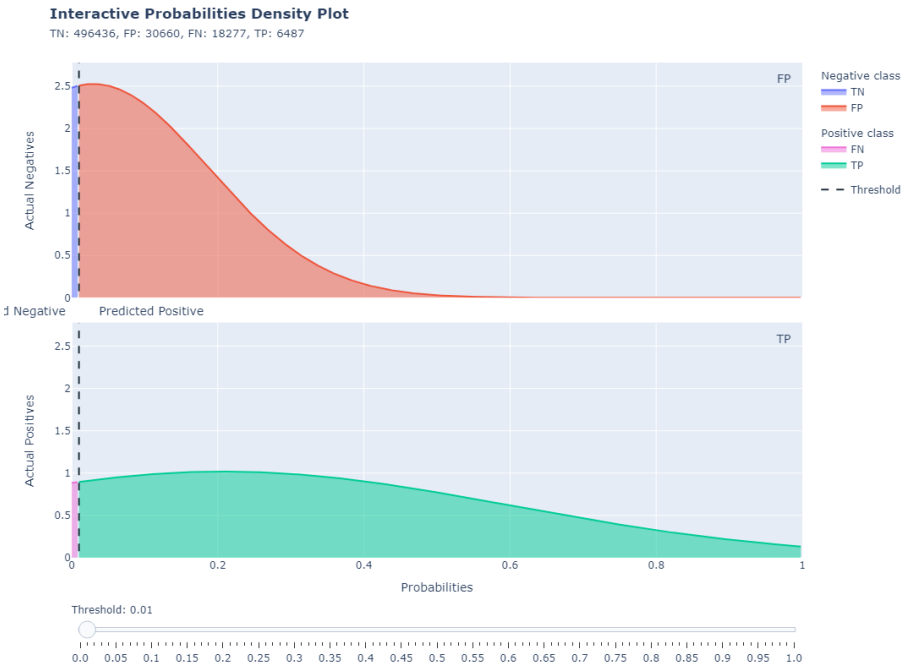
Variable Metric	Value
accuracy	0.9337
matthews_corr_coef	0.1794
f1_score	0.2135
precision	0.2282
recall	0.2005
cohens_kappa	0.179
geometric_mean	0.4406

Invariant Metric	Value
roc_auc	0.6508
pr_auc	0.1255
brier_score	0.0597

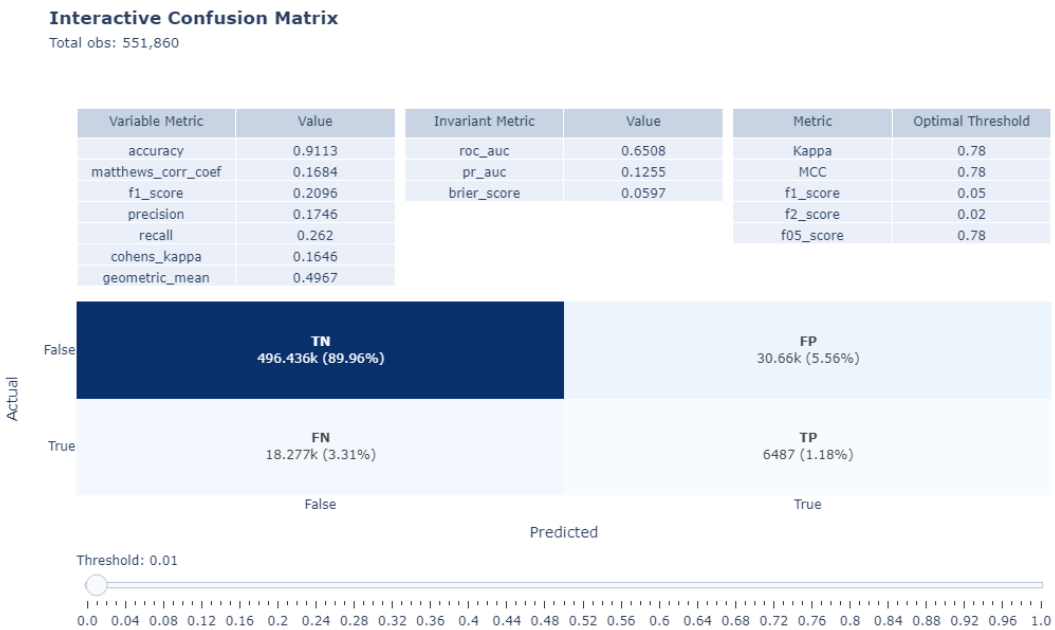
Metric	Optimal Threshold
Kappa	0.78
MCC	0.78
f1_score	0.05
f2_score	0.02
f05_score	0.78

(b) *Matrica zabune naivnog Bayesovog modela*

Slika 4.16 *Distribucija vjerojatnosti i matrica zabune za naivni Bayesov model*



(a) *Distribucija vjerojatnosti naivnog Bayesovog modela*

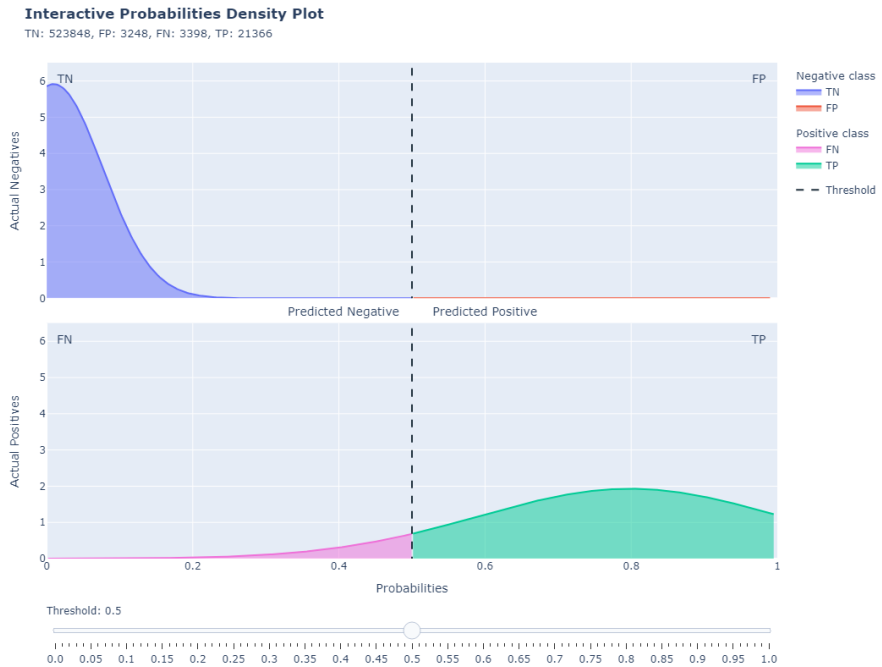


(b) *Matrica zabune naivnog Bayesovog modela*

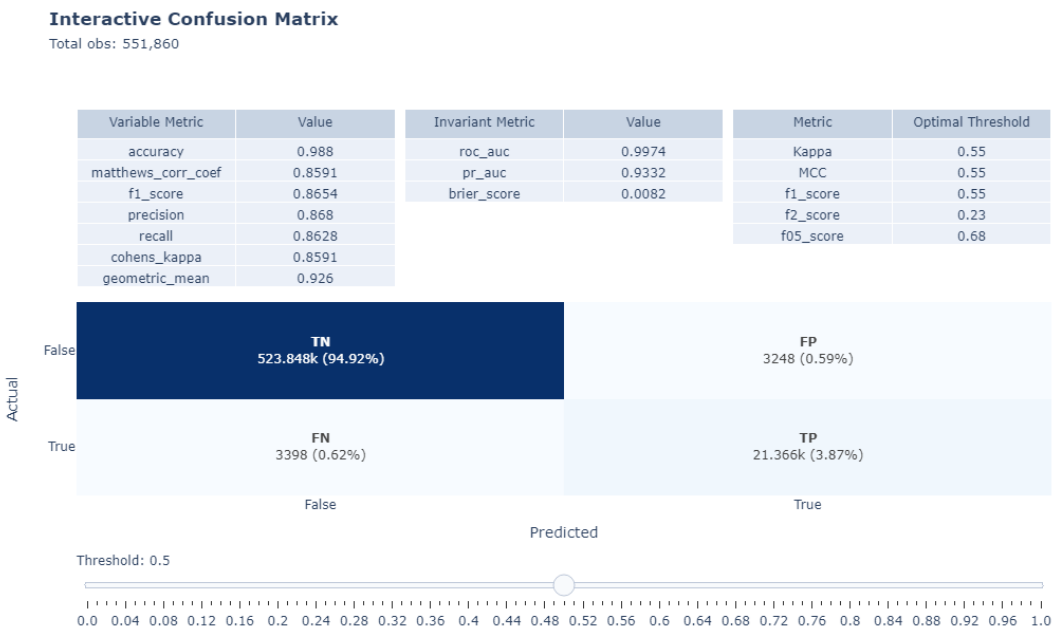
Slika 4.17 *Naivni Bayesov model - optimalni prag za ROC krivulju*

Za model treniran algoritmom slučajne šume, na slici 4.18a je prikazana distribucija vjerojatnosti, koja za razliku od distribucije vjerojatnosti modela treniranog naivnim Bayesovim klasifikatorom (slika 4.16a) pokazuje puno uspješniju klasifikaciju napada i normalnog prometa. Ta činjenica se može potvrditi pomoću matrice zabune i vrijednosti klasifikacijskih metrika prikazanih na slici 4.18b. Iz prikazanih metrika može se uočiti dosta visoka vrijednost geometrijske sredine (0.926). Međutim, ta vrijednost se može dodatno povećati s pomicanjem praga. Optimalni prag za model slučajne šume iznosi 0.17 te se s tim pragom vrijednost geometrijske sredine povećala na 0.9886 (slika 4.19b). Međutim, kao posljedica povećanja geometrijske sredine, smanjila se vrijednost preciznosti sa 0.868 na 0.691. To smanjenje preciznosti je direktna posljedica povećanja vrijednosti odziva sa 0.863 na 0.998, što je i očekivana pojava iz razloga što se geometrijska sredina računa pomoću vrijednosti True Positive Rate-a, odnosno odziva koji ima maksimalnu vrijednost za optimalni prag ROC krivulje. Kao posljedica pomicanja praga, na distribuciji vjerojatnosti se također može uočiti veći broj True Positive uzoraka. S jedne strane, pomicanje praga može biti korisno, primarno radi detekcije većeg broja napadačkih uzoraka kojih je u ovom skupu podataka znatno manje. Međutim, za postizanje toga, znatno se smanjuje preciznost, odnosno povećava se broj FP uzoraka.

Što se tiče modela treniranih algoritmima stabla odluke (slika 4.20a) i KNN (slika 4.22a), može se uočiti da su njihove distribucije vjerojatnosti gotovo identične s distribucijom vjerojatnosti modela slučajne šume (slika 4.18a). Dodatno, optimalni pragovi za algoritme stabla odluke (slika 4.21) i KNN (slika 4.23) iznose 0.06 i 0.11 respektivno, čije pomicanje kao i kod modela slučajne šume za posljedicu ima povećanje vrijednosti geometrijske sredine te povećanje odziva i smanjenje preciznosti. Sličnost performansi ta tri modela je također uočljiva iz njihovih ROC krivulja prikazanih na slici 4.15, na temelju čega se sličnost njihovih distribucija vjerojatnosti mogla i pretpostaviti.

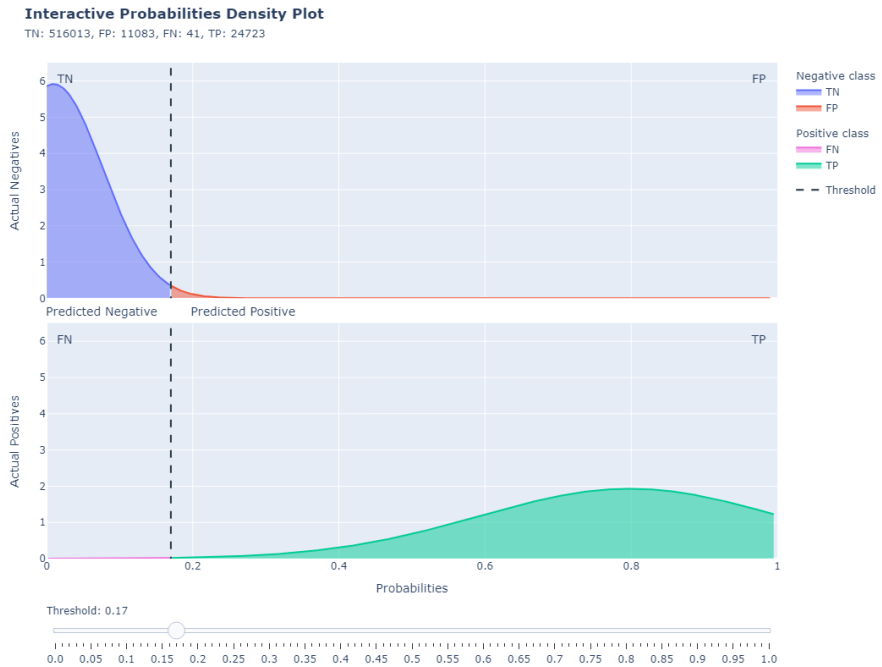


(a) *Distribucija vjerojatnosti modela slučajne šume*

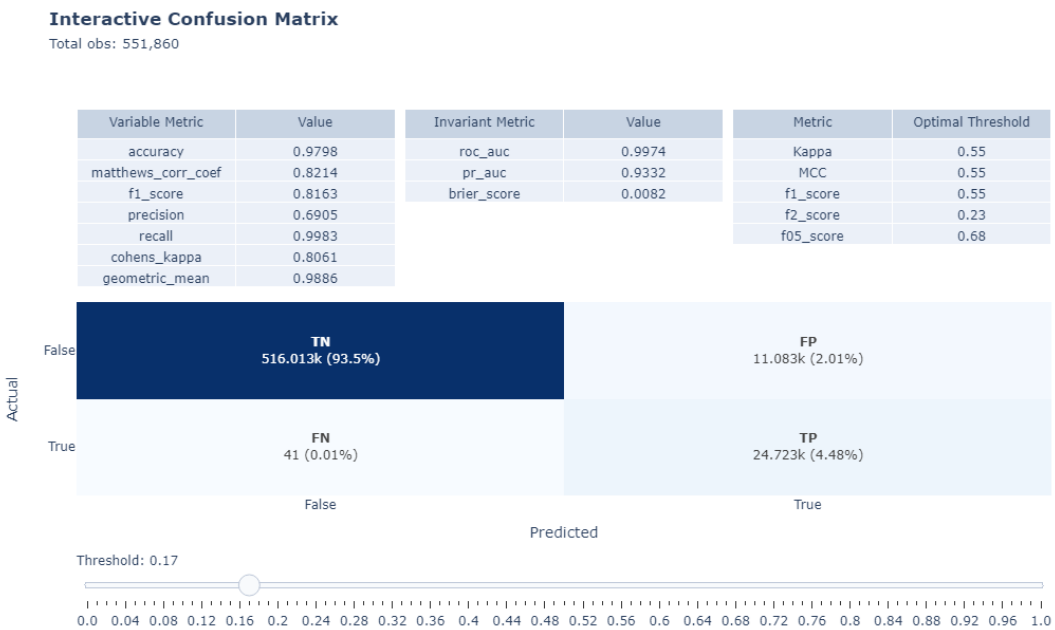


(b) *Matrica zabune modela slučajne šume*

Slika 4.18 *Distribucija vjerojatnosti i matrica zabune za model slučajne šume*

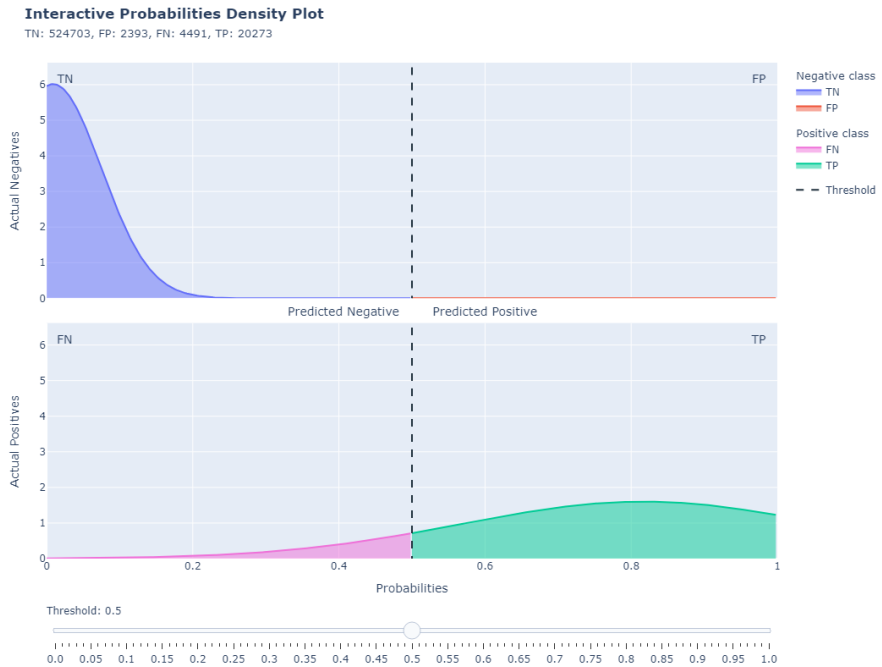


(a) *Distribucija vjerojatnosti modela slučajne šume*

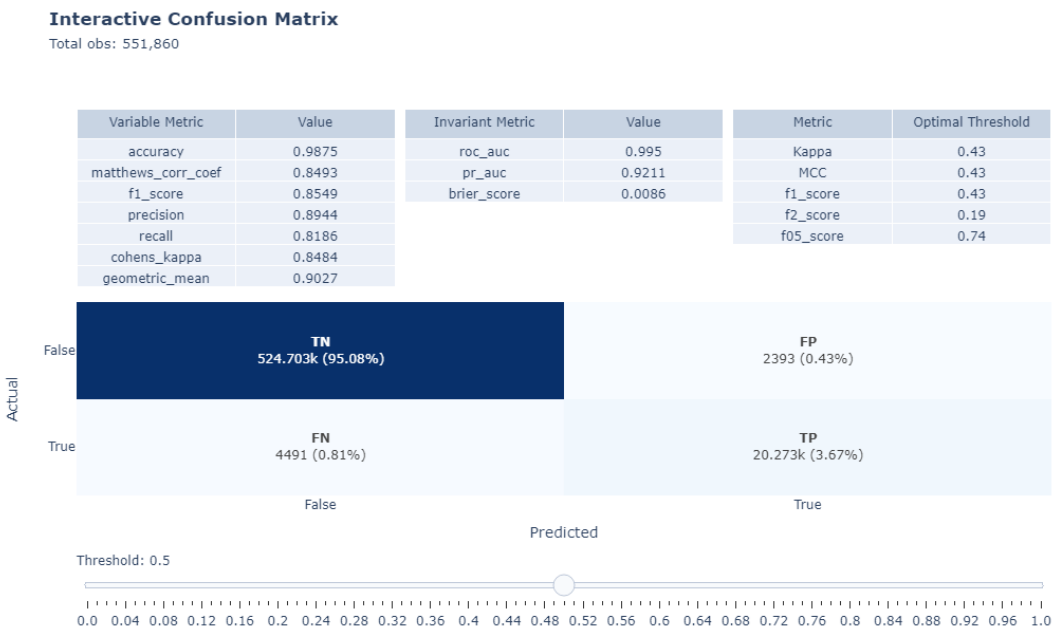


(b) *Matrica zabune modela slučajne šume*

Slika 4.19 *Model slučajne šume - optimalni prag za ROC krivulju*

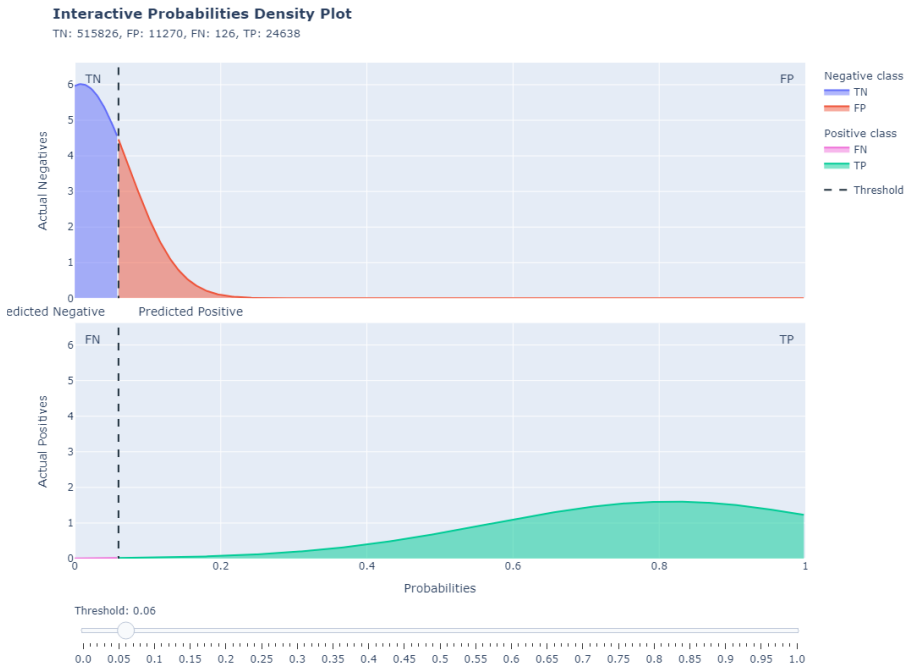


(a) *Distribucija vjerojatnosti modela stabla odluke*

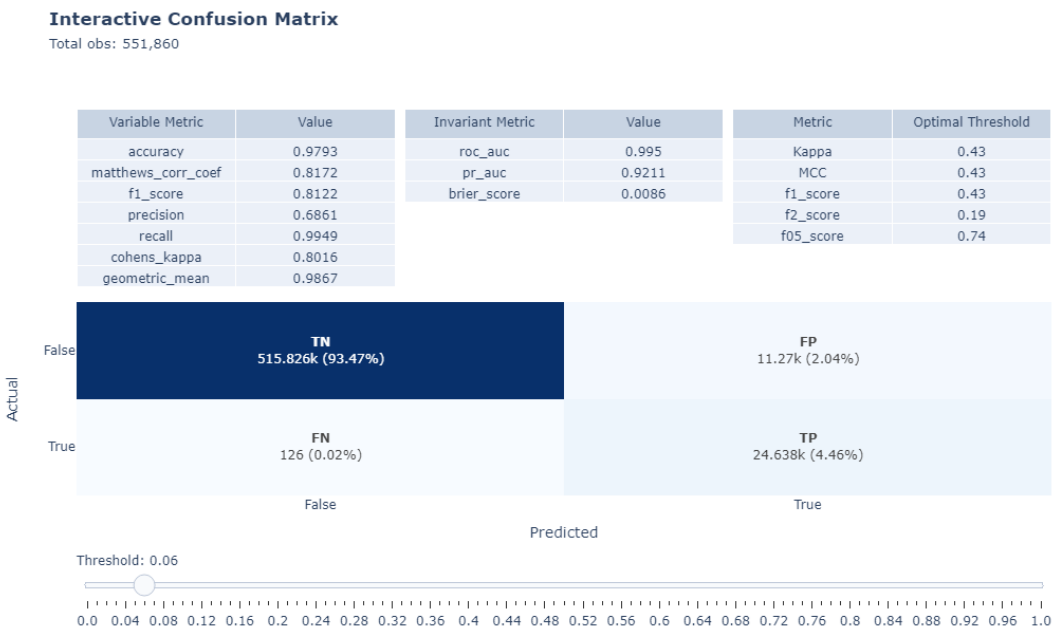


(b) *Matrica zabune modela stabla odluke*

Slika 4.20 *Distribucija vjerojatnosti i matrica zabune za model stabla odluke*

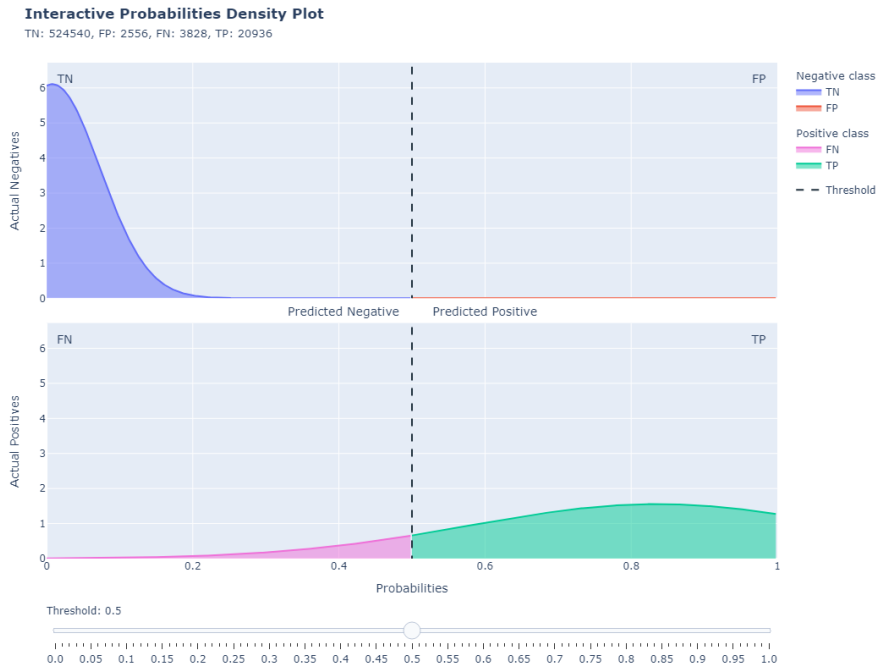


(a) *Distribucija vjerojatnosti modela stabla odluke*

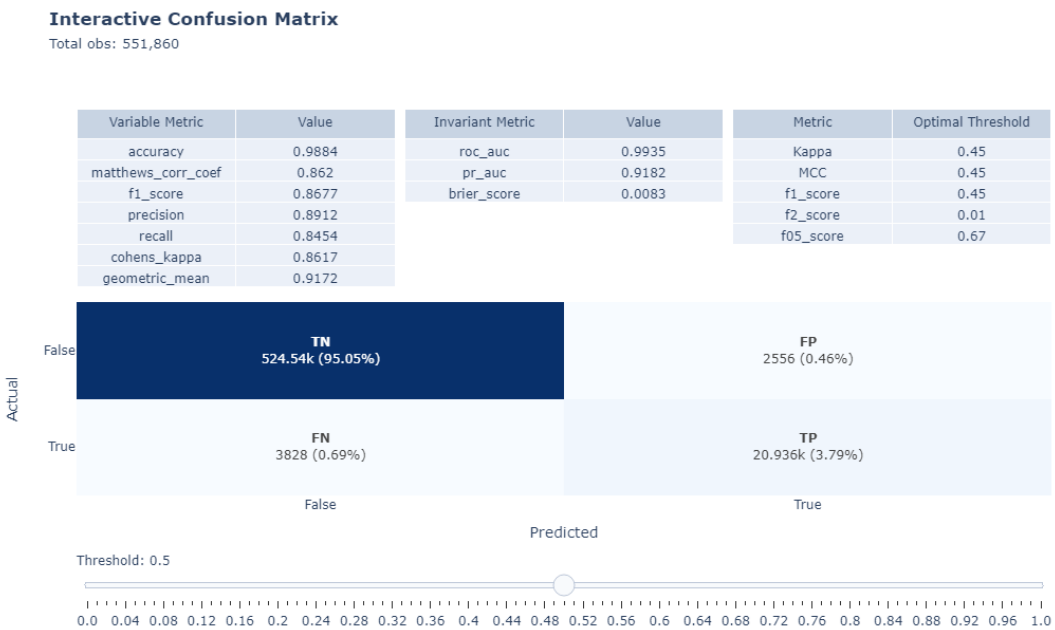


(b) *Matrica zabune modela stabla odluke*

Slika 4.21 *Model stabla odluke - optimalni prag za ROC krivulju*

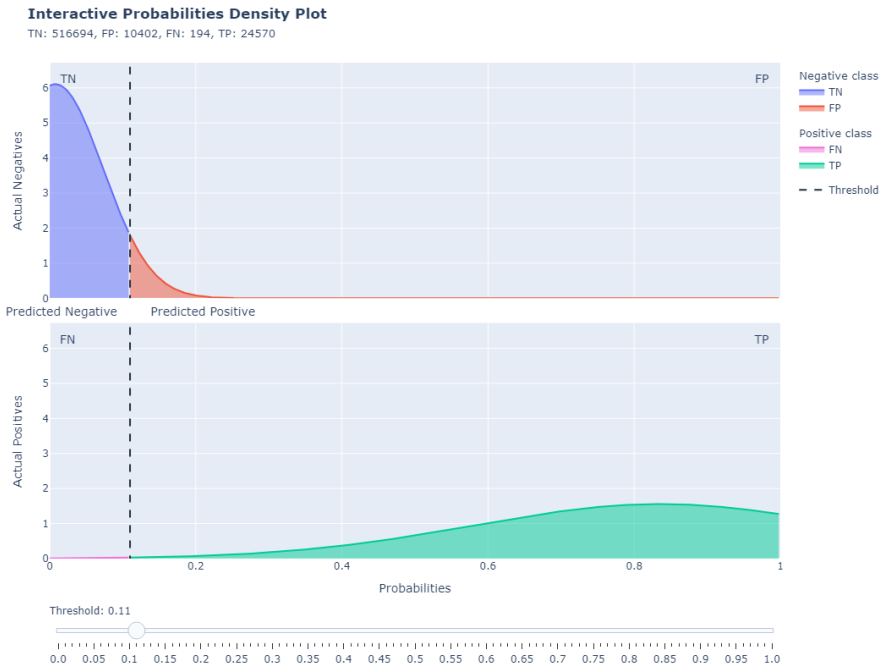


(a) *KNN distribucija vjerojatnosti*

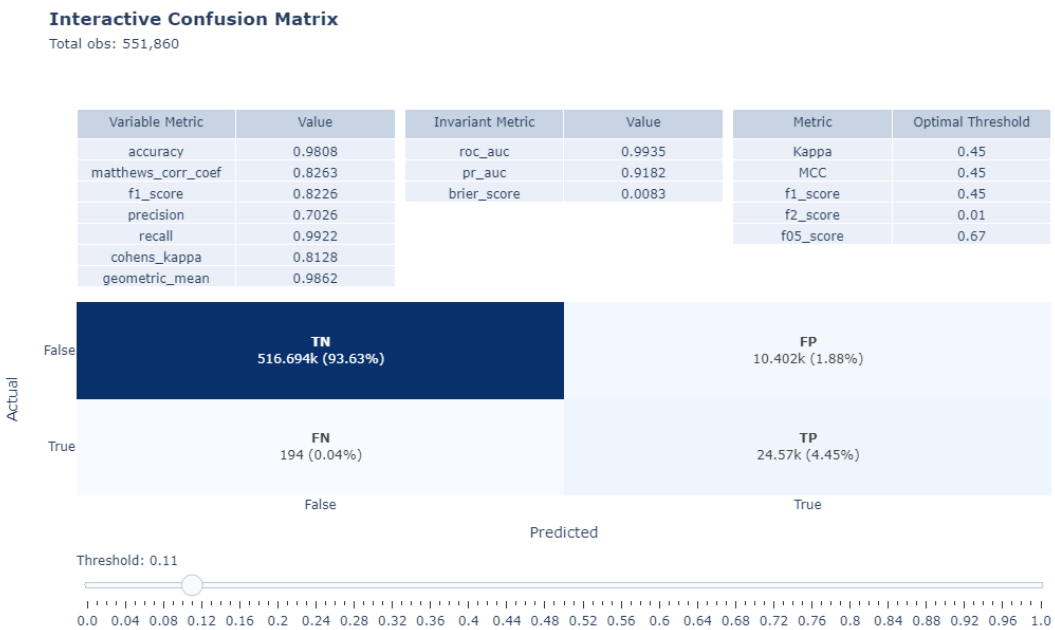


(b) *KNN matrica zabune*

Slika 4.22 *Distribucija vjerojatnosti i matrica zabune za KNN model*



(a) *KNN distribucija vjerojatnosti*



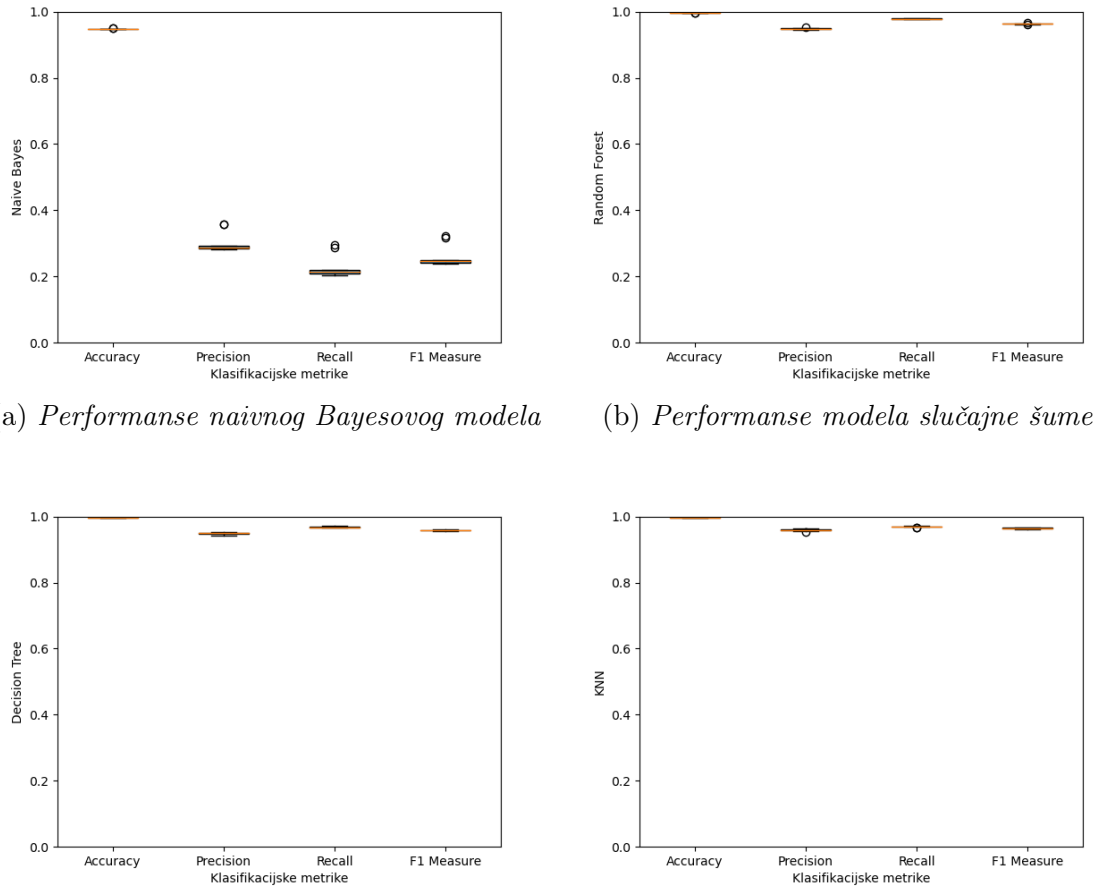
(b) *KNN matrica zabune*

Slika 4.23 *KNN model - optimalni prag za ROC krivulju*

4.3 Procjena modela strojnog učenja - NF-UNSW-NB15-V2

Nakon treniranja i testiranja modela strojnog učenja nad skupovima podataka sa 12 značajki, u ovom poglavlju isti proces je proveden nad skupom podataka koji sadrži dodatnu 31 značajku uz prethodnih 12. Navedene značajke su prikazane na slici 2.1. S obzirom da su se kod prethodnih skupova podataka izbacile značajke identifikatora toka te iste značajke su izbačene i iz novog skupa podataka. Također, pored značajki identifikatora toka izbačene su značajke koje se odnose na *Time To Live (TTL)* mjeru zbog ekstremne korelacije za oznakama *(engl. Labels)* uzoraka (navedeno u izvoru [1]). Također, slično kao i kod NF-UNSW-NB15, ovaj skup podataka sadrži 3.98% uzoraka određenih kao napad te 96.02% uzoraka određenih kao normalan promet.

Provođenjem procesa treniranja i testiranja različitih modela strojnog učenja, dobiveni su rezultati prikazani na slici 4.24. Iz prikazanih rezultata može se uočiti velika sličnost sa rezultatima dobivenim pomoću NF-UNSW-NB15 skupa podataka (slika 4.13). Prva sličnost je vidljiva u lošim rezultatima naivnog Bayesovog modela, nastalih kao posljedica utjecaja neujednačenog skupa podataka. Utjecaj neujednačenog skupa podataka je vidljiv iz loših rezultata metrika preciznosti, odziva i F1-mjere te visoke vrijednosti metrike točnosti. Druga sličnost se može uočiti usporedbom rezultata modela treniranih algoritmima slučajne šume, stabla odluke i KNN. Može se reći da su se performanse tih algoritama samo poboljšale s dodavanjem novih značajki te povećanjem broja uzoraka. To se može potvrditi usporedbom metrike F1-mjere, koja je prilikom treniranja navedenih modela nad NF-UNSW-NB15 skupom podataka iznosila oko 0.87, dok treniranjem nad NF-UNSW-NB15-V2 skupom podataka iznosi oko 0.96.



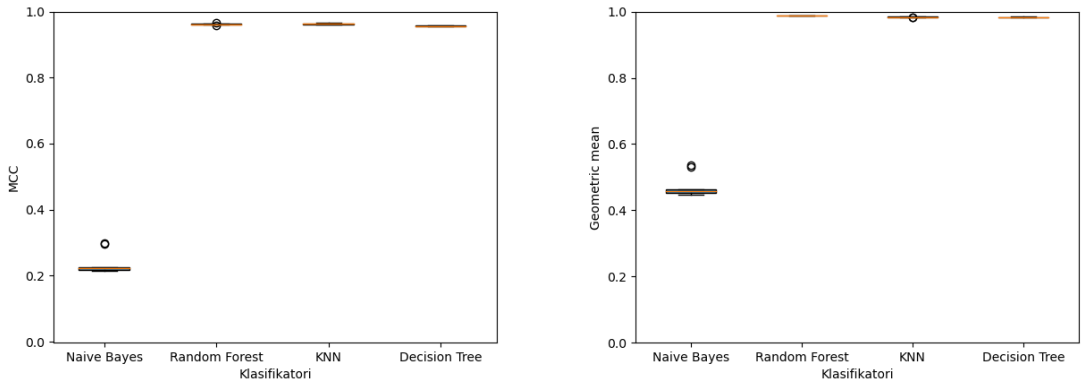
(a) *Performanse naivnog Bayesovog modela* (b) *Performanse modela slučajne šume*

(c) *Performanse modela stabla odluke* (d) *KNN performanse*

Slika 4.24 *Usporedba performansi algoritama*

Pored prethodno analiziranih klasifikacijskih metrika, na slici 4.25 mogu se uočiti vrijednosti metrika geometrijske sredine i MCC-a, koje za modele trenirane algoritmima slučajne šume, stabla odluke i KNN iznose oko 0.98 i 0.96 respektivno. Prema tome, u usporedbi sa metrikama geometrijske sredine i MCC-a dobivenih pomoću NF-UNSW-NB15 skupa podataka (slika 4.14), može se reći da se uspješnost navedenih klasifikatora u razlikovanju i napada i normalnog prometa samo poboljšala.

Što se tiče ROC krivulja (slika 4.26), u usporedbi sa NF-UNSW-NB15 skupom podataka (slika 4.15), jedina veća promjena je uočljiva kod naivnog Bayesovog modela

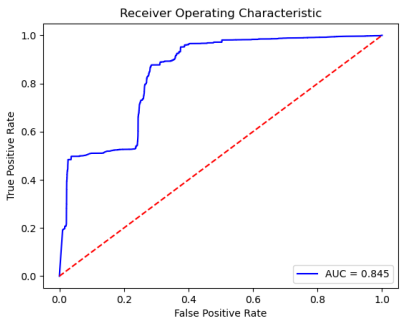


(a) *Matthewsov korelacijski koeficijent algoritama*

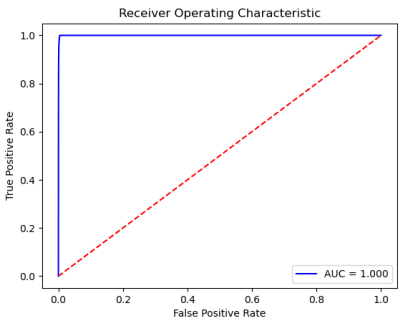
(b) *Geometrijska sredina algoritama*

Slika 4.25 *Usporedba MCC i G-Mean vrijednosti algoritama*

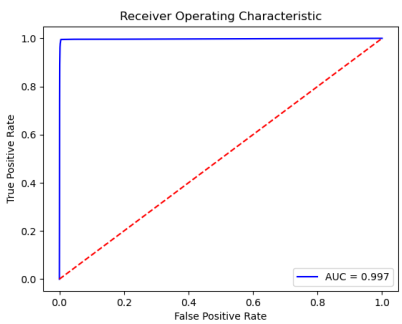
kod kojeg se oblik krivulje znatno poboljšao. Također, AUC vrijednost naivnog Bayesovog modela se povećala sa 0.651 (slika 4.15a) na 0.845 (slika 4.26a). Međutim, u usporedbi sa ostalim modelima, naivni Bayesov model nije mjerodavan. Drugim riječima, AUC vrijednosti modela slučajne šume, stabla odluke i KNN su uz malo veću vrijednost od NF-UNSW-NB15 modela, približno jednake vrijednosti 1.0. Kao posljedica povećanja AUC vrijednosti NF-UNSW-NB15-V2 modela, može se uočiti savršena ROC krivulja modela treniranog algoritmom slučajne šume (slika 4.26b).



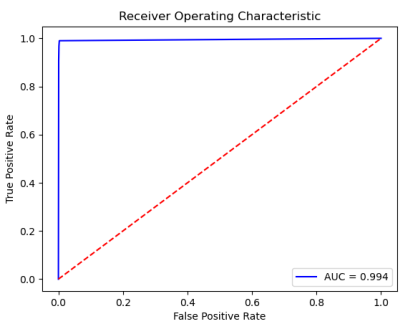
(a) *ROC-AUC naivnog Bayesovog*



(b) *ROC-AUC modela slučajne šume modela*



(c) *ROC-AUC modela stabla odluke*



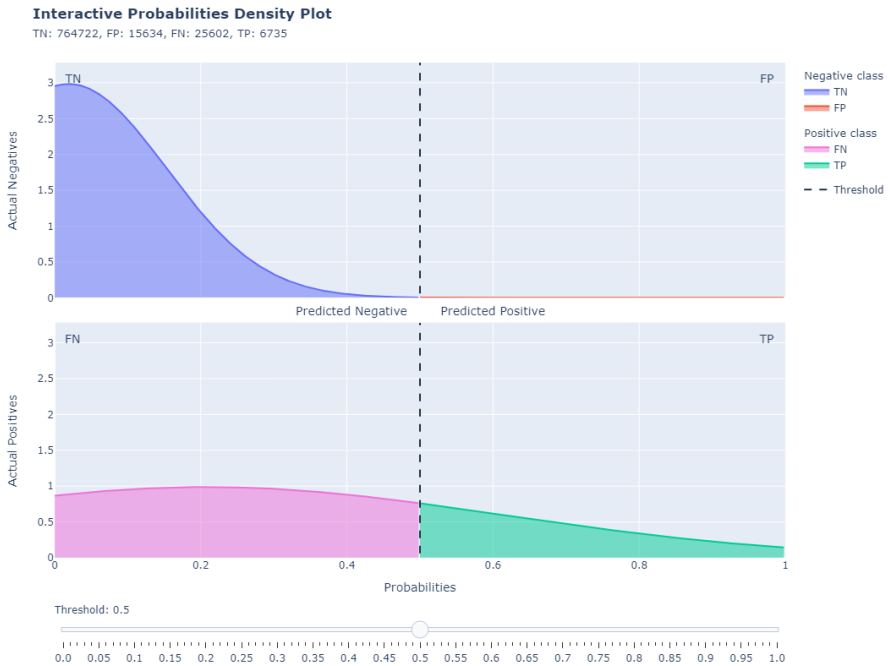
(d) *KNN ROC-AUC*

Slika 4.26 *Usporedba ROC-AUC krivulja algoritama*

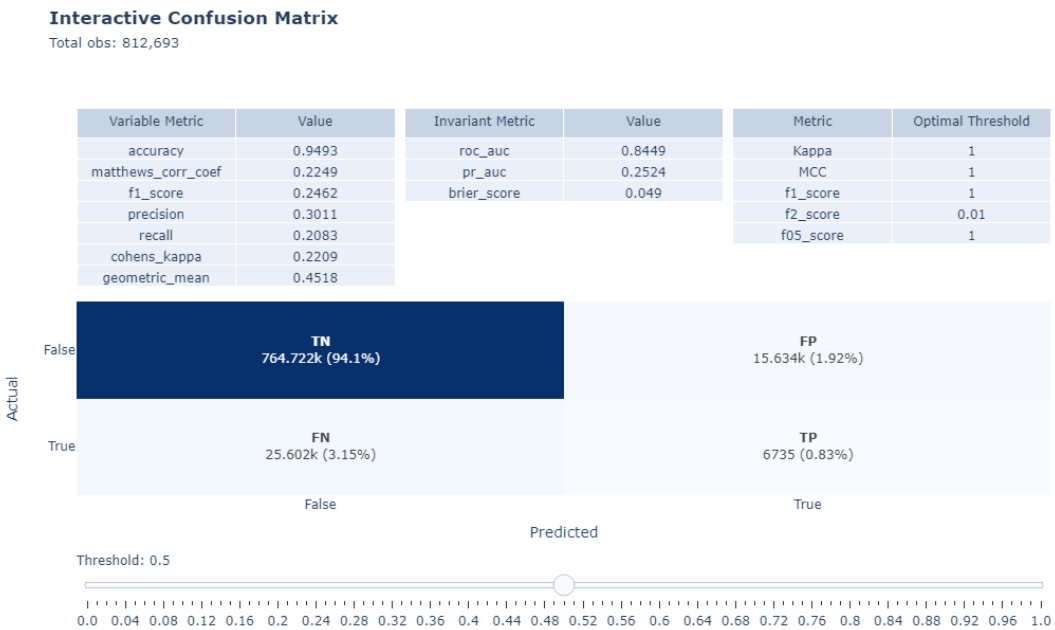
4.3.1 Distribucija vjerojatnosti testnog skupa podataka

Nakon prethodne analize vrijednosti klasifikacijskih metrika, dodatan uvid u performanse treniranih modela je moguće ostvariti povezivanjem njihove distribucije vjerojatnosti i matrice zabune.

Što se tiče naivnog Bayesovog modela, na slici 4.27 se mogu vidjeti distribucija vjerojatnosti te odgovarajuća matrica zabune. Iz distribucije vjerojatnosti (slika 4.27a) se može uočiti veći broj uzoraka klasificiranih kao FN, što ukazuje na lošiju uspješnost klasifikacije napadačkih uzoraka. Kao posljedica toga, na slici 4.27b se mogu vidjeti niske vrijednosti metrika preciznosti (0.301), odziva (0.208) i F1-mjere (0.246). Prema tome, radi neujednačene uspješnosti detekcije napada i normalnog prometa, vrijednost geometrijske sredine je također dosta niska (0.452). U svrhu poboljšanja uspješnosti detekcije napadačkih uzoraka te istovremenim povećanjem vrijednosti geometrijske sredine, potrebno je pomaknuti prag klasifikacije uzoraka. Optimalan prag za povećanje vrijednosti geometrijske sredine za naivni Bayesov model iznosi 0.01 (slika 4.28). Iz toga se može zaključiti da vrijednost optimalnog praga teži prema nuli, što uz neznatno povećanje vrijednosti geometrijske sredine sa 0.452 na 0.458, ukazuje na iznimno loše performanse modela.



(a) *Distribucija vjerojatnosti naivnog Bayesovog modela*



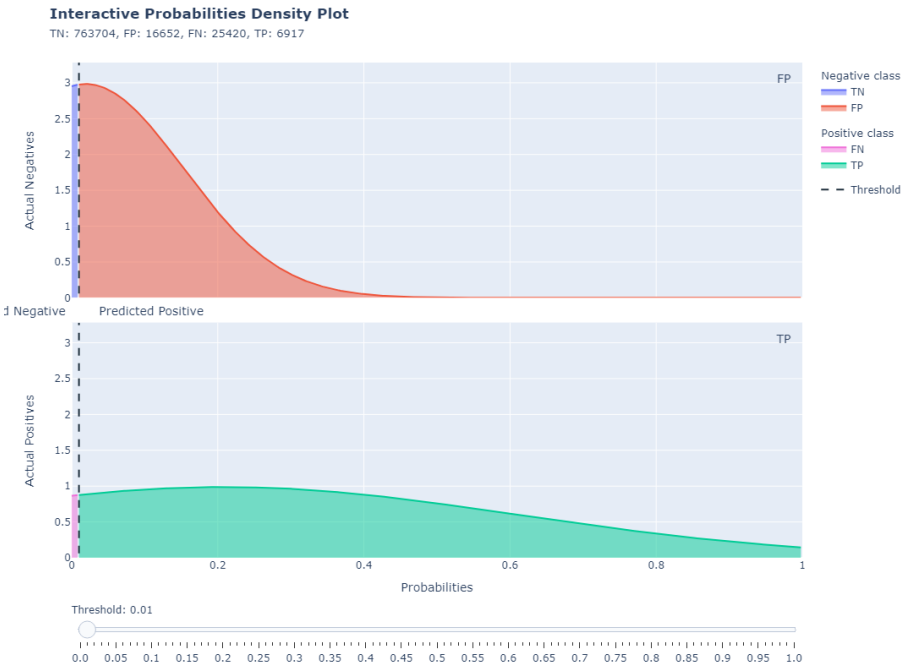
Variable Metric	Value
accuracy	0.9493
matthews_corr_coef	0.2249
f1_score	0.2462
precision	0.3011
recall	0.2083
cohens_kappa	0.2209
geometric_mean	0.4518

Invariant Metric	Value
roc_auc	0.8449
pr_auc	0.2524
brier_score	0.049

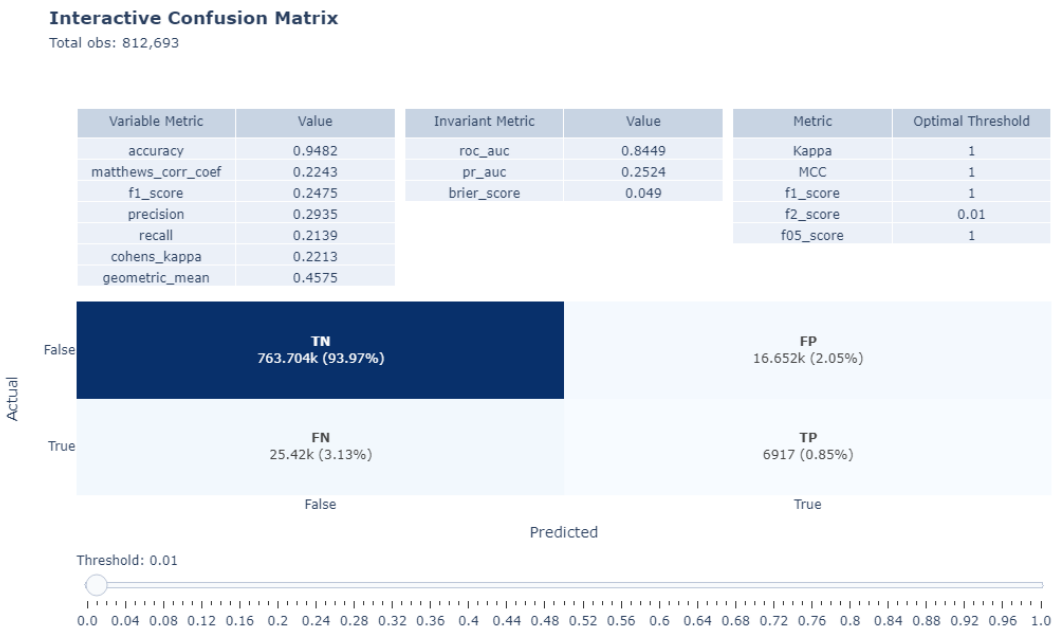
Metric	Optimal Threshold
Kappa	1
MCC	1
f1_score	1
f2_score	0.01
f05_score	1

(b) *Matrica zabune naivnog Bayesovog modela*

Slika 4.27 *Distribucija vjerojatnosti i matrica zabune za naivni Bayesov model*



(a) *Distribucija vjerojatnosti naivnog Bayesovog modela*

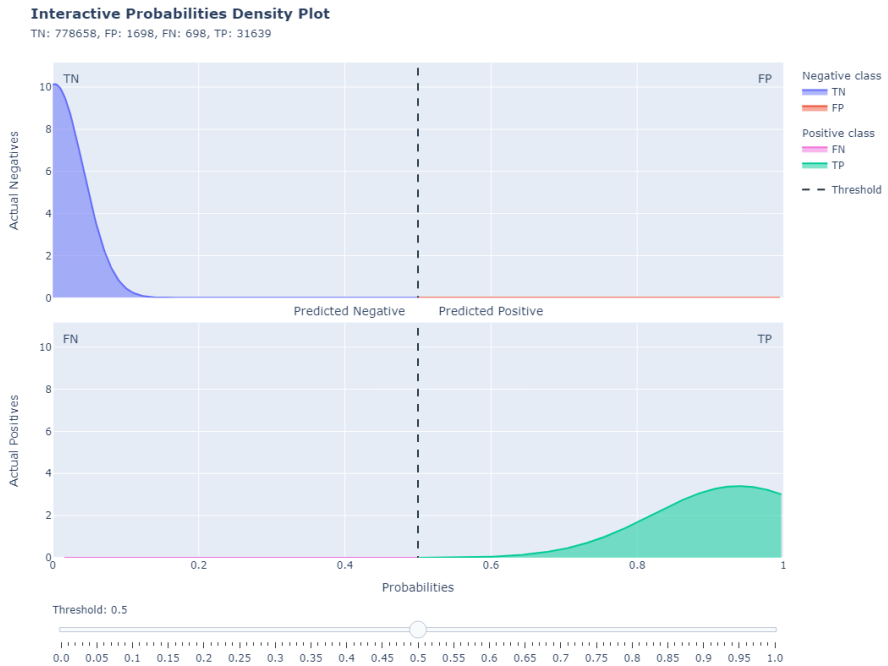


(b) *Matrica zabune naivnog Bayesovog modela*

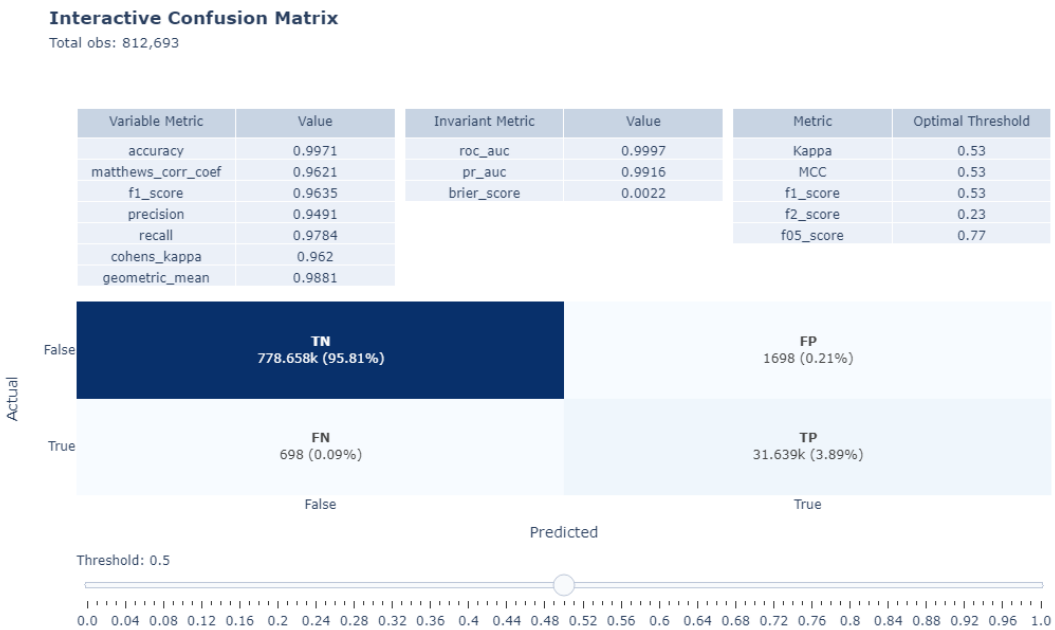
Slika 4.28 *Naivni Bayesov model - optimalni prag za ROC krivulju*

Na temelju prethodno prikazanih ROC krivulja (slika 4.26) te analiziranih vrijednosti klasifikacijskih metrika za modele trenirane algoritmima slučajne šume, stabla odluke i KNN, može se reći da ti modeli imaju približno iste performanse. Ovu izjavu mogu potvrditi distribucije vjerojatnosti tih modela prikazane na slikama 4.29a (slučajna šuma), 4.31a (stablo odluke) i 4.33a (KNN). Međutim, pored distribucija vjerojatnosti, u ovom slučaju potrebno je pobliže sagledati matrice zabune navedenih modela kako bi se uočile određene razlike u performansama.

Za model treniran algoritmom slučajne šume na slici 4.29b, može se uočiti broj uzoraka klasificiranih kao FN (698) te broj uzoraka klasificiranih kao FP (1698). Pored toga, potrebno je također obratiti pozornost na vrijednosti odziva (0.978) i preciznosti (0.949). Navedene vrijednosti vrijede za postavljeni prag od 0.5. Nakon pomicanja praga na 0.08 u svrhu povećanja vrijednosti geometrijske sredine, na slici 4.30b mogu se uočiti promjene u prethodno navedenim metrikama. Drugim riječima, broj FN uzoraka je snižen na 4, dok se broj FP uzoraka povećao na 3533. Iz ovakvih klasifikacijskih rezultata se može zaključiti da je model slučajne šume, uz prag optimalan za ROC krivulju, uspio točno klasificirati gotovo sve napadačke uzorke, što ukazuje na iznimno dobre performanse s obzirom na odnos broja napadačkih uzoraka i uzoraka normalnog prometa u skupu podataka. Ovakav zaključak potvrđuje i veća vrijednost odziva (0.999) u odnosu na preciznost (0.902), što ukazuje na veći broj uzoraka klasificiranih kao napad. Što se tiče modela stabla odluke i KNN, situacija je slična. Prije pomicanja praga, matrica zabune modela stabla odluke sadrži 1075 FN uzoraka te 1606 FP uzoraka (slika 4.31b). Nakon pomicanja praga na 0.04, broj FN uzoraka je pao na 164, dok se broj FP uzoraka povećao na 3501 (slika 4.32b). Također, za KNN model, matrica zabune sadrži 1042 FN te 1367 FP uzoraka prije promjene praga (slika 4.33b), dok su se nakon promjene praga na 0.33 te vrijednosti promijenile na 323 i 2300 respektivno (slika 4.34b). Na temelju rezultata za navedene modele, može se reći da je po pitanju detekcije što većeg broja napada, najbolji model slučajne šume s optimalnim pragom. Međutim, po pitanju minimiziranja broja FP uzoraka te održavanja relativno dobre uspješnosti detekcije napada, najbolji model bi bio onaj treniran KNN algoritmom uz optimalni prag.

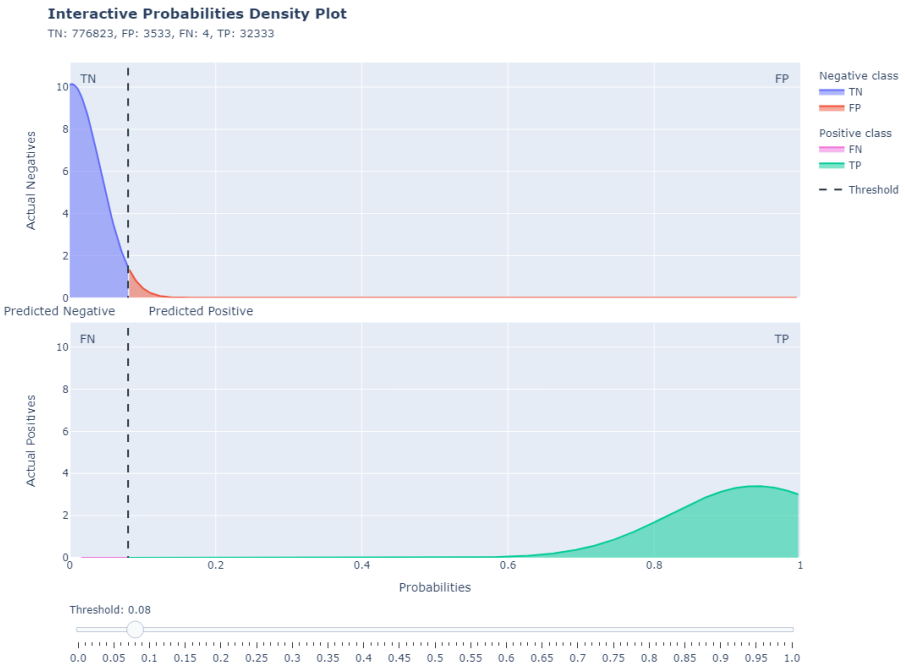


(a) *Distribucija vjerojatnosti modela slučajne šume*

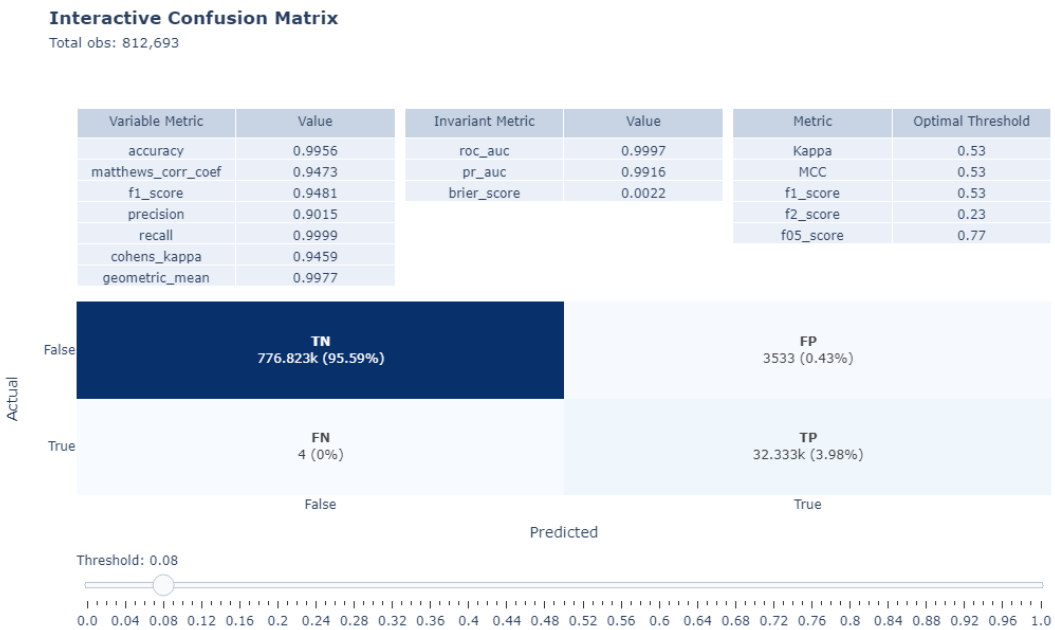


(b) *Matrica zabune modela slučajne šume*

Slika 4.29 *Distribucija vjerojatnosti i matrica zabune za model slučajne šume*

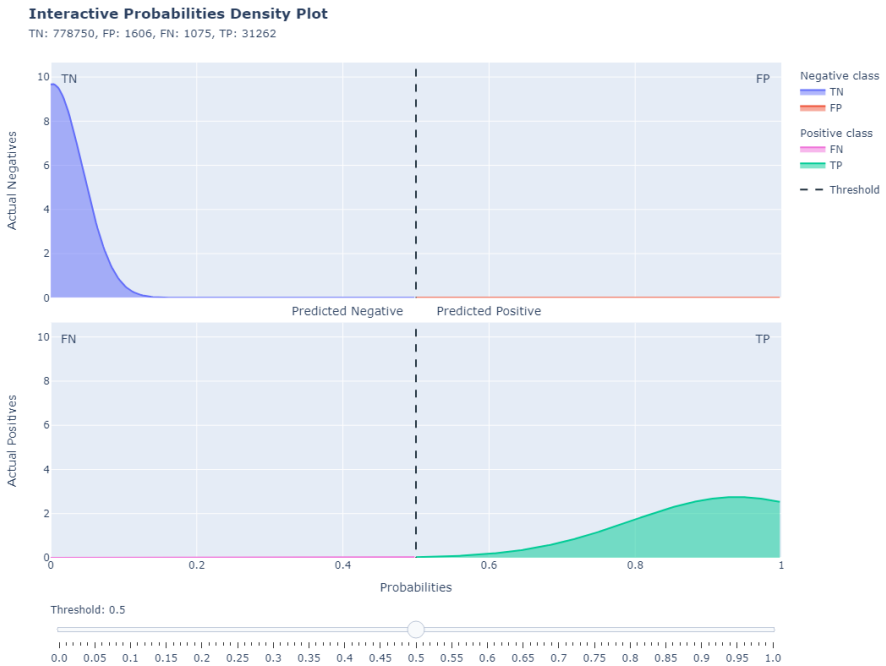


(a) *Distribucija vjerojatnosti modela slučajne šume*

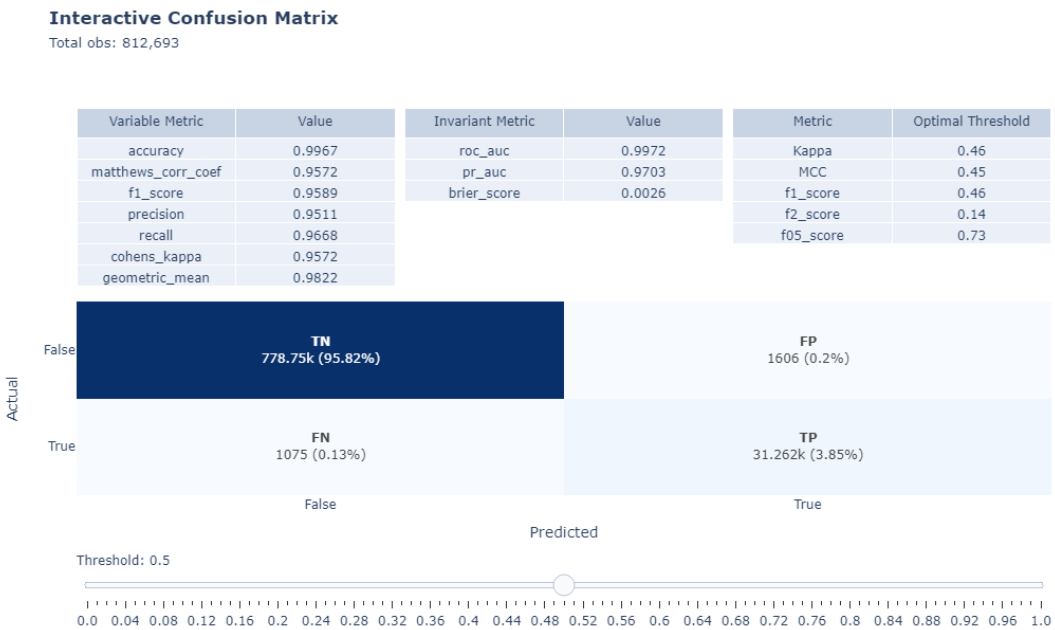


(b) *Matrica zabune modela slučajne šume*

Slika 4.30 *Model slučajne šume - optimalni prag za ROC krivulju*

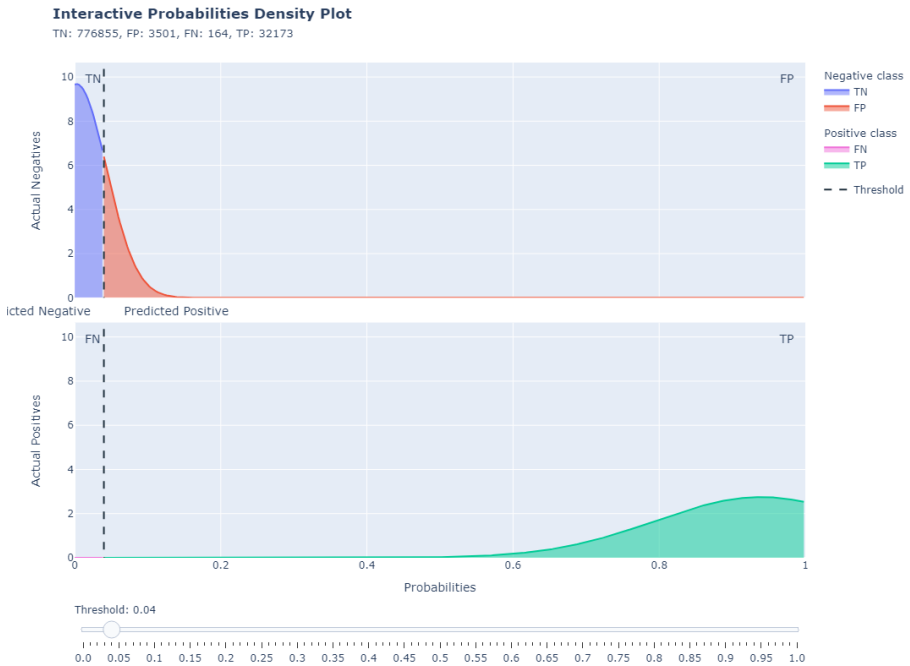


(a) *Distribucija vjerojatnosti modela stabla odluke*

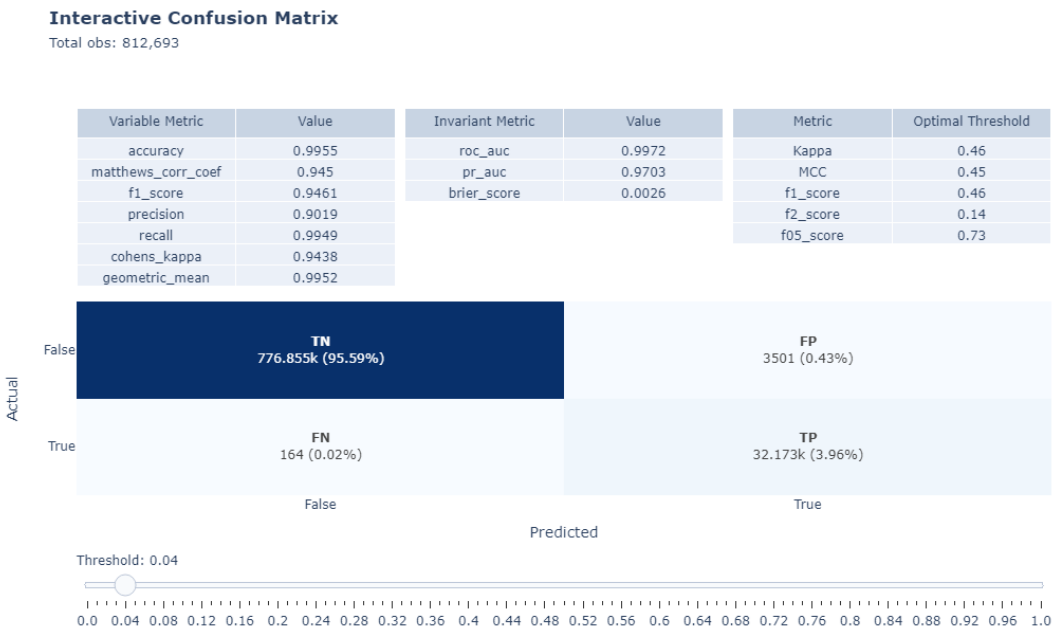


(b) *Matrica zabune modela stabla odluke*

Slika 4.31 *Distribucija vjerojatnosti i matrica zabune za model stabla odluke*

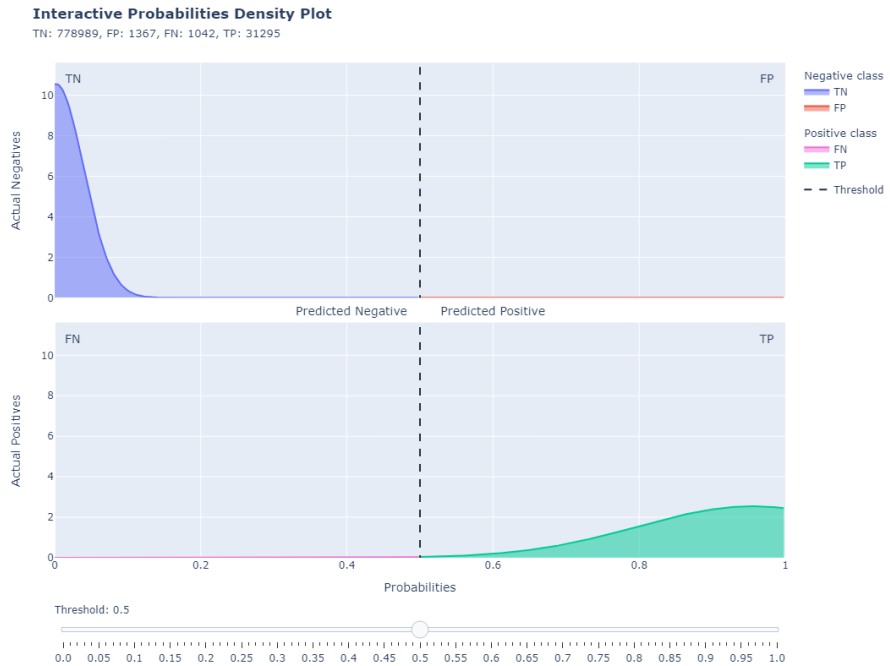


(a) *Distribucija vjerojatnosti modela stabla odluke*

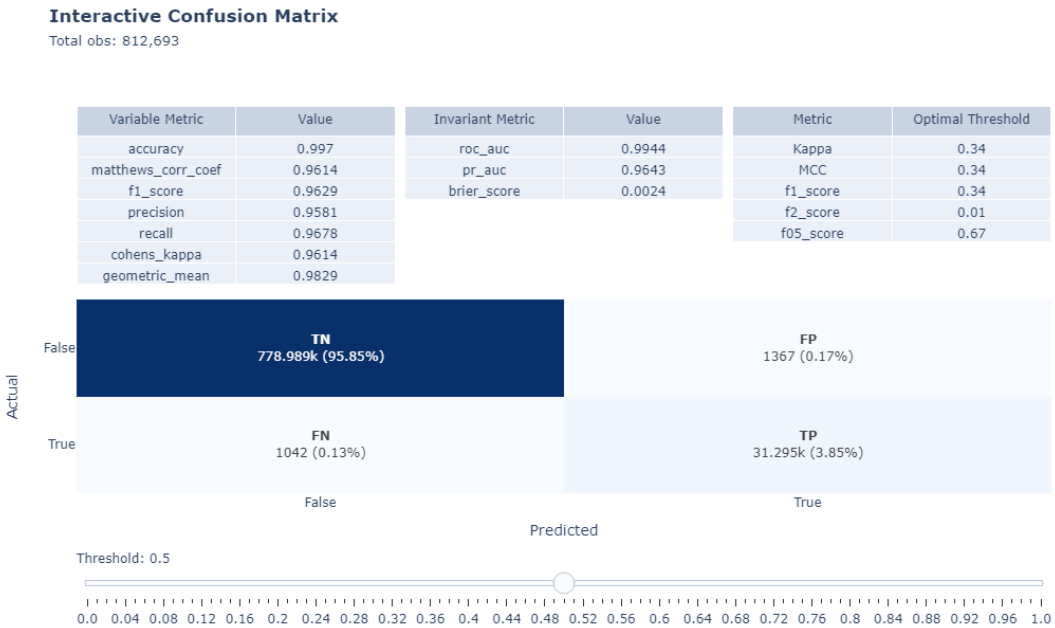


(b) *Matrica zabune modela stabla odluke*

Slika 4.32 *Model stabla odluke - optimalni prag za ROC krivulju*

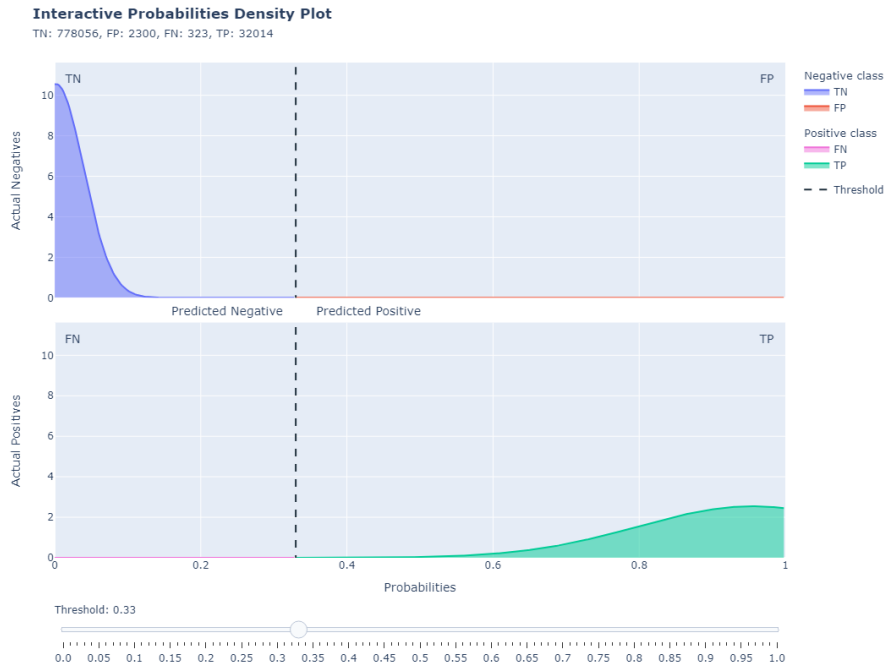


(a) *KNN distribucija vjerojatnosti*

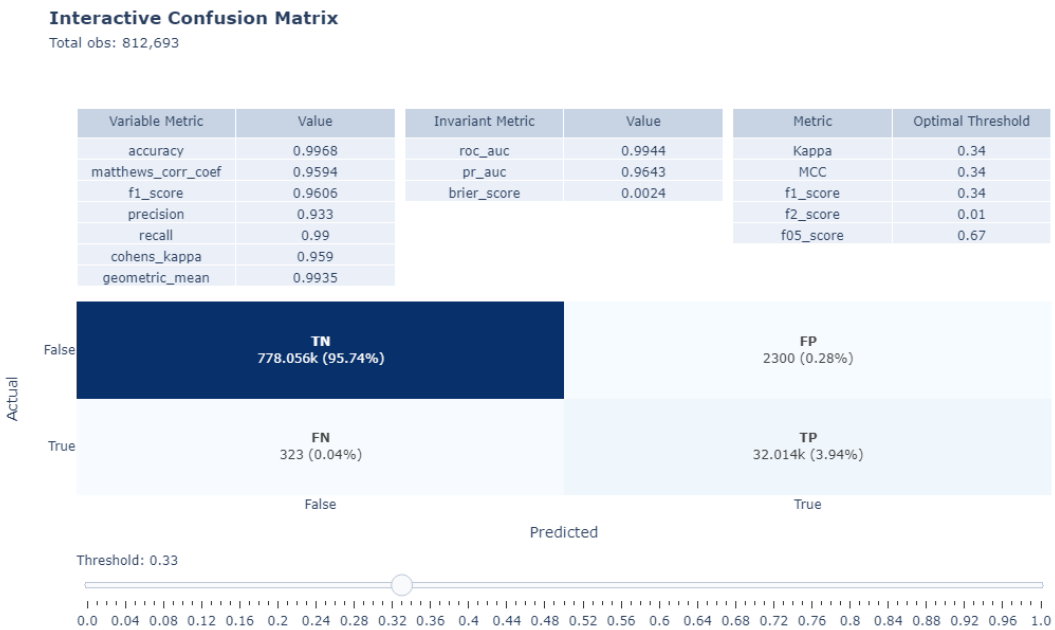


(b) *KNN matrica zabune*

Slika 4.33 *Distribucija vjerojatnosti i matrica zabune za KNN model*



(a) *KNN distribucija vjerojatnosti*



(b) *KNN matrica zabune*

Slika 4.34 *KNN model - optimalni prag za ROC krivulju*

4.4 Usporedba performansi modela strojnog učenja

Nakon detaljnije analize performansi modela treniranih pomoću 3 različita skupa podataka, u tablicama 4.1, 4.2 i 4.3 se mogu vidjeti vrijednosti odabranih klasifikacijskih metrika, na temelju kojih se može odrediti najbolji model za određeni skup podataka. Važno je napomenuti da su prikazani rezultati dobiveni preko vrijednosti klasifikacijskog praga od 0.5. Također, u tablici je navedeno vrijeme predviđanja u sekundama, što predstavlja vrijeme klasificiranja svih uzoraka iz skupa za testiranje.

Za modele trenirane algoritmima slučajne šume, stabla odluke i KNN pomoću NF-BoT-IoT skupa podataka (tablica 4.1), može se reći da imaju slične vrijednosti metrika točnosti, F1-mjere i geometrijske sredine. Prema tome, navedeni modeli se mogu usporediti prema AUC vrijednostima na temelju kojih se može eliminirati model treniran KNN algoritmom, čija je AUC vrijednost najmanja (0.828). Dalje, uspoređujući modele trenirane algoritmima slučajne šume i stabla odluke, zbog gotovo identičnih AUC vrijednosti, bolji model je u ovom slučaju onaj treniran algoritmom stabla odluke. Razlog ovakvog odabira se svodi isključivo na kompleksnost klasifikatora. Drugim riječima, algoritam slučajne šume za treniranje modela koristi više stabala odluke što ga čini kompleksnijim i računalno zahtjevnijim za izvršavanje. To se ujedno može potvrditi i većim vremenom predviđanja od modela treniranog algoritmom stabla odluke, koji sam po sebi predstavlja jedno stablo odluke, koje je po pitanju omjera performansi i korištenih računalnih resursa u ovom slučaju bolji odabir nego algoritam slučajne šume. U usporedbi sa modelom stabla odluke, za naivni Bayesov model se može uočiti da ima niže vrijednosti metrika točnosti, F1-mjere i AUC, ali zato ima veću vrijednost geometrijske sredine. Iako bi se zbog veće vrijednosti geometrijske sredine te time ujednačenije uspješnosti predviđanja normalnog prometa i napada, kao najbolji model trebao uzeti onaj treniran naivnim Bayesovim klasifikatorom, radi što uspješnije detekcije napadačkih uzoraka kao najbolji model se može definirati onaj treniran algoritmom stabla odluke.

Tablica 4.1 *Performanse modela strojnog učenja (NF-BoT-IoT)*

Algoritam	**Točnost**	**F1-mjera**	**G-mean**	**AUC**	**Vrijeme predviđanja (s)**
Naivni Bayes	0.940	0.969	0.867	0.793	0.065
Slučajna šuma	0.987	0.994	0.725	0.987	0.241
Stablo odluke	0.987	0.994	0.727	0.985	0.048
KNN	0.988	0.994	0.729	0.828	3.844

Što se tiče modela treniranih pomoću NF-UNSW-NB15 skupa podataka (tablica 4.2), može se uočiti da je model treniran naivnim Bayesovim klasifikatorom najlošiji, odnosno ima iznimno loše vrijednosti metrika F1-mjere, geometrijske sredine te AUC. Pored toga, ostali modeli imaju vrlo slične vrijednosti klasifikacijskih metrika. Kod modela treniranog KNN algoritmom, može se uočiti dosta duže vrijeme predviđanja u usporedbi sa algoritmima slučajne šume i stabla odluke, na temelju čega se on ne može definirati kao najbolji. Uspoređujući modele slučajne šume i stabla odluke, zbog veće razlike u vrijednostima metrika geometrijske sredine i F1-mjere, kao najbolji model se u ovom slučaju može definirati onaj treniran algoritmom slučajne šume.

Tablica 4.2 *Performanse modela strojnog učenja (NF-UNSW-NB15)*

Algoritam	**Točnost**	**F1-mjera**	**G-mean**	**AUC**	**Vrijeme predviđanja (s)**
Naivni Bayes	0.934	0.214	0.441	0.651	0.078
Slučajna šuma	0.988	0.865	0.926	0.997	0.295
Stablo odluke	0.988	0.855	0.903	0.995	0.095
KNN	0.988	0.868	0.917	0.994	26.477

Kao i kod NF-UNSW-NB15 skupa podataka, kod NF-UNSW-NB15-V2 (tablica 4.3) je odnos performansi između modela isti, jedino su vrijednosti klasifikacijskih metrika za NF-UNSW-NB15-V2 veće, što je posljedica povećanja broja uzoraka te povećanja broja značajki. Što se tiče treniranih modela, naivni Bayesov model se može eliminirati radi loših performansi, dok se KNN model može eliminirati radi iznimno dugog vremena predviđanja. Prema tome, zbog približno istih vrijednosti klasifikacijskih metrika između modela slučajne šume i stabla odluke te zbog znatno manjeg vremena predviđanja od strane modela stabla odluke, za najbolji model se može uzeti onaj treniran algoritmom stabla odluke. Također, zbog najboljih performansi te iznimne brzine prilikom klasificiranja uzoraka, model stabla odluke treniran pomoću NF-UNSW-NB15-V2 skupa podataka je najbolji od svih prethodno navedenih modela.

Tablica 4.3 *Performanse modela strojnog učenja (NF-UNSW-NB15-V2)*

Algoritam	**Točnost**	**F1-mjera**	**G-mean**	**AUC**	**Vrijeme predviđanja (s)**
Naivni Bayes	0.949	0.246	0.452	0.845	0.065
Slučajna šuma	0.997	0.964	0.988	1.000	0.439
Stablo odluke	0.997	0.959	0.982	0.997	0.087
KNN	0.997	0.963	0.983	0.994	147.320

Poglavlje 5

Zaključak

U ovome radu uspoređeno je 12 modela strojnog učenja za detekciju neovlaštenog pristupa u mrežnom prometu. Skupovi podataka korišteni za treniranje i testiranje modela su iznimno neujednačeni. Drugim riječima, skup podataka NF-BoT-IoT sadrži 97.69% napadačkih uzoraka, dok NF-UNSW-NB15 i NF-UNSW-NB15-V2 sadrže 95.54%, odnosno 96.02% uzoraka određenih kao normalan promet respektivno. Prije provođenja samog procesa strojng učenja, iz skupova podataka su uklonjene značajke identifikatora toka i TTL značajke radi nepristranog prikaza sposobnosti modela u učenju potrebnih obrazaca. Nakon provedenog procesa strojnog učenja pomoću algoritama naivnog Bayesovog klasifikatora, stabla odluke, slučajne šume i KNN, dobivene su klasifikacijske metrike za procjenu treniranih modela. Na temelju lošijih vrijednosti klasifikacijskih metrika, uspostavilo se da naivni Bayesov klasifikator zbog njegove pretpostavke o nezavisnosti između značajki nije dobar odabir za treniranje modela. Modeli trenirani KNN algoritmom imaju visoke vrijednosti klasifikacijskih metrika što KNN algoritam čini dobrim odabirom. To ujedno potvrđuju i visoke vrijednosti geometrijske sredine i MCC-a, čime se može opovrgnuti slutnja o potencijalnoj pristranosti modela prema dominantnoj klasi u slučaju neujednačenog skupa podataka. Najbolje performanse su ostvarili modeli trenirani pomoću algoritama stabla odluke i slučajne šume. Svi modeli trenirani pomoću tih algoritama imaju gotovo savršenu ROC krivulju, što algoritme stabla odluke i slučajne šume čini iznimno pouzdanima u situacijama kada su i napadački i normalni uzorci većinski. Za sve modele je također izračunat optimalan prag klasifikacije za ROC krivulju, iz

čega se mogu vidjeti bolje performanse modela u predviđanju i napada i normalnog prometa.

Pored samih performansi modela, potrebno je također napomenuti da se korišteni skupovi podataka razlikuju po vrstama napada u mrežnom prometu, čime primjena modela u realnom okruženju može biti ograničena. Prema tome, prilikom implementacije modela u neko stvarno okruženje, model je potrebno trenirati nad većim skupom podataka koji može pokriti širi spektar vrsta napada. Također, s obzirom da je kod treniranih modela jedan od neizbježnih problema detekcija normalnog prometa kao napadačkog (False Positive), implementacija modela u neki sustav bi se mogla ograničiti na upozorenje o potencijalnom kibernetičkom napadu ili kao dodatna mjera autentifikacije u slučaju detekcije abnormalnog ponašanja korisnika.

Literatura

[1] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a standard feature set for network intrusion detection system datasets," *Mobile Networks and Applications*, vol. 27, no. 1, pp. 357–370, nov 2021.

[2] "Smartdraw.com," , s Interneta, https://www.smartdraw.com/decision-tree/, 15. svibnja 2023.

[3] R. Abilash, "Applying random forest (classification) - machine learning algorithm from scratch with real datasets," , s Interneta, https://rb.gy/121ly, 15. svibnja 2023.

[4] "What is the k-nearest neighbors algorithm?" , s Interneta, https://www.ibm.com/topics/knn, 15. svibnja 2023.

[5] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow datasets for machine learning-based network intrusion detection systems," in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Springer International Publishing, 2021, pp. 117–135.

[6] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, jul 2019.

[7] *An Extensible NetFlow v5/v9/IPFIX Probe for IPv4/v6.*, , s Interneta, https://www.ntop.org/guides/nprobe/cli_options.html, 15. studenog 2022.

[8] Y. Gong, *Detecting Worms and Abnormal Activities with NetFlow, Part 2*, , s Interneta, https://community.broadcom.com/symantecenterprise/viewdocument/detecting-worms-and-abnormal-activi?CommunityKey=1ecf5f55-9545-44d6-b0f4-4e4a7f5f5e68&tab=librarydocuments/, 19. kolovoza 2023.

[9] N. Vlajic, M. Andrade, and U. Nguyen, "The role of dns ttl values in potential ddos attacks: What do the major banks know about it?" *Procedia Computer Science*, vol. 10, pp. 466–473, 2012, aNT 2012 and MobiWIS 2012. , s Interneta, https://www.sciencedirect.com/science/article/pii/S1877050912004176

[10] M. Sarhan, S. Layeghy, and M. Portmann, "Evaluating standard feature sets towards increased generalisability and explainability of ml-based network intrusion detection," *arXiv preprint arXiv:2104.07183*, 2021.

[11] S. Majić, "Primjena strojnog učenja u svrhu detektiranja anomalija u streaming podacima (diplomski rad)," Master's thesis, 2017.

Sažetak

U ovom završnom radu je opisan postupak kreiranja modela strojnog učenja za detekciju anomalija u mrežnom prometu, pri čemu se pod detekcijom anomalija misli na detekciju neovlaštenog pristupa. Opisani su skupovi podataka NF-BoT-IoT, NF-UNSW-NB15 i NF-UNSW-NB15-V2 te njihove značajke. Opisan je postupak provođenja procesa strojnog učenja, te je taj proces primijenjen za algoritme naivni Bayesov klasifikator, stablo odluke, slučajna šuma i KNN. Navedeni klasifikatori su korišteni za treniranje modela nad 3 različita skupa podataka, te se na temelju klasifikacijskih metrika odredio najbolji model za detekciju anomalija u mrežnom prometu. Najbolji model je treniran pomoću algoritma stabla odluke i NF-UNSW-NB15-V2 skupa podataka. Po pitanju primjene u stvarnom svijetu, navedeni model je ograničen na prepoznavanje vrsta napada koje obuhvaća skup podataka za treniranje. Prema tome, za detekciju više vrsta napada potrebno je koristiti puno veći skup podataka koji sadrži širi spektar napadačkih uzoraka. Također, radi neizbježnih *False Positive* vrijednosti, navedeni model je potrebno implementirati isključivo kao sustav upozorenja na potencijalni napad na temelju kojeg se mogu poduzeti daljnje akcije.

Ključne riječi **— strojno učenje, kibernetička sigurnost, sustav za detekciju neovlaštenog pristupa**

Abstract

This thesis describes the process of creating a machine learning model for detection of anomalies in network traffic, whereby the anomaly detection refers to the intrusion detection. NF-BoT-IoT, NF-UNSW-NB15 and NF-UNSW-NB15-V2 datasets and their features are described. The procedure of implementing the machine learning process is described, and this process is applied to Naive Bayes, Decision Tree, Random Forest and KNN classifiers. The specified classifiers were used to train the model on 3 different datasets, and based on the classification metrics, the best model for detecting anomalies in network traffic was determined. The best model was trained using the Decision Tree algorithm and the NF-UNSW-NB15-V2 dataset. In terms of real-world application, the mentioned model is limited to recognizing the

types of attacks that comprise the training dataset. Therefore, for the detection of more types of attacks, it is necessary to use a much larger dataset that contains a wider spectrum of attack patterns. Also, due to the inevitable False Positive values, the mentioned model should be implemented exclusively as a warning system for a potential attack based on which further actions can be taken.

Keywords **— machine learning, cybersecurity, intrusion detection system**

Dodatak A

Github repozitorij - programski kod

https://github.com/Marko132001/ML-based-Intrusion-Detection-System