Primjena tehnika obrade prirodnog jezika u računalnoj sigurnosti

Bilić, Ružica

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Humanities and Social Sciences / Sveučilište u Zagrebu, Filozofski fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:131:620944

SVEUČILIŠTE U ZAGREBU
FILOZOFSKI FAKULTET
ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE ZNANOSTI
Ak. god. 2021./2022.

Ružica Bilić

Primjena tehnika obrade prirodnog jezika u računalnoj sigurnosti

Završni rad

Mentor: doc. dr. sc. Ivan Dunđer, mag. inf.

Zagreb, lipanj 2022.

Izjava o akademskoj čestitosti

Izjavljujem da je ovaj rad rezultat mog vlastitog rada koji se temelji na istraživanjima te objavljenoj i citiranoj literaturi. Izjavljujem da nijedan dio rada nije napisan na nedozvoljen način, odnosno da je prepisan iz necitiranog rada, te da nijedan dio rada ne krši bilo čija autorska prava. Također izjavljujem da nijedan dio rada nije korišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

_____

(potpis)

Sadržaj

1. Uvod

Računalna obrada prirodnog jezika (*engl. Natural Language Processing*) podrazumijeva skup tehnika koji omogućuju primjenu raznih postupaka i algoritama strojnog učenja nad govornim ljudskim jezikom zapisanom u digitalnom tekstualnom obliku. Tehnike obrade se najčešće koriste za stvaranje aplikacija koje pomažu u interakciji čovjek-stroj (poput govornih asistenata), aplikacija za razumijevanje i analizu tekstualnog sadržaja na internetu, optimiziranje tražilica i naslovnih stranica društvenih mreža. U tehnike obrade prirodnog jezika spadaju i svi principi skupljanja podataka, sastavljanja korpusa, čišćenje podataka od praznih vrijednosti kako bi se kreirao valjan skup podataka nad kojim se mogu dalje primijeniti razni načini obrade i manipuliranja tekstualnim podacima. Međutim, osim prethodno navedenih primjena, tehnike obrade prirodnog jezika se koriste u računalnoj sigurnosti, što će biti obrađeno u ovom završnom radu.

Tekstualni podaci predstavljaju važan aspekt računalne sigurnosti, od samog teksta sadržaja aplikacija do izvornog koda i zbog toga obrada prirodnog jezika nalazi svoju primjenu u gotovo svim načinima prepoznavanja sigurnosnih propusta. Pojam računalne sigurnosti predstavlja širok spektar zaštite, detekcije, prevencije i reakcije na sigurnosne propuste u digitalnom okruženju. Sigurnosne prijetnje mogu uključivati različite aspekte informacijskih tehnologija, od softwarea do hardwarea, detekcije računalnih virusa, sigurnosnih propusta u operativnim sustavima i u web aplikacijama. U ovom radu će se obraditi primjena tehnika obrade prirodnog jezika u računalnoj sigurnosti s obzirom na internetsko okruženje, što uključuje razne aspekte sigurnosti na internetu i sigurnosti samih web aplikacija.

Rad će biti podijeljen na teorijski i praktični dio. U teorijskom dijelu će se objasniti osnovni koncepti obrade prirodnog jezika u sintezi sa strojnim učenjem i klasifikacijama te općeniti principi računalne sigurnosti na primjerima odabranih web aplikacija. Sve ranije navedeno će se proučavati na primjerima URL-ova web stranica, e-mail porukama i izvornom kodu web aplikacije. Za svaki od primjera, u praktičnom dijelu rada, koristit će se odgovarajući skupovi podataka pomoću kojih će se moći identificirati različite vrste sigurnosnih propusta te će se vršiti i klasifikacija malicioznih domena , kao i sadržaja u e-mail porukama. Maliciozne, tj. zlonamjerne poruke (eng. phishing) su one kojima se od korisnika nastoje izvući osjetljivi podaci, kao što su lozinke, adrese, osobni podaci, ili ostvariti financijska korist (Kovač,

Dunđer, Seljan, 2022[1]). Zlonamjerne poruke mogu se razlikovati prema sadržaju, komunikacijskom kanalu i vrsti ciljanog korisnika. Programsko rješenje za praktični dio rada će biti prikazano u Pythonu, s detaljnim objašnjenjima koda i izlaznim podacima. Na kraju rada slijedi zaključak, popis literature, popis slika te sažeci sa ključnim riječima na hrvatskom i engleskom jeziku.

[1] Kovač, A.; Dunđer, I.; Seljan, S. An overview of machine learning algorithms for detecting phishing attacks on electronic messaging services. Proceedings MIPRO, 2022.

2. Osnovni principi računalne obrade prirodnog jezika

Da bi se mogli prepoznati izazovi primjene tehnika obrade prirodnog jezika nad tekstualnim podacima u domeni sigurnosti, najprije je potrebno objasniti osnovne principe kojima će se obrada vršiti kroz ovaj rad. Za početak je potrebno objasniti što je prirodni jezik. Najjednostavnije rečeno, to je način na koji se ljudi sporazumijevaju te je osnovno sredstvo komunikacije. Preciznija definicija bi bila da je jezik međusobno dogovoreni skup protokola koji uključuje riječi/zvukove koje koristimo za međusobnu komunikaciju[2].

Obrada prirodnog jezika područje je umjetne inteligencije koje strojevima daje mogućnost čitanja, razumijevanja i izvlačenja značenja iz ljudskih jezika[3]. Kako je u digitalnom okruženju prisutnost tekstualnih podataka koji se odnose na prirodni jezik velika, tako se može i zaključiti da su upravo takvi podaci važni i za sigurnost samih korisnika pa je primjena strojnog učenja nad takvim podacima potrebna, od čega se u suštini i sastoje tehnike obrade prirodnog jezika. Osim nad prirodnim jezikom, tehnike obrade prirodnog jezika se mogu koristiti i nad ostalim tekstualnim podacima poput izvornih kodova, koji se u nekim slučajevima moraju pročistiti od posebnih znakova da bi se mogli koristiti u takvoj obradi i klasificiranju podataka, kako navode Krstić, Seljan i Zoroja (2019[4]) i Dunđer, Pavlovski i Seljan (2020[5]).

2.1. Teorijska podloga tehnika obrade prirodnog jezika za tekstualne podatke u računalnoj sigurnosti

Kao što je prethodno navedeno u ranijim poglavljima, u domeni računalne sigurnosti se najčešće obrađuju tekstualni podaci (prirodni jezik ili niz znakova). Zbog toga će se u ovom radu staviti naglasak na tehnike obrade prirodnog jezika isključivo za tekstualne podatke.

Osnovni pojmovi za obradu tekstualnih podataka su:

- *Tokenizacija rečenice*
- *Tokenizacija riječi*

[2]Ghosh, Gunning. *Natural Language Processing Fundamentals.* 2019.
URL: https://www.packtpub.com/product/natural-language-processing-fundamentals/9781789954043
(3.7.2021.)
[3]ODSC – Open Data Science, *An Introduction to Natural Language Processing (NLP).* 2019.
URL: https://medium.com/@ODSC/an-introduction-to-natural-language-processing-nlp-8e476d9f5f59
(3.7.2021.)
[4] Krstić, Ž., Seljan, S., Zoroja, J. Visualization of Big Data Text Analytics in Financial Industry: A Case Study of Topic Extraction for Italian Banks. Proceedings of EntreNova, 2019, 67-75.
[5] Dunđer, I., Pavlovski, M., Seljan, S. Computational Analysis of a LiteraryWork in the Context of Its Spatiality. World Conference on Information Systems and Technologies, 2020, 252-261.

- *Lematizacija i oblikovanje teksta*
- *Stop-riječi*
- *Regularni izrazi*
- *Vreća riječi (engl. bag-of-words model)*
- *TF-IDF*[6]

2.1.1. Tokenizacija rečenice

Tokenizacija rečenice se odnosi na podjelu niza pisanog jezika u njegove sastavne rečenice. Najjednostavniji princip na koji tokenizacija djeluje je podjela prema točkama (ili drugim završnim interpunkcijskim znakovima) između rečenica, gdje će se od određenog teksta stvoriti lista rečenica na osnovi toga što će se tokeniziranjem pretpostaviti da točka (ili neki drugi završni interpunkcijski znak) označava kraj jedne rečenice, to jest da odvaja dvije rečenice. Međutim, pri tokenizaciji rečenice nekog teksta javlja se problem kratica iza kojih često slijedi znak točke[7]. Da bi se uspješno izbjegla zabuna, moguće je stvoriti listu najčešćih kratica u danom jeziku ili se mogu koristiti regularni izrazi. U oba slučaja, korištenje dodatnih leksikona ili regularnih izraza će povećati točnost u znatnom postotku[8].

Tokenizacija rečenice ne nalazi svoju široku primjenu u obradi prirodnog jezika za računalnu sigurnost, osim u slučajevima kada se radi o spam e-mail porukama i korisnički generiranim unosima koji koriste prirodni jezik. Razlog tomu je što se većina tehnika obrade prirodnog jezika posebno prilagođava za određene programske jezike i kodove te nije potrebno vršiti prepoznavanje samih rečenica prirodnog jezika, iako se principi tokenizacije (uključujući i tokenizaciju rečenice) mogu primijeniti nad bilo kakvim tekstualnim podacima. Razlika će biti u izlaznim rezultatima, gdje se kod tradicionalne primjene dobivaju rečenice, a kod primjene u računalnoj sigurnosti, najčešće niz znakova koji predstavljaju isječak važnog koda ili URL-a.

---

[6]Shukra, Iriondo. *Natural Language Processing (NLP) with Python — Tutorial*. 2020.
URL:https://pub.towardsai.net/natural-language-processing-nlp-with-python-tutorial-for-beginners-1f54e610a1a0
(3.7.2021.)
[7]Yordanov, V. *Introduction to Natural Language Processing for Text.* 2018.
URL: https://towardsdatascience.com/introduction-to-natural-language-processing-for-text-df845750fb63
(3.7.2021.)
[8]Grefenstette, Tapanainen. *What is a word, What is a sentence? Problems of Tokenization.* 1997.
URL: https://www.dfki.de/~neumann/qa-course/grefenstette94what.pdf
(3.7.2021.)

2.1.2. Tokenizacija riječi

Tokenizacija riječi se odnosi na podjelu niza pisanog jezika na njegove sastavne riječi. U hrvatskom jeziku kao i u ostalim jezicima latinične abecede i mnogim drugim jezicima koji koriste neki oblik latinične abecede, razmak (*engl. whitespace*) je dobar pokazatelj gdje jedna riječ prestaje, a druga počinje u nekom odabranom tekstu[9]. Tokenizacijom riječi pomoću razmaka se dobivaju tokeni kao izlazni rezultat koji u prirodnom jeziku predstavljaju odgovarajuće riječi.

Nastavno na primjenu u računalnoj sigurnosti, tokenizacija pomoću razmaka se može koristiti u više svrha, od kojih je najpopularnija analiza datoteka zapisnika (*engl. log file*), s kojim se izlaznim rezultatima tokeni dalje uglavnom grupiraju i nadalje koriste za klasifikaciju datoteka zapisnika[10].

2.1.3. Lematizacija i oblikovanje teksta

Cilj lematizacije i oblikovanja teksta je smanjiti količinu različitih oblika iste riječi, a ponekad i derivativno povezane oblike riječi na zajednički osnovni oblik[11]. Lematizacija je potrebna kod obrade prirodnog teksta i jezika jer tekstualni dokumenti često koriste različite oblike iste riječi. Primjena lematizacije je srodna postupku korjenovanja (*engl.* stemming) i uključuje odsijecanje krajeva riječi, kao i uklanjanje završetaka (sufiksa) uz upotrebu rječnika i morfološke analize riječi. Na navedeni način se dobiva osnovna forma riječi (lema) ili korijen riječi (stem)[12].

Lematizacija, u obradi tekstualnih podataka u odnosu na računalnu sigurnost, se najčešće koristi u analizi i predikcijama računalnih propusta, kao i računalnih prijetnji koji se mogu zajedničkim imenom nazvati sigurnosnim incidentima (*engl. cyber events*)[13]. Tehnike obrade

---

[9]Yordanov, V. *Introduction to Natural Language Processing for Text.* 2018.
URL: https://towardsdatascience.com/introduction-to-natural-language-processing-for-text-df845750fb63
(3.7.2021.)
[10]Perkins, J. *NLP FOR LOG ANALYSIS – TOKENIZATION.* 2018.
URL: https://streamhacker.com/category/insight-engines/
(3.7.2021.)
[11]Yordanov, V. *Introduction to Natural Language Processing for Text.* 2018.
URL: https://towardsdatascience.com/introduction-to-natural-language-processing-for-text-df845750fb63
(3.7.2021.)
[12]Manning, Raghavan, Schütze. *Introduction to Information Retrieval.* 2008.
URL: https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html
(3.7.2021.)
[13]Law Insider, *Cyber Event definition.*
URL: https://www.lawinsider.com/dictionary/cyber-event

prirodnog jezika se mogu primijeniti na klasifikaciju i grupiranje opisa incidenata te se daljnjom obradom može ustanoviti kako dobivena klasifikacija odgovara prioritetu događaja. Izlaznim rezultatima takve klasifikacije se mogu napraviti predviđanja koja će sadržavati preporuke za bolje sprečavanje, razumijevanje ili kontrolu događaja[14].

2.1.4. Stop-riječi

Stop-riječi predstavljaju skup znakova u obliku riječi prirodnog jezika koje se filtriraju prije ili nakon obrade tekstualnih podataka. Razlog uklanjanja stop-riječi je što mogu stvarati šum pri primjeni strojnog učenja nad tekstualnim podacima. Stop-riječi se uglavnom odnose na najčešće riječi, kao što su „i“, „je“ „su“, u nekom jeziku, međutim ne postoji univerzalni popis stop-riječi te ovisno o aplikaciji, popis se može mijenjati[15].

Primjena stop-riječi u računalnoj sigurnosti se može odnositi na onu primjenu koja odgovara klasičnoj obradi prirodnog jezika, što bi bilo detektiranje spam e-mail poruka pomoću obrade korištenog prirodnog jezika unutar tijela e-mail poruke. U tom slučaju bi se koristio popis najčešćih riječi unutar jezika na kojem su spam e-mail poruke napisane te bi se riječi s tog popisa uklonile iz tekstualnih podataka nad kojima će se raditi klasifikacija e-mail poruka na „spam“ i „ham“[16]. Stop-riječi se također mogu koristiti kod klasificiranja sigurnosnih incidenata (napada na računalnu mrežu), ali i u određenim slučajevima analize izvornog koda. Upotreba stop-riječi kod analize izvornog koda pomoću tehnika obrade prirodnog jezika se ne odnosi na klasičan popis stop-riječi, kao kod prijašnjih primjera (najčešće riječi u određenom prirodnom jeziku). U analizi izvornog koda popis stop-riječi bi se sastojao od najčešćih i nesignifikantnih (za računalnu sigurnost) dijelova koda, pa bi takav popis zapravo djelomično bio u domeni lematizacije, a ne klasičnog popisa stop-riječi jer se takav popis ne bi sastojao od najčešćih riječi nekog jezika nego od pojedinih isječaka koda, posebnih znakova, riječi i slova.

(3.7.2021.)
[14]Sharma, A. *Applying Data Science to Cybersecurity Network Attacks & Events*. 2019.
URL: https://www.kdnuggets.com/2019/09/applying-data-science-cybersecurity-network-attacks-events.html
(3.7.2021.)
[15]Yordanov, V. *Introduction to Natural Language Processing for Text.* 2018.
URL: https://towardsdatascience.com/introduction-to-natural-language-processing-for-text-df845750fb63
(3.7.2021.)
[16]Thorpe, Schwartz. *USING MACHINE LEARNING TO IDENTIFY PHISHING ATTACKS.* 2017.
URL:https://assets.ctfassets.net/kdr3qnns3kvk/5VuciSuo36Ac2eeyU02ywE/941f363e586c71cda688aca307b851fd/Thorpe-PACISE2017.pdf
(3.7.2021.)

2.1.5. Regularni izrazi

Pojam regularnih izraza predstavlja niz znakova koji definiraju obrazac pretraživanja. Neki od osnovnih regularnih izraza su:

- *. – odgovara bilo kojem znaku, osim novog retka*
- *\ w – podudarna riječ*
- *\ d – podudarni broj*
- *\ s – podudaranje s razmakom*
- *\ W – ne podudara se s riječju*
- *\ D – podudaranje nije znamenka*
- *\ S – podudaranje nije razmak*
- *[abc] – odgovara bilo kojem od a, b ili c*
- *[^ abc] – ne podudara se s a, b ili c*
- *[a-g] – podudaranje znaka između a i g* [17]

Korištenjem regularnih izraza se može primijeniti dodatno filtriranje nad tekstualnim podacima.

Regularni izrazi su pronašli svoju široku primjenu u računalnoj sigurnosti, ponajprije u analizi zapisnih datoteka kao i u provjeri korisničkih unosa u web forme. Također se mogu postaviti pravila za vatrozid i filtriranje prometa pomoću regularnih izraza te se mogu koristiti za skeniranje i identifikaciju virusa stvaranjem opisa koji odgovaraju određenim karakteristikama virusa[18]. U sintezi tehnika obrade prirodnog jezika i računalne sigurnosti, koriste se u gotovo svim aplikacijama za analizu izvornog koda, detektiranje spam e-mail poruka i prepoznavanje malicioznih URL linkova.

2.1.6. Vreća riječi (bag-of-words model)

Model vreće riječi je tehnika izdvajanja značajki u tekstu te opisuje pojavu svake riječi u tekstu. Proces izdvajanja značajki je potreban jer algoritmi strojnog učenja ne mogu izravno raditi sa sirovim tekstom već se tekstualni podaci moraju pretvoriti u vektore brojeva. Da bi se mogao

[17]Yordanov, V. *Introduction to Natural Language Processing for Text.* 2018.
URL: https://towardsdatascience.com/introduction-to-natural-language-processing-for-text-df845750fb63
(3.7.2021.)
[18]Li, V. *Regular Expressions: A Quick Intro for Security Professionals* 2020.
URL: https://dzone.com/articles/regular-expressions-a-quick-intro-for-security-pro
(3.7.2021.)

koristiti ovaj model potrebno je osmisliti rječnik poznatih riječi (tokena) te odabrati mjeru prisutnosti poznatih riječi. Razlog zašto se ovaj model naziva „vrećom" je jer se sve informacije o redoslijedu i strukturi riječi odbacuju te je fokus isključivo na izlaznom rezultatu koji odgovara na pitanje javlja li se određena riječ u dokumentu ili ne. Na pitanje gdje se riječ nalazi u dokumentu ovaj model ne odgovara. Za ovaj model je jako važno izraditi odgovarajući rječnik, čije je podatke, nakon stvaranja, potrebno dodatno obraditi tehnikama čišćenja teksta kao što su upotreba stop-riječi, lematizacija, ignoriranje (uklanjanje) interpunkcije[19].

U računalnoj sigurnosti model vreće riječi se uglavnom koristi za detektiranje spam e-mail poruka, detektiranje računalnih događaja na mreži kao i automatsku klasifikaciju prijavljenih sigurnosnih propusta.

2.1.7. TF-IDF

Pojam TF-IDF (frekvencija termina-inverzna frekvencija dokumenta) se može definirati kao kratica za frekvenciju termina i inverznu frekvenciju dokumenta, a to je statistička mjera koja se koristi za procjenu važnosti riječi za dokument u zbirci ili korpusu[20]. Procjena važnosti riječi za dokument u zbirci ili korpusu se postiže množenjem dvije metrike: koliko se puta riječ pojavljuje u dokumentu i obrnuta učestalost riječi u skupu dokumenata. TF-IDF svoju primjenu nalazi u automatiziranoj analizi teksta te je također korisna metrika za bodovanje riječi za algoritme strojnog učenja za obradu prirodnog jezika, što znači da svoju primjenu nalazi i u izgradnji modela za prepoznavanje spam e-mail poruka[21]. Problem kojem TF-IDF pristupa i kojeg rješava je problem s bodovanjem učestalosti riječi koji se očituje u tomu što najčešće riječi u dokumentu počinju imati najviše bodova. Što znači da takve riječi ne sadrže veliku informacijsku vrijednost za model u usporedbi s nekim rjeđim riječima. Formula glasi:

[19] Yordanov, V. *Introduction to Natural Language Processing for Text.* 2018.
URL: https://towardsdatascience.com/introduction-to-natural-language-processing-for-text-df845750fb63
(3.7.2021.)
[20] Yordanov, V. *Introduction to Natural Language Processing for Text.* 2018.
URL: https://towardsdatascience.com/introduction-to-natural-language-processing-for-text-df845750fb63
(3.7.2021.)
[21]Stecanella, B. *What Is TF-IDF?* 2019.
URL: https://monkeylearn.com/blog/what-is-tf-idf/
(3.7.2021.)

$$W_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

$W_{x,y}$ – pojam x unutar dokumenta y

$tf_{x,y}$ – frekvencija pojma x unutar dokumenta y

df_x – broj dokumenata koji sadrže pojam x

N – ukupan broj dokumenata

Ako se formula podijeli na više dijelova dobije se:

- Frekvencija pojma (*engl. Term Frequency*) (TF): bodovanje učestalosti riječi u trenutnom dokumentu:

$$TF(pojam) = \frac{broj\ pojava\ pojma\ u\ dokumentu}{ukupan\ broj\ predmeta\ u\ dokumentu}$$

- Inverzna frekvencija dokumenta (*engl. Inverse Document Frequency)* (IDF): bodovanje koliko je riječ rijetka u dokumentima

$$IDF(pojam) = \log\left(\frac{ukupan\ broj\ dokumenata}{broj\ dokumenata\ koji\ sadrže\ pojam}\right)$$

- Prethodno navedene formule se mogu upotrijebiti za izračunavanje TF-IDF rezultata:

$$TFIDF(pojam) = TF(pojam) \times IDF(pojam)$$[22]

Primjena principa TF-IDF se može izvesti pomoću Python koda i različitih programskih proširenja. Primjena u računalnoj sigurnosti seže od detektiranja virusa, malicioznih URL-ova, spam i prevarantskih e-mail poruka, do analize i grupiranja sigurnosnih incidenata, s napomenom da se najčešće koristi kao vektorizator za promjenu iz abecednih u numeričke vrijednosti. Vektorizator je funkcija koja pretvara tekst u vektore da bi se izbjegao rad nad pojedinačnim vrijednostima posebno, umjesto čega se radi na skupu vrijednosti (vektor) odjednom.

[22]Yordanov, V. *Introduction to Natural Language Processing for Text.* 2018.
URL: https://towardsdatascience.com/introduction-to-natural-language-processing-for-text-df845750fb63
(3.7.2021.)

3. Računalna sigurnost na web-u i tekstualni podaci

Računalna sigurnost na web-u je povezana sa svim aspektima računalne sigurnosti općenito. Sigurnost operativnih sustava, sustava u oblaku te sigurnost web aplikacija, kao i aplikacija za mobilne uređaje je povezana s tekstualnim podacima pomoću kojih su aplikacije izgrađene, koriste se i unaprjeđuju. Zbog toga je potrebno definirati što je to računalna sigurnost, koja sva polja obuhvaća i utjecaj tekstualnih podataka na sigurnosne incidente i načine obrade korisničkih unosa u web aplikacijama.

3.1. Definicija računalne sigurnosti

„Računalna sigurnost je zbirka alata, politika, sigurnosnih koncepata, sigurnosnih zaštitnih mjera, smjernica, pristupa upravljanju rizicima, radnji, obuke, najboljih praksi, osiguranja i tehnologija koje se mogu koristiti za zaštitu računalnog okruženja organizacije i imovine korisnika. Imovina organizacije i korisnika uključuje povezane računarske uređaje, osoblje, infrastrukturu, aplikacije, usluge, telekomunikacijske sustave i ukupnost prenesenih i/ili pohranjenih informacija u računalnom okruženju. Računalna sigurnost nastoji osigurati postizanje i održavanje sigurnosnih svojstava organizacije i imovine korisnika protiv relevantnih sigurnosnih rizika u računalnom okruženju. Opći sigurnosni ciljevi uključuju dostupnost, integritet, koji može uključivati autentičnost te povjerljivost“[23].

Iz prethodno navedene definicije se može zaključiti da su principi računalne sigurnosti neizostavna stavka svakodnevnog korištenja računalnim komponentama i uslugama zbog čega se osmišljavaju razne strategije u domeni računalne sigurnosti za zaštitu korisnika i organizacija. U računalnoj sigurnosti se može govoriti o dva tipa primjene tehnika računalne sigurnosti, protumjerama i odgovorima na sigurnosne incidente. Pojam protumjera se može definirati kao skup strategija za stvaranje slojeva zaštite za obranu od računalnog kriminala, uključujući računalne napade koji pokušavaju pristupiti podacima korisnika ili organizacije, promijeniti ih ili uništiti. Protumjere se mogu podijeliti s obzirom na tip infrastrukture koja se štiti:[24]

- sigurnost kritične infrastrukture – prakse zaštite računalnih sustava, mreža i druge imovine na koju se društvo oslanja;

[23]ITU-T, *Definition of cybersecurity*. 2008.
URL: https://www.itu.int/en/ITU-T/studygroups/com17/Pages/cybersecurity.aspx
(3.7.2021.)
[24]IBM, *What is cybersecurity.* 2021.
URL: https://www.ibm.com/topics/cybersecurity
(3.7.2021.)

- sigurnost mreže – sigurnosne mjere za zaštitu računalne mreže od uljeza, uključujući žičane i bežične (Wi-Fi) veze;
- sigurnost aplikacija – procesi koji pomažu u zaštiti aplikacija koje rade lokalno i u oblaku. Sigurnost bi se trebala ugrađivati u aplikacije u fazi dizajniranja, uzimajući u obzir način na koji se postupa s podacima;
- sigurnost u oblaku – posebno povjerljivo računanje koje šifrira podatke u oblaku u stanju mirovanja (u pohrani), u pokretu (dok putuje do, iz i unutar oblaka) i u upotrebi (tijekom obrade) za podršku privatnosti kupaca, poslovnim zahtjevima i usklađenosti s propisima standardima;
- sigurnost podataka – mjere zaštite podataka, poput Opće uredbe o zaštiti podataka ili GDPR-a, koje štite podatke od neovlaštenog pristupa, izlaganja ili krađe;
- obrazovanje krajnjih korisnika – izgradnja svijesti o sigurnosti u cijeloj organizaciji radi jačanja sigurnosti krajnjih točaka što bi moglo uključivati obučavanje korisnika za prepoznavanje sumnjivih privitaka e-mail poruka;
- oporavak od računalnog proboja – alati i postupci za reagiranje na neplanirane događaje računalnih incidenata.

3.2. Tekstualni podaci u računalnoj sigurnosti web aplikacija

Kao što je ranije navedeno, zaštita web aplikacija spada u protumjere računalne sigurnosti. Sigurnost web aplikacija se odnosi na izgradnju web stranica koje bi imale ugrađene sigurnosne kontrole kako bi se zaštitile od mogućih napada. Jedna od takvih kontrola su liste poželjnih ili nepoželjnih (*engl. whitelist/blacklist*) nizova znakova koji se mogu unijeti u web forme aplikacije. Izgradnja ovakve liste i njena implementacija osiguravaju web aplikaciju od poznatih napada, pogotovo napada sa klijentske strane, poput „*Cross-site scripting*“ (XSS) napada, gdje se različiti dijelovi JavaScript koda mogu unijeti u web formu aplikacije i izazvati proboj računalne sigurnosti. Također, ovakve liste štite i od napada na baze podataka web aplikacije i neautorizirani pristup podacima iz baze, poput „*SQL Injection*“ napada. Liste mogu sadržavati i neželjene IP adrese, kojima se pomoću regularnih izraza zabranjuje/onemogućuje pristup određenoj web aplikaciji, kao i listu obrazaca koji se definiraju kao lažni korisnici (*engl. bots*). Najpoznatija takva lista koju koristi većina web stranica je Međunarodna lista paukova i

botova IAB/ABC (*engl. IAB/ABC International Spiders and Bots List*)[25]. O raznim vrstama napada na web aplikacije, kao i o OWASP Top 10 će se govoriti u narednim poglavljima.

Tekstualni podaci u računalnoj sigurnosti, osim kodova i listi, se također odnose na uporabu prirodnog jezika na webu. Skoro pa svaka web aplikacija je izgrađena za primjenu korisničkih unosa. Taj unos je uglavnom baziran na govornom jeziku korisnika. Prirodni jezik se može klasificirati u svrhe računalne sigurnosti da bi se prepoznala računalna prijevara na osnovu principa takve komunikacije. Na primjer, prevarantske e-mail poruke će se većinom korisniku obratiti sa „Dragi korisniče“ (*engl. dear user, dear sir, dear madam*) zbog toga što napadači nemaju pristup bazi podataka stvarnih korisnika neke web aplikacije pa samim time ne mogu pristupiti bazi podataka koja sadrži imena, prezimena i korisnička imena korisnika aplikacije. Na osnovu takvih i sličnih riječi (koje su previše općenite, ukazuju na hitnost bez imenovanja korisnika kojem se hitnost ukazuje) se mogu klasificirati e-mail poruke na zlonamjerne i uredne. Većina prikupljanja osjetljivih podataka o korisnicima su tekstualni podaci (poruke, ime, prezime, korisničko ime, lozinke) te je zbog toga očito da je nad takvim podacima primjenjiv bar dio tehnika obrade prirodnog jezika, ako ne i izravno (u slučaju kad se ne definiraju riječi i smislene rečenice), onda neizravno upotrebom nekih od tehnika za pročišćavanje tekstualnih podataka.

3.3. Računalna sigurnost na webu i OWASP Top 10

OWASP je kratica za Open Web Application Security Project, internetsku zajednicu koja proizvodi članke, metodologije, dokumentaciju, alate i tehnologije na polju sigurnosti web aplikacija. OWASP Top 10 je lista 10 najčešćih ranjivosti aplikacija. Također pokazuje njihove rizike, utjecaje i protumjere koje se mogu primijeniti kako bi se takve ranjivosti uspješno izbjegle. Lista se ažurira svake tri do četiri godine. Top 10 OWASP ranjivosti u 2021. godini su:[26]

- injekcija – SQL Injekcija vrsta je sigurnosnog napada u kojem zlonamjerni napadač umetne ili ubrizga upit putem ulaznih podataka (često jednostavnim ispunjavanjem obrasca na web mjestu) s klijentske strane na poslužitelj. Ako upit bude prihvaćen,

[25]IAB/ABC, *International Spiders and Bots List*, URL: https://www.iab.com/guidelines/iab-abc-international-spiders-bots-list/ (3.7.2021.)
[26]Open Web Application Security Project, *OWASP Top Ten – 2017 The Ten Most Critical Web Application Security Risks.* 2017.
URL: https://owasp.org/www-pdf-archive/OWASP_Top_10-2017_%28en%29.pdf.pdf
(3.7.2021.)

napadač može čitati podatke iz baze podataka, dodavati nove podatke, ažurirati podatke, brisati neke podatke prisutne u bazi podataka, izdavati naredbe administratora za izvršavanje privilegiranih zadataka baze podataka;

- prekinuta autentifikacija – slučaj u kojem je sustav provjere autentičnosti web aplikacije neispravan i može rezultirati nizom sigurnosnih prijetnji. To je moguće ako protivnik izvrši napad grubom silom (*engl. brute force*) da bi se prerušio u korisnika, dopuštajući korisnicima da koriste slabe lozinke koje su ili riječi iz rječnika ili uobičajene lozinke poput „12345678“, „lozinka“ i slično;
- izloženost osjetljivih podataka – podrazumijeva da se pohranjeni osjetljivi podaci propuštaju zlonamjernim napadačima. Ti podaci mogu uključivati osobne podatke kao što su ime, adresa, spol, datum rođenja, osobni identifikacijski brojevi, financijski podaci poput broja računa, brojeva kreditnih kartica i zdravstvenih podataka;
- XML vanjski entiteti (XXE) – podrazumijeva vrstu ranjivosti web aplikacije koja raščlanjuje XML ulazne podatke. Izvodi se kada se ulazni podaci u obliku XML-a odnose na vanjski entitet, ali ga obrađuje slabi XML parser, što može uzrokovati krivotvorenje zahtjeva na strani poslužitelja kao i otkrivanje osjetljivih podataka;
- neispravna kontrola pristupa – može rezultirati nenamjernim curenjem informacija, mijenjanjem detalja drugih korisničkih računa, manipulacijom metapodacima, neovlaštenim pristupom API-ju. Pravilno postavljena kontrola pristupa određuje ograničenja ili granice u kojima korisnik smije raditi. Na primjer, administratorske (*engl. root*) privilegije obično se daju administratoru, a ne stvarnim korisnicima;
- pogrešne konfiguracije sigurnosti – napadaču obično daju puni pristup sustavu, što rezultira komprimiranjem sustava. Web aplikacija mogla bi biti ranjiva na takve napade ako ima slabo konfigurirana dopuštenja za usluge u oblaku;
- Cross-site skriptiranje (XSS) – ove vrste napada se događaju kada se zlonamjerna skripta ubrizga putem web aplikacije (uglavnom sa strane preglednika) i pošalje se drugom korisniku koji ne može znati da kod nije dio web stranice i zbog toga se skripta izvršava;
- nesigurna deserializacija – serijalizacija u web aplikacijama obično se koristi za baze podataka. Ako web aplikacija deserijalizira neovlaštene predmete koje opskrbljuje napadač, aplikacija postaje ranjiva na napad. Ako je napad uspješan, napadač će moći izvršiti daljinsko izvršavanje koda (*engl. remote code execution*);

- korištenje komponenata s poznatim ranjivostima – ako se u razvoju web aplikacije koristi komponenta koja je sama po sebi ranjiva na prijetnje zbog neispravnog koda tada će i web aplikacija biti propusna na različite vrste računalnih prijetnji. Ova vrsta propusta se najčešće događa kada se koriste stare verzije komponenata ili ugniježđene vrijednosti;
- nedovoljno bilježenje i nadzor – se odnosi na bilježenje sigurnosnih incidenata te ako se stvore datotečni zapisi svih poznatih incidenata u određenoj organizaciji, što će omogućiti pravovremenu reakciju na buduće sigurnosne propuste.

Svaka od navedenih ranjivosti se bazira na tekstualnim podacima, neke od njih sadrže i prirodni govorni jezik, a neke su niz znakova koda. U sljedećim poglavljima ovog rada će se obraditi i klasifikacija dijelova izvornog koda koji uzrokuje XSS stoga je tu vrstu računalne ranjivosti potrebno dodatno objasniti.

3.3.1. Cross-site skriptiranje (XSS)

Cross-site skriptiranje (poznato i kao XSS) je ranjivost web sigurnosti koja napadaču omogućuje da ugrozi interakcije koje korisnici imaju s nekom web aplikacijom. Omogućuje napadaču da zaobiđe politiku podrijetla (*engl. origin policy*) koja je stvorena za međusobno odvajanje različitih web stranica. Ranjivosti skriptiranja na više web lokacija omogućavaju napadaču da se pretvori u korisnika koji je žrtva takvog napada te da izvrši sve radnje koje korisnik inače može izvršiti[27]. Ako je korisnik žrtva imao privilegirani (administratorski) pristup unutar aplikacije, tada bi napadač mogao steći potpunu kontrolu nad svim funkcijama i podacima aplikacije[28].

Postoje tri glavne vrste XSS napada[29]:

- *odraženi XSS (engl. Reflected XSS), gdje zlonamjerna skripta dolazi iz trenutnog HTTP zahtjeva;*

[27]PortSwigger. *Cross-site scripting.* 2021.
URL: https://portswigger.net/web-security/cross-site-scripting
(3.7.2021.)
[28]Kirsten, S. *Cross-site scripting (XSS).* 2021.
URL: https://owasp.org/www-community/attacks/xss
(3.7.2021.)
[29]PortSwigger. *Cross-site scripting.* 2021.
URL: https://portswigger.net/web-security/cross-site-scripting
(3.7.2021.)

- *pohranjen XSS (engl. Stored XSS), gdje zlonamjerna skripta dolazi iz baze podataka web mjesta;*
- *DOM-utemeljeni XSS (DOM-based XSS), gdje ranjivost postoji u kodu na strani klijenta, a ne u kodu na strani poslužitelja*.

S obzirom na to da će se u ovom radu obraditi analiza koda (u smislu klasifikacije) uzimajući u obzir ranjivost pohranjenog XSS-a, potrebno je detaljnije objasniti princip funkcioniranja pohranjenog XSS-a.

Pohranjeni XSS (poznat i kao trajni ili XSS drugog reda) nastaje kada aplikacija prima podatke iz nepouzdanog izvora i uključuje te podatke u svoje HTTP odgovore. Takvi podaci su tekstualni podaci koji se uglavnom unose putem HTTP zahtjeva (poput komentara na društvenim mrežama, blogovima ili forumima). Podaci također mogu stići iz drugih nepouzdanih izvora kao što su aplikacije za web poštu koje prikazuju poruke primljene putem SMTP-a ili aplikacija za nadgledanje mreže koja prikazuje paketne podatke iz mrežnog prometa[30].

Jedan od najjednostavnijih primjera XSS ranjivosti je upotreba aplikacije oglasne ploče na kojoj se korisnicima omogućuje slanje poruka (notifikacija) koje se prikazuju svim drugim korisnicima, te bi uređivanje i slanje takvih tekstualnih podataka izgledalo ovako:[31]

 Pozdrav, ovo je moja poruka! ,

gdje se između oznaka , koje označavaju paragraf teksta, unose željeni tekstualni podaci koji se zatim objavljuju. Ako aplikacija ne vrši nikakvu drugu obradu podataka (primjerice ako se ne koriste sigurnosne crne i bijele liste), napadač se lako može ulogirati kao obični korisnik i poslati poruku koja napada druge korisnike:

 <script>alert('zlonamjernaskripta');</script>

Prethodno navedena skripta radi po principu unosa malicioznih JavaScript kodova u web forme predviđene za unos običnog teksta (poput komentara, objava) od strane korisnika. Ako web

[30]Kirsten, S. *Cross-site scripting (XSS).* 2021.
URL: https://owasp.org/www-community/attacks/xss
(3.7.2021.)
[31]PortSwigger. *Cross-site scripting.* 2021.
URL: https://portswigger.net/web-security/cross-site-scripting
(3.7.2021.)

aplikacija ne provjerava i ne sanira unose korisnika, skripta se pokrene na web stranici te se prikaže ostalim korisnicima. Stoga bi svaka aplikacija trebala koristiti kodiranje korisničkih unosa, kao i crnu listu u kojoj bi se nalazili izrazi i znakovi poput: „<script>“, „„“, „ /“, „*“, „'“.

Ručno testiranje za prepoznavanje pohranjenih XSS ranjivosti je zahtjevno jer se moraju testirati sve relevantne ulazne točke preko kojih podaci ulaze u obradu aplikacije te sve izlazne točke gdje bi se ti podaci mogli pojaviti u odgovorima aplikacije. Ulazne točke uključuju parametre ili druge podatke unutar URL-a upita i tijela poruke. Rute za napad koje postoje ovise o funkcionalnosti koju aplikacija primjenjuje: aplikacija za web poštu obrađivat će podatke primljene u e-pošti; aplikacija koja prikazuje naslovnicu neke društvene mreže će obrađivati podatke sadržane u objavama korisnika, web portali će prikazivati komentare svojih korisnika itd. Izlazne točke za pohranjene XSS napade su svi mogući HTTP odgovori koji se vraćaju bilo kojoj vrsti aplikacije u bilo kojoj situaciji. Prvi korak u testiranju pohranjenih XSS ranjivosti je lociranje veza između ulazne i izlazne točke, pri čemu se podaci predani ulaznoj točki emitiraju s izlazne točke. Međutim, takav način testiranja može biti izazov zbog toga što podaci predani u bilo kojoj ulaznoj točki bi se mogli emitirati s bilo koje izlazne točke. Na primjer, neka korisnička imena ili unosi bi se mogli pojaviti kao vidljivi samo nekim korisnicima aplikacije – onima koji imaju pristup događajima i promjenama unutar aplikacije (*engl. audit-log*) pa bi zbog toga bilo moguće da takav XSS napad ostane neotkriven. Sveobuhvatna identifikacija veza između ulaznih i izlaznih točaka uključivala bi testiranje svake permutacije zasebno, što bi značilo da bi se određena vrijednost trebala postaviti u ulaznu točku te pratiti njeno kretanje do izlazne točke. Zatim bi se trebalo utvrditi pojavljuje li se navedena vrijednost u izlaznoj točki[32]. Takav način je znatno nepraktičan zbog broja mogućih izlaznih točaka na samo jednu ulaznu točku te nije moguće praktično i brzo na taj način testirati web aplikacije s više stranica. Umjesto toga, može se testirati kroz točke unosa podataka, podnoseći određenu vrijednost u svaku i nadgledajući odgovore aplikacije kako bi se otkrili slučajevi u kojima se prijavljena vrijednost pojavljuje[33]. Osim unosa podataka, ova vrsta računalne ranjivosti se može testirati i pomoću statične analize koda, gdje se obraća pozornost na korištenje već poznati nesigurnih funkcija i formi u izvornom kodu. Ovaj proces se može i automatizirati (do određene mjere i s obzirom na veličinu i dostupnost skupa podataka).

[32]PortSwigger. *Stored XSS.* 2021.
URL: https://portswigger.net/web-security/cross-site-scripting/stored
(3.7.2021.)
[33]PortSwigger. *Stored XSS.* 2021.
URL: https://portswigger.net/web-security/cross-site-scripting/stored (3.7.2021.)
(3.7.2021.)

4. Klasifikacija malicioznih domena pomoću tehnika obrade prirodnog jezika

Po zadnjim statistikama o digitalnoj populaciji svijeta može se vidjeti da trenutno 59,5% svjetske populacije koristi internet[34]. S mnogim brojem internet stranica, običnom korisniku je teško raspoznati legitimne URL-ove od onih zlonamjernih. Stoga će se u ovom poglavlju istražiti načini na koje bi se URL-ovi mogli klasificirati.

4.1. Anatomija URL-a

URL je skraćenica za jedinstveni lokator resursa (*engl. Uniform Resource Locator*). URL je u osnovi adresa web stranice koja se nalazi na WWW-u. URL-ovi se obično prikazuju u adresnoj traci web preglednika[35]. URL može sadržavati ili protokol za prijenos hiperteksta (HTTP) ili zaštićeni protokol za prijenos hiperteksta (HTTPS). Ostale vrste protokola uključuju protokol za prijenos datoteka (FTP), jednostavni protokol za prijenos pošte (SMTP).

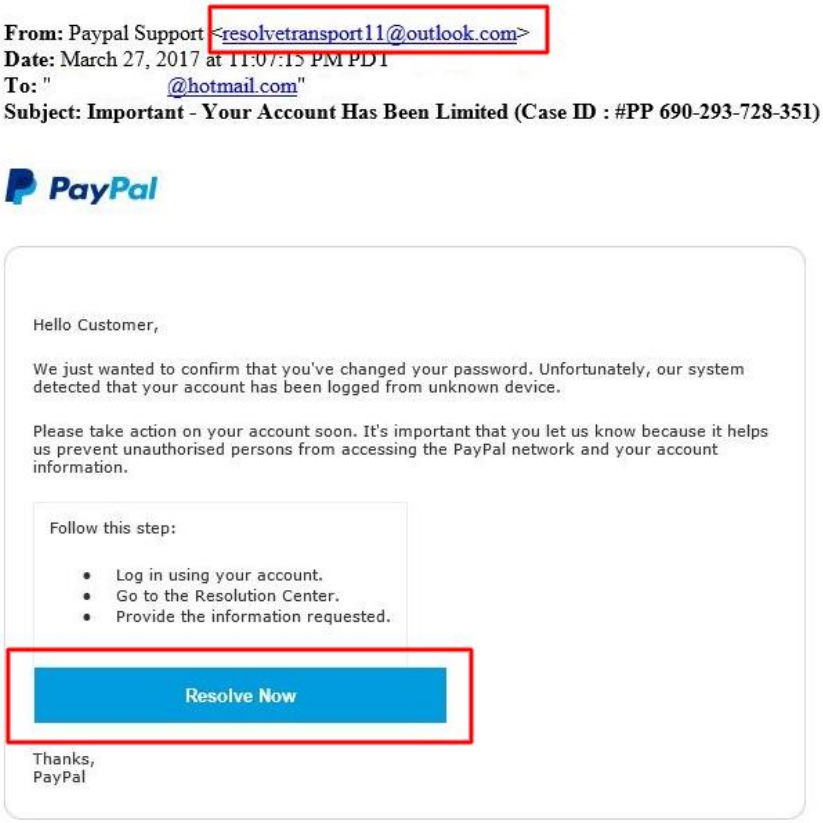
4.2. Tipovi malicioznih URL-ova

Zlonamjerni URL-ovi vode korisnike do loših ili problematičnih web stranica koje uglavnom pokušavaju ukrasti osobne podatke korisnika, poput OIB-a, broja bankovne kartice, vjerodajnica i sl. Obrana od takvih URL-ova se najčešće vrši preko antivirusnih programa u kojima se maliciozni URL-ovi nalaze u takozvanim crnim listama te se na taj način korisniku onemogućuje pristup istim, ili se prikaže upozorenje da web stranica nije sigurna. Postoji nekoliko glavnih tipova malicioznih URL-ova koji se koriste za klasifikaciju i grupiranje u crne liste[36]. Osnovni tipovi su:[37]

[34]Statista, *Global digital population as of January 2021.* 2021.
URL: https://www.statista.com/statistics/617136/digital-population-worldwide/
(3.7.2021.)
[35]Cross, M. *Uniform Resource Locator.* 2014.
https://www.sciencedirect.com/topics/computer-science/uniform-resource-locator
(3.7.2021.)
[36]Palo Alto Networks Inc., *Malicious URL Categories.* 2021.
URL: https://docs.paloaltonetworks.com/pan-os/9-1/pan-os-admin/url-filtering/url-categories/url-category-best-practices.html
(3.7.2021.)
[37]Palo Alto Networks Inc., *Malicious URL Categories.* 2021.
URL: https://docs.paloaltonetworks.com/pan-os/9-1/pan-os-admin/url-filtering/url-categories/url-category-best-practices.html (3.7.2021)

- URL-ovi za preuzimanje putem pokreta – su URL-ovi koji promiču nenamjerno preuzimanje softvera s web stranica. Preuzimanje se uglavnom pokrene pri prvom pristupu na takvu web stranicu ili samim pomakom miša po web stranici (bez korisničke interakcije ili klika). Ovakvi URL-ovi uglavnom započnu preuzimanje štetnih softvera na korisničko računalo[38];
- *Command and Control* URL-ovi – su URL-ovi koji su povezani sa zlonamjernim softverom koji povezuje ciljno računalo s poslužiteljem za upravljanje. Oni se razlikuju od URL-ova koji se mogu kategorizirati kao zlonamjerni, jer ne moraju uvijek biti povezani i upravljani virusom, ali većinom jesu[39];
- URL-ovi za krađu identiteta (*engl. phishing URLs*) – predstavljaju oblik URL-ova koji vrše napad na korisnika i pritom kradu osjetljive korisničke podatke. Često se URL maskira kao legitiman te se dalje prenosi ili šalje korisnicima preko e-mail poruka. Način na koji se lažni URL maskira je uglavnom izostavljanje par znakova u URL-u, koje korisnik neće primijetiti[40] (primjerice umjesto facebook.com, napadači će napraviti adresu sličnu originalnoj koja bi bila facbook.com). Međutim većina URL-ova za krađu identiteta se može lako prepoznati samim pogledom na URL.

[38]Kaspersky, *What Is a Drive by Download*. 2021.
URL: https://www.kaspersky.com/resource-center/definitions/drive-by-download
(3.7.2021.)
[39]Palo Alto Networks Inc., *Command and Control Explained*
URL: https://www.paloaltonetworks.com/cyberpedia/command-and-control-explained
(3.7.2021.)
[40]Barry, C. *Email threat types: URL phishing.* 2021.
URL: https://blog.barracuda.com/2021/02/03/email-threat-types-url-phishing/
(3.7.2021.)



Slika 1. Prikaz prevarantskog URL-a[41]

Po vizualnom sadržaju web stranice iz Slike 1., moglo bi se pretpostaviti da se radi o PayPal web formi, međutim provjerom URL-a je jasno da navedena web stranica nema URL koji PayPal službeno koristi.

4.3. Klasifikacija malicioznih URL-ova primjenom tehnika obrade prirodnog jezika

Podaci za klasifikaciju su preuzeti iz javnog skupa podataka sa Kaggle-a[42]. Skup podataka se sastoji od URL-ova koji su podijeljeni na "normalne" URL-ove i maliciozne URL-ove. Označeni su sa 0 i 1, gdje 0 predstavlja normalan URL, a 1 predstavlja maliciozan URL. Cijeli

[41]ITRC: Identity Theft Resource Center, *PAYPAL PHISHING E-MAIL.* 2017.
URL: https://www.idtheftcenter.org/images/Pay_pal.jpg
(3.7.2021.)
[42]Patel, D. *Malicious and Benign Websites. 2019.*
URL: https://www.kaggle.com/deepsworld/malicous-and-benign-websites
(3.7.2021.)

skup podataka je uređen i smanjen je broj URL-ova na 111 URL-ova, obraćajući pažnju na kategorije, zbog lakšeg i bržeg treniranja modela.

Budući da su podaci strukturirani i unaprijed označeni, dalje se može prijeći na izdvajanje određenih dijelova iz podataka. Primarno će se izdvojiti leksičke značajke temeljene na poslužitelju. Koristiti će se programski jezik Python te proširenja NumPy, re, pandas, NLTK (paket za obradu teksta i jezika – *Natural Language Toolkit*) i modeli iz sklearn paketa (za strojno učenje pomoću kojih će se stvarati modeli i predikcije). Python kod će se pisati u Jupyter Notebooks. Nakon učitavanja skup podataka u radnu knjižicu, prvo što je potrebno napraviti je pretvoriti CSV datoteku u podatkovni okvir pomoću pandas paketa da bi se moglo dalje obrađivati podatke. Zatim se podatkovni okvir čisti od duplikata u podacima te se stvaraju rječnici. Nakon procesa čišćenja i uređivanja podataka može se vidjeti prvih pet redova podatkovnog okvira (Slika 2.) te zatim slijedi obrada tekstualnih podataka pomoću tehnika obrade prirodnog jezika.

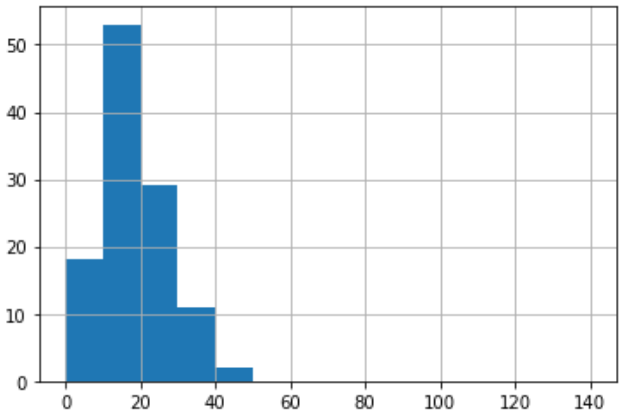
	label	url	category_id
0	0	https://www.drive.google.com	0
1	0	www.gmail.google.com	0
2	0	http://facebook.com	0
3	0	https://yahoo.com	0
4	1	001web.net	1

Slika 2. Prikaz podatkovnog okvira

Da bi se podaci mogli obraditi pomoću algoritama strojnog učenja potrebno ih je pretvoriti u vektore, zbog toga što algoritmi strojnog učenja rade isključivo s numeričkim značajkama[43].

URL-ovi u skupu podataka imaju različitu dužinu, te je potrebno predstaviti broj URL-ova za svaku različitu duljinu (Slika 3.).

[43]Pantola, P. *Natural Language Processing: Text Data Vectorization.* 2018.
URL: https://medium.com/@paritosh_30025/natural-language-processing-text-data-vectorization-af2520529cf7 (3.7.2021.)



Slika 3. Vizualizacija duljine niza znakova u skupu podataka

Iz prikazanog histograma može se uočiti da većina URL-ova ima duljinu manju od 20 znakova, što se može iskoristiti u kodu za tokenizaciju teksta i stvaranje modela „vreća riječi“. Funkcijom tokeniziranja se stvaraju tokeni, to jest nizovi znakova odvojeni po različitim posebnim znakovima (odvajaju se po: /, ., -) te se zatim uklanjaju nastavci domena, npr. „.com“. Već postojeći model će se poboljšati korištenjem TF-IDF čijom se primjenom smanjuje utjecaj nekih riječi. Iako učestalo u obradi teksta tehnikama obrade prirodnog jezika, u ovom slučaju se ne uklanjaju točke i posebni znakovi zbog toga što se upravo tim posebnim znakovima zlonamjerni URL-ovi služe da bi se naizgled činili kao legitimni, stoga je za model važno da se uključe u klasifikaciju. Nakon izrade modela, slijedi treniranje u kojem će se koristiti logistička regresija koja predstavlja klasifikacijski model. Logistička regresija je postupak modeliranja vjerojatnosti ishoda s obzirom na ulaznu varijablu te najčešće modelira binarni ishod[44].

Za potrebe izrade klasifikacijskog modela potrebno je podijeliti skup podataka na skup za treniranje i testni skup. Skupom za treniranje se poučava, tj. trenira model te se zatim testnim skupom procjenjuje točnost modela (Slika 4.).

[44]Edgar, Manz. *Research Methods for Cybersecurity*. 2017.
URL: https://www.sciencedirect.com/topics/computer-science/logistic-regression
(3.7.2021.)

```
točnost učenje = 1.0
točnost test = 0.6521739130434783
```

Slika 4. Prikaz rezultata modeliranja

Kao što se može vidjeti prema rezultatu na Slici 4., točnost je nešto iznad 50%, što nije idealan rezultat (poželjna je točnost od barem 70%). Razlog navedenom rezultatu je (vjerojatno) smanjivanje broja URL-ova u originalnom skupu podataka te se može voditi pretpostavkom da ako se poveća broj URL-ova u početnom skupu podataka, samim time se povećava i skup podataka za treniranje kojim se trenira model iz čega bi proizašao zaključak da bi se točnost povećala. To će se provjeriti tako što će se broj URL-ova sa 111 povećati na 1000 dodavajući URL-ove iz originalnog skupa podataka u smanjeni skup podataka (CSV datoteke).

```
točnost učenje = 0.9425
točnost test = 0.925
```

Slika 5. Prikaz rezultata modeliranja nad proširenim skupom podataka

Iz Slike 5., može se vidjeti da se točnost znatno povećava proširenjem skupa podataka. Model se zatim može primijeniti nad bilo kojim nizom znakova koji se unesu u listu koda (Slika 6.).

```
#primjena predviđanja na inputu unošenjem urlova u listu stringova:
X_predict = ['123movies.php','www.pltneki/profiles/myprofile.exe', 'https://www.google.net']
X_predict = vectorizer.transform(X_predict)
y_Predict = clf.predict(X_predict)
print(y_Predict)
```

```
[1 1 0]
```

Slika 6. Provjera točnosti modela unošenjem nove liste URL-ova

Rezultat primjene modela nad nekim nizom znakova se prikazuje pomoću nula i jedinica (0=normalna domena, 1=maliciozna domena). Iz primjera (Slika 6.) može se uočiti da je model osjetljiv na sumnjive ekstenzije (exe, php) kao i na previše numeričkih znamenki u URL-u te da https:// automatski prepoznaje kao sigurnu adresu.

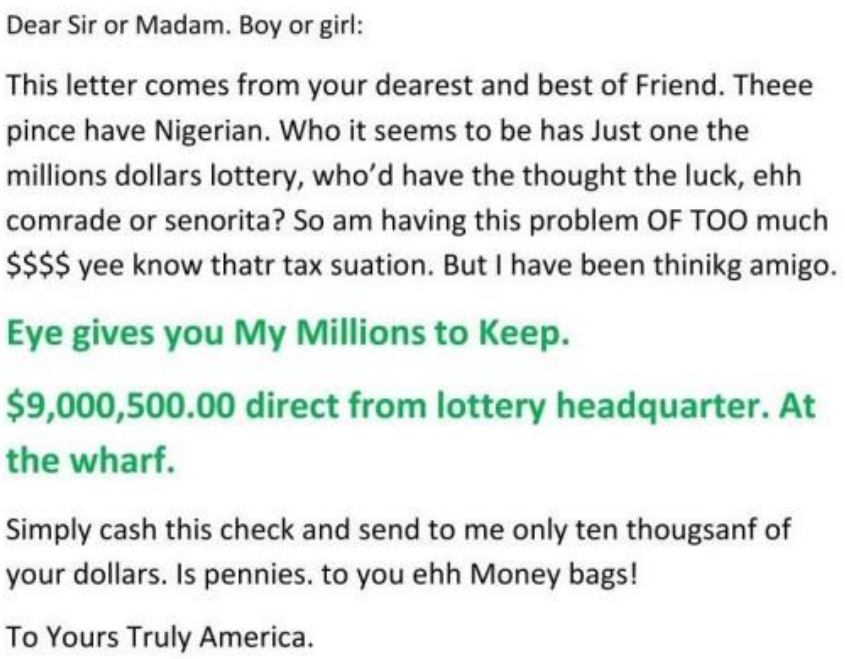
5. Detekcija prevarantskih e-mail poruka primjenom tehnika obrade prirodnog jezika

Phishing je jedna od najčešćih metoda računalnog kriminala, a odnosi se na lažne i prevarantske e-mail poruke u čijem se sadržaju napadač lažno predstavlja te često traži korisnika da uplati novac, pošalje osobne podatke ili se u tijelu poruke šalju poveznice na čiji klik se korisniku instalira virus na osobno računalo[45].

5.1. Sadržaj prevarantskih e-mail poruka

Prevarantske e-mail poruke se često mogu prepoznati po sadržaju teksta koji se nalazi u samim porukama. Često se može utvrditi da je e-mail poruka prevara ako sadrži neispravnu gramatiku i loš pravopis[46]. Međutim takve prevarantske poruke su često lako prepoznatljive i dio su nefiltriranog slanja poruke bilo kakvim korisnicima e-mail pošte (što znači da takve poruke nisu upućene zaposlenicima određene tvrtke ili ustanove, nego se šalju bilo kojim javno dostupnim e-mail adresama). Najpoznatiji primjer takve *phishing* kampanje je prevara nigerijskog princa (Slika 7.).

[45] Phishing.org, *What Is Phishing?*
URL: https://www.phishing.org/what-is-phishing
(3.7.2021.)
[46] NCSC.GOV.UK, *Phishing attacks: defending your organization.* 2021.
URL: https://www.ncsc.gov.uk/guidance/phishing
(3.7.2021.)


Dear Sir or Madam. Boy or girl:

This letter comes from your dearest and best of Friend. Theee pince have Nigerian. Who it seems to be has Just one the millions dollars lottery, who'd have the thought the luck, ehh comrade or senorita? So am having this problem OF TOO much $$$$ yee know thatr tax suation. But I have been thinikg amigo.

Eye gives you My Millions to Keep.

$9,000,500.00 direct from lottery headquarter. At the wharf.

Simply cash this check and send to me only ten thougsanf of your dollars. Is pennies. to you ehh Money bags!

To Yours Truly America.

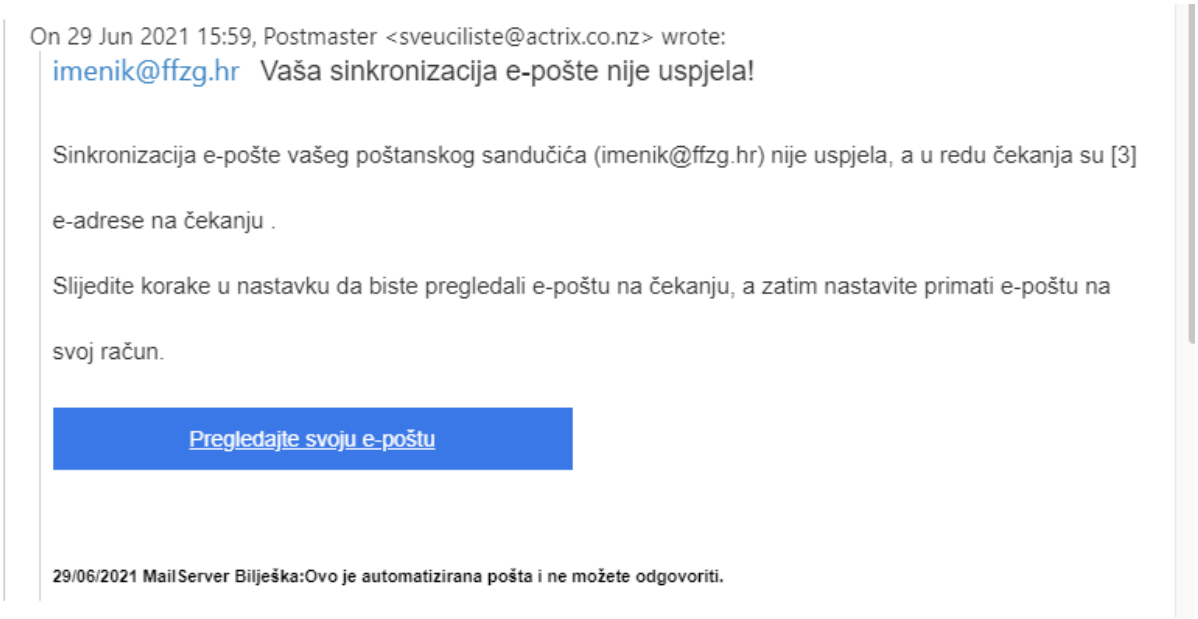
Slika 7. Primjer phishing e-mail poruke[47]

Kao što se može vidjeti u prethodnoj slici, gramatika i pravopis su loše konstruirani, a i koristi se „*Dear sir, madam, boy or girl*" što može biti pokazatelj da pošiljatelj nema informaciju o identitetu primatelja.

Osim očitih prevarantskih e-mail poruka, također postoje i one čiji je sadržaj nešto sofisticiraniji. U takvim porukama se pošiljatelj najčešće pretvara da je neka web usluga koju primatelj već koristi (poput online trgovina, online novčanika i sl.). Sadržaj takvih poruka je često uvjerljiv te je često istaknuta hitnost u poruci. Traže se osobni podaci primatelja e-mail poruke ili pak uplate novca te se također traži da korisnik klikne na link u poruci koji često vodi do malicioznih web stranica koje mogu instalirati viruse na operativni sustav korisnika ili izvršiti krađu identiteta. Najbolji način za uočavanje ovakvih prevarantskih poruka je obratiti pozornost na e-mail adresu pošiljatelja (s obzirom da će se naslov, ime pošiljatelja i sadržaj poruke činiti legitimnima). E-mail adresa takvog pošiljatelja je često sa javnih domena, poput „@gmail.com" te je potrebno imati na umu da online trgovine, tvrtke i banke nikada neće za svoju e-mail adresu koristiti adresu sa javne domene[48]. Također se znaju koristiti slične e-mail

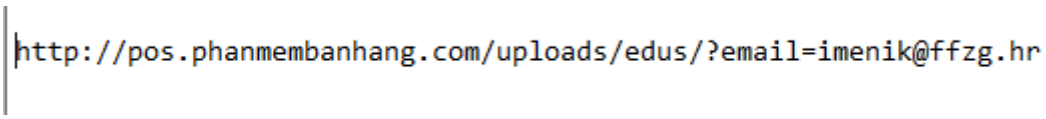
[47]Todorovic, A. *Nigerian Prince Scam Email*
URL: https://thepepsigeneration.com/nigerian-prince-scam-email/
(3.7.2021.)
[48]NCSC.GOV.UK, *Phishing attacks: defending your organization.* 2021.
URL: https://www.ncsc.gov.uk/guidance/phishing (3.7.2021.)

adrese, koje će imati par znakova viška ili manjka od one ispravne i izvorne. Primjer takve prevarantske e-mail poruke se može vidjeti na Slici 8.



Slika 8. Prikaz phishing e-mail poruke poslane djelatnicima i studentima Filozofskog fakulteta Sveučilišta u Zagrebu

Kao što se može vidjeti na ranijoj slici, sadržaj e-mail poruke se čini legitiman, gramatika je ispravna, a semantički gledano sadržaj ima smisla. Uočava se i da je e-mail poruka poslana korisnicima iz „imenik@ffzg.hr“. U sadržaju poruke se također nalazi veza na „Pregledajte svoju e-poštu“, prema čemu bi korisnik pretpostavio da će ga veza odvesti na pretinac e-mail pošte (iako se primatelj poruke već nalazi u pretincu e-pošte). Daljnjom inspekcijom veze, može se vidjeti da izgleda kao na Slika 9.



Slika 9. URL veze iz phishing e-mail poruke

Prva stavka koja se može uočiti je da URL ne odgovara pretincu e-pošte za ffzg.hr te da veza nije sigurna (koristi se http protokol umjesto https). Daljnjom inspekcijom e-mail poruke se također može uočiti da e-mail adresa pošiljatelja ne odgovara e-mail adresi domene ffzg.hr –

nego je domena „actrix.co.nz“, što nije ispravna i službena e-mail domena Filozofskog fakulteta Sveučilišta u Zagrebu.

5.2. Izrada modela za prepoznavanje *phishing* e-mail poruka primjenom tehnika obrade prirodnog jezika

Skup podataka koji će se obraditi, klasificirati i na temelju kojih će se izgraditi model za prepoznavanje su skupljeni iz autoričinih e-mail poruka. Podaci će se obraditi u programskom jeziku Python, koristeći NLTK, pandas, numpy, sklearn i worldcloud proširenja. Model će u obzir uzimati samo tekstualni sadržaj tijela e-mail poruke na temelju čega će se prepoznavati prevarantski sadržaji, što znači da se neće u obzir uzimati e-mail adrese pošiljatelja te će se moći primijeniti standardne tehnike obrade prirodnog jezika gdje se naglasak stavlja na riječi određenog jezika.

Za početak se učitava skup podataka te se pretvara u podatkovni okvir. Zatim se izlučuje osnovi izgled podataka koji su važni za daljnju obradu teksta. Iz Slike 10., može se vidjeti da se skup podataka sastoji od dvije za obradu važne kolone od kojih kolona „kategorija“ sadrži klasifikaciju pojedinačne poruke na „spam“ ili „ham“ po redovima (*spam* kao prevarantska poruka, *ham* kao poruka bezazlene komunikacije). Kolona „poruka“ se sastoji od sadržaja e-mail poruka, od kojih je svaka pojedinačna e-mail poruka zapisana u novom retku.

	kategorija	poruka
0	spam	Neseseri odabrala je vas! Dragi korisnice, ima...
1	spam	Pogodite tko se vratio? 50% popusta na artikle...
2	spam	Toplinski val ne posustaje, a nogometna grozni...
3	spam	Dragi korisniče, vaša pošiljka iz pošte ne mož...
4	spam	OBAVIJEST O PRIMJENI ČLANKA 54b. (Izuzeće od o...

Slika 10. Prikaz podatkovnog okvira inicijalnog skupa podataka

Zatim se počinje s primjenom osnovnih tehnika obrade prirodnog jezika kao što su uklanjanje posebnih i interpunkcijskih znakova, pretvaranje teksta poruke u mala slova te uklanjanje stop-riječi. Dodaje se i treća kolona (stupac) za praćenje duljine niza znakova u pojedinom retku.

Nakon obrade skupa podataka uklanjanjem interpunkcijskih znakova i pretvaranjem u mala slova, izgled skupa podataka se može vidjeti na Slici 11.

	kategorija	poruka	duljina
0	spam	neseseri odabrala je vas dragi korisnice imajt...	151
1	spam	pogodite tko se vratio 50 popusta na artikle k...	78
2	spam	toplinski val ne posustaje a nogometna groznic...	300
3	spam	dragi korisniče vaša pošiljka iz pošte ne može...	149
4	spam	obavijest o primjeni članka 54b izuzeće od obv...	153

Slika 11. Prikaz podatkovnog okvira nakon primjene tehnika obrade prirodnog jezika

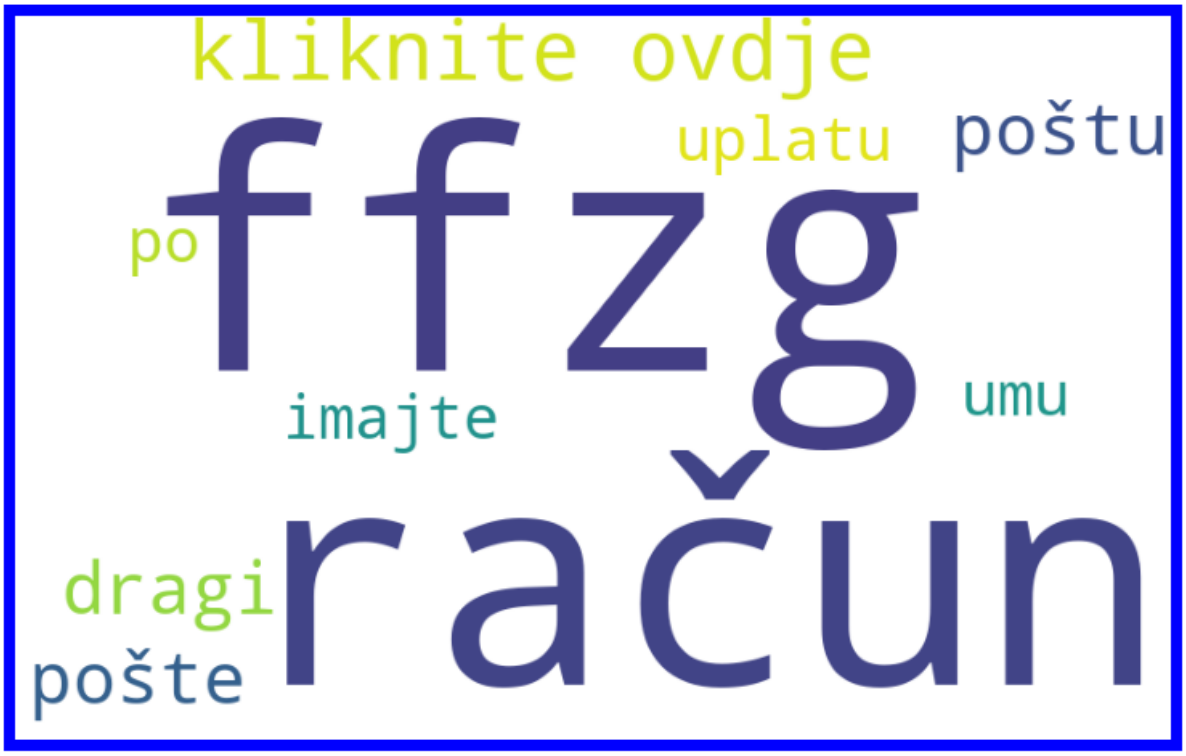
Zatim se uklanjaju stop-riječi. Međutim, kako u NLTK korpusu ne postoje predefinirane stop-riječi za hrvatski jezik, a da bi se demonstriralo dodavanje dodatnih stop-riječi u NLTK korpus (umjesto jednostavne obrade pomoću niza znakova i regularnih izraza), učitava se korpus stop-riječi za engleski jezik te se dodaje popis hrvatskih stop-riječi, preuzet sa Dabra[49]. Izgled skupa podataka nakon uklanjanja stop-riječi te nova duljina niza znakova nakon uklanjanja (stupac „cisto_dulj“) se može vidjeti na Slici 12.

	kategorija	poruka	duljina	cisto_dulj
0	spam	neseseri odabrala dragi korisnice imajte umu p...	151	126
1	spam	pogodite vratio 50 popusta artikle ponovno ima...	78	57
2	spam	toplinski val posustaje nogometna groznica uvi...	300	250
3	spam	dragi korisniče pošiljka pošte dostavljena upl...	149	108
4	spam	obavijest primjeni članka 54b izuzeće obveze i...	153	125

Slika 12. Prikaz podatkovnog okvira nakon uklanjanja stop-riječi

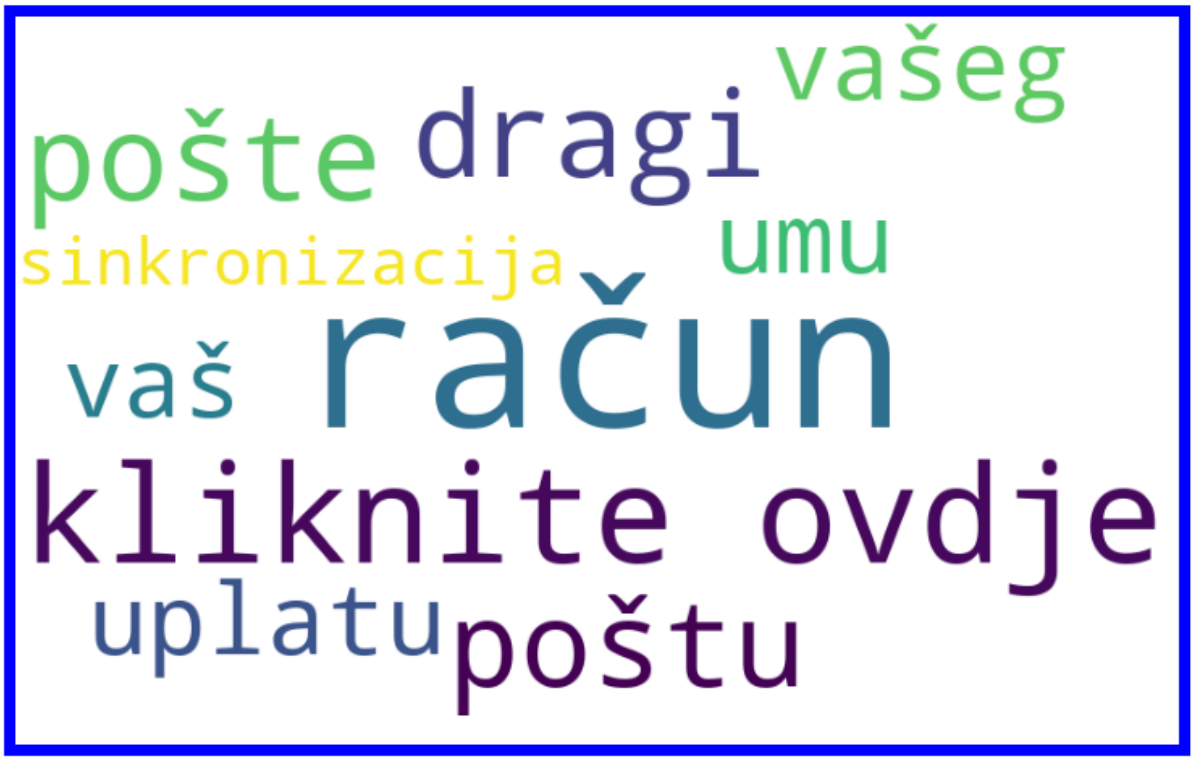
[49] Dabar, *Eliminacija stop-riječi u pretraživanju.* 2020.
URL: https://dabar.srce.hr/faq-page/254
(3.7.2021.)

Pomoću worldcloud-a se zatim mogu vizualizirati najčešće riječi poruka koje su označene kao „spam“ prevarantske poruke. Rezultat se može vidjeti na Slici 13.



Slika 13. Vizualizacija najčešćih riječi u porukama klasificiranima kao “spam”

Iz Slike 13. može se vidjeti da se na popisu nalaze neke očekivane riječi i fraze poput „kliknite ovdje“, „uplatu“, „račun“. Međutim zbog ograničenosti podataka (25% prevarantskih e-mail poruka u skupu podataka sadržavaju niz znakova „ffzg“), pojavljuje se i kratica „ffzg“ i „račun“ kao dvije najčešće riječi i indikatori prevarantske e-mail poruke, što u praksi nije pravi indikator prevarantskog e-maila. Stoga će se u listu stop-riječi naknadno uvrstiti „ffzg“, „po“ (stop-riječ u hrvatskom jeziku, previđena u inicijalnoj listi) i „imajte“ (riječ koja je česta bez obzira na vrstu poruke te ne može biti indikator prevarantske poruke). Rezultati se mogu vidjeti na Slici 14.



Slika 14. Vizualizacija najčešćih riječi u porukama klasificiranima kao “spam” nakon dodatnog uklanjanja novo-definiranih stop-riječi

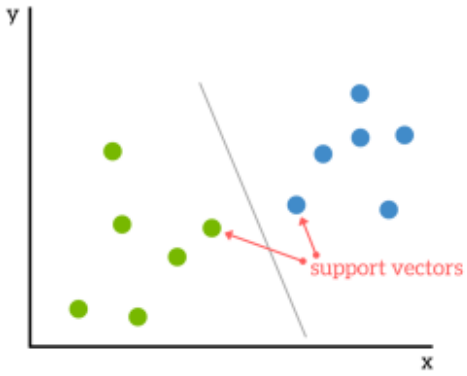
Nakon dodatnog uklanjanja stop-riječi dobiva se preciznija i semantički točnija vizualizacija najčešćih riječi u prevarantskim e-mail porukama. Umjesto najčešćih riječi prije uklanjanja, koje su bile „ffzg“, „račun“, „kliknite ovdje“, nakon dodatne obrade najčešće riječi su „račun“, „kliknite ovdje“, „uplatu“, što se znatno više preklapa s općenitim znanjem o sadržaju prevarantskih e-mail poruka (prevaranti često traže uplate na račun, uporabe hiperveza sa tekstom „kliknite ovdje“ te se šalju lažna upozorenja o kompromitiranom računu e-mail pošte).

Nakon vizualizacije najčešćih riječi u prevarantskim e-mail porukama, prelazi se na izgradnju modela. Najprije je potrebno, da bi se mogle primijeniti tehnike obrade prirodnog jezika na skupom podataka, pretvoriti abecedne poruke u numerički oblik pomoću TF-IDF vektorizatora. Zatim se poruke uklapaju i uzimaju se kao unos za model dok se kategorije (spam, ham) uzimaju kao izlazni rezultat.

Za postotak točnosti će se koristiti algoritam stroja potpornih vektora (*engl. Support Vector Machine, SVM*). Koristit će se linearni SVM klasifikator koji djeluje tako da napravi ravnu

liniju između dvije klase. To znači da će sve podatkovne točke na jednoj strani crte predstavljati jednu kategoriju, a podatkovne točke na drugoj strani crte stavit će se u drugu kategoriju[50].

SVM se temelji na ideji pronalaženja hiperravan koja dijeli skup podataka u dvije klase. Hiperravan je linija koja linearno odvaja i klasificira skup podataka, a što su podatkovne točke skupa podataka dalje od hiperravni to su ispravnije klasificirane. Potporni vektori predstavljaju podatkovne točke najbliže hiperravni te ako se uklone mogu promijeniti položaj same hiperravni. Zbog toga se smatraju značajnim elementima skupa podataka[51]. Na Slici 15., može se vidjeti prikaz hiperravni i potpornih vektora.



Slika 15. Hiperravan i potporni vektori[52]

Nakon podjele skupa podataka u testni skup i skup za treniranje, primjenjuje se SVM algoritam te se vrši predikcija nad testnim skupom gdje se dobiva točnost od 80% (Slika 16.).

[50]hr.ilusionity.com, *SVM Vodič za strojno učenje - Što je algoritam strojnog vektora podrške, objašnjen s primjerima*
URL:https://hr.ilusionity.com/638-svm-machine-learning-tutorial-what-is-the-support-vector-machine-algorithm-explained-with-code-examples
(6.7.2021.)
[51]Bambrick, N. *Support Vector Machines: A Simple Explanation.* 2016.
URL: https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html
(3.7.2021.)
[52]Bambrick, N. *Support Vector Machines: A Simple Explanation.* 2016.
URL: https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html
(3.7.2021.)

```
X_train,x_test,Y_train,y_test = train_test_split(X,y,random_state=8)


SVM.fit(X_train,Y_train)
y_pred = SVM.predict(x_test)

print ('Rezultat = > ', accuracy_score(y_test,y_pred))
```

```
Rezultat = >  0.8
```

Slika 16. Prikaz rezultata modeliranja

S navedenim rezultatima treba napomenuti da je inicijalni skup podataka poprilično ograničen. Sastoji se od manjeg broja e-mail poruka te bi se proširenjem skupa podataka, prvotno „ham“ sadržajem, a zatim i u manjoj potrebnoj količini „spam“ sadržajem, točnost znatno povećala zbog povećanog skupa podataka za treniranje modela.

6. Klasifikacija JavaScript izvornog koda s obzirom na „Cross-site scripting“ ranjivost

XSS (*engl. Cross-site scripting*) spada u jedne od najčešćih računalnih napada i sigurnosnih ranjivosti web aplikacija. Očituje se u web aplikacijama koje dopuštaju (ili ne saniraju) korisničke unose u web formama. Vrste ovakvih napada su objašnjene u ranijim dijelovima ovoga rada te ćemo se u ovom poglavlju koncentrirati na izradu modela i objašnjenje skupa podataka.

6.1. Skup podataka za izradu klasifikacijskog modela za XSS i obrada teksta primjenom tehnika obrade prirodnog jezika

Podaci za izradu modela su prikupljeni sa OWASP XSS CheatSheet web stranice[53]. Skup podataka sadrži dva tipa JavaScript kodova, one maliciozne i benigne kodove. Izvorni skup podataka se sastoji od dvije kolone, „kod“ i „label“. U „kod“ stupcu se nalazi niz znakova koji predstavlja JavaScript kod, a u „label“ kolumni se označavaju kodovi iz prve kolumne sa 0 i 1, gdje 0 predstavlja benigni kod, a 1 predstavlja maliciozni JavaScript kod. Nakon učitavanja skupa podataka u Jupyter Notebooks i pretvaranja u podatkovni okvir, izgled podataka se može vidjeti na Slici 17.

	Unnamed: 0	kod	label
0	0	<a href="/wiki/File:Socrates.png" class="i...	0
1	1	<tt onmouseover="alert(1)">test</tt>	1
2	2	\t Steeri...	0
3	3	\t <cite ...	0
4	4	\t . <a href="/wiki/Digital_object_iden...	0

Slika 17. Prikaz podatkovnog okvira

[53]OWASP Cheat Sheet Series, *Cross Site Scripting Prevention Cheat Sheet.* 2021.
URL: https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html (3.7.2021.)

Umjesto vektoriziranja pomoću TF-IDF, kao na ostalim primjerima u ovom radu, niz znakova iz kolone „kod“ će se pretvoriti u kodne točke znaka (*engl. character code points*), tj. ASCII kod ili kod drugog odgovarajućeg znakovnika. Nastavno na daljnju obradu tekstualnih podataka u koloni „kod“ neće se uklanjati posebni znakovi jer su upravo takvi nizovi znakova indikatori malicioznosti koda u većini slučajeva. Također je za detekciju malicioznosti koda važan odnos posebnih znakova i abecednog teksta. Zatim se preoblikuje niz podataka da bi se mogao nad njima upotrijebiti CNN (*engl. Convolutional Neural Network*) model (Slika 18.).

```
data = arr.reshape(arr.shape[0], 100, 100, 1)
```

```
data.shape
```

```
(1021, 100, 100, 1)
```

Slika 18. Prikaz koda kojim se preoblikuje niz podataka

6.2. Izrada modela za prepoznavanje „Cross-site scripting“ koda

Za izradu modela koristiti će se CNN ili konvolucijska neuronska mreža u Pythonu pomoću proširenja Keras. Konvolucijske neuronske mreže predstavljaju algoritam dubokog učenja koji može primiti neki unos, dodijeliti važnost različitim aspektima unosa (tekstualnim podacima) i biti u stanju razlikovati jednu vrstu unosa od druge[54]. U izgradnji modela koristi se sekvencijski model budući da će CNN biti linearni niz slojeva.

Nakon odabira modela i slojeva (Slika 19.), odabire se točnost kao izlazna metrika, kao optimizacijski algoritam se koristi adam[55], a kao funkcija gubitka se koristi binarna unakrsna entropija (za određivanje je li predikcija dobra ili ne). Zatim se podaci dijele u skup za treniranje i testni skup.

[54]Saha, S. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. 2018.
URL: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
(3.7.2021.)
[55] Kingma, Ba. *Adam: A Method for Stochastic Optimization*. 2014.
URL: https://arxiv.org/abs/1412.6980
(3.7.2021.)

```
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense, Activation, Conv2D, MaxPooling2D,Flatten,Dropout,MaxPool2D, BatchNormalization
```

Slika 19. Prikaz preuzimanja TensorFlow paketa, korištenja sekvencijskog modela i odabira slojeva

Treniranjem modela te izračunom matrice zabune (*engl. confusion matrix*) dobivaju se rezultati vidljivi na Slici 20., iz čega se može vidjeti da je točnost (*engl. accuracy*) modela 80%, preciznost (*engl. precision*) 82%, a odziv (*engl. recall*) 82%[56].

```
Accuracy : 0.8048780487804879
Precision : 0.8181818181818182
Recall : 0.8181818181818182
```

Slika 20. Prikaz točnosti, preciznosti i odziva modela nakon treniranja

Točnost modela predstavlja omjer ispravno klasificiranih podataka u odnosu na ukupan broj podataka, dok preciznost prikazuje omjer ispravno pozitivnih i zbroja ispravno pozitivnih i lažno pozitivnih podataka, a odziv se odnosi na omjer ispravno pozitivnih podataka i zbroja ispravno pozitivnih i lažno pozitivnih podataka[57].

Nakon treniranja modela, također se iteriranjem nad testnim podacima može uočiti broj točno predviđenih i krivo predviđenih niza znakova (Slika 21.).

[56]Arshaad, S. *Sentiment Analysis / Text Classification Using CNN (Convolutional Neural Networ*k). 2019.
URL: https://towardsdatascience.com/cnn-sentiment-analysis-1d16b7c5a0e7
(3.7.2021.)
[57]Shung, *K. P. Accuracy, Precision, Recall or F1?*. 2018.
URL: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9
(3.7.2021.)

```
true=0
false=0

for i in range(len(pred)):
    if pred[i] == testY[i]:
        true+=1
    else:
        false+=1

print("točno predviđeno :: ", true)
print("krivo predviđeno :: ", false)
```

```
točno predviđeno ::  165
krivo predviđeno ::  40
```

Slika 21. Prikaz for loop-a i rezultata iteriranja nad testnim podacima

Korištenjem navedenog modela, mogao bi se bilo kakav korisnički tekstualni unos na formama web aplikacije ili dio izvornog koda nakon korisničkog unosa, automatski klasificirati kao maliciozan ili benigan JavaScript kod u odnosu na *Cross-site scripting* ranjivost.

7. Zaključak

U ovome radu prikazani su razni načini primjene tehnika računalne obrade prirodnog jezika u računalnoj sigurnosti na webu. Naglasak se stavlja na sinergiju korištenja tehnika obrade prirodnog jezika i različitih algoritama za izradu modela za klasifikaciju, predikciju i detekciju sigurnosnih incidenata. Neovisno o tome koji se algoritam strojnog učenja koristi za treniranje modela, važno je uočiti da kada skupovi podataka uključuju nizove znakova, gotovo uvijek je potrebno koristiti tehnike obrade prirodnog jezika kako bi se nad tekstualnim podacima mogao primijeniti određeni algoritam strojnog učenja. S obzirom da su podaci koji se dobivaju analizom sigurnosnih incidenata u znatnom postotku u obliku niza znakova, tehnike obrade prirodnog jezika mogu biti primijenjene na gotovo svaki aspekt računalne sigurnosti, tekst ne mora biti u obliku prirodnog govornog jezika, već može biti bilo kakav niz znakova. Međutim, za uspješnu primjenu takvih metoda potrebna je velika količina skupova podataka povezanih s individualnim sigurnosnim incidentima. Neke vrste skupova podataka je lako prikupiti, a odnose se na popularizirane sigurnosne ranjivosti na internetu, poput OWASP Top 10 i malicioznih URL-ova, koji se sastoje od niza znakova koji uglavnom ne predstavljaju prirodni jezik u upotrebi. Znatno je teže pristupiti tekstualnim podacima koji su u tekstualnom obliku nekog prirodnog govornog jezika, a odnose se na sigurnosne ranjivosti na internetu. Dakle, više se skupova podataka može prikupiti za analiziranje semantike programerskog koda sa aspekta računalne sigurnosti nego za analiziranje semantike prirodnog jezika sa aspekta računalne sigurnosti. Kada se uzme u obzir broj govornih jezika u svijetu, očito je da će za engleski jezik postajati najveći skup podataka za semantičku analizu s obzirom na sigurnost. Ipak, obrada prirodnog jezika predstavlja tehnike koje se sve intenzivnije koriste u računalnoj sigurnosti. U ovom radu je obrađena primjena nad web aplikacijama, međutim svoju primjenu navedene tehnike nalaze u svim vrstama sigurnosnih incidenata. Dakle, tehnike obrade prirodnog jezika predstavljaju sredstvo pomoću kojeg se može podatkovna znanost primijeniti na sigurnosne probleme te će svijetu računalne sigurnosti ovi alati donijeti nove mogućnosti, kako za obranu od zlonamjernih aktivnosti, tako i za stvaranje novih vrsta računalnih napada.

8. Literatura

Arshaad, S. *Sentiment Analysis / Text Classification Using CNN (Convolutional Neural Networ*k). 2019.
URL: https://towardsdatascience.com/cnn-sentiment-analysis-1d16b7c5a0e7
(3.7.2021.)

Bambrick, N. *Support Vector Machines: A Simple Explanation*. 2016.
URL: https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html
(3.7.2021.)

Barry, C. *Email threat types: URL phishing.* 2021.
URL: https://blog.barracuda.com/2021/02/03/email-threat-types-url-phishing/
(3.7.2021.)

Cross, M. *Uniform Resource Locator.* 2014.
https://www.sciencedirect.com/topics/computer-science/uniform-resource-locator
(3.7.2021.)

Dabar, *Eliminacija stop-riječi u pretraživanju.* 2020.
URL: https://dabar.srce.hr/faq-page/254
(3.7.2021.)

Edgar, Manz. *Research Methods for Cybersecurity*. 2017.
URL: https://www.sciencedirect.com/topics/computer-science/logistic-regression
(3.7.2021.)

Ghosh, Gunning. *Natural Language Processing Fundamentals.* 2019.
URL: https://www.packtpub.com/product/natural-language-processing-fundamentals/9781789954043
(3.7.2021.)

Grefenstette, Tapanainen. *What is a word, What is a sentence? Problems of Tokenization.* 1997.
URL: https://www.dfki.de/~neumann/qa-course/grefenstette94what.pdf
(3.7.2021.)

IAB/ABC, *International Spiders and Bots List*, URL: https://www.iab.com/guidelines/iab-abc-international-spiders-bots-list/ (3.7.2021.)

IBM, *What is cybersecurity.* 2021.
URL: https://www.ibm.com/topics/cybersecurity
(3.7.2021.)

ITU-T, *Definition of cybersecurity*. 2008.
URL: https://www.itu.int/en/ITU-T/studygroups/com17/Pages/cybersecurity.aspx
(3.7.2021.)

Kaspersky, *What Is a Drive by Download*. 2021.
URL: https://www.kaspersky.com/resource-center/definitions/drive-by-download
(3.7.2021.)

Kingma, Ba. *Adam: A Method for Stochastic Optimization*. 2014.
URL: https://arxiv.org/abs/1412.6980
(3.7.2021.)

Kirsten, S. *Cross-site scripting (XSS).* 2021.
URL: https://owasp.org/www-community/attacks/xss
(3.7.2021.)

Law Insider, *Cyber Event definition.*
URL: https://www.lawinsider.com/dictionary/cyber-event
(3.7.2021.)

Li, V. *Regular Expressions: A Quick Intro for Security Professionals* 2020.
URL: https://dzone.com/articles/regular-expressions-a-quick-intro-for-security-pro
(3.7.2021.)

Manning, Raghavan, Schütze. *Introduction to Information Retrieval.* 2008.
URL: https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html
(3.7.2021.)

NCSC.GOV.UK, *Phishing attacks: defending your organization.* 2021.
URL: https://www.ncsc.gov.uk/guidance/phishing
(3.7.2021.)

ODSC – Open Data Science, *An Introduction to Natural Language Processing (NLP).* 2019.
URL: https://medium.com/@ODSC/an-introduction-to-natural-language-processing-nlp-8e476d9f5f59
(3.7.2021.)

Open Web Application Security Project, *OWASP Top Ten – 2017 The Ten Most Critical Web Application Security Risks.* 2017.
URL: https://owasp.org/www-pdf-archive/OWASP_Top_10-2017_%28en%29.pdf.pdf
(3.7.2021.)

OWASP Cheat Sheet Series, *Cross Site Scripting Prevention Cheat Sheet.* 2021.
URL: https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html
(3.7.2021.)

Palo Alto Networks Inc., *Malicious URL Categories.* 2021.
URL: https://docs.paloaltonetworks.com/pan-os/9-1/pan-os-admin/url-filtering/url-categories/url-category-best-practices.html
(3.7.2021.)

Palo Alto Networks Inc., *Command and Control Explained*
URL: https://www.paloaltonetworks.com/cyberpedia/command-and-control-explained
(3.7.2021.)

Pantola, P. *Natural Language Processing: Text Data Vectorization*. 2018.
URL: https://medium.com/@paritosh_30025/natural-language-processing-text-data-vectorization-af2520529cf7
(3.7.2021.)

Patel, D. *Malicious and Benign Websites. 2019.*
URL: https://www.kaggle.com/deepsworld/malicous-and-benign-websites
(3.7.2021.)

Perkins, J. *NLP FOR LOG ANALYSIS – TOKENIZATION.* 2018.
URL: https://streamhacker.com/category/insight-engines/
(3.7.2021.)

Phishing.org, *What Is Phishing?*
URL: https://www.phishing.org/what-is-phishing
(3.7.2021.)

PortSwigger. *Cross-site scripting.* 2021.
URL: https://portswigger.net/web-security/cross-site-scripting
(3.7.2021.)

PortSwigger. *Stored XSS.* 2021.
URL: https://portswigger.net/web-security/cross-site-scripting/stored

(3.7.2021.)

Saha, S. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way.* 2018.
URL: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
(3.7.2021.)

Sharma, A. *Applying Data Science to Cybersecurity Network Attacks & Events*. 2019.
URL: https://www.kdnuggets.com/2019/09/applying-data-science-cybersecurity-network-attacks-events.html
(3.7.2021.)

Shukra, Iriondo. *Natural Language Processing (NLP) with Python — Tutorial*. 2020.
URL:https://pub.towardsai.net/natural-language-processing-nlp-with-python-tutorial-for-beginners-1f54e610a1a0
(3.7.2021.)

Shung, K. P. *Accuracy, Precision, Recall or F1?*. 2018.
URL: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9
(3.7.2021.)

Statista, *Global digital population as of January 2021.* 2021.
URL: https://www.statista.com/statistics/617136/digital-population-worldwide/
(3.7.2021.)

Stecanella, B. *What Is TF-IDF?* 2019.
URL: https://monkeylearn.com/blog/what-is-tf-idf/
(3.7.2021.)

Thorpe, Schwartz. *USING MACHINE LEARNING TO IDENTIFY PHISHING ATTACKS.* 2017.
URL:https://assets.ctfassets.net/kdr3qnns3kvk/5VuciSuo36Ac2eeyU02ywE/941f363e586c71cda688aca307b851fd/Thorpe-PACISE2017.pdf
(3.7.2021.)

Yordanov, V. *Introduction to Natural Language Processing for Text.* 2018.
URL: https://towardsdatascience.com/introduction-to-natural-language-processing-for-text-df845750fb63
(3.7.2021.)

Todorovic, A. *Nigerian Prince Scam Email*
URL: https://thepepsigeneration.com/nigerian-prince-scam-email/
(3.7.2021.)

ITRC: Identity Theft Resource Center, *PAYPAL PHISHING E-MAIL.* 2017.
URL: https://www.idtheftcenter.org/images/Pay_pal.jpg
(3.7.2021.)

hr.ilusionity.com, *SVM Vodič za strojno učenje - Što je algoritam strojnog vektora podrške, objašnjen s primjerima*
URL:https://hr.ilusionity.com/638-svm-machine-learning-tutorial-what-is-the-support-vector-machine-algorithm-explained-with-code-examples
(6.7.2021.)
Kovač, A.; Dunđer, I.; Seljan, S. An overview of machine learning algorithms for detecting phishing attacks on electronic messaging services. Proceedings MIPRO, 2022.

Dunđer, I., Pavlovski, M., Seljan, S. Computational Analysis of a LiteraryWork in the Context of Its Spatiality. World Conference on Information Systems and Technologies, 2020, 252-261.

Krstić, Ž., Seljan, S., Zoroja, J. Visualization of Big Data Text Analytics in Financial Industry: A Case Study of Topic Extraction for Italian Banks. Proceedings of EntreNova, 2019, 67-75.

9. Popis slika

Primjena tehnika obrade prirodnog jezika u računalnoj sigurnosti

Sažetak

Završni rad na temu primjene tehnika obrade prirodnog jezika u računalnoj sigurnosti prikazuje da navedene tehnike mogu biti upotrijebljene za više od obrade ljudskog prirodnog jezika te da se mogu koristiti na bilo kakvim tekstualnim podacima. Postoji više načina primjene u računalnoj sigurnosti koji će se razraditi u ovom radu, od generiranja algoritma za klasifikaciju domena do modela za identifikaciju zlonamjernih email poruka. Korištenjem tehnika obrade prirodnog jezika, izvršit će se i analiza ranjivosti izvornog koda na način da se prvo odrede uzorci funkcija povezanih sa već poznatim sigurnosnim propustima u izvornim kodovima. Fokus u ovom radu će biti sigurnost web aplikacije (sigurnost na web-u) i za sve navedene primjene trebat će prikupiti korpuse i primijeniti neke od standardnih operacija nad tekstom, kao što su parsiranje, stvaranje semantičkih mreža i lematizacije. Prikupljeni podaci koji se odnose na izvorni kod će se klasificirati na temelju OWASP Top Ten. Nakon prikupljanja korpusa i pripreme teksta, softversko rješenje identifikacije i klasifikacije će se prikazati pomoću programskog jezika Python.

Ključne riječi: računalna obrada teksta, strojno učenje, Python, OWASP Top 10, računalna sigurnost, NLP

Application of natural language processing techniques for cybersecurity

Summary

The bachelor thesis on the application of natural language processing techniques in cybersecurity shows that these techniques can be used for more than human natural language processing and that they can be used on any textual data. There are several ways of application in computer security that will be developed in this thesis, from generating a domain classification algorithm to a model for identifying malicious e-mail messages. Using natural language processing techniques, source code vulnerability analysis will also be performed by first determining patterns of functions related to already known security vulnerabilities in source codes. The focus in this work will be web application security (security on the web) and for all of the above applications corpora will need to be compiled and data will be processed by some of the standard text operations, such as parsing, creating semantic networks and tokenization. The data collected related to the source code will be classified based on the OWASP Top Ten. After collecting the corpus and preparing the text, a software solution for identification and classification will be shown using the Python programming language code.

Key words: computational text analysis, machine learning, Python, OWASP Top 10, cybersecurity, NLP