UNIVERZITET U NIŠU
ELEKTRONSKI FAKULTET

**Danijel D. Čabarkapa**

# Nova metoda detekcije DDoS napada primenom softverski definisanih mreža

DOKTORSKA DISERTACIJA

Niš, 2023.

UNIVERSITY OF NIŠ
FACULTY OF ELECTRONIC ENGINEERING

**Danijel D. Čabarkapa**

# A New Method of Detecting DDoS Attacks Using Software-Defined Networks

DOCTORAL DISSERTATION

Niš, 2023.

## Podaci o doktorskoj disertaciji

| | |
|---|---|
| Mentor: | dr Dejan Rančić, redovni profesor<br>Elektronski fakultet Niš, Univerzitet u Nišu |
| Naslov: | Nova metoda detekcije DDoS napada primenom softverski definisanih mreža |
| Rezime: | Ova disertacija je rezultat detaljnog istraživanja metoda detekcije i identifikacije DDoS napada odbijanjem mrežnih servisa. Naučna opravdanost istraživanja zasniva se na činjenici da se ovaj aktuelni tip napada sve više izvršava u okviru softverski definisanih mreža, koje predstavljaju sasvim novu i sve važniju paradigmu mrežnog upravljanja.<br>Predložena je i analizirana nova metoda za detekciju anomalija i DDoS napada koja primenjuje kombinovani pristup koji obuhvata proračun entropije mrežnih atributa i primenu algoritama nadgledanog mašinskog učenja. Proračun entropije kao metrike visokog nivoa, primenjen je na ivičnom OpenFlow sviču mreže, kako bi se realizovala brza detekcija napada, dok se algoritmi nadgledanog mašinskog učenja izvršavaju na kontroleru, čime je postignuta preciznija detekcija, smanjen broj lažnih alarma i izvršena efikasna klasifikacija mrežnog saobraćaja.<br>Detaljnom eksperimentalnom analizom koja je izvršena za simulacionu topologiju softverski definisane mreže, dobijeni su rezultati koji pokazuju da predložena metoda detekcije DDoS napada postiže visok stepen efikasnosti i tačnosti klasifikacije. Takođe, predloženo rešenje poseduje karakteristiku opštosti, pa ima mogućnost da detektuje različite napade zasnovane na preplavljivanju. |
| Naučna oblast: | Elektrotehničko i računarsko inženjerstvo (Računarstvo i informatika) |
| Naučna disciplina: | Bezbednost računarskih mreža |
| Ključne reči: | detekcija napada, entropija, napad odbijanjem servisa, softverski definisane mreže, nadgledano mašinsko učenje, bezbednost mreža |
| UDK: | 004.7.056.5:004.722(043.3) |
| CERIF klasifikacija: | T120: Sistemski inženjering, računarska tehnologija |
| Tip licence Kreativni zajednice: | **CC BY-NC-ND** |

**Data on Doctoral Dissertation**

| | |
|---|---|
| Doctoral Supervisor: | PhD Dejan Rančić, associate professor<br>University of Niš, Faculty of Electronic Engineering |
| Title: | A new method of detecting DDoS attacks using software-defined networks |
| Abstract: | This dissertation is the result of a detailed research of detection and identification of DDoS attacks by denying network services. The scientific justification of the research is based on the fact that this important type of attack is increasingly carried out within software-defined networks, which represent a completely new and increasingly important paradigm of network management.<br>A new method for the detection of anomalies and DDoS attacks is proposed and analyzed, which applies a combined approach that includes the entropy calculation of network attributes and the application of supervised machine learning algorithms. Entropy calculation as a high-level metric was applied on the edge OpenFlow network switch to realize fast attack detection, while supervised machine learning algorithms were executed on the controller, which achieved more accurate detection, reduced the number of false alarms and performed effective classification of network traffic.<br>The detailed experimental analysis performed for the simulation topology of the software-defined network, obtained results that show that the proposed DDoS attack detection method achieves a high degree of efficiency and classification accuracy. Also, the proposed solution has the characteristic of generality, so it has the ability to detect different flooding attacks. |
| Scientific Field: | Electrical and Computer Engineering (Computer Science) |
| Scientific Discipline: | Computer networks security |
| Key Words: | intrusion detection, entropy, distributed denial of service, software defined networks, supervised machine learning, network security |
| UDC: | 004.7.056.5:004.722(043.3) |
| CERIF Classification: | T120 System engineering, computer technology |
| Creative Commons License Type: | **CC BY-NC-ND** |

*Sa iskrenošću se zahvaljujem mentoru dr Dejanu Rančiću, redovnom profesoru Elektronskog fakulteta u Nišu, na velikoj podršci i korisnim savetima, koji su bili od suštinskog značaja za izradu ove doktorske disertacije.*

*Disertaciju posvećujem supruzi Marjani, bez čije ljubavi, podrške, strpljenja i razumevanja ovo istraživanje ne bi bilo moguće.*

*Danijel Čabarkapa*

*Šabac, 2023.*

# SADRŽAJ

# Spisak slika

# Spisak tabela

# Lista skraćenica

| | |
|---|---|
| **SDN** | Software Defined Networks |
| **DoS** | Denial of Service |
| **DDoS** | Distributed Denial of Service |
| **NIDS** | Network Intrusion Detection System |
| **OSI** | Open System Interconnection |
| **PDU** | Packet Data Unit |
| **CIA** | Confidentiality, Integrity, Availability |
| **NIST** | National Institute of Standards and Technology |
| **DRDoS** | Distributed Reflected Denial of Service |
| **P2P** | Peer-to-Peer |
| **IRC** | Internet Relay Chat |
| **TCP/IP** | Transmission Control Protocol/Internet Protocol |
| **UDP** | User Datagram Protocol |
| **ICMP** | Internet Control Message Protocol |
| **PoD** | Ping of Death |
| **IPX** | Internetwork Packet Exchange |
| **DNS** | Domain Name System |
| **SNMP** | Simple Network Management Protocol |
| **SMTP** | Simple Mail Transfer Protocol |
| **SRDDoS** | Slow Rate DDoS |
| **RDDoS** | Ransom DDoS |
| **QoS** | Quality of Services |
| **ONF** | Open Network Foundation |
| **NOS** | Network Operating System |
| **NFV** | Network Functions Virtualization |
| **FT** | Flow Table |
| **TCAM** | Ternary Content-Addressable Memory |
| **GbE** | Gigabit Ethernet |
| **RIB** | Routing Information Base |
| **PaaS** | Platform-as-a-Service |
| **API** | Application Programming Interface |
| **SBI** | SouthBound Interface |

| | |
|---|---|
| **NBI** | NorthBound Interface |
| **OFA** | OpenFlow Agent |
| **MAC** | Media Access Control |
| **ASIC** | Application-Specific Integrated Circuit |
| **TLS** | Transport Layer Security |
| **AI** | Artificial Intelligence |
| **ML** | Machine Learning |
| **DL** | Deep Learning |
| **TP** | True Positive |
| **TN** | True Negative |
| **FP** | False Positive |
| **FN** | False Negative |
| **TNR** | True Negative Rate |
| **FPR** | False Positive Rate |
| **ANN** | Artificial Neural Network |
| **RNN** | Recurrent Neural Network |
| **DCRNN** | Diffusion Convolutional RNN |
| **ARP** | Address Resolution Protocol |
| **KDN** | Knowledge Defined Network |
| **SVM** | Support Vector Machine |
| **k-NN** | K-Near Neighbors |
| **NB** | Naïve Bayes |
| **RF** | Random Forest |
| **DT** | Decision Tree |
| **R2L** | Root to Local |
| **RSMQ** | Redis Simple Message Queue |
| **HAC** | Hierarchical Agglomerative Clustering |
| **TSK-FS** | Takagi-Sugeno-Kang Fuzzy System |
| **FPGA** | Field-Programmable Gate Array |
| **DPPSN** | Deeply Programmable Packet-Switching Node |
| **HTTPS** | Hyper Text Transfer Protocol Secure |
| **SSH** | Secure Shell Protocol |
| **FTP** | File Transfer Protocol |
| **PCA** | Principle Component Analysis |
| **TTL** | Time-to-Live |
| **IDE** | Integrated Development Environment |

# 1. UVOD

Tokom poslednje dve decenije svedoci smo neverovatno brzog razvoja Interneta i računarskih mreža, koji su postali deo svakodnog života čoveka i važna komponenta u mnogim sferama njegove aktivnosti. Prema statistici kompanije Cisco Systems, do 2023. godine oko 66% ukupne svetske populacije koristi Internet ili različite mrežne servise, sa prognoziranim trendom značajnog rasta u narednim godinama [1]. Nove tehnologije kao što su virtuelizacija, koncept računarstva u oblaku (*Cloud Computing*), Internet stvari IoT (*Internet of Things*) i centri velikih podataka (*Big Data*) uzrokovali su da su prenos, skladištenje i upravljanje sve većom količinom podataka postali veliki izazov za kompanije i provajdere usluga. Snažan razvoj ovih tehnologija praćen je sve većim brojem korisnika, zahtevima za prenos podataka sve bržim komunikacionim linkovima, novim mrežnim servisima, kao i novim načinima za skladištenje, prenos i obradu velike količine podataka.

Integracija novih mrežnih tehnologija i servisa sa tradicionalnim mrežama, troškovi njihovog razvoja, održavanja i upravljanja konstantno su u porastu. Tradicionalne konvergentne TCP/IP mreže po svojoj prirodi koriste zajednički medijum za prenos svih tipova saobraćaja, pa je za ispunjenje zahteva novih mrežnih servisa potrebno izvršiti čitav niz promena, kako u fizičkoj, tako i u logičkoj mrežnoj strukturi. Prenos sve veće količine podataka, kao i integracija novih protokola i servisa dovodi do niza problema u sferama skalabilnosti, bezbednosti, pouzdanosti i fleksibilnosti tradicionalnih mreža [2].

Da bi se pomenuti problemi rešili, bilo je neophodno primeniti nove načine upravljanja mrežama i njihovim resursima, i omogućiti automatizaciju, programabilnost i efikasnije načine upravljanja mrežnim saobraćajem unutar kompleksnih topologija. Navedene zahteve moguće je rešiti primenom softverski definisanih mreža SDN (*Software Defined Networks*), kao nove mrežne paradigme, koja je u ovu oblast donela brojna unapređenja. Najznačajnija osobina SDN arhitekture jeste razdvajanje kontrolne ravni i ravni podataka, kao i logička centralizacija SDN kontrolera, kojom je omogućeno praćenje stanja mreže na

globalnom nivou, što olakšava razvoj i implementaciju naprednih mrežnih funkcija, servisa i aplikacija [3, 4].

Glavna osobina SDN mreža tj. razdvajanje kontrolne ravni od ravni podataka sa gledišta njihove bezbednosti predstavlja dodatni istraživački izazov. Postoje izvesne bezbednosne specifičnosti SDN mreža koje nisu inherentno prisutne u tradicionalnim mrežama. SDN mreže implementiraju funkcije mrežne bezbednosti na drugačiji način u odnosu na tradicionalne, prvenstveno kroz automatsko programiranje i konfigurisanje mrežnih uređaja. SDN bezbednosne procedure primenjuju proaktivno praćenje, analizu, otkrivanje i prevenciju pretnji i napada, pre svega u virtuelnim mrežnim infrastrukturama centara podataka [5]. Bezbedonosne SDN procedure mogu se efikasno prilagođavati novonastalim promenama mrežne infrastrukture, tako da pružaju dugoročnu zaštitu visokog nivoa protiv različitih novonastalih mrežnih pretnji, anomalija i ranjivosti.

Sa dinamičnim razvojem novih mrežnih tehnologija i servisa, mrežni napadi su postali sve kompleksniji i sofisticiraniji, ali se u poslednje vreme izdvajaju distribuirani napadi odbijanjem servisa DDoS (*Distributed Denial of Service*), botnet napadi, ucenjivački (*ransomware*) i napadi preplavljivanjem. DDoS napadi od samog početka razvoja Interneta predstavljaju jed nu od najvećih pretnji, a tokom vremena, usavršavanjem i ubrzanim razvojem mrežnih tehnologija, dolazi do velike učestanosti ovih napada izuzetno velikih intenziteta i stepena destruktivnosti [6]. Rešenja za pravovremenu detekciju i sprečavanje DDoS napada danas su od jako velikog značaja za pravilno funkcionisanje mreža i dostupnost njihovih naprednih servisa. Sa tim ciljem je i sprovedeno ovo istraživanje.

Određeni broj savremenih rešenja za detekciju DDoS napada zasniva se na otkrivanju anomalija i koristi statističke osobine mrežnog saobraćaja. Entropija koja je u ovom radu izabrana kao metoda za detekciju anomalija, ima karakteristiku jednostavnosti implementacije i opštosti, tako da može da detektuje različite topive napada. Međutim, njen glavni nedostatak jeste pojava određenog broja lažnih alarma, koji izazivaju promenu njene vrednosti na sličan način kao i stvarni DDoS napad. Ciljevi ovog istraživanja su kako što efikasnije upotrebiti entropiju za detekciju anomalija, poboljšati brzinu i efikaso izvršiti klasifikaciju mrežnih napada primenom mašinskog učenja u SDN mrežnom okruženju.

Kroz nekoliko delova predstavljen je naučni doprinos ove disertacije, primenljiv na oblast bezbednosti SDN mreža. Predložena je nova metode za detekciju DDoS napada kombinovanim postupkom, koji je realizovan kroz dva modula. Prvi modul vrši brzu

detekciju anomalija jer se realizuje algoritmom koji se izvršava na programabilnom ivičnom sviču SDN mreže, čime se kontroler rasterećuje dodatnog procesiranja podataka, a sam postupak detekcije DDoS napada koristi hardverske resurse OpenFlow sviča. Dobijen signal promene entropije i broja mrežnih tokova inicira SDN kontroler, koji isključuje portove svičeva sa kojih su napadi primljeni. Drugi modul koristi algoritme mašinskog učenja kako bi se analizom statistike fluktuacija mrežnih tokova, dobio jednostavan obučavajući skup kojim se klasifikuje saobraćaj kao legitimni ili maliciozni, tj. DDoS napad.

Simulaciona SDN mreža je upotrebljena za eksperimentalne testove, što daje veće mogućnosti promene parametara nego na realnim mrežama. Softverski je generisan legitimni mrežni saobraćaj, a zatim saobraćaj određenog intenziteta koji simulira TCP-SYN flood i ICM flood napad. Prvi korak je bila analiza mrežnog saobraćaja sprovedena nad javno dostupnim skupovima podataka, a zatim je ta analiza realizovana nad prikupljenim podacima iz simulacione SDN mrežne topologije. Izvršeno je poređenje dobijenih rezultata klasifikacije i predikcije DDoS napada za oba skupa mrežnih podataka.

Predstavljena metoda detekcije DDoS napada je jedna od retkih koja kombinuje entropiju i tehnike nadgledanog mašinskog učenja, a koja je primenljiva u softverski definisanim mrežnim topologijama sa velikim brojem mrežnih čvorova i intenzivnim mrežnim saobraćajem. Predložena metoda takođe doprinosi razvoju novih postupaka rešavanju problema bezbednosti SDN mreža, kao i analize različitih struktura mrežnog saobraćaja.

## 1.1. Ciljevi i značaj istraživanja

Polazna osnova istraživanja realizovanog u ovoj disertaciji je činjenica da je moguće izvršiti efikasnu detekciju DDoS napada u mrežama sa intenzivnim mrežnim saobraćajem, efikasnom primenom naprednih algoritama nadgledanog mašinskog učenja, u kombinaciji sa metodom detekcije napada koja se bazira na promeni entropije. Za ovakvu detekciju DDoS napada moguće je razviti efikasno rešenje koje će omogućiti detekciju mrežnih anomalija i tokom normalnog rada mrežnih okruženja, pri čemu bi se očuvala dostupnost i efikasnost njihovih servisa. Druga pretpostavka koristi činjenicu da se programabilnošću mrežnog OpenFlow sviča, može postići manje kašnjenje tokom procesa detekcije napada.

Istraživanje u ovoj disertaciji je u skladu sa naučnom opravdanošću, zasnovanom na sve većoj potrebi implementacije novih tehnologija u oblasti bezbednosti računarskih mreža, sa ciljem realizacije efikasnih rešenja za detekciju različitih klasa napada u softverski definisanim mrežama. Naučni ciljevi disertacije su uslovljeni istaživanjima sa elementima statističke analize mrežnih podataka, entropije podataka, algoritama mašinskog učenja i programabilnosti softverski definisanih mreža. Osnovni ciljevi istraživanja su predstavljeni kroz sledeće faze:

- Analiza postojećeg stanja u oblasti detekcije DDoS napada u softverski definisanim mrežnim strukturama, pregled značajnih rešenja koja se odnose na rešavanje problema detekcije ovih napada, kao i analiza njihove primenljivosti
- Predlog nove metode za detekciju DDoS napada zasnovane na analizi parametara mrežnog saobraćaja, primenom kombinovane metode proračuna entropije i algoritama nadgledanog mašinskog učenja
- Dizajn algoritma za detekciju mrežnih anomalija, koji proračunom entropije i mrežnih tokova na ivičnom sviču izvršava znatno bržu detekciju napada, dok se na SDN kontroleru izvršava algoritam nadgledanog mašinskog učenja, kako bi se povećao broj ispravnih detekcija i klasifikovao saobraćaj na legitiman i maliciozan
- Testiranje, primena posebno izabranih algoritama nadgledanog mašinskog učenja, kao i analiza nad javnim skupovima podataka, kao i detaljna analiza izvršena nad podacima prikupljenim iz softverski simuliranog SDN mrežnog okruženja

Značaj ovog istraživanja se ogleda kroz definisanje i primenu nove metode detekcije DDoS napada, za koju je mrežnom simulacijom i detaljnom analizom prikupljenih podataka potvrđeno da može da omogući visok nivo bezbednosti naročito u mrežnim okruženjima specifičnim isključivo za softverski definisane mreže. Pored toga, značaj istraživanja je potvrda pretpostavke da je predloženom metodom moguće značajno povećati brzinu detekcije i efikasnost mrežnih klasifikatora, kao i realizovati softversko rešenje koje će biti u stanju da efikasno detektuje različite tipove DDoS napada, uključujući i one koji su u obučavajućem skupu podataka opisani malim brojem atributa.

## 1.2. Pregled sadržaja disertacije po poglavljima

Disertacija je organizovana u deset poglavlja.

U **poglavlju 1**, pored uvodnih razmatranja, navedeni su značaj i ciljevi istraživanja.

U **poglavlju 2** opisani su osnovni koncepti mrežnih napada odbijanjem servisa. Predstavljeni su osnovni mehanizmi izvršavanja DDoS napada na računarske mreže, i izvršena njihova taksonomija prema različitim kriterijumima. Takođe, prikazani su osnovni statistički podaci o DDoS napadima i ukratko razmotreni najdestruktivniji zabeleženi napadi.

U **poglavlju 3** prikazan je koncept softverski definisanih mreža, kao nove i specifične mrežne arhitekture. Kroz opis referentnog modela i mehanizama mrežne komunikacije, predstavljene su najvažnije karakteristike, prednosti kao i nedostaci ovog mrežnog koncepta, u okviru kojeg je realizovano istraživanje.

U **poglavlju 4** razmatrani su glavni aspekti bezbednosti softverski definisanih mreža. Utvrđeni su osnovni tipovi i mehanizmi DDoS napada u okviru ovih mreža, i analizirana je metodologija njihovog izvršavanja na konkretnom primeru softverski definisane mrežne topologije.

U **poglavlju 5** izložen je pregled postojećih istraživanja i rešenja koja su u relaciji sa ovom disertacijom. Navođenjem referenci koje se odnose na najznačajnije pristupe rešavanju problema detekcije DDoS napada, a posebno radova u kojima su opisana rešenja problematike napada na softverski definisana mrežna okruženja, koji su osnova ovog istraživanja.

U **poglavlju 6** razmatrane su najvažnije teorijske osnove i tehnologije koje su se koristile za realizaciju predloženog rešenja detekcije napada. Prvi deo poglavlja opisuje entropijske metode za detekciju napada, a koje su primenjene u prvom delu rešenja, tokom faze predobrade mrežnih podataka. Drugi deo poglavlja prezentuje detaljan pregled algoritama nadgledanog mašinskog učenja, koji su upotrebljeni u drugom delu rešenja za detekciju napada. Takođe, opisani su osnovni kriterijumi za ocenu efikasnosti sistema za detekciju napada, a koji se zasnivaju na principima mašinskog učenja.

**Poglavlje 7** detaljno predstavlja predloženo rešenje za detekciju DDoS napada. U skladu sa sprovedenim fazama istraživanja, ovo poglavlje se sastoji od nekoliko celina. U prvom delu je prikazana arhitektura predloženog rešenja, sa detaljnim opisom oba modula: za

proračun entropije i klasifikaciju napada mašinskim učenjem. Predstavljen je koncept rešenja, metodologija njegovog razvoja, kao i postupci analize i pretprocesiranja specifičnih mrežnih podataka. U drugom delu je izložena specifikacija hardvera i softvera simulacionog mrežnog okruženja. Opis mrežne topologije u okviru koje je primenjeno rešenje za detekciju napada izložen je u trećem delu. Algoritam za proračun entropije i detekciju anomalija detaljno je predstavljen u četvrtom delu. U petom delu prezentovani su javni skupovi podataka i skup podataka generisan iz simulacione topologije, koji su zatim korišćeni u algoritmima mašinskog učenja.

U **poglavlju 8**, na osnovu predloženog rešenja za detekciju DDoS napada, predstavljeni su rezultati dobijeni detaljno izloženim eksperimentalnim simulacionim testovima. Na osnovu ovih rezultata, analizirano je predloženo rešenje u kontekstu klasifikacije i predikcije napada

**Poglavlje 9** prezentuje zaključke koji se mogu definisati iz ove disertacije.

U **poglavlju 10** izloženi su potencijalni pravci daljeg razvoja i predloženi mogući naredni koraci koji mogu biti zanimljivi u ovoj oblasti istraživanja.

# 2. MREŽNI NAPADI ODBIJANJEM SERVISA

U ovom poglavlju opisani su osnovni principi izvršavanja napada na računarske mreže, a sa posebnim fokusom na napade zasnovane na odbijanju mrežnih servisa. Pojmovi anomalija mrežnog saobraćaja i kategorizacija napada objašnjeni su u prvom delu poglavlja. Prikazana klasifikacija DDoS napada nastala je kao sažetak većeg broja radova koji su se koristili u ovom istraživanju. Posebna pažnja posvećena je analizi DDoS napada preplavljivanjem, pošto su takvi napadi upotrebljeni u eksperimentalnom delu istraživanja. U završnom delu poglavlja navedeni su značajniji statistički podaci o DDoS napadima, a zatim su ukratko opisani primeri sa najvećim zabeleženim intenzitetima napada.

## 2.1. Anomalije mrežnog saobraćaja

Definicija sistema za detekciju mrežnih napada NIDS (*Network Intrusion Detection System*) polazi od činjenice da je to pasivni sistem koji se najčešće implementira preko drugog mrežnog uređaja koji mu šalje kopiju saobraćaja na analizu, pri čemu se formirani tok saobraćaja nastavlja da prosleđuje ka internom delu mreže [7]. U višeslojnom OSI (*Open System Interconnection*) referentnom mrežnom modelu postoje značajne razlike između napada na različite ravni, pa je u praksi gotovo nemoguće realizovati jedinstven metod detekcije koji bi bio primenljiv za sve tipove napada. Metode detekcije koje funkcionišu samo za jednu klasu napada imaju prednosti zbog jednostavnosti rešenja i manjeg kašnjenja, ali za savremene mrežne sisteme sa širokim spektrom napada ove metode nisu najbolje rešenje [8].

Prema načinu analize podataka, NIDS sistemi se klasifikuju kao: sistemi zasnovani na potpisu (*signature-based, misuse*), na detekciji anomalija (*anomaly-based*) i hibridni sistemi (*hybrid-based*) [9-11]. Detekcija zasnovana na potpisu koristi jedinstvene obrasce, formirane od strane već poznatih napada. Obrasci su definisani vrednostima PDU (*Packet Data Unit*) polja u zaglavljima mrežnih paketa. Vrši se poređenje PDU polja pristiglih paketa sa obrascem poznatog napada. Ovaj metod detekcije je jednostavan za implementaciju, ima

nizak nivo pogrešnih detekcija, ali nije efikasan u mrežama gde se često javljaju novi tipovi napada. Takođe, ovi sistemi zahtevaju dobro poznavanje funkcionisanja mreže, kao i osobina varijacija u načinima izvršavanja napada, što praktično nije uvek jednostavno realizovati. U literaturi je prepoznat određeni broj radova sa predloženim rešenjima za detekciju napada koja se zasnovaju isključivo na potpisima [12, 13].

Anomalija se definiše kao svaka promena ponašanja saobraćaja u mreži, a koja odstupa od prethodno definisanog legitimnog ili normalnog ponašanja, i ima potencijal da naruši funkcionalnost mreže. Legitimni mrežni saobraćaj određen je vrednostima mrežnih parametara merenih u određenom vremenskom intervalu u kojem uređaj koji ga generiše nije sigurnosno ugrožen, i radi na način koji je definisao njegov proizvođač. Ukoliko se dolazni saobraćaj značajno razlikuje od unapred definisanog legitimnog, tada se on identifikuje kao anomalija [14]. U okviru dostupne literature, uglavnom se navodi šest osnovnih kategorija detekcije anomalija: statistička, klasifikaciona, tehnika najbližih suseda, klasterovanje, tehnika zasnovana na teoriji informacija i na teoriji spektra [15, 16]. Istraživanje u ovoj disertaciji koristi statističku analizu parametara mrežnog saobraćaja u realnom vremenu. Na osnovu prikupljenih mrežnih podataka prvo se definiše model legitimnog saobraćaja, a nakon toga se statističkim zaključivanjem određuje da li nove instance tokova saobraćaja odgovaraju tom definisanom modelu. U najvećem broju rešenja za detekciju napada koriste se statistički parametri saobraćaja kao što su entropija, korelacija, standardna devijacija, kovarijansa itd [17]. Glavni problem statističke detekcije jeste na koji način odrediti graničnu vrednost (*threshold*) između legitimnog saobraćaja i anomalije. Ako statističke vrednosti odabranih parametara saobraćaja prelaze definisanu graničnu vrednost, tada se detektuju anomalije. Ako je granična vrednost suviše mala, dolazi do detekcije velikog broja lažnih alarma, dok velika granična vrednost stvara veliki broj lažno negativnih rezultata.

Anomalije povezane sa bezbednošću mreže su važne u kontekstu održivosti njenog pravilnog funkcionisanja. Takve anomalije obično nastaju kao posledica višestrukih aktivnosti nekog mrežnog čvora koji ugrožava dostupnost mrežne infrastrukture ili nekog njenog servisa. Aktivnosti koje narušavaju bezbednost mreže mogu biti prikupljanje podataka o mrežnim resursima, preplavljivanje mreže velikim nekorisnim saobraćajem ili neovlašćeno preuzimanje akreditiva mrežnih korisnika. Ovaj oblik anomalija se uglavnom poistovećuje sa terminom napada, jer se zasniva na malicioznim aktivnostima koje utiču na bezbednost mreže.

## 2.2. **Napadi na računarske mreže**

Napad na računarske mreže definiše se kao skup različitih aktivnosti koje utiču na poverljivost, integritet i raspoloživost njihovih servisa i resursa, definisanih u okviru CIA trijade (*Confidentiality, Integrity, Availability*). CIA trijada definiše metode obezbeđenja i kontrole bezbednosti informacija i sistema, kao i karakterizaciju pretnji, ranjivosti i sigurnosnih procesa u informacionim sistemima [18]. Mada se CIA trijada navodi kao osnovni cilj bezbednosti mreža, u dostupnoj literaturi se koriste dodatni pojmovi kao što su autentičnost, privatnost, neporicanje i kontrola, kojima se bezbednost dodatno definiše.

Napad na računarsku mrežu je specifična aktivnost izvršena sa ciljem prekida njenog funkcionisanja ili neovlašćenog pristupa njenom zaštićenom delu ili resursu. Napadi se uglavnom manifestuju kao onemogućavanje ili otežavanje pristupa mrežnim servisima i resursima, ili kao kompromitovanje informacija i poverljivih podataka. Napadi mogu biti usmereni prema različitim delovima mreže, mogu se izvršiti na bilo kom sloju OSI modela, tako da ciljevi napada mogu biti mrežni uređaji, aplikacije ili servisi kod kojih je upotrebljen neki sigurnosni propust.

Postoji više definicija pojma detekcije napada, ali je verovatno najsadržajniju dao NIST (*National Institute of Standards and Technology*), Američki nacionalni institut za standarde i tehnologiju, koji je detekciju napada opisao kao "postupak praćenja događaja koji se dešavaju u računarskoj mreži i njihovu analizu na pojedine znake napada, a koji su u principu postupci ugrožavanja poverljivosti, integriteta i raspoloživosti primenjenih sigurnosnih mehanizama posmatranog računara ili mreže" [19].

Koje će se procedure za detekciju napada sprovesti zavisi od više faktora, a naročito od tipa mrežnog napada. Treba uzeti u obzir da su napadi mogući u svim ravnima OSI modela, i da je njihov glavni cilj da utiču na rad mnogih mrežnih servisa ili protokola. Napadi na ravan aplikacija koriste ranjivosti programskog kôda aplikacije, pa su njihovi efekti uglavnom ograničeni na servise koje ta aplikacija direktno koristi. Ovoj grupi pripadaju napadi koji mogu da iskoriste nedostatke kôdova za proveru određenih mrežnih parametara, a najčešci je napad prelivanjem bafera (*buffer overflow*). Iako se ovi napadi mogu sprečiti izmenom porogramskog kôda, autori u radovima [20, 21] predlažu rešenja kod kojih se detekcija napada izvršava van samih aplikacija. Napadi na nižim ravnima OSI modela koriste

nedostatke programskog kôda sigurnosnih mehanizama mrežnih protokola. Ovi napadi se izvršavaju prosleđivanjem specifičnih mrežnih paketa, ili izmenom komunikacije u okviru protokola, ali na način koji značajno odstupa od standardnih procedura. Napadi skeniranjem vrše skeniranje IP adresa, portova, protokola ili otkrivaju različite forme potencijalnih ranjivosti, bezbednosnih propusta ili problema u mrežnoj komunikaciji, koji bi se mogli iskoristi za napad. Napadi izviđanjem vrše prikupljanje poverljivih podataka o mrežnom okruženju ili o korisnicima. U napadima društvenog inženjeringa koriste se društvene mreže kako bi se došlo do poverljivih podataka o bezbednosnim akreditivima korisnika (autorizacija i autentifikacija). Napadi trajnog pristupa realizuju se tako što napadač koristi specifične procedure kojima nastoji da pristupi mreži ili delu mreže backdoor softverom, i da nakon toga izvrši određene maliciozne aktivnosti. Napadači često koriste dobro poznavanje klijent-server arhitekture, i preuzimanjem akreditiva korisnika ili maskiranjem prisustva neovlašćeno pristupaju zaštićenim delovima mreže ili poverljivim podacima.

Detaljniji prikaz i karakteristike prethodno navedenih tipova napada predstavljeni su u referentnoj literaturi [22-26].

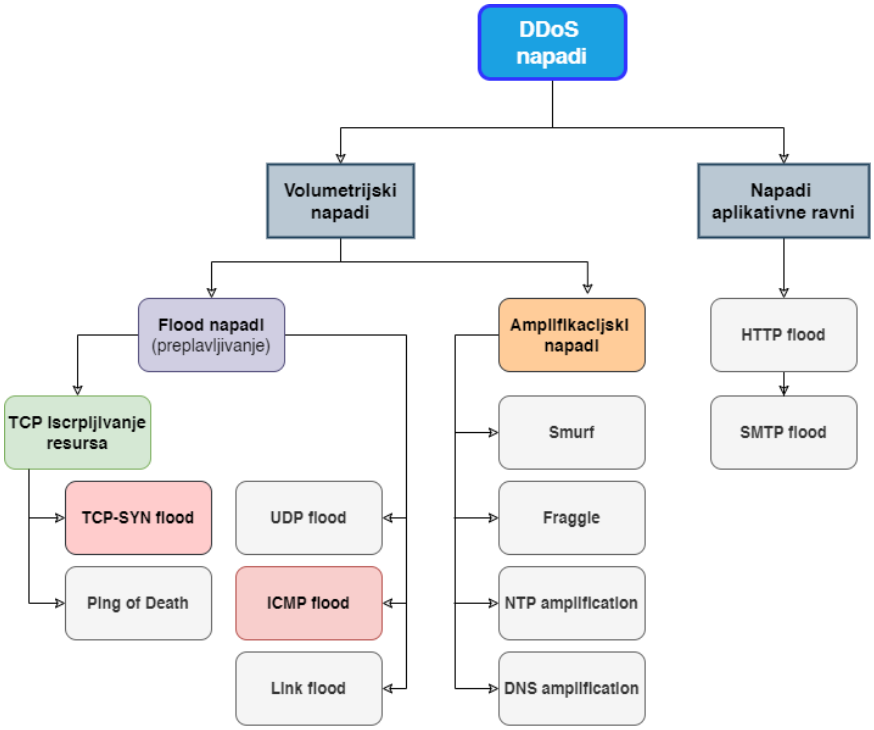
## 2.3. **DDoS napadi na računarske mreže**

Veliki broj malicioznih aktivnosti direktno utiče na dostupnost, tako da se legitimnim korisnicima ograničava ili ukida pristup mrežnim resursima ili servisima. Legitimni korisnici uglavnom poseduju određeni nivo ovlašćenja, pa se svaka aktivnost koja prelazi ta ovlašćenja smatra nelegitimnom. Ako ovakvi napadi potiču iz jednog izvora, tada se radi o DoS (*Denial of Service*) napadima. Predmet ovog istrživanja su distribuirani napadi odbijanjem servisa (DDoS), koji uglavnom potiču od velikog broja izvora. DDoS napadi koriste nedostatke funkcionisanja aktivnih mrežnih uređaja (svičevi, ruteri, serveri), komunikacionih linkova, protokola ili aplikacija. Privremeno ili trajno odbijanje servisa najčešće predstavlja sprečavanja legitimnog rada servera (web, e-mail, DNS, DHCP, database, file server…), koje se izvršava slanjem saobraćaja velikog intenziteta ili velikog broja zahteva za uspostavljenjem konekcije, koje ciljani server ili sistem ne mogu da podrže. Posledica slanja tako velike količine podataka dovodi do iscrpljivanja hardverskih resursa servera, prvenstveno memorije ili procesora. Pored iscrpljivanja hardverskih resursa, zahtevi za procesiranjem velike količine

podataka uzrokuju preopterećenje ili zagušenje pristupnih serverskih linkova, pa je pristup njihovim mrežnim servisima delimično ili potpuno prekinut [27].

DDoS napadi koriste postupke lažiranja (*spoofing*), jer se identiteti napadača ili neki njihovi mrežni parametri namerno skrivaju ili maskiraju slanjem velikog broja paketa sa geografski udaljenih ili dislociranih uređaja. Iako se napadi obično izvršavaju slanjem paketa podataka sa više računara, stvarni napadač ne mora biti uključen u napad. Agregacija mrežnog saobraćaja koji stiže sa velikog broja udaljenih računara stvara kritični protok, koji na kraju dovodi do prekida mrežnih servisa. Problem detekcije DDoS napada nije jednostavno rešiti i on je uslovljen načinima razdvajanja legitimnog i malicioznog saobraćaja. Izolovanje mrežnog saobraćaja koji je deo napada je složen postupak, jer se količina podataka koju šalje pojedinačni napadač najčešće neznatno razlikuje od intenziteta legitimnog saobraćaja. Osim toga, ovako formiran saobraćaj često sadrži legitimne zahteve za pojedinim mrežnim servisima na napadnutom uređaju, pa je postupak odbacivanja isključivo tog tipa paketa značajno otežan. Dodatnu problematiku predstavlja činjenica da DDoS napadi najčešće ne koriste posebno kreiran format sadržaja zaglavlja paketa. Napadi se obično izvršavaju slanjem paketa putem nadgledanja standardne komunikacije na određenoj računarskoj mreži.

U dostupnoj literaturi postoje različite klasifikacije DDoS napada, a sve u cilju boljeg razumevanja mehanizama njihovog izvršavanja i razvoja što efikasnijih rešenja zaštite. Sa razvojem tehnologije računarskih mreža, mehanizmi DDoS napada su se intenzivno menjali i prilagođavali novim tehnologijama, tako da njhova jedinstvena klasifikacija ne postoji. Značajno je istraživanje iz 2004. godine, u kojem autori predlažu jednu od prvih nespecifičnih klasifikacija DDoS napada, koja u obzir uzima nivo automatizacije napada, uticaj napada na ciljne sisteme i dinamiku stepena napada [28]. Istraživanja u okviru rada [29] predlažu klasifikaciju DDoS napada zasnovanu na smanjenju propusnog opsega komunikacionih linkova i iscrpljivanju mrežnih resursa. Autori u okviru rada [30] predlažu klasifikaciju DDoS napada na osnovu četiri faktora: nivoa automatizacije, dinamike napada, stepena ranjivosti i uticaja na ciljeve napada. Jedna od novijih klasifikacija, koju su autori predložili u radu [31] razlikuje četiri klase napada: napadi na ravan aplikacija, napadi zasnovani na protokolu, volumetrijski napadi i napadi male brzine izvšavanja. Na slici 2.1 prikazana je klasifikacija DDoS napada, formirana kao sveobuhvatna analiza mehanizama njihovog izvršavanja na osnovu većeg broja radova proučenih tokom ovog istraživanja.




**Slika 2.1:** Klasifikacija mehanizama izvršavanja DDoS napada

Volumetrijske (*volumetric*) napade karakteriše prisustvo ogromnog broja paketa u mreži. Napadači pokušavaju da iskoriste sav raspoloživi propusni opseg mreže ili hardverske resurse rutera, sviča ili servera preopterećujući ih intenzivnim malicioznim saobraćajem, tako da dolazi do smanjenja ili potpunog prekida legitimnog saobraćaja.

Napadi ravni aplikacija koriste slabosti ili softverske greške koje postoje u sistemu, protokolu ili mrežnim aplikacijama. Napadi najčešće pokušavaju da utiču na mrežne servise iscrpljivanjem procesora, memorije ili resursa za skladištenje na ciljnim serverima na kojima se izvršava mrežna aplikacija.

Napadi preplavljivanjem (*flooding*) su podgrupa DDoS napada kod kojih napadači generišu i šalju saobraćaj velikog intenziteta ka određenom uređaju, a u cilju značajnog smanjenja propusnog opsega pristupnih linkova, ili sprečavanja ispravnog rada računarske mreže ili nekog njenog dela. Napadi ove vrste razmatrani se u okviru ovog istraživanja i mehanizmi njihovog izvršavanja su detaljno opisani u narednom poglavlju.

Amplifikacijski (pojačani, reflektovani) DRDoS (*Distributed Reflected Denial of Service*) napadi koriste posredničke uređaje (reflektore) koji imaju ulogu da višestruko

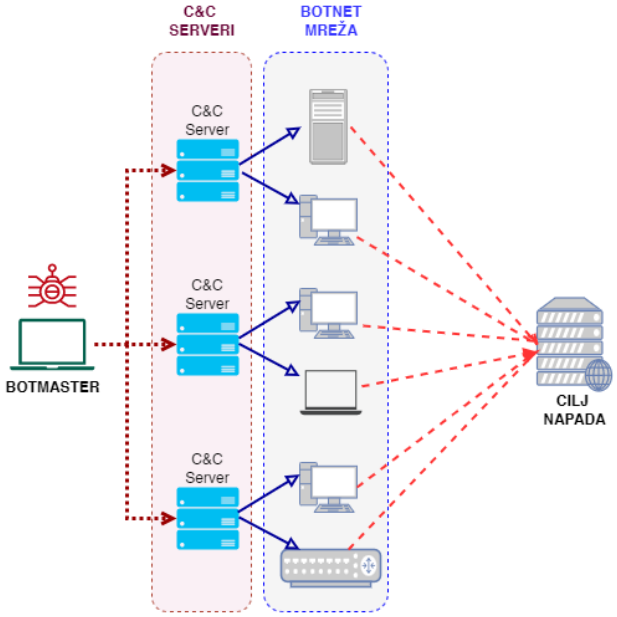
pojačavaju proopusni opseg i količinu generisanog saobraćaja ka cilju napada. Reflektori mogu biti ruteri ili serveri otvorenog tipa (DNS, Web ili NTP), koji na kratke mrežne upite šalju obimne odgovore. Agenti koji šalju upite, lažiraju IP adresu pošiljaoca, što rezultira da se takvi obimni odgovori preusmeravaju ka cilju napada. Ruteri kao reflektorski uređaji preko emisione (*bradcast*) IP adrese prosleđuju zahteve svim uređajima u delu mreže (*subnet*), na koje oni odgovoraju, smatrajući IP adresu u zahtevu izvornom. Lažiranjem izvorne IP adrese napadača postiže se prosleđivanje višestrukih odgovora ka cilju napada.

## 2.4. **Mehanizmi izvršavanja DDoS napada**

Jedna od najznačajnijih specifičnosti DDoS napada je njihovo izvršavanje mrežom daljinski dislociranih i kontrolisanih uređaja. Mreža takvih uređaja naziva se botnet (*botnet*), a uređaji unutar nje koji obično sadrže maliciozni kôd su bot ili zombi (*zombie*) uređaji. Maliciozni programski kôd ne nanosi štetu bot računaru, već mu je cilj da ostane potpuno neprimetan, tako da bot ne zna kada zaista učestvuje u DDoS napadu. Bot računari dejstvuju u sinhronizmu, dok napadača ostaje prikriven ili potpuno anoniman. Kao što je prikazano na slici 2.2, u procesu DDoS napada učestvuju sledeći mrežni elementi:

- C&C (*Command and Control*) ili kontrolni serveri (*handlers*)
- botovi ili agenti (*agents*)

DDoS napade kontroliše botmaster uređaj koji komunicira i upravlja botovima preko C&C servera kako bi se izvršio napad. Na C&C serverima se izvršavaju programske strukture koje definišu cilj, vreme, dužinu i druge parametre napada, koji se prosleđuju botovima. Na osnovu dobijenih instrukcija, botovi generišu ponavljajuće napade i preusmeravaju ih prema ciljanoj mreži, delu mreže ili sistemu. Različiti načini implementacije C&C servera opisani su u radovima [32, 33], pa se može zaključiti da oni kao posredni uređaji imaju vrlo značajnu ulogu u DDoS napadima. Realizuju se kroz nekoliko formi. Centralizovani C&C server koristi vezu sa botmasterom kako bi prosledio komande botovima. Karakteriše ga jednostavnost upravljanja, brz odgovor na mrežne zahteve, kao i otpornost na otkaze (*fault-resistance*) u jednooj tački, što je i njegov glavni nedostatak.

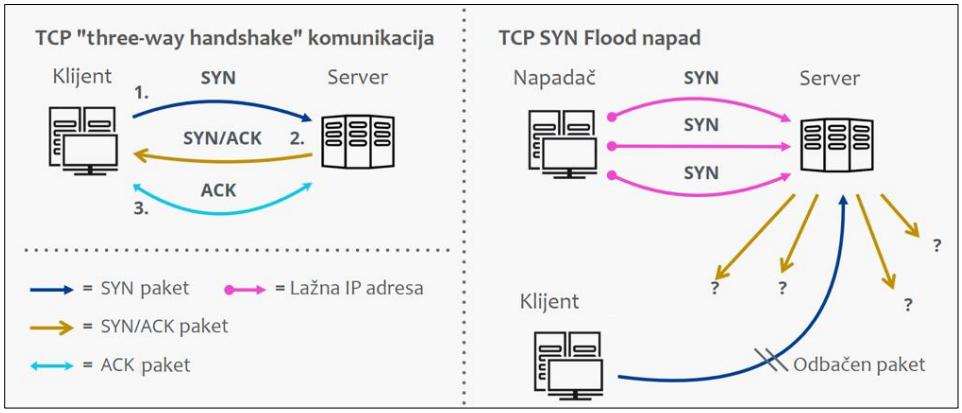



**Slika 2.2:** Prikaz DDoS botnet mreže i mehanizma napada

Decentralizovani C&C server koristi P2P (*Peer-to-Peer*) komunikacioni protokol, čime se postiže veća mrežna fleksibilnost pošto svaki čvor može da ima ulogu servera ili bota. Upravljanje ovom mrežom je složeno jer ne postoji centralni server preko kojeg bi se prosleđivale botmaster komande. IRC (*Internet Relay Chat*) je tekstualno zasnovan chat-sistem koji korisnicima omogućava komunikaciju sa više učesnika putem specifičnog kanala konverzacije. Omogućava povezivanje stotina korisnika preko višestrukih servera, pa napadač može da sakrije svoje prisustvo zahvaljujući velikom mrežnom saobraćaju koji ovi serveri imaju. O IRC softverskim platformama Trinity v3 i Kaiten više detalja može se pronaći u radu [34]. HTTP ili Web-botnet topologija koristi HTTP komunikacioni protokol za povezivanje botmastera sa C&C serverima i slanje komandi botovima, tako da svaki bot periodično šalje web zahteve. Web botovi se skrivaju unutar legitimnog saobraćaja, a kontrolišu se enkriptovanom komunikacijom preko HTTP (port 80) ili HTTPS (port 443) protokola. Porast broja uređaja i količine mrežnog saobraćaja zahteva linkove visoke propusnosti u delu jezgra komunikacione mrežne topologije. To omogućava da intenzitet mrežnog protoka prema krajnjem cilju napada može biti veći od 1Tb/s. Nedostatak autentifikacije kod nekih mrežnih servisa omogućava da DDoS napadi koriste lažne izvorne IP adrese, što značajno povećava anonimnost izvora napada. Za detekciju napada problem uglavnom predstavlja distribuirani način upravljanja i kontrole u internet mrežnoj infrastrukturi.

### 2.4.1. DDoS napadi preplavljivanjem

TCP/IP je glavni komunikacioni protokol računarskih mreža, i koristi princip "trostrukog rukovanja" (*three-way-handshake*) za formiranje sigurne isporuke mrežnih paketa od izvora do odredišta. Ovaj način komunikacije često se koristi za pokretanje DDoS napada preplavljivanjem, pri čemu se generiše intenzivan saobraćaj kako bi se onemogućio pristup mreži ili njenom delu. DDoS napadi se najčešće javljaju u formi TCP-SYN, UDP ili ICMP preplavljivanja. Na slici 2.3 prikazan je mehanizam izvršavanja TCP-SYN flood napada. Napad koristi ranjivost procesa uspostavljanja TCP veze i manipulaciju SYN polja u zaglavlju paketa. Napadač šalje ciljanom serveru veliki broj zahteva za formiranje TCP veze, koji u svom zaglavlju nose vrednost SYN polja i lažnu izvornu IP adresu. Osobina TCP konekcije je takva da server ima određeni broj linkova, istovremeno otvorenih na portu preko kojeg dolazi saobraćaj. Nakon prijema zahteva od napadača, server rezerviše potrebne resurse i uz otvorenu komunikaciju čeka na SYN-ACK poruku od klijenta, kojom će veza biti potvrđena. Server ne prima SYN-ACK poruku potvrde veze, što rezultira zadržavanjem rezervisanih resursa servera. Nakon što na server stigne veliki broj SYN poruka od napadača, on više neće imati slobodnih dodatnih resursa za formiranje nove veze, pa će svi novi klijenti biti odbijeni usled dostizanja maksimalnog broja mogućih veza. Dolazi do iscrpljivanja resursa servera, nakon čega se prelazi u stanje nedostupnosti, usled zasićenja propusnog opsega njegovog pristupnog linka.




**Slika 2.3:** Prikaz TCP komunikacije (levo) i mehanizma TCP-SYN napada (desno)

Za razliku od TCP-SYN napada, UDP (*User Datagram Protocol*) preplavljivanjem se putem botnet mreže šalje veliki broj paketa sa lažnim IP adresama ka jasno naznačenim ili na slučaj odabranim portovima ciljanog servera. Ovaj napad ima jednostavnii mehanizam izvršavanja, jer UDP protokol koristi samo četiri polja u steku zaglavlja, bez retransmisije i potvrde isporučenih paketa (*connectionless*), pa se često koristi u amplifikacijskim napadima. Nakon prijema UDP paketa, server će pokušati identifikaciju aplikacije odredišnog porta. Usled lažne izvorne IP adrese i neprepoznavanja aplikacije porta, server formira ICMP paket, koji predstavlja poruku o nedostupnosti porta. Usled velikog povećanja broja formiranih paketa, kao i prosleđenih poruka odgovora ka napadaču, ciljani server će biti prezasićen pristiglim brojem poruka, što će značajno uticati na dostupnost njegovih servisa i ispravno funkcionisanje.

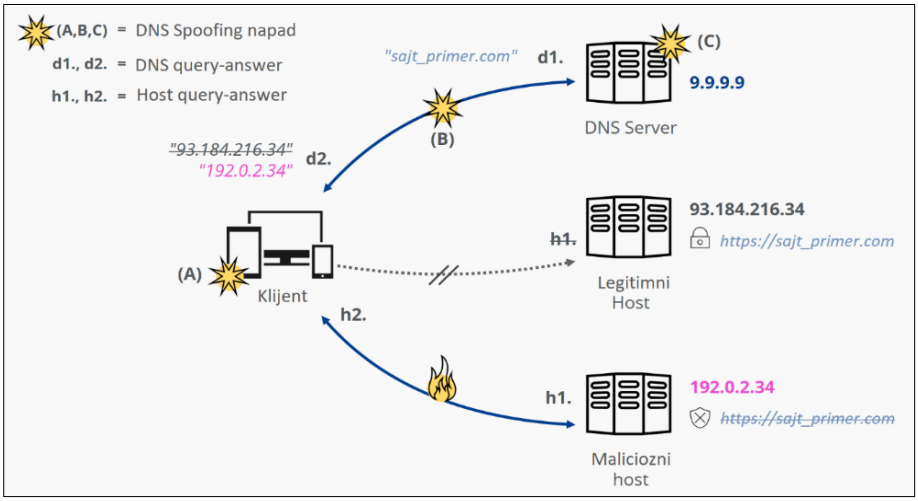
ICMP flood napad se izvršava tako što napadač preko botnet mreže ili reflektorskog uređaja šalje veliki broj *Echo* poruka putem dijagnostičkog ICMP (*Internet Control Message Protocol*) protokola. U ovom napadu se koristi *PING* komanda, kojom se od ciljanog uređaja zahteva da pošalje *ICMP Echo Reply* odgovor da je dostupan. Pošto napadač koristi lažnu izvornu IP adresu, napadnuti uređaj će slati jako veliki broj odgovora, što prevazilazi propusni opsega pristupnog linka. Kroz jako veliki broj *PING* poruka i njihove ciklične zahteve, napadač može iskoristiti ceo propusni opseg mreže samo za DDoS napad. O ovom mehanizmu napada više se govori u okviru rada [35].

Napad tipa “ping smrti” PoD (*Ping of Death*) predstavlja specifičan napad iscrpljivanja TCP resursa, kod kojeg napadač šalje ka ciljanom uređaju mrežne pakete koji prevazilaze njihovu maksimalno dozvoljenu veličinu. Definisana standardom, veličina IPv4 paketa sa zaglavljem iznosi 65535 B. Napad se najčešće izvršava slanjem *PING* paketa većih od 65535 B, mada se može inicirati bilo kojim protokolom koji koristi IP datagrame kao što je TCP, UDP ili stariji IPX (*Internetwork Packet Exchange*) protokol. Napadač vrši fragmentaciju poslatih paketa. Na strani napadnutog uređaja vrši se njihovo ponovno sastavljanje. Pošto ukupna veličina primljenog paketa premašuje dozvoljeni maksimum, dolazi do prelivanja bafera, restarta ili prekida rada napadnutog uređaja. Napad preplavljivanjem linka LFA (*Link Flooding Attack*) je vrsta DDoS napada kod kojeg napadač ima mogućnost da izoluje deo mreže, bez direktnog napada na ciljani uređaj. U prvoj fazi napadač pažljivo bira ciljano područje i servere-mamce (*decoy servers*) oko njega. Zatim se u drugoj fazi formira detaljna topologija sa svim putanjama od botova do ciljanih servera. U

poslednjoj fazi, napadač bira parove bot/server-mamac kako bi preko njih izolovao napadnutu oblast slanjem saobraćaja niskog intenziteta. O ovim, kao i PoD napadima vršena su detaljnija istraživanja u okviru radova [36, 37].

## 2.4.2. Amplifikacijski DDoS napadi

Amplifikacijski DDoS napadi koriste posredničke uređaje koji višestruko pojačavaju mrežni saobraćaj generisan od napadača. DNS (*Domain Name System*) serveri se često koriste kao reflektori za slanje velike količine poruka ka ciljanom uređaju. Osnovna uloga DNS servisa je da vrši mapiranje simboličkih imena domena u IP adrese. Napadač koristi mehanizam "krađe IP adrese" žrtve, uz istovremeno slanje zahteva pretrage (*lookup*) ka javnom DNS serveru. Odgovor DNS servera zavisi od opcija u zahtevu pretrage napadača. Nakon preuzimanja IP adrese žrtve od strane napadača, zahtev pretrage se istovremeno šalje na više javnih DNS servera. Napadnuti uređaj dobija odgovor u vidu velike količine podataka koja iscrpljuje propusni opseg njegovih pristupnih linkova. Na slici 2.4 prikazan je mehanizam izvršavanja DNS spoofing napada.




**Slika 2.4:** Mehanizam izvršavanja DNS napada

U prvom koraku (d1.) klijentski računar (pretraživač) šalje zahtev DNS serveru (IP=9.9.9.9) za mapiranje domena "*sajt_primer.com*" u odgovarajuću IP adresu. Klijent dobija odgovor od DNS servera (d2.), ali sa lažnom IP adresom, tako da se ne formira veza sa legitimnim serverom (IP=93.184.216.34) na kojem je hostovan "*sajt_primer.com*". U koraku (h1.) klijent šalje zahtev hostu sa lažnom IP adresom (IP=192.0.2.34), dok u koraku (h2,) lažni host vraća klijentu web stranicu koja izgleda kao legitimna, ali sa lažnim imenom domena kojem nedostaje bezbednosni sertifikat. (A), (B), i (C) su različite tačke napada, na strani klijenta (ili rutera), u okviru mrežne konekcije, i na strani DNS servera. Pojedini oblici amplifikacijskih napada funkcionišu dodatnim infiltriranjem napadača ili upotrebom softvera za takve aktivnosti. SNMP (*Simple Network Management Protocol*) je verzija UDP protokola koji se koristi za pristup i upravljanje štampačima, svičevima, ruterima ili firewall uređajima preko porta 161. Ovim napadom šalje se veliki broj zahteva sa lažnom izvornom IP adresom, na koju ovi uređaji odgovaraju velikim brojem odgovora koji uzrokuju preopterećenje propusnog opsega njihovih pristupnih linkova. U okviru istraživačkih radova [38, 39], izvršena je detaljna analiza poznatih napada tipa Smurf i Fraggle.

### 2.4.3. DDoS napadi ravni aplikacija

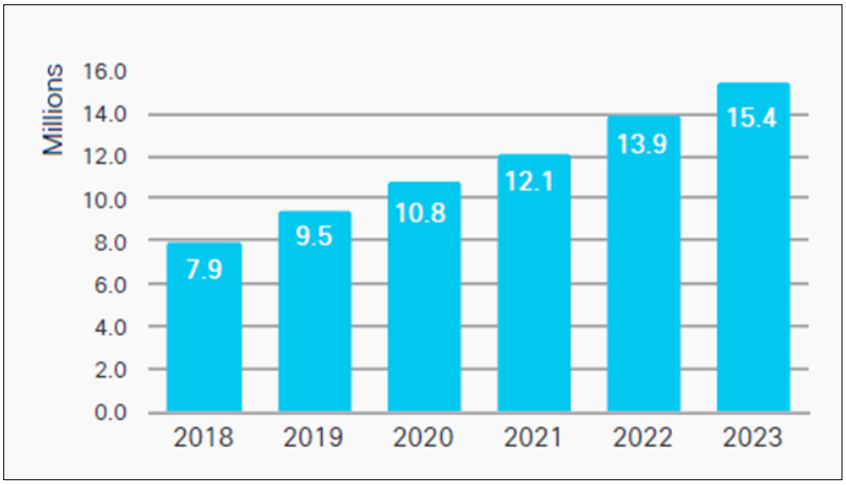
HTTP napadima preplavljivanja spečava se pristup web serveru slanjem velikog broja GET zahteva ka ciljanoj web stranici. Web server će nakon napada pokušati da odgovori na višestruke GET zahteve, pri čemu neće doći do obrade ACK paketa potvrde od strane napadača, što će web server dovesti u status čekanja. Web server održava vezu na čekanju dodeljujući joj fiksni vremenski period, a to čini za svaki zahtev napadača i svaku otvorenu vezu koja čeka potvrdu. Pošto napadač šalje mnogo zahteva ka serveru, on koristi sve raspoložive komunikacione resurse kako bi sve te zahteve obradio, što dovodi do njegove nedostupnosti za legitimne korisnike. SMTP (*Simple Mail Transfer Protocol*) flood napadi imaju sličan mehanizam izvršavanja kao i većina napada preplavljivanjem, jer slanjem velikog broja zahteva ka e-mail serverima sprečavaju pristup istim, stvarajući preopterećenje serverskih propusnih linkova.

DDoS napadi male brzine SRDDoS (*Slow Rate DDoS*) zasnivaju se na slanju kratkih tokova (strimova) podataka ka ciljnoj aplikaciji ili serveru. Za razliku od npr. napada grubom

silom (*brute-force*), ovi napadi ne zahtevaju velike hardverske resurse za izvršavanje. Pošto generišu saobraćaj vrlo malog intenziteta, teško ih je identifikovati kao maliciozne, i razdvojiti od legitimnog saobraćaja. Dok se standardni DDoS napadi mogu brzo detektovati, ovi napadi mogu ostati neotkriveni duži vremenski period, usporavajući ili odbijajući servise ciljanog uređaja. Varijante ovog napada su Slowloris i R.U.D.Y napadi. Ucenjivački RDDoS (*Ransom DDoS*) je trenutno vrlo aktuelan tip napada i evidentan je značajno povećan broj njihovih izvršavanja. Cilj napada je uglavnom pokušaj da se iznudi novac od pojedinca ili grupa korisnika. Napadač može da izvrši DDoS napad, a zatim da pošalje zahtev za plaćanje kako bi zaustavio napad, ili može prvo poslati poruku o zahtevu za plaćanjem u kojoj preti DDoS napadom [40].

## 2.5. **Statistika izvršavanja DDoS napada**

Intenzivnija istraživanja o DDoS napadima počela su pre dvadeset godina, i odmah se došlo da zaključka da će za postupke detekcije i zaštitie biti neophodna napredna i sveobuhvatna rešenja. Prvi detektovani DDoS napad dogodio se 1996. godine kada je internet provajder Panix nekoliko dana bio van funkcije usled pogođenosti TCP-SYN flood napadom. Vremenom su se DDoS napadi značajno menjali, kako po tehnikama izvršavanja, tako i po broju, a taj trend je vrlo intenzivan i danas.

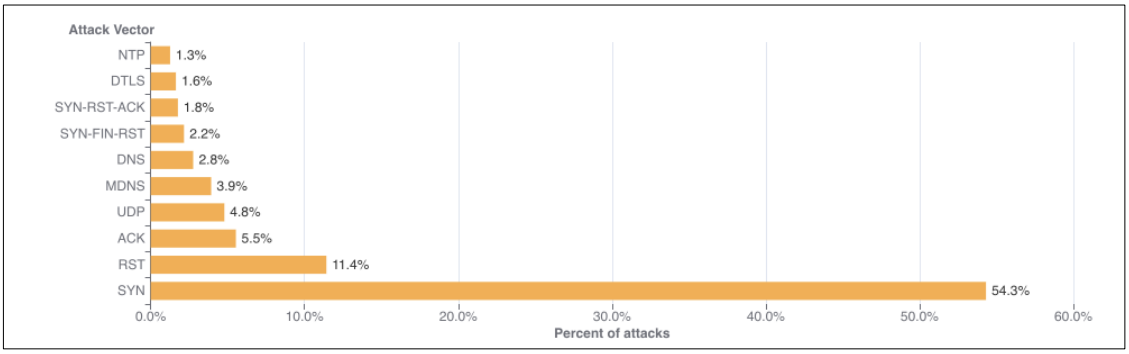



**Slika 2.5:** Hronološka statistika izvršenih DDoS napada (izvor: Cisco Systems [1])

Prema izveštaju kompanije Cisco Systems, predviđanja su da će se ukupan broj DDoS napada duplirati sa 7.9 miliona tokom 2018. godine, na nešto više od 15 miliona do 2023. Godine. Ova statistika je prikazana na slici 2.5.

Statistički podaci i analize o DDoS napadima pokazuju trendove značajnog rasta ne samo njihovog broja, već i botnet mreža koje generišu saobraćaj velikog protoka koji destruktivno utiče na ciljane sisteme. DDoS napad sa protokom 1 Gbps je dovoljan da čak i veliki internet provajderi budu van funkcije. Ovako veliki protoci pri DDoS napadima rezultat su povećanja propusnih opsega komunikacionih linkova u slojevima jezgra mreža, kao i intenzivnog razvoja cloud računarstva i naprednih provajderskih centara podataka. Zahvaljujući velikim botnet mrežama koje mogu da sadrže stotine hiljada, ili čak milione bot-uređaja, poslednjih godina su detektovani DDoS napadi sa intenzitetima mrežnog saobraćaja koji premašuju 1 Tbps.

Prema izveštaju kompanije Cloudflare za treće tromesečje 2021. godine, dominantan je bio TCP-SYN napad preplavljivanjem, koji predstavlja primer DDoS napada u ovom istraživanju. Ovaj tip napada imao je 54,3% učešća u ukupnom broju registrovanih mrežnih napada. Broj TCP RST/ACK napada preplavljivanjem je značajno povećan u odnosu na period iz 2020 god. RST flood napadi izvršeni su u 11,4% slučajeva, a dalje slede ACK flood napadi sa oko 5.5%, UDP flood napadi sa 4.8% itd. Evidentno je da je najveći broj registrovanih DDoS napada tokom 2021. godine bio zasnovan na mehanizmima preplavljivanja, što govori o važnosti istraživanja i pronalaženju efikasnih rešenja za detekciju ovih napada. Statistika DDoS napada za treći kvartal 2021. god. prikazana je na slici 2.6.




**Slika 2.6:** Statistika DDoS napada za Q3 2021. godine (izvor: Cloudflare [41])

Najveći registrovani DDoS napad dogodio se 2020. godine na kompaniju Google, kada je detektovan UDP amplifikacijski napad od strane tri kineska internet provajdera (ASN 4134, 4837, 58453 i 9394). Tom prilikom napadnuto je na hiljade Google IP adresa, napad je trajao šest meseci i dostigao maksimalni mrežni protok od 2.54 Tbps. Napadač je koristio nekoliko mreža za lažiranje 167 Mpps na 180.000 dostupnih CLDAP, DNS i SMTP servera, koji su zatim slali ogroman broj odgovora, čime su potpuno ili delimično izbačeni iz funkcije

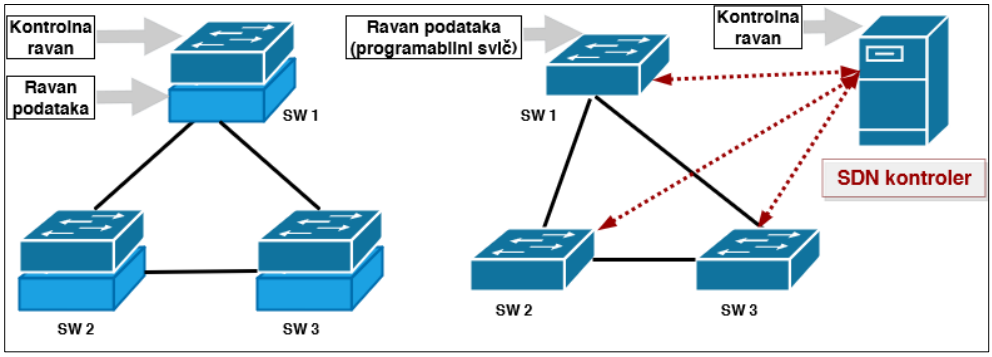
Mirai botnet napad iz 2016. godine pokazao je svu snagu i sofisticiranost mehanizama DDoS napada. Mirai napad je postao značajan nakon napada na DNS servere provajdera Dyn i francuskog hosting provajdera OVH, što je uticalo na to da web stranice pojedinih velikih svetskih kompanija (Twitter, PayPal, HBO, Amazon, GitHub) postanu nedostupne. Ovi napadi su izvršeni generisanim mrežnim protokom od 1 Tbps, i u tom trenutku to su bili najveći ikad zabeleženi DDoS napadi. Veliki rast popularnosti Mirai je doživeo kada je javno objavljen njegov izvorni kôd 2016 godine, što je napadačima omogućilo da prilagode napade i stvore nove sofisticiranije verzije ovog napada.

Cloud mreža kompanije Amazon AWS (*Amazon Web Services*), bila je cilj velikog DDoS napada u februaru 2020. Godine. To je bio najveći napad koji je ciljao na neidentifikovanog AWS korisnika koristeći tehniku CLDAP refleksije (*Connectionless Lightweight Directory Access Protocol*). Ovaj način napada zasniva se na ranjivim CLDAP serverima treće strane i povećava količinu podataka poslatih na IP adresu žrtve u opsegu od 50-70 puta. Ovo je bio trodnevni napad, koji je dostigao rekordnih 2.3 Tbps. Iako je poremećaj Amazonovih AWS servisa uzrokovan ovim napadom bio daleko manji nego što se očekivalo, sam opseg napada i implikacije na AWS hosting klijente bili su vrlo značajni.

# 3. SOFTVERSKI DEFINISANE MREŽE

Današnje računarske mreže zahtevaju primenu efikasnih, primenljivih i fleksibilnih rešenja za vrlo raznolike i kompleksne mrežne probleme. Činjenica je da se mrežni hardver značajno sporije razvija od softvera, i obično su potrebne godine da bi se novi standard usvojio i postao primenljiv. U literaturi postoji veći broj radova u kojima se pokušalo pronaći efikasnije rešenje za pojedine mrežne probleme, kao što su npr. rutiranje, problem mobilnosti, povećanje kvaliteta servisa QoS (*Quality of Services*), bezbednosti itd. Većina tih rešenja su često bolja od postojećih, ali je retko koje definisano kao standard i primenjeno u praksi. Razlozi su uglavnom sveprisutnost tradicionalnih standarda, diskutabilna kompatibilnost sa postojećim rešenjima ili problem interoperabilnosti hardvera različitih proizvođača. Arhitektura TCP/IP mreža nije dizajnirana tako da može da upravlja složenim mrežama sa velikim brojem uređaja koji zahtevaju konstantno održavanje. Najvažnija ograničenja standardnih TCP/IP mreža su: kompleksna statična arhitektura, nedoslednost korisničkih naloga i polisa, problem skalabilnosti, nedostatak standarda i zavisnost od dobavljača opreme i softvera. Zbog navedenih ograničenja, istraživanja su se fokusirala na softversko upravljanje mrežama i njihovu programabilnost. Ideja o softverski programabilnim mrežama nije nova. Dokument kompanije AT&T iz 1987. godine predstavljao je prvi opis SDN koncepta [42]. Arhitektura ravni (slojeva) SDN mreže prvi put je prikazana u radu [43], u kojem je opisano razdvajanje kontrolne ravni od ravni podataka. Pregledni rad [44] smatra se značajnim, jer su autori sistematično predstavili važnost koncepta SDN mreža i OpenFlow protokola, kao i glavne istraživačke pravce njihovog daljeg razvoja.

Protokoli koji su donekle koristili principe SDN mreža bili su SNMP (*Simple Network Management Protocol*), Telnet, SSH (*Secure Shell*) i Netconf, koji su se putem softverskih agenata ili kolekcija konfiguracionih skriptova koristili za daljinski pristup ili podešavanje mrežnih uređaja. Iako postoji više različitih definicija SDN mreže, njenu najopštiju formulaciju dao je konzorcijum mrežnih operatera ONF (*Open Network Foundation*), koji kaže da je to "arhitektura koja razdvaja kontrolnu ravan mreže od ravni za prosleđivanja podataka, i u kojoj kontrolna ravan upravlja sa nekoliko uređaja" [45].




**Slika 3.1**: Arhitektura tradicionalne i ekvivaalentne SDN mreže

Na slici 3.1 prikazana je osnovna razlika u arhitekturi tradicionalne TCP/IP i SDN mreže. Nasuprot TCP/IP mrežama, kod kojih je svaki mrežni uređaj nezavisan kontrolni entitet, SDN mrežni koncept koristi centralizovanu kontrolnu ravan preko koje komunicira sa ostalim mrežnim uređajima, i koja određuje pravila prosleđivanja podataka u okviru mreže. Ulogu kontrolne ravni preuzima centralizovani namenski server ili SDN kontroler, koji definiše „inteligenciju“ mreže. SDN koristi mrežni hardver jedino u segmentu koji treba da omogući značajne mrežne performanse. Centralizovanost kontrolera i mogućnost programabilnosti ovih mreža, omogućava efikasnija i fleksibilnija rešenja, kao i nove načine dizajna mreža i implementacije mrežnih servisa.

Najvažnija karakteristika SDN arhitekture jeste razdvajanje kontrolne ravni od ravni podataka. Logička centralizovanost SDN kontrolera ima nekoliko prednosti, a pre svega se ogleda u mogućnosti praćenja stanja mreže na globalnom nivou, što olakšava razvoj i implementaciju naprednih mrežnih funkcija, servisa i aplikacija. Automatizovano ažuriranje mrežnih pravila SDN mreže manje je podložno greškama, u poređenju sa tradicionalnim konfigurisanjem mrežnih uređaja. Programabilnošću visokog nivoa, SDN mreža koristi logički mrežni model koji se od njene fizičke implementacije može značajno razlikovati, ali koji znatno olakšava njenu funkcionalnost i upravljanje. Jedan od osnovnih mrežnih postulata – nezavisnost softvera od hardvera, omogućen je kombinovanjem SDN i NFV (*Network Functions Virtualization*) virtuelizovanih mrežnih funkcija. Zahvaljujući programabilnosti i definisanju izolovanih NFV mreža kroz kontrolnu ravan, SDN arhitektura pruža mogućnosti za razvoj i implementaciju potpuno nove mrežne infrastrukture neophodne za funkcionisanje dinamičkih mrežnih servisa. SDN osobina programabilnosti značajno olakšava mrežnu

automatizaciju i upravljanje uz smanjenje troškova održavanja, dok mogućnost brže reakcije na različite probleme znatno poboljšava dostupnost mrežnih servisa [46].

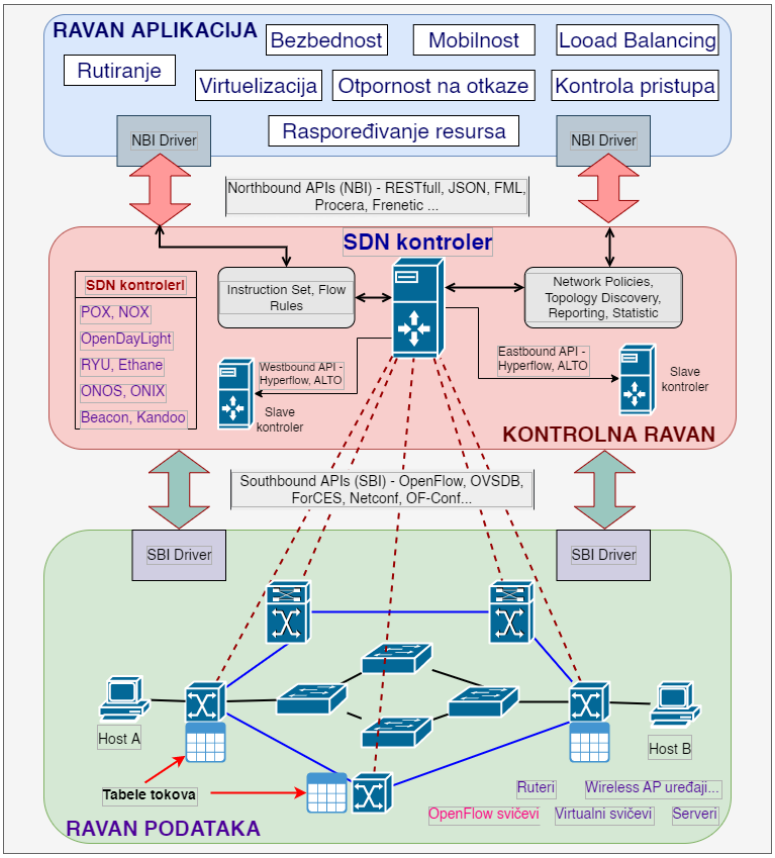
SDN mrežna arhitektura koristi tri apstrakcije: prosleđivanje, distribuiranje i specifikaciju. Apstrakcija prosleđivanja treba da omogući da je prenos podataka od strane aplikacija ili kontrolera potpuno nezavisan od mrežnog hardvera. OpenFlow protokol je praktična realizacija te apstrakcije, i on je ekvivalent „drajvera mrežnog uređaja“ u operativnom sistemu kontrolera NOS (*Network Operating System*). Za realizaciju apstrakcije distribuiranja odgovorna je ravan distribuiranja, koja predstavlja deo NOS softvera, i koji ima dve funkcije. Prva je instaliranje kontrolnih komandi u SDN svičeve, dok je druga prikupljanje informacije o globalnom statusu mreže, a koje se mogu dobiti iz ravni podataka. Apstrakcija specifikacije omogućava apstrakciju mrežnih aplikacija. SDN mreže ovu funkcionalnost realizuju kroz virtuelizaciju i programabilnost, mapiranjem apstraktne konfiguracije definisane od strane aplikacije u oblik pojednostavljenog mrežnog modela.

Glavni cilj SDN mrežne arhitekture je omogućiti cloud servisima, provajderima, mrežnim inženjerima i administratorima brzu reakciju na bilo kakvu promenu mrežnih zahteva putem centralizovanog upravljačkog kontrolera. SDN implementira različite mrežne tehnologije, kako bi se što efikasnije podržala virtuelizovana serverska rešenja i zadovoljile potrebe velikih propusnih opsega savremenih centara podataka. SDN arhitektura definiše postupke dizajna, kreiranja i upravljanja mrežama, tako da se razdvoji upravljanje mrežom od prosleđivanje podataka u okviru nje. Na taj način mrežno upravljanje je postalo programabilno i automatizovano, pa se SDN infrastruktura može proširiti za aplikacije velike propusnosti i efikasno primeniti u novim mrežnim tehnologijama kao što su cloud računarstvo i mobilne mreže.

U početnom delu ovog istraživanja, proučen je značajan broj radova koji se bave tematikom SDN mreža, ali se izdvajaju [47-49] u kojima se na sistematičan i detaljan način izlaže pregled trenutnog stanja u ovoj oblasti. U okviru istraživanja [50] su razmatrane prednosti SDN mreža u odnosu na standardne. Kada je reč o komercijalnim implementacijama SDN mreža, za potrebe ovog istraživanja proučena je mreža pod nazivom Orion, razvijena od strane kompanije Google. Orion je distribuirano rešenje SDN kontrolne ravni zasnovano na mikro-servisima, koje se koristi za povezivanja udaljenih centara podataka, i koje omogućava skalabilnost, servise vrlo velikih mrežnih protoka, korelaciju sa postojećim mrežama, i unapređeno koordinisano upravljanje mrežom centara podataka [51].

## 3.1. SDN referentni mrežni model

ONF je definsao SDN referentni mrežni model kao strukturu koja se sastoji iz tri ravni: ravni podataka, kontrolne ravni i ravni aplikacija. Na slici 3.2 je prikazan SDN mrežni model, gde su pored ravni prikazane i njihove osnovne funkcionalnosti.




**Slika 3.2:** SDN referentni mrežni model

Ravan podataka ili infrastrukturna ravan sadrži uređaje čija je primarna funkcija prosleđivanje podataka. Svičevi, ruteri, serveri, bežični AP (*access point*) ili drugi gateway uređaji su međusobno povezani (žično ili bežično) formirajući SDN mrežu. Uloga ovih uređaja je prosleđivanje i izvođenje raznih operacija nad paketima podataka. Sadrže tabelu tokova FT (*Flow Table*) sa uputstvima šta treba uraditi sa određenim paketom, menjaju ili prosleđuju pakete, ažuriraju tabele tokova ili komuniciraju sa kontrolerom. Svoj rad zasnivaju na protočnom procesiranju paketa (*pipeline processing*) i prikupljaju statističke podatke o mrežnom saobraćaju. Svičevi su osnovni mrežni uređaji ravni podataka. Mogu biti

komercijalnog tipa, ali su uglavnom zasnovani na OpenFlow protokolu. Mogu imati formu GbE (*Gigabit Ethernet*) svičeva za mala preduzeća, ili se koriste kao 100GbE svičevi u velikim centrima podataka. Tabele tokova čuvaju se u internoj adresibilnoj TCAM (*Ternary Content-Addressable Memory*) memoriji, koja može imati relativno mali kapacitet (do 8000 tokova), pa do preko $10^6$ tokova u 100GbE svičevima za velike centre podataka. Načini upravljanja TCAM memorijom svičeva nije predmet ovog istraživanja, ali je opisan u većem broju stručnih radova. U zavisnosti od hardverskih specifikacija, SDN implementacija svičeva može biti: standardnim PC hardverom (Pantou, OpenFlowClick), mrežnim hardverom otvorenog tipa (NetFPGA, ATCA) ili su to komercijalni svičevi (NEC PF5240, Pica8 3920 itd). Detaljan i analitičan pregled SDN svičeva opisan je u radovima [52,53].

Primarna uloga kontrolne ravni je softversko upravljanje tokovima podataka, kako bi se implementirale različite procedure kojima se može na efikasniji način upravljati mrežom. SDN kontroler donosi odluke na osnovu globalnog statusa mreže, a ne na osnovu statusa mrežnih uređaja. SDN kontrolerom upravlja NOS operativni sistem, koji takođe popunjava i održava tabele rutiranja RIB (*Routing Information Base*), ali i vrši konfigurisanje ravni podataka, kroz pravila za prosleđivanje paketa. Mrežne funkcije se realizuju softverski na strani kontrolera, a primenjuju se preko svičeva prevođenjem u instrukcije tokova. Izmena sadržaja mrežnih paketa se ne vrši od strane kontrolera, već je to uloga sviča, koji pored izmene vrši i prosleđivanje paketa. U kontrolnoj ravni može postojati više kontrolera koji se organizuju u formu master-slave topologije. U oblasti SDN mreža, danas se koristi veći broj različitih kontrolera, kako komercijalnih, tako i verzija otvorenog programskog kôda. Za razvoj rešenja za detekciju DDoS napada predstavljenog u ovom radu, upotrebljen je POX kontroler sa nešto manjim brojem funkcionalnosti u odnosu na neke druge, ali sa jednostavnijim API-em [54]. Najzastupljeniji kontroleri koji se koriste u savremenim mrežnim okruženjima su NOX, Ryu, FloodLight, OpenDayLight, ONOS, ODL itd. Autori su u referentnim radovima [55, 56] istraživali njihove karakteristike i različite načine primene.

SDN API (*Application Programming Interface*) interfejsi omogućavaju visok nivo programabilnosti i automatizacije mreže. Koriste programske jezike višeg nivoa, nezavisni su od mrežnog hardvera i obezbeđuju punu interoperabilnost SDN uređaja. U kontrolnoj ravni definisani su API interfejsi: SBI (*Southbound Interface*) za komunikaciji svič-kontroler, NBI (*Northbound Interface*) za komunikaciju kontroler-aplikacija, i *EastBound/WestBound* za internu komunikaciju kod višekontrolerskih SDN topologija.

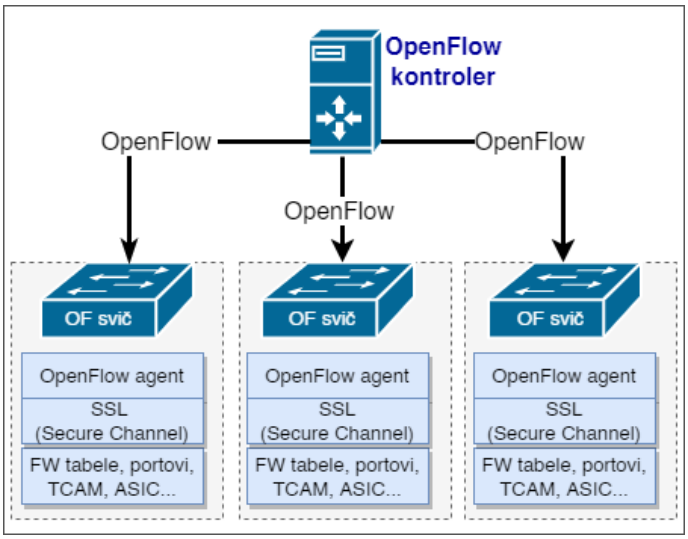
NBI interfejs obezbeđuje komunikaciju između ravni aplikacija i kontrolera. Svaka mrežna aplikacija komunicira preko App Logic modula i NBI drajvera, formirajući na taj način viši nivo apstrahovane mrežne kontrole. NBI predstavlja nestandardizovanu apstrakciju SDN ekosistema, tako da svaki kontrolor definiše sopstveni komunikacioni API. Savremene SDN mreže uglavnom koriste open-source interfejse tipa REST/RESTful ili JSON, ili programske jezike FML, Frenetic i Procera, kao i specijalizovane API interfejse za ad hoc mreže. Iako je REST API trenutno najviše u upotrebi, ONF konzorcijum ne isključuje mogućnost standardizacije, i predlaže razvoj novih API modula otvorenog kôda. Autori u [57] daju vrlo iscrpan prikaz aktuelnih NBI API interfejsa. SBI interfejs se koristi u komunikaciji između svičeva, i u okviru njega implementirani su protokoli koji su definisani standardima. Najvažniji SBI protokol je OpenFlow, a detaljnije je opisan u narednoj sekciji. U literaturi se navodi da se kao SBI interfejsi, pored OpenFlow protokola, uglavnom koriste OVSDB, ForCES, OpFlex i NETCONF protokoli [58-61].

Ravan aplikacija SDN referentnog modela definiše načine na koji aplikacije realizuju mrežne funkcije. Aplikacije su apstrakcije mrežnih funkcija i izvršavaju se programski i tehnikom virtuelizacije. Ravan aplikacija obezbeđuje PaaS (*Platform-as-a-Service*) servis, koji je osnova razvojnog modela cloud računarstva. Savremene SDN mreže imaju širok spektar mrežnih aplikacija, koje se koriste za kontrolu upravljanja, rutiranje, bezbednost mreže, virtuelizaciju, balansiranje opterećenja, raspoređivanje mrežnih resursa, kontrolu mobilnosti itd. SDN aplikacije se uglavnom koriste kao sopstvena razvojna rešenja ili su to često third-party rešenja [62].

## 3.2. **SDN mrežna komunikacija**

OpenFlow protokol je prvobitno imao ulogu L2 i L3 protokola u komercijalnim svičevima i ruterima, ali je sa razvojem SDN mreža implementiran kao primarni SBI komunikacioni interfejs. ONF asocijacija je prvu zvaničnu OpenFlow specifikaciju objavila 2009. godine, a tekuća verzija 1.5.1 objavljena je 2016. godine [63]. OpenFlow je standardizovan komunikacioni protokol SDN uređaja i može se ugraditi u mrežni hardver različitih proizvođača.

SDN svičevi svoje funkcije zasnivaju na komunikaciji sa kontrolerom i obradi tabele tokova. Tabela tokova sviča se sastoji od: zaglavlja sa kriterijumima za izbor paketa (*packet matching*), brojača o statistici za svaki izabrani paket i akcije koja označava uputstvo šta treba uraditi sa selektovanim paketom. OpenFlow svičevi preko OFA agenata (*OpenFlow Agent*) komuniciraju sa kontrolerom putem zaštićene komunikacije (*secure channel*). Pravila prosleđivanja paketa se definišu vrednostima polja zaglavlja na L2, L3 i L4 slojevima, kao i uputstvima šta je potrebno uraditi sa svim pristiglim paketima kod kojih su vrednosti u tabeli tokova jednake vrednostima stvarnih polja u zaglavljima. Akcija podrazumeva prosleđivanje paketa na izlazni port, na sve portove sviča, odbacivanje paketa ili prosleđivanje paketa kontroleru. Jednom instrukcijom moguće je realizovati višekriterijumski izbor, ili koristiti vrednosti za više različitih zaglavlja istovremeno, pa je za realizovanje instrukcije neophodno da svi kriterijumi budu zadovoljeni. Nakon dobijanja novog pravila prosleđivanja, OFA agent proverava da li je tabela tokova sviča popunjena. Ako nije, novo pravilo se upisuje u nju, a ako jeste, pravilo se odbacuje i kontroleru se šalje poruka o grešci. Svičevi mogu lokalno da iz tabele tokova brišu postojeća pravila prosleđivanja, kako bi se stvorio prostor za unos novih. Hardverske komponente sviča ASIC (*Application-Specific Integrated Circuit*) procesor i TCAM memorija se koriste za čuvanje sadržaja tabela tokova. Slika 3.3 prikazuje detaljniju OpenFlow komunikaciju između kontrolera i sviceva.




**Slika 3.3:** Komunikacija između OpenFlow kontrolera i svičeva

U savremenim SDN mrežama u upotrebi su dva načina implementacije OpenFlow protokola. Prvi način koristi namenski hardver i softver realizovan kao OpenFlow svič, i koji koristi komunikaciju sa kontrolerom za sve aktivnosti. Drugim načinom se OpenFlow protokol implementira kao dodatna funkcionalnost postojeće mrežne opreme koja poseduje određen hardver i softver. SDN mrežni koncept podrazumeva maksimalnu redukciju funkcija L2 svičeva TCP/IP mrežnog modela. Za razliku ood klasičnih svičeva koji koriste MAC (*Media Access Control*) tabele i postupke njihovog "učenja" (*learning*), tabela tokova OpenFlow sviča se menja u skladu sa instrukcijama kontrolera. Na taj način je npr. moguće koristiti hardverske FPGA svičeve visokih performansi i velikih propusnih opsega, kao što je opisano u istraživanju [64].

OpenFlow protokolom (ver. 1.5.1) definisano je preko 30 tipova poruka koje se razmenjuju između kontrolera i svičeva, ali su najznačajnije sledeće:

- (Flow_Mod poruke): Koriste se za izmenu statusa sviča. Pravila definisana ovim porukama podrazumevaju parametre prioriteta i vremena važenja, kako bi se dodatno uticalo na procedure njihove primene u svičevima
- (Packet_In poruke): Definišu poruke koje svičevi prosleđuju kontrolerima, kao i način njihovog slanja, bilo putem precizno definisanog pravila, ili za opciju kada ne postoji pravilo koje se ne poklapa sa poljima u zaglavljima mrežnih paketa
- (Packet_Out poruke): Poruke kojima kontroleri šalju procesirane pakete prema svičevima. Najčešće se koriste kako bi se Packet_In poruke pristigle na kontroler dalje prosledile prema ravni podataka nakon obrade
- (Port_Status poruke): Predstavljaju asinhrone poruke koje svičevi šalju kontrolerima kako bi signalizirali promenu stanja nekog od svojih portova

Bezbedna komunikacija između OpenFlow sviča i SDN kontrolera predstavlja osnovu ispravnog funkcionisanja OpenFlow uređaja u mreži. Za komunikaciju OpenFlow svič-kontroler koristi se zaštićena TLS komunikacija, kako bi se garantovala visoka bezbednost prenosa podataka, što je detaljnije opisano u narednom poglavlju. Maksimalna bezbednost TLS komunikacije omogućena je tako da i kontroler i svič poseduju svoj par RSA kriptografskih ključeva, pa se tokom uspostavljanja veze između njih razmenjuju sertifikati radi postizanja faktora autentifikacije. Za TCP i UDP komunikaciju, OpenFlow protokol koristi port 6653, kako je navedeno u RFC dokumentu za ovaj standard [65].

# 4. BEZBEDNOST SOFTVERSKI DEFINISANIH MREŽA

Budući da je istraživanje u okviru ove disertacije usmereno na detekciju anomalija mrežnog saobraćaja koje su nastale kao rezultat različitih napada, u ovom poglavlju objašnjene su posledice napada na funkcionisanje SDN mreže, kao i mehanizmi izvršavanja DDoS napada na njene različite ravni. Kompleksnost i raznovrsnost napada predstavljeni su taksonomijama, dok je posebna pažnja posvećena mehanizmu DDoS napada na kontroler, koji predstavlja najčešći tip maliciozne aktivnosti u aktuelnim SDN mrežama.

## 4.1. Karakteristike napada na SDN mreže

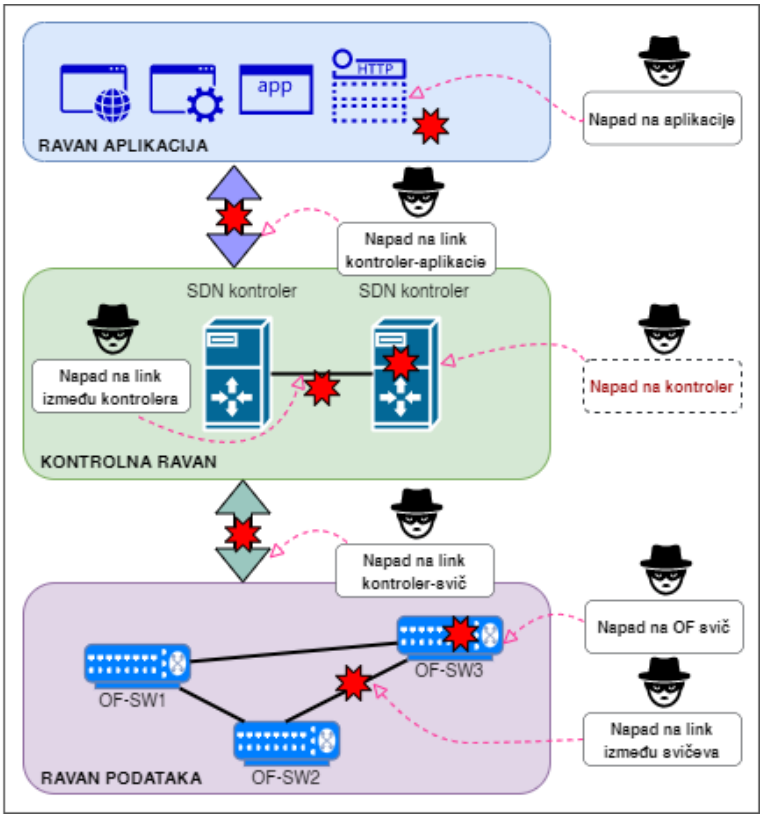
Fundamentalna karakteristika SDN arhitekture tj. razdvojenost kontrolne ravni od ravni podataka predstavlja sa gledišta bezbednosti dodatni istraživački izazov. SDN mreže omogućavaju implementacije funkcija bezbednosti na drugačiji način u odnosu na tradicionalne mreže, što stvara potrebu za drugačijim mehanizmima osiguravanja (*network hardening*) ovih mreža. U dostupnoj literaturi naglašava se važnost sigurnosnih analiza SDN mreža, i predlažu nova rešenja, sa posebnim akcentom koji je usmeren ka sigurnosnim aspektima OpenFlow protokola. Istraživanje u okviru rada [66] daje pregled ranjivosti uzrokovane razdvajanjem kontrolne ravni od ravni podataka SDN mreže.

ONF konzorcijum navodi nekoliko ključnih faktora bezbednosti SDN arhitekture [67]:

- postojanje bezbednosnih rizika u okviru centralizovane SDN kontrole mreže
- postojanje dvostranih efekata programabilnosti SDN mreža
- kompleksnost SDN integracije tradicionalnih mrežnih protokola (NAT, DNS, BGP), bez prethodne analize kompatibilnosti bezbednosnih aspekata
- postojanje relacija nepoverenja međusobno povezanih mrežnih domena

Napadači mogu iskoristiti ove bezbedonosne faktore pokretanjem različitih napada. Poreklo ovih napada ne mora nužno biti od strane spoljašnjeg izvora, kao u slučaju DDoS napada realizovanih botnet mrežom. Postoji čitava grupa napada koji mogu iskoristiti ranjivosti slojevite SDN arhitekture. Ranjive tačke SDN mreže mogu biti komunikacioni linkovi (naročito SBI interfejs) preko kojih napadač može pristupiti kontroleru i izvršiti napad, čime se funkcionalnost mreže izlaže riziku. Napadi se mogu izvršiti na ravan podataka, pri čemu su ciljevi napada OpenFlow svičevi ili linkovi između njih. Takođe, napadi se mogu izvršiti na ravan aplikacija, ili na NBI interfejs. Ipak, najčešći tip napada se realizuje kao napad na kontrolnu ravan, pri čemu se želi ciljano uticati na kontroler i njegove funkcionalnosti.

Razvoj novih mrežnih tehnologija i povećanje brzine komunikacionih linkova u SDN mrežnim okruženjima uzrokuje nove tipove napada i načine njihovog izvršavanja. Evidentne su različite kategorizacije napada na SDN mreže, mada se u referentnoj literaturi one uglavnom svode na podelu prema ravnima mreže u kojima se ti napadi izvršavaju. Na slici 4.1 prikazana je lokalizacija napada u okviru SDN mrežnog referntnog modela.




**Slika 4.1:** Lokalizacija napada u SDN mrežnom modelu

SDN mreže poseduju izvesne specifičnosti koje mogu biti izazov za potencijalne napadače i predstavljati ozbiljan bezbednosni problem. Prva specifičnost je softver za upravljanje mrežom, koji je podložan programskim greškama ili ranjivostima. Sledeća specifičnost je centralizovanost kontrolerske logike, jer napadač koji ima pristup serverima ili NOS sistemu kontrolera, može preuzeti kontrolu nad celom mrežom. Programabilnost i softver otvorenog kôda daju napadačima mogućnost reprogramiranja mreže, iako omogućavaju razvoj najefikasnijih procedura mrežnog upravljanja. Kontroler prikuplja podatke samo sa podređenih uređaja, pa je analiza tih podataka jednostavnija nego kod tradicionalnih mreža. Za detekciju anomalija, kontroler može da koristi kolaborativnu detekciju, što je za tradicionalne mreže teže primenljivo rešenje. Važan segment istraživanja mrežne bezbednosti predstavlja način analize podataka. U radu [68] autori su izvršili poređenje forenzičkih postupaka analize podataka u tradicionalnoj i SDN mreži. Prikupljanje podataka u realnom vremenu i njhova analiza se izvode jednostavnije u SDN mrežama, jer kontroler prikuplja statističke podatke o tokovima podataka sa svičeva, dok se u TCP/IP mrežama vrši na osnovu analize zaglavlja paketa podataka. Pored protokola rutiranja, u kontroleru se najčešće istovremeno nalazi i softver za obradu podataka, što značajno pojednostavljuje mrežnu analizu.

U ravni podataka dominantni su napadi na OpenFlow svičeve, ili na linkove između njih. Napadi na svičeve se fokusiraju na tabele tokova ili njihova hardverska ograničenja, dok napadi na linkove narušavaju prenos nezaštićenih podataka, naročito kod bežičnih mreža. Najveći broj napada dešava se u okviru kontrolnoj ravni, pa su mogući napadi na kontroler ili na linkove između njih. U distribuiranim mrežnim topologijama sa više kontrolera, sve češće se javljaju napadi usmereni na East/WestBound interfejse. Napadi na ravan aplikacija često koriste sigurnosne propuste ili neautorizovan pristup third-party softveru, kako bi se preko NBI interfejsa zlonamerni kôd ubacio u kontroler i narušile njegove funkcije. Značajna istraživanja u kojima su definisane kategorizacije, principi analize podataka, kao i sistematični pregledi različitih mehanizama napada na SDN mreže publikovana su u radovima [69-72]. U ovim radovima opisane su najvažnije kategorije rešenja za detekciju napada, kao i savremene strategije za njihovo ublažavanje (*attack mitigation*). Na osnovu analize pomenutih radova, izvedena je klasifikacija napada na SDN mreže, prikazana u tabeli 4.1. Pored prikaza ravni u kojima se izvršavaju, dat je kraći opis ovih napada, uz navođenje najvažnijih mehanizama koje oni koriste kako bi se realizovali.

**Tabela 4.1:** Klasifikacija napada na SDN mreže

| SDN RAVAN | TIP NAPADA | OPIS NAPADA |
|---|---|---|
| RAVAN PODATAKA | Lažni tokovi podataka | Podložnost uređaja u ravni podataka lažnim tokovima podataka |
| | Flooding napadi | Tabele tokova i memorijski kapaciteti OpenFlow svičeva su ograničeni |
| | Preuzimanje ili kompromitovanje kontrolera | Bezbednost ravni podataka isključivo zavisi od bezbednosti kontrolera |
| | "Man-in-the-middle" napadi | TLS kao opcija i složenost TLS konfigurisanja |
| KONTROLNA RAVAN | DDoS napadi | Centralizovano upravljanje i hardverska ograničenja kontrolera, "vidljivost" kontrolera |
| | Neovlašćen pristup kontroleru | Nepostojanje efikasnih mehanizama kontrole pristupa aplikacijama koje komuniciraju sa kontrolerom |
| | Skalabilnost i dostupnost | Centralizovan kontroler kao samostalan entitet uzrokuje probleme skalabilnosti i dostupnosti |
| RAVAN APLIKACIJA | Autorizacija i autentifikacija | Nepostojanje adekvatnih mehanizama autentifikacije i autorizacije za standardne i third-party aplikacije |
| | Unos lažnih pravila mrežnih tokova | Maliciozne aplikacije koje je teško proveriti unose lažna pravila mrežnih tokova |
| | Kontrola pristupa | Kompleksne procedure za podešavanja kontrole pristupa za third-parti aplikacije |

Napade na ravan podataka karakterišu određene specifičnosti. OpenFlow sviče ima ograničen memorijski prostor za čuvanje tabele tokova i instrukcija dobijenih od kontrolera. Pošto on nema ulogu donošenja odluka o prosleđivanju saobraćaja, bezbednosni izazov predstavlja njegovo pravovremeno prepoznavanje lažnih ili malicioznih tokova. Lažni tokovi podataka generišu određenu vrstu mrežnog saobraćaja koji je unutar mreže nepoželjan. Svičevi takođe imaju ograničen broj unosa novih tokova koji se čuvaju u njegovoj bafer memoriji. Usled toga su svičevi skloni napadima saturacije bafera.

OpenFlow protokol ne poseduje sopstvene bezbednosne mehanizme, pa se zaštićena komunikacija između svičeva i kontrolera opciono može uspostaviti preko TLS (*Transport Layer Security*) kriptografskog protokola. ONF preporučuje upotrebu TLS od verzije 1.2 [73]. Autori u [74] navode složenost konfigurisanja TLS protokola koja, podrazumeva generisanje site-wide sertifikata, sertifikata kontrolera i svičeva, privatnih ključeva i postupaka razmene ključeva između uređaja. Sve su to razlozi zbog kojih je na mnogim OpenFlow svičevima TLS protokol opciona funkcija, što je bitan faktor bezbednosti SDN mreže.

Usmeravanjem i kontrolisanjem mrežnih tokova, napadač može izvršiti osluškivanje mreže ili "man-in-the-middle" napad. Autori u radu [75] navode da je stepen destruktivnosti "man-in-the-middle" napada u OpenFlow mrežama mnogo veći u odnosu na tradicionalne mreže, zbog stalne povezanosti i nedostatka autentifikacije TCP kontrolnog linka.

Bezbednost SDN kontrolora predstavlja izazov iz perspektive njegove sposobnosti da autentifikuje aplikacije i autorizuje resurse koje aplikacije koriste. Zbog toga je neophodna autentifikacija aplikacija pre njihovog pristupa mreži i njenim resursima, a na NBI interfejsu implementiranje prilagođenih bezbednosnih mehanizama. Međutim, bezbednosne procedure koje se implementiraju prema tipu aplikacija u dostupnoj literaturi skoro da i ne postoje. Ako se od kontrolera zahteva da čuva pravila tokova za celu mrežnu topologiju, onda on lako može postati "usko grlo". Autori su u [76] zaključili da najveći broj implementacija kontrolera u brzim 10 Gbps mrežama nije u stanju da procesira ogroman broj tokova složenih SDN topologija. Nedostatak skalabilnosti omogućava ciljanim napadima da izazovu preopterećenje kontrolne ravni, koje destuktivno utiče na funkcionisanje mreže. Pošto brzina procesiranja tokova u kontroleru zavisi od njegovih hardverskih resursa, centralizovan kontroler kao samostalan entitet postaje jedinstvena tačka otkaza (*single point of failure*) u slučajevima DDoS napada. Iako se kao logično rešenje nameće upotreba topologije distribuiranih kontrolera, u istraživanju [77] je pokazano da to nije uvek dobro rešenje usled pojave kaskadnih kvarova kontrolera, koji nastaju zbog njihovog neadekvatnog balansiranja mrežnog opterećenja. Bezbednosni problemi mogu da nastanu usled konfiguracionih neusaglašavanja u topologijama sa više kontrolera, što rezultuje pojavom potencijalnih inter-federativnih konflikata [78]. U slučaju promene mrežnog statusa, obaveštenje o tome neće dobiti svi kontroleri istovremeno. Zbog toga pojedine aplikacije (npr. firewall), mogu imati nepredvidivo ponašanje usled nesinhronizovanosti ažuriranja informacija sa više kontrolera.

Pošto se većina SDN mrežnih funkcija može implementirati kroz aplikacije, postojanje i pristup malicioznih aplikacija mreži ili njenim resursima mogu biti vrlo destruktivni. Raznolikost i mnoštvo third-party aplikacija kreiranih u različitim razvojnim okruženjima i programskim modelima, dovodi do ograničenja u interoperabilnosti i kolizija u bezbednosnim strategijama. Provera autentičnosti aplikacija u SDN mrežama predstavlja jedan od većih izazova. Najveći broj aplikacija koje se izvršavaju na kontrolerima sadrži većinu funkcija kontrolne ravni. Međutim, aplikacije i softver kontrolera uglavnom ne potiču od istog izvora ili su otvorenog kôda, tako da aplikacije nasleđuju kontrole pristupa mrežnim resursima bez odgovarajućih sigurnosnih mehanizama zaštite od zlonamernih aktivnosti. Autori su u radu [79] zaključili da nema efikasnih mehanizama za uspostavljanje odnosa poverenja između SDN kontrolera i aplikacija. Zlonamerna aplikacija može biti izuzetno destruktivna, jer kontroler preko nje definiše apstrakcije koje se prevode u konfiguracione mrežne komande. U

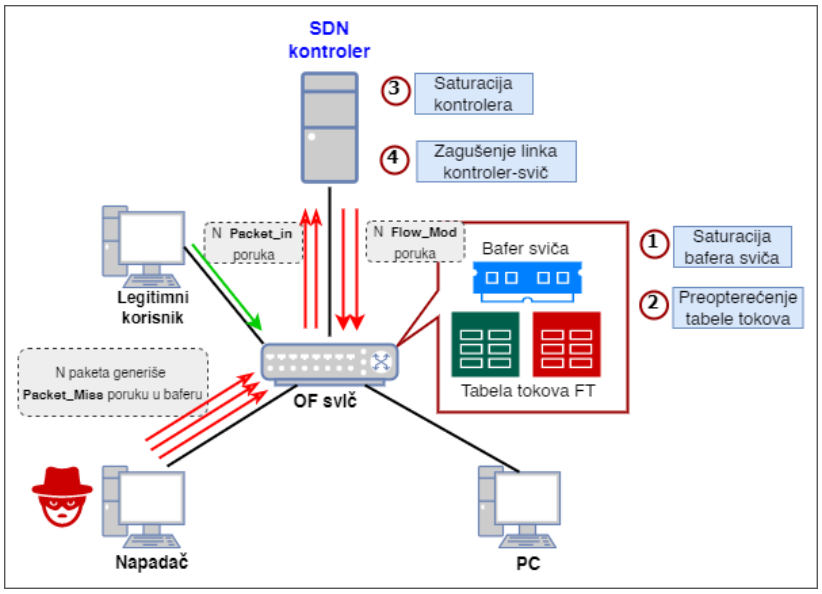
slučaju ugroženosti servera aplikacija na kojem se čuvaju korisnički podaci, akreditivi legitimnih korisnika mogu se iskoristiti za unošenje ovlašćenih, ali lažnih mrežnih tokova. Postoje načini za sertifikovanje mrežnih uređaja, ali još uvek nisu dostupni mehanizmi za sertifikovanje mrežnih aplikacija. Pošto je mrežna funkcionalnost implementirana preko aplikacija, centralizovani sistem za sertifikaciju SDN aplikacija je neophodan, ali takvo rešenje u referentnoj literaturi još uvek nije dostupno.

## 4.2. **Mehanizmi DDoS napada na SDN mreže**

DDoS napadi predstavljaju jedan od najčešćih oblika malicioznih aktivnosti koje utiču na performanse i ponašanje SDN mreže. Kada govorimo o ovim napadima, činjenica je da je kontroler najčešća tačka njihove aktivnosti. Ako napadač ima pristup kontroleru, može preuzeti kontrolu nad mrežnim uređajima preko kojih dalje može pokrenuti potencijalne napade ka ostalim ravnima mreže. OpenFlow svičevi imaju drugačije mehanizme upravljanja podacima u odnosu na standardne L2 svičeve, pa je uticaj DDoS napada na njih specifičan. Svaki paket se u OpenFlow sviču procesira na osnovu tabele tokova, a zatim se šalje na određeni izlazni port. Procesiranje podrazumeva poređenje pristiglih paketa sa jasno definisanim pravilima prosleđivanja. SDN uređaji koriste proaktivni ili reaktivni način rada. Kod proaktivnog načina, kontroler prvo razlaže mrežne polise u pravila tokova, a zatim ih instalira u svičeve tokom faze uspostavljanja mreže (*bootstrap*). U reaktivnom modu, kontroler definiše i instalira pravila prosleđivanja samo onda kada mu svič pošalje zahtev za njihovo ažuriranje. Ovaj mod omogućava svičevima da se brže prilagode dinamici mreže, tako da ne moraju da održavaju velike tabele tokova. DDoS napadi na SDN mreže koriste hardverska ograničenja procesora ili TCAM memorije svičeva ili kontrolera, ali i osobinu sporog procesiranja paketa, kao posledice nedostatka OpenFlow protokola.

Posledice DDoS napada na SDN mreže su saturacije bafera sviča, preopterećenje tabele tokova, saturacija kontrolera i zagušenje linka kontroler-svič. Na slici 4.2 prikazan je uticaj DDoS napada preplavljivanjem na SDN mrežu. Napadač pristupa portu sviča preko računara ili botnet mreže i generiše napad slanjem *N* paketa. U tabeli tokova sviča desiće se *N* "promašaja" (*Table_Miss*), jer svič neće uspeti da procesira sve pristigle pakete. Svič će genetisati *N Packet_In* poruka koje će proslediti kontroleru. Svaka *Packet_In* poruka sadrži

informacije o dolaznom paketu (zaglavlje, buffer_id, in_port, korisni podaci itd…). Kao odgovor, kontroler će sviču poslati *N Flow_Mod* poruku. Svaka *Flow_Mod* poruka sadrži instrukciju prosleđivanja paketa, koja se upisuje u TCAM memoriju sviča. Nakon prijema *Flow_Mod* poruke, svič proverava da li postoji dovoljno memorijskog prostora za upis novog pravila. Ako ima, paket se dalje prosleđuje na osnovu novog pravila, a u suprotnom svič odbacuje paket i obaveštava kontroler o prepunjenosti memorije slanjem *Table_Full* poruke.




**Slika 4.2:** Uticaj DDoS napada preplavljivanjem na SDN mrežu

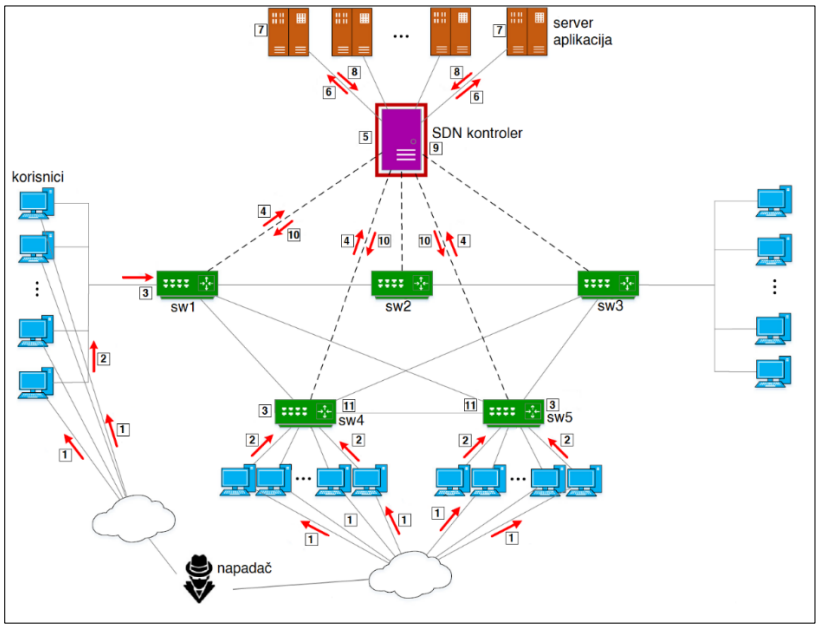
1) (Saturacija bafera sviča) TCAM memorija sviča ima ograničen kapacitet, tako da može da čuva određeni broj mrežnih tokova, najčešće od 1500 do 3000. U kataloškim podacima za H3C S5820V2 svič, navedeno je da može čuvati do 3000 tokova u TCAM memoriji, tako da napadač može relativno lako da iscrpi memoriju DDoS napadom, što je analizirano istraživanjem [80]. Kada ne postoji poklapanje polja zaglavlja dolaznog paketa sa pravilom prosleđivanja u TCAM, svič će sačuvati deo paketa u svom baferu, dok će zaglavlje paketa proslediti kontroleru. Kada je bafer pun, svič će celokupan paket proslediti kontroleru kao *Packet_In* poruku. DDoS napadom se šalje veliki broj paketa koji se ne poklapaju sa pravilima prosleđivanja, tako da svič kontroleru prosleđuje veliki broj celih *Packet_In* poruka. Hardversko ograničenje bafera sviča dovodi do njegove saturacije. U uputstvima [81, 82] se mogu pronaći podaci da svič serije HP 6600 ima mogućnost generisanja do 1000 *Packet_In* poruka, dok serija Cisco Nexus 3000 svičeva može generisati do 5000 ovih poruka.

2) (Preopterećenje tabele tokova sviča) U slučaju nepoklapanja sa pravilom u tabeli tokova, svič šalje *Packet_In* poruku kontroleru kako bi dobio novu instrukciju o prosleđivanju. Svako novo pravilo toka je vremenski ograničeno, i nakon određenog vremena biće zamenjeno novim. Napadač to može iskoristiti i generisati lažne pakete, na koje će kontroler reagovati slanjem novih zahteva kontroleru. Novi unosi tokova zameniće stare, tako da će sva pravila tokova biti zamenjena, pa će se tabela tokova sviča popuniti lažnim pravilima prosleđivanja. Mrežni tokovi legitimnog saobraćaja biće odbačeni, usled popunjenosti tabele tokova sviča. U literaturi se navodi nekoliko rešenja za problem preopterećenja tabele tokova. Autori u radu [83] koriste dinamički in/out algoritam balansiranja pod imenom DIOB/LFLU. Algoritam održava razliku između rule-in i rule-out pravila tokova. Razlika između njih se održava tako da bude manja ili jednaka nuli. U slučaju poruke o popunjenosti tabele tokova, algoritam povlači pojedine unose iz tabele koji su istekli, i brojač se postavlja na nulu. Na taj način algoritam pokušava da održi ravnotežu novih unosa u tabeli. Nedostatak rešenja je taj što je brzina pristiglih paketa tokom napada mnogo veća od brzine izvršavanja algoritma balansiranja, pa je predloženi algoritam u praksi teško primenljiv.

3) (Saturacija kontrolera) Tokom DDoS napada preplavljivanjem, kada veliki broj *Packet_In* poruka stigne od sviča u kontroler, kontroler će procesirati veliki broj lažnih zahteva, pa će doći do iscrpljivanja njegovih resursa i dalje nemogućnosti obrade mrežnih paketa. Ovo može da dovede do degradacije celokupne SDN mreže. Problem preopterećenja hardverskih resursa kontrolera i njegov uticaj na brzinu detekcije anomalija je tema prvog dela ovog istraživanja.

4) (Zagušenje linka kontroler-svič) Slanje velikog broja celih *Packet_In* poruka dovodi do značajnog smanjenja propusnog opsega linka kontroler-svič. Određeni broj rešenja ovog problema zasniva se na ograničenju propusnog opsega ovog linka. U radu [84] razvijen je mehanizam ograničenja propusnog opsega pod imenom FlowSec. Preko OpenFlow sviča meri se propusni opseg i broj prenetih paketa kroz link kontroler-svič. Ako brzina prenosa pređe definisani prag, paket se odbacuje. Ovim mehanizmom filtrira se i legitimni mrežni saobraćaj, što je bitan nedostatak. Nešto drugačije rešenje problema zagušenja linka pod imenom Flowfence, predloženo je u okviru rada [85]. Prema ovom rešenju, svičevi prate mrežne aktivnosti na svojim portovima, i u slučaju zagušenja obaveštavaju kontroler. Kontrolor na osnovu prikupljene statistike tokova sa svih mrežnih svičeva šalje komandu ograničenja protoka tačno određenom sviču. Karakteristika ovog rešenja je što vrši ograničenje propusnog opsega linka, ali nema mogućnost potpunog sprečavanja DDoS napada.

Slika 4.3 prikazuje mehanizam izvršavanja DDoS napada na kontroler, dok su u tabeli 4.2 označene faze napada i ukratko opisane aktivnosti napada u svakoj od njih.




**Slika 4.3:** Mehanizam DDoS napada na SDN kontroler

**Tabela 4.2:** Faze mehanizma DDoS napada na SDN kontroler

| Faza napada | Aktivnost napada |
|---|---|
| ① | Napadač šalje komandne instrukcije bot-računarima koji šalju vrlo veliki broj paketa vremenski ograničenog zaglavlja, ka što većem broju svičeva u mreži, umesto ka jednom ciljnom sviču |
| ② | Bot-računari šalju DDoS pakete ka svičevima u mreži |
| ③ | Napadnuti svičevi analiziraju pristigle DDoS pakete, i pretražuju tabele tokova kako bi pronašli poklapanje sa pravilom toka dobijenim sa kontrolera |
| ④ | Zbog vremenskog ograničenja zaglavlja primljenih DDoS paketa, svič ne može pronaći poklapanja u tabeli tokova, enkapsulira takve pakete i kao neodgovarajuće ih prosleđuje kontroleru u vidu Packet_In poruka |
| ⑤ i ⑥ | Kontroler analizira Packet_In poruke, izdvaja neodgovarajuće pakete i prosleđuje ih serveru aplikacija |
| ⑦ i ⑧ | Server aplikacija odlučuje kako će procesirati neodgovarajuće pakete i odluku o tome šalje kontroleru |
| ⑨ i ⑩ | Kontroler enkapsulira jednu ili više Flow_Mod poruka, na osnovu odluka servera aplikacija, i te poruke prosleđuje sviču |
| ⑪ | Svič dodaje jedan ili više novih unosa pravila tokova na osnovu Flow_Mod poruka sa kontrolera |

# 5. PREGLED POSTOJEĆIH ISTRAŽIVANJA

U ovom poglavlju predstavljena su određena rešenja koja se odnose na problematiku detekcije DDoS napada u SDN mrežama. Kako bi se postigli ciljevi istraživanja, pročitan je i analiziran veći broj naučno-istraživačkih radova. Aktuelnost teme uzrokovala je brojna istraživanja, pa je fokus značajnog broja novijih radova usmeren ka problematici primene mašinskog učenja za klasifikaciju mrežnog saobraćaja. Važno je napomenuti da se sa povećanjem broja i načina izvršavanja DDoS napada, javlja sve veći broj radova u kojima se predlažu hibridna, ili rešenja koja su spoj različitih tehnika detekcije, a određeni broj radova i istraživanja fokusiran je na tehnike detekcije DDoS napada koje su kombinovani metod entropije i principa mašinskog učenja.

Jedan od doprinosa ovog istraživanja jeste rešavanje problema preopterećenja SDN kontrolera i smanjenje kašnjenja procesiranja mrežnih paketa. U dostupnim izvorima, određeni broj radova bavi se ovom problematikom. Sa značajnim razvojem brzih hardverskih komponenti poslednjih godina, sve je veći broj rešenja koja koriste softverski algoritme realizovane na hardverskim platformama, pa se eksperimenti vrše direktno na programabilnim FPGA (*Field-Programmable Gate Array*) komponentama. U istraživanju [86] je predložena hibridna programabilna DPPSN (*Deeply Programmable Packet-Switching Node*) arhitektura, zasnovana na FPGA tehnici, kojom se prevazilazi problem ograničenja OpenFlow SDN svičeva, implementacijom novih protokola i unapređenih funkcija procesiranja mrežnih paketa. Na taj način rešava se osnovni problem ograničenja OpenFlow svičeva koji nastaje zbog spore pretrage u njihovim tabelama prosleđivanja. U radu [87] je predloženo FPGA rešenje primenljivo za detekciju napada u mrežama centara podataka sa velikim propusnim opsezima. Rešenje je moguće implementirati na glavni mrežni tok, pa se može primeniti za detektovanje nekoliko tipova anomalija.

U okviru israživanja [88], autori su koristili kolaborativnu tehniku za detekciju i sprečavanje DDoS napada preplavljivanjem u SDN mrežama. Rešenje integriše sflow-RT aplikaciju i Snort softver kako bi se detektovali napadi na višestruke SDN kontrolere. RSMQ (*Redis Simple Message Queue*) mehanizam je upotrebljen za deljenje pravila detekcije DDoS

napada između višestrukih RYU kontrolera. Rezultati simulacije pokazuju da kolaborativno rešenje primenjeno na višestruke kontrolere daje bolje rezultate detekcije DDoS napada, u poređenju sa samostalnim sistemom detekcije zasnovanim na sflow-RT ili Snort softveru. Predloženi metod efikasno smanjuje propusni opseg SDN mreže i preopterećenje kontrolera tokom DDoS detekcije.

U sistematičnom radu [89], napadi su analizirani kroz mehanizme njihovog izvršavanja, pa je njihova klasifikacija izvršena prema dinamici napada i kriterijumu stepena automatizacije. Ovaj rad je bitan za problem detekcije, jer naglašava da se napadi sa promenljivom brzinom izvršavanja mogu značajno prilagoditi, kako bi zaobišli mehanizme detekcije i to smanjenjem intenziteta napada u precizno definisanom vremenskom intervalu.

Autori u okviru istraživanja opisanog u [90] predlažu model za detekciju DDoS napada u SDN mrežama realizovan u dva nivoa. U okviru prvog nivoa implementiran je mehanizam rane detekcije napada zasnovan na entropiji, koji analizira broj mrežnih paketa i IP adresa. U drugom nivou za dodatnih šest atributa primenjeno je mašinsko učenje sa C4.5 stablom odlučivanja, kako bi se povećala tačnost detekcije napada.

U istraživanju [91], upoređeni su rezultati dva postupka detekcije DDoS napada. Prvi koristi princip detekcije Šenonovom entropijom, dok drugi koristi Tsallis entropiju kao metod detekcije. Za generisanje mrežnog saobraćaja upotrebljen je ns-2 mrežni simulator. CUSUM proračun kumulativnih suma je upotrebljen za detekciju početka i kraja napada, koji je dalje primenjen na vremenski signal entropije. Određivanjem procenta ispravnih/lažnih detekcija, kao i vremena kašnjenja promene statusa detekcije, izvršena je evaluacija performansi ovih postupaka DDoS detekcije. Zaključeno je da Tsallis detektor pokazuje značajno bolje karakteristike detekcije DDoS napada u odnosu na Šenonov detektor.

U okviru vrlo značajnog rada za ovo istraživanje [92], predložena je metoda detekcije anomalija i napada koja koristi proračun entropije nad određenim atributima tokova mrežnog saobraćaja. Prvo su instance saobraćaja grupisane prema već definisanim profilima, a zatim je nad njima primenjen algoritam hijerarhijskog aglomerativnog klasterovanja HAC (*Hierarchical Agglomerative Clustering*). Izvršeno je opisivanje svake mrežne komunikacije preko pridruživanja klasterima normalnog saobraćaja, što je upotrebljeno za identifikovanje, a zatim i detekciju anomalija. Postupak komunikacionih potpisa je dalje upotrebljen za identifikovanje mrežnih anomalija i postizanje preciznosti detekcije. Za mašinsko učenje upotrebljen je princip sličnih klastera, pri čemu su, u zavisnosti od unapred određenog praga

udaljenosti, definisani komunikacioni profili za određene tokove podataka. Na osnovu izvedenih rezultata, zaključeno je da je predložena metoda efikasna za poređenje mrežnih tokova sa unapred utvrđenim profilima, a da je generisanim potpisima pojedinačnih profila, moguće izvršiti efikasno detektovanje novih mrežnih profila.

U istraživanju koje je opisano u [93], izvršeno je upoređivanje rezultata za dve metode detekcije DDoS napada. Jedna metoda koristi princip Šenonove entropije, dok je druga konfigurisana tako da efikasno prepoznaje TCP-SYN flooding napade. I u ovom slučaju je primenjen proračun kumulativnih suma za detekciju početka i kraja napada, ali primenjen na vremenski signal entropije. Određivanjem broja ispravnih i lažnih detekcija, kao i kašnjenja u promene statusa detekcije, eksperimentalno je pokazano da metoda zasnovana na Šenonovoj entropiji beleži lošije karakteristike od metode prilagođene određenom tipu napada. Zaključak je da detekcija napada zasnovana na entropiji daje korektne rezultate i za druge tipove napada, za čiju bi detekciju trebalo formirati sasvim drugačiji detektor.

U istraživanju [94], za detekciju DDoS napada upotrebljen je princip fazi-logike. Predloženo je rešenje za detekcije napada se sastoji iz dve faze. U prvom delu vrši se klasični postupak proračuna entropije mrežnih paketa, dok se u drugom delu na već izračunate vrednosti entropije primenjuje TSK-FS (*Takagi-Sugeno-Kang Fuzzy System*) sistem. Rezultati su pokazali da TSK-FS detekcija napada daje veoma nizak procenat lažnih detekcija, uz značajno povećanje osetljivosti i robusnosti. U okviru rada predložena je implementacija optimizovane verzije TSK-FS detektora u realnom vremenu, koja bi se realizovala sa hardverskim komponentama i mogla praktično primeniti na postojećoj mrežnoj opremi.

U preglednom radu [95] autori su predstavili najznačajnija rešenja za detekciju DDoS napada koja su implementirana u SDN cloud scenarijima. Posebna pažnja povećena je analizi SDN i cloud mrežne arhitekture. Osim toga, dat je pregled istraživačkih radova i glavnih problema vezanih za identifikaciju i suzbijanje DDoS napada u ovim okruženjima.

U radu [96] je prezentovan novi NB-PSDP model koji omogućava decentralizovano procesiranje i razmenu podataka putem nezavisnog kanala između svičeva i kontrolera, kako bi se rešili problemi preopterećenja i pada performansi složene SDN mrežne topologije. Model koristi metodu izbora odgovarajućih atributa kako bi se smanjio njihov broj tokom izdvajanja, i minimizovao prenos podataka kroz kanale. Pored toga, kao efikasan i brz klasifikator, upotrebljen je Naïve Bayes algoritam mašinskog učenja. Predloženi framework je implementiran putem Mininet emulatora, kojim je formirano odgovarajuće SDN mrežno

okruženje. Eksperimentalni rezultati su pokazali da sistem postiže tačnost detekcije različitih napada od 98,46%, uz visoku pouzdanost i integritet u okviru kompleksnih SDN mreža.

Autori u svom istraživanju [97] predlažu novu metodu za detekciju DDoS napada, primenljivu u SDN mrežama. U svrhu detekcije, višeslojni algoritam potpornih vektora (SVM) je korišćen kao klasifikator. Za veću tačnost i smanjenje trajanja faze testiranja, razvijen je modul za analizu glavnih komponenti kernela KPCA, kao i poseban modul u formi genetskog algoritma GA. KPCA modul ma ulogu izdvajanja glavnih atributa iz DDoS skupa podataka, dok se GA koristi za optimizaciju različitih parametara SVM algoritma. Kako bi se smanjio šum uzrokovan razlikama među atributima, predložena je poboljšana funkcija kernela N-RBF, čijom je upotrebom smanjeno vreme trening faze modela učenja. Eksperimentalni rezultati su pokazali da za DDoS skup podataka, KPCA postiže veću efekasnost od standardno korišćene PCA tehnike za redukciju dimenzionalnosti izbora atributa. Postignuta je tačnost predloženog modela od 98,97%.

U radu [98], autori predlažu rešenje za detekciju napada zasnovano na proračunu entropije mrežnih atributa i primene unapređenog SVM algoritma nadgledanog mašinskog učenja. Rešenje predstavlja hibridnu tehniku detekcije, i izvršeno je njeno poređenje sa individualnim metodama detekcije napada. Za procenu predložene metode korišćen je javno dostupni skup podataka DARPA, koji predstavlja kombinaciju legitimnog mrežnog saobraćaja i saobraćaja u uslovima napada. Analiza rezultata pokazuje da hibridna metoda detekcije napada zasnovana na proračunu entropije postiže veću tačnost detekcije anomalija i daje manji broj lažnih alarma, u poređenju sa sistemom koji koristi samo SVM algoritam mašinskog učenja.

Autori u svom radu [99] izlažu dva pristupa za detekciju DDoS napada u SDN mrežama. Prvi pristup koristi proračun entropije odredišnih IP adresa i statistiku mrežnih tokova kako bi se precizno razdvojio legitimni saobraćaj od malicioznog, koji predstavlja DDoS napad. Drugi pristup koristi algoritam slučajne šume, model nadgledanog mašinskog učenja kojim se vrši precizna klasifikacija saobraćaja na legitiman i nelegitiman, tj. kada je mreža pod DDoS napadom. Rad [100] primenjuje tehniku dubinskog učenja za otkrivanje DDoS napada, pri čemu su skupovi atributa klasifikacionih algoritama paketa slični onima koji su upotrebljeni u ovom radu. Rezultati su pokazali da je oko 95% malicioznog saobraćaja i oko 99% legitimnog mrežnog saobraćaja ispravno klasifikovano.

# 6. TEORIJSKE OSNOVE ISTRAŽIVANJA

U ovom poglavlju prezentovane su teorijske osnove mehanizama detekcije DDoS napada na kojima se zasniva istraživanje sprovedeno u ovoj disertaciji. Kako bi se lakše razumeo i pratio dalji sadržaj rada, prvo će biti opisan pojam entropije, koja je izabrana kao primerni metod za detekciju anomalija u SDN mreži. Zatim će biti prikazane teorijske osnove i objašnjeni najvažniji pojmovi postupaka nadgledanog mašinskog učenja, koji su u drugom delu rada upotrebljeni za klasifikaciju DDoS napada.

## 6.1. Entropija kao metoda detekcije DDoS napada

Entropiju kao pojam koji definiše meru neuređenosti sistema, detaljno je opisao i prilagodio teoriji informacija Klod Šenon 1948. godine (Šenonova entropija) [101]. On je entropiju definisao uvođenjem pojma verovatnoće događaja, koja predstavlja osnovu teorije informacija. Prvo je definisan alfabet tj. skup koji sadrži simbole $A_i$, $i=0...M-1$, pri čemu postoji konačan broj od $M$ različitih simbola. Pojedine osobine izvora podataka mogu biti merljive jedino ako su poznate verovatnoće pojavljivanja pojedinih simbola $p_i=p(A_i)$, $i=0 ... M-1$. Pri tome, važi činjenica da je skup svih događaja (simbola) tzv. siguran događaj. Za svaki siguran događaj važi ograničenje dato izrazom (6.1):

$$\sum_{i=0}^{M-1} p_i = 1 \tag{6.1}$$

Što je poruka verovatnija, zaključak je da ona nosi manju količinu informacija i obrnuto. Količina informacija koju mogu nositi pojedinačne poruke može se definisati kao recipročna vrednost verovatnoće pojavljivanja posmatrane poruke. Ako je poruka verovatnija, ona nosi manju količinu informacija i obrnuto. To znači da je entropija najveća za potpuno slučajno generisane podatke, a najmanja kada se generišu potpuno predvidivi podaci. Iako poruke sa malom verovatnoćom nose veliku količinu informacija, to ne znači da je izvor

takvih poruka naročito efikasan. To je osnovni razlog što se za izvor informacija može definisati entropija kao mera prosečne količine informacija, ili kao srednja mera neizvesnosti izvora. Za sisteme kod kojih važe pretpostavke da je količina informacija koju nosi siguran događaj jednaka nuli, a količina koju nosi malo verovatan događaj veoma velika, može se definisati matematički model po kom se količina informacija $Q_i$ koju nosi poruka $A_i$ sa verovatnoćom pojavljivanja $p_i$ određuje izrazom (6.2):

$$Q_i \propto \log(\frac{1}{p_i}) \tag{6.2}$$

Šenonova entropija uzima u obzir prethodni izraz, a za skup $X$ od $n$ promenljivih $\{x_1, x_2, x_3, \dots x_n\}$ čije su verovatnoće dešavanja $p(x_1), p(x_1), \dots p(x_n)$ definisana je izrazom (6.3):

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log\big(p(x_i)\big) \tag{6.3}$$

U teoriji informacija osnova logaritma je broj 2, pa su za digitalni izvor koji generiše samo dva moguća simbola poruke 0 i 1 ("da" i "ne"), verovatnoće događaja $p_1=p_2=0.5$ (količina informacija koju nosi svaki simbol) i entropija je na osnovu (6.3) jednaka jedinici. Za događaj kada oba slučaja imaju istu verovatnoću, vrednost entropije je maksimalna, nesigurnost ishoda događaja je najveća, pa je informacioni sadržaj najveći. Ako je ishod događaja siguran, tada nema neodređenosti pa je vrednost entropije nula. Promena vrednosti verovatnoća događaja u smislu njihovog ujednačavanja uzrokovaće porastom entropije, pri čemu su mogući ishodi međusobno zavisni. To znači da povećanje verovatnoće ishoda jednog događaja smanjenje verovatnoću ishoda drugog. Nasuprot tome, promena vrednosti verovatnoća događaja u smislu povećanja njihove razlike uzrokovaće smanjenjem entropije, jer će jedan od ishoda događaja biti značajno izvesniji od ostalih.

Najbitnije osobine entropije su sledeće:

- Nenegativnost: $\forall p(x_i) \in [0,1]\ \ H(X) > 0$;
- Simetričnost: $H(p(x_1), p(x_2), \dots) = H(p(x_2), p(x_1), \dots)$;
- Maksimalnost: $H(p(x_1), \dots, p(x_n)) \leq H(\frac{1}{n}, \dots, \frac{1}{n}) = \log_a n$;
- Aditivnost: $H(X,Y) = H(X) + H(Y)$ (za skupove X i Y koji su nezavisni);

Šenonova entropija je kompromis između postignutih uticaja iz glavne mase događaja koji se često pojavljuju i uticaja ređih događaja. Za preciznije određivanje entropije koriste se parametarizovane entropijske generalizacije, tj. Rényi i Tsallis entropije [102, 103]. Za obe vrste ovih entropija koristi se promenljivi parameter skaliranja $q$. Isticanje čestih događaja prepoznaje se promenom $q$ parametra ka većim pozitivnim vrednostima, dok je njegova promena ka negativnim vrednostima znak retkih događaja. U slučaju negativne vrednosti parametra $q$ vrednosti entropije su veće od 1. Ova osobina se koristi kod podešavanja osetljivosti detektora anomalija. Tsallis i Rényi vrednosti entropije $H_R(X)$ i $H_T(X)$ se računaju prema izrazima (6.4) i (6.5):

$$H_T(X) = \frac{1}{1-q}\left(\sum_{i=1}^{n} p(x_i)^q - 1\right) \tag{6.4}$$

$$H_R(X) = \frac{1}{1-q}\, log_b\left(\sum_{i=1}^{n} p(x_i)^q\right) \tag{6.5}$$

Entropija predstavlja jedinstvenu metodu za detekciju anomalija ili napada u mrežama, pa je razumevanju njenih karakteristika posvećen značajan deo ovog rada. Polazi se od pretpostavke da se varijacije entropije u analizi mrežnog saobraćaja smatraju pouzdanim znakom da su se dogodile anomalije. Predloženo rešenje u ovom radu zasniva se na prikupljanju mrežnih podataka, za koje se u prvom koraku određuje vrednost entropije. Proračun entropije je moguće primeniti za detekciju anomalija, jer postoji određena korelacija među atributim polja zaglavlja mrežnih TCP/IP paketa. Polja zaglavlja paketa koja se koriste za analizu entropije su: izvorna/odredišna IP adresa, izvorni/odredišni broj porta, tip protokola, dužina paketa (*payload size*), polja flegova itd. Entropija se određuje poređenjem njene vrednosti za određena polja zaglavlja paketa jednog mrežnog toka, sa vrednostima polja drugog toka, pri čemu se detektuju razlike u slučajnosti raspodele. Vrednosti entropije izračunavaju se za slučaj legitimnog mrežnog statusa (kada nema napada), a zatim za slučaj kada se izvršava napad. Za legitimni status mreže, entropija polja zaglavlja ima vrednosti u određenom opsegu. Tokom napada, ovaj opseg vrednosti može da se promeni u velikoj meri. U zavisnosti od toga za koje polje zaglavlja paketa se računa, vrednost entropije se može značajno smanjiti ili povećati, pa se ta promena koristi kako bi se detektovao napad. Ako su promene vrednosti entropije velike, to može biti indikacija da je došlo do DDoS napada. Za definisani vremenski interval, entropijom je moguće odrediti ujednačenost raspodele

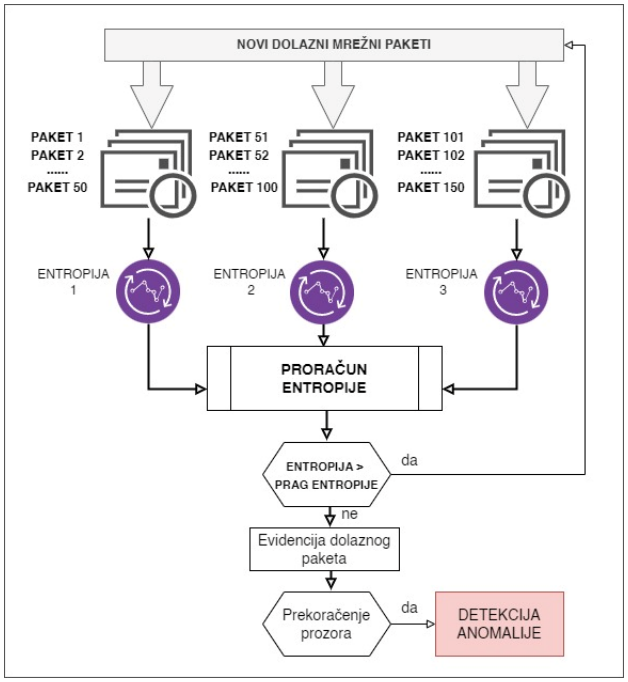
vrednosti za polja zaglavlja. Tokom vremenskih instanci podataka, ovim postupkom se prati stanje raspodele vrednosti entropije polja zaglavlja paketa. Postupak agregacije se vrši u određenim vremenskim intervalima (epohama) tako da se za određeno polje zaglavlja mrežnog paketa formira vremenski tok podataka.

Za proračun entropije značajno je koje polje zaglavlja paketa treba izabrati. Ukoliko određeno polje zaglavlja ima veliki broj vrednosti, njega treba svesti na što minimalniji skup. Skup vrednosti polja mora biti kompromisno rešenje, jer veliki skup otežava računanje entropije, dok je za mali skup vrednosti potrebno povećati vremenski prozor uzorkovanja entropije. Proračun entropija se najčešće sprovodi za polja odredišne ili izvorne IP adrese, ali treba uzeti u obzir da je opseg ovih adresa veliki i da su ove adrese dinamičke tj. promenljive. Zbog toga se ne koristi cela 32-bitna IP adresa, već jedan njen deo koji je zajednički za posmatranu mrežu, kao npr. adresa podmreže. Kao polje za proračun entropije mogu se koristiti brojevi izvornih ili odredišnih portova, kojih je manje nego IP adresa i koji su takođe dinamički. Polja flegova su značajna kao parametar, jer se pomoću njih mogu detektovati pojedini tipovi anomalija. U proračunu entropije obično se koristi ACK polje kao indikator TCP-SYN napada, ili se koriste polja zaglavlja tipa FIN i SYN. Moguće je kombinovati više polja kako bi se povećala preciznost detekcije, ali se pri njihovom izboru mora uzeti u obzir korelacioni parametrar. Problem izbora višestrukih polja u proračunu entropije je istraživan u radovima [104, 105].

Prag entropije (*threshold*) definiše se kao vrednost koja je u opsegu [0,1] i označava do koje vrednosti se može smanjiti entropija izmerena u vremenskom intervalu, pre nego što se može klasifikovati kao napad. Kod većine metoda za detekciju napada prvo se meri vrednost entropije za legitiman saobraćaj, a zatim entropija za slučaj napada. Poreeđenjem ove dve vrednosti, donose se zaključci o mogućim anomalijama ili napadima. Izbor praga entropije može znatno da utiče na tačnost detekcije. Niža vrednost praga entropije omogućava da mreža funkcioniše i sa neoptimalno izabranim parametrima, pa se ova vrednost definiše u slučajevima DDoS napada preplavljivanjem, kada nastaje drastičan pad vrednosti entropije. Tako se povećava mogućnost neispravne detekcije (*false negative*) i ignoriše aktivan napad. Nemogućnost detekcije napada može se odraziti na detekciju tokom kasnijih napada, pošto se izmerena vriednost entropije definiše kao nova bezbedna vrednost, i to se posmatra kao legitimno ponašanje mreže. Za uspešnu detekciju napada vriednost entropije mora biti ispod vrednosti praga, a od intenziteta napada zavisi broj neispravnih detekcija ili lažnih alarma.

Proračun vrednosti entropije na mrežnom uređaju hardverski je zahtevan postupak, naročito u slučaju kompleksnih SDN topologija sa velikim brojem uređaja i velikim mrežnim protokom. Entropija se određuje analizom paketa, prema redosledu njihovog pristizanja na port uređaja. Da bi se smanjili hardverski zahtevi proračuna entropije na uređaju, primenjuje se metod uzorkovanja, pri čemu se paketi analiziraju u tačno definisanom vremenskom intervalu (*time windows*) ili epohi, dok se jedan deo paketa ne analizira. Mada se na taj način smanjuje tačnost detekcije, kompromisom između dužine prozora i broja analiziranih paketa može se formirati sistem sa prihvatljivim nivoom tačnosti detekcije. Ako je uzorkovanje češće od optimalnog, nepotrebno se analizira veliki broj paketa, pa je proces detekcije spor i složen. Nasuprot tome, ako je uzorkovanje ređe od optimalnog, dolazi do gubljenja bitnih podataka i nemogućnosti detekcije anomalija. Nad prikupljenim instancama tokova moguće je formirati vremenski tok entropije svakog mrežnog atributa. Svaka vrednost odgovara izračunatoj entropiji u jednom vremenskom prozoru. Na taj način moguća je detaljnija analiza karakteristika mrežnog saobraćaja, uz mogućnost lakog uočavanja promena koje ukazuju na pojavu mrežnih anomalija. Veličina prozora se određuje tako da bude manja ili jednaka broju krajnjih čvorova mreže. Ako je prozor suviše veliki, onda kraći DDoS napadi mogu ostati skriveni, pa izračunata entropija možda neće prikazati jasnu razliku između legitimnog i saobraćaja napada. Međutim, ako je prozor premali, tada vrednosti entropije mogu biti suviše osetljive na promene u mreži. To znači da bilo koja mala promena u mreži može označavati napad, iako se on u stvarnosti nije desio. Ovo ima za posledicu stvaranje velokog broja lažnih alarma. U određenom broju radova analizira se uticaj veličine prozora na tačnost detekcije napada, a detaljnija analiza izvršena je u [106, 107].

Na slici 6.1 prikazan je postupak proračuna entropije za veličinu prozora od 50 paketa. Sličan postupak je upotrebljen u prvom delu predloženog rešenja za detekciju napada. Kada se na portu sviča, servera ili hosta detektuje više paketa nego što je određeno veličinom prozora, onda to predstavlja indikaciju da se u mreži dešavaju DDoS napadi. Svaki prozor sadrži 50 paketa, i za svaki od njih se izračunava entropija prema njihovom dolaznom redosledu, tako da se dobija njen vremenski tok. Prag entropije se određuje na osnovu opsega njenih varijacija, u uslovima legitimnog mrežnog saobraćaja. Ako je vrednost entropije veća od praga, procesiranje paketa se nastavlja sa sledećim dolaznim paketima, a ako je ispod definisanog praga, paket se evidentira i proverava da li je došlo do prekoračenja prozora. Ako jeste, onda je to slučaj kada se detektuje potencijalna anomalija ili napad.




**Slika 6.1:** Tok proračuna entropije za definisani prozor paketa

Iako primena samo entropijskih postupaka detekcije DDoS napada može biti vrlo efikasna za određene slučajeve i mrežne uslove, u radovima [108, 109] autori su pokazali da su oni ipak nedovoljno preciznni. Predloženo rešenje u okviru ove disertacije polazi od entropijski zasnovane metode, kao početne instance rešenja za detekciju DDoS napada, koja je zatim unapređena algoritmima nadgledanog mašinskog učenja, kako bi se obezbedila efikasnija, brža, i preciznija metoda detekcije mrežnih anomalija i napada.

## 6.2. Detekcija DDoS napada nadgledanim mašinskim učenjem

Razvoj veštačke inteligencije AI (*Artificial Intelligence*) poslednjih godina doveo je do njene primene u različitim oblastima istraživanja. Osnovni cilj oblasti mašinskog učenja ML (*Machine Learning*), kao podoblasti veštačke inteligencije, jeste izučavanje i razvoj sistema i algoritama koji upotrebom ulaznih podataka primarno izvršavaju određene zadatke predikcije ili klasifikacije. Uopštenije objašnjenje bi bilo da se mašinskim učenjem prikupljaju znanja iz mašinski čitljivih informacija, tako da se može „naučiti" određeno

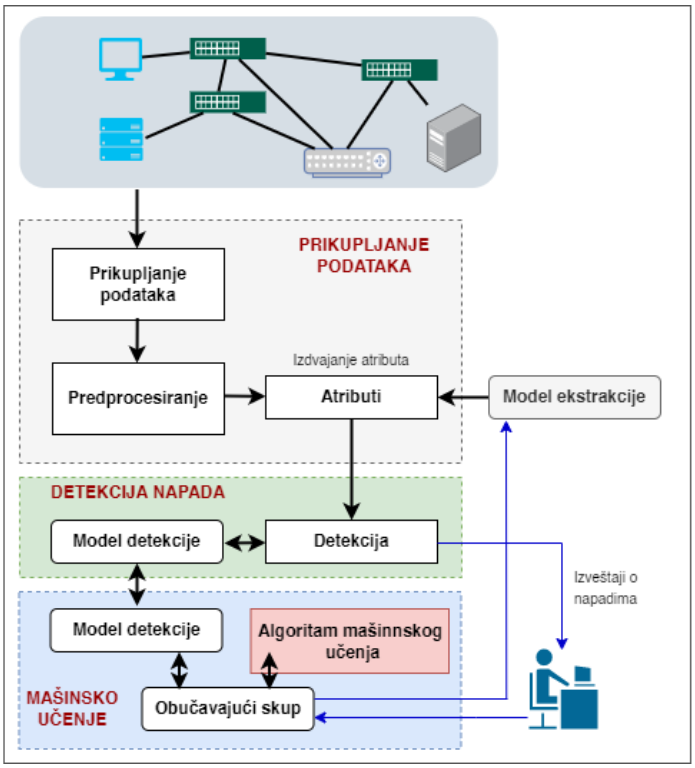
znanje iz ulaznih podataka. Mašinsko učenje se može definisati i kao složeni skup različitih metoda kojima se preko unapred zadatih informacija utvrđuje znanje, prepoznaju slične informacije, izvršava klasifikacija ili izračunavaju performanse sistema. Najopštija definicija mašinskog učenja data od strane Toma Mičela kaže da "računarski program može da uči na osnovu iskustva, u odnosu na performanse ili klasu zadatka, ukoliko se njegove performanse tokom izvršavanja zadatka poboljšavaju zahvaljujući usvojenom znanju o iskustvu" [110].

Ako bismo prethodnu definiciju primenili na sisteme za detekciju mrežnih napada, ona bi mogla da glasi "Sistem za detekciju napada uči da klasifikuje događaje (zadatke Z), mera performansi zadataka (P) je tačnost klasifikacije, a iskiustvo (I) je zadati obučavajući skup pravila". Mašinsko učenje u računarskim mrežama još uvek zahteva dodatna istraživanja, ali i rešavanje problema njegove implementacije u realna mrežna okruženja. Matematička složenost algoritama predstavlja ograničavajući faktor njihove primene, jer sporost izvršavanja i greške u modelima učenja mogu dovesti do prekida mrežnih servisa i velikih troškova. Potrebni su modeli većeg stepena generalizacije, koji se mogu prilagoditi uslovima dinamičnih i vrlo promenljivih mrežnih uslova. Iako neka od rešenja pokazuju da model učenja, primenjen u jednoj mreži može u određenoj meri imati dobre performanse i u drugoj, to je i dalje istraživački izazov, jer većina algoritama mašinskog učenja koristi pretpostavku da mrežne podatke definišu iste karakteristike verovatnoće i raspodele, što u savremenim mrežama to uglavnom nije slučaj. Osim toga, pojedini modeli učenja, a posebno oni koji se oslanjaju na principe dubokog učenja DL (*Deep Learning*) i dalje su nepoznanica, i neophodna su dodatna istraživanja kako bi se detaljno proučile njihove karakteristike.

Pošto se ovo istraživanje bavi problemima detekcije i klasifikacije DDoS napada, mašinsko učenje je jedna od primarnh metoda upotrebljenih za analizu podataka. Osnovu najvećeg broja savremenih rešenja za detekciju napada baziranih na mašinskom učenju čini nekoliko uzastopnih koraka, koji se sastoje od sledećih aktivnosti:

- formulacije problema
- formiranja simulacione mrežne topologije
- prikupljanja mrežnih podataka i izdvajanja atributa
- predprocesiranja i normalizacije neobrađenih podataka
- formiranja modela mašinskog učenja i obučavanja
- procene i testiranja modela mašinskog učenja

Mada su razvijena i predložena različita rešenja za detekciju DDoS napada koja se baziraju na mašinskom učenju, iz dostupne literature se zaključuje da se nova rešenja konstantno razvijaju i unapređuju [111, 112]. Primer modela za detekciju napada koji koristi principe mašinskog učenja, a na kojem se delimično zasniva istraživanje u ovoj disertaciji prikazan je na slici 6.2. Model je predstavljen u okviru rada [113], i sadrži tri celine ili modula. Nakon formiranja simulacione mrežne topologije, prvi modul vrši prikupljanje mrežnih podataka i izdvaja njihove atribute. Prikupljanje podataka vrši se senzorskim uređajima (mrežni adapter, ruter, AP pristupni uređaj…). Nakon predprocesiranja podataka, softverski se izdvajaju bitni mrežni atributi i vrši se konverzija "sirovih" podataka u vektore atributa koji se mogu procesirati algoritmima mašinskog učenja. Drugi modul formira model detekcije na osnovu primenjenih algoritama mašinkog učenja. Ako se koristi nadgledano učenje, tada algoritam tokom faze obučavanja (treninga) formira model na osnovu obučavajućeg skupa (*training set*), u kojem su svi primerci (instance) označeni, tj. dodeljena im je oznaka klase kojoj pripadaju. Treći modul vrši detekciju napada u realnom vremenu, i koristi prethodni modul, kao i izdvojene vektore atributa da bi izvršio klasifikaciju mrežnog saobraćaja na legitimni i maliciozni.



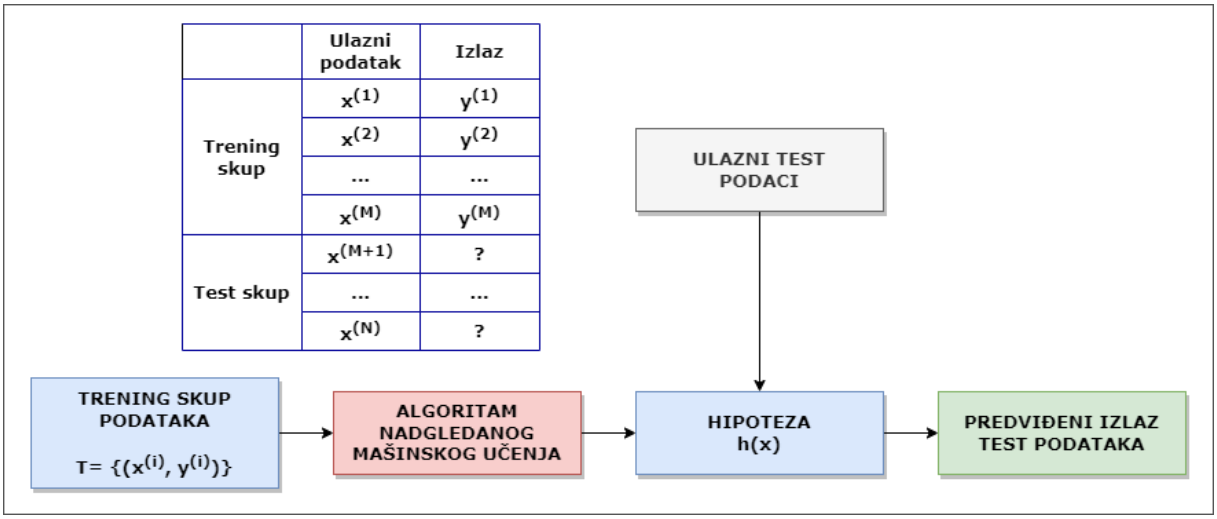

**Slika 6.2:** Model za detekciju napada zasnovan na mašinskom učenju [113]

Formulacija problema podrazumeva precizno definisanje tipa i obima napada, kao i određivanje kategorije modela mašinskog učenja (klasifikacija, regresija, klasterovanje…). Na taj način određuje se tip podataka koje treba prikupljati, kao i koji model učenja treba odabrati. Pogrešna apstrakcija problema može dovesti do izbora neogdovarajućeg modela mašinskog učenja, što uglavnom rezultuje njegovim lošim performansama i greškama u postupcima klasifikacije napada.

Predprocesiranje podataka predstavlja vrlo značajan korak postupka mašinskog učenja. Karakteristika mrežnih podataka prikupljenih u realnom vremenu jeste da su oni "neobrađeni", i kao takvi nisu pogodni za algoritme mašinskog učenja. Da bi se utvrdila ispravnost i efikasnost izvršavanja algoritma učenja, formiraju se dva skupa podataka - obučavajući i skup za testiranje. Uloga obučavajućeg skupa je prvenstveno optimizacija modela mašinskog učenja, dok se skup za testiranje koristi za procenu konačnog modela. U fazi testiranja modela, formiranim skupom podataka vrši se procena njegove efikasnosti izvršavanja, kao i procena grešaka. Ako su performanse modela zadovoljavajuće, model može da se koristi za predikciju budućih podataka. Bitna činjenica je da se parametri za skaliranje atributa i redukciju njihove dimenzionalnosti, dobijaju samo iz obučavajućeg skupa podataka, a isti parametri se kasnije ponovo primenjuju za transformisanje skupa podataka za testiranje, kao i bilo koje instance novih podataka. Vrlo važan i zahtevan deo analize podataka predstavlja izbor atributa kojima se opisuje konkretan mrežni domen. Do sada je predložen veći broj metoda za izbor mrežnih atributa, a nove metode se i dalje razvijaju [114, 115].

Klasifikacija algoritama mašinskog učenja najčešće podrazumeva tri metode: nadgledano učenje (*supervised learning*), nenadgledano učenje (*unsupervised learning*) i učenje podsticanjem (*reinforcement learning*). U upotrebi su i algoritmi polunadgledanog učenja (*semi-supervised learning*), ka oi učenje kombinovanim tehnikama.

Većina savremenih rešenja za detekciju napada u mrežama pripada grupi nadgledanog mašinskog učenja, pa je istraživanje u okviru ove disertacije usmereno ka ovom modelu učenja. Koristeći prethodno naučene relacije među podacima, algoritmima nadgledanog mašinskog učenje vrši se generalizacija podataka. Na osnovu matematičkih modela i velikih skupova podataka, model može da predvidi ponašanje sistema. Termin "nadgledan" se odnosi na skup ulaznih obućavajućih ili trening podataka, pri čemu je željeni izlaz već poznat, a algoritam može da "nauči" na koji način neobeleženom podatku može dodeliti tačnu izlaznu vrednost. Na slici 6.3 dat je šematski prikaz procesa nadgledanog mašinskog učenja.




**Slika 6.3:** Proces nadgledanog mašinskog učenja

U prikazanom modelu nadgledanog učenja, $x^{(i)} \in R^n$ predstavlja vektor ulaznih podataka (*feature vector*), dok je $y^{(i)} \in R$ odgovarajuća izlazna (ciljna) vrednost. Svaki uređeni par $(\boldsymbol{x}^{(i)}, y^{(i)})$ predstavlja $i$-ti trening primerak. Konačan skup ovih trening primeraka podataka označen je kao trening skup $T=\{(\boldsymbol{x}^{(i)}, y^{(i)})\}$, $i=1,\ldots, M$. U trening skupu $T$, definisano je $M$ trening vektora oznake $\boldsymbol{x}^{(i)}=[x_1{}^{(i)}, x_2{}^{(i)},\ldots, x_n{}^{(i)}]^T$, a svakom vektoru je pridružena unapred poznata vrednost izlazne promenljive $y^{(i)}$. U trening skupu $T$ indeks svakog objekta "$(i)$" predstavlja indeks mrežnog podatka tj. njegovog deskriptora ili atributa. Glavni cilj nadgledanog mašinskog učenja je na koji način uz pomoć trening skupa $T$ "naučiti" funkciju $h(x)$ da na najbolji mogući način aproksimira vrednosti izlaznih promenljivih $y^{(i)}$. Određivanje funkcije $h(\boldsymbol{x})$ svodi se na problematiku njene matematičke aproksimacije, pa se funkcija $h(\boldsymbol{x}):R^n \rightarrow R$ još naziva i predikcija ili hipoteza. Funkcija predikcije ili hipoteze se formira unapred poznatim obučavajućim skupom podataka, a zatim primenom algoritma nadgledanog mašinskog učenja, tako da se parametri funkcije $h(\boldsymbol{x})$ određuju procesom obučavanja modela [116]. U realnim okruženjima se najčešće umesto termina "naučiti funkciju predikcije" koristi izraz "treniranje modela".

Izvršena je podela nadgledanog mašinskog učenja na klasifikacione i regresione modele, prema vrsti izlaznih vrednosti. Postupkom klasifikacije podaci se svrstavaju u neku od predefinisanih, diskretnih klasa, tj. definišu se oznake (labele) klasa novih instanci (objekata) na osnovu ranijih opažanja. Za algoritme nadgledanog mašinskog učenja potreban je sveobuhvatan obučavajući skup podataka, koji treba da sadrži sve instance podataka. Nad

obučavajućim skupom podataka vrši se mapiranje ulaznih podataka u izlazne. Takođe, obučavajući i test skup moraju biti formirani po principu "osnovnih istinitosnih podataka", kako bi se postupcima nadgledanog učenja kreirala optimalna funkcija mapiranja.

Problem detekcije napada predstavlja primer klasifikacije, pri čemu se nadgledanim učenjem model obučava da može da razlikuje dve klase mrežnog saobraćaja – legitimni i maliciozni (kada je izvršen napad). Mrežni podaci nad kojima se izvršava klasifikacija sadrže nezavisne promenljive tj. mrežne atribute, kao i zavisne (objašnjavajuće) promenljive (atribute klase). Atributi klase su u formi diskretnih vrednosti, i broj klasa određen je skupom tih vrednosti. Klasifikacioni model se testira nad ulaznim test podacima, a validacija obučavajućeg modela se vrši nekom od metrika klasifikacije. Za detekciju napada, potrebno je da se tokom obučavanja modela postigne visok nivo klasifikacione tačnosti. Sastavni deo postupka razvoja modela klasifikacije je njegova evaluacija, i njen cilj je da se pronađe najpogodniji model koji će efikasno klasifikovati budući mrežni saobraćaj tj. moći da izvrši predikciju mrežnog saobraćaja. Više o klasifikacionim modelima nadgledanog učenja, primenjenim u rešenjima za detekcije DDoS napada izloženo je u [117] i [118].

Pojam prenaučenosti (*overfitting*) modela klasifikacije svodi se problematiku njegove tačnosti. Prenaučenost dovodi do "previše naučenog modela", koji se uglavnom ne može primeniti za predikciju budućih podataka. Prenaučenim modelom definišu se karakteristični slučajevi i eventualne greške u podacima. Uzrok prenaučenosti je u osnovi suviše složen model učenja u odnosu na veličinu primenjenog skupa podataka. Kod složenih modela sa velikom reprezentativnom snagom, uzimaju se u obzir svi atributi podataka, uključujući i eventualne greške. Nasuprot njima, jednostavniji modeli manje reprezentative snage, imaju mogućnost generalizacije budućih podataka. Takođe, kod previše pojednostavljenih klasifikacionih modela, nije uvek moguće pronaći karakteristične šablone podataka, što ima kao posledicu nedovoljnu naučenost (*underfitting*). Prethodno navedeni pojmovi su najčešći problemi u slučaju primene mašinskog učenja u detekciji napada, pa se u radovima [119] i [120] može pronaći više podataka o uticaju složenosti modela na tačnost klasifikacije.

Mašinsko učenje u detekciji napada karakteriše da svaki atribut sadrži deo znanja koje se može iskoristiti za detekciju, mada treba imati u vidu da pojedini atributi više utiču na tačnost klasifikacije, dok drugi imaju manji uticaj. Izbor atributa od značaja ima veliki uticaj na performanse klasifikatora. Osnovna uloge algoritma za odabir atributa su pronalaženje podskupova ulaznih atributa koji mogu adekvatno predstavljati ulazne podatke, redukovanje

neodgovarajućih podataka, kao i postizanje dovoljno tačnih rezultata predviđanja [121]. Danas je u upotrebi više metoda za izbor atributa, mada je evidentan konstantan razvoj novih. Za svaki konkretan slučaj, jedan od zahtevnijih zadataka takođe predstavlja način izbora uslovno optimalnog algoritma. Prema istraživanju u [122], izbor atributa postupkom njihove redukcije značajno utiče na brojne parametre: poboljšanje mogućnosti predikcije, smanjenje vremena proračuna, smanjenje efekata dimenzionalnosti podataka, utiče na bolje razumevanje podataka itd. Da bi se smanjili hardverski zahtevi složenih algoritama, obučavanje klasifikatora se vrši skupom iz kojeg se uklanjaju atributi koji imaju mali doprinos detekciji. Ako je sistem obučen skupom redukovane dimenzionalnosti, on će mnogo brže klasifikovati nepoznate instance, ali je količina znanja koja se koristi za klasifikaciju manja. Jednostavnom redukcijom obučavajućeg skupa pravi se kompromis između brzine klasifikacije i tačnosti detekcije i taj postupak nije prihvatljiv u sistemima koji zahtevaju visoku tačnost detekcije i postojanje malog broja lažnih alarma. Metoda detekcije napada zasnovana na veštačkim neuronskim mrežama ANN (*Artrifical Neural Networks*) postiže veću brzinu klasifikacije, ali koristi manji broj značajnih atributa iz obučavajućeg skupa, što se može zaključiti na osnovu radova [123] i [124].

U okviru ovog istraživanja za evaluaciju klasifikacije korišćeno je pet algoritama nadgledanog mašinskog učenja: algoritam potpornih vektora SVM (*Support Vector Machines*), naivni Bajesov algoritam NB (*Naïve Bayes*), algoritam k-najbližih suseda K-NN (*k-Nearest Neighbour*), algoritam zasnovan na stablu odlučivanja DT (*Decision Tree*) i algoritam slučajne šume RF (*Random Forest*). Ukratko su opisane osnovne karakteristike ovih algoritama, jer bi njihova detaljna analiza prevazišla okvire ovog rada.

**Algoritam potpornih vektora SVM:** Predstavlja linearni klasifikator koji pronalazi optimalnu hiperravan kojom se vrši razdvajanje podataka različitih klasa [125]. Polazi se od pretpostavke da za linearno razdvojive podatke postoji beskonačno mnogo hiperravni koje mogu da izvrše razdvajanje podataka iz obučavajućeg skupa. Hiperravan sa maksimalnom marginom razdvajanja vršiće optimalno raspoređivanje novih instanci podataka. Granični potporni vektori su oni kojima se ograničava širina margine. Za definisanje klasifikacionog modela u obzir se uzima linearna kombinacija potpornih vektora, dok se prostale tačke podataka odbacuju. Pošto je broj izabranih potpornih vektora SVM algoritmom uglavnom mali, ovaj klasifikator je prihvatljiviji kod manjih skupova podataka sa većim brojem atributa. Pogrešno klasifikovane instance podataka, dovode do toga da SVM klasifikator teško

pronalazi optimalnu razdvajajuću hiperravan. Za rešavanje ovog problema koristi se princip "mekih margina", koje u određenoj meri dozvoljavaju pogrešno klasifikovane primerke iz obučavajućeg skupa podataka. U slučaju da nije moguće razdvajanje podataka, SVM algoritam preslikava podatke u višedimenzionalni prostor atributa (*feature space*), i u okviru njega definiše jedinstvenu hiperravan razdvajanja. Nakon toga, algoritam koristi specifičnu funkciju jezgra (kernel) za određivanje novih tačaka u višedimenzionalnom prostoru. Ovom funkcijom se formira višedimenzionalni prostor atributa sa klasifikovanim instancama obučavajućeg skupa. U okviru istraživanja [126] i [127] predstavljena su rešenja klasifikacije DDoS napada bazirana na SVM algoritmu, realizovana za SDN mrežno okruženje.

**Näive-Bayes algoritam NB:** Ovaj algoritam, definisan kao "naivni", polazi od pretpostavke da su atributi uslovno nezavisni tj. da su vrednosti atributa za datu klasu nezavisne u odnosu na ostale klase. Osnovu NB algoritma predstavlja modelovanje raspodele ciljne promenljive ili klase ***y*** za zadate vrednosti ***x***, uz korišćenje Bajesove teoreme (6.6):

$$p(y|x) = \frac{p(y,x)}{p(x)} = \frac{p(x|y)p(y)}{p(x)} \tag{6.6}$$

pri čemu *p(y)* predstavlja prethodnu (apriornu) verovatnoću klase (*prior probability*), *p(y|x)* je posteriorna ili uslovna verovatnoća klase (*posterior probability*), *p(y,x)* je zajednička verovatnoća klase i podatka (*joint probaility*), dok je *p(x|y)* funkcija izvesnosti (*likelihood function*) [125]. Klasifikacijom podatak treba svrstati u klasu koja je za njega najverovatnija, tj. ima najveću posteriornu verovatnoću. Da bi se izvršila klasifikacija, potrebno je *x* i *y* tretirati kao slučajne promenljive, znati vrednosti verovatnoća *p(x|y)* i *p(y)*, tj. maksimizirati vrednost *p(y,x)* koristeći Bajesovu teoremu posteriorne verovatnoće. Polazi se od pretpostavke da su sve promenljive kategoričke i numeričke i da ih je potrebno transformisati primenom metode diskretizacije. Precizna procena uslovnih verovatnoća za svaku moguću kombinaciju klasa i vrednosti atributa je bitan problem, jer zahteva veliki obučavajući skup. Verovatnoća *p(x)* se lako određuje za jednodimenzionalne prostore, ali u višedimenzionalnim je potrebno eksponencijalno mnogo podataka za njen proračun. Verovatnoća *p(y)* se lako procenjuje iz obučavajućeg skupa, a za procenu uslovne verovatnoće *p(x|y)* koristi se NB klasifikator. Neke od prednosti NB algoritma su jednostavanost, primena na male skupove podataka, efikasanost i laka interpretacija. NB klasifikatori u nekim slučajevima mogu da nadmaše kompleksnije, npr. algoritme bazirane na stablu odlučivanja, naročito za skupove podataka sa manje atributa, što je u okviru istraživanja [128] detaljno analizirano.

**Algoritam stabla odlučivanja DT:** Ovaj algoritam spada u kategoriju iterativnih prediktivnih modela. Razdvajanje skupova podataka uzima u obzir vrednosti atributa, tj. njihov značaj za sam klasifikacioni problem. Svaki čvor stabla odlučivanja, osim listova, sadrži po jedan test koji može da ima više ishoda. Svakom ishodu odlučivanja odgovara po jedna grana stable, koja dalje vodi do sledećeg čvora. Listovi su definisani vrednostima koja su procenjena od strane stabla. U zavisnosti od procene ishoda, instanca se prosleđuje granom ka sledećem čvoru, tako da se postupak rekurzivno nastavlja sve do lista čija vrednost predstavlja traženo predviđanje. U svakom od čvorova vrši se provera vrednosti pojedinačnih atributa podataka [125]. Postoje mnoge varijacije osnovnog algoritma, a najznačajniji su ID3, CART i C4.5 [129]. CART metoda koristi raspoložive podatke o ulazno/izlaznim promenljivim i kreira binarno stablo, pri čemu se grananje slogova u čvorovima vrši prema funkciji definisanoj za svaki ulazni atribut. CART algoritam koristi sva grananja kako bi pronašao onaj sa najvećom tačnošću modela. Za svaki mrežni atribut određuje se najbolje grananje u svakom čvoru, a "pobednik" se bira putem Gini indeksa GI (*Gini Index*), koji procenjuje sa kolikom efikasnošću atribut deli uzorke klase u direktnim potomcima. Rezultat podele skupa atributa dovodi do toga da su u sama stabla odlučivanja uključeni samo određeni podskupovi ulaznih skupova atributa. DT algoritam predstavlja dobro rešenje kada je potrebno jednostavno grafičko predstavljanje ili interpretacija posmatranog modela, bez obzira na to da li se koristi regresiono ili klasifikaciono stablo odlučivanja.

**Algoritam k-najbližih suseda K-NN:** Zasnovan je na memoriji ili instancama i spada u tzv. lenje metode učenja (*lazy learning*) [125]. Kod metode učenja instancama se umesto izvođenja eksplicitnog modela upoređuju sve nove instance sa onim instancama koje se nakon faze obučavanja čuvaju u memorijskim strukturama. Najjednostavniji oblik memorijske strukture predstavljen je u formi višedimenzionalnog prostora atributa, sa formatom inicijalnog vektora, pri čemu je svaka instanca obučavajućeg skupa predstavljena kao tačka u prostoru. K-NN algoritmom se vrši klasifikacija tačke posmatranja u odnosu na to kako su susedi klasifikovani, tj. prema kriterijumu "najbližeg suseda". Svaka nova instanca se poredi sa instancama obučavajućeg skupa putem specifične k-NN metrike, pri čemu joj se dodeljuje klasa koja se najčešće javlja između *K* instanci. K-NN metrikom definiše se međusobna udaljenost između instanci na osnovu vrednosti njihovih atributa, a kao kriterijum se koristi princip intuitivnosti sličnih instanci, što znači da ako su instance sličnije, njihovo međusobno rastojanje je manje i obrnuto.

K-NN klasifikatorom se vrši izbor dva parametra: broj suseda *K* i rastojanje *d*. Izbor ovih parametara nije jednostavan postupak. Ako se u razmatranje uzme mali broj suseda, klasifikator će koristiti samo par tačaka podataka, što će dovesti do nestabilnosti usled varijacija tih par susednih tačaka. Nasuprot tome, za slučaj velikog broja suseda, odstupanja susednih klasa od stvarne mogu biti značajna. Preporuka je da se za skupove podataka male dimenzionalnosti, broj suseda *K* definiše na neku od vrednosti između 5 i 10. Dodatni nedosatatak K-NN algoritma je način pravilnog izbora funkcije rastojanja. Nedostatak eksplicitnog modela ima izvesne prednosti i nedostatke. Za složene modele tokom dodavanja novih podataka ne treba vršiti ažuriranje klasifikatora, tako da je usklađivanje približavanja na lokalnom nivou značajno olakšano. Za veće skupove podataka, neophodno je koristiti metod indeksiranja za nalaženje najbližih suseda. U [130] i [131] autori predlažu rešenja detekcije DDoS napada upotrebom KNN klasifikatora, primenjena na specifične SDN topologije.

**Algoritam slučajne šume RF:** Ovaj algoritam sadrži kolekciju klasifikatora zasnovanih na stablu odlučivanja $f_m(X)$, m=1,2, …M, pri čemu su *m* broj stabala, *M* je veličina podskupova instanci i atributa, dok je *X* ulazni vektor. Algoritam se zasniva na jednostavnoj formi agregacije stabla odlučivanja, kao i definisanim brojem na slučaj izabranih atributa u svakom čvoru grananja. Svaka kolekcija ili ansambl sadrži *m* stabala koji su procesirani putem različitih obučavajućih skupova. Broj *m* je diskutabilan parametar i pokazuje svojstvo da se za veći broj stabala odlučivanja dobijaju bolji rezultati i smanjuje predprilagođavanje modela, ali uz cenu vremena procesiranja. Za svako stablo se formira oznaka (glas) klasifikacije za ulazni vektor *X*. Rezultat algoritma klasifikacije je proizvod glasanja svakog stable u modelu. Za formiranje kolekcije RF stabala najzastupljeniji je metod pakovanja (*bagging*). Ovaj metod koristi obučavanje svakog stabla modela nezavisnim i ravnomernim podskupovima uzoraka, koji su izdvojeni iz obučavajućeg skupa. Obučavanje stabala različitim podskupovima vrši se iz razloga smanjenja korelacije grešaka, što se naknadno može korigovti postupkom agregacije. Klasifikator bira najpopularnije izglasane klase, dobijene od svih prediktivnih stabala sveobuhvatne šume. Stablo odluke podrazumeva izbor kriterijuma izbora atributa i metoda za "potkresavanja stabla". Algoritam slučajne šume se može koristiti i za slučajeve procene važnosti atributa. Izbor RF atributa se vrši ili proračunom ukupne značajnosti atributa, ili određivanjem GI indeksa. Veći broj referentnih radova je fokusiran na izbor ovog klasifikatora kao osnove za detekciju napada, a radovi [132, 133] predstavljaju primere njegove implementacije.

Tabela 6.1 daje kratak pregled prednosti i nedostataka algoritama mašinskog učenja koji su upotrebljeni za potrebe ovog istraživanja.

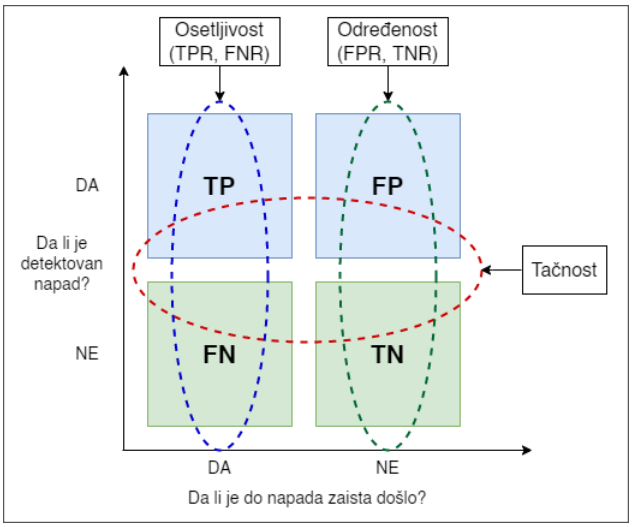
**Tabela 6.1:** Karakteristike algoritama nadgledanog mašinskog učenja

| Algoritam | Prednosti | Nedostaci |
|---|---|---|
| SVM | – Rad sa višedimenzionalnim podacima<br>– Učenje iz malih uzoraka podataka<br>– Rad sa nelinearnim atributima | – Sporost algoritma<br>– Hardverska zahtevnost<br>– Problem sa izborom kernela<br>– Preklapanje klasa |
| NB | – Jednostavnost algoritma<br>– Brza klasifikacija<br>– Skalabilnost i učenje iz malih skupova podataka | – Loše performanse za nereprenzetativne podatke<br>– Rad sa nezavisnim podacima<br>– Rad sa kontinualnim podacima |
| DT | – Jednostavnost algoritma<br>– Interakcije među atributima<br>– Trening skup sa kontinualnim i diskretnim podacima | – Osetljivost na podatke<br>– Problem prevelike podešenosti |
| KNN | – Jednostavna implementacija<br>– Lak za razumevanje<br>– Ne zahteva obučavajući skup<br>– Fleksibilnost izbora udaljenosti | – "Prokletstvo dimenzionalnosti"<br>– Kompleksan izbor *K* vrednosti<br>– Memorijska i vremenska zahtevnost |
| RF | – Rad sa velikim skupovima podataka<br>– Rešava problem prenaučenosti<br>– Manja osetljivost na podatke | – Sporost obučavajućeg skupa<br>– Komplikovan koncept predikcije |

Primarne kategorije (metrike) za evaluaciju performansi sistema za detekciju napada baziranog na mašinskom učenju su: dijagram reakcije, tačnost (*accuracy)*, preciznost (*precision*), osetljivost (*recall, sensitivity*), određenost (*specificity*) i F1-koeficijent.

Dijagram reakcije je karakterističan za oblast mašinskog učenja, i služi za procenu ispravnosti ili tačnosti predloženog modela. Za problem klasifikacije mrežnog saobraćaja, u slučaju dve ili više klasa, dijagramom se definišu različiti slučajeve alarma i odgovora sistema za detekciju na njih. Izgled dijagrama reakcije za slučaj dve klase prikazan je na slici 6.4. X-osa daje odgovor na pitanje "Da li se napad zaista desio?", "DA" predstavlja slučaj da je postojao napad, dok "NE" znači da napada nije bilo. Y-osa je odgovor sistema na napad tj. "Da li je napad detektovan?", "DA" označava da je sistem reagovao na napad kao stvaran, dok "NE" znači da sistem nije reagovao. Dijagram razlikuje četiri različita slučaja:

- stvarno pozitivni TP (*True Positive*) ili pravi alarm: važi za slučaj kada je sistem uspešno detektovao napad
- lažno pozitivni FP (*False Positive*) ili lažni alarm: važi za slučaj da je sistem detektovao nepostojeći napad kao stvarni
- lažno negativn FN (*False Negative*) ili propušteni alarm: označava slučaj kada sistem nije uspešno detektovao postojeći napad
- stvarno negativni TN (*True Negative*) ili ispravno legitiman: označava da sistem nije detektovao nepostojeći napad, tj. da se radi o ispravnoj detekciji normalne aktivnosti




**Slika 6.4:** Dijagram reakcije sistema za detekciju napada za dve klase

Efikasnost sistema detekcije određuje se na osnovu broja pojavljivanja lažno pozitivnih i lažno negativnih alarma. U skladu sa tim, definiše se osetljivost TPR (*True Positive Rate*) kao metrika koja označava koliki je procenat stvarnih napada dijagnostikovan algoritmom mašinskog učenja. Izračunava se prema izrazu (6.7) kao odnos broja stvarno detektovanih napada, i zbira pravih i propuštenih alarma:

$$TPR = \frac{TP}{TP + FN} = 1 - FNR \tag{6.7}$$

Sistem visoke osetljivosti ima nisku učestanost lažno negativnih alarma FNR (*False Negative Rate*), što znači da taj sistem mali broj napada pogrešno prepoznaje kao legitimne mrežne aktivnosti.

Određenost TNR (*True Negative Rate*) je metrika suprotna osetljivosti, pa se definiše kao učestalost pojave ispravno detektovanih aktivnosti, tj. kao odnos ispravno detektovanih legitimnih mrežnih aktivnosti i zbira stvarno negativnih i lažnih alarma, prema izrazu (6.8):

$$TNR = \frac{TN}{FP + TN} = 1 - FPR \tag{6.8}$$

FPR (*False Positive Rate*) predstavlja broj lažno pozitivnih alarma. Sistem za detekciju napada visoke određenosti ima malu FPR vrednost, što znači da takav sistem mali broj dozvoljenih aktivnosti pogrešno prepoznaje kao napade.

Tačnost je metrika koja se definiše kao odnos svih rezultata (pozitivnih i negativnih) koji su ispravni, i izračunava se prema izrazu (6.9):

$$\text{Tačnost} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.9}$$

Ako se od sistema za detekciju zahteva visoka tačnost, onda se među metrikama pravi kompromis u izboru, kako bi se detektovao mali broj lažnih i propuštenih alarma i dostigao visok nivo osetljivosti i određenosti.

Preciznost je metrika koja se takođe koristi kao mera efikasnosti detekcije i određuje koliki procenat instanci jedne klase koje su klasifikovane kao pozitivne, pripada toj klasi. Izračunavanje preciznosti vrši se na osnovu izraza (6.10):

$$Preciznost = \frac{TP}{TP + FP} \tag{6.10}$$

Metrika F1 definiše harmonijsku sredinu između kombinovanih metrika preciznosti i osetljivosti. Polazi se od stava da harmonijska sredina dva broja *x* i *y* uvek teži da bude bliža manjem od njih, pa se može zaključiti da veća vrednost F1 označava visoku preciznost i osetljivost. F1 metrika je robusnija mera za procenu performansi klasifikatora od tačnosti, jer pokazuje veći stepen otpornosti za slučajeve sa visokom tačnošću, i relativno velikom greškom. F1 metrika se izračunava prema izrazu (6.11):

$$F1 = \frac{2 * Preciznost * Odziv}{Preciznost + Odziv} \tag{6.11}$$

Korelaciona metoda, kao važna metrika mašinskog učenja, vrši procenu stepena korelacije među skupovima podataka. Izračunavanjem koeficijenata korelacije proverava se relevantnost atributa ili skupa atributa, i na taj način se procenjuje koliko promene u jednom

skupu podataka imaju uticaj na neki drugi skup podatka. Osnovni princip korelacije bazira se na činjenici da podskup atributa treba da sadrži isključivo one atribute koji su u visokoj korelaciji sa izlaznim podacima, dok međusobno atributi ne moraju biti jako korelisani [134]. Pirsonov koeficijent korelacije *R(i)* najčešće se koristi kao merilo stepena korelacije podataka. Izračunava se kao kovarijansa, predstavljena brojem standarnih devijacija posmatranih atributa. Može se odrediti izrazom (6.12):

$$R(i) = \frac{cov(x_i, y)}{\sqrt{var(x_i)\ var(y)}} \tag{6.12}$$

u kojem $x_i$ predstavlja sve vrednosti posmatranog atributa iz prostora *X*, dok je *Y* slučajna promenljiva izlaza *y*. U ovom izrazu je:

$$cov(x_i, \mathrm{y}) = \sum_{i=1}^{N} (x_i - \overline{x}_i)(y_i - \bar{y}) \tag{6.13}$$

$$var(x_i) = \sum_{i=1}^{N} (x_i - \overline{x_i})^2 \tag{6.14}$$

$$var(y) = \sum_{i=1}^{N} (y_i - y)^2 \tag{6.15}$$

*N* predstavlja broj trening primera u skupu, $\overline{x_i}$ definiše aritmetičku sredinu uzorka za izabrani atribut, dok je $\overline{y}$ aritmetička sredina uzorka izlaznih vrednosti. Za slučaj kontinualnih vrednosti atributa postoji ograničenje da se može odrediti isključivo linearna zavisnost između posmatanog atributa i ciljne vrednosti, iako su u okviru rada [135] opisani načini kako se ovo ograničenje moguće prevazići. Za nelinearne skupove podataka, model neće formirati dobre rezultate. Koeficijent korelacije može imati vrednosti iz opsega [-1, +1], pri čemu granične vrednosti -1 i +1 ukazuju na potpunu linearnost, dok vrednost 0 upućuje na potpuno odsustvo linearnosti modela.

# 7. PREDLOŽENA METODA DETEKCIJE NAPADA

Za potrebe razvoja metode detekcije DDoS napada u SDN mrežnom okruženju predstavljene u okviru ove disertacije, obezbeđeno je specifično simulaciono okruženje koje je omogućilo neophodne uslove za implementaciju algoritma proračuna entropije mrežnog saobraćaja, primenu algoritama mašinskog učenja, rad sa specifičnim skupovima podataka, kao i efikasano formiranje rezultata i njihovu reprezentativnu grafičku ili računarsku analizu. Za realizaciju rešenja detekcije DDoS napada korišćeni su Mininet mrežni emulator, softver za virtuelizaciju, kao i novije verzije softvera specifične za ovu oblast istraživanja. Dodatno su razvijene programske komponente koje su omogućile da se istraživanje uspešno realizuje, kao i da se dobiju konkretni rezultati kojima se definišu najvažniji naučno-istraživački doprinosi predloženog rešenja.

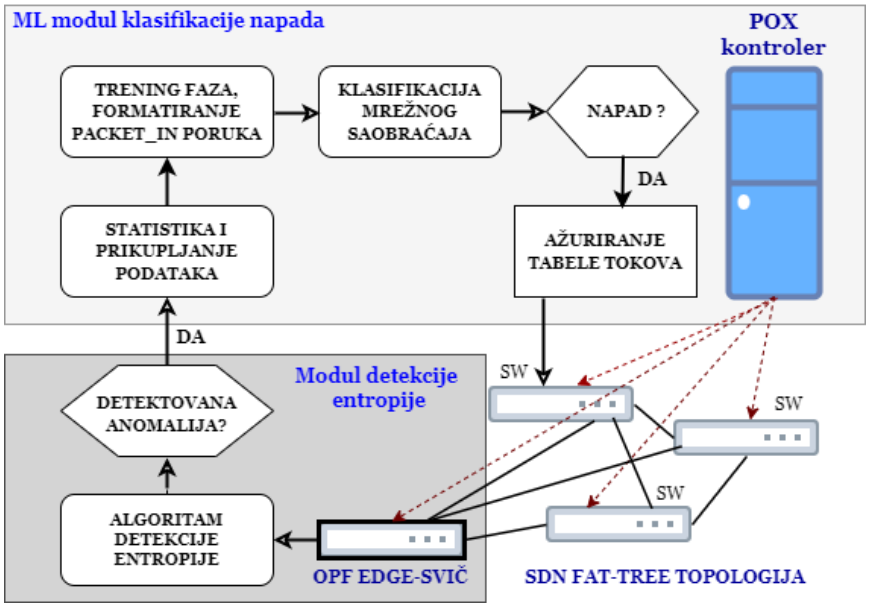
## 7.1. Opis predložene metode detekcije DDoS napada

Predloženom metodom za detekciju napada bilo je potrebno pre svega omogućiti da ona postiže dobru tačnost za različite tipove DDoS napada, da bude efikasna, skalabilna i fleksibilna. Takođe, bilo je neophodno uzeti u obzir procesorske i memorijski zahteve, ali i propusne opsege mrežnih linkova, kako bi predložena metoda mogla efikasno geneisati podatke u realnom vremenu. Metodama koje omogućavaju analizu sadržaja mrežnih paketa, moguće je npr. dobiti detaljne informacije o potpisima napada, koji se zatim mogu upotrebiti za formiranje modela detekcije. Treba uzeti u obzir činjenicu da je, nakon potvrde da je došlo do DDoS napada, proces identifikovanja napadača uglavnom otežan, usled nedostatka informacija kojima bi se određena mrežna konekcija mogla povezati sa identitetom korisnika koji je učestvovao u napadu. U slučajevima kada se koristi enkriptovana komunikacija, praktično je nemoguće takav sadržaj analizirati i ukoliko ne postoji sveobuhvatnost detektovanih potpisa napada, može se dogoditi potpuni izostanak njegove detekcije, jer napadači koriste dodatne mehanizme kako bi mrežni paketi bili što složeniji.

Najveći broj rešenja detekcije DDoS napada u SDN mrežama, počevši od onih koji koriste standardne statističke metode i algoritme mašinskog učenja, pa do onih koji su kombinacija različitih metoda ili metoda dubinskog učenja, koriste prednosti i napredne mogućnosti centralizovanih kontrolera kako bi se poboljšala tačnost detekcije. Uloga kontrolera je višestruka i kao što je objašnjeno u poglavlju 3.1, podrazumeva obradu i prikupljanje mrežnih podataka, praćenje statistike mrežnih tokova ili klasifikaciju napada zasnovanu na mašinskom učenju, što je često upotrebljeno rešenje u poslednje vreme. U slučaju kompleksnih mreža sa velikim brojem uređaja i protocima, SDN kontroler je često "kritična tačka" usled njegovog preopterećenja, koje obično stvara dodatno kašnjenje tokom procesa detekcije napada. Istovremeno, sa izvršavanjem klasifikacionih algoritama, razmenjuje se veliki broj poruka između kontrolera i svičeva, čime se dodatno ograničava propusni opseg SBI interfejsa. Jedan od ciljeva predloženog rešenja u ovom istraživanju jeste razvoj modularne arhitekture, kojom bi se smanjilo preopterećenje kontrolera i opseg komunikacije prema svičevima, uz značajno poboljšanje brzine detekcije napada i njene tačnosti. Metoda detekcije napada koristi osobinu programabilnosti OpenFlow svičeva i njihove hardverske resurse, kako bi se deo analize mrežnog saobraćaja izvršavao ne ivičnom sviču (*edge switch*) umesto na SDN kontroleru. Ivični svič SDN mreže predstavlja uređaj ivičnog sloja mreže, obično blizak izvoru napada, i u ovom rešenju izabran je za detekciju anomalija proračunom entropije. U radovima [136, 137] autori su istraživali različite mogućnosti poboljšanja bezbednosti SDN mreža kroz programabilne OpenFlow svičeve. Uz programabilnost svičeva lociranih u ravni podataka SDN mreže, koriste se i pojedine dodatne tehnike kao rešenje problema preopterećenosti kontrolera. Rešenja se zasnivaju ili na efikasnom filtriranju zahteva koji se šalju ka kontroleru sa ivičnih svičeva, ili se koristi firewall sa specifičnom hibridnom FPGA/CPU arhitekturom ravni podataka. Kao jedno od rešenja navodi se CPP (*Controller Protection Protocol*) protokol koji koristi mehanizam filtriranja lažnih zahteva koji se analiziraju i odbacuju na programabilnom ivičnom sviču, čime se sprečava preopterećenje kontrolera i SBI interfejsa [138].

Predložena metoda detekcije DDoS napada realizovana je kao višemodularna struktura, i šematski je prikazana na slici 7.1. Osnovu rešenja čine dva modula:

1) modul za detekciju anomalija i proračun entropije realizovan na ivičnom sviču
2) modul za klasifikaciju DDoS napada, realizovan na kontroleru i zasnovan na principima mašinskog učenja sa nadgledanjem




**Slika 7.1:** Šematski prikaz predloženog rešenja za detekciju DDoS napada

(Modul za detekciju anomalija i proračun entropije): Svi OpenFlow svičevi SDN mreže su pod nadzorom jednog centralizovanog POX kontrolera i čine jedinstven mrežni domen. Za model detekcije napada izabrana je specifična Fat-Tree SDN topologija, formirana Mininet emulacionim softverom. Jedan od OpenFlow svičeva topologije izabran je kao ivični (*edge-switch*), i na njemu se izvršava posebno razvijen algoritam za detekciju anomalija i proračun entropije. Način funkcionisanja ovog algoritma objašnjen je u poglavlju 7.4. Kako bi se napadi mogli detektovati, prvo se formira legitimni mrežni saobraćaj, a zatim se generišu TCP-SYN i ICMP napadi preplavljivanjem. Algoritmom se vrši nadgledanje tokova podataka, analizira se broj prenetih paketa i izračunava entropija odredišnih IP adresa dolaznih paketa. Ivični svič detektuje svaku promenu entropije i broja prenetih paketa u mreži, a podatak o detektovanoj mrežnoj anomaliji direktno prosleđuje POX kontroleru. Kriterijumi za detekciju anomalija su razlika entropije legitimnog i saobraćaja napada, kao i broj kontrolnih poruka koje se procesiraju u sviču. POX kontroler, kao centralni upravljački mrežni čvor, u legitimnim uslovima upravlja svakim svičem u domenu, što podrazumeva prosleđivanje podataka, proračune putanja i ažuriranja tabela prosleđivanja svih svičeva. Nakon detektovanja anomalije, kontroler vrši prekid tokova napada ažuriranjem tabele prosleđivanja sviča sa kojeg je napad došao. Nakon prijema informacije o nastanku anomalije od strane ivičnog sviča, kontroler odmah započinje proces klasifikacije, kako bi se utvrdilo da li je

nastala anomalija DDoS napad ili ne. U okviru ovog modula, analizirani su brzina detekcije i opterećenost kontrolera za oba tipa DDoS napada, kao i poređenje dobijenih rezultata sa centralizovanim rešenjem, tj. slučajem kada se detekcija entropije izvršava na POX kontroleru, a ne na ivičnom sviču.

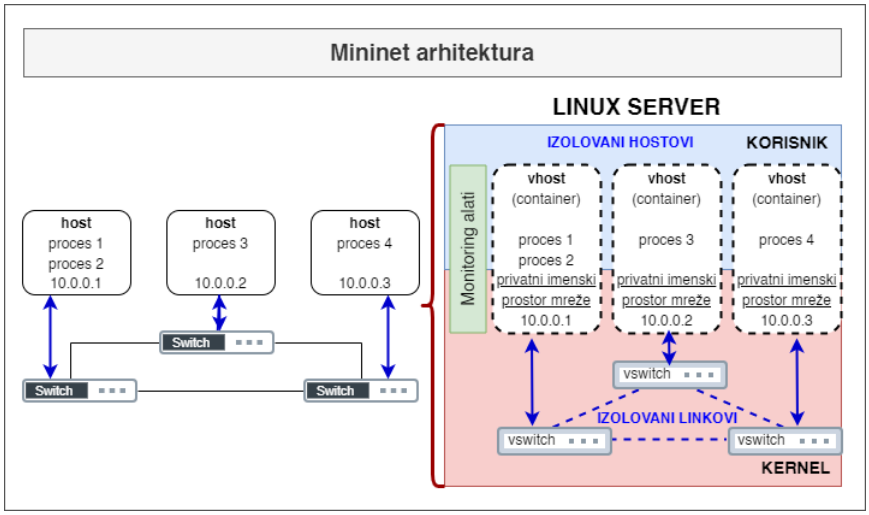
(Modul za klasifikaciju napada): Drugi modul koristi principe nadgledanog mašinskog učenja kako bi se na osnovu prikupljenih podataka iz SDN mreže kreirao obučavajući skup, koji algoritmi koriste da bi klasifikovali mrežne napade sa određenom tačnošću. Prema scenarijima napada, šalje se veliki broj paketa sa lažnim IP adresama kako bi se formirao veliki broj *Packet_In* poruka koje svičevi procesiraju i šalju kontroleru. *Packet_In* poruke, definisane OpenFlow protokolom i objašnjene u poglavlju 3.2, predstavljaju osnovne podatke koji su u ovoj metodi detekcije upotrebljeni za klasifikaciju. Skoro sva postojeća rešenja detekcije napada, dostupna u referentnoj literaturi, uglavnom koriste podatke zaglavlja TCP paketa, tj. polja sa atributima (karakteristikama) koji opisuju mrežnu komunikaciju. Od ovih atributa formiraju se obučavajući skupovi kojima se model detekcije "uči" da prepozna napade. Obučavajući skupovi su često hardverski zahtevni, prikupljanje mrežnih podataka je složen postupak sa kompleksnim procedurama njihovog predprocesiranja i normalizacije, a sa ciljem da ih algoritmi mašinskog učenja mogu pravilno interpretirati.

Modul za klasifikaciju napada formiran je sa ciljem da se upotrebi što jednostavniji model zasovan na nadgledanom mašinskom učenju, kako bi se poboljšale sveukupne performanse rešenja. Izbor skupa podataka za klasifikaciju detaljnije je objašnjen u poglavlju 7.5.2. POX kontroler vrši statističku analizu mrežnog saobraćaja, kao i konverziju podataka u format razumljiv algoritmima učenja. U fazi obučavanja, prikupljanje broja *Packet_In* poruka vrši se posebno za regularan i za saobraćaj napada, uz njihovo memorisanje i analizu za svaki mrežni čvor, u okviru definisanog vremenskog intervala. Na taj način se formira jedinstven obučavajući skup podataka koji omogućava brzu klasifikaciju napada, uz niske hardverske zahteve i veliku tačnost. Kontroler vrši anaalizu i prikupljanje poruka koje se razmenjuju između kontrolera i svičeva, i od njih se formiraju liste *Packet_In* poruka određenog formata. Iz ovih listi izdvajaju se uzorci koji se koriste kao podaci obučavajućeg skupa. Svaki uzorak u obučavajućem skupu predstavlja ponašanje mrežnih uređaja i definiše mrežni status. Uzorci sadrže podatke o broju prenetih *Packet_In* poruka, pa svaka veća promena njhovog broja indicira potenijalno dešavanje mrežnog napada.

## 7.2. **Izbor simulacionog okruženja**

Jedan od značajnih izazova u istraživanju SDN mreža predstavljaju načini testiranja njihovih servisa ili aplikacija u realnim okruženjima. S obzirom na sve specifičnosti SDN mreža, bez mogućnosti njihovog adekvatnog testiranja i analize, njihova primena postaje otežana i diskutabilna. Iako danas postoji veči broj mrežnh simulatora koji značajno olakšavaju procedure testiranja SDN mreža, praksa pokazuje određen stepen netačnosti i grešaka u analizi podataka. Najčešći uzrok su pojednostavljeni i ograničeni mrežni modeli, kao i neodgovarajuća podešavanja mrežnih parametara. Testiranje SDN mreža u realnim uslovima je uglavnom zahtevan postupak, jer to podrazumeva niz izmena u strukturama kompleksnih strukturnih mreža provajdera. Provajderi ili centri podataka obično ne žele bilo kakve modifikacije svojih mrežnih infrastruktura, i zahtevaju da se one prethodno detaljno testiraju na višestrukim prototipovima ili test-mrežama. Hardversko testiranje SDN mreža zahteva dodatnu nabavku specifične opreme i troškove, što nije ekonomski opravdano [139].

Zbog svih ograničenja i složenosti testiranja SDN mreža na realnim mrežnim strukturama, problem se uglavnom rešava tehnikom emulacije, koja predstavlja softversku imitaciju, tj. formiranje kopije realne mreže unutar virtuelizacione softverske platforme. Mininet je softver baziran na Linuxu, koji sa visokim stepenom tačnosti može da emulira SDN mreže sa velikom broju OpenFlow svičeva, kontrolera ili hostova. Omogućava kreiranje novih mrežnih funkcionalnosti koje se mogu testirati na složenim SDN topologijama sa realnim saobraćajem, a formirani programski kôd i testne skripte se mogu primeniti na realnu mrežnu strukturu. U široj naučnoj zajednici koja se bavi istraživanjima u oblasti računarskih mreža, Mininet se navodi kao nezamenljiv mrežni emulator za testiranje SDN mreža, pa je on za potrebe ovog istraživanja upotrebljen u inicijalnoj fazi za simulaciju mrežne topologije. Za testiranje SDN mreža i njihovih servisa, Mininet koristi kombinaciju virtuelizacije, API funkcija i CLI komandnog interfejsa na jednoj fizičkoj mašini. Zasnovan na OpenFlow protokolu, koristi Python API interfejse kojima je moguće definisati scenarije koji mogu oponašati realna mrežna okruženja. Referentna literatura prepoznaje veći broj radova u kojima se SDN mreža koristi u okviru Mininet emulatora. Autor ove disertacije je u okviru istraživanja [140] izvršio poređenje karakteristika ONOS i RYU kontrolera, koji su deo Mininet viruelizacione SDN mreže zasnovane na topologiji stabla. Na ovoj formi topologije zasniva se simulaciona SDN mreža u ovom istraživanju.




**Slika 7.2:** Arhitektura Mininet mrežnog emulatora

Arhitektura Mininet emulatora prikazana je na slici 7.2. SDN mreža se može predstaviti kao jedinstvena portabilna virtuelna mašina, pokrenuta ili modifikovana od strane više korisnika. Virtuelizacijom procesa izvršava se emulacija SDN mreže, tako da se na jednom sistemu može pokrenuti na stotine hostova i svičeva koristeći jedan kernel i korisnički kôd. Mininet za svaki izolovani virtuelni host (*vhost*) ili svič (*vswitch*) formira njegov imenski prostor i pojedinačne procese sa zasebnim mrežnim interfejsima, tabelama rutiranja i ARP (*Address Resolution Protocol*) tabelama, i povezuje ih preko virtuelnih (*veth*) Ethernet interfejsa. U emuliranoj mreži mogu se izvršavati realne mrežne aplikacije, podržane od strane Linux operativnog sistema koji pokreće Mininet. Kontrola Mininet hostova sprovodi se putem zaštićene SSH (*Secure Shell*) komunikacije. Mrežni programi na hostovima šalju pakete podataka preko emuliranih (*veth*) interfejsa, čija se kašnjenja i propusni opsezi mogu dodatno podešavati ili konfigurisati. Preko komandnog CLI interfejsa moguće je kontrolisati ili upravljati celom SDN mrežom, komunicirati sa hostovima i svičevima, ili testirati funkcionalnosti linkova ili mrežnih servisa. Pošto Mininet hostovi pokreću standardni Linux mrežni softver, moguć je razvoj i primena virtuelizovanih scenarija serverskih mrežnih servisa. To podrazumeva korišćenje softverskih kontejnera i njihovu upotrebu u cloud okruženjima. Mininet omogućava jednostavnu implementaciju Docker ili Solaris kontejnera, čime se dodaju nove i značajne funkcionalnosti za emulirane mrežne topologije ili cloud servise. Ovo je aktuelna tema, i u okviru radova [141, 142] istraživane su mogućnosti implementacije kontejnerskog softvera u kompleksnim višekontrolerskim Mininet virtuelizacionim topologijama.

Razvoj metode detekcije DDoS napada u ovom istraživanju zahtevao je specifično simulaciono okruženje, rad sa posebnim skupovima mrežnih podataka, kao i upotrebu različitih algoritama mašinskog učenja za klasifikaciju napada. Specifikacija hardvera i softvera simulacije prikazana je u tabeli 7.1.

**Tabela 7.1:** Specifikacija hardvera i softvera mrežne simulacije

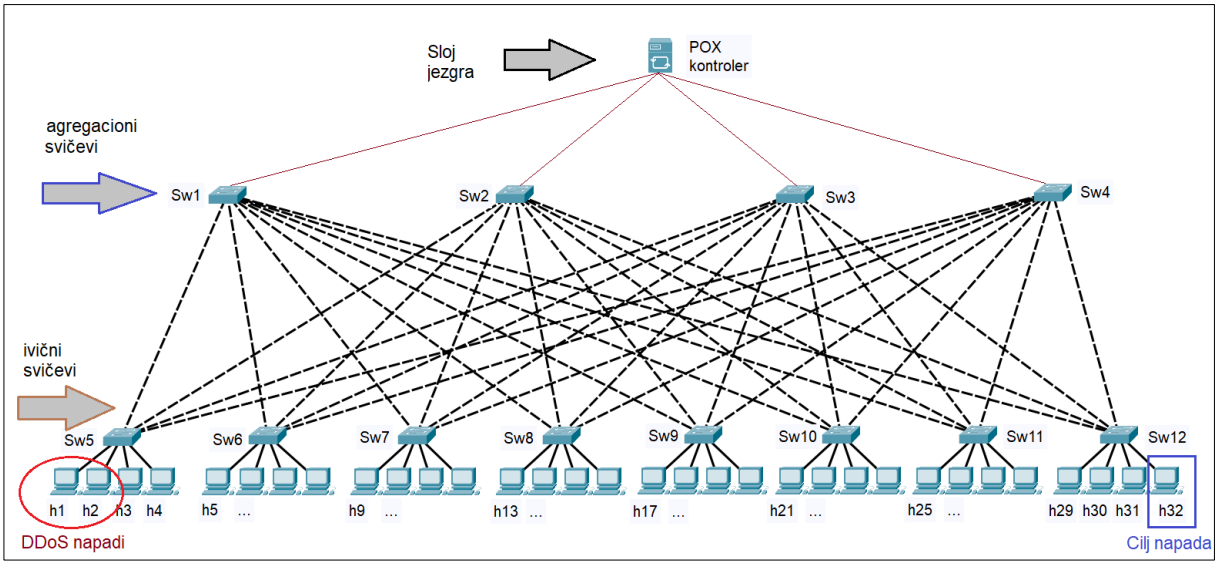
| | | **PC** | **Virtual Machine (VM)** |
|---|---|---|---|
| **Hardver** | Procesor | AMD Ryzen 5 3600, 3.6 GHz (6-core) | 6-core CPU |
| | RAM | 32 GB, DDR4 | 8 GB DDR4 |
| | SSD disk | Samsung 970, 1 TB | 20 GB |
| **Softver** | OS | Windows 11, v21H2 | OS Xubuntu 20.04.1 |
| | | | VirtualBox 6.1.16, r140961 |
| | | | Mininet 2.3.0d6 |
| | | | POX 0.5.0 |
| | | | Spyder IDE 5.4.0 |
| | | | Python 3.8.5 |
| | | | Wireshark 3.4.14 |
| | | | Scapy 2.4.5 |

Simulacija je realizovana na desktop raačunaru pod Windows 11 (ver. 21H2) operativnim sistemom. Bilo je neophodno obezbediti virtuelizaciono okruženje kako bi se u okviru njega izvršavala Mininet emulacija SDN topologije. Pošto virtuelizacija po svojoj prirodi zahteva veće hardverske resurse, hardverske specifikacije računara su bile sledeće: procesor AMD Ryzen 5 radnog takta od 3,6GHz, memorija DDR4 kapaciteta 32GB i SSD disk kapaciteta 1TB. Takođe, bilo je potrebno u okviru BIOS čipa omogućiti opciju SVMM (*Secure Virtual Machine Mode*) za podršku AMD-V virtuelizacionog moda procesora. Ulogu hipervizora obezbedio je VirtualBox 6.1.16 softver, u okviru kojeg je formirana virtuelna mašina pod Xubuntu 20.04 distribucijom (kernel 5.4), a na kojoj se izvršava Mininet 2.3 mrežni emulator, sa podrškom za OpenFlow protokol (verzija 1.3). Virtuelna mašina je konfigurisana tako da koristi 6-jezgarni procesor, 8GB RAM memorije i 20GB skladišnog prostora diska. Osnovni razlozi izbora Xubuntu distribucije za virtuelizaciju bili su njegova brzina izvršavanja, korišćenje minimalnih sistemskih resursa i XFCE interfejs koji je jednostavan, stabilan i konfigurabilan. Pregledni rad [143] navodi razloge zbog kojih je Xubuntu distribucija vrlo čest izbor za SDN mrežnu virtuelizaciju.

## 7.3. **Izbor topologije SDN mreže**

Izbor i formiranje SDN mrežne topologije je bio prvi korak u razvoju predložene metode detekcije napada. Kao primarna forma mrežne strukture u ovom istraživanju izabrana je Fat-Tree (FT) topologija, koja je podvrsta topologije stabla (*tree-network topology*). Nekoliko je razloga zbog kojih je izabran ovaj tip SDN topologije. Osnovni razlog je što se Fat-Tree topologija može programski kreirati, implementirati i testirati u okviru Mininet okruženja, bez obzira na njen nivo složenosti. Takođe, autor ove disertacije koristio je rezultate predstavljene u sopstvenom radu [143], gde su analizirane karaktristike skalabilnosti RYU i POX kontrolera kojima su definisani centralni upravljački uređaji Fat-Tree topologije. Fat-Tree topologija kao slojevita hijerarhijska struktura svičeva, rutera i računara, predstavlja vrlo često rešenje u cloud sistemima, mrežama provajdera ili centrima podataka, u kojima je mrežno upravljanje zasnovano na SDN konceptu. Dodatni razlog izbora ove topologije je taj da se u praksi izvršava veliki broj DDoS napada upravo na korporacijske mreže i centre podataka, u kojima je SDN arhitektura dominantna. Takođe, ovaj tip topologije karakteriše praktičnost i efikasnosti njene realizacije, kao i mogućnost da se napad detektuje u blizini njegovog izvora, gde je moguće primeniti neke dodatne mere zaštite. Za realna mrežna okruženja važi da je detekcija napada pouzdanija što je bliža njihovom izvoru, a tome doprinosi manji IP adresni prostor mrežnih čvorova, manji memorijski i procesorski resursi za praćenje mrežnog saobraćaja, jednostavnija primena mera reakcije na napade itd. Izvori napada uglavnom potiču iz javnih domena ili interneta, neposredno iza mesta njihove detekcije. Univerzitetske ili kampus mreže MAN (*Metropoliten Area Network*) su posebno ranjive na DDoS napade, jer imaju veliki broj korisnika i lokalnih mreža na relativno malom prostoru. Zbog svog modela otvorenosti, MAN mreže su laka meta za potencijalne napadače, koji imaju višestruke mogućnosti da u njima instaliraju DDoS botnet agente.

Na slici 7.3 prikazana je Fat-Tree topologija koja je u realizaciji rešenja izabrana kao simulaciona SDN mreža. Programski je formirana višeslojna SDN topologija, čiji se pojedini detalji razlikuju od topologije stabla kod klasičnih mreža. Topologija predstavlja višeslojnu hijerarhijska mrežnu strukturu koja se sastoji od sloja jezgra, kojeg čini centralni POX kontroler, sloja agregacije koju čine svičevi *Sw1-Sw4* i ivičnog sloja koji se sastoji od svičeva *Sw5-Sw12*. Svičevi su bazirani na OpenFlow protokolu i međusobno su povezani višestrukim linkovima, dok je svaki ivični svič povezan sa četiri hosta koji mogu imati ulogu servera.




**Slika 7.3:** Fat-Tree simulaciona SDN mrežna topologija

Izabrana Fat-Tree topologija ima specifičnu formu u kojoj je *n* portova svičeva (*n=8*) u ivičnoj ravni povezano sa *n/2* hostova. Preostalih *n/2* portova je povezano sa *n/2* svičeva u ravni agregacije. Osnovnu ćeliju topologije (*pod*) čine *n/2* agregaciona sviča, *n/2* ivična sviča i hostovi povezani sa njima. U standardnoj topologiji stabla, u ravni jezgra postoji $(n/2)^2$ svičeva sa *n* portova, pri čemu je svaki od njih povezan sa svakom od *n* ćelija, ali je za potrebe ove simulacije upotrebljena jednostavnija forma, koja u ravni jezgra ima jedan POX kontroler, koji je povezan sa svih 12 svičeva. DDoS napadi se izvršavaju sa hostova *h1* i *h2*, dok je kao cilj napada fizički najudaljeniji *h32* host. Dodatne prednosti izbora ove topologije su: isti protok podataka u svim delovima SDN mreže, svaka ravan ima isti protok, svaki port sviča podržava istu brzinu kao i krajnji host, kao i redundantnost i funkcija skalabilnosti jer topologija može da koristi $(n)^3/4$ hostova. Ograničenja upotrebe ove topologije postoje, mada se prvenstveno ogledaju u kompleksnosti međusobno povezivanih mrežnih uređaja, što nameće potrebu za prekomernim kabliranjem usled velikog broja svičeva i hostova.

Simulaciona topologija formirana je razvojem Python programskog kôda, što je omogućilo da se na relativno jednostavan način ona implementira u Mininet emulator. Mininet poseduje predefinisane mrežne profile topologija koji se jednostavno mogu koristiti ili modifikovati, ali je u ovom slučaju SDN topologija definisana parametrima u okviru --*custom* repozitorijuma Mininet emulatora. Python programski kôd simulacione Fat-Tree topologije prikazan je na slici 7.4.

```
1  # Custom FatTree topologija K=3
2
3  from mininet.topo import Topo
4
5  class FatTree( Topo ):
6
7      def __init__( self, half_ports = 2, **opts ):
8
9          # Default clanovi class
10         Topo.__init__(self, **opts)
11
12         aggrs = []
13         hnum = 0
14         snum = 0
15
16         # Aggregacioni svicevi
17         for i in range(half_ports):
18             snum += 1
19             aggrs.append(self.addSwitch('s%s' % snum))
20
21         # Top svicevi
22         for i in range(half_ports*2):
23             snum += 1
24             sw = self.addSwitch('s%s' % snum)
25
26             # Povezivanje Top sa Agregacionim
27             for j in range(half_ports):
28                 self.addLink(sw, aggrs[j])
29
30             # Formiranje hostova i linkova
31             for j in range(half_ports):
32                 hnum += 1
33                 host = self.addHost('h%s' % hnum)
34                 self.addLink(sw, host)
35
36 topos = { 'ftb-topology-dc': FatTree }
37
```

**Slika 7.4:** Python kôd za generisanje Fat-Tree mrežne topologije

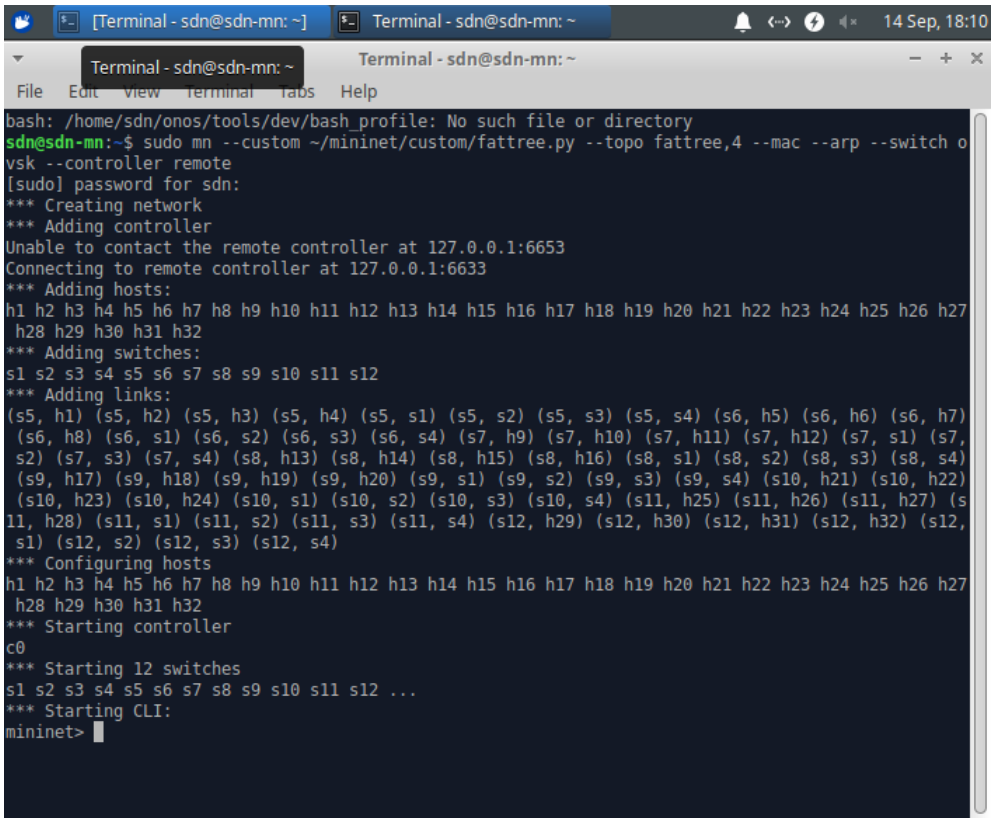
Zbog složenosti topologije i većeg broja svičeva, hostova i linkova, korišćen je API visokog nivoa, sa apstrakcijom mrežnih formi (šablona). U programskom kôdu *Topo* predstavlja osnovnu Mininet klasu za kreiranje parametarizovanih formata topologije (linija 3). U liniji 5 definisana je osnovna klasa. Postoji više načina kako se ova klasa može koristiti, ali je ovde definisana klasa kao podrazumevani host, a zatim su klase i konstruktori povezani sa Mininet emulatorom. U programskom kôdu *__init__* je rezervisana metoda i predstavlja konstruktor klase (linija 10). Promenljiva *self* predstavlja instancu klase i povezuje atribute topologije sa datim argumentima. Metode *self.addSwitch()*, *self.addLink()* i *self.addHost()* vrše unošenje svičeva i hostova u topologiju i njihovo međusobno povezivanje (linije 17-34). Linkovi topologije su u formi virtuelnih konekcija, mada je kôd apliakcija i mrežnog steka (uključujući TCP stek) potpuno identičan onom koji funkcioniše u realnom okruženju.

Inicijalizacija POX kontrolera je početni korak za programsko formiranje SDN topologije, i vrši se unošenjem Mininet CLI komande:

```
$ sudo pox/pox.py forwarding.l2_learning openflow.discovery --eat-early-
packets  openflow.spanning_tree --no-flood --hold-down
```

Nakon inicijalizacije kontrolera, izvršeno je importovanje kôda topologije u Mininet okruženje (modul *custom*). Na slici 7.5 prikazan je izgled Mininet interfejsa nakon generisane topologije, koji daje informacije o svim povezanim sviječevima, hostovima i linkovima, kao i o centralnom POX kontroleru. Importovanje kôda topologije u Mininet emulator vrši se unošenjem CLI komande:

```
$ sudo mn --custom ~/mininet/custom/ftb-topology-dc.py --topo fattree,4 --
mac --arp --switch ovsk --controller remote
```




**Slika 7.5:** Fat-Tree topologija u Mininet mrežnom emulatoru

Sviječevi u Fat-Tree topologiji prosleđuju saobraćaj na osnovu OpenFlow 1.3 protokola, i svakom od njih dodeljen je jedinstven port. Mininet automatski dodeljuje jedinstvene MAC adrese hostovima, dok im se IP adrese dodeljuju iz opsega privatnih adresa (10.0.0.0/24). Kombinacija IP/MAC adresa za host *h1* je (10.0.0.1/00:00:00:00:00:01), za host *h2* (10.0.0.2/00:00:00:00:00:02) itd. Sviječevi su sa POX kontrolerom povezani putem virtuelne loopback IP adrese 127.0.0.1, pri čemu kontroler za komunikaciju koristi port 6633.

Zbog velikog broja linkova u simulacionoj topologiji, dolazi do mogućnosti stvaranja intenzivnih emisionih oluja (*bradcast storms*), koje dovode do značajnog smanjenja mrežnih performansi. Moguć je nastanak višestrukih mrežnih petlji i beskonačnog kruženja paketa kroz mrežu, usled dobijanja istog frejma na različitim portovima sviča i neprestanog ažuriranja njegove MAC tabele. Protokol razapinjućeg stabla STP (*Spanning-Tree Protocol*) vrši nadgledanje mreže kako bi pronašao sve veze i sprečava pojavu ovih petlji blokiranjem redundantnih linkova [144]. STP algoritam formira logičku strukturu stabla od korena (*root*) do svih čvorova u SDN mreži, i kao takav ugrađen je kao Python skript u POX kontroler. STP funkcionalnost je omogućena OpenFlow protokolom, podrazumevano je deo kontrolerskog softvera i navodi se *$openflow.spanning_tree --no-flood* komandom tokom inicijalizacije POX kontrolera.

## 7.4. **Modul za detekciju entropije mrežnog saobraćaja**

Teorijske osnove entropije predstavljene u poglavlju 6.1 su bile polazna tačka razvoja modula za detekciju mrežnih anomalija. Pri tome je trebalo uzeti u obzir činjenicu da se entropija, kao važna metrika za analizu mrežnog saobraćaja, zasniva na verovatnoći da će svaki sledeći paket biti upućen prema istom cilju, tj. istoj IP adresi i da je svaki mrežni čvor moguće odredišta paketa. Pošto zaglavlje mrežnog paketa sadrži jednu IP adresu pošiljaoca i primaoca, različiti ishodi tokom DDoS napada su međusobno zavisni, tj. promena broja paketa u jednom mrežnom čvoru dovodi do smanjenja broja paketa u preostalim čvorovima. Apsolutna vriednost broja paketa nema veliki značaj za promenu vrednosti entropije, ali je relativni odnos između konstantnog broja primalaca paketa jako važan. Ova činjenica značajno otežava bilo kakvu detekciju napada, jer metrika uzima u obzir samo broj paketa. Promenom vrednosti entropije moguće je jednostavno detektoavti intenzivno povećanje broja paketa upućenih ka malom broju primalaca. Za razvoj algoritma za detekciju anomalija i analizu entropije u ovom radu, upotrebljeni su detalji istraživanja prikazani u [145].

U modulu za detekciju anomalija, entropija upotrebljena kao ocena neizvesnosti slučajne promenljive, čija se promena odražava na karakteristike mrežnog saobraćaja. Pored određivanja promene entropije odredišnih IP adresa, algoritam koristi proračun mrežnih tokova kako bi doneo odluku o detektovanom napadu. Ako se pretpostavi da skup $X=(X_1, X_2,$

*..., $X_m$)* predstavlja prostor odredišnih IP adresa ivičnog sviča, tada je $X_i$ broj paketa sa jednom odredišnom IP adresom za period detekcije $\Delta t$. Verovatnoća da će paket imati istu određenu odredišnu IP adresu tokom perioda detekcije može se odrediti prema izrazu (7.1):

$$p_i = \frac{X_i}{\sum_{i=1}^{m} X_i} \tag{7.1}$$

Pod pretpostavkom da je legitiman mrežni saobraćaj uvek prisutan, i da je broj aktivnih mrežnih čvorova $m$ uvek veći od jedan, normalizovana entropija odredišne IP adrese se može izračunati prema izrazu (7.2):

$$H(X) = \frac{-\sum_{i=1}^{m} p_i \, log(p_i)}{\log(m)} \tag{7.2}$$

Algoritam koristi dve vrednosti entropije, za dva stanja mreže. Prva vrednost $H_n(X)$ predstavlja entropiju za legitiman mrežni saobraćaj, dok je $H_a(X)$ vrednost entropije izračunata tokom trajanja DDoS napada. U situaciji legitimnog stanja mreže, promene vrednosti entropije su relativno male. Tokom DDoS napada, mrežni saobraćaj sa istom odredišnom IP adresom će se naglo povećati, što će rezultovati manjom vrednošću entropije, pa se kao kriterijum za detekciju anomalija koristi razlika entropija, data izrazom (7.3):

$$H_n(X) - H_a(X) > \theta \tag{7.3}$$

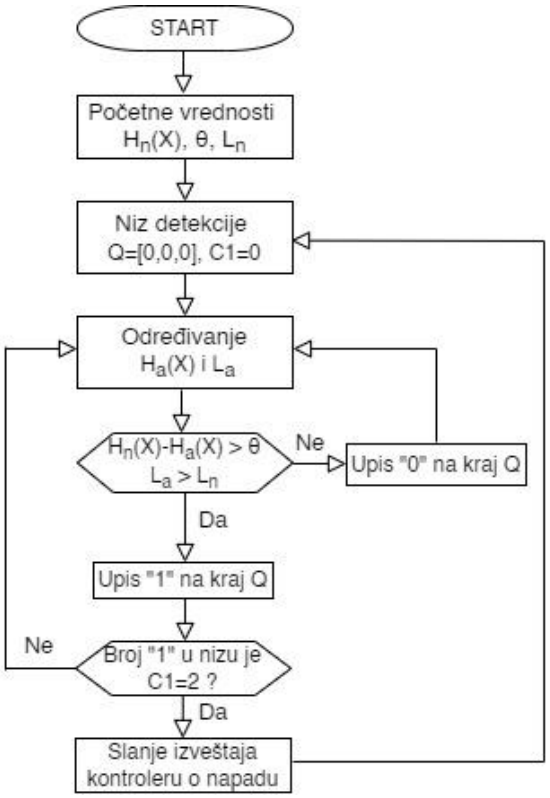
Vrednost $\theta$ određena je prema statistici za entropiju u uslovima legitimnog statusa mreže, tj. kada ne postoji DDoS napad. Tokom TCP-SYN i ICMP flood napada, generiše se veliki broj paketa sa lažnim IP adresama, pa će promena entropije određene odredišne IP adrese na ivičnom sviču biti značajno povećana. Drugi uslov za detekciju anomalija koji algoritam koristi jeste broj mrežnih tokova sviča $L$. Za broj tokova legitimnog saobraćaja $L_n$, uslov za detekciju anomalije u mreži se računa izrazom (7.4):

$$L > L_n \tag{7.4}$$

Vrednost za $L_n$ je određena na osnovu maksimalnog broja tokova na sviču tokom legitimnog statusa mrežne topologije i ovaj parameter uzima u obzor broj prenetih *Packet_In* poruka koje ivični svič razmenjuje sa kontrolerom. Podaci o broju ovih poruka upotrebljeni su kao obučavajući skup modela mašinskog učenja u okviru modula za klasifikaciju napada.

Algoritam za detekciju mrežnih anomalija prikazan je na slici 7.6. Izvršava se na ivičnom sviču *Sw8* simulacione toplogije. Algoritam vrši detekciju napada poređenjem vrednosti entropije odredišnih adresa i broja tokova za legitimni saobraćaj i za slučaj mrežnog

napada. Cilj algoritma je da se iskoriste hardverski resursi sviča koji će procesirati mrežne podatake i izvršiti proračun entropije na što efikasniji način, kako bi se postigla što brža detekcija anomalija i SDN kontroler oslobodio dodatnog procesiranja mrežnih podataka.




**Slika 7.6:** Algoritam detekcije entropije na ivičnom sviču

Nakon definisanja vrednosti entropije i broja tokova sviča za legitiman saobraćaj, algoritam nadgleda saobraćaj definisanog mrežnog domena u realnom vremenu, i pravovremeno obaveštava kontroler o detektovanim anomalijama. Kontoler za svaki svič u mreži definiše tabelu tokova na osnovu globalnog pregleda postojeće mrežne topologije. Takođe, kontroler šalje svičevima instrukcije o ažuriranju njihovih tabela tokova, kao i proračun optimalne putanje podataka do odredišnih uređaja. Fluktuacije u kašnjenju pojedinačnih mrežnih paketa između izvornog i odredišnog čvora (*jitter*), kao i latencija mrežnog saobraćaja mogu biti u opsegu nekoliko destina do nekoliko stotina milisekundi, što je analizirano u više istraživanja [146]. Da bi se izbegla pomenuta kašnjenja, entropija se izračunava za male podintervale, tipično od 0.1 sec. Za interval od *1 sec*, vremenski prozor sadrži *M* podintervala (tipično 10), tako da je $\Delta t=1s$. Veličina prozora je značajan parametar

entropije, jer bi kratki događaji mogli ostati neprimećeni ako bi se izvršili tokom jednog podintervala. To je način kako da se smanje kašnjenja, jer se idirektno vrši usrednjavanje rezultata. Ovaj princip se koristi u sličnom istraživanju [147], u kojem se za svaki podinterval formira raspodela odredišnih IP adresa kako bi se izračunala entropija.

Algoritam koristi generisan niz $Q$ dužine 3, za određivanje statusa detekcije. U slučaju kada se detektuje razlika entropije veća od vrednosti $\theta$ i broj tokova veći od graničnog $L_n$, vrši se upis "1"na kraj niza. U slučaju da ovi zahtevi nisu ispunjeni, na kraj niza $Q$ se upisuje "0". Postojanje dva upisa "1" u okviru niza $Q$, indikator je nastanka anomalije u mreži. U tom trenutku, ivični svič odmah šalje izveštaj kontroleru, a niz $Q$ se resetuje. Kao poslednji korak, ako je sumnjivi mrežni saobraćaj potvrđen kao napad, svaki takav tok podataka se prekida i vrši se ažururanje tabele tokova sviča sa kojeg je takav saobraćaj došao.

## 7.5. Modul mašinskog učenja za klasifikaciju napada

### 7.5.1. Generisanje mrežnog saobraćaja

Nakon formiranja simulacione topologije, sledeći korak je bio generisanje mrežnog saobraćaja za scenario regularnog, kao i za slučaj DDoS napada. DDoS napadi su inicirani sa hostova $h_1$ i $h_2$, a preko izlaznog porta ivičnog sviča *Sw8* prosleđeni su ka odredišnom hostu $h_{32}$ koji predstavlja cilj napada. Simulacija je izvršena za TCP-SYN i ICMP flood napade. Za generisanje legitimnog saobraćaja, kao i za saobraćaj napade korišćen je Scapy softver [148, 149]. Scapy omogućava interaktivno kreiranje i manipulaciju mrežnim paketima. U mrežnim istraživanjima uglavnom se koristi za slanje paketa i osluškivanje mrežnog saobraćaja, ali i za složenije zadatke poput skeniranja mrežnih portova ili identifikaciju operativnog sistema na udeljenim računarima. Takođe, može se koristiti za simulaciju DDoS saobraćaja na infrastrukturnoj i ravni aplikacija. Scapy predstavlja programski modul za Python jezik, što znači da njegov programski kôd sadrži gotove funkcije koje se mogu koristiti preko Python interpretera. Ove funkcije se koriste zajedno sa standardnim Python funkcijama, tako da se mrežni saobraćaja može relativno jednostavno sooftverski generisati.

Na slici 7.7 prikazan je deo Python kôda kojim je generisan napad koji je preko hostova *h1* i *h2* injektovan u simulacionu topologiju. Ovaj tip napada je korišćen u prvom delu metode detekcije, pri izvršavanju algoritma detekcije anomalija. Pošto jedna grupa TCP-SYN napada koristi lažne IP adrese, prvo se formira slučajno izabrana IP adresa iz opsega mogućih adresa, isključivši opsege privatnih adresa. Klasifikacioni modeli u ovom istraživanju koriste analizu tačnosti za različite intenzitete napada. Intenzitet napada se definiše kao odnos broja paketa napada $N_{attack}$ i ukupnog broja prenetih paketa $N_{total}$ u određenom vremenskom intervalu i izračunava se na osnovu izraza (7.5):

$$Attack_{intensity} = \frac{N_{attack}}{N_{total}} \times 100 \qquad (7.5)$$

Za napad intenziteta od 25%, interval legitimnog saobraćaja je 0,1 sec, dok je interval napada 0,025 sec. To u simulaciji predstavlja slučaj kada se u okviru vremenskog intervala od ukupnog broja, šalje 25% paketa napada ka odredišnom hostu. U klasifikacionim algoritmima biiće analizirani scenariji kada su intenziteti napada 5%, 25% i 50%.

```
30  def main():
31    for i in range (1,5):
32      mymain()
33      time.sleep (10)
34  #slanje generisanih IP adresa
35  def mymain():
36
37  # dobijanje IP adrese na koju se šalju paketi napada ...
38    dstIP = sys.argv[1:]
39    print dstIP
40    src_port = 80
41    dst_port = 1
42
43  # slanje paketa preko porta eth0
44    interface = popen('ifconfig | awk \'/eth0/ {print $1}\'').read()
45
46    for i in xrange(0,500):
47  # formiranje paketa
48      packets = Ether()/IP(dst=dstIP,src=sourceIPgen())/UDP(dport=dst_port,sport=src_port)
49      print(repr(packets))
50
51  # interval slanja paketa (sec)...25%
52      sendp( packets,iface=interface.rstrip(),inter=0.025)
53
54  #main
55
56
57  if __name__=="__main__":
58    main()
```

**Slika 7.7:** Deo programskog kôda za generisanje DDoS napada

### 7.5.2. Skupovi podataka simulacione topologije

Pored postojanja legitimnog mrežnog saobraćaja u simulacionoj topologiji, bilo je potrebno formirati reprezentativne skupove podataka (*datasets*) koji će predstavljati određene tipove DDoS napada. Na osnovu dostupne literature, takvi skupovi podataka su osnova većine rešenja za detekciju napada. Potreba za ovim skupovima podataka proizilazi iz prirode mašinskog učenja, koje koristi definisane skupove podataka kojima se model obučava da prepozna tip mrežne aktivnosti. Za analizu performansi modela nadgledanog mašinskog učenja u drugom modulu predložene metode detekcije napada, bilo je neophodno obezbediti adekvatne skupove podataka koji bi predstavljali primere DDoS napada preplavljivanjem ili njihove varijacije. Predloženi metod detekcije je realizovan i analizirane su njegove performanse uz pomoć dva skupa podataka:

- skupova podataka simulacione topologije
- javno dostupnih skupova podataka

Vrenosti atributa polja u zaglavlju TCP/IP paketa (izvorna/odredišna IP, MAC ili broj porta, entropija izvornih/odredišnih IP adresa, TCP flegovi…) se izdvajaju iz mrežnog saobraćaja putem različitog softvera (tcpdump, Wireshark, Tshark itd.). Ovi podaci se izdvajaju kao skupovi podataka posebno za legitiman, i za saobraćaj napada. Zatim se od ovih podataka formira .csv (*comma-separated values*) fajl koji se dalje koristi kao obučavajući skup u modelu mašinskog učenja. Algoritmi mašinskog učenja koji koriste podatke prikupljene iz simulacione mreže, karakteriše veća brzina izvršavanja i jednostavnost analize, ali oni poseduju i izvesna ograničenja. Ovi skupovi obično sadrže malu količinu podataka, što značajno utiče na tačnost predikcije modela mašinskog učenja. Takođe, oni sadrže attribute malog broja tipova napada, što je u suprotnosti sa stvarnim stanjem u mrežama, koje karakteriše njihova velika raznolikost. Mali broj tipova napada može pomoći napadačima da saznaju kakvo je normalno ponašanje mehanizma detekcije napada, tako da mogu kreirati napad koji će replicirati ovo ponašanje. Mali broj izdvojenih atributa iz zaglavljla TCP paketa često je nedovoljan za opis ponašanja različitih tipova napada. Ograničenje predstavlja i to što se navedeni atributi izdvajaju isključivo iz zaglavlja paketa, bez detaljnije analize korisnih podataka (*payload*). Polja zaglavlja paketa moguće je jednostavno modifikovati tako da liče na polja legitimnog mrežnog saobraćaja. Napadač može ugraditi maliciozni kôd unutar korisnog dela paketa kako bi pokrenuo R2L (*Root to Local*) napad, kojim je moguće iskoristiti

ranjivosti operativnih sistema ili softvera radi sticanja sistemskih administratorskih (*root*) ovlašćenja. Primeri R2L napada su npr. prelivanje bafera (*buffer overflow*), rootkit i SQLAttack napadi.

Skup podataka simulacione Fat-Tree topologije formiran je kroz nekoliko faza. Prvi korak je zahtevao formiranje virtuelne simulacione toplogije, a zatim su u okviru nje generisana dva tipa mrežnog saobraćaja, legitimni i saobraćaj napada. Prvi tip mrežnog saobraćaja je regularni TCP saobraćaj maksimalnog mrežnog protoka u trajanja od 50 sec. Drugi tip saobraćaja je kratkotrajni napad preplavljivanjem, intenziteta 1000 paketa/sec, koji se šalju na svakih 5 sec. Tokom legitimnog statusa mreže, host *h1* šalje ka hostu *h32* samo legitimni TCP saobraćaj maksimalnog mrežnog protoka. Tokom napada, sa hosta *h2* šalju se periodični paketi napada ka hostu *h32*. Lažne IP adrese dovode do izvršavanja DDoS mehanizma napada i formiranja velikog broja *Packet_In* poruka koje svičevi prosleđuju POX kontroleru u kratkom vremenskom periodu.

Analizom podataka dobijenih iz simulacione mreže, pošlo se od pretpostvake da je moguće formirati jednostavan skup podataka sa malim brojem mrežnih atributa, i na taj način pojednostaviti proces njihovog prikupljanja, analize i pretprocesiranja, što je i zahtev modela mašinskog učenja. Pošto se nadgledano mašinsko učenje izvršava kroz procese obučavanja i testiranja, izbor skupa podataka dobijenih iz simulacione mreže je važan korak, jer značajno utiče na performanse modela klasifikacije. Jednostavnost rešenja je postignuta korišćenjem isključivo *Paket_In* poruka, koje predstavljaju osnovu komunikacije između svičeva i kontrolera, a posebno u slučajevima izvršavanja napada. Fluktuacija ili značajna promena broja ovih poruka primljenih sa svičeva u određenom vremenskom intervalu opisuje ponašanje mrežnih tokova i indikator je nastanka mrežnih anomalija. Da bi se ovakav skup podataka kreirao, potrebno je izvršiti monitoring, analizu i prikupljanje *Packet_In* poruka za oba tipa mrežnog saobraćaja. Za ovaj deo rešenja, upotrebljen je programski kôd modela mašinskog učenja prezentovan u [150], pri čemu je on prilagođen Fat-Tree simulacionom okruženju i mrežnim uslovima, kako bi se *Packet_In* poruke prikupile sa svih mrežnih svičeva, a klasifikacija saobraćaja izvršila na POX kontroleru.

POX kontroler analizom saobraćaja prikuplja podatke, tako što formira liste pristiglih *Packet_In* poruka sa svičeva, u okviru definisanog vremenskog intervala (slota). Podaci se prikupljaju za oba mrežna statusa. Liste *Packet_In* poruka kreiraju se tokom 30 vremenskih slotova, trajanja 10 sec. pa se celokupni skup podataka formira tokom perioda od 300 sec.

Svaka lista koju formira kontroler sadrži upisane brojeve primljenih *Packet_In* poruka, i nakon 30 unosa vrši se resetovanje liste i unos novih podataka. Nakon formiranja listi podataka, kontroler prelazi na proces formiranja uzoraka u okviru test-ciklusa. Kontroler na osnovu prethodnih koraka i putem formiranih listi generiše uzorke skupa podataka, a svaki uzorak sadrži varijaciju broja *Packet_In* poruka. Svaki uzorak u obučavajućem skupu reprezentuje ponašanje mrežnih uređaja tokom vremena. Osnovni podatak uzorka je broj *Packet_In* poruka koje su primljene preko MAC adresa mrežnih uređaja tokom vremenskog slota. Svaki uzorak, pored broja *Packet_In* poruka nosi podatak o identifikatoru klase kojoj pripada, pa se na taj način označavaju legitimni uzorci ili uzorci za status napada. Svaki uzorak sadrži 31 promenljivu, 30 brojeva paketa (listi) i identifikator klase. Za analizu uticaja broja uzoraka u obučavajućem skupu na tačnost detekcije modela mašinskog učenja, formirani su skupovi podataka koji sadrže respektivno 100, 200, 400, 800, 1600, 3200, 6400, 12800 i 25600 uzoraka. U sekciji 8.2.2 izvršena je uporedna analiza tačnosti svih pet klasifikatora nadgledanog mašinskog učenja, za obučavajući skup od 25600 uzoraka.

Pošto svaki uzorak sadrži broj primljenih *Packet_In* poruka, on je osnovni i jedini atribut koji definiše podatke za model mašinskog učenja. Nakon formiranja klasifikovanih grupa uzoraka, faza predprocesiranja podrazumeva uklanjanje njihovih dupliranih ili praznih polja. Nakon toga vrši se podela uzoraka na obučavajući i test skup u odnosu (80:20)%, što je standarda podela u većini rešenja zasnovanih na mašinskom učenju. Nakon što se izvrši obučavanje klasifikatora sa odabranim uzorcima, preostali uzorci se koriste za evaluaciju rezultata.

### 7.5.3. Javno dostupni skupovi podataka

Većina novijih rešenja za detekciju napada zasniva se na korišćenju poznatih i javno dostupnih skupova podataka. Ovi skupovi podataka (poznati i kao sintetički), nastali su u okviru istraživanja sistema za detekciju napada, i sadrže tokove mrežnog saobraćaja sa većim brojem označenih atributa koji predstavljaju različite scenarije napada, a koji su prikupljeni u određenom vremenskom periodu od nekoliko dana ili nedelja. Javni skupovi podataka mogu da sadrže podatke generisane DoS/DDoS napadima, napadima grubom silom, skeniranjem, botnet napadima, infiltracijom itd. Ipak, najveći broj ovih skupova ne predstavlja realno

mrežno stanje, je im nedostaje raznolikost u tipovima napada čije podatke sadrže. Pošto ovi skupovi često čine mrežne podatke provajdera, prikupljene u određenom vremenskom periodu, postavlja se pitanje privatnosti i pravnih normi njihovog objavljivanja i upotrebe. Ipak, izbor javnih skupova podataka treba pažljivo razmotriti, jer najveći broj njih ne daje prihvatljivu tačnost kada se koriste u modelima mašinskog učenja. U najvećem broju rešenja za detekciju napada, koriste se sledeći javni skupovi podataka: KDDCUP'99, CICIDS2017, Kyoto, LBNL, UMASS, UNSW-NB15, ADFA, DEFCON [151-154] itd. Godinama unazad istraživanja u oblasti mrežne bezbednosti bazirala su se na upotrebi KDD99 skupa podataka. Tokom vremena on je zamenjen novijim i sveobuhvatnijim skupovima podataka, koji su rešili probleme zastarelosti mrežnih atributa i klasa napada koji su njima obuhvaćeni, zatim probleme nebalansiranosti prisustva instanci pojedinih napada itd. CICIDS-2017 i InSDN su javni skupovi podataka koji su u ovom radu upotrebljeni za evaluaciju performansi modula mašinskog učenja.

CICIDS-2017 predstavlja javno dostupan skup podataka, koji sadrži instance legitimnog saobraćaja i saobraćaja većine savremenih tipova napada. Ovaj skup podataka može se slobodno preuzeti sa kanadskog CIC instituta za sajber bezbednost [155]. Bazira se na na dvosmernim mrežnim tokovima sa po 80 različitih atributa koji u značajnom obimu definišu realan mrežni saobraćaj. CICIDS-2017 sadrži označene podatake koji reprezentuju scenarije savremenih napada, uključujući napade grubom silom, DoS/DDoS, botnet, napade skeniranjem, Web napade i napade infiltracijom. Ovaj skup emulira niz različitih mrežnih procesa koji koriste HTTP, HTTPS, FTP i SSH komunikacione protokole. Mrežni saobraćaj je prikupljan tokom pet radnih dana, pri čemu mrežni tokovi prikupljeni ponedeljkom označavaju legitimni saobraćaj. Pošto ovaj rad razmatra mehanizme DDoS napada, upotrebljen je mrežni saobraćaj prikupljen tokom petka, a koji se nalazi u okviru *Friday-WorkingHours.pcap* foldera, i koji simulira instance DDoS napada. Podaci su u *.csv* formatu, kako bi bili prepoznatljivi za algoritme mašinskog učenja.

InSDN je javno dostupan skup podataka, koji je delimično rešio problem nedostatka adekvatnih skupova podataka koji su dobijeni direktno iz SDN mreža. Predstavljen je u radu [156]. Specifičnost ovog skupa podataka je što sadrži instance napada koji se izvršavaju u kontrolnoj ravni, kao i u ravni podataka SDN mreže. Klase napada generisane su u Mininet virtuelnom SDN okruženju, koje je formirano od četiri VMware hipervizora i jednog ONOS kontrolera. InSDN skup podataka je za ovo istraživanje izabran jer sadrži scenarije DDoS

napada preplavljivanjem, kao što su TCP-SYN, UDP ili ICMP napadi. Tcpdump softverom izvršeno je prikupljanje podataka o mrežnim tokovima za svaku kategoriju saobraćaja, preko porta SDN kontrolera. Za ekstrakciju atributa tokova u ovom skupu podataka, upotrebljen je CICFlowMeter softver. InSDN skup podataka sadrži više od 80 statističkih atributa tokova, svrstanih u 56 kategorija. Ukupan broj instanci podataka je 343.939, od čega 68.424 predstavlja legitimni saobraćaj, dok broj instanci napada iznosi 275.515. Skup podataka sadrži 73.529 instanci DDoS napada, koje su inicijalno upotrebljene u delu ovog rada koji se odnosi na klasifikaciju napada algoritmima mašinskog učenja.

### 7.5.4. Softver za razvoj predloženog rešenja

Za razvoj metode detekcije naapada u ovoj disertaciji korišćen je različit softver, ali je celokupan programski kôd pisan u Python jeziku i Spyder razvojnom okruženju. Python svakako spada u grupu najpopularnijih programskih jezika koji se koristi u oblastima mašinskog učenja, analize podataka, velikih podataka, a sve veću primenu ima i u oblasti programabilnih računarskih mreža. U oblasti SDN mreža, Python predstavlja osnovni programski jezik, jer je u njemu pisan kôd za OpenFlow protokol, kao i kôd za većinu kontrolerskih operativnih sistema i aplikacija. Python je u ovom istraživanju korišćen kroz nekoliko modula u okviru Spyder IDE okruženja. Pandas je softverski paket prvenstveno namenjen za rad sa podacima. Intenzivno se koristi za naučno-istraživački rad, jer se lako integriše u različite simulacione modele. Primenjuje se u radu sa tabelarnim podacima, podacima sa vremenskim serijama, matricama i različitim formama podataka koji se dobijaju statističkim merenjima. U ovom istraživanju primenjen je za tabelatni prikaz formiranih uzoraka *Packet_In* poruka simulacione mreže. Spyder IDE (*Integrated Development Environment)* je open-source softverska platforma, inicijalno razvijena od strane MIT Instituta. Ovaj softver je izabran za istraživanje zbog jednostavnosti primene i podrške za većinu važnijih algoritama nadgledanog učenja. Omogućava rad sa različitim formatima podataka, kao i za određivanje njihove strukture i karakteristika. Spyder IDE ima različite mogućnosti za obrade podataka: pretprocesiranje, klasifikaciju, klasterovanje, primenu asocijativnih pravila itd. U okviru ovog rada, upotrebljen je za rad sa atributima u eksperimentalnom delu istraživanja, kao i za grafički prikaz dobijenih rezultata.

# 8. REZULTATI I ANALIZA PREDLOŽENE METODE

U ovom poglavlju su prikazani i analizirani rezultati dobijeni simulacijom SDN mrežne topologije. Izvršeno je nekoliko eksperimenata, za različite tipove DDoS napada, a zatim je sprovedena evaluacija predloženog rešenja za detekciju napada i klasifikaciju mrežnog saobraćaja. U prvom delu poglavlja razmatrani su rezultati modula detekcije anomalija na ivičnom sviču, kroz analizu parametara brzine detekcije, vrednosti entropije i iskorišćenja hardverskih resursa kontrolera. Zatim su analizirani rezultati u kontekstu klasifikacije i predikcije napada, pa je poređenje rezultata izvršeno za nekoliko skupova podataka i algoritama nadgledanog mašinskog učenja.

## 8.1. Rezultati i analiza modula za detekciju entropije i anomalija

Osnovni cilj metode detekcije u ovom istraživanju je kako postići što veći procenat uspešnosti detekcije DDoS napada. Bilo je potrebno detaljno proučiti faktore koji u većoj ili manjoj meri utiču na nivo uspešnosti. Kako bi se postigla što veća tačnost predloženog algoritma detekcije, trebalo je jasno definisati parametre koji određeni mrežni saobraćaj označavaju kao anomalije tj. napade. Za DDoS napade koji se zasnivaju na preplavljivanju, efikasnost detekcije zavisi prvenstveno od performansi odredišnih uređaja, tako da definisanje praga entropije koji će biti granica između legitimnog i saobraćaja napada često predstavlja zahtevan zadatak. Mada bilo koji neželjeni saobraćaj treba tretirati kao napad, pojedini napadi će na strani odredišnog uređaja biti obrađeni i odbačeni, čak i u slučaju da ne budu prepoznati od strane mrežnih procesnih elemenata. Za odredišne uređaje manjih hardverskih mogućnosti, čak i malo povećanje intenziteta legitimnog saobraćaja može dovesti do prekida rada njihovih servisa. Ovo je posebno bitno za funkcionisanje OpenFlow svičeva. Grupe podataka koje se koriste za testiranje napada, koriste označene maliciozne mrežne pakete, kako bi se naknadno

mogla izvršiti efikasna evaluacija dobijenih rezultata. Za volumetrijske DDoS napade važi da je teže sprovesti analizu graničnih slučajeva detekcije, jer se ovi napadi mogu u potpunosti sastojati od legitimnog saobraćaja. Za potrebe testiranja rešenja u ovom radu, generisani su specifični napadi koji se značajno izdvajaju iznad pozadinskog mrežnog saobraćaja. U rešenjima dostupnim u literaturi, kao primeri napada koriste se veći i dugotrajniji napadi, što dovodi do toga da detekcija nije dovoljno osetljiva. Predložena metoda detekcije je realizovana na taj način, da se tokom jednog simulacionog eksperimenta može detektovati veći broj događaja, na osnovu kojih se mogu odrediti brojne vrednosti ili procenti tačnih odnosno pogrešnih detekcija. Rešenje je moguće prilagoditi za kratke usmerene napade, jer se promenom brzine detekcije ovi napadi mogu uspešno detektovati, dok bi u legitimnim mrežnim uslovima bili bi neotkriveni.

Modul za detekciju entropije i anomalija je procesorski i memorijski zahtevan postupak, naročito u kompleksnim topologijama sa velikim mrežnim protokom. Detekcija se zasniva na analizi paketa, i svič ih procesira redom onako kako pristižu preko ulaznog porta. Da bi se smanjili hardverski zahtevi za procesiranje paketa, primenjena je tehnika uzorkovanja, gde se paketi analiziraju u tačno određenim vremenskim intervalima ili prozorima. Iako se na taj način donekle smanjuje tačnost detekcije, formiran je modul sa zadovoljavajućim nivoom pouzdanosti detekcije. Modul detekcije koristi vrednost prozora od $\Delta t=1\ sec.$, tako da prikupljene instance tokova saobraćaja formiraju vremenski tok vrednosti entropija svakog atributa, pri čemu svaka vrednost odgovara izračunatoj entropiji u jednom vremenski definisanom prozoru. Pojam brzine detekcije povezan je sa funkcionisanjem kontrolera, OpenFlow svičeva, kao i OpenFlow protokola, o čemu je razmatrano u poglavlju 3. Na osnovu specifikacije OpenFlow protokola, kontroler omogućava programabilnost OpenFlow svičeva instaliranjem predefinisanih aktivnosti koje se izvršavaju na njima. Međutim, kada aktivnost sviča nije dovoljna da procesira sve dolazne pakete, on ih prosleđuje kontroleru kao *Packet_In* poruke. Ovaj tip poruka dozvoljava svičevima da sprovode određene akcije nad paketima koje nisu predviđene OpenFlow specifikacijama. Vrlo čest problem predstavlja koncentracija *Packet_In* poruka koje se šalju od sviča ka kontroleru. Svič nakon prijema paketa traži njihovo poklapanje sa pravilom prosleđivanja u svojoj internoj tabeli. Ako takvo poklapanje ne postoji, podrazumevanim pravilom se paket šalje kontroleru putem *offp_controller* porta. Na kraju se svi dolazni paketi enkapsuliraju kao OpenFlow poruke i šalju kontroleru u formi predefinisanih *Packet_In* poruka.

Najveći broj radova koji se bave proučavanjem uticaja DDoS napada na SDN mreže, fokusiran je na kontrolnu ravan i kontroler. Ovaj deo istraživanja bio je usmeren ka ravni podataka, i uticaju napada na SDN svičeve i krajnje uređaje i predstavljao je pokušaj da se celokupan proces detekcije anomalija izvrši na svičevima, kako bi se dalja procedura predikcije napada mašinskim učenjem usmerila direktno na kontroler. U istraživanju su namenski generisani napadi na određene ciljeve, a cela simulacija je realizovana Mininet emulatorom, jer on olakšava rad sa višestrukim mrežnim parametrima virtuelnih svičeva i rad sa tabelama prosleđivanja paketa svičeva. Takođe, Mininet obezbeđuje kompatibilnost sa OpenFlow protokolom, funkcionalnosti L3 mrežne ravni i lakoću migracije simulacionog kôda u realno mrežno okruženje.

Postupak analize modula za detekciju anomalija sproveden je kroz nekoliko faza, u okviru kojih su analizirani sledeći mrežni parametri:

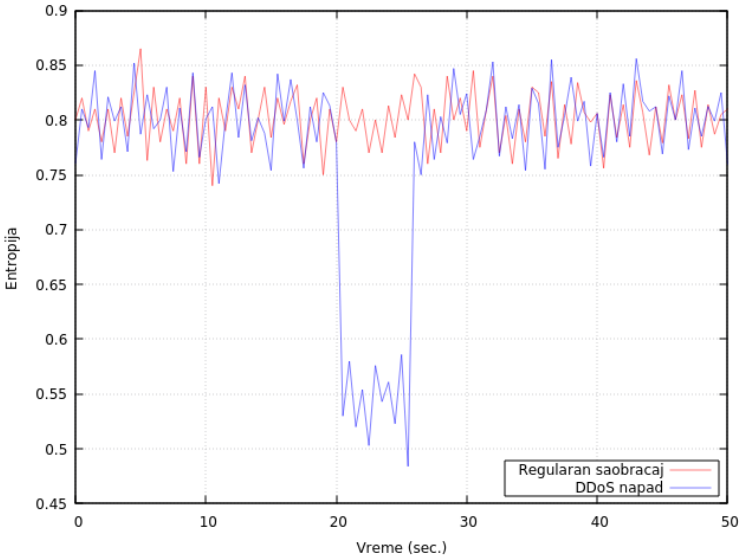
- broj mrežnih tokova i *Packet_In* poruka ivičnog sviča
- određivanje razlike entropije odredišnih IP adresa
- brzine detekcije anomalija

**Tabela 8.1:** Specifikacije modula za detekciju anomalija

| Parametar | Tip/Vrednost |
|---|---|
| Tip napada (flooding) | TCP-SYN, ICMP |
| Intenzitet napada | 1000 paketa/s |
| Trajanje napada | 5 sec. (od 20-25 sec.) |
| Entropija regularnog saobraćaja | 0.8 |
| Koeficijent razlike entropije $\theta$ | 0.2 |
| Vremenski prozor | 1 sec. |

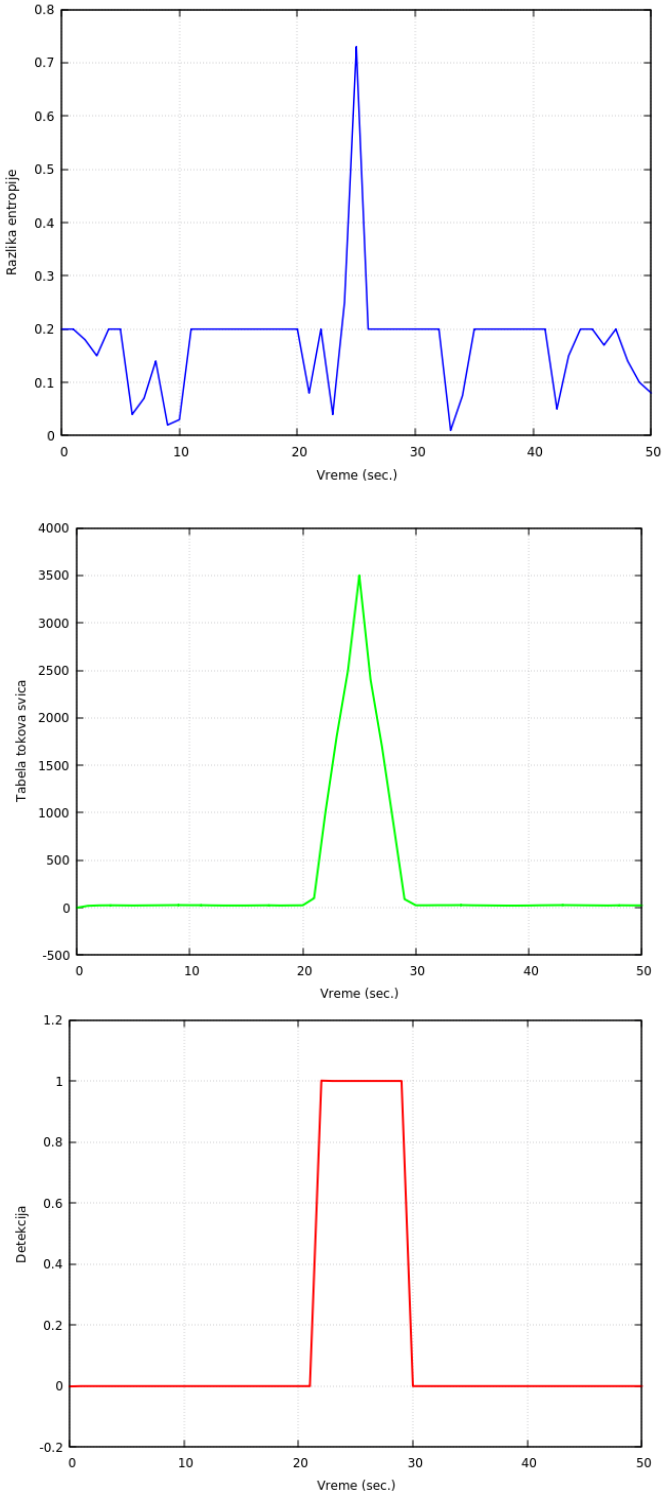
Specifikacije simulacije i modula detekciije napada prikazane su u tabelama 7.1 i 8.1. Eksperimentalna Fat-Tree topologija sadrži POX kontroler, 12 OpenFlow svičeva i 32 hosta. Napad se izvršava sa hosta *h2*, preko ivičnog sviča *Sw8*, koji u topologiji ima ulogu sviča na kojem se izvršava algoritam detekcije anomalija. DDoS napadi su realizovani generiisanjem TCP-SYN i ICMP saobraćaja, i analizirana je detekcija anomalija za ova dva specifična tipa napada preplavljivanjem. Napadi su formirani kao kratki periodični napadi u trajanju od 5 sekundi, od 20 do 25 sekunde od početka regularnog mrežnog saobraćaja. Trajanje simulacije je 50 sec. Cilj napada je host *h32*. Putem hosta *h1* generisan je legitimni mrežni saobraćaj.

Entropija je u prvom koraku određena za ovaj tip saobraćaja i varijacije njenih vrednosti tokom simulacije su prikazane na slici 8.1. Na osnovu ovih promena, određena je vrednost 0.8 za entropiju legitimnog saobraćaja, dok je za prag entropije usvojena vrednost 0.2. Ova vrednost je određena na osnovu uslova entropije algoritma za detekciju anomalija, koji inicijalno izračunava razliku entropija odredišnih IP adresa za legitiman i saobraćaj napada.

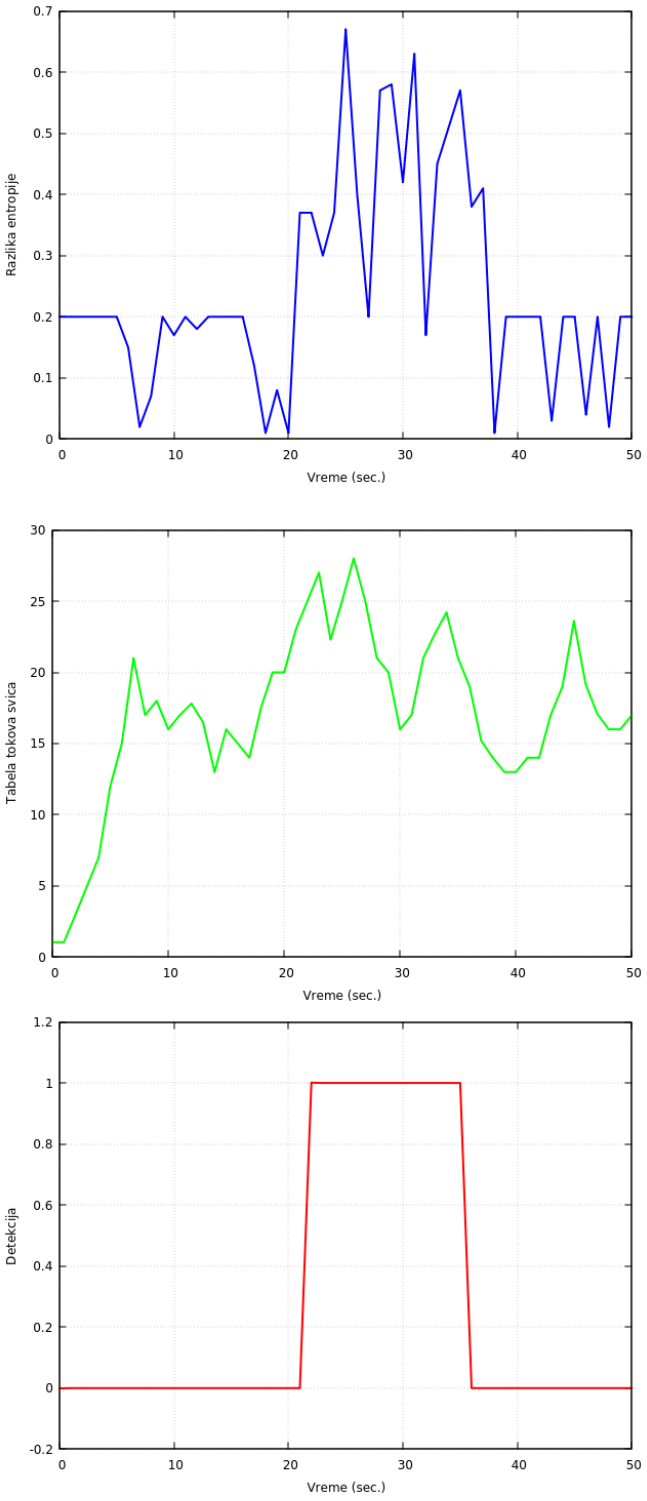



**Slika 8.1:** Varijacije entropije modula za detekciju anomalija

(Analiza TCP-SYN napada preplavljivanjem) Mehanizam TCP-SYN napada je objašnjen u poglavlju 2.4.1. Karakteriše ga da napadač šalje veliki broj zahteva za formiranje TCP veze, koji u svom zaglavlju nose vrednost SYN polja i lažnu izvornu IP adresu, izabranu slučajno. Ovo je tip napada lažnom IP adresom (*spoofed*) i može se izvršiti kao distribuirani direktni napad. Ako su napadi distribuirani, onda se oni obično šalju sa više različitih hostova (*h1* ili *h2* u SDN simulacionoj topologiji). Pošto *h1/h2* šalju veliki broj paketa sa lažnim adresama, ivični svič *Sw8* nije u mogućnosti da ih procesira, tj. da pronađe njihovo poklapanje u svojoj tabeli tokova. Posledica toga je slanje velikog broja *Packet_In* paketa ka kontroleru, što iscrpljuje njegove resurse. Za slučaj ovog napada, razlika entropije odredišnih IP adresa nije prevelika, kao što se vidi na slici 8.2 (a). Varijacija entropije postoji od 21 sekunde, kada je pokrenut napad. Međutim, broj tokova u ivičnom sviču *Sw8* jako je povećan. Algoritam detektuje anomaliju tek nakon ispunjenja oba uslova (prag entropije i broj tokova), tako da već u 22 sekundi kontroler prima poruku o detekciji. Nakon poruke sa sviča, kontroler pristupa ažuriraju tabela tokova svičeva i automatski odbija pakete napada.




**Slika 8.2:** Rezultati detekcije TCP-SYN napada na ivičnom sviču: razlika entropije (gore), broj tokova (sredina) i vreme detekcije (dole)




**Slika 8.3:** Rezultati detekcije ICMP napada na ivičnom sviču:
razlika entropije (gore), broj tokova (sredina) i vreme detekcije (dole)

(Analiza ICMP napada preplavljivanjem) Mehanizam ICMP napada je objašnjen u poglavlju 2.4.1. Tokom ICMP napada, veliki broj paketa sa specifičom odredišnom IP

adresom se generiše u vrlo kratkom vremenskom periodu. Na osnovu slike 8.3 (a) u uslovima legitimnog saobraćaja, razlika entropije je ispod vrednosti 0.2. Nakon pokretanja napada u 20 sekundi, razlika entropije odredišnih IP adresa se brzo povećava. Istovremeno, kao što je prikazano na slici 8.3 (b), broj tokova tabele prosleđivanja sviča se ne menja značajnije u odnosu na uslove regualarnog saobraćaja, ali varijacije su evidentne i utiču na prag detekcije algoritma. Na osnovu algoritma koji se izvršava na ivičnom sviču *Sw8,* vreme detekcije je prikazano na slici 8.3 (c). Od 22 sekunde svič šalje kontroleru poruku o detektovanoj anomaliji. Međutim, vreme detektovanja anomalija se u ovom slučaju produžava, pa će svič detektovati napade do 35 sekunde, kada vrednost entropije drastično opada. Za ICMP fnapad, vreme detekcije anomalija je u odnosu na prethodni slučaj produženo čak za 9 sekundi.

Jedan od doprinosa modula za detekciju anomalija u ovom istraživanju predstavlja efikasniji pristup rešenju problema degradacija performansi kontrolera (*controller overhead*), uzrokovanih prevelikim brojem procesiranja paketa koji se izvršavaju tokom komunikacije sa OpenFlow svičevima. Analiza je izvršena poređenjem zauzetosti POX procesora za predloženi model detekcije na sviču, i centralizovanog rešenja kod kojeg se i detekcija i predikcija napada dešavaju istovremeno na kontroleru. Sa stanovišta hardverskih resursa, centralizovano rešenje je zahtevnije jer kontroler sistemom prozivanja (*pooling*), pored upravljanja i komunikacije sa svičevima, vrši istovremeno detekciju anomalija proračunom entropije i mrežnih tokova, a nakon njihove detekcije izvršava klasifikaciju napada algoritmima mašinskog učenja. U ovom slučaju, ivični svič nema nikakvu dodatnu ulogu osim prosleđivanja paketa. Problem raspodele procesorskih resursa kod SDN mrežnih hipervizora istraživan je npr. u radu [157], pri čemu je predložen prediktivni linearni model skaliranja portova, kojim se postiže minimalna degradacija performansi kontrolera.

Prikaz svih aktivnih Mininet procesa u okviru simulacione virtuelne topologije i njihovih identifikatora (PID) dobijen je komandom:

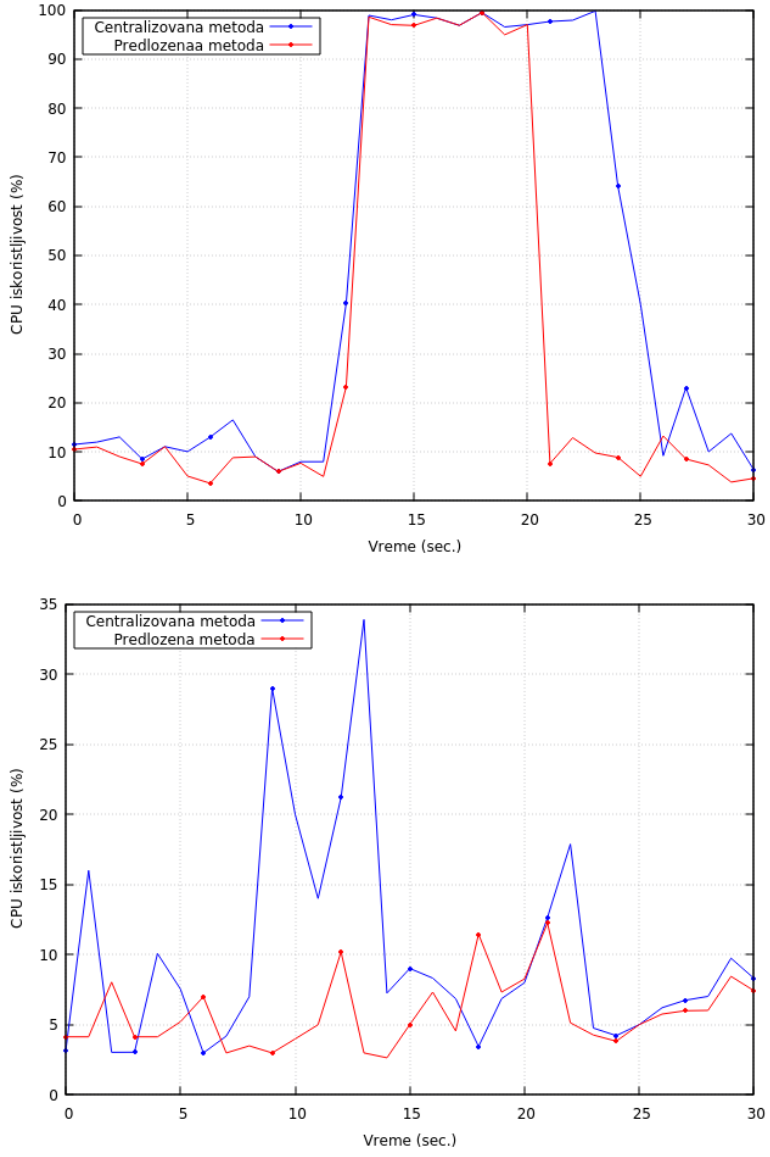
```
$ps aux | less
```

```
root      1465  0.0  0.0  21340  2048 pts/2    Ss+  14:04   0:00 bash --norc -is mininet:c0
root      1469  0.0  0.0  21340  2040 pts/3    Ss+  14:04   0:00 bash --norc -is mininet:h1
root      1473  0.0  0.0  21340  2044 pts/5    Ss+  14:04   0:00 bash --norc -is mininet:h2
root      1478  0.0  0.0  21336  2036 pts/6    Ss+  14:04   0:00 bash --norc -is mininet:s1
```

U odzivu na komandu, prvi red se odnosi na kontroler u topologiji (`mininet:c0`). Druga kolona predstavlja identifikatore procesa. Statistika o zauzetosti procesora kontrolera pre i tokom DDoS napada dobijena je komandom redirekcije na sviču:

```
$top -p 1465 > contr_cpu.txt
```

Slika 8.4 prikazuje zauzetost procesora POX kontrolera za oba tipa izvršenih TCP-SYN i ICMP napada preplavljivanjem . U ovom slučaju, napadi su pokrenuti sa hosta *h1* ka hostu *h32*, u vremenskom intervalu od 15 sekundi, intenzitetom od 1000 paketa/sec, i to prvo za centralizovani model, a zatim za predloženi model detekcije. Upotrebljeni su podrazumevani parametri terminal komandi, tako da su podaci o procesorskoj aktivnosti prikupljani svake sekunde.




**Slika 8.4:** Zauzetost procesora POX kontrolera za TCP-SYN flood napad (gore) i ICMP flood napad (dole)

Za TCP-SYN napad, pored procesiranja velikog broja zahteva i preplavljivanja tabele prosleđivanja, svič komunicira sa kontrolerom kako bi od njega dobio instrukcije o načinu dalje obrade paketa. Tokom ovog napada, zauzetost procesora može da se poveća i do 100%, što je predstavljeno slikom 8.4 (a), U predloženom rešenju, zahvaljujući algoritmu detekcije, kontroler reaguje samo u slučaju prijema signala o nastaloj anomaliji od strane ivičnog sviča. Na taj način se postiže da je zauzetost procesora kontrolera tokom napada značajno niža u odnosu na rešenje sa centralizovanom formom. Može se zaključiti da je trajanje vršne vrednosti zauzetosti procesora za predloženo rešenje detekcije kraće 5 sekundi u odnosu na vrednost dobijenu centralizovanom metodom detekcije.

U drugom slučaju, tokom ICMP napada, u mreži se javlja samo veliki broj lažnih ICMP paketa koji dovode do preopterećenja pristupnog linka sviča. Pošto ivični svič već ima formiranu tabelu prosleđivanja, nema potrebe za dodatnom komunikacijom sa kontrolerom, pa je kao rezultat toga ujednačenost zauzetosti procesora, bez većih varijacija. Kao što se vidi na slici 8.4 (b), procesorska zauzetost za obe metode raste, ali je njena prosečna vrednost za naš predloženi metod detekcije značajno niža od one koja se dobija primenom centralizovane metode detekcije.

## 8.2. **Rezultati i analiza modula za klasifikaciju napada**

U ovom delu disertacije prikazani su i analizirani rezultati modula za klasifikaciju DDoS napada, zasnovanog na mašinskom učenju. Prvo je analiziran model klasifikacije realizovan putem dva javna skupa podataka, a zatim j izvršeno poređenje dobijenih rezultata sa rezultatima klasifikacije za simulacionu Fat-Tree topologiju. Potom su analizirani rezultati koji su dobijeni modifikacijom parametara algoritama nadgledanog mašinskog učenja.

### 8.2.1. Rezultati i analiza za javne skupove podataka

Za detekciju napada neophodno je poznavanje višestrukih karakteristika mrežnog saobraćaja, koje se mogu dobiti prikupljanjem podataka iz mrežnih tokova. Prikupljanje podataka iz realne mrežne topologije najčešće je problematičan postupak, jer zahteva pristup

uređajima putem parametara autorizacije, uz enkriptovane komunikacione linkove. Rešenja za detekciju napada koja koriste podatke iz realnih mreža su malobrojna, pa se najveći broj njih fokusira ka javno dostupnim skupovima podataka. Uprkos istraživanjima koja se sprovode u oblasti detekcije napada, još uvek postoji mnogo izazova kada je reč o razvoju ovih sistema u okviru SDN mrežnih standarda. Činjenica je da postoji jako mali broj dostupnih skupova podataka koji su generisani direktno iz SDN mreže, i koji bi mogli da se iskoriste kao obučavajući skupovi, ili za evaluaciju modela klasifikacije napada. Iako jako mali broj rešenja koristi podatke iz realne SDN mreže i formira prihvatljiv skup podataka, većina tih skupova definiše samo pojedine tipove napada, ne uzimajući u obzir vektore napada koji mogu postojati u različitim ravnima SDN mreža.

U ovom delu rada razmatrani se rezultati klasifikacije i tačnosti detekcije napada primenom mašinskog učenja na javne dostupne skupove podataka. Kao što je napomenuto u poglavlju 7.5.1, za ocenu tačnosti detekcije predloženog rešenja, izabrana su dva skupa podataka, InSDN i CICIDS-2017. U tabeli 8.2 prikazana je raspodela mrežnih tokova za legitiman i saobraćaj napada, a iz oba skupa izdvojene su samo instance tokova koje reprezentuju DDoS napade.

**Tabela 8.2:** Mrežni tokovi InSDN i CICIDS-2017 skupova podataka

| Skup podataka | Legitimni tokovi | Tokovi napada | Ukupno |
|---|---|---|---|
| InSDN (DDoS) | 68424 | 73529 | 141953 |
| CICIDS-2017 (Friday-WorkingHours-DDoS) | 97718 | 128027 | 225745 |

Princip nadgledanog mašinskog učenja koristi obučavajući skup podataka, kako bi se iz prošlih iskustava kroz obuku, predvidelo buduće ponašanje sistema. U slučaju detekcije napada, algoritmom mašinskog učenja sistem se obučava da prepozna i razlikuje saobraćaj DDoS napada od legitimnog. Izabrane javne skupove podataka karakteriše nekoliko osobina. Za oba skupa podataka razmatran je samo legitimni i saobraćaj DDoS napada, dok drugi tipovi instanci tokova nisu uzeti u razmatranje. Stoga je izvršeno filtriranje povezanih instanci iz skupova podataka, kao i predprocesiranje, kako bi se formirao obučavajući skup podataka. U analizi modela mašinskog učenja korišćena je činjenica da intenzitet napada u okviru obučavajućeg skupa ima određeni uticaj na tačnost predikcije. Intenzitet napada definiše se kao količnik broja tokova napada i ukupnog broja svih tokova. U analizi modela korišćene su

tri grupe obučavajućih skupova podataka, sa intenzitetima napada od 5%, 25% i 50%. Činjenica je da nisu svi nezavisni atributi tokova jednako važni pri detekciji anomalija za određenu klasu napada. Određeni atributi jedne klase napada mogu biti relevantni, dok sa aspekta druge klase to ne moraju biti. Pošto oba skupa podataka imaju veliki broj atributa, primenjuje se princip njihove redukcije kako bi se analizirale performanse kada se u obučavajućem skupu koriste samo ključni atributi. Uzima se u obzir vrednost korelacije *R* među atributima, pri čemu su ključni oni koji imaju veću pozitivnu ili negativnu vrednost. Izbor vrednosti korelacije je uglavnom komporomis između broja atributa i tačnosti klasifikacije, jer velika apsolutna vrednost *R* znači manji broj atributa i nižu tačnost. Izbor manjeg broja atributa je poželjan, jer se na taj način štede sistemski resursi potrebni za obradu podataka, i pojednostavljuje se njihova priprema.

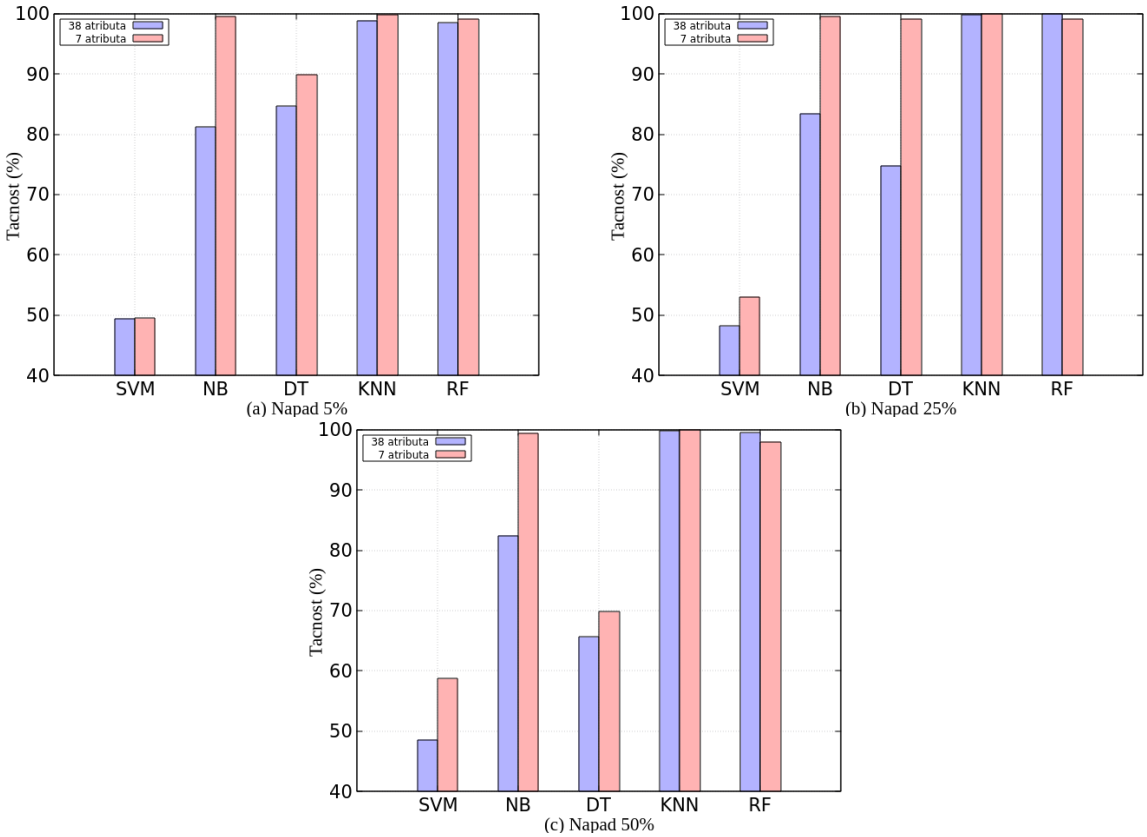
Za InSDN skup podataka značajno je da su pojedini atributi tokova potpuno nevažni za slučaj detekcije DDoS napada. Pretprocesiranje atributa za ovaj skup podataka izvršeno je na osnovu istraživanja opisanog u [158], pri čemu je izvršena selekcija irelevantnih atributa, kako bi se u okviru modela koristili samo osnovni atributi na osnovu kojih bi se izvršila evaluacija njegovih performansi. Npr. atribut *fwd_psh_flags* ima uvek vrednost 0, bez obzira da li se radi o legitimnom saobraćaju ili napadu. U skladu sa tim, usvojena su sledeće ograničenja atributa: a) atributi koji imaju apsolutnu vrednost korelacije manju od 0.1 su irelevantni, pa se razmatra ukupno 38 atributa tokova b) atributi koji imaju vrednost korelacije veću od 0.5, pa se koristi ukupno 7 atributa: *flow_id*, *protocol*, *timestamp*, *init_bwd_win_byts*, *flow_pkts*, *bwd_pkts* i *pkt_len_- min*. Izborom relevantnih atributa, formirane su dve grupe, većeg i manjeg obima. Tačnost klasifikacije za tri različita intenziteta napada, kao i dve grupe atributa prikazana je u tabeli 8.3, a rezultati su u formi dijagrama prikazani na slici 8.5.

Na osnovu tabele 8.3 izvodi se zaključak da se najveća tačnost postiže KNN algoritmom za manji broj atributa tokova (99,98%), kao i za intenzitet napada od 25%. Promena intenziteta napada u obučavajućem skupu nema veliki uticaj na tačnost klasifikacije za većinu algoritama. Najveća razlika dobijena je za DT algoritam i manji broj atributa, što se donekle može objasniti većom prenaučenošću modela na koje je ovaj algoritam naročito osetljiv. Ako se pogledaju podaci u istoj grupi skupa podataka, zaključuje se da razmatranje većeg broja atributa u fazi obučavanja dovodi do dva kompromisna ishoda – veće tačnosti ili prenaučenosti modela. Za najveći broj algoritama nadgledanog mašinskog učenja važi da više atributa u obučavajućem skupu daje malo bolje rezultate tačnosti predikcije. U slučaju InSDN

skupa podataka, tačnost metode detekcije opada sa povećanjem broja atributa koji se uzimaju u razmatranje tokom postupka klasifikacije.

**Tabela 8.3:** Vrednosti za tačnost klasifikacije InSDN skupa podataka. Podebljani su odgovarajući maksimumi

| ML klasifikator | Napad 5% | | Napad 25% | | Napad 50% | | Razlika | |
|---|---|---|---|---|---|---|---|---|
| | Broj atributa | | Broj atributa | | Broj atributa | | Broj atributa | |
| | 38 | 7 | 38 | 7 | 38 | 7 | 38 | 7 |
| SVM | 49,42% | 49,50% | 48,25% | 53,04% | 48,58% | 58,75% | 0,84% | 9,25% |
| NB | 81,19% | 99,56% | 83,45% | 99,52% | 82,44% | 99,35% | 1,25% | 0,21% |
| DT | 84,65% | 89,85% | 74,75% | 99,14% | 65,63% | 69,79% | **19,29%** | **22,06%** |
| KNN | **98,85%** | **99,86%** | 99,83% | **99,98%** | **99,76%** | **99,97%** | 0,91% | 0,11% |
| RF | 98,46% | 99,07% | **99,95%** | 99,06% | 99,45% | 97,93% | 0,99% | 1,14% |




**Slika 8.5:** Metrika tačnosti detekcije DDoS napada za InSDN skup podataka

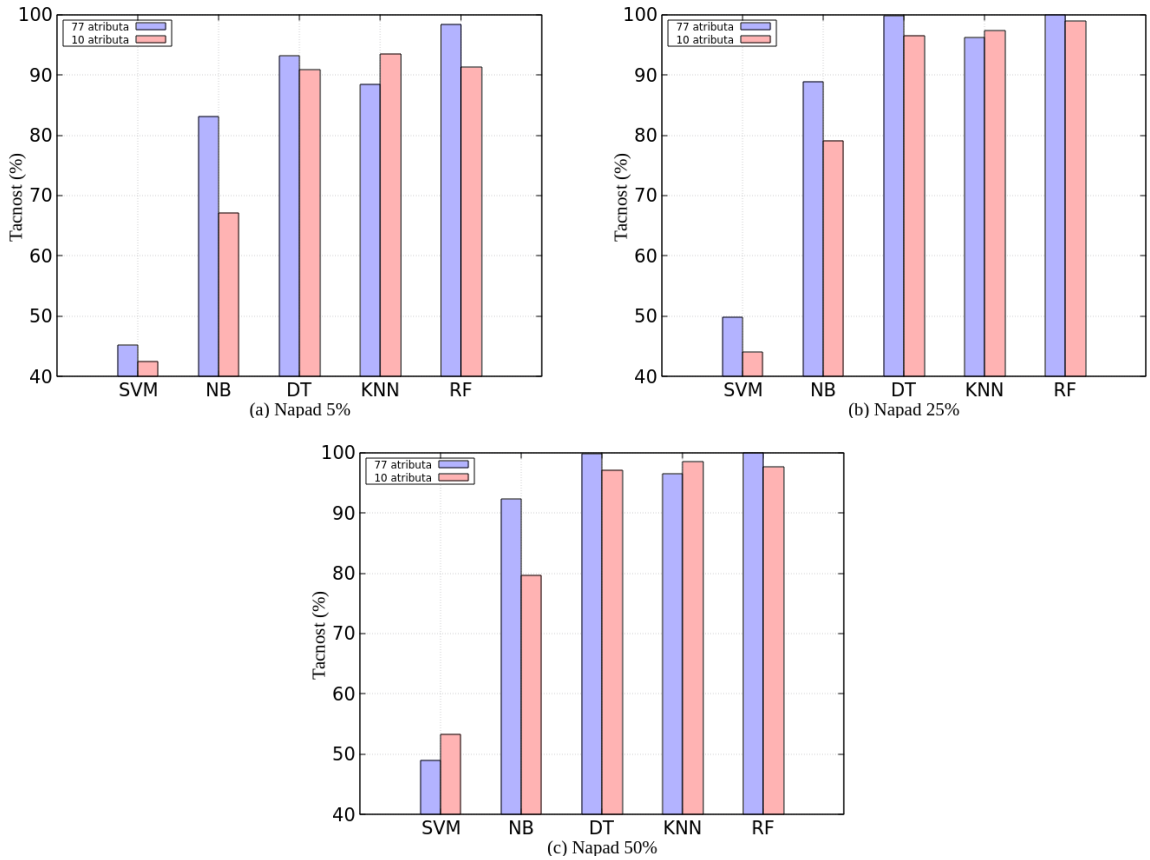
CICIDS-2017 skup podataka je generisan na osnovu mrežnih tokova u okviru tradicionalne računarske mreže, za razliku od InSDN skupa, kod kojeg su podaci formirani virtuelnom SDN emuliranom mrežom. Utvrđeno je da algoritmi mašinskog učenja imaju različito ponašanje i tačnost predikcije za ova dva tipa mreža. CICIDS-2017 sadrži podatke koji simuliraju vrlo kompleksne tipove napada. Podaci su organizovani u 5 grupa (generisani su tokom 5 radnih dana), a za ovaj deo rada analiziran je skup podataka DDoS napada, i može se preuzeti kao *Friday-WorkingHours.pcap* fajl. Opravdanost izbora ovog skupa podataka je u tome što on verno simulira saobraćaj realne mrežne topologije. Za evaluaciju tačnosti modela ne koristimo ceo skup podataka, već samo instance koje se odnose na DDoS napade. Kao i prethodni, i ovaj skup podataka sadrži veći broj atributa tokova, pri čemu određeni atributi nemaju važnost za slučaj klasifikacije mrežnog saobraćaja. CICIDS-2017 sadrži 78 standardnih atributa i jednu oznaku klase. Dva atributa (kolone) u *.csv* fajlu nose naziv *Fwd Header Length*, pa nakon uklanjanja jednog, analizu prvo vršimo sa 77 atributa tokova. Poboljšanje tačnosti modela postignuto je upotrebom reprezentativnijih i diskiminativnijih atributa. Na osnovu podataka iz istraživanja izvršenog u radu [159], za predprocesiranje i redukciju dimenzionalnosti je upotrebljena PCA (*Principle Component Analysis*) analiza, kojom je broj atributa za evaluaciju modela smanjen na 10. Dobijene vrednosti za tačnost klasifikacije za tri intenziteta napada, za standardne i redukovane atribute prikazana je u tabeli 8.4. Dobijene vrednosti tačnosti klasifikacije grafički su prikazane na slici 8.6.

**Tabela 8.4:** Vrednosti za tačnost klasifikacije CICIDS-2017 skupa podataka. Podebljani su odgovarajući maksimumi

| ML klasifikator | Napad 5% | | Napad 25% | | Napad 50% | | Razlika | |
|---|---|---|---|---|---|---|---|---|
| | Broj atributa | | Broj atributa | | Broj atributa | | Broj atributa | |
| | 77 | 10 | 77 | 10 | 77 | 10 | 77 | 10 |
| SVM | 45,22% | 42,53% | 49,85% | 44,04% | 48,98% | 53,35% | 3.76% | 10,82% |
| NB | 83,12% | 67,16% | 88,85% | 79,12% | 92,24% | 79,66% | **9,12%** | **12,5%** |
| DT | 93,15% | 90,85% | 99,73% | 96,54% | 99,73% | 97,09% | 6,58% | 6,24% |
| KNN | 88,45% | **93,47%** | 96,13% | 97,29% | 96,46% | **98,47%** | 8,01% | 5,00% |
| RF | **98,35%** | 91,27% | **99,94%** | **98,03%** | **99,87%** | 97,61% | 1,52% | 6,34% |

Na osnovu dobijenih rezultata evidentno je da KNN i RF algoritmi imaju zadovoljavajuću tačnost za oba skupa podataka, dok je SVM algoritam uglavnom

neprihvatljiv. Najveća vrednost tačnosti dobijena je za RF algoritam, koja za 77 atributa i intenzootet napada od 25% iznosi visokih 99,94%.




**Slika 8.6:** Metrika tačnosti detekcije DDoS napada za CICIDS-2017 skup podataka

Poređenjem rezultata za tačnost predikcije oba javna skupa podataka, moguće je izvesti sledeće zaključke:

- Korišćenje više atributa u obučavajućem skupu obično daje veću tačnost, mada određeni algoritmi pokazuju karakteristike prenaučenosti
- Redukcija i upotreba isključivo jako korelisanih atributa tokova je opcija koja se može detaljnije razmatrati, jer je potrebno pronaći kompromisno rešenje između tačnosti i veličine obučavajućeg skupa. Smanjenje obučavajućeg skupa i broja značajnih atributa se postiže jednostavnijim predprocesiranjem podataka i manjim hardverskim zahtevima za procesiranje

- U slučajevima kada je postignuta tačnost preko 90%, njena vrednost se neznatno razlikuje od vrednosti u obučavajućim skupovima 25%, odnosno 50% intenziteta napada
- Obučavajući skup sa većim intenzitetom napada i većim brojem atributa generalno postiže nešto bolje rezultate tačnosti predikcije
- Dobijene vrednosti za tačnost reprezentuju ukupne performanse algoritama nadgledanog mašinskog učenja, ali one mogu biti neprecizne u uslovima specifičnih mrežnih uslova i okruženja. Zbog toga je potrebno istovremeno koristiti više metrika mašinskog učenja, kako bi se postigla preciznija evaluacija performansi sistema
- Pre implementacije modela nadgledanog mašinskog učenja, potrebno ih je detaljno proučiti, posebno za primenu u uslovima specifičnih SDN mrežnih topologija, jer je funkcionisanje pojedinih algoritama u tradicionalnim mrežama potpuno drugačije u uslovima SDN mrežnih scenarija

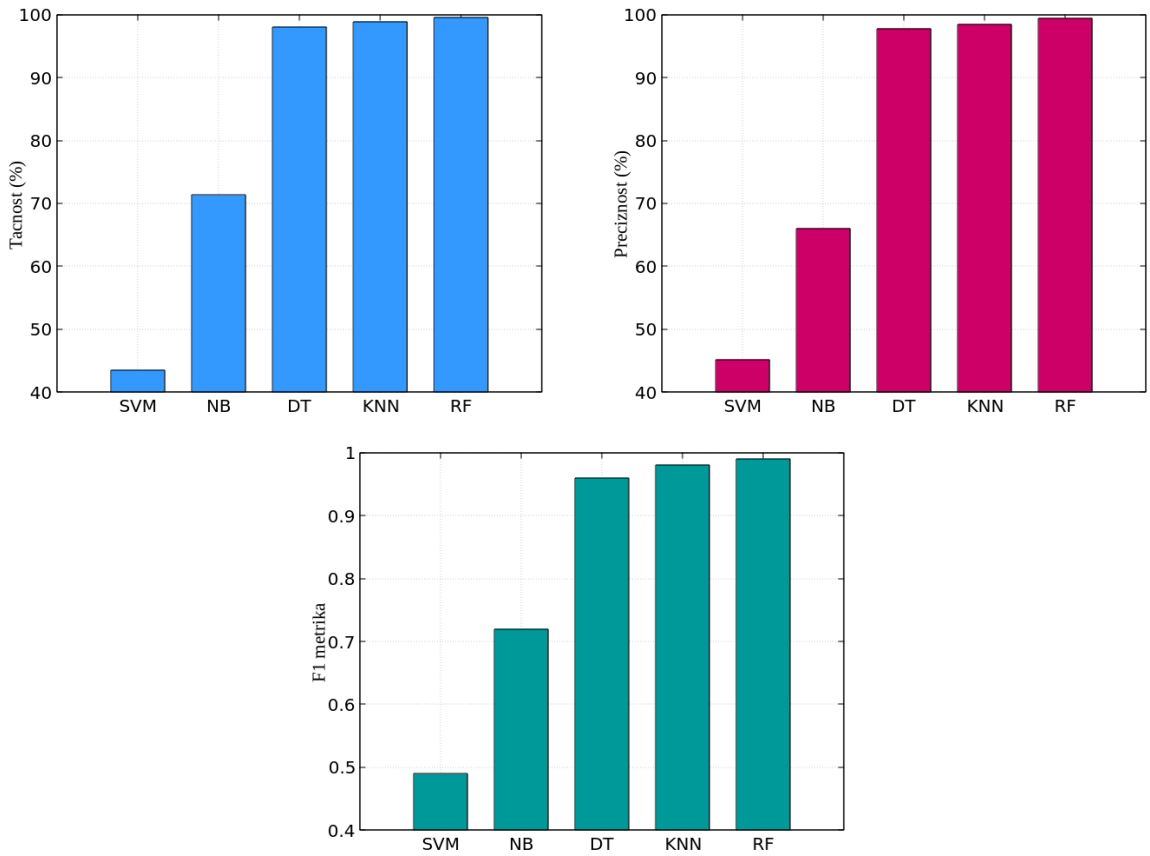
### 8.2.2. Rezultati i analiza za simulacionu topologiju

U ovom delu istraživanja razmatrani se rezultati klasifikacije napada za Mininet emuliranu Fat-Tree topologiju, predstavljenu u poglavlju 7.3. Obučavajući skup dobijen je na osnovu analize *Packet_In* poruka, prikupljenih u trajanju od 300 sec. podeljenih u 30 vremenskih slotova trjanja 10 sec. Za potrebe formiranja obučavajućeg skupa, za razliku od javih skupova podataka, koristi se samo statistika o broju *Packet_In* poruka koje kontroler evidentira od strane mrežnih čvorova. Za scenario napada uzimaju se u obzir i legitimni i saobraćaj mrežnog napada. Osnovu evaluacije tačnosti klasifikacije predstavlja broj uzoraka obučavajućeg skupa, pa se koriste skupovi podataka sa brojem uzoraka od 100 do 25600. Jedan od ciljeva analize je analizirati tačnost algoritama mašinskog učenja koji koriste ove različite obučavajuće skupove podataka. U svakom obučavajućem skupu, regularni i uzorci napada su u odnosu 50:50 kako bi se izbegao problem nedovoljnog podudaranja (*bias*), koji se se javlja kao greška za slučaj kada se koristi jednostavan modela za rešavanja kompleksnog realnog problema. Za grupu od 25600 uzoraka, odnos regularnih i uzoraka napada proveravamo i za niži udeo (5% i 25%) uzoraka napada, jer u realnim mrežama odnos

malicioznog i legitimnog saobraćaja ima malu vrednost. Podela podataka na obučavajući i test skup je u odnosu 80%:20%, odnosno 20480 uzoraka za obuku i 5120 uzoraka za validaciju od ukupno 25600 uzoraka.

**Tabela 8.5:** Vrednosti metrika mašinskog učenja za simulacionu SDN topologiju. Obučavajući skup ima 25600 uzoraka

| ML klasifikator | Tačnost | Preciznost | F1-metrika |
|---|---|---|---|
| SVM | 43,57% | 45,18% | 0.49 |
| NB | 71,34% | 66,03% | 0,72 |
| DT | 98,03% | 97,72% | 0,96 |
| KNN | 98,88% | 98,44% | 0,98 |
| RF | **99,56%** | 99,40% | 0,99 |




**Slika 8.7:** Metrike mašinskog učenja za simulacionu SDN topologiju (tačnost, preciznost i F1 metrika)

Na slici 8.7 prikazane su metrike nadgledanog mašinskog učenja za simulacionu SDN topologiju. Prikazana je tačnost, preciznost i F1-metrika za obučavajući skup od 25600 uzoraka. Metrika tačnosti pokazuje udeo tačno klasifikovanih uzoraka od skupa svih uzoraka koji se klasifikuju. Tačnost kao metrika nije dovoljno informativna za procenu klasifikatora, jer ne ukazuje na tačnost klasifikacije svakog tipa napada, tako da se uz tačnost razmatra i metrika preciznosti. Preciznost određuje udeo tačno klasifikovanih pozitivnih alarma u odnosu na sve alarme detekcije koji su klasifikovani kao pozitivni. Pošto metrike tačnosti i preciznosti ne uzimaju u obzir i tačne negativne alarme, one nisu uvek dovoljne za poređenje klasifikatora i izbor optimalnog za problem detekcije napada. Iz tog razloga prikazana je F1-metrika, koja uzima u razmatranje i tačne pozitivne i tačne negativne alarme, i na taj način predstavlja otežinjenu vrednost preciznosti i odziva i omogućava pouzdanije poređenje. Na osnovu analize klasifikacije napada u prethodnom slučaju, zaključak je da se varijacije tačnosti smanjuju sa povećanjem obučavajućeg skupa, što znači da predikcija postaje tačnija sa dovoljno velikim obučavajućim skupom. Pojedini algoritmi, kao npr. KNN i RF, imaju trend dostizanja blizu 100% tačnosti, u slučaju kada se koristi veliki obučavajući skup, dok se drugi, kao npr. SVM model ne menjaju značajnije. Nisu sve klasifikacione tehnike prilagođene predloženom obučavajućem skupu u ovom modelu detekcije napada, pa je neophodna detaljnija analiza. Može se zaključiti da su DT, KNN i RF tri algoritma koja su najbolje prilagođena predloženom obučavajućem skupu. RF postiže tačnost od 99,56%, što je najveća vrednost među svih pet tehnika nadgledanog učenja koje razmatramo.

Prethodna analiza simulacione topologije pokazuje da se pojedinim algoritmima mašinskog učenja (kao npr. KNN ili RF) može postići velika tačnost samo sa jednim mrežnim atributom u obučavajućem skupu, što znači da se za klasifikaciju mogu primeniti jednostavnije funkcije za razlikovanje malicioznog i legitimnog saobraćaja. Izbor manjeg broja atributa u obučavajućem skupu ima određene prednosti:

- formiranje obučavajućih skupova postaje jednostavnije jer se ne moraju detaljno ispitivati mrežni tokovi
- faza obučavanja postaje jednostavnija, jer nema potrebe za definisanjem kritičnih atributa
- faza obučavanja zahteva manje hardverske resurse usled jednostavnijeg obučavajućeg skupa

Promena parametara u algoritmima nadgledanog učenja je vrlo često primenjivana metoda za poboljšanje performansi sistema za detekciju napada. Različiti modeli primenjeni na istu mrežnu topologiju daju potpuno različite rezultate tačnosti. Uzroci leže uglavnom u karakteristikama obučavajućeg skupa, kao i u algoritmima koji se koriste za klasifikaciju. Kako bi se ispitao uticaj algoritama na ishod klasifikacije, primenjene su izmene parametara određenih algoritama. U tabeli 8.6 prikazani su rezultati tačnosti za KNN, DT i RF algoritme, za osnvni model algoritma i za model sa izmenjenim parametrima. Takođe, za svaki klasifikator upotrebljeni su obučavajući skupovi sa tri najveće vrednosti uzoraka. Među tri analizirana algoritma, osnovni DT algoritam ima najveću tačnost od 98,89%. U navedenim algoritmima promenjeni su sledeći parametri:

- Za KNN algoritam promenjen je parametar moda udaljenosti, pa je upotrebljen standardizovani "Euclidan to city block" model, čime je dobijeno malo poboljšanje (0,60% za skup sa 6400 uzoraka)
- Za DT, izvršena je izmena "MinParentSize" parametra, čije su vrednosti menjane u opssegu od 2 do 10. Najbolji rezultati su dobijeni za vrednost 5 ovog parametra (0,96% za obučavajući skup sa 6400 uzoraka)
- Za SVM algoritam koristili smo RBF (*Radial Basis Function*) kernel, namenjen rešavanju problema male dimenzionalnosti i za nestruktuirane podatke. Evidentno je značaajno pobolljšanje tačnosti, koje za obučavajući skup sa 25600 uzoraka iznosi 35,53%

**Tabela 8.6:** Tačnost klasifikacije za izmenjene parametre klasifikatora

| ML klasifikator | Trening uzorci | Osnovni model | Poboljšani model | Poboljšanje |
|---|---|---|---|---|
| KNN | 6400 | 98,87% | 99,47% | 0,60% |
| | 12800 | 98,24% | 98,78% | 0,54% |
| | 25600 | 98,88% | 99,03% | 0,15% |
| DT | 6400 | 97,86% | 98,81% | 0,96% |
| | 12800 | 98,89% | 99,57% | 0,92% |
| | 25600 | 98,03% | 99,76% | 1,73% |
| SVM | 6400 | 48,85% | 61,79% | 20,94% |
| | 12800 | 47,44% | 66,43% | 28,50% |
| | 25600 | 43,57% | 67,59% | 35,53% |

Rezultati dobijeni klasifikacionim algoritmima mašinskog učenja pokazuju da je predložena metoda detekcije napada primenljiva na SDN mrežne topologije i da postiže zadovoljavajuće rezultate tačnosti klasifikacije. Pokazano je da je DDoS napade moguće detektovati jednim atributom u obučavajućem skupu, umesto upotrebe višestrukih atributa. Podaci obučavajućeg skupa su prikupljeni iz Mininet emulacione SDN mreže, a virtuelizacijom mreže omogućena je detaljna analiza podataka, kao i implementacija naprednih algoritama predikcije. U analizi klasifikacije upotrebljeni su obučavajući skupovi sa različitim brojem uzoraka, i upoređene su performanse pet algoritama mašinskog učenja. Rezultati pokazuju da performanse primenjenih tehnika nadgledanog mašinskog učenja značajno variraju pri istim scenarijima testiranja, dok se modifikacijom parametara klasifikacionih algoritama ovi rezultati mogu poboljšati u određenoj meri. Dobijeni rezultati pokazuju da od svih modela, najbolje rezultate tačnosti detekcije postiže RF algoritam, jer postiže vrednost od 99,56%.

Za nadgledano mašinsko učenje potreban je veliki broj obeleženih podataka za obučavajući skup, kako bi mogle da se detektuju različite vrste DDoS napada. Kao što je napomenuto u poglavlju 5. tj. u pregledu postojećih istraživanja u ovoj oblasti, performanse modela mašinskog učenja koji koriste iste skupove podataka značajno se razlikuju. Preporuka je da se pre razvoja modela procene mogućnosti predikcije različitih tehnika mašinskog učenja, kako bi se pronašao najodgovarajući model za konkretan mrežni scenario. Glavni nedostatak predloženog modela klasifikacije je njegova primena isključivo za DDoS napade preplavljivanjem. Specifičnost neoznačenih podataka i mali broj atributa dovode do toga da se ovom metodom ne mogu detektovati nevolumetrijski napadi, što naročito važi za DDoS napade male brzine izvršavanja.

# 9. ZAKLJUČAK

U ovoj disertaciji je izvršena analiza problematike bezbednosti specifičnih SDN mrežnih okruženja. Osnovu istraživačkog rada predstavljala je realizacija nove metode detekcije posebne klase napada realizovanih odbijanjem mrežnih servisa (DDoS). Dosadašnja istraživanja u ovoj oblasti uglavnom su se fokusirala na tradicionalne mreže i njihove servise. Zahvaljujući programabilnosti i automatizovanom upravljanju u okviru SDN mreža, oblast razvoja postupaka detekcije i sprečavanja DDoS napada je dodatno aktuelizovana, tako da se pojavljaju nova rešenja koja koriste sasvim drugačije metode detekcije napada. U dostupnoj literaturi nije dostupan veliki broj radova u kojima su rešenja za detekciju DDoS napada primenjena isključivo na SDN mreže. Evidentno je da su potrebna intenzivnija istraživanja u ovoj oblasti, s obzirom na dinamičnost mehanizama DDoS napada i veliku raznolikost u načinima njihovog izvršavanja.

Predložena metoda daje drugačiji uvid u problematiku detekcije DDoS napada i upućuje na njenu originalnost kroz više aspekata. U okviru nekoliko poglavlja disertacije, metodološki su prezentovani teorijski principi predložene metode detekcije DDoS napada. Kroz različite faze razvoja rešenja, primenjeni su različiti pristupi, potpomognuti softverskom simulacijom, specifičnom za SDN topologiju.

U početnom poglavlju rada opisane su i klasifikovane anomalije i napadi u SDN mrežama, pri čemu je dat iscrpniji opis mehanizama DDoS napada koji se izvršavaju prvenstveno na kontrolerima. Objašnjen je uticaj ovih napada na osnovne SDN procesne elemente, uz detaljniji prikaz procesa komunikacije između OpenFlow svičeva i kontrolera.

Osnovne karakteristike entropije, kao i njene prednosti i mane u kontekstu primene za detekcije mrežnih anomalija i napada, prikazane su u poglavlju koje se bavi teorijskim osnovama predloženog rešenja detekcije.

Opis simulacionog okruženja i namenski kreirane SDN topologije, predstavljeni su u zasebnom poglavlju disertacije. Jedno poglavlje istraživanja opisuje karakteristike skupova podataka upotrebljenih u delu nadgledanog mašinskog učenja. Najvažniji deo ovog

istraživanja odnosi se na opis i realizaciju nove metode za detekciju DDoS napada i anomalija, što je detaljno prikazano u poglavlju 7.

U prvom delu rešenja razvijen je i implementiran algoritam za detekciju mrežnih anomalija, koji proračunom entropije i mrežnih tokova na ivičnom sviču izvršava znatno bržu detekciju napada. Predloženo rešenje je jedno od retkih koje koristi hardverske resurse ivičnog sviča SDN mreže, kako bi se procesorski zahtevan proračun entropije izvršio u okviru hardvera sviča, a kontroler oslobodio dodatnog procesiranja mrežnih podataka. Dobijeni rezultati za dva tipa napada preplavljivanjem pokazuju brzu detekciju napada i mali broj detektovanih lažnih alarma.

Rezultati dobijeni klasifikacionim algoritmima mašinskog učenja pokazuju da je predložena metoda detekcije napada primenljiva na SDN mrežne topologije i da postiže dobre rezultate tačnosti klasifikacije. Dobijeni rezultati pokazuju da se najveća tačnost detekcije od 99,56% postiže RF algoritmom, što ukazuje na visok procenat detektovanih DDoS napada. Osnovna prednost ovog dela predložene metode je dokaz da je DDoS napade moguće detektovati malim brojem atributa obučavajućg skupa, umesto upotrebe višestrukih atributa koji zahtevaju kompleksnije procesure i veće hardverske resurse za izvršavanje.

U ovom istraživanju je potvrđeno da se predložena metoda detekcije DDoS napada može uspešno primeniti u SDN topologijama sa vrlo velikim brojem procesnih mrežnih uređaja i kompleksnim strukturama. Tipičan primer ovakvih okruženja su cloud infrastrukture provajdera ili velikih centara podataka, kod kojih je SDN mrežni koncept sve dominantniji i kod kojih je u poslednje vreme evidentan rast njihove mrežne infrastrukture.

# 10. PRAVCI DALJEG RAZVOJA

Postoji nekoliko pravaca daljeg istraživanja i razvoja:

- Preciznija podešavanja predložene metode detekcije DDoS napada, kako bi se dodatno smanjio broj lažnih alarma pri velikim protocima regularnog mrežnog saobraćaja, što je karakteristično za SDN mreže. Predloženi metod detekcije mogao bi biti primenljiv za slučaj dodatnog kombinovanja različitih mrežnih atributa
- Istraživanje mogućnosti primene mehanizama dubinskog učenja na predloženi model detekcije DDoS napada
- Hardverska realizacija predloženog detektora DDoS napada, koji bi se potencijalno mogao ugraditi u postojeću komercijalnu mrežnu opremu sa većim ili manjim stepenom kompatibilnosti
- S obzirom da je u radu prikazano da se klasifikacija napada može primeniti na vrlo kompleksne SDN mrežne topologije, zanimljiv pravac istraživanja moglo bi biti kombinovanje predložene metode minimalnih atributa modela mašinskog učenja sa vrednostima nekih drugih parametara mrežnog saobraćaja ili nekih drugih polja iz zaglavlja TCP/IP paketa
- Analiza mogućnosti implementacije predloženog modela u SDN topologije sa distribuiranom strukturom kontrolera. U okruženjima sa više kontrolera, istraživanja bi mogla biti fokusirana na segment međukontrolerske komunikacije i mehanizma raspoređivanja istanci DDoS napada na pojedinačne kontrolerske module, čime bi se omogućilo formiranje detaljnog statusa entropije, a što bi dalje zahtevalo implementaciju mehanizama redundanse i balansiranja mrežnog saobraćaja između kontrolera

# LITERATURA


[1] Cisco Annual Internet Report (2018-2023), Cisco piublic, White paper, https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf

[2] D. F. Macedo, A. L. dos Santos and G. Pujolle, "From TCP/IP to convergent networks: challenges and taxonomy," in IEEE Communications Surveys & Tutorials, vol. 10, no. 4, pp. 40-55, 2008, doi: 10.1109/SURV.2008.080405.

[3] P. Goransson, C. Black, T. Culver, "Software Defined Networks - A Comprehensive Approach," Second Edition, Morgan Kaufmann, 2016, ISBN: 978-0-12-804555-8

[4] M. Mousa, A. M. Bahaa-Eldin and M. Sobh, "Software Defined Networking concepts and challenges," 2016 11th International Conference on Computer Engineering & Systems (ICCES), 2016, pp. 79-90, doi: 10.1109/ICCES.2016.7821979.

[5] ONF Foundation, "SDN Security Considerations in the Data Center," ONF Solution Brief, 2013, https://opennetworking.org/wp-content/uploads/2013/05/sb-security-data-center.pdf

[6] Cloudflare DDoS threat report for 2022 Q4, https://blog.cloudflare.com/ddos-threat-report-2022-q4/

[7] R. Koch and M. Golling, "Architecture for evaluating and correlating NIDS in real - World networks," 5th International Conference on Cyber Conflict (CYCON 2013), Tallinn, Estonia, pp. 1-20, 2013

[8] S. W. Cadzow, "Security mechanisms in converged networks," 2005 The First IEE International Conference on Commercializing Technology and Innovation (Ref. No. 2005/11044), 2005, pp. 0_96-D5/6, doi: 10.1049/ic:20050607

[9] M. H. Bhuyan, D. K. Bhattacharyya and J. K. Kalita, "Network Anomaly Detection: Methods, Systems and Tools," in IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 303-336, 2014, doi: 10.1109/SURV.2013.052213.00046.

[10] G. Sebestyen, A. Hangan, Z. Czako and G. Kovacs, "A taxonomy and platform for anomaly detection," 2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), 2018, pp. 1-6, doi: 10.1109/AQTR.2018.8402710.

[11] A. Qayyum, M. H. Islam and M. Jamil, "Taxonomy of statistical based anomaly detection techniques for intrusion detection," Proceedings of the IEEE Symposium on Emerging Technologies, pp. 270-276, 2005, doi: 10.1109/ICET.2005.1558893.

[12] M. Al-Asli and T. A. Ghaleb, "Review of Signature-based Techniques in Antivirus Products," 2019 International Conference on Computer and Information Sciences (ICCIS), 2019, pp. 1-6, doi: 10.1109/ICCISci.2019.8716381.

[13] I. P. Saputra, E. Utami and A. H. Muhammad, "Comparison of Anomaly Based and Signature Based Methods in Detection of Scanning Vulnerability," 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), pp. 221-225, 2022., doi: 10.23919/EECSI56542.2022.9946485.

[14] D. Pleskonjić, N. Maček, B. Đorđević, M. Carić, "Sigurnost računarskih mreža", Beograd, Viša elektrotehnička škola, 2006, ISBN 86-85081-16-5

[15] S. Guo, Y. Liu and Y. Su, "Comparison of Classification-based Methods for Network Traffic Anomaly Detection," 2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2021, pp. 360-364, doi: 10.1109/IMCEC51613.2021.9482274.

[16] H. M. Deylami, R. C. Muniyandi, I. T. Ardekani, A. Sarrafzadeh, "Taxonomy of malware detection techniques: A systematic literature review," 14th Annual Conference on Privacy, Security and Trust, 2016, pp. 629-636, doi: 10.1109/PST.2016.7906998.

[17] S. Latha and S. J. Prakash, "A survey on network attacks and Intrusion detection systems," 4th International Conference on Advanced Computing and Communication Systems (ICACCS), 2017, pp. 1-7, doi: 10.1109/ICACCS.2017.8014614.

[18] H. Bidgoli, "Handbook of Information Security: Threats, Vulnerabilities, Prevention, Detection, and Management," vol. 3, John Wiley & Sons Inc., New Jersey, USA, 2006.

[19] K. Scarfone, P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS), Recomm. of the National Institute of Standards and Technology", NIST Special Publication 800-94, https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=50951

[20] O. Joldžić, Z. Đurić, "Prijedlog rješenja za detekciju i klasifikaciju sigurnosnih propusta web aplikacija," Info M - Časopis za informacionu teh. i multimedijalne sisteme, vol. 9, no. 34, pp. 42–52, 2010. https://infom.fon.bg.ac.rs/index.php/infom/article/view/759

[21] B. Maros, I. Homoliak, M. Kacic and H. Petr, "Detection of network buffer overflow attacks: A case study," 47th International Carnahan Conference on Security Technology (ICCST), 2013, pp. 1-4, doi: 10.1109/CCST.2013.6922067.

[22] O. Al-Jarrah and A. Arafat, "Network Intrusion Detection System using attack behavior classification," 5th International Conference on Information and Communication Systems (ICICS), pp. 1-6, 2014, doi: 10.1109/IACS.2014.6841978.

[23] A. Alshamrani, "Reconnaissance Attack in SDN based Environments," 27th International Conference on Telecommunications (ICT), 2020, pp. 1-5, doi: 10.1109/ICT49546.2020.9239510.

[24] W. H. Allen, G. A. Marin and L. A. Rivera, "Automated detection of malicious reconnaissance to enhance network security," Proceedings IEEE SoutheastCon., pp. 450-454, 2005, doi: 10.1109/SECON.2005.1423286.

[25] P. Y. Leonov et al. "The Main Social Engineering Techniques Aimed at Hacking Information Systems," Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), pp. 0471-0473, 2021, doi: 10.1109/USBEREIT51232.2021.9455031.

[26] X. Wang, K. Zheng, X. Niu, B. Wu, C. Wu, "Detection of command and control in advanced persistent threat based on independent access," IEEE International Conference on Communications (ICC), pp. 1-6, 2016, doi: 10.1109/ICC.2016.7511197.

[27] Christos Douligeris, Aikaterini Mitrokotsa, "DDoS Attacks and Defense Mechanisms: Classification and State-of-the-Art," Computer Networks, The International Journal of Computer and Telecommunications Networking, vol. 44, Issue 55, pp. 643–666, 2004, https://doi.org/10.1016/j.comnet.2003.10.003

[28] Jelena Mirkovic, Peter Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," ACM SIGCOMM Computer Communication Review, vol. 34, Issue 2, 2004, pp. 39-53, https://doi.org/10.1145/997150.997156

[29] S. M. Specht, R. B. Lee, "Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures," Proceedings of the ISCA, 17th International Conference on Parallel and Distributed Computing Systems, USA, 2004.

[30] T. Mahjabin, Y. Xiao, G. Sun, J. Wangdong, "A survey of distributed denial-of-service attack, prevention, and mitigation techniques," International Journal of Distributed Sensor Networks, vol. 13, pp. 1-32, 2017, doi: 10.1177/1550147717741463

[31] M. De Donno, A. Giaretta et al., "A taxonomy of distributed denial of service attacks," International Conference on Information Society (i-Society), pp. 100-107, 2017, doi:10.23919/i-Society.2017.8354681.

[32] D. K. Bhattacharyya, J. K. Kalita, "DDoS Attacks: Evolution, Detection, Prevention, Reaction, and Tolerance," CRC Press, 2016, https://doi.org/10.1201/b20614

[33] R.Tandon, "A survey of distributed denial of service attacks and defenses." arXiv preprint, (2020). https://arxiv.org/abs/2008.01345

[34] E. Alomari et al. "Botnet-based Distributed Denial of Service (DDoS) Attacks on Web Servers: Classification and Art," International Journal of Computer Applications, vol. 49, no.7, pp. 24-32, 2012, doi: 10.5120/7640-0724

[35] Y. K. Shaheen, M. al Kasassbeh, "A Proactive Design to Detect Denial of Service Attacks Using SNMP-MIB ICMP Variables," 2nd Int. Conference on new Trends in Computing Sciences (ICTCS), pp. 1-6, 2019, doi: 10.1109/ICTCS.2019.8923045.

[36] F. Yihunie et al. "Analysis of ping of death DoS and DDoS attacks," 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT), pp. 1-4, 2018, doi: 10.1109/LISAT.2018.8378010.

[37] J. Wang et al. "Detecting and Mitigating Target Link-Flooding Attacks Using SDN," in IEEE Transactions on Dependable and Secure Computing, vol. 16, no. 6, pp. 944-956, 2019, doi: 10.1109/TDSC.2018.2822275.

[38] A. Sahi, D. Lai, Y. Li, M. Diykh, "An Efficient DDoS TCP Flood Attack Detection and Prevention System in a Cloud Environment," in IEEE Access, vol. 5, pp. 6036-6048, 2017, doi: 10.1109/ACCESS.2017.2688460.

[39] M. Ozkan-Okay, R. Samet, Ö. Aslan, D. Gupta, "A Comprehensive Systematic Literature Review on Intrusion Detection Systems," in IEEE Access, vol. 9, pp. 157727-157760, 2021, doi: 10.1109/ACCESS.2021.3129336.

[40] V. D. M. Rios, P. R. M. Inacio et al. "Detection and Mitigation of Low-Rate Denial-of-Service Attacks: A Survey," in IEEE Access, vol. 10, pp. 76648-76668, 2022, doi: 10.1109/ACCESS.2022.3191430.

[41] V. Ganti, O. Yoachimik: DDoS Attack Trends for Q3 2021, The Cloudflare Blog, https://blog.cloudflare.com/ddos-attack-trends-for-2021-q3/

[42] J. L. Cummings, K. R. Hickey, B. D. Kinney, "AT&T Network Architecture Evolution," 1987 AT&T Technical Journal, vol. 66, Issue 3, pp. 2-12, 1987. https://doi.org/10.1002/j.1538-7305.1987.tb00205.x

[43] L. Yang, R. Dantu, T. A. Anderson, R. Gopal, "Forwarding and Control Element Separation (ForCES) Framework, " RFC 3746, pp. 1-40, 2004, doi: 10.17487/RFC3746.

[44] B. A. A. Nunes et al. "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," in IEEE Communications Surveys & Tutorials, vol. 16, no. 3, pp. 1617-1634, 2014, doi: 10.1109/SURV.2014.012214.00180.

[45] ONF: Open Networking Foundation, Software-Defined Networking (SDN) Definition, https://www.opennetworking.org/sdn- definition/

[46] P. L. Ventre et al., "Deploying SDN in GÉANT production network," 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 1-2, 2017, doi: 10.1109/NFV-SDN.2017.8169862.

[47] D. Kreutz, F. M. V. Ramos et al., "Software-Defined Networking: A Comprehensive Survey," in Proceedings of the IEEE, vol. 103, no. 1, pp. 14-76, 2015, doi: 10.1109/JPROC.2014.2371999.

[48] B. Raghavan et al., "Software-defined internet architecture: decoupling architecture from infrastructure," 11th ACM Workshop on Hot Topics in Networks, HotNets-XI, ACM, pp. 43-48, 2012, doi: 10.1145/2390231.2390239.

[49] P. Goransson, C. Black, T. Culver, "Software Defined Networks A Comprehensive Approach," 2nd Edition, Morgan Kaufmann Publishers, 2016.

[50] R. V. Nunes, R. L. Pontes and D. Guedes, "Virtualized network isolation using Software Defined Networks," 38th Annual IEEE Conference on Local Computer Networks, pp. 683-686, 2013, doi: 10.1109/LCN.2013.6761310.

[51] A. D. Ferguson, et al., "Orion: Google's Software-Defined Networking Control Plane," in Proceedings of the 18th USENIX Symposium on Networked Systems Design and Implementation, 2021, https://www.usenix.org/conference/nsdi21/presentation/ferguson

[52] E. Kaljic, A. Maric, P. Njemcevic, M. Hadzialic, "A Survey on Data Plane Flexibility and Programmability in Software-Defined Networking," in IEEE Access, vol. 7, pp. 47804-47840, 2019, doi: 10.1109/ACCESS.2019.2910140.

[53] K. Tantayakul, R. Dhapou, B. Paillassa et al., "Experimental analysis in SDN open source environment," 14th International Conference (ECTI-CON), pp. 334-337, 2017, doi: 10.1109/ECTICon.2017.8096241.

[54] Github NOX Repo, https://github.com/noxrepo/

[55] Liehuang Zhu et al., "SDN Controllers: Benchmarking & Performance Evaluation," arXiv:1902.04491 [cs.NI], 2019, doi =10.48550/ARXIV.1902.04491

[56] Y. Zhao ez al., "On the performance of SDN controllers: A reality check," IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN), pp. 79-85, 2015, doi: 10.1109/NFV-SDN.2015.7387410.

[57] M. G. Alabarce, A. Bravalheri, P. P. Mariño, "INSPIRING-SNI: Investigating SDN Programmability Improving Optical South-and-North-Bound Interfaces," 22nd Int. Conference (ICTON), 2020, pp. 1-4, doi: 10.1109/ICTON51198.2020.9203409.

[58] OVSDB: The Open vSwitch Database Management Protocol, RFC 7047, 2013, https://tools.ietf.org/html/rfc7047

[59] ForCES: Forwarding and Control Element Separation (ForCES) Protocol Extensions, RFC 7391, 2014, https://tools.ietf.org/html/rfc7391

[60] OpFlex Control Protocol, https://datatracker.ietf.org/doc/html/draft-smith-opflex-00

[61] NETCONF: Network Configuration Protocol (NETCONF), RFC 6241, 2011, https://tools.ietf.org/html/rfc6241

[62] C. Banse, S. Rangarajan, "A Secure Northbound Interface for SDN Applications," IEEE Trustcom/BigDataSE/ISPA, pp. 834-839, 2015, doi: 10.1109/Trustcom.2015.454.

[63] OpenFlow Switch Specification, Version 1.5.1 (Protocol version 0x06), ONF TS-025, https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf

[64] A. Kalyaev, E. Melnik, "FPGA-based approach for organization of SDN switch," 9th International Conference on Application of Information and Communication Technologies (AICT), pp. 363-366, 2015, doi: 10.1109/ICAICT.2015.7338580.

[65] IANA (Internet Assigned Numbers Authority), OpenFlow assignment for 6653 TCP/UDP port, RFC 6335, 2013, https://www.rfc-editor.org/rfc/rfc6335.html

[66] D. Kreutz, F. M. V. Ramos, P. Verissimo, "Towards Secure and Dependable Software-Defined Networks," HotSDN '13: Proceedings of the second ACM SIGCOMM workshop, pp. 55-60, 2013, https://doi.org/10.1145/2491185.2491199

[67] Open Networking Foundation: Principles and Practices for Securing Software-Defined Networks, ONF-TR511, 2015, www.opennetworking.org

[68] Haopei Wang et al., "Towards Fine-grained Network Security Forensics and Diagnosis in the SDN Era," CCS '18: Proceedings of the 2018 ACM SIGSAC Conference, pp. 3-16, 2018, https://doi.org/10.1145/3243734.3243749

[69] S. Scott-Hayward, S. Natarajan, S. Sezer, "A Survey of Security in Software Defined Networks," in IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 623-654, 2016, doi: 10.1109/COMST.2015.2453114.

[70] M. B. Jiménez, D. Fernandez et al, "A Survey of the Main Security Issues and Solutions for the SDN Architecture," in IEEE Access, vol. 9, pp. 122016-122038, 2021, doi: 10.1109/ACCESS.2021.3109564.

[71] I. Ahmad, S. Namal, M. Ylianttila, A. Gurtov, "Security in Software Defined Networks: A Survey," in IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2317-2346, 2015, doi: 10.1109/COMST.2015.2474118.

[72] M. Hajizadeh, T. V. Phan, T. Bauschert, "Probability Analysis of Successful Cyber Attacks in SDN-based Networks," 2018 IEEE Conference on NFV-SDN, 2018, pp. 1-6, doi: 10.1109/NFV-SDN.2018.8725664.

[73] ONF: OFCONFIG 1.2 - OpenFlow Management and Configuration Protocol, ONF TS-016, 2014, https://opennetworking.org/wp-content/uploads/2017/07/of-config-1.2.pdf

[74] S. Midha, K. Triptahi, "Extended TLS security and Defensive Algorithm in OpenFlow SDN," 9th International Conference on Cloud Computing, Data Science & Engineering, pp. 141-146, 2019, doi: 10.1109/CONFLUENCE.2019.8776607.

[75] E. de la Hoz et al., "Detecting and defeating advanced man-in-the-middle attacks against TLS," 6th International Conference On Cyber Conflict (CyCon 2014), pp. 209-221, 2014, doi: 10.1109/CYCON.2014.6916404.

[76] Phillip Porras et al., "Securing the Software-Defined Network Control Layer," Proceedings of the NDSS Symposium, USA, 2015

[77] J. Park, W. Yoon, "SDN-based heterogeneous network architecture with Multi-Controllers," 22nd International Conference on Advanced Communication Technology (ICACT), pp. 559-561, 2020, doi: 10.23919/ICACT48636.2020.9061391.

[78] Q. Ma, L. Dong, X. Jiang, G. Zhu, "Research on anomaly detection method for SDN multi-controller," International Conference on Information Science, Parallel and Distributed Systems, pp. 136-139, 2021, doi: 10.1109/ISPDS54097.2021.00034.

[79] R. Durairajan, J. Sommers, P. Barford, "Controller-Agnostic SDN debugging," in CoNEXT Proceedings of the 2014 Conference on Emerging Networking Experiments and Technologies, pp. 227-233, 2014, https://doi.org/10.1145/2674005.2674993

[80] H3C S5820V2 Switch Series Support, https://www.h3c.com/en/Support/

[81] HP Switch Software: OpenFlow v1.3 Administrator Guide K/KA/WB15.17, Edition 2, 2015, https://support.hpe.com/hpesc/public/docDisplay?docId=emr_na-c04656675

[82] Cisco OpenFlow Agent for Nexus 3000 and 9000 Series Switches, 2020, https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus/openflow/b_openflow_agent_nxos_n3kn9k.pdf

[83] W. You, K. Qian, Y. Qian, "Software-defined network flow table overflow attacks and countermeasures," International Journal of Soft Computing and Networking, vol. 1, no. 1, pp. 70-81, 2016, https://doi.org/10.1504/IJSCN.2016.077044

[84] M. Kuerban et al., "FlowSec: DOS Attack Mitigation Strategy on SDN Controller," 2016 IEEE International Conference on Networking, Architecture and Storage (NAS), pp. 1-2, 2016, doi: 10.1109/NAS.2016.7549402.

[85] A. F. M. Piedrahita et al., "Flowfence: a denial of service defense system for software defined networking," Global Information Infrastructure and Networking Symposium (GIIS), pp. 1-6, 2015, doi: 10.1109/GIIS.2015.7347185.

[86] E. Kaljic, A. Maric, P. Njemcevic, "An implementation of a deeply programmable SDN switch based on a hybrid FPGA/CPU architecture," 18th International Symposium INFOTEH-Jahorina, pp. 1-6, 2019, doi: 10.1109/INFOTEH.2019.8717768.

[87] S. Pati et al., "Design and Implementation of an FPGA Architecture for High-Speed Network Feature Extraction," 2007 International Conference on Field-Programmable Technology, pp. 49-56, 2007, doi: 10.1109/FPT.2007.4439231.

[88] O. E. Tayfour, M. N. Marsono, "Collaborative Detection and Mitigation of Distributed Denial-of-Service Attacks on Software-Defined Network," Mobile Networks and Applications 25, pp.1338–1347, 2020, https://doi.org/10.1007/s11036-020-01552-0

[89] C. Douligeris, A. Mitrokotsa, "DDoS attacks and defense mechanisms: a classification," Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology, pp. 190-193, 2003, doi: 10.1109/ISSPIT.2003.1341092.

[90] K. Muthamil Sudar, P. Deepalakshmi, "A two level security mechanism to detect a DDoS flooding attack in software-defined networks using entropy-based and C4.5 technique," Journal of High Speed Networks, Vol. 26, issue 1, pp. 55-76, 2020, https://doi.org/10.3233/JHS-200630

[91] A. Lakhina et al., "Mining anomalies using traffic feature distributions," ACM SIGCOMM Computer Communication Review, vol. 35, issue 4, pp 217-228, 2005, https://doi.org/10.1145/1090191.1080118

[92] V. Timcenko, S. Gajin, "Machine learning enhanced entropy-based network anomaly detection," Advances in Electrical and Computer Engineering, vol. 21, no. 4, pp. 51-60, 2021, doi:10.4316/AECE.2021.04006

[93] I. Basicevic, S. Ocovaj, M. Popovic, "Evaluation of entropy-based detection of outbound denial-of-service attacks in edge networks," Security and Communication Networks, vol. 8, issue 5, pp. 837-844, 2015, doi: 10.1002/sec.1040

[94] D. Parfenov et al., "Development of a solution for identifying network attacks based on adaptive neuro-fuzzy networks ANFIS," Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), pp. 491-495, 2021, doi: 10.1109/USBEREIT51232.2021.9455115.

[95] S. Dong, K. Abbas, R. Jain, "A Survey on Distributed Denial of Service (DDoS) Attacks in SDN and Cloud Computing Environments," in IEEE Access, vol. 7, pp. 80813-80828, 2019, doi: 10.1109/ACCESS.2019.2922196.

[96] A. H. Janabi, T. Kanakis, M. Johnson, "Overhead Reduction Technique for Software-Defined Network Based Intrusion Detection Systems," in IEEE Access, vol. 10, pp. 66481-66491, 2022, doi: 10.1109/ACCESS.2022.3184722.

[97] K. S. Sahoo et al., "An Evolutionary SVM Model for DDOS Attack Detection in Software Defined Networks," in IEEE Access, vol. 8, pp. 132502-132513, 2020, doi: 10.1109/ACCESS.2020.3009733.

[98] Myo Myint Oo et al., "Advanced Support Vector Machine- (ASVM-) Based Detection for Distributed Denial of Service (DDoS) Attack on Software Defined Networking (SDN)", Journal of Computer Networks and Communications, vol. 2019, article ID 8012568, 2019, https://doi.org/10.1155/2019/8012568

[99] R. Swami, M. Dave, V. Ranga, "Detection and Analysis of TCP-SYN DDoS Attack in Software-Defined Networking," Wireless Personal Communications 118, pp. 2295-2317, 2021, https://doi.org/10.1007/s11277-021-08127-6

[100] Q. Niyaz, W. Sun, A. Y. Javaid, "A Deep Learning Based DDoS Detection System in Software-Defined Networking (SDN) " ICST Transactions on Security and Safety, vol. 4, no. 12, pp. 153-515, 2017, doi = {10.4108/eai.28-12-2017.153515},

[101] Shannon, C. E., "A Mathematical Theory of Communication," The Bell System Technical Journal, vol. 27, pp. 379-423, 623-656, (1948)

[102] A. Rényi, "On measures of information and entropy," Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability 1960, pp. 547-561, (1961)

[103] C. Tsallis, "Possible generalization of Boltzmann-Gibbs statistics," Journal of Statistical Physics 52, pp. 479-487, doi:10.1007/BF01016429 (1988).

[104] S. Khan, A. Gani, A. W. A. Wahab, et al., "Feature Selection of Denial-of-Service Attacks Using Entropy and Granular Computing," Arabian Journal for Science and Eng. 43, pp. 499-508, 2018, https://doi.org/10.1007/s13369-017-2634-8

[105] C. Wang, J. Zheng, X. Li, "Research on DDoS Attacks Detection Based on RDF-SVM," 10th International Conference on Intelligent Computation Technology and Automation (ICICTA), pp. 161-165, 2017, doi: 10.1109/ICICTA.2017.43.

[106] I. Basicevic, S. Ocovaj, "Application of entropy formulas in detection of denial-of-service attacks," International Journal of Communication Systems 32(1-2):e4067, 2019, doi:10.1002/dac.4067

[107] Y. Zhou et al. "Research on DDoS Attack Detection based on Multi-dimensional Entropy," IEEE 9th International Conference on Computer Science and Network Technology (ICCSNT), pp. 65-69, 2021, doi: 10.1109/ICCSNT53786.2021.9615450.

[108] S. Gajin, V. Timcenko, "Comparison of entropy-based and machine learning approaches in intrusion detection" in 11th International Conference ICIST2021, pp.113-118, 2021, https://www.eventiotic.com/eventiotic/library/paper/651

[109] I. Juma, S. Gajin, "Entropy-based network traffic anomaly classification method resilient to deception," Computer Science and Information Systems, vol. 19, no.1, pp. 87-116, 2022, doi:10.2298/CSIS201229045I

[110] Tom M. Mitchell "The Discipline of Machine Learning", CMU-ML-06-108, Carnegie Mellon University, 2006, http://www.cs.cmu.edu/~tom/pubs/MachineLearning.pdf

[111] S. Atasever, İ. Özçelık, Ş. Sağiroğlu, "An Overview of Machine Learning Based Approaches in DDoS Detection," 2020 28th Signal Processing and Communications Applications Conference (SIU), pp. 1-4, 2020, doi: 10.1109/SIU49456.2020.9302121.

[112] A. Aljuhani, "Machine Learning Approaches for Combating Distributed Denial of Service Attacks in Modern Networking Environments," in IEEE Access, vol. 9, pp. 42236-42264, 2021, doi: 10.1109/ACCESS.2021.3062909.

[113] Xin Xu, "Adaptive Intrusion Detection Based on Machine Learning: Feature Extraction, Classifier Construction and Sequential Pattern Prediction," International Journal of Web Services Practices, vol.2, no.1-2, pp. 49-58, 2006.

[114] S. Vattikuti et al., "DDoS Attack Detection and Mitigation using Anomaly Detection and Machine Learning Models," 2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), pp. 1-6, 2021, doi: 10.1109/CSITSS54238.2021.9683214.

[115] Y. Feng et al., "Feature Selection for Machine Learning-Based Early Detection of Distributed Cyber Attacks," IEEE 16th Internat. Conference DASC, 2018, pp. 173-180, doi: 10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00040.

[116] M. A. El Mrabet et al., "Supervised Machine Learning: A Survey," 2021 4th International Conference on Advanced Communication Technologies and Networking (CommNet), pp. 1-10, 2021, doi: 10.1109/CommNet52204.2021.9641998.

[117] M. M. Raikar et al., "Data Traffic Classification in Software Defined Networks (SDN) using supervised-learning," Procedia Computer Science, vol. 171, 2020, pp. 2750-2759, https://doi.org/10.1016/j.procs.2020.04.299.

[118] D. Vallejo-Huanga, S. Vizcaíno, "LAN Network Optimization after a DDoS Attack Detected with Supervised Learning," 2022 IEEE 2nd International Conference on Computer Communication and Artificial Intelligence (CCAI), pp. 108-114, 2022, doi: 10.1109/CCAI55564.2022.9807697.

[119] F. Rustam, M. F. Mushtaq, A. Hamza at al., "Denial of Service Attack Classification Using Machine Learning with Multi-Features," Electronics 2022, 11, (22):3817. https://doi.org/10.3390/electronics11223817

[120] E. Fahlman, F. Ostlund, "Complexity and its effect on Classification Accuracy in Multi Class Classification Problems" Degree project in Computer Science and Engineering, KTH Royal Institute of Technology, 2022.

[121] T. Radivilova et al., "Classification Methods of Machine Learning to Detect DDoS Attacks," 2019 10th IEEE International Conference IDAACS, pp. 207-210, 2019, doi: 10.1109/IDAACS.2019.8924406.

[122] G. V. Patil, K. V. Pachgare et al., "Feature Reduction in Flow Based Intrusion Detection System," 2018 3rd IEEE International Conference (RTEICT), 2018, pp. 1356-1362, doi: 10.1109/RTEICT42901.2018.9012554.

[123] M. Botha, R. V. Solms, "Utilizing Neural Networks for Effective Intrusion Detection," Published in ISSA 2004, pp. 1-15, 2004.

[124] Z. E. Mrabet et al., "Detection of the False Data Injection Attack in Home Area Networks using ANN," 2019 IEEE International Conference on Electro Information Technology (EIT), pp. 176-181, 2019, doi: 10.1109/EIT.2019.8834036.

[125] M. Nikolić, A. Zečević, Mašinsko učenje, Skripta za predavanja iz predmeta Mašinsko učenje, Matematički fakultet Beograd, http://ml.matf.bg.ac.rs/readings/ml.pdf

[126] K. S. Sahoo et al., "An Evolutionary SVM Model for DDOS Attack Detection in Software Defined Networks," in IEEE Access, vol. 8, pp. 132502-132513, 2020, doi: 10.1109/ACCESS.2020.3009733.

[127] G. Shang-fu, Z. Chun-lan, "Intrusion detection system based on classification," 2012 IEEE International Conference on Intelligent Control, Automatic Detection and High-End Equipment, pp. 78-83, 2012, doi: 10.1109/ICADE.2012.6330103.

[128] R. F. Fouladi et al., "Frequency based DDoS attack detection approach using naive Bayes classification," 39th International Conference on Telecommunications and Signal Processing (TSP), pp. 104-107, 2016, doi: 10.1109/TSP.2016.7760838.

[129] J. Zhou et al., "Multi-Traffic Features Network Intrusion Detection Algorithm Based on C4.5," 18th International Computer Conference ICCWAMTIP, pp. 548-552, 2021, doi: 10.1109/ICCWAMTIP53232.2021.9674129.

[130] Y. Xu, H. Sun, F. Xiang, Z. Sun, "Efficient DDoS Detection Based on K-FKNN in Software Defined Networks," in IEEE Access, vol. 7, pp. 160536-160545, 2019, doi: 10.1109/ACCESS.2019.2950945.

[131] S. Dong, M. Sarem, "DDoS Attack Detection Method Based on Improved KNN With the Degree of DDoS Attack in Software-Defined Networks," in IEEE Access, vol. 8, pp. 5039-5048, 2020, doi: 10.1109/ACCESS.2019.2963077.

[132] Y. Chen, J. Hou, Q. Li, H. Long, "DDoS Attack Detection Based on Random Forest," 2020 IEEE International Conference on Progress in Informatics and Computing (PIC), pp. 328-334, 2020, doi: 10.1109/PIC50277.2020.9350788.

[133] J. Tan et al. "DDoS detection method based on Gini impurity and random forest in SDN environment," International Conference on Security, Pattern Analysis, and Cybernetics（SPAC), pp. 601-606, 2021, doi: 10.1109/SPAC53836.2021.9539920.

[134] M. A. Hall, "Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning" in in Proceedings of the 17th International Conference on Machine Learning (ICML '00), pp. 359–366, 2000.

[135] G. I. Guyon, A. Elisseeff, "An Introduction to Variable and Feature Selection," Journal of Machine Learning Research, vol. 3, pp. 1157-1182, 2003.

[136] Y. Qian, P. Bhattacharya, W. You, K. Qian, "Security Threat Analysis of SDN Switch Flow Table," 2018 27th International Conference on Computer Communication and Networks (ICCCN), pp. 1-2, 2018, doi: 10.1109/ICCCN.2018.8487385.

[137] D. Balagopal, X. A. K. Rani, "NetWatch: Empowering software-defined network switches for packet filtering," 2015 International Conference (iCATccT), pp. 837-840, 2015, doi: 10.1109/ICATCCT.2015.7456999.

[138] B. Pandya, S. Parmar, Z. Saquib, A. Saxena, "Framework for securing SDN southbound communication," 2017 International Conference (ICIIECS), pp. 1-5, 2017, doi: 10.1109/ICIIECS.2017.8275912.

[139] J. Alcorn, S. Melton, C. E. Chow, "SDN On-The-Go (OTG) physical testbed," 2017 IEEE Conference on Dependable and Secure Computing, pp. 202-208, 2017, doi: 10.1109/DESEC.2017.8073808.

[140] Čabarkapa, D., Rančić, D., Pavlović, P., Milićević, M. "Investigating The Impact of Tree-based Network Topology on the SDN Controller Performance," Facta Universitatis, Series: Automatic Control and Robotics, 1(1), pp. 025-035, 2022, doi:https://doi.org/10.22190/FUACR211223003C

[141] Y. Xu et al., "SDN docker: Enabling application auto-docking/undocking in edge switch," 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 864-869, 2016, doi: 10.1109/INFCOMW.2016.7562199.

[142] C. Costache et al., "Software-defined networking of Linux containers," 2014 RoEduNet Conference 13th Edition: Networking in Education and Research Joint Event RENAM 8th Conference, pp. 1-4, 2014, doi: 10.1109/RoEduNet-RENAM.2014.6955310.

[143] D. Cabarkapa, D. Rancic, "Performance Analysis of Ryu-POX Controller in Different Tree-Based SDN Topologies," Advances in Electrical and Computer Engineering, vol.21, no.3, pp.31-38, 2021, doi:10.4316/AECE.2021.03004

[144] Po. Chi et al., "SDN Migration-An Efficient Approach to Integrate OpenFlow Networks with STP-Enabled Networks," International Computer Symposium (ICS), pp. 148-153, 2016, doi:10.1109/ICS.2016.0038

[145] I. Basicevic, N. Blazic, S. Ocovaj, "On the use of principal component analysis in the entropy based detection of denial-of-service attacks," Security and Privacy, vol. 5, issue 2, 2022, e193, doi:10.1002/spy2.193

[146] K. Phemius, M. Bouet, "OpenFlow: Why latency does matter," 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), Ghent, Belgium, pp. 680-683, 2013

[147] P. Berezinski, B. Jasiul, M. Szpyrka, "An Entropy Based Network Anomaly Detection Method," Entropy 2015, vol. 17, issue 4, pp. 2367-2408, 2015. doi:10.3390/e17042367

[148] Scapy Documentation, ver. 2.5.0, 2023, https://scapy.readthedocs.io/en/latest/

[149] Rohith Raj S et al., "SCAPY- A powerful interactive packet manipulation program," 2018 International Conference on Networking, Embedded and Wireless Systems (ICNEWS), pp. 1-5, 2018, doi: 10.1109/ICNEWS.2018.8903954.

[150] Abhishek Gupta (2023) "Machine Learning with MATLAB", version 1.3.0.1, https://www.mathworks.com/matlabcentral/fileexchange/42744-machine-learning-with-matlab

[151] A. N. Nazarov, A. K. Sychev, M. Voronkov, "The Role of Datasets when Building Next Generation Intrusion Detection Systems," 2019 (WECONF), 2019, pp. 1-5, doi: 10.1109/WECONF.2019.8840124.

[152] D. Protic, M. Stankovic, "Anomaly-Based Intrusion Detection: Feature Selection and Normalization Influence to the Machine Learning Models Accuracy," European Journal of Formal Sciences and Engineering, vol. 3, issue 1, 2020.

[153] S. S. Panwar et al., "An Intrusion Detection Model for CICIDS-2017 Dataset Using Machine Learning Algorithms," 2022 International Conference (ICACCM), pp. 1-10, 2022, doi: 10.1109/ICACCM56405.2022.10009400.

[154] N. Moustafa, J. Slay, "The Significant Features of the UNSW-NB15 and the KDD99 Data Sets for Network Intrusion Detection Systems," 2015 4th International Workshop on (BADGERS), pp. 25-31, 2015, doi: 10.1109/BADGERS.2015.014.

[155] CIC-Canadian Institute for Cybersecurity, Intrusion Detection Evaluation Dataset (CIC-IDS2017), https://www.unb.ca/cic/datasets/ids-2017.html

[156] M. S. Elsayed et al., "InSDN: A Novel SDN Intrusion Dataset," in IEEE Access, vol. 8, pp. 165263-165284, 2020, doi: 10.1109/ACCESS.2020.3022633.

[157] N. Đerić et al., "SDN Hypervisors: How Much Does Topology Abstraction Matter?," 2018 14th International Conference on CNSM, pp. 328-332, 2018.

[158] Jin Wang, Liping Wang. 2022. "SDN-Defend: A Lightweight Online Attack Detection and Mitigation System for DDoS Attacks in SDN" Sensors 22, no. 21: 8287. https://doi.org/10.3390/s22218287

[159] M. Injadat et al., "Multi-Stage Optimized Machine Learning Framework for Network Intrusion Detection," in IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1803-1816, June 2021, doi: 10.1109/TNSM.2020.3014929.

# BIOGRAFIJA AUTORA

Danijel Čabarkapa je rođen 8. juna 1969. godine u Prijepolju. Živi i radi u Šapcu. Osnovnu o srednju školu završio je u Prijepolju. Diplomirao je 1996. godine na Fakultetu tehničkih nauka Univerziteta u Novom Sadu, na odseku za elektroniku i telekomunikacije. Na istom fakuletetu je 2008. godine stekao zvanje Diplomirani inženjer elektrotehnike i računarstva - master.

Profesionalnu karijeru započeo je u kompanijama „Indas“ i “Jevon” iz Novog Sada, gde je u periodu od 1998. do 2002. godine bio zaposlen na poslovima projektovanja i održavanja mrežnih sistema. Od 2003. do 2014. godine radio je u Tehničkoj školi u Šapcu kao nastavnik grupe predmeta iz elektrotehnike i računarstva. Na Odseku za medicinske i poslovno-tehnološke studije Akademije strukovnih studija Šabac zaposlen je od 2014. kao saradnik u nastavi, a zatim kao asistent. Učestvovao je u pripremi i izvođenju računarskih vežbi iz predmeta Računarske mreže, Bezbednost i zaštita podataka, Arhitektura računara i operativni sistemi, Informacioni sistemi i Održavanje računarskih sistema.

Njegova istraživačka delatnost fokusirana je prvenstveno na oblast računarskih mreža i komunikacija. Teme istraživanja obuhvataju oblasti softverski definisanih mreža, bezbednosti računarskih mreža, kao i NDN i VANET mreža. Iz navedenih oblasti istraživanja objavio je više publikacija u međunarodnim i nacionalnim časopisima, kao i radove na skupovima međunarodnog i nacionalnog značaja.

## IZJAVA O AUTORSTVU

Izjavljujem da je doktorska disertacija, pod naslovom

**Nova metoda detekcije DDoS napada primenom softverski definisanih mreža**

koja je odbranjena na Elektronskom fakultetu Univerziteta u Nišu:

- rezultat sopstvenog istraživačkog rada;
- da ovu disertaciju, ni u celini, niti u delovima, nisam prijavljivao/la na drugim fakultetima, niti univerzitetima;
- da nisam povredio/la autorska prava, niti zloupotrebio/la intelektualnu svojinu drugih lica.

Dozvoljavam da se objave moji lični podaci, koji su u vezi sa autorstvom i dobijanjem akademskog zvanja doktora nauka, kao što su ime i prezime, godina i mesto rođenja i datum odbrane rada, i to u katalogu Biblioteke, Digitalnom repozitorijumu Univerziteta u Nišu, kao i u publikacijama Univerziteta u Nišu.

U Nišu, _____

Potpis autora disertacije:

Danijel D. Čabarkapa

**IZJAVA O ISTOVETNOSTI ELEKTRONSKOG I ŠTAMPANOG OBLIKA DOKTORSKE DISERTACIJE**

Naslov disertacije:

**Nova metoda detekcije DDoS napada primenom softverski definisanih mreža**

Izjavljujem da je elektronski oblik moje doktorske disertacije, koju sam predao za unošenje u **Digitalni repozitorijum Univerziteta u Nišu**, istovetan štampanom obliku.

U Nišu, _____

Potpis autora disertacije:

Danijel D. Čabarkapa

## IZJAVA O KORIŠĆENJU

Ovlašćujem Univerzitetsku biblioteku „Nikola Tesla“ da u Digitalni repozitorijum Univerziteta u Nišu unese moju doktorsku disertaciju, pod naslovom:

**Nova metoda detekcije DDoS napada primenom softverski definisanih mreža**

Disertaciju sa svim prilozima predao sam u elektronskom obliku, pogodnom za trajno arhiviranje.

Moju doktorsku disertaciju, unetu u Digitalni repozitorijum Univerziteta u Nišu, mogu koristiti svi koji poštuju odredbe sadržane u odabranom tipu licence Kreativne zajednice (Creative Commons), za koju sam se odlučio.

1. Autorstvo **(CC BY)**
2. Autorstvo – nekomercijalno **(CC BY-NC)**
3. **Autorstvo – nekomercijalno – bez prerade (CC BY-NC-ND)**
4. Autorstvo – nekomercijalno – deliti pod istim uslovima **(CC BY-NC-SA)**
5. Autorstvo – bez prerade **(CC BY-ND)**
6. Autorstvo – deliti pod istim uslovima **(CC BY-SA)**

U Nišu, _____

Potpis autora disertacije:

D. Čabarkapa

Danijel D. Čabarkapa