Primjena umjetne inteligencije u području kibernetičke sigurnosti

Aleksić, David

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

David Aleksić

PRIMJENA UMJETNE INTELIGENCIJE U PODRUČJU KIBERNETIČKE SIGURNOSTI

DIPLOMSKI RAD

Zagreb, rujan 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI
POVJERENSTVO ZA DIPLOMSKI ISPIT

Zagreb, 23. svibnja 2023.

Zavod: **Zavod za informacijsko komunikacijski promet**
Predmet: **Sigurnost i zaštita informacijsko komunikacijskog sustava**

DIPLOMSKI ZADATAK br. 7322

Pristupnik: **David Aleksić (0135253806)**
Studij: Promet
Smjer: Informacijsko-komunikacijski promet

Zadatak: **Primjena umjetne inteligencije u području kibernetičke sigurnosti**

Opis zadatka:

U okviru diplomskog rada potrebno je analizirati dosadašnja istraživanja i mogućnosti primjene umjetne inteligencije u području kibernetičke sigurnosti. Potrebno je prikupiti podatke i formirati podatkovni skup kibrnetičkih napada i razviti model strojnog učenja za njihovu detekciju. Razvijeni model potrebno je validirati i ocijeniti njegove performase te izvesti zaključke.

Mentor:

dr. sc. Ivan Cvitić

Predsjednik povjerenstva za diplomski ispit:

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

PRIMJENA UMJETNE INTELIGENCIJE U PODRUČJU KIBERNETIČKE SIGURNOSTI

USING ARTIFICIAL INTELLIGENCE IN CYBER SECURITY

DIPLOMSKI RAD

Mentor: doc. dr. sc. Ivan Cvitić

Student: David Aleksić
JMBAG: 0135253806

Zagreb, rujan 2023.

PRIMJENA UMJETNE INTELIGENCIJE U PODRUČJU KIBERNETIČKE SIGURNOSTI

SAŽETAK:

Rad donosi pregled mogućnosti primjene metoda umjetne inteligencije u području kibernetičkih znanosti. Strojno učenje i duboko učenje su dvije najčešće korištene metode umjetne inteligencije za zaštitu sustava. Rad prikazuje kratki pregled algoritama i metoda strojnog i dubokog učenja nakon čega su izrađeni modeli strojnog učenja za prepoznavanje internetskih adresa iza kojih se kriju mogući *phishing* napadi.

KLJUČNE RIJEČI: Umjetna inteligencija, Strojno učenje, Duboko učenje, *Phishing* napadi

USING ARTIFICIAL INTELLIGENCE IN CYBER SECURITY

SUMMARY:

This thesis shows the possibilities of using methods of Deep Learning and Machine Learning in the field of Cybersecurity. Machine Learning and Deep Learning are two most commonly used methods of Artificial Intelligence that are used for Cybersecurity. Thesis gives short summary of types of algorithms and methods that are used for Machine Learning and Deep Learning after which the models based on Machine Learning for detection of phishing attacks are built.

KEY WORDS: Artificial Intelligence, Machine Learning, Deep Learning, Phishing attacks

Sadržaj

1. Uvod

Uz sve veći broj uređaja i međusobnu povezanost uređaja povećava se moguće područje napada na sustave. Kibernetička sigurnost u digitalnom dobu postaje sve važniji faktor i jedan od najtežih faktora za ostvariti s obzirom da se iz dana u dan povećavaju brojevi i vrste napada i napadača na sustave. Kibernetička sigurnost je skup postupaka, mjera i određenih standarda s kojima se nastoji održati pouzdanost tijekom korištenja proizvoda ili usluga u kibernetičkom području. U izvršavanju mjera, postupaka i standarda sve je češća upotreba umjetne inteligencije (engl. *Artificial Intelligence,* AI).

Umjetna inteligencija je područje znanosti koje omogućuje tehničkim sustavima prepoznavanje okruženja u kojemu se nalaze, prikupljanje podataka iz okoline te rješavanje zadanih problema [1]. Podvrste umjetne inteligencije koje se mogu koristiti kako bi se zaštitili suvremeni digitalni sustavi od sve češćih kibernetičkih prijetnji i napada su strojno učenje (engl. *Machine Learning,* ML) i duboko učenje (engl. *Deep Learning,* DL). Detekcija napada/prijetnji, detekcija upada te mogućnost obrade velike količine podataka su među najčešćim aktivnostima za koje se koristi umjetna inteligencija prilikom zaštite kibernetičkih sustava.

Svrha izrade ovog diplomskog rada je prikazati mogućnosti primjene metoda umjetne inteligencije i algoritama strojnog učenja u području kibernetičke sigurnosti tj. prikazati načine na koji se isti mogu koristiti za zaštitu od kibernetičkih napada kao što su *phishing* napadi. Cilj istraživanja je izraditi model strojnog učenja koji će na temelju dostupnih podatkovnih skupina podataka naučiti prepoznati kibernetičku prijetnju kao što je *phishing* napad, te na temelju izrađenog modela prikazati analizu dobivenih rezultata.

Ovaj diplomski rad podijeljen je u sedam cjelina:

1. Uvod
2. Dosadašnja istraživanja
3. Mogućnosti primjene umjetne inteligencije u kibernetičkoj sigurnosti
4. Prikupljanje i predobrada podataka
5. Primjena metoda strojnog učenja u kibernetičkoj sigurnosti
6. Analiza rezultata
7. Zaključak

Drugo poglavlje rada sadrži pregled dosadašnji istraživanja vezanih uz primjenu umjetne inteligencije u području kibernetičke sigurnosti. U prikazanim radovima dan je pregled algoritama strojnog učenja i dubokog učenja koji su korišteni u sustavima detekcije napada. Kroz pregled dosadašnjih radova definirani su pojmovi *phishing* napada, sistem za detekcije provale u sustave (engl. *Intrusion Detection System,* IDS) te ostale razne vrste napada na sustave.

Poglavlje „Mogućnosti primjene umjetne inteligencije u kibernetičkoj sigurnosti“ daje dodatan prikaz područja umjetne inteligencije koja se mogu koristiti u području kibernetičkih znanosti. Područja AI koja se mogu koristiti su: strojno učenje, duboko učenje te umjetne neuronske mreže (engl. *Artificial Neural Networks,* ANN). U ovome dijelu rada napravljena je podjela na najkorištenije algoritme strojnog učenja prema izvorima te je objašnjeno njihovo korištenje u području kibernetičke sigurnosti. Raspisana je i podjela na vrste umjetnih neuronskih mreža u kombinaciji sa algoritmima dubokog učenja te njihovo korištenje u kibernetičkoj sigurnosti. Za navedene algoritme strojnog učenja navedeni su konkretni primjeri korištenja u kibernetičkoj sigurnosti u sustavima kao što su IDS-ovi ili filteri za prepoznavanje *spam* poruka.

Četvrto poglavlje rada odnosi se na prikupljanje odnosno preuzimanje podatkovnog skupa podataka koji će se koristiti za izradu modela strojnog učenja za prepoznavanje *phishing* internetskih stranica. Poglavlje se sastoji od dijela u kojemu je opisan odabrani skup podataka. Ovdje se nalaze podaci o autorima, stranici sa koje je skup preuzet te osnovne informacije o samome skupu kao npr. raspodjela URL-ova (engl. *Uniform Resource Locator,* URL) na legitimne i nelegitimne itd.. Nakon osnovnih informacija o skupu napravljena je analiza programskog alata Weka koji je korišten za postupak predobrade podatkovnog skupa kako bi isti bio pogodan za korištenje sa pojedinim algoritmima strojnog učenja. Na kraju ovog poglavlja opisan je postupak učitavanja preuzetog podatkovnog skupa podataka i proces predobrade tog skupa u Weka programskom alatu.

U petome poglavlju prikazani su postupci izrade modela strojnog učenja za detekciju *phishing* internetskih adresa. Postupak izrade modela obavljen je putem Google-ove aplikacije unutar clouda koja se zove Google Colaboratory. Za izradu modela koristio se programski jezik Python i njegove knjižnice (engl. *library*). Korišteni algoritmi za izradu modela su: stablo odlučivanja (engl. *Decision Tree,* DT), algoritam logističke regresije (engl. *Logistic Regression*), algoritam potpornih vektora (engl. *Support Vector Machine,* SVM), slučajne

šume (engl. *Random Forest*, RF) i algoritam K-najbližih susjeda (engl. *K-nearest neighbors*, KNN)

Šesto poglavlje nastavlja se na prethodno poglavlje i odnosi se na analizu i interpretaciju rezultata dobivenih iz prijašnjeg poglavlja. U poglavlju je odrađena analiza pojedinačnih rezultata te međusobna usporedba rezultata prikazana vizualno sa grafovima.

Posljednje poglavlje je zaključno poglavlje u kojemu se nalazi kratki pregled rada te pregled dobivenih rezultata istraživanja i zaključci dobiveni provedbom analize dobivenih rezultata. Nakon zaključka u radu se nalazi popis literature te slika i tablica korištenih za izradu rada.

2. Dosadašnja istraživanja

Mogućnosti primjene algoritama strojnog učenja i umjetne inteligencije općenito u području kibernetičke sigurnosti su sve veće, što je rezultat tehnološkog napretka i značajnog povećanja resursa za prikupljanje i obradu podataka. Sukladno navedenome u kontinuiranom je porastu i broj istraživanja mogućnosti primjena istih u području kibernetičke sigurnosti.

Rad autora Dasgupta D., Akhtar Z. i Sen S. pod nazivom: „*Machine learning in cybersecurity: a comprehensive survey*“ iz 2022. godine daje uvid u događaje koji su se dogodili u području kibernetičke sigurnosti, a koji su povezani sa primjenom strojnog učenja. U radu su autori naveli vrste i podvrste kibernetičkih napada te načine obrane od istih.

U vrstu napada kompromitirajućeg korisničkog pristupa (engl. *User access compromise*) spadaju *phishing* napadi. Prema autorima *phishing* napadi smatraju se trikovima kojima maliciozni korisnici pokušavaju iskoristiti nepažnju ili neznanje korisnika te na taj način žele postići maliciozne radnje kao što su krađa korisničkih podataka ili preuzimanje malicioznog softvera na korisničko računalo.

U radu su navedeni načini obrane od raznih vrsta napada. Sistem detekcije provale u sustave jedan je od sistema koji se koristi kako bi se spriječili kibernetički napadi. Ovisno o namjeni i načinu detekcije IDS može se podijeliti na dvije metode. Prva metoda je metoda bazirana na detekciji (engl. *Detection-based method*) čije su prednosti brza detekcija zlouporabe i mala količina tzv. „lažnih uzbuna“ odnosno prijavljivanja provala u sustav za poznate napade, dok joj je mana što ima veliku količinu pogrešnih prijava provala u sustav za ranije nepoznate napade na sustave. Ova metoda radi na principu pronalaženja anomalija unutar mrežnog prometa. Druga metoda IDS-a koja se spominje u radu je metoda bazirana na izvoru podataka (engl. *Data source-based method*). Mana ove metode je što ona ne može prepoznati anomalije unutar mrežnog prometa dok joj je prednost što ovakav sustav IDS-a može precizno detektirati ponašanje koje se događa unutar programa, datoteka, itd. Kao jedan od načina obrane od *phishing* napada na korisnike autori ovoga istraživanja navode metodu filtriranja ulazne e-pošte na strani poslužitelja i korisnika.

Autori rada daju detaljan pregled rezultata istraživanja drugih autora za istraživanja slična njihovome koja su provedena u razdoblju od 2013. godine do 2018. godine. 2015. godine zadatak rada bio detekcija napada uskraćivanja usluge (engl. *Denial of Service,* DoS), detekcija napada „korisnik prema izvoru“ (engl. *User-to-Root,* U2R) te detekcija napada „udaljeni prema lokalnom“ (engl. *Remote-to-Local,* R2L) koristeći strojno učenje točnije algoritma stabla

odlučivanja. Sveukupni rezultat detekcije odnosno preciznost detekcije koja je postignuta u uočavanju navedenih napada iznosila je 92.62%. Od svih navedenih istraživanja najbolji rezultat ostvaren je 2017. godine sa preciznošću detekcije provale u sustav od 100%. Rezultat je postignut na privatnoj skupini podataka (engl. *dataset*) sa učilišta u Loughboroughu. Prilikom istraživanja je korišten algoritam/metoda strojnog učenja zvana metoda potpornih vektora.

Prema tablici prikazanoj u radu najbolje rezultate za detekciju napada ostvaruju istraživanja pri kojima je korišten SVM algoritam samostalno ili u kombinaciji sa algoritmom *decision tree.* U radu je napravljen kratki pregled setova podataka nad kojima su odrađena spomenuta istraživanja. Jedan od setova podataka nad kojima se provodilo istraživanje u detekcije napada je NSL-KDD dataset, koji je u sebi sadržavao tri podseta. Slika 1. prikazuje podjelu NSL-KDD skupa podataka i količinu podataka za pojedinu vrstu napada. Prvi set pod nazivom $KDDTrain^{+}$ služio je za treniranje modela i sastojao se od 125 973 podatka, preostala dva podseta su $KDDTest^{-21}$ i $KDDTest^{+}$ koji su se koristili za fazu testiranja [2].

	Total	Normal	DoS	Probe	R2L	U2R
$KDDTest^{+}$	22544	9711	7458	2421	2754	200
$KDDTest^{-21}$	11850	2152	4342	2402	2754	200
$KDDTrain^{+}$	125973	67343	45927	11656	995	52

Slika 1. Pregled skupa podataka NSL-KDD

Izvor: [2]

U zaključku rada navodi se kako je se korištenje algoritama strojnog učenja i umjetne inteligencije pokazalo učinkovito u okruženju kibernetičke sigurnosti te očuvanju integriteta podataka, ali da su i maliciozni korisnici otkrili mogućnosti korištenja ML i AI kako bi okrenuli stvari u svoju korist i naštetili korisničkim sustavima [2].

Rad pod nazivom „*A comparison study of machine learning techniques for phishing detection*“ autora: Kolla J., Praneeth S., Sameed Baig M. i Reddy Karri G. bazira se isključivo na *phishing* napade i metode strojnog učenja za detekciju napada. Analizirani su dobiveni rezultati detekcije pojedinog algoritma strojnog učenja korištenog za detektiranje *phishing* napada. Autori navode kako *phishing* napad spada u grupu napada socijalnog inženjeringa u kojima se korisnicima žele ukrasti osobni podaci kao što su zaporke, PIN i brojevi kreditnih kartica itd. Opisuje se postupak provedbe *phishing* napada putem e-pošte, poziva ili SMS poruke i način na koji se maliciozni korisnik skriva iza malicioznog linka. *Phishing* napadi su

najučestalija vrsta napada prema autorima, u radu se navodi kako je američka obavještajna agencija FBI (engl. *Federal Bureau of Investigation,* FBI) u razdoblju od 2013. do 2018. godine izgubila 2.3 milijarde dolara zbog *phishing* mailova tj. napada.

U ovome istraživanju autori su koristili skupine podataka za nadzirano strojno učenje. Dvije glavne podvrste nadziranog strojnog učenja su klasifikacija i regresija, skup podataka korišten prilikom istraživanja pogodniji je za klasifikacijsku metodu. Cilj korištenja strojnog učenja u ovome istraživanju je prepoznavanje phishing napada. Da bi to uspio model strojnog učenja mora klasificirati URL koji se nalazi u dolaznoj poruci kao lažni tj. phishing ili kao legitiman pravi link bez maliciozne funkcije na temelju odrađenog treninga.

Slika 2. prikazuje tablicu iz rada sa dobivenim rezultatima istraživanja. Iz tablice je vidljivo kako su najbolji rezultati ostvareni korištenjem algoritma slučajne šume. Ovo je algoritam koji se može koristiti za klasifikaciju i regresiju. Ovaj algoritam dao je najbolju efikasnost prepoznavanja napada u testnoj fazi sa postotkom preciznošću od 90%, dok je u fazi treniranja bio među slabijima sa 81,837%. Ostala četiri korištena algoritma imali su slične rezultate. SVM algoritam i algoritam višeslojnog perceptrona (engl. *Multilayer perceptron*) imali su preciznost od 86,667%, a algoritam stabla odlučivanja i XGBoost (engl. *Extreme Gradient Boosting*) dali su preciznost od 83,333% [3].

	ML Model	Train Accuracy	Test Accuracy
1	Random Forest	81.837	90.000
2	Multilayer Perceptrons	85.646	86.667
4	SVM	80.000	86.667
3	XGBoost	88.027	83.333
0	Decision Tree	81.701	83.333

Slika 2. Pregled rezultata preciznošću algoritama strojnog učenja

Izvor: [3]

Uz pregled rezultata u zaključku rada navedeno je kako su autori morali izraditi vlastitu skupinu podataka koja se sastojala od 9,375 URL-ova od čega su 5,653 bili legitimni stvarni linkovi bez maliciozne namjere i ostatak od 3,722 URL-ova su bili *phishing* URL-ovi [3].

Sljedeći rad „*Classification Methods of Machine Learning to Detect DDoS Attacks*“, autora Radivilova T., Kirichenko L., Ageiev D. te Bulakh V. prikazuje korištenje algoritama strojnog učenja za detekciju napada uskraćivanjem resursa i distribuiranih napada uskraćivanjem resursa (engl. *Distributed Denial-of-Service,* DDoS) Detekcija DDoS i DoS napada vrši se na način da se prilikom razmjene prometa unutar mreže vrši analiza istog prometa i cilj je detektirati anomalije u tom prometu. IDS se u mreži uglavnom nalazi iza vatrozida (engl. *Firewall*) koji sam analizira promet i propušta ga dalje kroz mrežu, a zatim dolazi do IDS-a. Detekcija anomalija u mrežnom prometu vrši se korištenjem algoritama strojnog učenja te statističkih metoda.

Za ovo istraživanje korišten je algoritam slučajne šume koji je dao najbolje rezultate u simulaciji i analizi koja je provedena prije samog istraživanja. Analizom dobivenih rezultata autori rada zaključuju kako je moguće primjenjivati algoritam strojnog učenja slučajne šume za detekciju DDoS napada samo u slučajevima kada prosječni udio prometa koji će se koristiti za napad iznosi 15-20% prosječnog sveukupnog prometa u mreži [4].

Rad „*Opportunities of Using Machine Learning Methods in Telecommunications and Industry 4.0 – A survey*“ autora: Peraković D., Periša M., Cvitić I., Zorić P., Kuljanić M.P. i Aleksić D. daje uvid u općenito korištenje umjetne inteligencije i strojnog učenja u području telekomunikacija. Neke od navedenih mogućnosti za korištenje AI i ML u telekomunikacijama su za prikupljanje podataka, interpretaciju prikupljenih podataka, kibernetička sigurnost itd.

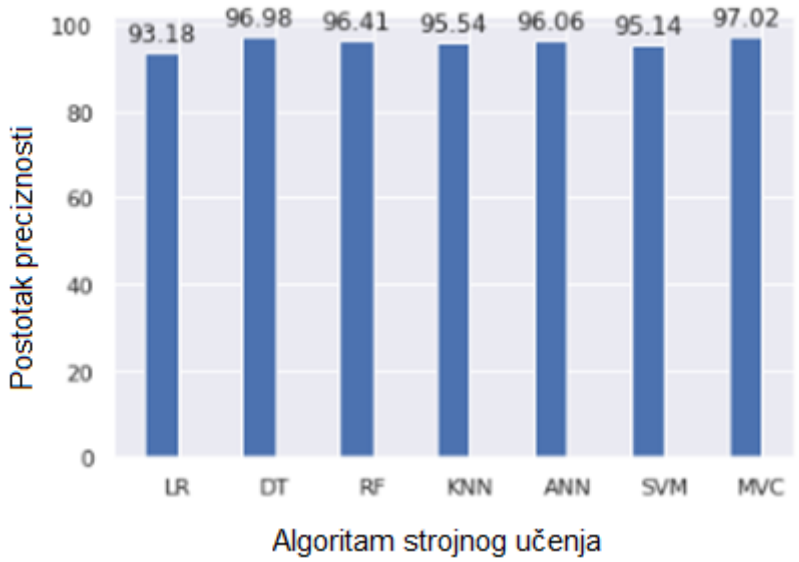
U radu se objašnjava razlika između tri vrste strojnog učenja: nadziranog, nenadziranog i ojačanog strojnog učenja. Za svaku vrstu navedene su i najpoznatije metode koje se koriste prilikom izrade modela strojnog učenja. U radu su navedeni programi tj. alati i njihove prednosti i nedostaci te općenite karakteristike alata koje se mogu koristiti za uporabu algoritama strojnog učenja prilikom izrade modela. Neki od navedenih alata su SciKit Learn, TensorFlow, PyTorch, Weka itd.

Rad sadrži primjere u kojima se strojno učenje koristi u području telekomunikacija kao na primjer u detekciji anomalija unutar mrežnog prometa te prepoznavanju stvarnog i malicioznog prometa kako bi se detektirao mogući DDoS napad na sustav. Još jedna od spomenutih mogućnosti korištenja algoritama ML je prilikom klasifikacije uređaja interneta stvari (engl. *Internet of Internet,* IoT). Primjenom regresijskog modela LogitBoost razvija se model strojnog učenja koji na temelju prikupljenih podataka i faze treninga može prepoznati o kojoj vrsti IoT uređaja je riječ na temelju vrste i količine podataka koja se prenosi od njega i prema njemu.

Moguća je primjena ML i u Industriji 4.0. Industrija 4.0 teži prema umrežavanju i samostalnom komuniciranju uređaja uz pomoć informacijsko komunikacijskih tehnologija. ML može se koristiti za predviđanje količine mrežnog prometa i u održavanju uređaja što je nužno i od velike prednosti ako je konačni cilj to da uređaji sami međusobno komuniciraju. Predviđanjem količine prometa može se doći do podataka koliko je uređaja i opreme potrebno da bi se ta količina prometa odvijala bez kašnjenja i zagušenja. Održavanje uređaja je neophodno kako bi se taj promet mogao odvijati konstantno jer ukoliko dođe do kvara jednog uređaja to može uzrokovati poteškoće za rad cijele mreže i cirkulaciju prometa kroz mrežu te je uz pomoć ML algoritama moguće pratiti stanje uređaja i na vrijeme reagirati ukoliko neki je neki od uređaja potrebno zamijeniti ili popraviti [5].

Autor Chawla A. u svom radu „*Phishing website analysis and detection using Machine Learning*“ provodi istraživanje sprječavanja *phishing* napada na sustave koristeći različite algoritme strojnog učenja. U radu navodi skup podataka od 11056 internetskih stranica sa po 30 parametara na temelju kojih će se validirati je li stranica stvarna ili je lažna. Neki od tih 30 parametara koji se koriste za analizu URL-a internetske stranice su [6]:

- Ima li stranica IP adresu (engl. *Having IP Address*) – parametar koji ukazuje je li internetska stranica registrirana na domenu ili ne.
- Duljina URL-a (engl. *URL Length*) – ukoliko je duljina URL-a kraća od 54 znakova stranica se može smatrati pouzdanom dok ako je raspon duljine znakova između 54 i 75 znakova stranica se smatra kao moguća prijetnja odnosno sumnjivom.
- Broj prosljeđivanja (engl. *Website Forwarding*) – ovaj parametar ukazuje na broj preusmjeravanja (engl. *redirecting*). Ukoliko je taj broj veći od jedan može se posumnjati na *phishing* URL.
- Onemogućen desni klik (engl. *Disabled Right Click*) – parametar koji pokazuje je li onemogućeno korištenje desnog klika. Ova mogućnost je od važnosti jer se uz korištenje desnog klika na mišu može koristiti mogućnost provjere stranice i provjera izvora koda te ako je ona onemogućena postoji velika mogućnost da ju je napadač onemogućio kako se ne bi mogle učiniti navedene radnje.
- Korištenje skočnih prozora (engl. *Using Pop-up Window*) – parametar koji govori traži li stranica popunjavanje podataka, informacija itd. u skočnim prozorima, ukoliko traži stranica se može klasificirati kao *phishing* stranica.
- Itd.



Slika 3. Usporedba rezultata preciznosti algoritama strojnog učenja u prepoznavanju URL phishing napada

Izvor: [6]

Autor je u radu koristio sedam različitih algoritama strojnog učenja. Oni su: algoritam logističke regresije, k-najbližih susjeda , potpornih vektora, stabla odlučivanja, slučajne šume, umjetnih neuralnih mreža i klasifikacija glasanja (engl. *Voting classifier*, MVC). Prema faktoru preciznosti prepoznavanja lažnog URL najbolje rezultate je dao algoritam klasifikacije glasanja sa 97,02%. Slika 3. prikazuje dobivene rezultate na temelju preciznosti prepoznavanja lažnih URL-ova.

3. Mogućnosti primjene umjetne inteligencije u kibernetičkoj sigurnosti

Umjetna inteligencija u području kibernetičke sigurnosti koristi se za detekciju napada i upada malicioznih kodova i korisnika u sustave te za prevenciju napada. Jedan od načina zaštita sustava primjenom umjetnom inteligencije može biti biometrijska zaštita kod koje uređaji korištenjem umjetne inteligencije daju ovlaštenje i pristup korisniku za korištenje sustava nakon što se uspješno obavi autentifikacijski postupak [7]. AI daje mogućnost korištenja naprednih alata i algoritama kako bi se spriječili napadi ili kako bi se njihove posljedice što više smanjile.

Umjetna inteligencija se sastoji od više područja znanosti koje zajedno spadaju u umjetnu inteligenciju. U kibernetičkoj sigurnosti najčešće se koriste strojno učenje, duboko učenje i umjetne neuronske mreže.

3.1. Kibernetička sigurnost

Kibernetička sigurnost je postupak koji za cilj ima zaštitu sustava, računalnih mreža i programa od kibernetičkih prijetnji i napada te neovlaštenog pristupa istima. Maliciozni korisnici imaju cilj upada u sustav te onesposobljavanje sustava i onemogućavanje kvalitetnog nastavka rada sustava, također još jedna maliciozna radnja koja im je u cilju jest i krađa raznih vrsta podataka koji se mogu nalaziti unutar sustava ili mreže. Cilj kibernetičke sigurnosti je svojim postupcima i mjerama smanjiti mogućnost ili u potpunosti ukloniti mogućnost malicioznim korisnicima ostvarenje njihovog cilja iako je nemoguće u potpunosti ukloniti tu mogućnost zbog konstantnog razvoja napada i tehnologija za maliciozne radnje sa sustavima [8].

Cilj zaštite računalnih sustava je zaštititi i očuvati tri osnovna načela sigurnosti [9]:

- Povjerljivost – načelo sigurnosti koje osigurava otkrivanje podataka i informacija samo i isključivo osobama koji su autorizirani
- Cjelovitost – načelo sigurnosti koje osigurava zaštitu od slučajnih ili namjernih promjena tj. modifikacija informacija koje mogu biti uzrokovane ljuskom ili pogreškom rada sustava
- Dostupnost – načelo sigurnosti koje osigurava da će informacije biti uvijek biti raspoložive namijenjenim korisnicima u traženom trenutku prema zadanim uvjetima.

Phishing napadi su web temeljeni napadi u kojima maliciozni korisnici na prevaru preusmjeravaju korisnika koji je meta napada na malicioznu internetsku stranicu. Na toj stranici žele iskoristiti neznanje korisnika te ih traže unos osobnih podataka kao što su npr. ime, prezime, datum rođenja, broj mobitela, PIN, lozinke itd. te na taj način vrše krađu podataka. Prema [10] dnevno u svijetu se pošalje otprilike 3.4 milijarde *spam* i *phishing* e-pošte, a u 2023. godini detektirano je 1.35 miliona *phishing* internetskih stranica dok je na kraju 2020. godine broj tih stranica iznosio 146 994 prema podacima sa izvora [11]. Zbog čega su *phishing* napadi među najučestalijim napadima na računalne sustave. *Phishing* napadi već su dugo poznati u svijetu kibernetičke sigurnosti te je već poznato kakav je oblik te kako izgleda *phishing* napad u bilo kojem obliku (SMS, e-pošta, poziv). *Phishing* napad koji dolazi putem e-pošte najlakše je prepoznati putem adrese pošiljatelja, s obzirom da se maliciozni korisnici u velikoj većini slučajeva pretvaraju da se radi o velikim tvrtkama, poduzećima. U njihovim adresama se uglavnom nalazi pogreška koja je namjerno napravljena u nadi kako meta napada neće primijetiti tu pogrešku. Na primjer umjesto „@microsoft.com“ maliciozni korisnik će napraviti adresu u kojoj će zamijeniti raspored dva slova pa bi ta adresa izgledala „@mircosoft.com“. Zatim još jedna česta karakterističnost je da pošta koja je poslana navodno od strane tvrtke kao što je i dalje na primjer Microsoft u adresi pošiljatelja uopće ne sadrži domenu tvrtke već je poslano sa javne domene pružatelja usluga elektroničke pošte kao što je Gmail. Te još jedna mogućnost prepoznavanja *phishing* napada je u tijelu poruke. Sadržaj tih poruka je uvijek napisan na isti način i korisniku javlja kako je osvojio nagradu te da klikom na poveznicu upiše svoje podatke kako bi mu se isporučila nagrada ili je sadržaj poruke vezan uz nekakav oblik plaćanja neke usluge ili proizvoda. Dakle sve poruke koje zahtijevaju od korisnika prelazak na drugu internetsku stranicu i unos osobnih podataka, a da ispunjavaju i prethodan dva uvjeta sa pogreškama unutar adresa pošiljatelja mogu se smatrati *phishing* napadima [12].

Pojavom umjetne inteligencije znatno se olakšava prepoznavanje i održavanje kibernetičke sigurnosti. Umjetna inteligencija je dovoljno razvijena kako bi olakšala rad sa velikim podatkovnim skupinama i na taj način znatno skratila resurs vremena koji je u nekim slučajevima od velike važnosti. Prednost umjetne inteligencije je u tome što s vremenom razvija i umjetna inteligencija koja može učiti. Također prednost je što se smanjuje čovjekov obujam posla jer velika većina onoga što je do nedavno čovjek morao samostalno obavljati u današnje vrijeme može obavljati umjetna inteligencija, a čovjek je taj koji će je nadzirati i kontrolirati u radu. Umjetna inteligencija i strojno učenje mogu se koristiti kod detekcije razne

vrste napada u sustave kao samostalni sustavi ili kao dijelovi nekog sustava za detekciju te na taj način također dopridonose održavanju kibernetičke sigurnosti.

3.2. Strojno učenje u području kibernetičke sigurnosti

Strojno učenje je područje znanosti umjetne inteligencije koje za cilj ima naučiti sustave načinu razmišljanja što sličnijem čovjeku u svrhu rješavanja postavljenih problema i zadaća [7]. Osnovna podjela strojnog učenja je na: nadzirano strojno učenje (engl. *Supervised Machine Learning*), nenadzirano strojno učenje (engl. *Unsupervised Machine Learning*) i ojačano strojno učenje (engl. *Reinforcement Machine Learning*). Svako od podvrsta strojnog učenja ima dodatnu podjelu na algoritme. Najčešće korišteni algoritmi strojnog učenja prema [13] su:

- Algoritam stabla odlučivanja – algoritam nadziranog strojnog učenja
- Skriveni Markovljev model (engl. *Hidden Markov model,* HMM) – statistički model koji se koristi prilikom strojnog učenja, prvotno je bio sastavni dio područja znanosti umjetne inteligencije koja je proučavala prepoznavanja govora
- K-srednjih vrijednosti algoritam klasteriziranja (engl. *K-means clustering*) – spada pod najpopularnije i jednostavnije algoritme nenadziranog strojnog učenja koji se mogu koristiti za kibernetičke probleme

Algoritam stabla odlučivanja često je prvi izbor prilikom odabira algoritma ukoliko je potrebno izvršiti detekciju upada ili detekciju *spam* poruka. Razlog odabira DT algoritma je u tome što ovaj algoritam ima dobre mogućnosti u identifikaciji pravila i uzoraka unutar mrežnog prometa i korištenju sustava te mu je lako uočiti anomalije koje se događaju unutar sustava i mrežnog prometa. Algoritam DT pogodan je iz razloga što je izrada modela uz njegovo korištenje jeftina i brza, a interpretacija dobivenih rezultata je laka za malu količinu podatkovnih skupova [13].

Metoda potpornih vektora ja algoritam strojnog učenja koji se koristi za metode klasifikacije i regresije. U području kibernetičke sigurnosti pogodan je za detektiranje anomalija unutar mrežnog prometa i detekciju uzoraka. Detekcija uzoraka je postupak koji se izvodi za napade kao što su phishing napadi koji imaju određeni uzorak koji se ponavlja u svakome od pokušaja napada na temelju čega se može zaključiti da je riječ o ovoj vrsti napada na sustav. Za razliku od algoritma DT rezultati se teže analiziraju jer je teže zaključiti na kojem principu je algoritam SVM došao do zaključaka i rezultata [13].

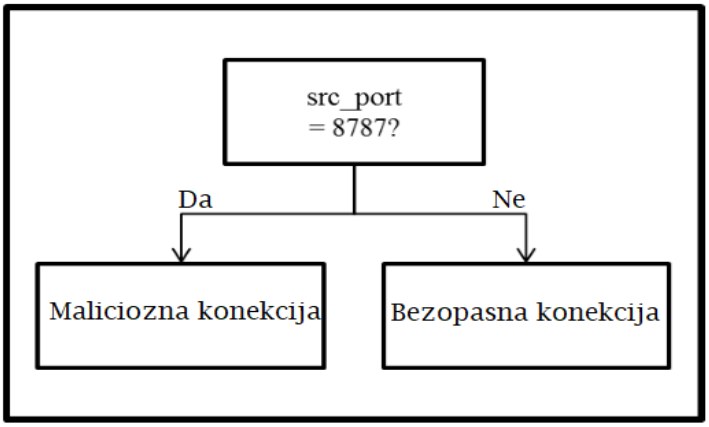
K-srednjih vrijednosti algoritam je algoritam nenadziranog strojnog učenja koji radi na jednostavnom principu u kojem podatke svrstava u skupine na temelju njihove sličnosti [13].

Prednosti k-srednjih vrijednosti algoritma osim jednostavnog principa rada je i mogućnost rada sa velikim podatkovnim skupinama što je čak i preporučljivo kako bi se dobio veći broj podataka iz kojih se mogu detaljnije odrediti sličnosti skupina podataka [14]. Ovaj algoritam u području kibernetičke sigurnosti koristi se za izradu modela IDS-ova, detekciju zloćudnih softvera (engl. *malware*) i filtriranje e-pošte odnosno *spam* poruka i stvarnih poruka. Također, može se koristiti za analiziranje log zapisa sa *proxy* servera i Captive portala u kojoj se prikupljeni podaci sa servera klasteriziraju (razvrstavaju u pripadajuće skupine podataka) korištenjem k-srednjih vrijednosti algoritma kako bi se analizirali trendovi korisnika uz pomoć kojih se zatim mogu detektirati anomalije te otkriti potencijalni maliciozni korisnici stranice [15].

Skriveni Markovljev algoritam češće se koristi za statističko predviđanje uzoraka. HMM je pouzdan alat za prepoznavanje slabih signala. Mana mu je što se podatkovni skup podataka nad kojima se provodi faza treniranja modela mora iznimno dobro definirati i jasno predstavljati problem te ti podaci moraju biti visokokvalitetni kako bi proces izrade modela bio kvalitetniji. U kibernetičkoj sigurnosti HMM algoritam se može koristiti za više mogućnosti zaštite, a ponajviše za detekciju upada [13].

3.2.1. Primjena algoritma stabla odlučivanja za sustave detekcije upada

Algoritam stabla odlučivanja koristi se prilikom postupka analiziranja mrežnog prometa u sistemima detekcije upada u sustave. DT algoritam definira se kao tehnologija prediktivnog modeliranja koja nastaje iz znanosti statistike i strojnog učenja na temelju kojih se izrađuje model strukture koja nalikuje stablu zbog kojega se ovaj algoritam zove algoritam stabla odlučivanja [16].



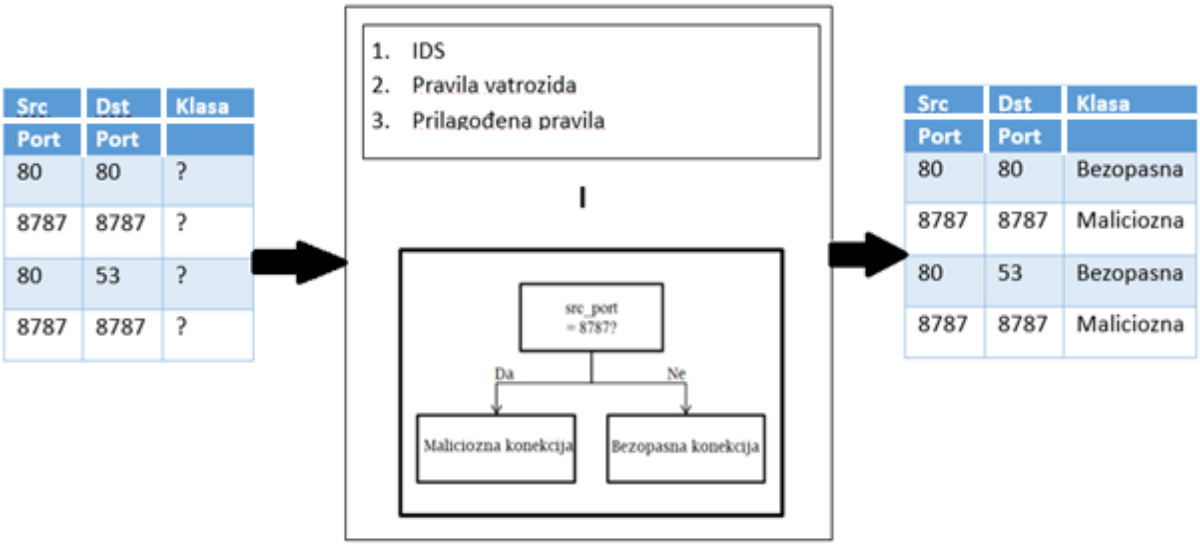
Slika 4. Primjer jednostavnog stabla odlučivanja

Izvor: [16]

Slika 4. prikazuje jednostavan primjer stabla odlučivanja u kojem se stablo odlučivanja temelji na odlučivanju o vrsti konekcije za port 8787 za koji je poznato da je riječ o malicioznoj konekciji.

Prema [17] DT algoritmi i ostali klasifikacijski algoritmi su testirani i korišteni u realnim okruženjima i izazovima kao dijelovi IDS-ova. Klasifikatore stabla odlučivanja velika većina istraživača i modelara napravila je pomoću tzv. *honeypots-a.* To su sustavi koji služe mamci unutar mreže kako bi se privukli i namamili maliciozni korisnici. Ti sustavi zajedno sa podacima iz normalne mrežne aktivnosti koriste se kako bi se razvio model klasifikatora algoritma stabla odlučivanja. Putem analize prikupljenih podataka stvara se lista pravila prema kojima se utvrđuje je li riječ o anomaliji unutar mreže tj. malicioznoj aktivnosti. Isto tako za prikupljanje podataka mogu se koristiti podaci prikupljeni prilikom penetracijskih testiranja sustava na način da se analiziraju postupci koje je tester izvodio nad sustavom i mrežom [17].

Prednost korištenja algoritam stabla odlučivanja je u tome što je to direktan proces koji se može odraditi u roku nekoliko sati ili mjeseci ovisno o tome na kolikoj skupini podataka se vrši proces izrade modela te koliko detaljno se želi istražiti i saznati o vrsti upada u sustav. Nakon procesa prikupljanja podataka, prikupljeni podaci prolaze kroz proces pred procesiranja u kojemu se obrađuju i pretvaraju u oblik koji je potreban kako bi se mogli koristiti u procesu izrade modela korištenjem DT algoritma. Sljedeći korak nakon predobrade podataka je proces treniranja modela, a zatim posljednji koraka analiziranje rezultata dobivenog modela u stvarnom okruženju [16].



Slika 5. Princip razmišljanja klasifikatora

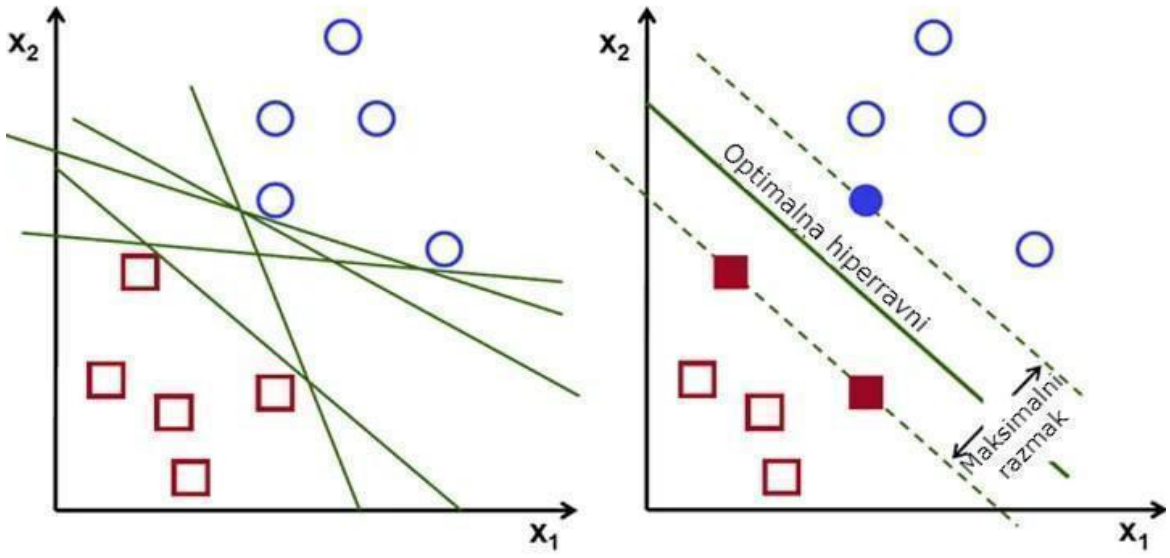
Izvor: [16]

Slika 5. prikazuje način na koji se klasificira promet uz korištenje DT algoritma. Na slici lijevo nalazi se tablica sa dolaznim novim mrežnim prometom i popisom portova za koje se ne zna kojoj klasi prometa pripada promet koji dolazi sa tih portova. Prometu koji dolazi može se vjerovati ili ne ovisno od izvora s kojeg dolazi. Dolazni promet prolazi kroz klasifikatorsku logiku u kojoj prolazi kroz pravila na vrhu kojih su pravila IDS sustava, zatim pravila unutar *firewall-a* i na kraju popis prilagođenih pravila ovisno o krajnjem cilju modela. Uz pravila prolazi i kroz stablo odlučivanja koje je napravljeno za npr. poznate portove kojima se može vjerovati ili ne može vjerovati. Na slici desno nalazi se tablica sa istim portovima nakon prolaska kroz klasifikator gdje se dobiva tablica sa klasificiranim mrežnim prometom i podacima i zna se točno koji promet maliciozan, a koji bezopasan i na temelju toga se može spriječiti upad u sustave i može se izvršiti detekcija upada u sustave [16].

3.2.2. Primjena SVM algoritma za sustave detekcije upada

Cilj SVM algoritma strojnog učenja jest pronaći optimalne razdvajajuće hiperravnine (engl. *Hyperplane*) koji će im omogućiti maksimalizaciju korištenja količine podataka u fazi treninga te će minimalizirati kompleksnost i rizik od *overfitting*-a [18]. Hiperravnina su granice odlučivanja koje se koriste kako bi se lakše klasificirali podaci primjenom SVM algoritma [7]. *Overfitting* je nepoželjni događaj koji se javlja u području strojnog učenja kada izrađeni model daje točne pretpostavke/odgovore za podatke sa kojima su odrađeni postupci treninga i evaluacije dok to nije slučaj za nove podatke s kojima model radi prilikom postavljanja u

stvarno vremensko okruženje [19]. Primjena SVM algoritma pogodna je za rad sa malim skupovima podataka za fazu treninga i pogodan je za analizu ekstremno velikih skupova podataka. Slika 6. prikazuje klasifikaciju i odabir optimalne hiperravnine koja se kao granica nalazi na jednakoj udaljenosti između dvije različite klase podataka. Slika prikazuje maksimalni razmak između klasificiranih skupina podataka unutar kojeg nema niti jednog podatka sa niti jedne strane skupine.



Slika 6. SVM algoritam

Izvor: [20]

Prilikom primjene SVM algoritma za izradu IDS sustava najprije se prikupljaju podaci za izradu modela. Nakon odabranog skupa podataka za izradu modela prvi postupak SVM klasifikacijskog procesa je inicijalna faza treninga i klasifikacijska faza. U ovim fazama bitno je znati da je SVM algoritam vrsta nadziranog strojnog učenja za koje je bitno poznavati skupove podataka, kako bi faze treninga bile učinkovite. To jest, bitno je znati točno koji podaci u skupu podataka predstavljaju maliciozne, a koje bezopasne podatke kako bi se model mogao ispravno naučiti. Također prije nego li se započne sa izradom modela odabrani skup podataka mora proći kroz proces pred obrade kako bi se podaci optimizirali za SVM algoritam [21].

Primjer procesa pred obrade skupine podataka sastoji se od postupaka [21]:

- Dodjeljivanja numeričke vrijednosti slikovnim značajkama
- Implementacija skalarne vrijednost s obzirom da podaci unutar podatkovnih skupova mogu varirati, zato se određuje skala za atribute u rasponu od -1 do 1

- Nazivi napada dodjeljuju se u jednu od dvije grupe, 0 je za podatke koji nisu maliciozni, a 1 za podatke koji u sebi sadrže određenu vrstu napada.

SVM je prema [21] jedan od najboljih algoritama strojnog učenja za detekciju anomalija i prepoznavanje neovlaštenih upada u mrežni promet. Za razliku od DT algoritma koji se isto koristi za IDS sustave SVM algoritam je dosta zahtjevniji, kompliciraniji i zahtjeva više vremenskog resursa za izradu [21].

3.2.3. K-srednjih vrijednosti za detekciju *spam* i *phishing* pošte

Phishing e-pošta svrstava se u *spam* kategoriju elektroničke pošte. *Spam* e-pošta je oblik neželjenih poruka koje korisnici dobivaju najčešće od raznih brendova, dućana i ostalih pošiljatelja, a u sebi sadrže propagandne sadržaje u obliku letaka za promociju proizvoda i usluga. Premda da svaki korisnik dnevno dobije nekolicinu *spam* poruka koriste se klasifikatori koji tu poštu automatski kategoriziraju u jednu od kategorija unutar korištene aplikacije ili platforme za elektroničku poštu [22]. Zbog velike količine dolaznih promotivnih poruka koje korisnik dobiva na dnevnoj razini, maliciozni korisnici *phishing* napade maskiraju kako bi njihova pošta bila što sličnija bezopasnim *spam* mailovima. Slika 7. prikazuje ovakav način *phishing* poruke koja korisnika traži ponovni unos i pokušaj plaćanja mjesečne pretplate za platformu Netflix jer je prijašnja uplata nije odobrena. U zaglavlju pristigle poruke i na vrhu slike vidljivo je kako adresa pošiljatelja nije legitimna adresa već je riječ o *phishing* poruci koja ima namjeru krađu korisničke kreditne kartice ili ostalih osobnih podataka.

Slika 7. Prikaz phishing poruke

Izvor: [23]

U postupku detekcije tj. izrade filtera za prepoznavanje *spam* pošte k-srednjih vrijednosti algoritam strojnog učenja ima zadaću prikupljene podatke razvrstati u K broj skupina na temelju zajedničkih karakteristika tih podataka. Princip rada algoritma je u opisan u pet koraka [24]:

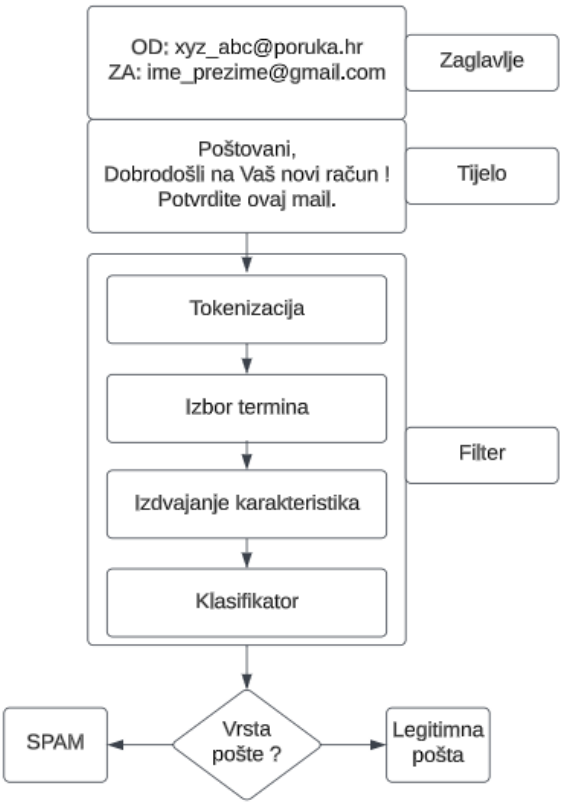
1. Označavanje broja skupina
2. Opisivanje različitih centroida za svaku K skupinu podataka – vrši se kako bi se oblikovali centroidi oko kojih bi se stvarale grupe sa sličnim karakteristikama
3. Ponavljanje pregledavanja karakteristika svakog podatka kako bi se odredilo kojoj skupini odnosno kojem je centroidu taj podatak najbliži, te se na taj način podatak dodjeljuje K skupini podataka čijem je centroidu najbliži
4. Nakon što se neki podatak pridruži skupini računaju se novi centroidi za novonastalu skupinu
5. Ponavljanje koraka 3. sve dok se ne dođe do toga da su centroidi svih K skupina statični tj. da se ponovnim računanjem njihova pozicija ne mijenja.

Kada se postigne da se centroidi svih K skupina podataka postanu stacionarni znači da je uspješno napravljen filter koji će na temelju tih dobivenih skupina podataka odlučivati radili se o *spam* i *phishing* pošti ili je riječ legitimnoj dolaznoj pošti. Karakteristike koje se u slučaju *spam* i *phishing* pošte gledaju ovise o arhitekturi e-pošte. Arhitektura e-pošte je jednostavna sastoji se od zaglavlja i tijela. U dijelu zaglavlja nalaze se adrese pošiljatelja i primatelja te naslov pošte. U tijelu se nalaze podaci poruke koja se šalje ili prima. Karakteristike prema kojima se stvaraju centroidi putem k-srednjih vrijednosti algoritma strojnog učenja prikupljaju se uglavnom iz zaglavlja pošte u kojima se za uzorak mogu uzimati dijelovi adresa pošte pošiljatelja. Te adrese su u velikoj većini slučajeva sumnjive već na pregled „golim okom“ pa se tako dosta lako može napraviti skupina podataka koji imaju sumnjive adrese pošiljatelja e-pošte. Karakteristike se još mogu prikupljati iz tijela pošte u kojima se također mogu nalaziti linkovi sa sumnjivim odredišnim adresama ili sumnjivim riječima unutar tijela e-pošte [24].

Pojednostavljeni prikaz arhitekture filtera kojim se vrši kategorizacija pošte nalazi se na slici 8. Na slici se vidi zaglavlje i tijelo kao arhitektura e-pošte iz kojih se vade podaci i karakteristike pomoću kojih se pošta filtrira. U samom filteru odvijaju se četiri postupka [24]:

- Tokenizacija – postupak u kojemu se ključne riječi izvlače iz konteksta cjelokupne poruke koja se npr. nalazi unutar tijela e-pošte
- Izbor termina (engl. *Term Selection*) – postupak u kojemu se izvučene riječi rangiraju prema važnosti
- Izdvajanje karakteristika (engl. *Feature Extraction*) – postupak u kojemu se detaljnije pregledavaju izvučene riječi te se dodatno reduciraju i dobiva se manja skupina podataka od važnosti
- Klasifikator – dio u kojemu se na temelju K znači algoritma određuje kojoj skupini K podatkovnih setova izvučeni podatak/riječ pripada

Na kraju se prilikom izlaska iz filtera dobije konačno podatak je li riječ o legitimnoj dolaznoj pošti ili je riječ o *spam/phishing* pristigloj pošti.



Slika 8. Arhitektura Spam filtera

Izvor: [24]

3.3. Duboko učenje i umjetne neuronske mreže u području kibernetičke sigurnosti

Duboko učenje prema [25] može se smatrati podvrstom strojnog učenja iako se općenito klasificira kao podvrsta umjetne inteligencije. Duboko učenje zajedno sa umjetnim neuronskim mrežama pokušava imitirati čovjekov način razmišljanja i učenja te na temelju tog razmišljanja izvršava razne zadaće [26].

Razlika između strojnog učenja i dubokog učenja je u tome što su modeli stvoreni korištenjem algoritama strojnog učenja naučeni prilagođavati se stanju u svojoj okolini i bez obzira na sitne promjene u okruženju ispravno izvršiti svoj zadatak. Dok su modeli nastali od kombinacije ANN i DL naučeni oponašati ljudski način razmišljanja samo isključivo za poznate situacije [27]. Prednosti korištenja DL u odnosu na ML je u tome što nestrukturirani podaci mogu biti zahtjevni i izazovni za algoritme strojnog učenja dok za algoritme dubokog učenja to ne predstavlja poteškoće. Još jedna od prednosti je u tome što model nastao kombinacijom ANN i DL algoritama nema potrebu za ljudskim interveniranjem u stvarnom okruženju dok je kod modela ML to potrebno [27].

Algoritmi dubokog učenja u kombinaciji sa ANN-om rade na principu slično ljudskome mozgu, ljudski mozak sastoji se od miliona međusobno povezanih neurona koji zajedno uče i procesuiraju informacije koje dolaze do mozga. Na sličan princip umjetne neuronske mreže sastoje se od više slojeva umjetnih neurona koji unutar računala/sustava zajedno rade i nastoje procesuirati dobivene podatke [27]. Duboka umjetna neuronska mreža sastoji se od tri glavna sloja. Prvi sloj je ulazni sloj (engl. *Input Layer*), ovaj sloj se sastoji od nekolicine čvorova koji ubacuju dolazne podatke u mrežu. Nakon ulaznog sloja nalazi se skriveni sloj (engl. *Hidden Layer*) u kojemu se mogu nalaziti stotine dodatnih skrivenih slojeva koji služe za analiziranje ulaznih podataka kako bi se model ispravno naučio. Na kraju mreže nalazi se izlazni sloj (engl. *Output Layer*) koji daje izlazne podatke nakon obrade i analize u skrivenom sloju. Broj čvorova u ovome sloju ovisi o količini izlaza na primjer ukoliko su izlazni podaci samo „DA“ ili „NE“ onda će se u izlaznom sloju nalaziti samo dva čvora [27].

Tri najčešće korištene vrste umjetnih neuronskih mreža prema [28] su: umjetna neuronska mreža s naprednim prijenosom (engl. *Feedforward neural networks,* FNNs), umjetna konvolucijska neuronska mreža (engl. *Convolutional neural networks,* CNNs) i rekurentna umjetna neuronska mreža (engl. *Recurrent neural networks,* RNNs). FNNs je najjednostavnija vrsta ANN sa linearnim protokom informacija kroz mrežu. FNNs se koriste za zadatke kao što su na primjer klasifikacija slika, prepoznavanje govora itd. CNNs je posebna vrsta neuronskih mreža koje se koriste za zadatke kao što su analiza videozapisa ili fotografija. Imaju mogućnost dobrog zapažanja detalja i učenja karakteristika iz videozapisa ili fotografija zbog čega se koriste prilikom detekcije objekata, segmentacije slika i ostalih zadataka. RNNs koriste se za prepoznavanje govora, automatsko prevođenje s jednog jezika na drugi jezik itd. iz razloga što imaju mogućnost obrade manjih dijelova podataka što znači i detaljniju obradu [28].

U području kibernetičke sigurnosti algoritmi dubokog učenja mogu se koristiti za detekciju DDoS napada, detekciju zlonamjernih softvera, detekciju botnet mreže, analizu mrežnog prometa i detekcije anomalija unutar istoga itd. [29].

Korištenje ANN i DL algoritama u detekciji DDoS napada u kibernetičkim sustavima prema [30] može se podijeliti na pet koraka. Prvi korak je kao i kod svakog od algoritama strojnog učenja prikupljanje podataka tj. podatkovnih setova koji u drugome koraku prolaze kroz proces predobrade. Proces predobrade je drugi korak u kojemu se vrši postupak u kojim se odabrani podaci „čiste“ zatim se izabiru ključne karakteristike unutar tih podataka te se podaci pripremaju za daljnju obradu. Sljedeći korak je podjela pripremljenih podataka u tri

skupine. Tri skupine u koje se dijele podaci su: skupina podataka za trening, skupina podataka za provjeru odnosno validacijsku fazu te skupina podataka za testiranje. Sve ove skupine podataka koriste se u fazi treniranja samo u različitim stadijima te faze. U četvrtome koraku koriste se podatkovne skupine za trening fazu i fazu validacije. Ovaj korak je korak u kojemu se obavlja dizajniranje i strukturiranje duboke umjetne neuronske mreže. U ovome koraku vrši se postupak podešavanja hiper parametara zajedno sa postupkom validacije kako bi se dobila ispravna i optimalna struktura mreže. Nakon što je odrađen proces određivanja optimalne strukture mreže odrađuje se proces evaluacije dizajniranog modela korištenjem testne skupine podataka [30].

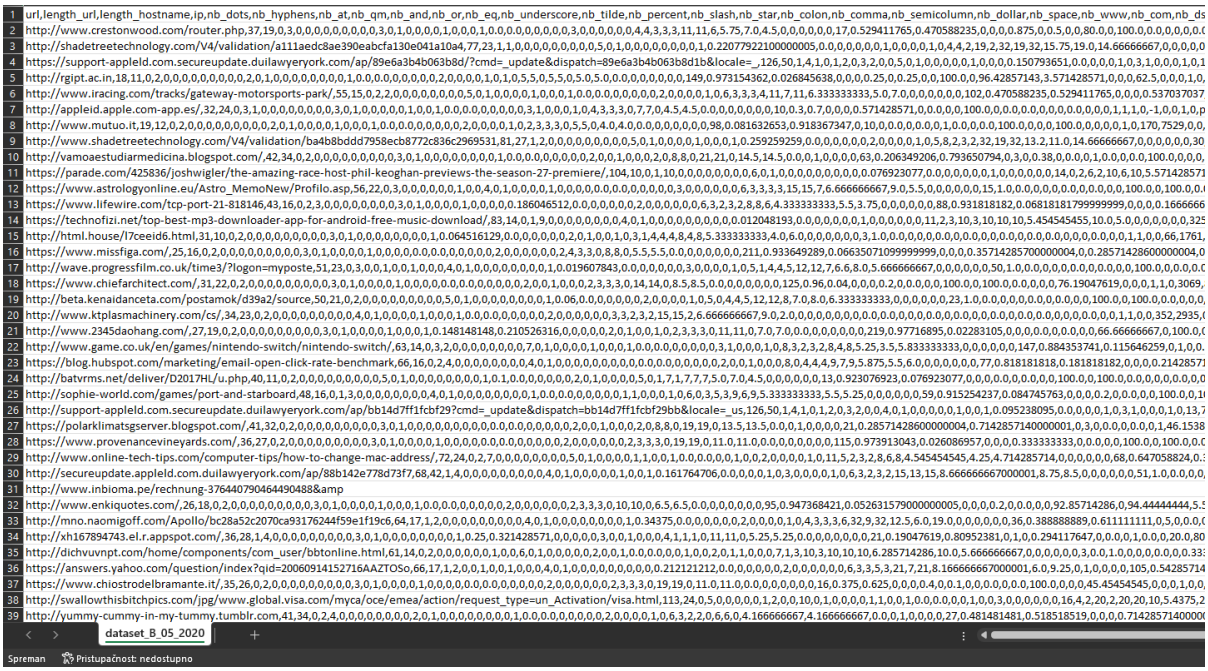
Ulazni sloj ANN-a ima zadaću primanja podataka iz podatkovne skupine i uglavnom jedan čvor unutar ulaznog sloja ANN-a odgovara jednoj dimenziji ili broju karakteristika unutar podataka. Skriveni sloj prima podatke iz ulaznog sloja. Ovaj sloj i ostali mogući skriveni slojevi unutar njega imaju zadaću odrediti odgovarajući broj čvorova kako bi se model mogao naučiti i kako bi mogao imati mogućnost učenja kompleksnih informacija. Izlazni sloj prima podatke nakon što oni prođu kroz sve slojeve ispred njega. Broj čvorova ovog sloja je vrijednost broja vjerojatnosti koja se kreće od 0.00 do 1.00. Nakon ovog koraka ide postupak podešavanja hiper parametara koji se vrši kako bi proces učenja bio kvalitetniji i bolji [30].

4. Prikupljanje i predobrada podataka

Korišteni podatkovni skup za ovo istraživanje kreirana je u svibnju 2020 godine. Autori javno dostupne podatkovne skupine su A. Hannousse i S. Yahiouche [31].

4.1. Odabir podatkovnog skupa

Podaci su objavljeni na internetskoj stranici "Mendeley Data" 28. rujna 2020. godine. Podatkovni skup stvoren je za razvoj modela umjetne inteligencije za detekciju *phishing* internetskih stranica. Sastoji se od 11430 URL adresa jednoliko podijeljenih u omjeru 50/50 posto. Što znači da je 50% URL legitimnih stvarnih internetskih stranica i URL adresa dok je preostalih 50% *phishing* adresa. Na stranici se nalaze dva dostupna podatkovna skupa od kojih je korišten drugi pod nazivom "dataset_B_05_2020.csv" [31]. Podaci se nalaze u Microsoft Excel tablici sa i prilikom otvaranja te tablice dobiva se prikaz kao na slici 9.

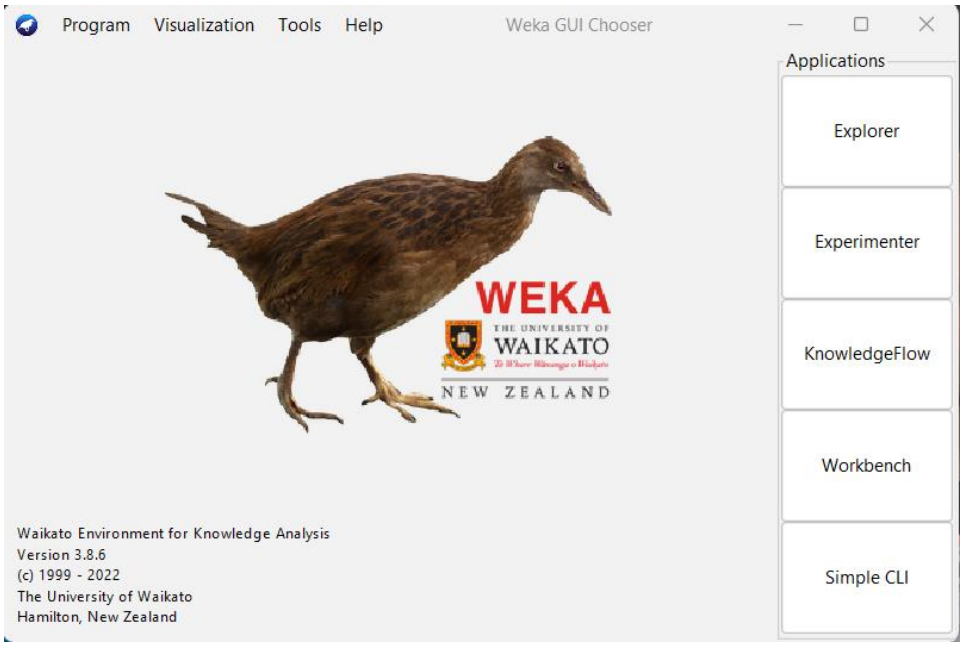


Slika 9. Prikaz podatkovne skupine unutar Microsoft Excel tablice

Nastavak datoteke „.csv" nije uobičajeni nastavak za Excel datoteku već je to datoteka zarezom odvojene vrijednosti (engl. *Comma-separated values*). Ova vrsta datoteke u kojima se nalazi veliki broj podataka te je pregled i rad sa tolikom količinom podataka olakšan koristeći ovu vrstu datoteke. Preuzeti podatkovni skup sastoji se od 89 značajki odnosno atributa za prepoznavanje *phishing* adrese. Neke od tih atributa su: dužina URL-a, IP adresa, broj znakova u adresi, broj točki koji se nalaze unutar URL-a, starost domene, količina mrežnog prometa koja se generira na stranici itd. [31].

4.2. Platforma za obradu podataka „Weka“

Alat Weka koristi se za zadatke kao što su rudarenje podacima (engl. *Data mining*) za koje koristi kolekciju algoritama strojnog učenja. Sadrži alate za predobradu, klasifikaciju, regresiju, klasterizaciju te vizualizaciju podataka i skupova. Kao programski jezik Weka alat koristi Java programski jezika. Prednosti ovog alata su što ima grafičko korisničko sučelje (engl. *Graphical user interface,* GUI) koje olakšava rad korisnicima koji nemaju iskustva sa radom Weke ili mogućnostima koje ona pruža. U velikoj većini slučajeva Weka alat se koristi za pripremu i predobradu podataka koji će se kasnije koristiti za određenu vrstu strojnog ili dubokog učenja [32].



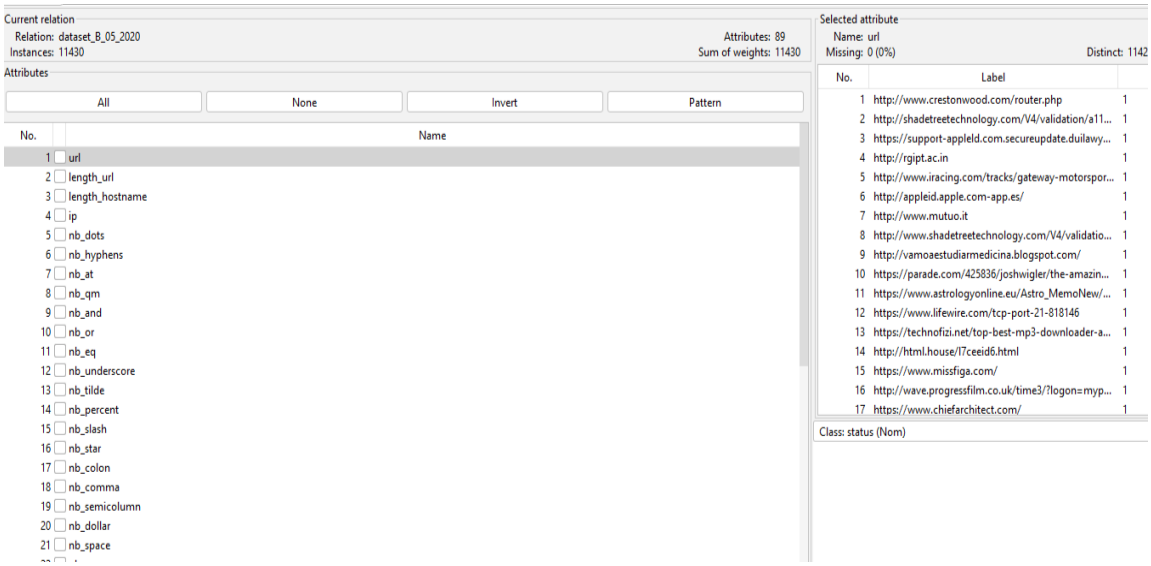
Slika 10. Polazni izbornik Weka 3 alata

Slika 10. prikazuje polazni izbornik koji se otvara prilikom pokretanja Weka 3 alata. Na desnoj strani izbornika nalazi se lista sa različitim „programima“ unutar Weke. Odabirom na karticu „*Explorer*“ korisnik ima mogućnost provedbe pokusa nad svojom podatkovnom skupinom, oblikovanjem podataka kako mu odgovara za njegove buduće radnje sa podatkovnom skupinom. U ovome dijelu programa moguć je proces predobrade podataka. Prilikom otvaranja te kartice otvara se novi prozor u koji je podijeljen na pet osnovnih kartica u kojima je moguće s podacima vršiti predobradu, klasifikaciju, klasterizaciju, vizualizaciju itd. Izborom „*Experimenter*“ korisniku se omogućuje provođenje pokusa sa podatkovnom skupinom i algoritmima strojnog učenja te analizom rezultata [33].

4.3. Proces predobrade podataka

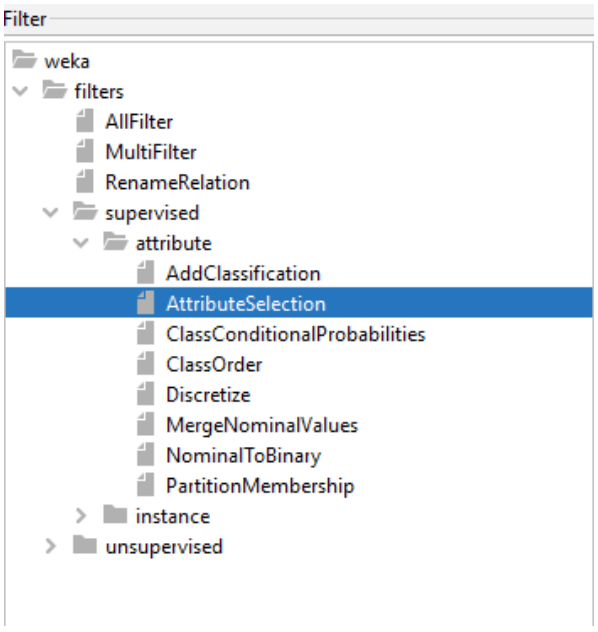
Proces predobrade podataka je od iznimne važnosti u postupku izrade modela strojnog učenja. Predobrada podataka uključuje čišćenje, transformiranja te pripremu podataka kako bi oni bili kompatibilni za rad sa određenom vrstom algoritama strojnog učenja te kako bi se isti ti podaci kasnije mogli analizirati. Proces predobrade podataka je važan jer podatkovni skupovi u velikoj većini slučajeva sadrže informacije tj. podatke koji stvaraju redundanciju te mogu biti nebitni. Ukoliko ima dosta podataka koji imaju iste vrijednosti dolazi do zalihosti ili redundancije koja usporava proces i povećava mogućnost pogreške ili izrade kvalitetnog modela. Postupak predobrade podataka može se podijeliti u više tehnika koje zajedno čine postupak predobrade. Čišćenje podataka jedna je od tih tehnika koja čisti odnosno briše ili prepravlja podatke koji su duplikati, imaju pogrešne vrijednosti ili kojima nedostaju vrijednosti. Druga tehnika je transformacija podataka kojom se podaci unutar podatkovnog skupa dorađuju na načine kako bi se povećala efikasnost i preciznost prilikom izrade modela strojnog učenja. Proces predobrade podataka prije izrade modela strojnog učenja važan je jer uvelike povećava preciznost izrađenih modela strojnog učenja jer se fokusira na važne podatke i smanjuje mogućnost nastanka šumova i zalihosti sa nepotrebnim podacima. Također, smanjuje mogućnost od *overfitting-a*, omogućava štednju vremena tijekom izrade modela jer je manja količina podataka za obradu te omogućava lakšu analizu i interpretabilnost dobivenih rezultata i modela [34].

Prvi korak u procesu predobrade podataka u Weka programskom alatu je učitavanje preuzetog podatkovnog skupa. Za ovo istraživanje korišten je programski alat Weka verzija 3.8.6 i podatkovni skup naziva „dataset_B_05_2020“. Prilikom pokretanja programa otvara se početni izbornik koji je prikazan na slici 10. Slika 11. prikazuje prozor nakon učitavanja podatkovnog skupa. Na lijevoj strani slike nalaze se učitani atributi podatkovnog skupa, a na desnoj se nalazi prvih 17 URL linkova od ukupno 11430. Iznad popisa atributa nalazi se kartica sa nazivom podatkovnog skupa, ukupnim brojem atributa te ukupnim brojem URL-ova.



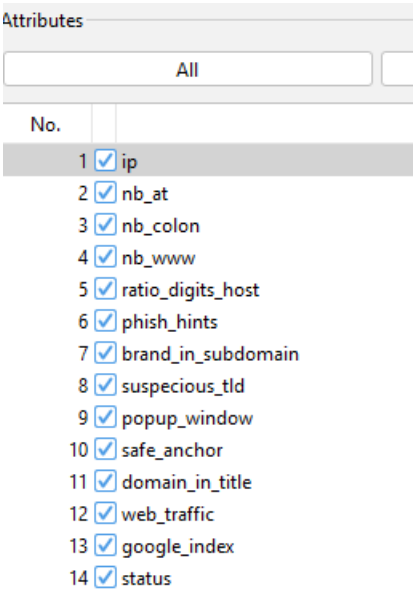
Slika 11. Prikaz učitanih podataka

Nakon učitavanja podataka moguće je odabrati atribute koji su nepotrebni te ih je moguće maknuti sa liste. Odabirom mogućnosti koja se nalazi unutar podmapa programskog alata Weka moguće je odabrati filtere koji automatski listu atributa skraćuju. Opcija odabira filtera korisniku daje mogućnost brisanja svih onih atributa koji nisu potrebni za izradu modela korištenjem željene vrste strojnog učenja. Slika 12. prikazuje padajući izbornik unutar programskog alata Weka sa filterima za nadzirano strojno učenje. U ovom slučaju nakon odabira filtera za nadzirano strojno učenje „*Attribute Selection*“ podatkovni skup tj. atributi unutar njega će se sa 89 smanjiti na 14.



Slika 12. Prikaz učitanih podataka

Slika 13. prikazuje 14 odabranih atributa koji su ostali primjenom filtera „*Attribute Selection*“. Među preostalim atributima su atribut IP adrese, atribut postojanja skočnih prozora itd. Nakon primjene filtera podatkovni skup je spreman za razvoj modela nadziranog strojnog učenja. Isti proces se može primijeniti sa podatkovnim skupom za primjenu filtera za nenadzirano strojno učenje. Nakon primjene filtera potrebno je novonastali podatkovni skup spremiti.



Slika 13. Atributi nakon primjene filtera

5. Primjena metoda strojnog učenja u području kibernetičke sigurnosti

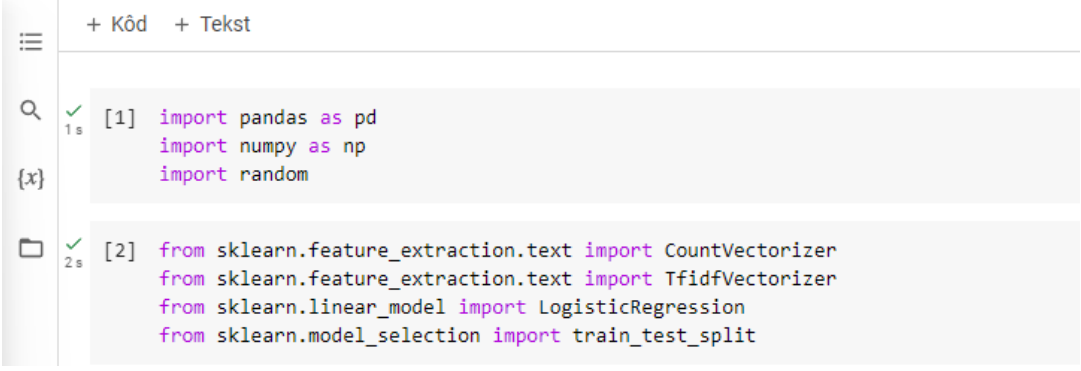
Na predobrađenim skupovima podataka provedeni su algoritmi strojnog učenja u svrhu razvoja modela koji će samostalno prepoznavati internetske adrese koje se smatraju *phishing* adresama i potencijalnim prijetnjama za sustave. Izrađeni modeli rade samo sa URL adresama koje se nalaze unutar preuzetog skupa podataka. Za izradu modela korištena je Google-ova platforma Google Colaboratory te programski jezik Python.

5.1. Platforma za primjenu strojnog učenja Google Colaboratory

Google Colaboratory ili kraće Colab je aplikacija unutar Google-ovog cloud servisa koja omogućuje korisnicima upisivanje i izvršavanje kodova programskog jezika Pythone bez potrebe za preuzimanjem programa kao što su Git, Python, UNIX Shell itd. Razlog izbora Google-ove platforme je taj što je ona pogodna za izradu modela strojnog učenja, edukaciju i analizu podataka, a uz to je i jednostavna za korištenje. Prednosti Google Colab-a su što je besplatna i s obzirom da je dio Google-ovog paketa svi kodovi, modeli, projekti se mogu pohraniti unutar Google Diska i na računalo. Uz to prednosti su mu još i što je povezan sa alatima kao što su TensorFlow, PyTorch itd. te stranicama koje sadrže skupove podataka kao što je na primjer Kaggle [35].

5.2. Izrada modela primjenom algoritma logističke regresije

Prvi korak u izradi model logističke regresije je učitavanje paketa strojnog učenja i knjižnica (engl. *Library*). Knjižnice koje se učitavaju su *„pandas“, „numpy“ i „random*“. NumPy se koristi za široki spektar matematičkih operacija koje se provode prilikom izrade modela strojnog učenja. Pandas je Pythonova knjižnica koja se koristi za rad sa podatkovnim skupovima i omogućuje mogućnosti kao što su čišćenje, istraživanje i manipulacija sa podacima unutar podatkovnih skupova. Na slici 14. prikazan je postupak učitavanja knjižnica te paketa strojnog učenja.



Slika 14. Učitavanje Python knjižnica i paketa strojnog učenja

Nakon uspješnog učitavanja knjižnica i paketa potrebno je učitati predobrađeni podatkovni skup. Postupak učitavanja podatkovnog seta i prikaz učitanih podataka iz podatkovnog seta prikazan je na slici 15.

```
from google.colab import files
uploaded = files.upload()
```

```
Choose Files  dataset_B_05_2020.csv
• dataset_B_05_2020.csv(text/csv) - 3661166 bytes, last modified: 7/20/2023 - 100% done
Saving dataset_B_05_2020.csv to dataset_B_05_2020 (2).csv
```

Dvaput kliknite (ili pritisnite enter) za uređivanje

```
[56] urls_data = pd.read_csv("dataset_B_05_2020.csv")
```

```
[57] type(urls_data)
```

```
pandas.core.frame.DataFrame
```

```
[80] urls_data.head()
```

	url	ip	nb_at	nb_dots	...	status
0	http://www.crestonwood.com/router.php	0	0	3	...	legitimate
1	http://shadetreetechnology.com/V4/validation/a...	1	0	1	...	phishing
2	https://support-appleId.com.secureupdate.duila...	1	0	4	...	phishing
3	http://rgipt.ac.in	0	0	2	...	legitimate

Slika 15. Učitavanje i prikaz učitanog podatkovnog skupa

Naredbom *urls_data = pd.read_csv(„dataset_B_05_2020.csv“)* vrši se učitavanje tj. upis podataka u skriptu nakon čega se sa naredbom *urls_data.head()* zahtjeva ispis točnije prikaz podataka unutar podatkovnog skupa. Ovom naredbom učitava se prvih 5 podataka u ovome slučaju popis URL-ova i atributa unutar podatkovnog skupa. Sa naredbom *urls_data.head(50)* moguće je zatražiti istu naredbu, ali za prvih 50 linija podataka u podatkovnom skupu.

Sljedeći korak u izradi modela je postupak izrade tokena. Tokeni se koriste za postupak tokenizacije tj. izvlačenja ključnih riječi iz podatka u ovom slučaju iz adresa. Na slici 16. prikazan je postupak izrade tokena. Kod za izradu tokena se sastoji od tri dijela. Prvi dio je kreiranje tri tokena. Prvi token je kreiran nakon pojavljivanja znaka „ / “ unutar internetske adrese, drugi token se odnosi za pojavljivanje znaka „ - “ unutar adrese te treći token se kreira za pojavljivanje točke unutar adrese. Drugi dio izrade tokena je uklanjanje redundantnih tokena i posljednji korak je uklanjanje mogućih tokena koji u sebi sadrže „ .com “ s obzirom da se taj nastavak učestalo pojavljuje u podatkovnom skupu.

```
[60] def makeTokens(f):
         tkns_BySlash = str(f.encode('utf-8')).split('/')
         total_Tokens = []
         for i in tkns_BySlash:
             tokens = str(i).split('-')
             tkns_ByDot = []
             for j in range(0,len(tokens)):
                 temp_Tokens = str(tokens[j]).split('.')
                 tkns_ByDot = tkns_ByDot + temp_Tokens
             total_Tokens = total_Tokens + tokens + tkns_ByDot
         total_Tokens = list(set(total_Tokens))
         if 'com' in total_Tokens:
             total_Tokens.remove('com')
         return total_Tokens
```

Slika 16. Kreiranje tokena

Nakon što su se uspješno izradili tokeni, sljedeći korak je izrada oznaka i atributa te korištenje izrađenih tokena. Slika 17. prikazuje odabir oznaka i atributa te postupak spremanja dobivenih vektora u varijablu X koja će se kasnije koristiti prilikom izrade modela.

```
✓ 0s [90] y = urls_data["status"]

✓ 0s [91] url_list = urls_data["url"]

✓ 0s [92] vectorizer = TfidfVectorizer(tokenizer=makeTokens)

✓ 0s [93] X = vectorizer.fit_transform(url_list)
```

Slika 17. Izrada oznaka i atributa

Na sljedećoj slici 18. prikazan je postupak podjele podatkovnog skupa u dvije grupe u omjeru 80/20. 80% podataka iz podatkovnog skupa koristit će se u fazi treniranja dok će se preostalih 20% podataka koristiti u fazi testiranja. Također na slici 18. prikazan je ispis preciznosti koja je ostvarena u fazi treniranja modela te ona iznosi 0.88539 to jest 88.54%.

```
[94] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

[95] logit = LogisticRegression()
     logit.fit(X_train, y_train)

     ▾ LogisticRegression
     LogisticRegression()

[96] print("Accuracy ",logit.score(X_test, y_test))

     Accuracy  0.8853893263342082
```

Slika 18. Podjela podataka u dvije skupine

Preostaje još izvršiti fazu testiranja izrađenog modela. Korišteni su preostalih 20% podataka za ovu fazu. Slika 19. prikazuje postupak faze testiranja model i rezultata preciznosti. Postupak testiranja se ponavlja dva puta za različite internetske adrese i za različitu količinu adresa. U prvom testiranju uzete su tri adrese dok su u drugome korištene četiri. Na kraju se ostvaruje postotak preciznosti predviđanja *phishing* adresa 0.90245 ili 90.24%.

```
[97] X_predict = ["https://www.provenancevineyards.com/", "http://appleid.apple.com-app.es/", "https://www.iracing.com/tracks/gateway-motorspo"

[98] X_predict = vectorizer.transform(X_predict)
     New_predict = logit.predict(X_predict)

[99] print(New_predict)

     ['legitimate' 'phishing' 'legitimate']

[100] X_predict1 = ["www.buyfakebillsonlinee.blogspot.com",
      "www.unitedairlineslogistics.com",
      "www.stonehousedelivery.com",
      "www.silkroadmeds-onlinepharmacy.com" ]

[101] X_predict1 = vectorizer.transform(X_predict1)
      New_predict1 = logit.predict(X_predict1)
      print(New_predict1)

      ['phishing' 'phishing' 'phishing' 'phishing']

[102] vectorizer = TfidfVectorizer()

[103] X = vectorizer.fit_transform(url_list)
      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

[104] logit = LogisticRegression()
      logit.fit(X_train, y_train)

      ▾ LogisticRegression
      LogisticRegression()

[106] print("Accuracy ",logit.score(X_test, y_test))

      Accuracy  0.9024496937882764
```

Slika 19. Faza testiranja i ispis rezultata

Ovaj model izrađen je prema modelu izrađenom na izvoru [36] sa vlastitim podacima i potrebnim prepravkama koda.

5.3. Izrada modela primjenom algoritma potpornih vektora

Za izradu modela korištenjem SVM algoritma koristi se isti podatkovni skup kao i za model koji je izrađen korištenjem metode logističke regresije. U prvom koraku izrade modela učitavaju se knjižnice i paketi strojnog učenja iz programa Scikit Learn. Za razliku od prethodnog modela logističke regresije u ovome primjeru od knjižnica učitava se samo „*numpy*“. Postupak učitavanja paketa, knjižnica i podatkovnog skupa prikazan je na slici 20.

```
[1] import pandas as pd
    from sklearn.model_selection import train_test_split
    from sklearn.svm import SVC
    from sklearn.metrics import classification_report
    from sklearn.metrics import precision_score
```

```
[2] from google.colab import files
    uploaded = files.upload()
```

Choose Files dataset_B_...2020(1).csv

- dataset_B_05_2020(1).csv(text/csv) - 496289 bytes, last modified: 8/1/2023 - 100% done

```
Saving dataset_B_05_2020(1).csv to dataset_B_05_2020(1).csv
```

Slika 20. Učitavanje podatka i podatkovnog skupa za SVM algoritam

Nakon učitavanja potrebnih podataka sljedeći korak je ispis svih stupaca unutar podatkovnog skupa u kojima se nalaze atributi prema kojima se razvija model. Naredbom „*data.columns*“ ispisuju se svi stupci. Nakon ispisa svi stupci osim posljednjeg tj. stupca „*status*“ uzimaju se kako bi se kreirali atributi dok se stupac „*status*“ uzima kao oznaka (engl. *Label*). Slika 21. prikazuje linije koda od 6 do 9 u kojima se obrađuju postupci podjele na oznake i atribute, ispis stupaca u podatkovnom skupu te u devetoj liniji koda koja je također prikazana na slici odrađuje se postupak podjele podataka podatkovnog skupa na podatke za fazu treninga i fazu testiranja u omjeru 80/20.

```
[6]  data.columns

     Index(['ip', 'nb_at', 'nb_colon', 'nb_www', 'ratio_digits_host', 'phish_hints',
            'brand_in_subdomain', 'suspecious_tld', 'popup_window', 'safe_anchor',
            'domain_in_title', 'web_traffic', 'google_index', 'status'],
           dtype='object')

[7]  df = pd.get_dummies(df, columns=['ip', 'nb_at', 'nb_colon', 'nb_www', 'ratio_digits_host', 'phish_hints',
             'brand_in_subdomain', 'suspecious_tld', 'popup_window', 'safe_anchor',
             'domain_in_title', 'web_traffic', 'google_index'])

[8]  X = df.drop('status', axis=1)
     y = df['status']

[9]  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
```

Slika 21. Rad sa atributima i oznakama te podjela podataka za faze treninga i testiranja

Kada se izvrši podjela podataka vrši se treniranje modela uz korištenje SVM algoritma i postupak prepoznavanja statusa podataka unutar podatkovnog skupa za trening fazu izrade modela. Ovaj korak prikazan je slikom 22.

```
✓ 1 min  [11] clf = SVC(kernel='linear')
              clf.fit(X_train, y_train)

              ▾ SVC
              SVC(kernel='linear')

✓ 26 s   [12] predictions = clf.predict(X_test)

✓ 23 s   [13] print("Training precision", clf.predict(X_test))

         Training precision ['phishing' 'legitimate' 'legitimate' ... 'phishing' 'phishing' 'phishing']
```

Slika 22. Faza treniranja modela SVM algoritma

Posljednji korak u izradi modela za prepoznavanje *phishing* URL adresa korištenjem SVM algoritma je izvršiti fazu testiranja. Faza testiranja rezultirala je sa prosječnom preciznošću prepoznavanja od 0.92576 tj. 92.58%. Prosječna preciznost je iz razloga što ovaj kod za SVM algoritam daje rezultate za prepoznavanje posebno legitimnih i posebno za *phishing* adrese. Preciznost prepoznavanja legitimnih adresa iznosi 93% dok za *phishing* adrese iznosi oko 91%. Slika 23. pokazuje postupak testne faze i ispis rezultata.

```
[14] y_pred = clf.predict(X_test)

[15] precision = precision_score(y_test, y_pred, pos_label='legitimate')

print("Precision:", precision)
print(classification_report(y_test, predictions))
```

```
Precision: 0.9257602862254025
              precision

  legitimate       0.93
    phishing       0.91
```

Slika 23. Ispis preciznosti prepoznavanja modela za SVM algoritam

Za pomoć prilikom izrade modela korištenjem SVM algoritma korišten je kod sa izvora [37] prilagođen za rad sa vlastitim podatkovnim skupom.

5.4. Izrada modela primjenom algoritma slučajnih šuma

Za izradu modela korištenjem algoritma slučajnih šuma potrebno je uvesti knjižnice „*numpy*“ i „*pandas*“ te paket strojnog učenja iz Scikit Learn-a za rad sa algoritmom slučajnih šuma. Nakon toga ponovno je potrebno učitati podatkovni skup podataka te podijeliti podatke za trening i testnu fazu kao i u svakome od primjera do sada.

Nakon inicijalnih postupaka izvršavaju se postupci treniranja modela i testiranja modela te faze provjere preciznosti unutar faze treniranja i faze testiranja modela. Na slici 24. prikazan je postupak treniranja i testiranja modela.

```
[7]  forest = RandomForestClassifier(n_estimators=10)

[8]  forest.fit(X_train,y_train)
```

```
RandomForestClassifier
RandomForestClassifier(n_estimators=10)
```

```
[9]  y_train_forest = forest.predict(X_train)
     y_test_forest = forest.predict(X_test)
```

Slika 24. Postupak treniranja i testiranja modela RF algoritma

U ovome primjeru „*n_estimators*“ koji inače označava broj stabla unutar algoritma slučajne šume postavljena je vrijednost 10. Vrijednost 10 je početna zadana vrijednost i znači da će algoritam prilikom izrade kreirati 10 različitih stabla.

Po završetku faze treniranja i testiranja ispisuju se dobiveni rezultati preciznosti prepoznavanja koji u ovom slučaju također imaju prosječnu vrijednost jer isto kao i u slučaju kod algoritma potpornih vektora računaju preciznost prepoznavanja legitimnih i *phishing* napada tj. adresa. Za legitimne adrese preciznost iznosi 94%, a za *phishing* adrese 93%. U ovome primjeru preciznost prepoznavanja na podatkovnom skupu označenom za trening fazu iznosi 0.973 tj. 97.3%. Na podatkovnom skupu za fazu testiranja preciznost prepoznavanja iznosi 0.933 ili 93.3%. Slika 25. prikazuje ispis rezultata dobivenih korištenjem algoritmom slučajnih šuma.

```
[10] acc_train_forest = metrics.accuracy_score(y_train,y_train_forest)
     acc_test_forest = metrics.accuracy_score(y_test,y_test_forest)
     print("Random Forest : Accuracy on training Data: {:.3f}".format(acc_train_forest))
     print("Random Forest : Accuracy on test Data: {:.3f}".format(acc_test_forest))
     print()

     Random Forest : Accuracy on training Data: 0.973
     Random Forest : Accuracy on test Data: 0.933
```

```
[11] print(metrics.classification_report(y_test, y_test_forest))

                  precision

       legitimate       0.94
         phishing       0.93
```

Slika 25. Ispis rezultata preciznosti prepoznavanja modela za RF algoritam

5.5. Izrada modela primjenom algoritma stabla odlučivanja

Za izradu modela primjenom DT algoritma potrebno je najprije iz Scikit Learn-a uvesti paket strojnog učenja za DT algoritam. Slika 26. prikazuje učitavanje paketa strojnog učenja, sljedeća linija koda pokazuje kreiranje modela koji se nakon kreiranja trenira, Prilikom kreiranja modela linija koda sadrži hiper parametar „*max_depth*=30“. Ovaj hiper parametar kod algoritma stabla odlučivanja utječe na kompleksnost cijelog algoritma. Što je vrijednost ovog hiper parametra veća izrada modela i obrada podataka postaje kompleksnija. Prilikom odabira vrijednosti treba paziti na jer može doći do pojava *underfittinga* i *overfittinga* koje mogu utjecati na rad i preciznost izrađenog modela [38].

```
[9]  from sklearn.tree import DecisionTreeClassifier

[10] tree = DecisionTreeClassifier(max_depth=30)

[11] tree.fit(X_train, y_train)

     DecisionTreeClassifier
     DecisionTreeClassifier(max_depth=30)

[12] y_train_tree = tree.predict(X_train)
     y_test_tree = tree.predict(X_test)
```

Slika 26. Učitavanje paketa ML i odabir hiper parametra

Dobivena preciznost prepoznavanja *phishing* adresa iznosi 0.91 odnosno 91%. U fazi treniranja modela postotak preciznosti iznosi 97%. Na slici 27. prikazan je postupak ispisa dobivenih rezultata prilikom izrade modela. Izrada modela napravljena je prema uputama i primjeru koji se nalazi na izvoru [37].

```
[16] acc_train_tree = metrics.accuracy_score(y_train,y_train_tree)
     acc_test_tree = metrics.accuracy_score(y_test,y_test_tree)
     print("Decision Tree : Accuracy on training Data: {:.3f}".format(acc_train_tree))
     print("Decision Tree : Accuracy on test Data: {:.3f}".format(acc_test_tree))
     print()

     Decision Tree : Accuracy on training Data: 0.979
     Decision Tree : Accuracy on test Data: 0.923

[17] print(metrics.classification_report(y_test, y_test_tree))

                  precision

       legitimate       0.93
         phishing       0.91
```

Slika 27. Ispis dobivenih rezultata za model izrađen primjenom DT algoritma

5.6. Izrada modela primjenom algoritma K-najbližih susjeda

KNN je algoritam nadziranog strojnog učenja koji se bazira na utvrđivanju sličnosti između podataka te ih prema tome svrstava u zasebne klastere. Prednosti korištenja KNN algoritma su njegova jednostavnost za korištenje i implementaciju, mogućnost rada sa velikim podatkovnim skupovima itd. [39].

Nakon učitavanja vrši se proces izrade modela. Za ovaj model karakterističan je hiper parametar "*n_neighbor=1*" Ovaj hiper parameter označava broj susjeda koji će glasati u koju skupinu se svrstava podatak koji se obrađuje. Zadana početna vrijednost je 5, a preporuka je staviti neparan broj kako nebi došlo do izjednačenog broja glasova prilikom raspodjele [40].

Slika 28. Prikazuje postupak treniranja modela i odabir vrijednosti "1" za hiper parameter "*n_neighbor*".

```
[16] knn = KNeighborsClassifier(n_neighbors=1)

[17] knn.fit(X_train,y_train)

     KNeighborsClassifier
     KNeighborsClassifier(n_neighbors=1)

[18] y_train_knn = knn.predict(X_train)
     y_test_knn = knn.predict(X_test)
```

Slika 28. Odabir vrijednosti hiper parametra i postupak treniranja/testiranja modela

Ostvareni rezultat prepoznavanja *phishing* internetskih adresa modela koji je izrađen primjenom KNN algoritma u fazi treninga iznosi 0.971 tj. 97.1% dok u fazi testiranja iznosi 85%. Ispisano izvješće rezultata ovog modela nalazi se na slici 29. Ovaj model je također izrađen prema primjeru koji se nalazi na izvoru [37].

```
acc_train_knn = metrics.accuracy_score(y_train,y_train_knn)
acc_test_knn = metrics.accuracy_score(y_test,y_test_knn)
print("K-Nearest Neighbors : Accuracy on training Data: {:.3f}".format(acc_train_knn))
print("K-Nearest Neighbors : Accuracy on test Data: {:.3f}".format(acc_test_knn))
print()
```

```
K-Nearest Neighbors : Accuracy on training Data: 0.971
K-Nearest Neighbors : Accuracy on test Data: 0.854
```

```
print(metrics.classification_report(y_test, y_test_knn))
```

```
              precision

  legitimate       0.88
    phishing       0.83
```

Slika 29. Ispis dobivenih rezultata za model izrađen primjenom KNN algoritma

6. Analiza rezultata

Ispisi rezultata i izvješća za modele koji su izrađeni prema [37] sastoje se od rezultata za prepoznavanje legitimnih i *phishing* adresa zasebno. U fazi treninga daju prosječnu preciznost za prepoznavanje *phishing* adresa. Svi modeli izrađeni su sa istim obrađenim podatkovnim skupom odnosno sa istim brojem atributa, istim brojem adresa itd. Za sve modele korištena je ista raspodjela podataka u omjeru 80% podataka za fazu treniranja modela, a 20% podataka za fazu testiranja.

6.1. Analiza rezultata preciznosti prepoznavanja

Gledajući svaki model zasebno najmanje odstupanje između preciznosti postignute prilikom faze testiranja i faze treninga ima algoritam potpornih vektora. Razlika između faze treninga i faze testiranja za SVM algoritam iznosi 1.6%. Na drugome mjestu je algoritam logističke regresije čija razlika između dvije faze iznosi 1.7%, nakon LR algoritma sljedeću veću razliku ima algoritam slučajne šume i ona iznosi 4%. Algoritam stabla odlučivanja ima razliku 6.9% između rezultata postignutih na fazama testiranja i treninga, a najveću razliku od 14.1% ima algoritam K-najbližih susjeda.

Najbolji postotak preciznosti prepoznavanja *phishing* URL adresa prilikom faze treniranja modela ima algoritam stabla odlučivanja. Izradom modela korištenjem DT algoritma u fazi treninga ostvarena je preciznost prepoznavanja od 97.90%. Modeli koji su izrađeni korištenjem algoritama slučajne šume i K-najbližih susjeda su na drugom i trećem mjestu sa 97.30% i 97.10%. Korištenjem algoritma SVM na fazi treninga ostvarena je preciznost od 92.60% dok je najlošije rezultate u ovoj fazi ostvario algoritam logističke regresije čija je preciznost iznosila 88.54%.

U fazi testiranja sa preostalih 20 posto podataka iz podatkovnog skupa najbolji postotak preciznosti prepoznavanja *phishing* URL adresa ostvario je model izrađen korištenjem algoritma slučajne šume. Postotak preciznosti tog modela iznosi 93.3%, na drugom mjestu nalaze se modeli izrađeni korištenjem algoritmima stabla odlučivanja i potpornih vektora sa postotkom prepoznavanja od 91%. Nakon njih nalazi se model logističke regresije koji je također blizu rezultatu prepoznavanja DT i SVM algoritama sa 90.24%. Najlošiji rezultati postignuti u fazi testiranja dobiveni su korištenjem modela K-najbližih susjeda čiji postotak prepoznavanja iznosi 83%.

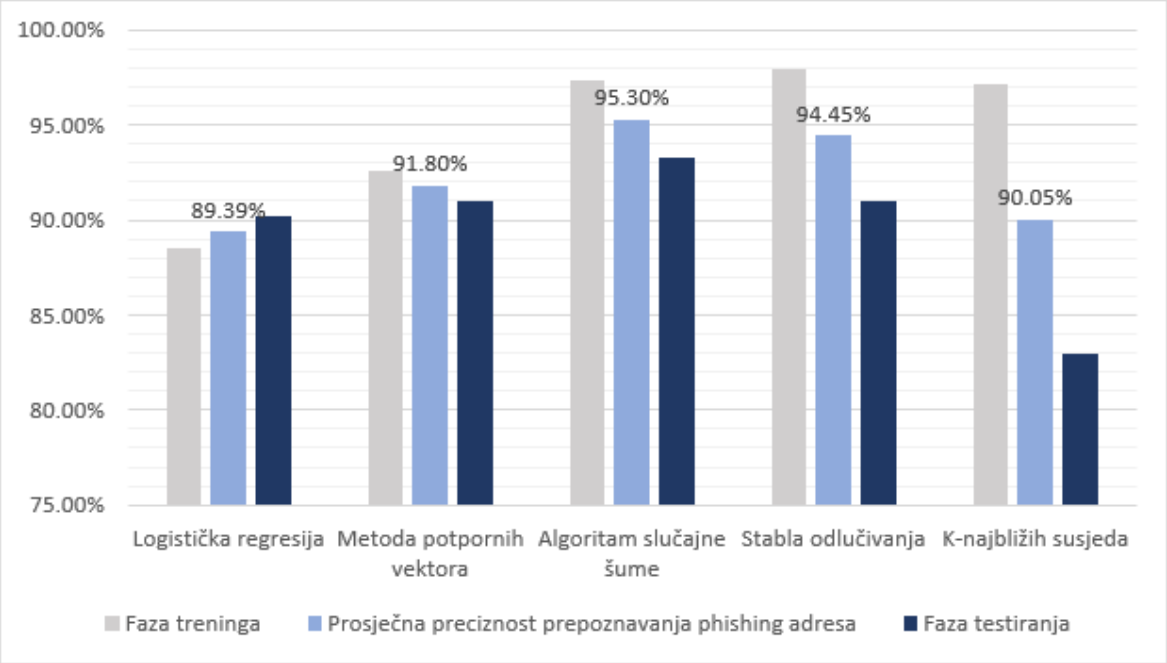
Tablica 1. Pregled dobivenih rezultata istraživanja

Algoritam strojnog učenja	Preciznost faze treninga	Preciznost faze testiranja
Logistička regresija	88.54%	90.24%
Metoda potpornih vektora	92.60%	91%
Slučajne šume	97.30%	93.3%
Stabla odlučivanja	97.9%	91%
K-najbližih susjeda	97.1%	83%

Tablica 1. prikazuje pregled dobivenih rezultata prepoznavanja *phishing* adresa tijekom faze treniranja i faze testiranja modela.

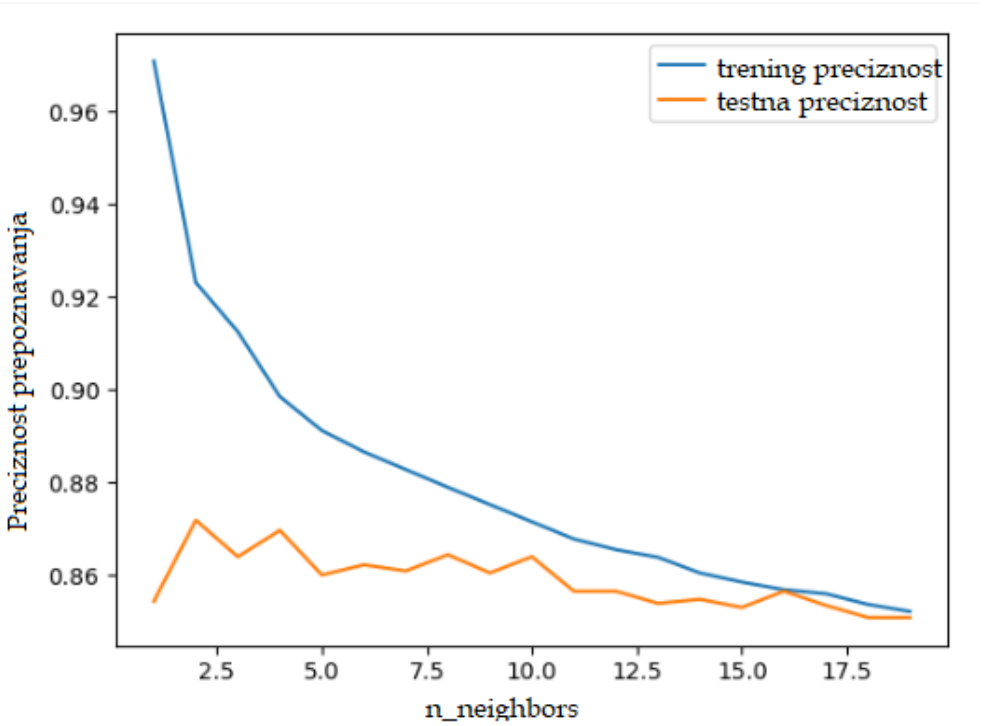
Analizom rezultata najneočekivanija je razlika koja je nastala između rezultata dobivenih tijekom faze treninga i testiranja kod razvoja modela K-najbližih susjeda. Analizom koda nije utvrđeno zbog čega je došlo do toliko velikog odstupanja u postocima preciznosti prepoznavanja.

Kada bi se gledala prosječna preciznost ukoliko se u obzir uzmu rezultati sa faze treninga i faze testiranja. Najbolje rezultate ostvaruje model izrađen korištenjem algoritma slučajne šume sa postotkom prepoznavanja od 95.30%, na drugom mjestu jako blizu RF algoritmu je model stabla odlučivanja koji u prosjeku rezultata dviju faza ima postotak od 94.45%. Na trećem mjestu je model SVM algoritma sa 91.80%, iza njega nalazi se model razvijen korištenjem KNN algoritma sa 90.50% te na posljednjem mjestu je model logističke regresije sa 89.39%. Slika 30. prikazuje grafički prikaz dobivenih rezultata.



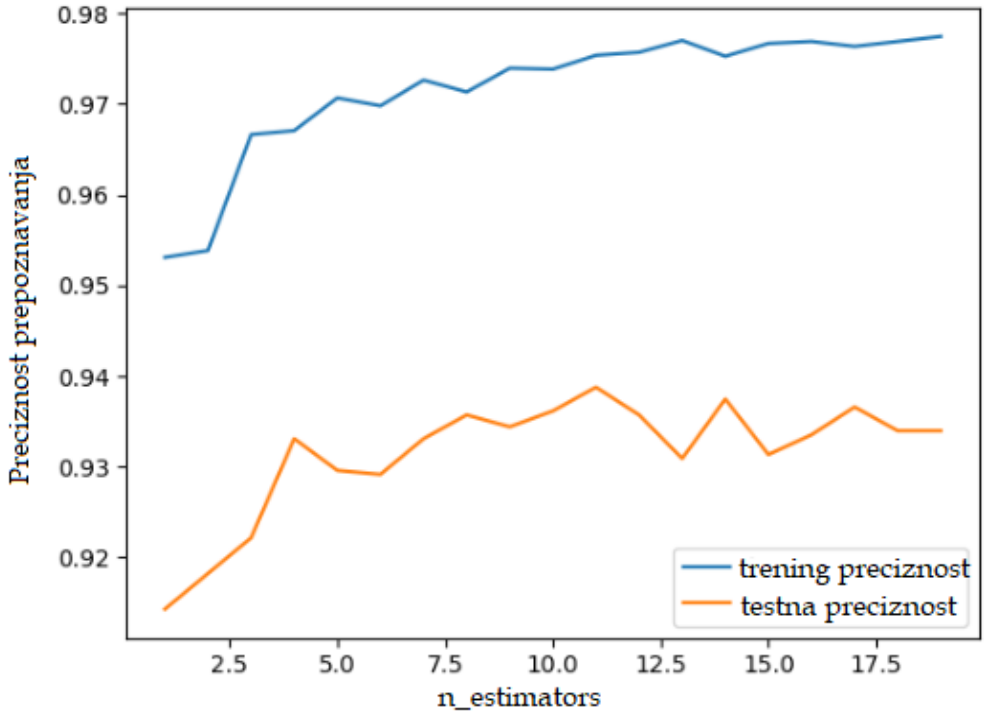
Slika 30. Grafički prikaz dobivenih rezultata

Slika 31. prikazuje grafički prikaz kako se mijenja rezultat preciznosti prepoznavanja *phishing* URL adresa za model izrađen korištenjem KNN algoritma u ovisnosti na promjenu hiper parametra „*n_neighbors*“. Iz grafa je vidljivo da što je veći broj susjeda točnost prepoznavanja se smanjuje za fazu treninga. Što se tiče rezultata unutar faze testiranja modela, rezultati osciliraju rastu i padaju oko vrijednosti od 0.86. Vidljivo je kako je u fazi treninga za uzetu vrijednost hiper parametra 1 preciznost preko 0.96 točnije 0.971, dok se sa povećanjem vrijednosti hiper parametra „*n_neighbors=15*“ preciznost znatno spušta do vrijednosti od 0.86.



Slika 31. Prikaz ovisnosti rezultata o vrijednosti hiper parametru "n_neighbors"

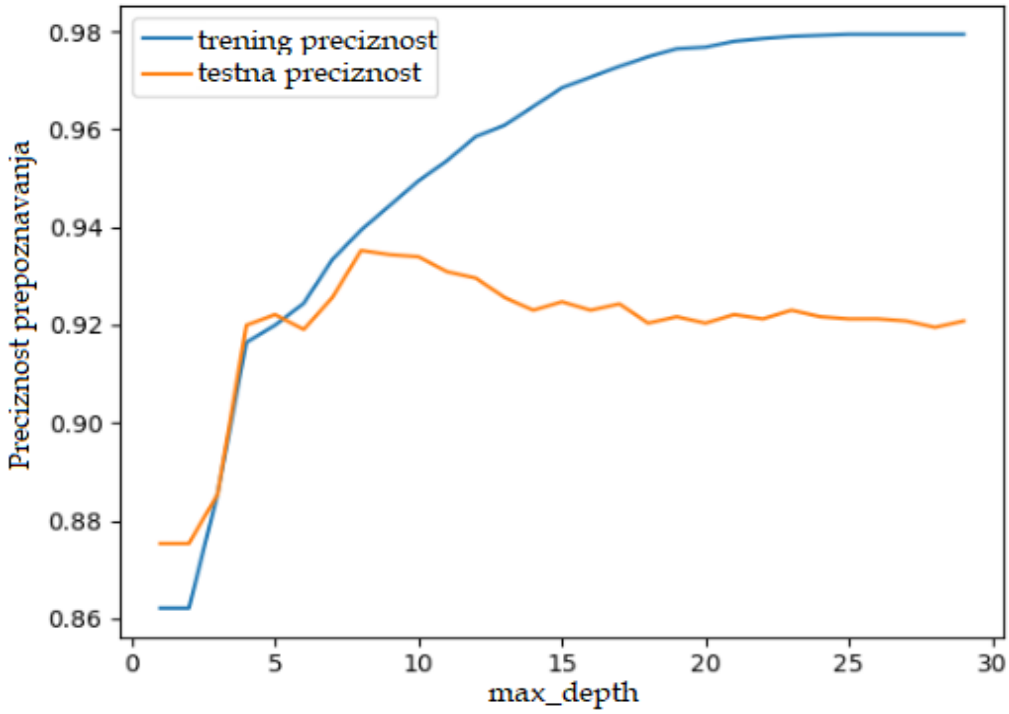
Sljedeći grafički primjer mogućih rezultata i trendova izrade modela je za model slučajne šume. Kod RF algoritma koristi se hiper parametar „*n_estimators*“ koji prikazuje broj stabla unutar šume. Prilikom izrade algoritma za vrijednost hiper parametra korištena je zadana vrijednost 10, a postignuta je preciznost od 0.973 za fazu treniranja odnosno 0.933 za fazu testiranja. Slika 32. prikazuje grafički prikaz ovisnosti rezultata preciznosti prepoznavanja *phishing* URL adresa u zavisnosti od promjene hiper parametra „*n_estimators*“.



Slika 32. Prikaz ovisnosti rezultata o vrijednosti hiper parametru "n_estimators"

Ako se u ovome primjeru također uzme promjena vrijednosti hiper parametra „*n_estimators=15*“ dobiveni rezultati se mijenjaju gotovo neznatno. Preciznost sa faze treninga bi se povećala porastom vrijednosti hiper parametra i za vrijednost od 15 iznosila bi 0.975. Dok bi se u fazi testiranja također promijenila, ali bi se smanjila na 0.93.

Za model izrađen primjenom algoritma stabla odlučivanja ključna je vrijednost hiper parametra „*max_depth*“. Prilikom izrade modela vrijednost parametra određena je na 30, a postignute preciznosti su 0.971 prilikom faze treninga i 0.91 za fazu testiranja. Slika 33. prikazuje kako se mijenja preciznost predviđanja ukoliko se parametar „*max_depth*“ smanji.



Slika 33. Prikaz ovisnosti rezultata o vrijednosti hiper parametru "max_depth"

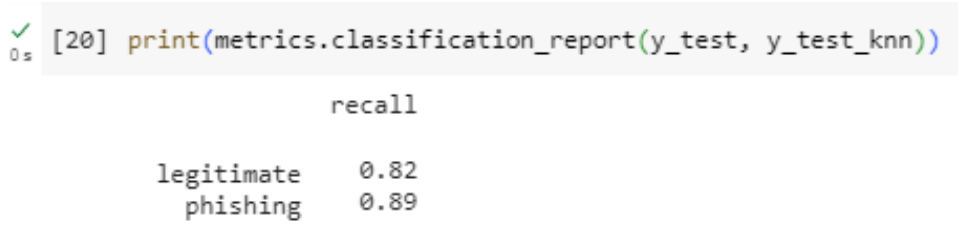
Na primjer ako se za vrijednost hiper parametra odabere „*max_depth=15*“ za fazu treninga dobiva se preciznosti od 0.969, a u fazi testiranja preciznost za tu vrijednost hiper parametra iznosi 0.924. Kod modela DT algoritma može se iščitati kako se sa smanjenjem hiper parametra, preciznost prepoznavanja smanjuje.

6.2. Analiza *recall* rezultata

Recall rezultat je vrijednost koja prikazuje mogućnost točnog predviđanja pozitivnog iz svih pozitivnih podataka, ova vrijednost još se zove i osjetljivost. Razlika između preciznosti prepoznavanja i *recall* vrijednosti je u tome što preciznost prepoznavanja govori o krajnjem postotku prepoznavanja, dok osjetljivost daje informaciju o tome koliko je model zapravo pozitivnih podataka prepoznao kao pozitivne. Na primjeru izrađenih modela za *phishing* detekciju to bi značilo da *recall* vrijednost daje informaciju o tome koliki je postotak prepoznavanja *phishing* adresa model ostvario od svih *phishing* adresa koje su mu ponuđene tijekom faze testiranja. Što je veća *recall* vrijednost veća to znači da je model strojnog učenja bolji jer ima veću osjetljivost na prepoznavanje [41].

Kod modela izrađenog prema KNN algoritmu *recall* vrijednost za legitimne URL iznosi 82% što znači da je model od svih legitimnih adresa koje su se koristile unutar faze testiranja prepoznao njih 82% kao legitimne dok je za ostale pogrešno odlučio. Za *phishing* adrese

recall vrijednost kod KNN modela iznosi 89% što je bolje u odnosu na legitimne URL adrese. Slika 34. prikazuje ispis rezultata *recall* vrijednosti za legitimne i *phishing* adrese KNN modela.



Slika 34. Ispis recall vrijednosti za KNN model

Tablica 2. Prikaz rezultata recall vrijednosti izrađenih modela

Algoritam strojnog učenja/*Recall* vrijednost	Legitimne adrese	*Phishing* adrese
Logistička regresija	89%	91%
Metoda potpornih vektora	91%	93%
Slučajne šume	93%	94%
Stabla odlučivanja	91%	93%
K-najbližih susjeda	82%	89%

Tablica 2. prikazuje dobivene rezultate *recall* vrijednosti za izrađene modele. Iz dobivenih rezultata vidljivo je kako su rezultati izrađenih modela približno isti osim kod modela koji je izrađen primjenom KNN algoritma koji ponovno ima lošije rezultate u odnosu na preostale modele kao što je imao i najveće odstupanje kod rezultata preciznosti predviđanja *phishing* adresa. Najbolju osjetljivost ostvario je model algoritma slučajne šume čije je vrijednost za prepoznavanje legitimnih adresa među legitimnim adresama iznosi 93% a za *phishing* adrese iznosi 94%.

6.3. Analiza „F1-mjera“ rezultata

F1 vrijednost predstavlja harmonijsku sredinu vrijednosti preciznosti prepoznavanja i *recall* vrijednosti. Od važnosti je jer daje informaciju o kvaliteti izlaza koju model daje. Omogućuje uvid u stanje modela te time mogućnost optimizacije preciznosti prepoznavanja ili osjetljivosti modela ovisno o potrebama korisnika [41].

Matematički izraz za izračunavanje F1 vrijednosti glasi:

$$F1 = \frac{2*Preciznost*Recall}{(Preciznost+Recall)}$$

```
[20] print(metrics.classification_report(y_test, y_test_knn))
0s

                  f1-score

  legitimate      0.85
    phishing      0.86
```

Slika 35. Ispis rezultata F1-mjere

Slika 35. prikazuje ispis dobivenih rezultata za F1-vrijednosti koje su dobivene primjenom KNN algoritma strojnog učenja. Iz slike je vidljivo kako je za legitimne adrese F1-mjera 85% dok je za *phishing* adrese 86%.

Tablica 3. Prikaz rezultata F1-mjere

Algoritam strojnog učenja	F1-mjera
Logistička regresija	89.6%
Metoda potpornih vektora	91.4%
Slučajne šume	94.4%
Stabla odlučivanja	93.2%
K-najbližih susjeda	87.8%

Tablica 3. prikazuje rezultate F1-mjere dobivene matematičkim putem korištenjem ranije napisanog izraza. Dobiveni rezultati pokazuju kako model izrađen korištenjem algoritma slučajne šume ima najveću i najbolju vrijednost F1-mjere koja iznosi 94.4% dok najlošiju vrijednost ima model izrađen prema KNN algoritmu što ima smisla prema se F1-mjera ovisi o vrijednostima preciznosti prepoznavanja i osjetljivosti modela u kojima je također ovaj model ostvario najslabije rezultate.

7. Zaključak

Rad donosi pregled dosadašnjih rezultata primjene umjetne inteligencije kao načina zaštite od kibernetičkih prijetnji. Pregledom dosadašnjih istraživanja daje se mogućnost analize rezultata i analiza razvoja metoda i algoritama strojnog učenja od vremena kada su ta istraživanja provedena do danas kada je ovo istraživanje izvršeno. Pregledom dosadašnjih istraživanja može se zaključiti kako je mogućnost primjene metoda umjetne inteligencije kao što su strojno učenje i duboko učenje širokog spektra. Korištenjem ML i DL algoritama mogu se razviti modeli koji će biti dijelovi sustava kao što su IDS za prepoznavanje neovlaštenih upada u sustave. Također kroz analizu dosadašnjih istraživanja vidljiva je razna mogućnost primjene algoritama kada je riječ o detekciji *phishing* napada koje je moguće prepoznati na više načina uz pomoć modela DT i ML.

Modeli koji su izrađeni u ovom radu naučeni su prema pet algoritama strojnog učenja: DT, KNN, SVM, RF i LR algoritmu. Najbolji prosječni rezultat prepoznavanja *phishing* URL adresa iz preuzetog i obrađenog podatkovnog skupa ostvario je model izrađen sa algoritmom slučajne šume. Također analizom ostalih dobivenih rezultata može se zaključiti kako je model izrađen korištenjem algoritma slučajne šume za ovo istraživanje najkvalitetnije izrađen i kako je najpodobniji za prepoznavanje *phishing* URL adresa iz podatkovnog skupa podataka.

Kod razvijenih modela rezultati su dali visoku učinkovitost u odrađivanju zadanog zadatka, ali sa mogućnošću za poboljšanjima. Kako bi se dodatno poboljšali rezultati prepoznavanja potencijalnih malicioznih adresa, za treniranje modela mogao bi se dodatno unaprijediti podatkovni skup iako ima već trenutno 89 atributa prema kojima se izrađuje model mogao bi se dodati još koji atribut koji se ne nalazi u skupu da bi se poboljšala preciznost modela. Jedan od tih atributa bi mogao biti npr. *domain rating* koji bi ukazivao na ocjenu web adrese i time utjecao na njezinu autentičnost i nemalicioznost.

U zaključku se još može navesti i mana izrađenih modela, a to je da izrađeni modeli imaju visoku točnost u radu i detekciji sa adresama unutar podatkovnog skupa dok za adrese koje nisu unutar podatkovnog skupa postotak preciznosti prepoznavanja *phishing* URL adresa pada.

Literatura

[1] Europski parlament. Što je umjetna inteligencija i kako se upotrebljava?. Preuzeto sa: https://www.europarl.europa.eu/news/hr/headlines/society/20200827STO85804/sto-je-umjetna-inteligencija-i-kako-se-upotrebljava [Pristupljeno: svibanj 2023.]

[2] Dasgupta D, Akhtar Z, Sen S. Machine Learning in Cybersecurity: a comprehensive survey. The Journal of Defensive Modeling & Simulation: Applications, Methodology, Technology 2022; 19(1) 57-106.

[3] Kolla J, Praneeth S, Sameed Baig M, Reddy Karri G. A comparison study of machine learning techniques for phishing detection. 2022; 4(1) : Advances on business and information system (E-ISSN: 2685-2543).

[4] Radivilova T, Kirichenko L, Ageiev D, Bulakh V. Classification Methods of Machine Learning to Detect DDos Attacks. 2019. 10th IEEE International Conference on Intelligent Dana Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS); p. 207-210.

[5] Peraković D, Periša M, Cvitić I, Zorić P, Kuljanić T. M, Aleksić D. Opportunities of Using Machine Learning Methods in Telecommunications and Industry 4.0 – A Survey. 7th EAI International Conference of Management of Manufacturing Systems.

[6] Chawla A. Phishing website analysis and detection using Machine Learning. 2022. International Journal of Intelligent Systems And Applications in Engineering (ISSN:2147-6799)

[7] Aleksić D. Mogućnosti primjene metoda strojnog učenja u području telekomunikacija. Sveučilište u Zagrebu; Fakultet prometnih znanosti. Završni rad. Preuzeto sa: https://repozitorij.fpz.unizg.hr/islandora/object/fpz%3A2355/datastream/PDF/view [Pristupljeno: lipanj 2023.]

[8] Springboard. What Is Cybersecurity? A Complete Overview Guide. O. Agbeleye. 2023. Preuzeto sa: https://www.springboard.com/blog/cybersecurity/what-is-cybersecurity/ [Pristupljeno: kolovoz 2023.]

[9] Cvitić I. *Temeljna terminoglogija sigurnosti informacijsko-komunikacijskog sustava.* Prezentacija. Sigurnost i zaštita informacijsko komunikacijskih sustava. Fakultet prometnih znanosti Sveučilišta u Zagrebu. 3. Ožujka 2022.

[10] Technopedia. 50+ Cybersecurity Statistics for 2023 You Need to Know – Where, Who & What is Targeted. N. Kolesnikov. 2023. Preuzeto sa: https://www.techopedia.com/cybersecurity-statistics [Pristupljeno: kolovoz 2023]

[11] Statista. Number of unique phishing sites detected worldwide from 3rd quater 2013 to 34th quater 2022. Preuzeto sa: https://www.statista.com/statistics/266155/number-of-phishing-domain-names-worldwide/ [Pristupljeno: kolovoz 2023]

[12] IT Governance. How to Spot a Phishing Email: With Examples. L. Irwin. 2022. Preuzeto sa: https://www.itgovernance.co.uk/blog/5-ways-to-detect-a-phishing-email [Pristupljeno: kolovoz 2023.]

[13] ENISA. Artificial Intelligence and Cybersecurity Research.2023. ENISA Research and Innovation Brief.

[14] k-Means Advantages and Disadvantages. Preuzeto sa: https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages [Pristupljeno: lipanj 2023.]

[15] Medium. K-Means cluster and it's use case in Cyber Security. Saha A. 2021. Preuzeto sa: https://arnabsaha1.medium.com/k-means-cluster-and-its-use-case-in-cyber-security-3abfaab2ec09 [Pristupljeno: lipanj 2023.]

[16] Markey J. Using Decision Tree Analysis for Intrusion Detection: A How-To-Guide. Preuzeto sa: https://sansorg.egnyte.com/dl/6edQgfwngE [Pristupljeno: lipanj 2023.]

[17] Kumar M, Hanumanthappa M, Kumar, T. V. S. Intrusion Detection System Using Decision Tree Algorithm.
Preuzeto sa: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6511281
[Pristupljeno: lipanj 2023.]

[18] Ghanem K, Aparicio-Navarro F. J, Kyriakopoulos K. G, Lambotharan S, Chambers J. A. Support Vector Machine for Network Intrusion and Cyber-Attack Detection. Preuzeto sa: https://core.ac.uk/download/288367383.pdf [Pristupljeno: lipanj 2023.]

[19] AWS. What IS OVerfitting?. Preuzeto sa: https://aws.amazon.com/what-is/overfitting/ [Pristupljeno: lipanj 2023.]

[20] Medium. SVM: Feature Selection and Kernels. Ippolito P. P. Preuzeto sa: https://towardsdatascience.com/svm-feature-selection-and-kernels-840781cc1a6c [Pristupljeno: lipanj 2023.]

[21] Jha J, Ragha L. Intrusion Detection System using Support Vector Machine. International Journal of Applied Information Systems (IJAIS) – ISSN: 2449-0868. Preuzeto sa: https://research.ijais.org/icwac/number3/icwac1342.pdf

[22] CERT.hr. SPAM. Preuzeto sa: https://www.cert.hr/19795-2/spam/ [Pristupljeno: lipanj 2023]

[23] Investopedia. Phishing: What it is And How to Protect Yourself. Hayes A. 2022. Preuzeto sa: https://www.investopedia.com/terms/p/phishing.asp [Pristupljeno: lipanj 2023.]

[24] Sharma A, Rastogi V. Spam Filtering using K mean Clustering with Local Feature Selection Classifier. 2014;108(10); International Journal of Computer Applications (0975-8887).
Preuzeto sa:https://research.ijcaonline.org/volume108/number10/pxc3900096.pdf

[25] Simplilearn. What is Deep Learning and How Does It Works. Reyes K. 2023. Preuzeto sa: https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-deep-learning#what_is_deep_learning [Pristupljeno: srpanj 2023.]

[26] AWS Amazon. What is Deep Learning ?. Preuzeto sa: https://aws.amazon.com/what-is/deep-learning/ [Pristupljeno: srpanj 2023.]

[27] Coursera. Deep Learning vs. Machine Learning: Beginner's Guide. 2023. Preuzeto sa: https://www.coursera.org/articles/ai-vs-deep-learning-vs-machine-learning-beginners-guide [Pristupljeno: srpanj 2023.]

[28] geeksforgeeks. Introduction to Deep Learning.
Preuzeto sa: https://www.geeksforgeeks.org/introduction-deep-learning/ [Pristupljeno: srpanj 2023.]

[29] Simplilearn. Exploring the Application of Deep Learning in Cyber Security. Rauch S. 2022. Preuzeto sa: https://www.simplilearn.com/application-of-deep-learning-in-cyber-security-article [Pristupljeno: srpanj 2023.]

[30] Khempetch T, Wuttidittachotti P. DDoS attack detection using deep learning. IAES International Journal of Artificial Intelligence (IJ-AI). 2021. 10(2); pp. 382-338. Preuzeto sa: https://ijai.iaescore.com/index.php/IJAI/article/view/20884/13116#

[31] MendeleyData. Web page phishing detection.
Preuzeto sa: https://data.mendeley.com/datasets/c2gw7fy2j4/2 [Pristupljeno: srpanj 2023.]

[32] Weka. Weka 3: Machine Learning Software in Java.
Preuzeto sa: https://www.cs.waikato.ac.nz/ml/weka/ [Pristupljeno: srpanj 2023.]

[33] Machine Learning Mastery. What is the Weka Machine Learning Workbench. Preuzeto sa: https://machinelearningmastery.com/what-is-the-weka-machine-learning-workbench/ [Pristupljeno: srpanj 2023.]

[34] Medium. Importance of Data Preprocessing in Machine Learning. A. Soumen. Preuzeto sa: https://levelup.gitconnected.com/importance-of-data-preprocessing-in-machine-learning-17045bc18d01 [Pristupljeno: kolovoz 2023.]

[35] Google Research. Colaboratory. Preuzeto sa:
https://research.google.com/colaboratory/faq.html [Prisutpljeno: srpanj 2023.]

[36] GitHub. Machine-Learning-In-Julia-JCharisTech. Preuzeto sa:
https://github.com/Jcharis/Machine-Learning-In-Julia-JCharisTech/blob/master/Detecting%20Malicious%20Url%20With%20Machine%20Learning%20In%20Python.ipynb [Pristupljeno: srpanj 2023.]

[37] GitHub. Phishing-Website-Detection-by-Machine-Learning-Tehniques. Preuzeto sa:
https://github.com/shreyagopal/Phishing-Website-Detection-by-Machine-Learning-Techniques/tree/master [Pristupljeno: srpanj 2023.]

[38] Scikit Learn. Importance of decision tree hyperparameters on generalization. Preuzeto sa: https://inria.github.io/scikit-learn-mooc/python_scripts/trees_hyperparameters.html [Pristupljeno: kolovoz 2023.]

[39]Java Point. K-nearest Neighbor(KNN) Algorithm fot Machine Learning. Preuzeto sa:
https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning
[Pristupljeno: srpanj 2023.]

[40] WordPress. Posts Tagged 'KneighborsClassifier explained'. Preuzeto sa: https://ashokharnal.wordpress.com/tag/kneighborsclassifier-explained/ [Pristupljeno: kolovoz 2023.]

[41] Vitalflux. Accuracy, Precision, Recall & F1-Score – Python Examples. A. Kumar. 2023. Preuzeto sa: https://vitalflux.com/accuracy-precision-recall-f1-score-python-example/#Terminologies_%E2%80%93_True_Positive_False_Positive_True_Negative_False_Negative [Pristupljeno: kolovoz 2023.]

Popis slika

Popis tablica

Sveučilište u Zagrebu
Fakultet prometnih znanosti
Vukelićeva 4, 10000 Zagreb

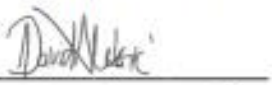
IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOSTI

Izjavljujem i svojim potpisom potvrđujem da je diplomski rad (vrsta rada) isključivo rezultat mojega vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu, a što pokazuju upotrijebljene bilješke i bibliografija. Izjavljujem da nijedan dio rada nije napisan na nedopušten način, odnosno da je prepisan iz necitiranog rada te da nijedan dio rada ne krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu završnog/diplomskog rada pod naslovom Primjena umjetne inteligencije u području kibernetičke sigurnosti, u Nacionalni repozitorij završnih i diplomskih radova ZIR.

Student/ica:

U Zagrebu, 1.9.2023.



(ime i prezime, *potpis*)