Upotreba umjetne inteligencije u NIDS sustavima

Zekan, Marko

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:508358

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Marko Zekan

UPOTREBA UMJETNE INTELIGENCIJE U NIDS SUSTAVIMA

DIPLOMSKI RAD

Varaždin, 2021.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Marko Zekan

Maticni broj: 43411/14-R

Studij: Organizacija poslovnih sustava

UPOTREBA UMJETNE INTELIGENCIJE U NIDS SUSTAVIMA

DIPLOMSKI RAD

Mentor :

Doc. dr. sc. Igor Tomičić

Varaždin, rujan 2021.

Marko Zekan

Izjava o izvornosti

Izjavljujem da je moj diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Ovaj rad istražuje uporabu metoda umjetne inteligencije na problem detekcije napada na mreži. Tradicionalni sustavi detekcije napada na mrežama (NIDS) pokazali su loše rezultate u detekciji novih napada te općenito detekciji napada u kompleksnim mrežnim infrastrukturama gdje teče mnogo podataka. Jedan od većih problema tradicionalnih sustava je taj što proizvode velik broj lažno pozitivnih alarma. Zbog toga se istraživači okreću pronalaženju novih metoda kojima bi unaprijedili NIDS sustave. Za tu svrhu u proteklom je desetljeću testirano stotine modela i tehnika strojnog i dubokog učenja na problem detekcije napada na mreži. Iako pokazuju izvrsne rezultate, ove metode često zahtijevaju mnogo više označenih podataka nego je to dostupno. Manjak označenih skupova podataka potaknuo je istraživače da pokušaju proizvesti sintetičke podatke uporabom generativnih suparničkih modela (GAN). Jedna takva metoda je nedavno razvijena EC-GAN metoda, koja je pokazala uspjeh u treniranju klasifikatora supstituiranjem manjka označenih podataka s umjetnim podacima generiranim GAN mrežom. U ovome radu implementiran je klasifikator duboke neuronske mreže koristeći EC-GAN metodu te su rezultati uspoređeni s dosadašnjim rezultatima iz ovog područja kao i sa rezultatima treniranja istog klasifikatora standardnom metodom. Rezultati pokazuju da relativno jednostavan klasifikator treniran EC-GAN metodom može konkurirati drugim rezultatima iz ovog područja iako je za njegovo treniranje bila potrebna desetina originalnog skupa podataka.

Ključne riječi: generativne suparničke mreže, EC-GAN, WCGAN-GP, sintetički podaci, klasifikacija, duboke neuronske mreže, polu-nadgledano učenje, sustavi za detekciju upada, CIC-DIS-2017

Sadržaj

1. Uvod

Objašnjenje pojmova

GAN (eng. *Generative Adverserial Networks*) - Generativne suparničke mreže

EC-GAN (eng. *External Classifier – GAN*) - GAN s vanjskim klasifikatorom

CGAN (eng. *Conditional GAN*) – Uvjetovani GAN

WGAN (eng. *Wasserstein GAN*) – Wasserstein GAN

WCGAN-GP (eng. *Wasserstein Conditional GAN with Gradient Penalty*) – Uvjetovani Wasserstein GAN sa gradijentnom kaznom

IDS (eng. *Intrusion Detection System*) - Sustav za otkrivanje upada

NIDS (eng. *Network Intrusion Detection System*) - Sustav za otkrivanje upada na mreži

UI - Umjetna inteligencija

DNN (eng. *Deep Neural Network*) – Duboka neuronska mreža

Uspjeh NIDS sustava koji se oslanjaju na strojno/duboko učenje za otkrivanje anomalija uvelike ovisi o podacima na kojima su modeli trenirani. Podaci za treniranje dubokih nadgledanih modela moraju biti obilni, kvalitetni, označeni i klase moraju biti dobro balansirane. Problem nastaje jer su takvi skupovi podataka vrlo teški i skupi za stvoriti [1].

Imajući to na umu, glavna motivacija za ovaj rad bila je upotrijebiti neku inovativnu metodu na problem klasifikacije mrežnog prometa koja će se moći nositi s navedenim problemima, a da se pritom ostvare konkurentni rezultati klasifikacije.

Navedene probleme i ciljeve ovaj rad nastoji riješiti uporabom inovativne EC-GAN [2] metode treniranja nadgledanog klasifikatora. Ova metoda polu-nadgledanog učenja nedavno je predstavljena kao rješenje za treniranje potpuno nadgledanog klasifikatora korištenjem kombinacije stvarnih i sintetički generiranih podataka u svrhe treniranja modela na malim, realističnim skupovima podataka [2].

Rezultati rada postignuti su treniranjem klasifikatora prvo standardnom zatim EC-GAN metodom nad progresivno sve manjem originalnom skupu podataka te su na kraju uspoređeni rezultati evaluacije oba modela kao i rezultati ostalih modela iz ovog područja.

U radu je pokazano kako klasifikator treniran EC-GAN metodom može konkurirati najboljim rezultatima iz ovoga područja iako je potrebno i do 90% manje podataka za treniranje te da premašuje rezultate ostvarene treniranjem klasifikatora standardnim tehnikama.

Pregled domene rada

Sustavi za detekciju upada na mreži (eng. *Network Intrusion Detection Systems* – NIDS) obavljaju zadatak analize paketa (eng. *packet sniffing*) i mrežnog prometa kako bi identificirali sumnjive aktivnosti i zabilježili relevantne informacije [1], [3]. Spadaju pod veći skup sustava za detekciju upada (eng. *Intrusion Detection Systems* – IDS) gdje se nalaze još i

HIDS (eng. *Host IDS*) i WIDS (eng. *Wireless IDS*) sustavi [1]. Nadalje, osim prema domeni na kojoj djeluju (mreža/domaćin), IDS sustavi se dijele i prema metodama detekcije napada. Dvije glavne podijele su na sustave koji rade na principu pronalaska malicioznih potpisa (eng. *Signature-based IDS*) i one koji rade temeljem pronalaska anomalija (eng. *Anomaly-based IDS*) [4]. Ovaj rad bavi se treniranjem klasifikatora koji bi se koristio za otkrivanje anomalija u mrežnom prometu.

Uporaba strojnog i dubokog učenja u NIDS sustavima se u proteklom desetljeću pokazala veoma efikasnom u predviđanju abnormalnosti [5]. Mogućnost tih modela da nauče prepoznavati kompleksne uzorke među podacima omogućuje im da detektiraju do sad ne viđene napade [6]. Brojni algoritmi poput: K najbližih susjeda, autoenkodera i generativnih suparničkih mreža korišteni su za detekciju napada, no ne postoji jedinstveno najbolje rješenje te svaki pristup ima svoje prednosti i mane. Na primjer, nadgledani (eng. *supervised*) algoritmi strojnog učenja rade bolje od nenadgledanih algoritama (eng. *unsupervised*) za ovaj zadatak [1], ali zahtijevaju ekstenzivno inženjerstvo značajki (eng. *feature engineering*) [5]. S druge strane algoritmi dubokog učenja ne zahtijevaju inženjerstvo značajki [5], ali su potrebne velike količine podataka za treniranje i validaciju. Negdje između nadgledanih i nenadgledanih algoritama nalaze se polu-nadgledani (eng. *semi-supervised*) algoritmi koji uglavnom koriste kombinirane označene i neoznačene podatke za učenje [7] te tako pokušavaju premostiti navedene probleme. Jedan primjer metode polu-nadgledanog učenja je ranije spomenuta EC-GAN metoda [2] koja kombinira nenadgledani GAN algoritam s nadgledanim DNN klasifikatorom.

Ključan dio EC-GAN metode je upravo njena mogućnost kreiranja novih podataka temeljem stvarnih podatka korištenjem GAN mreže. Ove mreže, prvi put predstavljene 2014. godine u [8], dosegle su veliku popularnost zbog njihove sposobnosti generiranja do sada neviđenih podataka visoke kvalitete. Glavna karakteristika GAN mreža je njihova uporaba dvije različite neuronske mreže, zvane generator i diskriminator, u suparničkom okružju [8] gdje diskriminator pokušava osuditi jesu li njegovi ulazni podaci stvarni ili generirani od strane generatora. Cilj treniranja je da generator trenira toliko kvalitetne podatke da ih diskriminator ne može razlikovati od stvarnih. GAN mreže su od svoga nastanka primijenjene na velik broj zadataka: procesiranje fotografija, otkrivanje lica, transfer tekstura [9] te danas postoji cijeli spektar specijaliziranih implementacija koje rješavaju specifične probleme poput: CycleGAN [10], StyleGAN [11], WGAN [12] itd. Jedna varijacija WGAN mreže zvana WCGAN-GP [13] korištena je za kreiranje sintetičkih podataka u implementaciji ovoga rada.

U originalnom radu [2] EC-GAN, odnosno generativna suparnička mreža s vanjskim klasifikatorom korištena je za generiranje i klasifikaciju X-Ray skupa podataka [14], pri čemu je za klasifikaciju korišten ResNet-18 algoritam [15] dok je za generiranje novih slika korišten DCGAN (eng. *Deep Convolutional GAN*) algoritam [16]. Navedena dva algoritma specijalizirani su za klasifikaciju i generiranje slika, stoga su u ovome radu supstituirani algoritmima koji obavljaju iste zadatke, ali za tablične podatke. Za potrebe generiranju sintetičkih tabličnih podataka korištena je ranije navedena WCGAN-GP metoda [13], dok je za klasifikaciju korištena duboka neuronska mreža (DNN).

Postavljanje problema

Problemi koje ovaj rad nastoji riješiti mogu se svrstati u dvije domene – problemi vezani uz umjetnu inteligenciju i problemi vezani uz NIDS sustave. Problemi UI, specifično klasifikacijskih metoda tj. nadgledanog učenja, tiču se samih podataka koji se koriste za treniranje. Ti podaci moraju biti kvalitetni, odnosno dobro balansirani, označeni i u slučaju dubokog učenja mora ih biti vrlo mnogo. Problem je u tome što je proces označavanja (eng. *labeling*) podataka od strane domenskih stručnjaka vrlo skup [1], tim više ako je potreban veliki skup za treniranje nekog modela dubokog učenja. Nadalje, problem ne leži samo u količini označenih podataka, nego i u njihovoj kvaliteti. Podaci moraju biti dobro balansirani, odnosno da bi model naučio dobro razlikovati pojedine klase skupa, potrebno je imati dovoljno članova svake klase kako bi model imao dovoljno primjera na kojima učiti [17]. Problem nebalansiranih klasa samo raste kako raste i skup podataka [17].

Problemi vezani uz domenu NIDS sustava tiču se stope lažnih pozitiva. Cilj svakog IDS-a je imati što manje lažnih alarma, tj. prepoznati maliciozne pakete/napade samo kad se uistinu događaju jer mnogo lažnih pozitiva može uzrokovati da pravi napadi ne dolaze do izražaja, odnosno da ih sigurnosni operateri previde [18].

Definiranje cilja rada

Cilj ovoga rada je dakle ponuditi rješenja za ranije navedene probleme. Glavni cilj rada može se sročiti kao „ostvariti rezultate klasificiranja mrežnog prometa koji konkuriraju drugim radovima iz ovog područja uz korištenje manje podataka za treniranje modela“. Za postizanje ovog cilja potrebno je ostvariti nekolicinu manjih ciljeva kao što su: prenamijeniti EC-GAN metodu za rad sa sintetičkim podacima i proizvesti sintezu tabličnih podataka koji se ne razlikuju od pravog skupa podataka.

Detalji oko metoda i tehnika kojima su ciljevi ostvareni mogu se pronaći u sljedećem poglavlju.

Glavni doprinosi

Glavni doprinosi ovoga rada su: primjena inovativne EC-GAN metode na klasifikaciju mrežnog prometa, prva primjena EC-GAN metode na klasifikaciju tabličnih podataka te uspješna primjena WCGAN-GP algoritma za sintezu podataka iz CIC-IDS-2017 [19] skupa podataka.

U ovom poglavlju dan je pregled tema kojima se rad bavi, navedeni su problemi koje rad rješava, glavni ciljevi te su na kraju predstavljeni njegovi glavni doprinosi.

U nastavku rada slijedi poglavlje u kojem su opisane metode i tehnike kojima rad ostvaruje navedene ciljeve. Nakon toga slijedi poglavlje o primjeni umjetne inteligencije u NIDS sustavima, gdje je dan pregled područja, spomenute su korištene metode kao i do sada ostvareni rezultati. Potom slijedi poglavlje posvećeno samoj EC-GAN metodi gdje je pobliže objašnjeno kako sama metoda i njeni sastavni dijelovi rade te koje probleme rješavaju. Ovo poglavlje također se dotiče tema sintetičkih podataka i klasifikatora. Zadnji dio poglavlja posvećen je opisu rješenja ovog rada.

Zadnja velika cjelina rada bavi se praktičnim dijelom rada. Objašnjena je implementacija EC-GAN rješenja te su istaknuti bitni i zanimljivi dijelovi implementacije. Također, dan je pregled procesa treniranja algoritama, njihove arhitekture, vrijednosti hiperparametara kao i pregled samog CIC-IDS-2017 skupa podataka koji se koristi za treniranje.

2. Metodologija

U ovom poglavlju dan je kratak pregled metoda i tehnika koje su korištene za postizanje ciljeva rada kao i metodologija kojom su dobiveni rezultati. Detaljniji pregled metoda i detalji vezani uz samu implementaciju nalaze se u petom poglavlju.

Većina problema navedenih u uvodu riješeno je uporabom EC-GAN metode. Problem malih podataka (eng. *low-sample dana*) EC-GAN rješava nadopunjavanjem malih skupova podataka s validnim sintetičkim podacima generiranim korištenjem GAN mreže. Nadalje, problem loše balansiranih podataka (eng. *class imbalance*) riješen je uvjetovanjem GAN mreže da generira one klase čijih članova ima malo, umjesto da nasumično generira klase skupa. Ovime se postiže da se originalni skup podataka nadopuni s više jedinki klasa malo članova. Za ove potrebe korištena je posebna vrsta GAN mreža zvana CGAN (eng. *Conditional GAN*) [20]] čija karakteristika je da može generirati točno određenu klasu podataka iz skupa za treniranje tako da se vrijednost te klase proslijedi kao ulaz u mrežu (eng. *input*).

Rješavanje problema treniranja modela na malim, loše balansiranim skupovima podataka trebalo bi imati za posljedicu i manju stopu lažnih pozitiva.

Za uspješno treniranje bilo kojeg modela strojnog učenja potrebno je odabrati prikladan skup podataka te ga pravilno procesirati prije samog treniranja kako bi se postigli što bolji rezultati. Pred-procesiranje odabranog skupa podataka (CIC-IDS-2017) odrađeno je tako su sve vrijednosti skupa prvo standardizirane, potom je odrađena PCA (eng. *Principal Component Analysis*) te je na kraju provedena i normalizacija vrijednosti.

Za potrebe generiranja sintetičkih tabličnih podataka EC-GAN metodom, korištena je specifična implementacija CGAN mreže zvana Wasserstein CGAN, odnosno WCGAN [21]. Ova metoda pokazala je dobre rezultate u sintezi tabličnih podataka u više područja [21], [22].

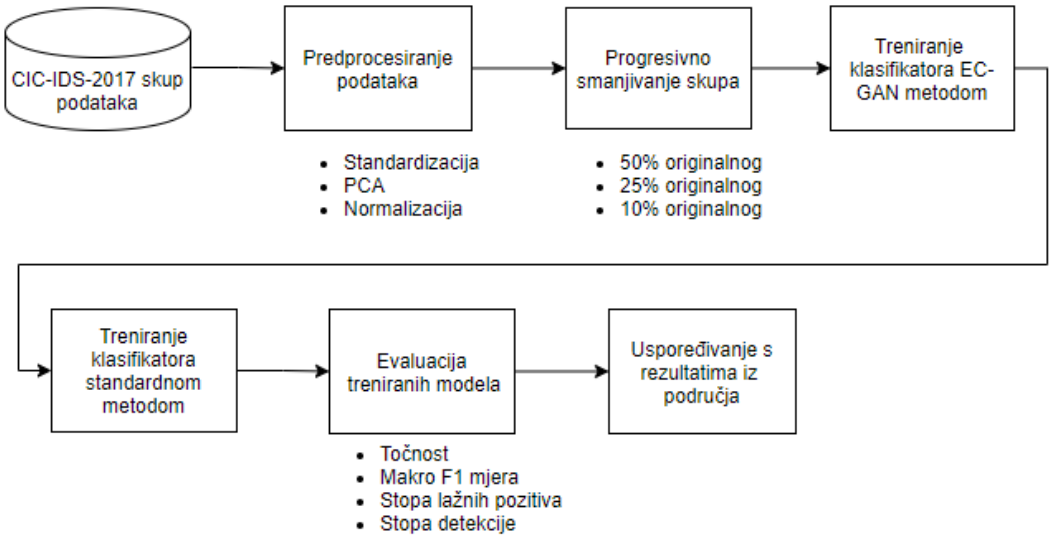
Smanjenim skupovima podataka prvo je treniran klasifikator EC-GAN metodom, a potom je isti klasifikator treniran standardnom metodom. Na ovaj način se mogu kasnije usporediti rezultati.

Rezultati rada uspoređeni su po nekoliko ključnih metrika:

- Točnost modela (eng. *accuracy*)
- Makro F1 mjera (eng. *macro F1 score*)
- Stopa lažnih pozitiva (eng. *false positive rate*)
- Stopa detekcije / osjetljivost (eng. *detection rate / sensitivity*)

Na kraju rada dan je pregled rezultata treniranja klasifikatora standardnom i EC-GAN metodom, a zatim su ti rezultati uspoređeni s jednim od boljih rezultata u ovom području.

Na slici 1 se može vidjeti tijek obavljanja aktivnosti koje su bile potrebne za dobivanje rezultata u ovome radu.



Slika 1: Dijagram tijeka provođenja eksperimenta ovoga rada [autorski rad]

3. Umjetna inteligencija u NIDS sustavima - pregled područja

Ovo poglavlje daje pregled područja uporabe umjetne inteligencije u NIDS sustavima. U nastavku je dan pregled literature, korištenih metoda i tehnika, skupova podataka te do sada ostvarenih rezultata.

U današnje doba velikog napretka informacijsko komunikacije tehnologije gdje skoro svako poslovanje u svojoj suštini ovisi o Internet-u, sigurnost mrežnih sustava bitnija je nego ikad prije. Za osiguravanje svoje infrastrukture u digitalnoj sferi tvrtke koriste sustave poput vatroštita, antivirusnih programa i IDS sustava. NIDS, odnosno sustavi za detekciju upada na mreži se u takvim sustavima koriste za otkrivanje napada na mreži koristeći tehnike analize mrežnih paketa i detekcije neuobičajenog ponašanja [3], [5].

Iako su u uporabi dugo vremena, sustavi za detekciju upada i dalje proizvode visoku stopu lažno pozitivnih alarma koja uzrokuje probleme detekcije stvarnih napada te stavlja veliki teret na sigurnosne analitičare. Osim navedenog, dodatan problem predstavlja nemogućnost današnjih IDS sustava da detektiraju nove napade [6]. Nadalje, nagle promjene u veličini mrežne infrastrukture i broju aplikacija na čvorovima postale su plodno tlo za pojavu novih sofisticiranih napada protiv kojih današnji IDS sustavi ne mogu štititi [5].

Kako bi riješili problem detekcije novih napada, stope lažnih pozitiva i napravili sustave prikladne današnjem dobu istraživači su se fokusirali na pronalazak boljih metoda detekcije. Za te potrebe istraživači su se okrenuli korištenju metoda strojnog i dubokog učenja [5]. Sustavi bazirani na ovim metodama pokazali su dobre rezultate u detekciji poznatih, ali i novih napada uz uvjet da su modeli trenirani na kvalitetnim podacima [6]. U nastavku je dan pregled korištenih metoda i tehnika strojnog i dubokog učenja koje se danas koriste u NIDS sustavima.

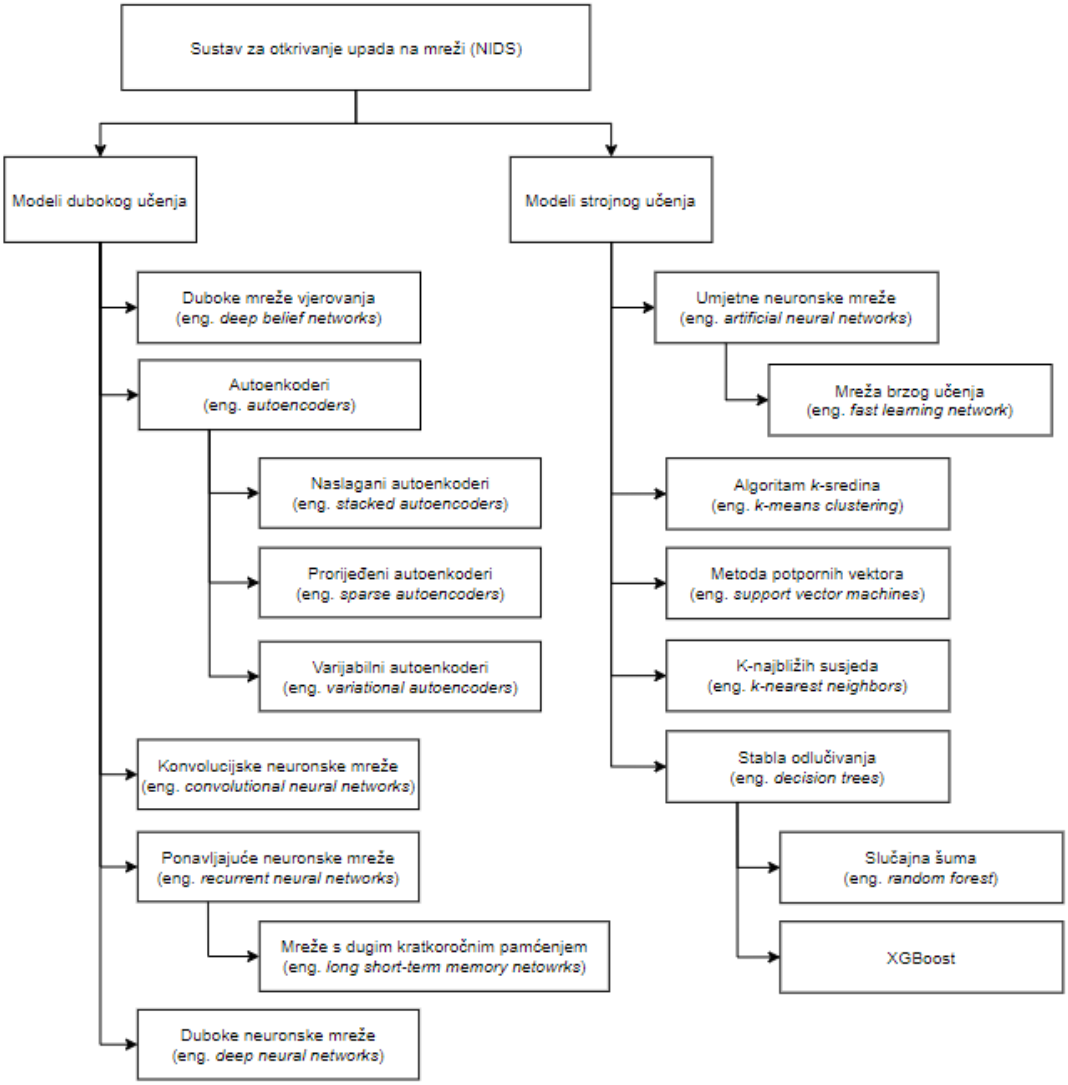
3.1. Korištene metode i tehnike

Područje uporabe metoda umjetne inteligencije u NIDS sustavima doseglo je nagli rast proteklih godina. Napravljeno je mnogo novih istraživanja [5], [6], [23]–[25], velik broj nadgledanih i nenadgledanih algoritama strojnog i dubokog učenja primijenjen je na problem detekcije mrežnih napada te je došlo do razvoja novih skupova podataka za treniranje tih modela, poput CIC-IDS skupova [26]. Ovaj dio rada posvećen je pregledu područja kroz nekoliko preglednih radova.

Ahmad, Kahn, Shiang, Abdullah i Ahmad u svome radu [5] daju pregled radova iz područja uporabe umjetne inteligencije u NIDS sustavima. Navode koje se metode UI koriste, koje zaključke su izvorni autori donijeli, nad kojim skupovima podataka su rezultati dobiveni te koje su bile metrike ocjene uspjeha tih modela. Ahmad i sur. [5] iznose nekoliko zanimljivih zaključaka u svom preglednom radu. Na primjer, 60% pregledanih radova koristi duboko učenja umjesto strojnog učenja za postizanje rezultata, od kojih se najviše koristi nenadgledana metoda autoenkodera. Autori navode kako preko 60% radova koristi metrike: točnost (eng. *ac-*

curacy), specifičnost (eng. *precision*), specifičnost (eng. *recall*) i F1 mjeru (eng. *F1 score*) za mjerenje uspješnosti rezultata. Također, zanimljiv je podatak da samo 12% pregledanih radova koristi CIC-IDS skupove podataka, iako Ring, Wunderlich, Scheuring, Landes i Hotho u svome radu [27] preporučuju njihovo korištenje.

Dijagram na slici 2 prikazuje taksonomiju metoda strojnog i dubokog učenja za korištenje u NIDS sustavima prema [5].



Slika 2: Dijagram taksonomije modela strojnog i dubokog učenja u NIDS sustavima [autoriski rad, prema [5]]

Lio i Lang u svojem preglednom radu iz 2019. godine [6] također daju pregled metoda strojnog i dubokog učenja u IDS sustavima. Dolaze do sličnih zaključaka kao Ahmad i sur. vezano uz korištene skupove podataka i evaluacijske metrike koje radovi koriste za prikaz rezultata. Lio i Lang navode iste metode kao Ahmad i sur., uz dodatak sljedećih:

- Ograničeni Boltzmannovi strojevi (eng. *Restricted Boltzman machines*)
- Generativne suparničke mreže
- Naivni Bayes
- Logistička regresija

Zadnji pregledni rad koji je analiziran u ovom poglavlju tiče se uporabe generativnih suparničkih modela u području kibernetičke sigurnosti. Dutta, Gohsh, Kyle, Totaro i Bayoumi u preglednom radu GAN mreža u sigurnosti [28] navode nekoliko područja uporabe ovih modela: skrivanje osjetljivih informacija, otkrivanje upada u kibernetičkom okružju i detekcija zloćudnog koda, stenografija, neuralna kriptografija i sigurnosna analiza. Autori navode šest radova koji koriste GAN mreže za detekciju upada. Generativni modeli u pitanju su:

- BiGAN (eng. *Bidirectional GAN*) – Dvosmjerni GAN [29]
- Vanilla GAN – obična implementacija GAN mreže
- Wasserstein GAN [12]
- Conditional GAN [20]

Analizom preglednih radova možemo zaključiti da je u području NIDS sustava korištena većina danas postojećih modela umjetne inteligencije. Gledano s najviše razine podijele prema nižima, spomenuti su modeli strojnog i dubokog učenja, zatim nadgledanog i nenadgledanog učenja kao i klasifikacijski modeli, modeli za klasteriranje, generativni modeli, konvolucijski, povratni kao i modeli regresijskog učenja.

3.2. Dosadašnji rezultati

U ovom poglavlju prikazani su rezultati koje su istraživači do sada ostvarili u kontekstu klasifikacije napada na mreži. Rezultati su također prikazani kroz nekoliko preglednih radova. Ranije je navedeno da se zaista velik broj metoda strojnog i dubokog učenja koristi za detektiranje anomalija. Zaključak koji je proizašao iz tih istraživanja je da modeli dubokog učenja nadmašuju modele strojnog za ovaj zadatak [1].

Nekoliko izvora [5], [6] navodi prednosti dubokih modela. Naime, duboki modeli izvrsno rade s velikim brojem podataka, mogu automatski naučiti unutarnju strukturu podataka te ne zahtijevaju inženjerstvo značajki dok klasični modeli strojnog učenja to zahtijevaju.

Unatoč tom zaključku, Maseer, Yusof, Bahaman, Mostafa i Foozy u preglednom radu iz 2021. [30] daju pregled metoda strojnog učenja na problem detektiranja anomalija u IDS sustavima koristeći CIC-IDS-2017 skup podataka. Autori su testirali desetak metoda strojnog učenja te pronalaze da metode poput k-najbližih susjeda, stabla odlučivanja, naivni Bayes i umjetne neuronske mreže daju najbolje rezultate u detektiraju Web napada, dok metode poput Samo-organizirajućih mapa i Algoritam k-sredina daju lošije rezultate. Autori su uspoređivali

modele prema metrikama preciznosti, specifičnosti, osjetljivosti, F1 mjeri te vremenima treniranja i testiranja.

Važno je istaknuti i rad Toupasa, Chamoua, Giannoutakisa, Dorsoua i Tzovarasa iz 2020. godine [31] u kojem su predstavili model dubokog učenja koji je postigao izvanredne rezultate. Autori su predstavili model duboke neuronske mreže namijenjen detekciji anomalija u mrežnom protoku. Za treniranje modela koristili su CIC-IDS-2017 skup podataka te u postigli sljedeće rezultate:

Tablica 1: Rezultati evaluacije dubokog modela [autorski rad, prema [31]]

Točnost	F1 mjera	Stopa lažnih alarma	Stopa detekcije
.9995	.9410	.0005	.9561

Rezultati evaluacije ovoga rada bit će uspoređeni s navedenim rezultatima.

Za kraj je bitno napomenuti da većina radova spomenutih u navedenim preglednim radovima i dalje koristi skupove podataka poput KDD-CUP-99 i NSL-KDD [5] iako su do sada razvijeni skupovi poput CIC-IDS koji sadrže primjere više napada kao i detaljnije meta podatke o napadima [27]. Korištenje starijih ili manje kvalitetnih skupova podataka može uvelike utjecati na performanse inače kompetentnih arhitektura modela, što implicira da neka arhitektura nije nužno lošija od druge nego nije trenirana na pravim podacima.

U nastavku rada slijedi poglavlje o samoj EC-GAN metodi treniranja klasifikatora.

4. EC-GAN metoda

Ovo poglavlje bavi se detaljnim opisom EC-GAN metode za treniranje nadgledanih klasifikatora. Opisana je razlog postojanja ove metode, njega upotreba, arhitektura, a zatim opis i arhitektura njezinih sastavnih dijelova. Na kraju poglavlja dan je opis rješenja ovoga rada.

EC-GAN, odnosno generativna suparnička mreža s vanjskim klasifikatorom predložena je 2021. godine u [2] kao rješenje za klasifikaciju malih, realističnih skupova podataka. Autor ističe kako postoje područja gdje su dostupni samo neoznačeni podaci koji su skupi za označiti, ali i područja gdje su sami podaci teški za skupiti no ne nužno i za označiti. Za ovaj problem autor daje primjer sakupljanja rendgenskih snimki gdje postoji mnoštvo koraka do izrade cjelovitog skupa podataka. Ovakvi kompleksni procesi sakupljanja su skupi i dovode do skupova podataka s malo članova (eng. *low-sample data*) koji uvelike utječu na djelotvornost dubokih modela [2].

Autor ističe da inovativnost EC-GAN metode lezi u tome sto se sintetički podaci generiraju paralelno s treniranjem klasifikatora te koriste povratnu informaciju od klasifikatora za daljnje treniranje. Također, sam pristup vanjskog klasifikatora je inovativan, druge metode koriste metodu dijeljene arhitekture (eng. *shared architecture method*) gdje se diskriminator ujedno koristi za treniranje generatora i kasnije kao klasifikator.

Kao glavne doprinose rada autor navodi: predstavljanje novog modela koji koristi GAN mreže za pomoć klasifikaciji korištenjem umjetnih podataka, intuitivnu arhitekturu modela koja razdvaja diskriminator i klasifikator te provođenje detaljnih eksperimenata nad ograničenim nadgledanim skupom podataka [2].

Izvorni rad pokazao je primjenu EC-GAN metode za treniranje klasifikatora naziva ResNet18 [15] koji je tip konvolucijske neuronske mreže. Generiranje umjetnih podataka za pomoć klasifikaciji obavljeno je korištenjem DCGAN algoritma [16] koji također u suštini koristi konvolucijske neuronske mreže kao generator i diskriminator. U radu se koristi Chest X-Ray [14] skup podataka za treniranje navedenih modela, kojeg karakterizira mali broj podataka za treniranje i validaciju, što ga čini prikladnim za testiranje navedenih prednosti EC-GAN metode.

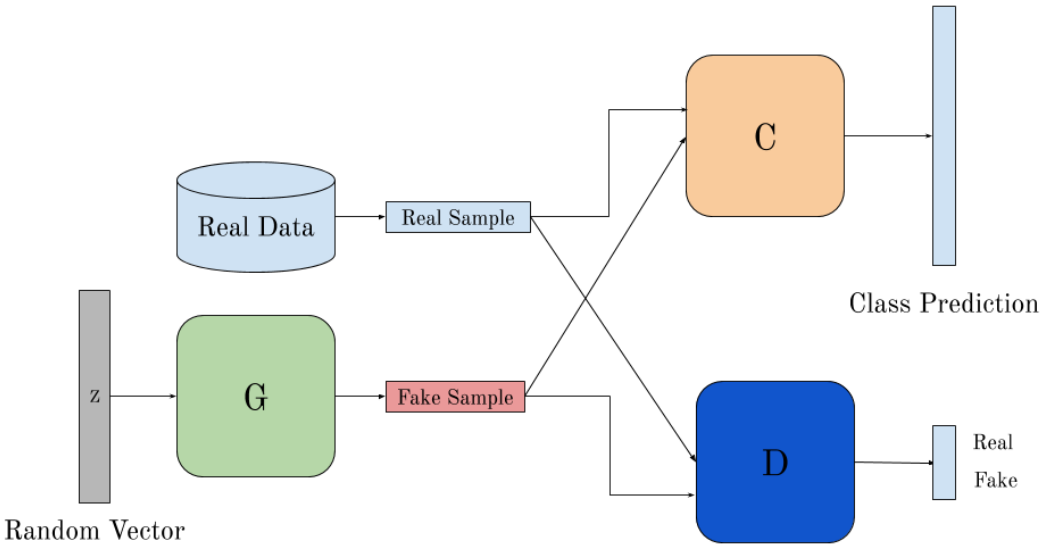
Također je važno napomenuti da kombinacija nadgledanog klasifikatora i nenadgledane GAN mreže čini EC-GAN metodom polu-nadgledanog učenja [2]. Takvim metodama nije pridodano jednako pažnje kao metodama nadgledanog i nenadgledanog učenja iako daju dobre rezultate.

4.1. Arhitektura EC-GAN modela

Na slici 3 nalazi se prikaz arhitekture EC-GAN algoritma te se mogu primijetiti ranije navedeni sastavni dijelovi: klasifikator i GAN (generator i diskriminator). U svakoj iteraciji treniranja generator dobiva vektor nasumičnih vrijednosti s normalnom distribucijom koje koristi kao sjeme za generiranje novih slika. Generirana slika se potom prosljeđuje diskriminatoru koji pokušava razaznati je li slika iz pravog skupa podataka ili je generirana od strane generatora.

Tako diskriminator uči razaznavati lažne od stvarnih slika.

U isto vrijeme klasifikator se trenira stvarnim podacima na standardan način. Zatim se klasifikator dodatno trenira s umjetno generiranim slikama kojima su pseudo-označavanjem dodijeljene oznake. Kako umjetno generirane slike nemaju oznake, a oznake su potrebne za treniranje klasifikatora, koristi se metoda pseudo-označavanja. To znači da se koristi sam klasifikator, u svom trenutnom stanju, kako bi se generiranim slikama dodijelio oznaku. Imajući na umu da klasifikator još nije treniran do kraja i da stoga ne daje najbolje rezultate, autor uvodi dodatne hiperparametre kako bi kontrolirao utjecaj pseudo-označenih podataka na djelotvornost modela.



Slika 3: Arhitektura EC-GAN modela iz izvornog rada [2]

Bitan dio implementacije klasifikatora EC-GAN metodom je njegova funkcija gubitka (eng. *loss function*). Autor predlaže intuitivnu novu funkciju gubitka koja implementira nove hiperparametre za kontroliranje utjecaja umjetnih podataka na treniranje modela. Predložena funkcija gubitka za model klasifikatora glasi:

$$L_C(x, y, z) = CE(C(x), y) + \lambda CE(C(G(z)), argmax(C(G(z))) > t)$$

Pri čemu je *x* stvaran podatak, *y* stvarna oznaka podatka, *z* umjetan podatak, λ je suparnička težina, *CE* funkcija unakrsne entropije, *C(x)* rezultat klasifikacije stvarnog podatka, *C(G(z))* rezultat klasifikacije umjetnog podatka i na kraju *t* predstavlja razinu pouzdanosti.

Vidimo kako je funkcija gubitka klasifikatora zapravo suma uobičajenih funkcija unakrsne entropije izračunatih za rezultate klasifikacije stvarnih i umjetnih podataka. Suparnička težina, predstavljena oznakom λ, je novi hiperparametar koji se koristi kako bi se umanjila težina (utjecaj) klasifikacije umjetnih podataka na rad modela. Bez ovog parametra, stvarni i umjetni podaci imali bi jednak utjecaj na krajnji rezultat što nije poželjno. Drugi predstavljeni hiperparametar je razina pouzdanosti *t*. Njome se određuje minimalna razina pouzdanost klasifikatora prilikom procesa pseudo-označavanja umjetnih podataka. Odnosno, ako klasifikator

nema dovoljno razinu pouzdanosti u to koja je oznaka umjetnog podatka, taj podataka se neće dalje koristiti za treniranje klasifikatora. Na ovaj način se izbjegava korištenje nisko kvalitetnih umjetnih podataka za treniranje modela.

Na kraju izvornog rada autor iznosi rezultate eksperimentiranja u kojima je pokazano da treniranje klasifikatora EC-GAN metodom nad progresivno manjim skupom podataka daje bolje rezultate klasifikatora treniranog standardnom metodom. Također, navedena je usporedba rezultata s drugim sličnim metodama od kojih EC-GAN pristup također daje bolje rezultate.

4.2. Kreiranje sintetičkih podataka

U izvornom radu, klasifikator treniran EC-GAN metodom koristi se za klasificiranje rendgenskih snimki (slika). Iz tog razloga korištene su specifične implementacije algoritama specijalizirane za rad sa slikama (ResNet18 i DCGAN). Upravo iz te činjenice proizlazi jedan od problema ovoga rada, naime skupovi podataka poput CIC-IDS-2017 su strukturirani tablični skupovi podataka. Algoritmi navedeni u izvornome radu nisu prikladni za rad s tabličnim podacima stoga ih je bilo potrebno supstituirati s algoritmima koji su za to specijalizirani.

Kreiranje sintetičkih tabličnih podataka još je relativno neistraženo područje uporabe GAN mreža te ne postoji mnogo primjera uspješne implementacije u te svrhe, pogotovo na području NIDS sustava. Međutim, nekolicina postojećih radova kojima je uspjelo pokazali su dobre rezultate. Jedan takav rad predstavili su Walia, Tierney i McKeever 2020. godine [22] u kojem su iskoristili WCGAN-GP kao metodu za kreiranje sintetičkih tabličnih podataka. Autori su usporedili tako kreirane skupove sa standardnim metodama generiranja podataka poput SMOTE metode. Rezultati pokazuju da su podaci generirani sa WCGAN-GP usporedivi sa stvarnim podacima za problem klasifikacije te da djeluju znatno bolje od podataka generiranih SMOTE tehnikom [22].

Zbog navedenih uspjeha odlučeno je da će WCGAN-GP implementacija biti korištena za kreiranje sintetičkih podataka u ovome radu¬. Alternative WGAN-u su bile implementacije TGAN [32] i CTGAN [33] koje se također koriste za kreiranje tabličnih podataka.

4.2.1. Uvjetovani Wasserstein GAN s gradijantnom kazom

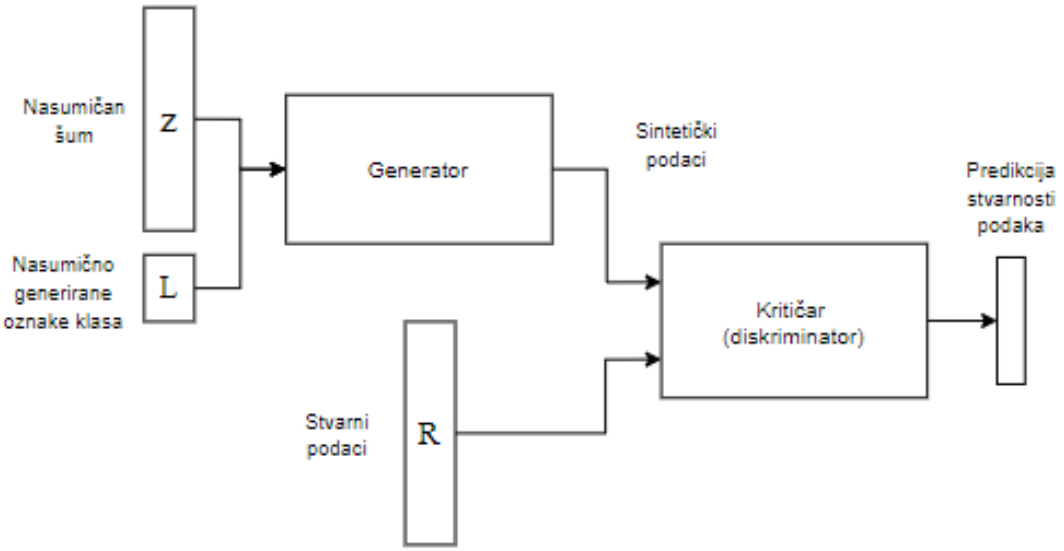
WCGAN-GP, odnosno uvjetovani Wasserstein GAN s gradijentnom kaznom, specijalna je implementacija GAN mreže koja, među ostalim, može biti korištena za kreiranje sintetičkih tabličnih podataka. Ovaj algoritam spada pod specijalnu vrstu GAN mreža zvanih uvjetovani GAN, odnosno CGAN. Njihova glavna karakteristika je da omogućuju ciljano generiranje klasa podataka, tj. CGAN generatoru je moguće narediti koju klasu podataka da generira iz nasumičnog vektora, umjesto da generira nasumičnu klasu [20].

Nadalje, Wasserstein u imenu ovog algoritma označava Wasserstein-ovu funkciju gubitka koja se koristi u ovoj implementaciji umjesto standardnih funkcija [12]. U izvornom radu, Arjovsky, Chintala i Bottu [12] navode kako se uporabom ovog algoritma postiže stabilnost treniranja i rješava se problem kolapsa mod-a (eng. *mode collapse*) što omogućuje lakše traženje

grešaka i optimalnih hiperparametara. Također, autori navode kako su daljnja istraživanja potrebna kako bi se pronašao optimalan način treniranja WGAN algoritma.

Nedugo nakon izvornog WGAN rada, izlazi rad [13] u kojem autori navode rješenje za optimalno treniranje WGAN algoritma. Autori predlažu korištenje gradijentne kazne (eng. *gradient penalty*) nad Wasserstein-skom funkcijom gubitka diskriminatora. Ovime su uspjeli postići još bolje rezultate i stabilnije treniranje WGAN algoritma. Pošto ova implementacija pokazuje do sada najbolje rezultate, ona će biti korištena dalje u radu.

Sama arhitektura ovoga algoritma može se vidjeti na slici 4. Bitno je napomenuti da se u literaturi vezanoj uz Wasserstein GAN, diskriminator mreža zapravo naziva „kritičar“ umjesto „diskriminator“, ali i dalje obavlja istu dužnost. Na slici 4 također se mogu vidjeti sastavni dijelovi WCGAN algoritma, koji su izgledom slični klasičnoj GAN implementaciji. Bitna razlika je dodatan ulaz u obliku oznake klase (eng. *label*) kod generatora, što čini ovaj algoritam CGAN-om. Daljnji slijed treniranja je standardan, generator koristi nasumičan šum i oznake za generiranje umjetnih podataka koji se potom daju kritičaru (diskriminatoru). Kritičar naizmjence dobiva umjetne i stvarne podatke te pokušava odrediti koji su koji. Sam kritičar je po prirodi binarni klasifikator te kao izlaz (eng. *output*) daje nula ili jedan, odnosno je li ulazni podatak stvaran ili umjetan.



Slika 4: Arhitektura WCGAN-GP modela implementiranog u ovome radu [autorski rad]

4.3. Klasifikacija mrežnog prometa

U ovom dijelu rada objašnjen je razlog korištenja duboke neuronske mreže za klasifikaciju mrežnog prometa. Ranije u radu dan je pregled mnoštva metoda, tehnika i algoritama koji se koriste za ovu svrhu te je većina njih mogla biti odabrana za treniranje EC-GAN metodom. Ta činjenica ostavlja mnogo prostora za buduća istraživanja na ovom području.

Postoji više razloga za odabir duboke neuronske mreže za treniranje EC-GAN metodom. Prvi od tih je razlog koji navode [5], [6], a to je da duboki modeli daju bolje rezultate od

klasičnih modela strojnog učenja. Drugi razlog je jednostavno implementacije i treniranja takvog modela, za razliku od drugih modela dubokog učenja poput autoenkodera ili GAN mreža, duboke neuronske mreže su relativno jednostavne za treniranje. Treći razlog tiče se rada spomenutog u 3. poglavlju [31] koji je pokazao izvrsne rezultate korištenjem duboke neuronske mreže.

Detalji oko same arhitekture i hiperparametara ovog modela dani su u sljedećem dijelu rada.

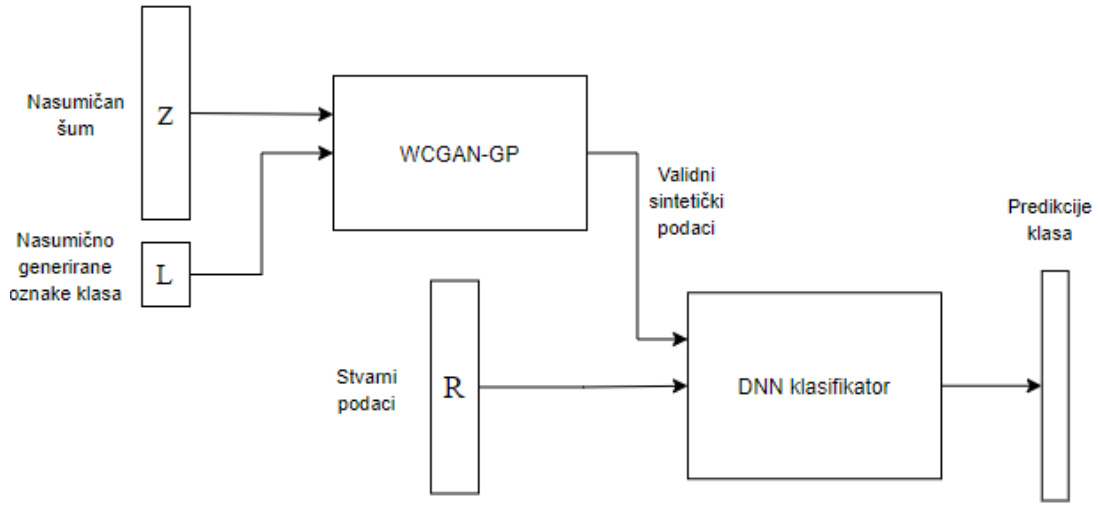
4.4. Opis rješenja

Ovaj dio rada sadrži detalje implementacije EC-GAN metode ovoga rada. Dan je pregled arhitektura sastavnih algoritama koji čine ovaj specifičan EC-GAN model te su objašnjeni koraci treniranja. U radu je do sada uglavnom bilo riječi općenito o korištenim metodama i tehnikama te o njihovoj teorijskoj pozadini. U nastavku je dan pregled arhitekture implementiranog modela kao i tablice u kojima su prikazani detalji implementacije svakog algoritma u smislu arhitekture i ostalih hiperparametara.

Na slici 5 se nalazi prikaz arhitekture EC-GAN modela koji je korišten za treniranje klasifikatora u ovome radu. Moguće je vidjeti do sada navedene metode za koje je spomenuto da su korištene u implementaciji rada. Bitno je napomenuti da su unutarnji dijelovi WCGAN-GP modela sa slike prikazani u ranijem poglavlju posvećenom tom algoritmu te da je jednak takav korišten i ovdje.

Sami koraci treniranja slični su kao i u općenitoj implementaciji EC-GAN metode iz četvrtog poglavlja uz jednu bitnu razliku. Kako se u ovoj implementaciji koristi uvjetovani GAN model (CGAN) za kreiranje sintetičkih podataka moguće je preskočiti korak pseudo-označavanja umjetnih podataka. Za generiranje umjetnih podataka WCGAN-GP algoritmom potrebno proslijediti oznake podataka koje će mreža generirati, iste te oznake mogu se kasnije koristiti za ažuriranje klasifikatora. Garanciju da te oznake korektno označavaju pripadni podatak nam daje mreža kritičara koja i tako mora propustiti generirani podatak kao „ispravan“ da bi on došao do klasifikatora za treniranje. Time se gubi potreba za korakom pseudo-označavanja umjetnih podataka – pošto su oznake već poznate.

Daljnji koraci treniranja isti su kao u općenitoj implementaciji EC-GAN metode. Generator generira sintetičke podatke, kritičar odlučuje jesu li oni stvarni, ako kritičar odluči da jesu to znači da je generator napravio dovoljno uvjerljive sintetičke podatke da zavara kritičara. Takvi validni umjetni podaci se zatim prosljeđuju klasifikatoru koji u jednoj iteraciji treniranja dobiva i umjetne i stvarne podatke. U svakoj iteraciji treniranja utezi (eng. *weights*) klasifikatora bivaju ažurirani temeljem ranije navedene EC-GAN funkcije gubitka klasifikatora.



Slika 5: Arhitektura EC-GAN modela implementiranog u ovome radu [autorski rad]

U tablici 2 nalaze se podaci o samoj implementaciji WCGAN-GP algoritma u ovom radu. Može se vidjeti da su generator i kritičar oboje duboke neuronske mreže s tri skrivena sloja. Kod generatora broj neurona skrivenih slojeva je progresivno sve veći, dok je kod diskriminatora progresivno sve manji kao što je to učinjeno u [22]. Zanimljiv je parametar *n_critic* koji označava koliko je puta kritičar (diskriminator) treniran u jednoj iteraciji treniranja. U revidiranom WGAN radu [13] autori navode kako algoritam daje bolje rezultate ako se kritičar trenira nekoliko puta više od samog generatora.

Tablica 2: Specifikacija arhitekture i hiperparametara za WCGAN-GP model [autorski rad]

Kritičar	Generator
Ulaz: dimenzija stvarnog skupa podataka	Ulaz: 32 (nasumičan šum)
1024, Leaky ReLU (alpha=0.2)	256, Leaky ReLU (alpha=0.2), Dropout(0.3)
512, Leaky ReLU (alpha=0.2)	512, Leaky ReLU (alpha=0.2), Dropout(0.3)
256, Leaky ReLU (alpha=0.2)	1024, Leaky ReLU (alpha=0.2), Dropout(0.3)
Izlaz - 1, bez aktivacijske funkcije	Izlaz - dimenzija stvarnih podataka, bez aktivacijske funkcije
Ostali hiperarametri:	
Stopa učenja: 0.0005	
Optimizator: Adam (beta_1=0.05, beta_2=0.9)	
Veličina jedne serije treniranja (eng.batch size): 128	
Broj epoha treniranja: 10	
n_critic parametar: 5	

U tablici 3 nalaze se detalji implementacije duboke neuronske mreže klasifikatora. Njezina arhitektura prilično je jednostavna, sastoji se od 3 skrivena sloja koji do početka do sredine

imaju sve veći broj neurona, a zatim on opet pada prema izlazu mreže. Za aktivaciju skrivenih slojeva korištena je *ReLU* funkcija, dok je za aktivaciju izlaza korištena *Softmax* funkcija.

Tablica 3: Specifikacija arhitekture i hiperparametara za DNN model [autorski rad]

Arhitektura
Ulaz: dimenzija stvarnog skupa podataka
128, ReLU, Dropout(0.3)
256, ReLU, Dropout(0.3)
128, ReLU, Dropout(0.3)
Izlaz: 15 - broj klasa stvarnog skupa podataka, Softmax
Optimizator: Adamax

Većina hiperparametara koji se nalaze u tablicama dobiveni su ekstenzivnim eksperimentiranjem te metodom pokušaja i promašaja dok se nije pronašla optimalna vrijednost.

Vrijednosti hiperparametara specifičnih za EC-GAN metodu, suparnički uteg i razina pouzdanja ostaju iste kao što su navedene u originalnom radu [2] (0.1 i 0.2 respektabilno). Prilikom implementacije testirane su različite vrijednosti ovih parametara no najbolje rezultate i dalje pokazuju originalne vrijednosti.

Zaključno s ovim dijelom završava dio rada koji je vezan uz teorijsku pozadinu te metode i tehnike korištene u ovom radu. U sljedećem poglavlju riječ je o praktičnom dijelu rada.

5. Praktični dio rada

Ovo poglavlje bavi se praktičnim dijelom rada. Poglavlje počinje s navođenjem korištenih tehnologija u implementaciji ovoga rada. Zatim su navedeni detalji oko korištenog skupa podataka, njegovog pred-procesiranja i pripreme za provođenje eksperimenta. Nakon toga slijedi dio koji opisuje proces treniranja klasifikatora EC-GAN metodom gdje je dan i pregled nekih zanimljivih dijelova programskog koda. Na kraju poglavlja navedeni su rezultati rada, odnosno usporedba s postojećim rezultatima u području kao i s modelom klasifikatora treniranim standardnom metodom. U nastavku slijedi popis korištenih tehnologija.

5.1. Korištene tehnologije

U svrhe implementacije praktičnog dijela ovoga rada korišteno je mnogo tehnologija iz Python ekosustava za razvoj modela umjetne inteligencije. Općenito cijela implementacija napisana je u jeziku Python 3.8 koristeći pretežito Keras biblioteku, a modeli su trenirani na grafičkoj kartici NVIDIA GeForce RTX 2070. Glavna implementacija EC-GAN modela napravljena je kao jedinstvena Jupyter bilježnica koja se može pronaći na Sljedećoj GitHub poveznici: https://github.com/marzekan/EC-GAN_NIDS/blob/main/ecgan/ecgan-nids.ipynb.

U nastavku slijedi popis korištenih tehnologija zajedno s kratkim opisom za što su korištene.

Python 3.8 (https://www.python.org/downloads/release/python-380/) – cijela implementacija napisana je u jeziku Python.

Tensorflow okvir (https://www.tensorflow.org/) – i njegov unutarnji okvir Keras korišteni su za izgradnju modela dubokog učenja.

Jupyter Notebook (https://jupyter.org/) – digitalne bilježnice korištene su kao razvojno okružje za razvoj, treniranje i testiranje modela.

CUDA (https://developer.nvidia.com/cuda-zone) – platforma pomoću koje su modeli trenirani na grafičkoj kartici.

Raznovrsni paketi iz Python ekosustava za umjetnu inteligenciju:

- Pandas (https://pandas.pydata.org/) – za analizu i obradu podataka.
- Numpy (https://numpy.org/) – za brzu obradu podataka.
- Scikit-learn (https://scikit-learn.org/stable/) – za evaluaciju uspješnosti klasifikatora.
- Matplotlib (https://matplotlib.org/) i Seaborn (https://seaborn.pydata.org/) – za kreiranje dijagrama i vizualizaciju.
- Tabular Evaluator (https://pypi.org/project/table-evaluator/) – za vizualnu evaluaciju sintetičkih tabličnih podataka.

5.2. Analiza skupa podataka

Ovo poglavlje bavi se preliminarnom analizom skupa podataka, gdje će ukratko biti objašnjeno podrijetlo i struktura CIC-IDS-2017 skupa podataka. Ovaj skup, nazvan prema Kanadskom institutu za Kibernetičku sigurnost (eng. *Canadian Institute for Cybersecurity*) na kojem je i razvijen za potrebe kreiranja jednog cjelovitog, modernog skupa podataka na području sustava za otkrivanje upada [19].

Skup podataka sadrži mrežni promet agregiran kroz nekoliko radnih dana tijekom kojih je simulirano 14 različitih napada. Skup sadrži i temeljnu neutralnu klasu naziva BENIGN koja predstavlja benigni, odnosno normalan promet tijekom kojeg se ne događa niti jedan napad. Zbrojeno napadi i benigni promet čine da ovaj skup podataka ima 15 različitih klasa. S obzirom na to da svaki redak sadrži i pripadnu klasu, odnosno naznačeno je kojoj klasi pripada, ovaj skup spada pod označene skupove podataka.

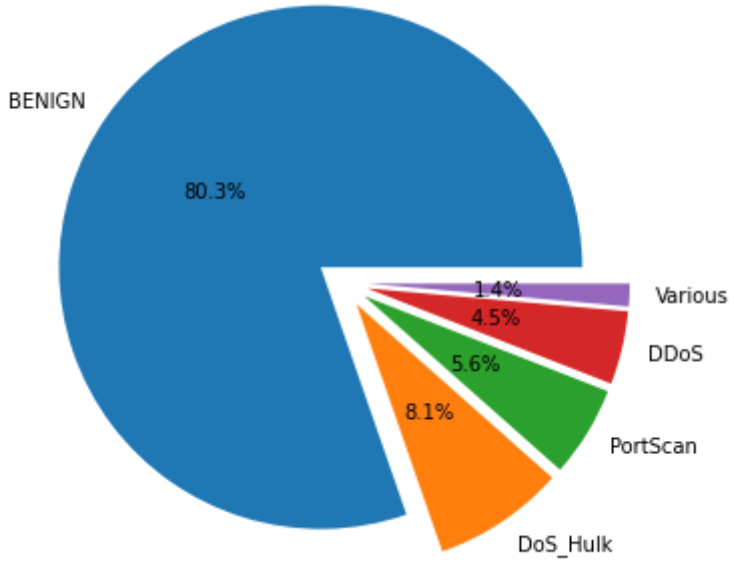
Prikaz članova svake klase, sortiran silazno, kao i relativan udio svake klase u cjelovitom skupu nalazi se u tablici 4.

Iz tablice 4 moguće je primijetiti da najveća klasa BENIGN čini čak 80% cijelog skupa podataka. Dok najmanja klasa Heartbleed čini manje od jedne tisućinke postotka cijelog skupa. Ovaj nesrazmjer u količinama članova (eng. *class imbalance*) klasa čini ovaj skup nebalansiranim. Činjenica da nekih klasa ima manje nego je minimalno potrebno da bi se trenirao bilo koji duboki model čini ovaj skup pogodnim za testiranje klasifikacije EC-GAN metodom.

Tablica 4: Prikaz broja članova klasa i njihovog udjela u CIC-IDS-2017 skupu podataka [autorski rad]

	Broj	Udio (%)
BENIGN	2271320	80.31
DoS_Hulk	230124	8.13
PortScan	158804	5.61
DDoS	128025	4.52
DoS_GoldenEye	10293	0.36
FTPPatator	7935	0.28
SSHPatator	5896	0.20
DoS_slowloris	5796	0.20
DoS_Slowhttptest	5499	0.19
Bot	1956	0.06
Web_Attack_Brute_Force	1507	0.05
Web_Attack_XSS	652	0.02
Infiltration	36	0.001
Web_Attack_Sql_Injection	21	0.0007
Heartbleed	11	0.0003

Na slici 6 nalazi se dijagram s relativnim veličinama klasa u CIC-IDS-2017 skupu podataka.



Slika 6: Dijagram udjela pojedine klase u skupu podataka CIC-IDS-2017 [autorski rad]

5.3. Priprema podataka za treniranje

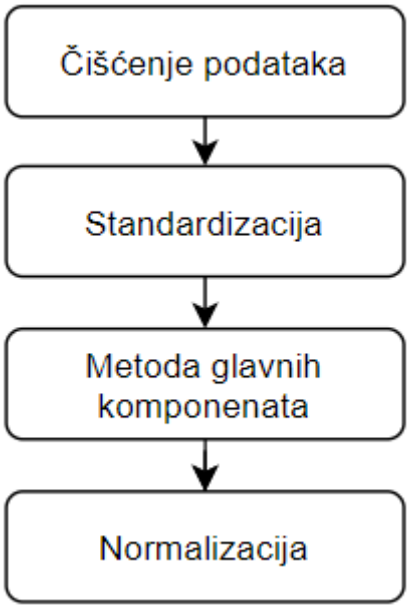
Ovaj dio rada tiče pripreme skupa podataka za treniranje. U nastavku je objašnjeno koji su koraci pred-procesiranja podataka obavljeni te kako su podaci dalje pripremljeni za eksperiment.

Pred-procesiranje podataka obuhvaća provodi se kroz metoda kojima se podaci pokušavaju činiti što pogodnijima za treniranje nekim algoritmom. Ovaj proces može čini samo takozvano čišćenje podataka, odnosno micanje NULL vrijednosti, brisanje redaka u kojima fale atributi te pretvaranje vrijednosti iz jednog tipa podatka u drugi. No češći je slučaj da je podatke potrebno dodatno obraditi nakon čišćenja nekom od čestih metoda: standardizacija, normalizacija, metoda glavnih komponenata (PCA) ili t-SNE. Navedene metode standardizacije i normalizacije mijenjaju distribuciju podataka u distribuciju koja je pogodna za treniranje neuronskim mrežama. Dok se metode poput PCA koriste za smanjivanje dimenzionalnosti podataka kako bi se smanjila kompleksnost treniranja uz uvjet da se ne promjeni značenje podataka.

Za potrebe treniranja modela u ovom radu skup podataka je temeljito procesiran. Proces odabira metoda za pred-procesiranje nije bio jednostavan. Bilo je potrebno napraviti mnogo iteracija procesiranja i uzastopnog treniranja modela takvim podacima da bi se odredilo koje metode daju najbolji rezultate. Nakon par desetaka pokušaja zaključeno je da model WCGAN-GP proizvodi najbolje rezultate kada je treniran podacima koji su prvo očišćeni, standardizirani, zatim im je reducirana dimenzija te su na kraju i normalizirani.

Slika 7 pokazuje slijed metoda kojima su podaci pred-procesirani za potrebe treniranja

WCGAN-GP algoritma, a potom i klasifikatora duboke neuronske mreže.



Slika 7: Ilustracija metode pred-procesiranja podataka kao priprema za treniranje modela [autorski rad]

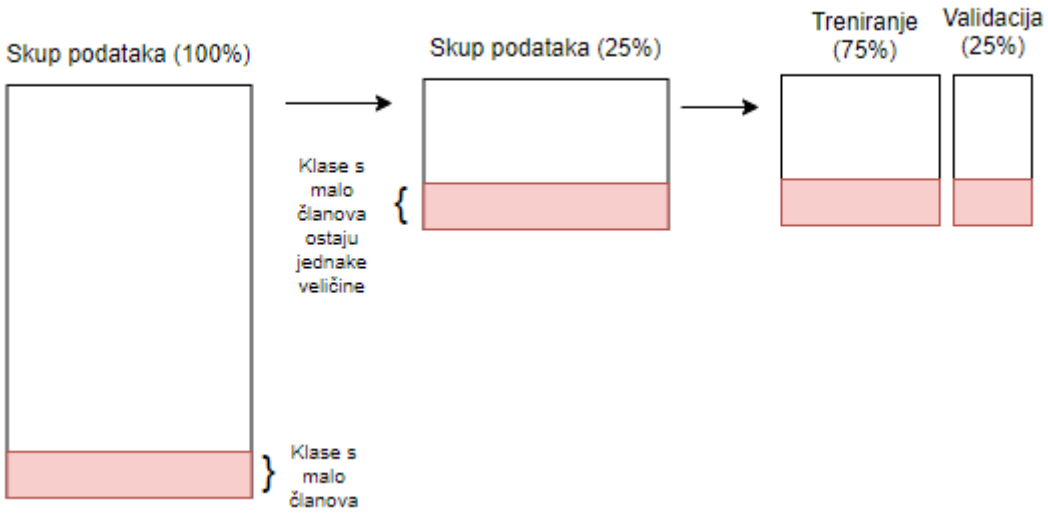
Sljedeći korak u pred-procesiranju podataka bio je kreirati nekoliko različitih skupova podataka za provođenje eksperimenta. Naime, bilo je potrebno napraviti progresivno sve manje skupove podataka kako bi se oponašali mali, realistični skupovi podataka iz stvarnog svijeta. Dodatan razlog smanjivanja skupova je kako bi se testirale granice EC-GAN metode, odnosno kako bi se pokušalo odrediti koliko je minimalno stvarnih podataka potrebno za postizanje dobrih rezultata.

Odlučeno je kako će se eksperiment s EC-GAN metodom provesti nad 10%, 25% i 50% CIC-IDS-2017 skupa podataka, dok će standardna metoda treniranja uključivati i cijeli originalni skup podataka. No kako je u prošlom poglavlju pokazano, skup podataka je veoma nebalansiran te neke klase samo desetak članova. Zbog toga je odlučeno da sve klase koje imaju manje od dvije tisuće članova neke biti srezane, nego će u svim iteracijama progresivno manjih skupova ostati jednake veličine. Sukladno tome, sve klase koje imaju više od dvije tisuće članova će biti srezane na postotak veličine koji odgovara toj iteraciji skupa za treniranje. Na primjer, u skupu koji čini 10% početnog skupa podataka, odabrano je 10% članova svake klase da čine taj skup, osim navedenih klasa s malo članova, koje će biti preuzete u svojoj cijelosti.

Za bolje objašnjenje navedenog razumijevanja, napravljena je ilustracija na slici 8.

Zadnji korak pred-procesiranja podataka bio je podijeliti skup podataka na skup za treniranje i skup za testiranje. Odlučeno je da će podaci biti odijeljeni u omjeru 75:25 pri čemu je 75% podataka rezervirano za treniranje.

Nakon zadnjeg koraka pred-procesiranja, podaci su spremni za treniranje modela, tj. za obavljanje eksperimenta.



Slika 8: Ilustracija metode razdvajanja podataka na skupove za treniranje i testiranje [autorski rad]

5.4. Treniranje modela

Ovaj dio rada bavi se implementacijom samog modela. U nastavku su pokazani bitni dijelovi implementacije poput implementacije inovativne funkcije gubitka EC-GAN metode, zatim poziva funkcije treniranja EC-GAN metode kao i funkcije treniranja klasifikatora standardnom metodom.

Modeli o kojima je riječ u ovom poglavlju već su nekoliko puta navedeni i opisani ranije u radu. Njihova arhitektura kao i vrijednosti hiperparametara mogu se pronaći u tablicama u četvrtom poglavlju. Modeli su izgrađeni korištenjem Tensorflow okvira u jeziku Python koristeći navedene tablice kao nacrt. Pregled cjelokupne praktične implementacije može se pronaći na Github repozitoriju ovoga rada (https://github.com/marzekan/EC-GAN_NIDS).

Prvi programski isječak koji će biti objašnjen vezan je uz poziv funkcije treniranja EC-GAN modela. Parametri koji se prosljeđuju ovoj funkciji su većinom hiperparametri WCGAN-GP ili samog EC-GAN modela. Osim njih kao parametar prosljeđuju se i pred-procesirani skup za treniranje podijeljen na sirovi skup podataka i njegove oznake (*x_train, y_train*). Sljedeći parametar nakon njih je broj klasa (*num_classes*) koji označava broj klasa označenog skupa podataka, u ovom slučaju to je broj napada + normalan promet u CIC-IDS-2017 skupu podataka.

Sljedeći parametar koji se prosljeđuje funkciji je veličina latentne dimenzije generatora, odnosno broj dimenzije vektora nasumičnog šuma kojeg generator koristi kao temelj za izradu umjetnih podataka. Ovaj hiperparametar kao i sljedeća dva (*batch_size* i *n_critic*) preuzeti su iz [13]. Zadnja dva hiperparametra vezana su uz EC-GAN metodu te se odnose na razinu pouzdanosti i suparničku težinu prilikom treniranja klasifikatora na umjetnim podacima.

```
gan = ECGAN(x_train,
            y_train,
            num_classes=15,
            latent_dim=32,
            batch_size=128,
            n_critic=5,
            conf_thresh=.2,
            adv_weight=.1
            )

gan.train(epochs=30)
```

Sljedeći programski isječak pokazuje implementaciju EC-GAN funkcije gubitka klasifikatora koju je autor predstavio u izvornom radu metode [2]. Parametar *y_true* predstavlja pravu klasu podatka dok parametar *y_pred* predstavlja klasu koju je model predvidio.

Druga linija koda ove funkcije je ključna, u njoj se odlučuje koja će predviđanja biti iskorištena za ažuriranje utega klasifikatora u sljedećoj liniji. Kao što je navedeno u izvorom radu [2] za treniranje se koriste samo predviđanja koja zadovoljavaju hiperparametar razine pouzdanosti. Odnosno, ako je pouzdanost klasifikatora u predviđanje neke klase preniska tada se to predviđanje neće koristiti za treniranje. Za kraj, treća linija koda ove funkcije implementira matematičku formulu funkcije gubitka za EC-GAN klasifikator. Moguće je vidjeti da se izračunati gubitak množi s drugim hiperparametrom EC-GAN metode – suparničkom težinom.

Bitno je napomenuti da se ova funkcija koristi samo za treniranje klasifikatora nad umjetnim podacima, dok se za treniranje nad stvarnima koristi uobičajena funkcija unakrsne entropije.

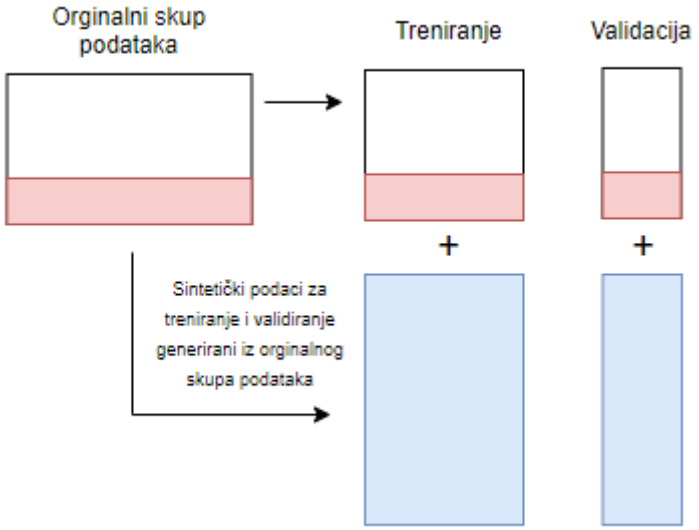
```
    def ecgan_loss(self, y_true, y_pred):
        max_values = tf.math.reduce_max(y_pred, axis=1)
        max_index = tf.where(tf.math.greater(max_values, self.conf_thresh))
        loss = self.adv_weight * self.cce_loss(y_true[max_index], y_pred[max_index])
        return loss
```

Ovime je završen pregled praktičnog dijela rada. U sljedećem dijelu rada predstavljeni su rezultati eksperimenata.

5.5. Rezultati rada

U ovom dijelu rada dan je pregled rezultata ranije navedenog eksperimenta. Dakle, klasifikator duboke neuronske mreže treniran je standardnom metodom, a zatim je njegova kopija trenirana korištenjem nove EC-GAN metode. Rezultati rada uspoređeni su s ranije navedenim radom Toupasa, Chamoua, Giannoutakisa, Dorsoua i Tzovarasa iz 2020. godine [31] koji su svoje rezultate predstavili kroz četiri metrike: točnost, stopa detekcije, stopa lažnih alarma i F1 mjera. Ove metrike su, među ostalima, standard za testiranje hipoteza u statistici i učestalo se koriste za evaluaciju klasifikatora u ovom području [34].

Sama evaluacija rada provedena je na skupu podataka za testiranje (eng. *testing set*) koji čini 25% cjelokupnog skupa za treniranje i testiranje modela. Bitno je za naglasiti da su



Slika 9: Ilustracija metode nadopunjavanja skupova za treniranje i testiranje sa sintetičkim podacima [autorski rad]

podaci za testiranje također nadopunjeni sintetičkim podacima, ali nakon samog treniranja. Razlog tome je isti kao i kod treniranja modela, broj članova nekih klasa jednostavno je premali za treniranje i za validaciju. Ako se uzme u obzir da neke klase imaju svega desetak članova od kojih se većina (75%) koristi za treniranje modela jasno je zašto je potrebno nadopuniti i skup za testiranje.

Za razliku od procesa treniranja, gdje su se sintetički podaci generirali paralelno s treniranjem klasifikatora, u procesu testiranja sintetički podaci se generiraju prije testiranja umjesto za vrijeme testiranja. Za njihovo kreiranje korišten je isti generator iz WCGAN-GP modela koji je korišten u treniranju modela. Za potrebe testiranja, kreirano je 10000 instanci svake klase osim klase BENIGN, jer ona i tako čini preko 80% skupa podataka. Kreirani podaci su zatim pomiješani sa stvarnim skupom za testiranje te je provedena evaluacija. Na slici 9 nalazi se ilustracija za bolje razumijevanje kako su korišteni sintetički podaci u evaluaciji modela.

U tablici 5 nalaze se rezultati navedene evaluacije klasifikatora kao i rezultati navedenog rada [31]. Jasno je kako je klasifikator treniran EC-GAN metodom pokazao bolje rezultate prema svim metrikama od klasifikatora treniranog standardnom metodom. Najbolje rezultate od modela treniranih u ovom radu pokazuje EC-GAN klasifikator koji je treniran na samo 25% originalnog skupa pravih podataka. Ova činjenica dokazuje korist EC-GAN metode za treniranje klasifikatora u slučajevima kada nije dostupno mnogo podataka. Najveću točnost od svih modela pokazuje model Toupas-a i suradnika koji je dostigao 99.95% točnosti. Njihovi rezultati pokazuju vrlo nisku razinu lažnih alarma koju je EC-GAN klasifikator također uspio dostići. Metrike u kojima EC-GAN klasifikator vodi su: F1 mjera i stopa detekcije.

Tablica 5: Rezultati evaluacije dva modela trenirana u ovome radu uspoređeni s rezultatima modela navrdenim u [31] [autorski rad]

	DNN treniran standardno				DNN treniran sa EC-GAN			DNN, prema [31]
(%) CIC-IDS-2017	10%	25%	50%	100%	10%	25%	50%	100%
Točnost (eng. accuracy)	.9771	.9806	.9844	.9852	.9897	.9895	.9876	.9995
Makro F1 mjera	.7266	.7247	.6351	.6289	.9924	.9995	.9901	.9410
Stopa lažnih pozitiva (FPR)	.0004	.0001	.0006	.0002	.0007	.0005	.0010	.0005
Stopa detekcije	.9255	.9414	.8985	.9218	.9933	.9897	.9852	.9562

6. Zaključak

Ovaj rad bavio se temom uporabe modela umjetne inteligencije u domeni sustava za detekciju upada na mreži. U samom uvodu dan je pregled problema na području NIDS sustava poput problema mnogo lažnih pozitiva i nemogućnosti sustava da uoče nove napade. Zatim je objašnjen i problem sa strane modela umjetne inteligencije, specifično klasifikacijskih algoritama koji zahtijevaju velike i dobro balansirane skupove podataka kako bi pokazali konkurentne rezultate. Navedeni problemi, odnosno želja za pronalaskom rješenja ujedno je i glavni cilj ovoga rada. Navedena je i motivacija autora za odabir ove teme koja ističe želju za uporabom inovativnih metoda na ovom području.

Nakon uvodnog dijela predstavljena je metodologija kojom će rad dobiti i predstaviti rezultate. Potom slijedi poglavlje o pregledu područja gdje su predstavljeni brojne metode, tehnike, algoritmi i modeli dubokog i strojnog učenja koji se koriste na ovom području. Kroz ovog poglavlje napravljen je i uvod u algoritme koji će se koristiti za postizanje rezultata ovoga rada. Sljedeće poglavlje bavi se inovativnom EC-GAN metodom koja se u radu koristi kao alternativna metoda treniranja nadgledanog klasifikatora. U poglavlju je sročen izvoran rad u kojem je metoda predstavljena i istaknuti su njegovi najvažniji dijelovi koji se direktno tiču ovog rada. U ovom poglavlju također su navedeni specifični algoritmi koji su implementirani u praktičnom dijelu rada WCGAN-GP i DNN klasifikator.

Zadnja velika cjelina ovog rada bavi se praktičnim dijelom rada. U ovom poglavlju dan je pregled CIC-IDS-2017 skupa podataka, opisan je proces pred-procesiranja podataka i njihove pripreme za provođenje eksperimenta. Zatim su pokazani neki zanimljivi dijelovi implementacije programskog koda te su isti i objašnjeni. Na kraju poglavlja prikazani su rezultati rada, odnosno prikazana je usporedba dva ista klasifikatora trenirana različitim metodama (EC-GAN i standardna) kao i njihova usporedba s uspješnim algoritmom iz ovog područja. Rezultati rada su uspješni i pokazuju da klasifikator treniran EC-GAN metodom po svojim rezultatima nadmašuje isti klasifikator treniran standardnom metodom.

Ovaj rad je među prvima koji koristi EC-GAN metodu za treniranje klasifikatora sintetičkim tabličnim podacima. Potrebno je napraviti još mnogo ispitivanja vezanih uz primjenu ove inovativne metode, kako bi se pokazala njena stvarna korist u ovom području. Imajući to na umu, rezultati ovoga rada su veoma uspješni i pokazuju da se korištenjem EC-GAN metode može trenirati nadgledani klasifikator čak i kada je izvorni skup podataka vrlo mali.

Popis literature

[1] N. Azeez, T. Bada, S. Misra, A. Adewumi, C. Vyver i R. Ahuja, „Intrusion Detection and Prevention Systems: An Updated Review," siječanj 2020., str. 685–696, Dostupno: ResearchGate, https://www.researchgate.net/. [pristupano: 06.09.2021.], ISBN: 978-981-32-9948-1. DOI: 10.1007/978-981-32-9949-8_48.

[2] A. Haque, „EC-GAN: Low-Sample Classification using Semi-Supervised Algorithms and GANs," *Proceedings of the AAAI Conference on Artificial Intelligence*, sv. 35, br. 18, str. 15 797–15 798, svibanj 2021., Dostupno: arXiv, https://arxiv.org/. [pristupano: 10.06.2021.] adresa: https://ojs.aaai.org/index.php/AAAI/article/view/17895.

[3] K. Kent, S. Chevalier, T. Grance i H. Dang, „Guide to Integrating Forensic Techniques into Incident Response," *NIST Special Publication*, siječanj 2006., Dostupno: ResearchGate, https://www.researchgate.net/. [pristupano: 06.09.2021.]

[4] B. Lutkevich, *Intrusion Detection System (IDS)*, en, Blog, veljača 2020. adresa: https://searchsecurity.techtarget.com/definition/intrusion-detection-system (pogledano 5. 9. 2021.).

[5] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah i F. Ahmad, „Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, sv. 32, br. 1, e4150, 2021., Dostupno: Wiley Online Library, https://onlinelibrary.wiley.com/. [pristupano: 13.06.2021.] DOI: https://doi.org/10.1002/ett.4150. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/ett.4150. adresa: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4150.

[6] H. Liu i B. Lang, „Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey," *Applied Sciences*, sv. 9, br. 20, 2019., Dostupno: MDPI, https://www.mdpi.com/. [pristupano: 13.06.2021.], ISSN: 2076-3417. DOI: 10.3390/app9204396. adresa: https://www.mdpi.com/2076-3417/9/20/4396.

[7] C. Chen, Y. Gong i Y. Tian, „Semi-supervised learning methods for network intrusion detection," Dostupno: ResearchGate, https://www.researchgate.net/. [pristupano: 07.09.2021.], studeni 2008., str. 2603–2608. DOI: 10.1109/ICSMC.2008.4811688.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville i Y. Bengio, „Generative Adversarial Networks," *Advances in Neural Information Processing Systems*, sv. 3, lipanj 2014., Dostupno: ResearchGate, https://www.researchgate.net/. [pristupano: 16.07.2021.] DOI: 10.1145/3422622.

[9] „Generative adversarial network: An overview of theory and applications," *International Journal of Information Management Data Insights*, sv. 1, br. 1, str. 100 004, 2021., Dostupno: Science Direct, https://www.sciencedirect.com/. [pristupano: 08.09.2021.], ISSN: 2667-0968. DOI: https://doi.org/10.1016/j.jjimei.2020.100004. adresa: https://www.sciencedirect.com/science/article/pii/S2667096820300045.

[10] J.-Y. Zhu, T. Park, P. Isola i A. A. Efros, „Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," *CoRR*, sv. abs/1703.10593, 2017., Dostupno: arXiv, https://arxiv.org/. [pristupano: 08.09.2021.] arXiv: 1703.10593. adresa: http://arxiv.org/abs/1703.10593.

[11] T. Karras, S. Laine i T. Aila, „A Style-Based Generator Architecture for Generative Adversarial Networks," *CoRR*, sv. abs/1812.04948, 2018., Dostupno: arXiv, https://arxiv.org/. [pristupano: 08.09.2021.] arXiv: 1812.04948. adresa: http://arxiv.org/abs/1812.04948.

[12] M. Arjovsky, S. Chintala i L. Bottou, *Wasserstein GAN*, Dostupno: arXiv, https://arxiv.org/. [pristupano: 16.07.2021.], 2017. arXiv: 1701.07875 [stat.ML].

[13] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin i A. C. Courville, „Improved Training of Wasserstein GANs," *CoRR*, sv. abs/1704.00028, 2017., Dostupno: arXiv, https://arxiv.org/. [pristupano: 16.07.2021.] arXiv: 1704.00028. adresa: http://arxiv.org/abs/1704.00028.

[14] D. Kermany, K. Zhang i M. Goldbaum, „Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification"," en, *Mendeley Data*, sv. V2, 2018., Dostupno: Mendeley Data, https://data.mendeley.com/datasets/. [pristupano: 08.09.2021.] DOI: 10.17632/rscbjbr9sj.2.

[15] K. He, X. Zhang, S. Ren i J. Sun, „Deep Residual Learning for Image Recognition," *CoRR*, sv. abs/1512.03385, 2015., Dostupno: arXiv, https://arxiv.org/. [pristupano: 08.09.2021.] arXiv: 1512.03385. adresa: http://arxiv.org/abs/1512.03385.

[16] A. Radford, L. Metz i S. Chintala, *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*, Dostupno: arXiv, https://arxiv.org/. [pristupano: 08.09.2021.], 2016. arXiv: 1511.06434 [cs.LG].

[17] X. Guo, Y. Yin, C. Dong, G. Yang i G. Zhou, „On the Class Imbalance Problem," *Fourth International Conference on Natural Computation, ICNC '08*, sv. Vol. 4, str. 192, listopad 2008., Dostupno: ResearchGate, https://www.researchgate.net/. [pristupano: 08.09.2021.] DOI: 10.1109/ICNC.2008.871.

[18] C.-Y. Ho, Y. Lin, Y.-C. Lai, I.-W. Chen, F.-Y. Wang i W.-H. Tai, „False Positives and Negatives from Real Traffic with Intrusion Detection/Prevention Systems," *International Journal of Future Computer and Communication*, sv. 1, str. 87–90, kolovoz 2012., Dostupno: ResearchGate, https://www.researchgate.net/. [pristupano: 09.09.2021.] DOI: 10.7763/IJFCC.2012.V1.23.

[19] I. Sharafaldin, A. H. Lashkari i A. A. Ghorbani, „Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization"," en, *4th International Conference on Information Systems Security and Privacy (ICISSP*, Portugal, siječanj 2018.

[20] M. Mirza i S. Osindero, „Conditional Generative Adversarial Nets," *CoRR*, sv. abs/1411.1784, 2014., Dostupno: IEEE Xplore, https://ieeexplore.ieee.org/. [pristupano: 13.07.2021.] arXiv: 1411.1784. adresa: http://arxiv.org/abs/1411.1784.

[21] Y. Yu, B. Tang, R. Lin, S. Han, T. Tang i M. Chen, „CWGAN: Conditional Wasserstein Generative Adversarial Nets for Fault Data Generation," *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dostupno: IEEE Xplore, https://ieeexplore.ieee.org/. [pristupano: 10.09.2021.], 2019., str. 2713–2718. DOI: 10.1109/ROBIO49542.2019.8961501.

[22] M. Walia, B. Tierney i S. Mckeever, „Synthesising Tabular Data using Wasserstein Conditional GANs with Gradient Penalty (WCGAN-GP)," Dostupno: ResearchGate, https://www.researchgate.n [pristupano: 16.07.2021.], prosinac 2020.

[23] D. Gomes i M. Neto, „Network Intrusion Detection Systems Design: A Machine Learning Approach," Dostupno: ResearchGate, https://www.researchgate.net/. [pristupano: 11.06.2021.], svibanj 2019. DOI: 10.5753/sbrc.2019.7413.

[24] M. Almseidin, M. Alzubi, S. Kovacs i M. Alkasassbeh, „Evaluation of machine learning algorithms for intrusion detection system," *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, Dostupno: IEEE Xplore, https://ieeexplore.ieee.org/. [pristupano: 13.06.2021.], 2017., str. 000 277–000 282. DOI: 10.1109/SISY.2017.8080566.

[25] E. Aminanto i K. Kim, „Deep Learning in Intrusion Detection System: An Overview," Dostupno: Semantic Scholar, https://semanticscholar.org/. [pristupano: 13.06.2021.], 2016.

[26] *CIC-IDS Datasets*, en, Scientifitc. adresa: https://www.unb.ca/cic/datasets/index.html (pogledano 10. 9. 2021.).

[27] M. Ring, S. Wunderlich, D. Scheuring, D. Landes i A. Hotho, „A Survey of Network-based Intrusion Detection Data Sets," *CoRR*, sv. abs/1903.02460, 2019., Dostupno: arXiv, https://arxiv.org/. [pristupano: 13.06.2021.] arXiv: 1903.02460. adresa: http://arxiv.org/abs/1903.02460.

[28] I. Dutta, B. Ghosh, A. Carlson, M. Totaro i M. Bayoumi, „Generative Adversarial Networks in Security: A Survey," Dostupno: ResearchGate, https://www.researchgate.net/. [pristupano: 11.09.2021.], listopad 2020. DOI: 10.1109/UEMCON51285.2020.9298135.

[29] J. Donahue, P. Krähenbühl i T. Darrell, *Adversarial Feature Learning*, Dostupno: arXiv, https://arxiv.org/. [pristupano: 11.09.2021.], 2017. arXiv: 1605.09782 [cs.LG].

[30] Z. Kamil, Y. Robiah, N. Bahaman, S. Mostafa i C. F. Mohd Foozy, „Benchmarking of Machine Learning for AnomalyBased Intrusion Detection Systems in the CICIDS2017 Dataset," *IEEE Access*, veljača 2021., Dostupno: ResearchGate, https://www.researchgate.net/. [pristupano: 11.09.2021.] DOI: 10.1109/ACCESS.2021.3056614.

[31] P. Toupas, D. Chamou, K. M. Giannoutakis, A. Drosou i D. Tzovaras, „An Intrusion Detection System for Multi-class Classification Based on Deep Neural Networks," *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019., str. 1253–1258. DOI: 10.1109/ICMLA.2019.00206.

[32] L. Xu i K. Veeramachaneni, „Synthesizing Tabular Data using Generative Adversarial Networks,” *CoRR*, sv. abs/1811.11264, 2018., Dostupno: arXiv, https://arxiv.org/. [pristupano: 23.06.2021.] arXiv: 1811.11264. adresa: http://arxiv.org/abs/1811.11264.

[33] L. Xu, M. Skoularidou, A. Cuesta-Infante i K. Veeramachaneni, „Modeling Tabular data using Conditional GAN,” *CoRR*, sv. abs/1907.00503, 2019., Dostupno: arXiv, https://arxiv.org/. [pristupano: 09.07.2021.] arXiv: 1907.00503. adresa: http://arxiv.org/abs/1907.00503.

[34] A. Tharwat, „Classification assessment methods,” *Applied Computing and Informatics*, sv. 17, br. 1, str. 168–192, siječanj 2021., Dostupno: Emerald Insight, https://www.emerald.com/insight. [pristupano: 09.09.2021.] DOI: 10.1016/j.aci.2018.08.003. adresa: https://doi.org/10.1016/j.aci.2018.08.003.

Popis slika

Popis tablica