UNIVERSITETI I PRISHTINËS

FAKULTETI I INXHINIERISË ELEKTRIKE DHE KOMPJUTERIKE

PUNIM DIPLOME

Tema:

Zhvillimi i një "Chrome Extension" për detektimin e "Phishing Sites" me ML

Mentori:
Prof. Dr. Blerim Rexha

Kandidati:
Leand Thaçi

Prishtinë, Dhjetor 2020

Abstrakt

Sulmet “phishing” janë mënyra më e thjeshtë për të marrë informacion të ndjeshëm nga përdoruesit. Qëllimi i mashtruesve është të marrin informacione kritike si emrin e përdoruesit, fjalëkalimin dhe detajet e llogarisë bankare. “Phishing Sites” kanë qenë , janë dhe do të jenë edhe në të ardhmen problem i madh për sigurinë kibernetike. Kjo pasi që, nëse sigurinë kibernetike e mendojmë si një zinxhir, këto faqe në Internet sulmojnë hallkën më të dobët të këtij zinxhiri , i cili është faktori njeri. Përderisa njeriu ka rol në gjykimin dhe vendimmarrjen për ofrimin e të dhënave të ndjeshme në faqet e Internetit, rreziku nuk mund të evitohet plotësisht, mirëpo mund të zvogëlohet. Pikërisht qëllimi i “Chrome Extension”, të zhvilluar në kuadër të temës së diplomës, është që të shërbej si një ndërmjetësues në mes përdoruesit dhe faqeve në Internet. Arsyeja e zgjedhjes së metodës me ML është që të parandalohen sulmet e quajtura “zero-day”, pra të parandalohen sulmet nga faqet e reja të cilat nuk i ka raportuar ende askush si faqe të rrezikshme, mirëpo që në të vërtet janë faqe phishing.

Abstract

Phishing attacks are a simple way to obtain sensitive information from users. The aim of the phishers is to acquire critical information like username, password and bank account details. Phishing Sites have been, are and will be even in the future a big problem for cyber security. This is because , if we imagine cyber security being a chain, these webistes attack the weakest link in the chain, that being the human factor. As long as humans have a role in judging and decision making for offering sensitive data in websites, the risk cannot be completely avoided, but it can be mitigated. The aim of this tool is to serve as a middleware between the user and malicious websites. The main reason I chose the ML method is to prevent “zero-day” attack, which means to prevent attacks from new sites which haven`t yet been reported as dangerous, but in reality are phishing sites.

Falënderime

Falënderoj familjen time për përkrahjen dhe motivimin e vazhdueshëm. Falënderoj të gjithë stafin e departamentit “Inxhinieri Kompjuterike” në Fakultetin e Inxhinierisë Elektrike dhe Kompjuterike, dhe veçanërisht Prof. Dr.Blerim Rexha për ndihmën dhe këshillimet gjatë realizimit të këtij punim , dhe mbi të gjitha motivimin dhe energjinë pozitive që e ka pasqyruar tek ne gjatë këtyre viteve.

Përmbajtja

ABSTRAKT **1**

ABSTRACT **2**

FALËNDERIME **3**

LISTA E FIGURAVE **5**

LISTA E TABELAVE **5**

LISTA E SHKURTESAVE **6**

LISTA E PJESËVE TË KODIT **6**

1. HYRJE **7**

1.1. HYRJE 7
1.2. MOTIVIMI 8
1.3. PËRSHKRIMI I PROBLEMIT 9

2. GJENDJA E TANISHME **11**

3. PHISHING **13**

3.1. ÇKA ËSHTË PHISHING? 13
3.2. LLOJET E SULMIT PHISHING. 14

4. MACHINE LEARNING **16**

4.1. ÇKA ËSHTË MACHINE LEARNING? 16
4.2. ALGORITMET NË MACHINE LEARNING 18
4.2.1. Support Vector Machine (SVM) *18*
4.2.2. kNN – Nearest Neighbor *20*
4.2.3. Random Forest *21*

5. RASTI I STUDIMIT - ZHVILIMI I "NOPHISH" PËR CHROME **22**

5.1. METODOLOGJIA DHE VEGLAT E PËRDORURA 22
5.2. IMPLEMENTIMI 23
5.2.1. Përzgjedhja e dataset-it *23*
5.2.2. Përcaktimi i veçorive në bazë së cilave do të klasifikohen faqet *23*
5.2.3. Përzgjedhja e algoritmit optimal për rastin konkret *29*
5.2.4. Zhvillimi i komponentës "server-side" për klasifikimin e faqeve *31*
5.2.5. Zhvillimi i komponentës "client-side" *35*
5.2.6. Zhvillimi i webserverit për ndërlidhjen e dy komponentave. *38*
5.3. PUNË EKSPERIMENTALE 39

6. DISKUTIME DHE KONKLUKZIONE **40**

6.1. REFLEKTIM KRITIK 40
6.2. PËRMIRËSIMET E MUNDSHME 40

BIBLIOGRAFIA DHE REFERENCAT **42**

Lista e figurave

Figura 1. Statistika të raportimit ditorë të faqeve Phishing (2020) [3]. 7
Figura 2. Statistika në periudhë 1 vjeçare (shihet rritja në muajt e fundit) [3] 8
Figura 3 Përdorimi i protokollit HTTPS nga faqet phishing [3] 9
Figura 4 Përdorimi i shfletuesve (browsers) në nivel global. [4] 10
Figura 5. 10 kompanitë më të imituara nga sulmuesit phishing. [5] 14
Figura 6 Disa nga llojet e sulmit phishing [7]. 14
Figura 7 Tre paradigmat e ML, dhe aplikimet e tyre. [9] 17
Figura 8 Dallimi në mes Deep Learning dhe Machine Learning [11] 18
Figura 9 Parimi i punës së SVM Linear [12] 19
Figura 10 Trajnimi i modelit duke përdorur më shumë fusha [10] 19
Figura 11 Dallimi mes llojeve të ndryshme të peshës së fqinjëve 20
Figura 12 Diagrami i Random Forest [14] 21
Figura 13 Arkitektura e veglës NoPhish 22
Figura 14 Statistikat e testimeve të tre algoritmeve. 29
Figura 15 Rezultatet e trajnimit të klasifikuesit me algoritmin Random Forest 32
Figura 16 Rëndësia e vetive bazuar në MDI 33
Figura 17 Rëndësia e veçorive bazuar në permutacione 34
Figura 18 Pamja e parë e veglës 36
Figura 19 Pamja e veglës pas parashikimit 37
Figura 20 Pamjet e veglës kur parashikimi është si phishing apo si faqe e dyshimtë 37
Figura 21 Testimi i performancës së NoPhish dhe krahasimi me vegla si Google Safe Browsing dhe WoT 39

Lista e tabelave

Tabela 1 Rezultatet e testimit të algoritmeve për precision, recall, confusion matrix. 30

Lista e shkurtesave

CS	Cyber Security
ML	Machine Learning
BEC	Business Email Compromise
EAC	E-mail Account Compromise
URL	Uniform Resource Locator
HTTPS	Hypertext Transfer Protocol Secure
SVM	Support Vector Machine
MDI	Mean Decrease in Impurity

Lista e pjesëve të kodit

Kodi 1 Përgatitja e të dhënave për sklearn. 31
Kodi 2 Deklarimi i klasifikuesit dhe parametrave të tij.....31
Kodi 3 Funksioni main i cili bën klasifikimin e faqeve 35
Kodi 4 Fajlli manifest.json i detyrueshëm për funksionimin e Chrome Extension.....36
Kodi 5 Ekzekutimi i klasifikuesit në webserver dhe kthimi i rezultateve 38

1. Hyrje

1.1. Hyrje

Ky punim diplome trajton problemin e zhvillimit të një “chrome extension” si një vegël e cila do të shërbej si një ndërmjetësues në mes përdoruesit dhe faqeve me qëllime të vjedhjes së informacionit.

Në ditët e sotme “Phishing” bëhet një fushë kryesore shqetësuese për studiuesit e sigurisë kibernetike sepse nuk është e vështirë të krijosh një faqe të rreme e cila duket aq afër faqes së vertet, sa një përdorues jo vigjilent të mos e dalloj nga e vërteta. Nëse kemi njohuri bazike mbi çështjen dhe jemi vigjilent gjatë shfletimit të faqeve, mund të identifikojmë faqet e Internetit të rreme por jo të gjithë përdoruesit mund të identifikojnë faqen e Internetit të rreme dhe përdorues të tillë bëhen viktima të sulmit të mashtrimit. Qëllimi kryesor i sulmuesit është të vjedhë kredencialet e llogarive bankare dhe fjalëkalimet. “FBI Internet Complaint Center” raporton se 26 miliardë dollarë kanë humbur globalisht në krimet BEC/EAC gjatë periudhës së qershorit 2016 dhe korrikut 2019 [1]. Sulmet phishing po bëhen të suksesshme për shkak të mungesës së vetëdijes së përdoruesit. Meqenëse sulmi i phishing shfrytëzon mangësinë e vigjilencës tek përdoruesit, është shumë e vështirë të evitohen plotësisht ato, por është shumë e rëndësishme të përmirësohen teknikat e detektimit të phishing me qëllim të mitigohet këto sulme.

Çdo 20 sekonda hostohet një faqe e re “phishing” [2]. Bazuar në këtë statistikë , dhe në raportimet ditore të faqeve phishing në një ndër faqet kryesore kundër pshishing www.phishtank.com (Fig.1), shihet se përkundër mbylljes së vazhdueshme të këtyre faqeve, numri i tyre është në rritje.

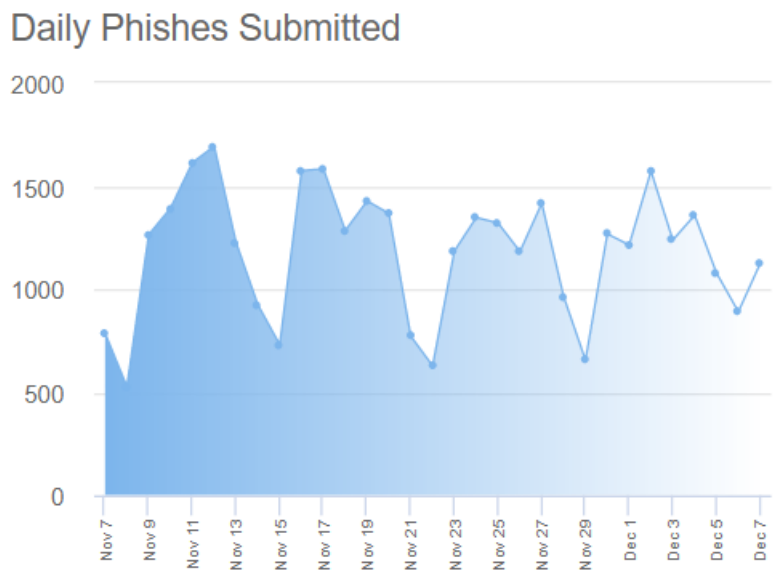

Figura 1. Statistika të raportimit ditorë të faqeve Phishing (2020) [3].

Bazuar në numrin e madh të faqeve phishing të reja çdo ditë ky punim diplome fokusohet në gjetjen e zgjidhjes së detektimit të faqeve phishing përmes ML. Arsyeja e zgjedhjes së kësaj metode (krahasuar me metodat tjera të cilat do të diskutohen ne Kapitullin 2.) është qëllimi që vegla të trajnohet në mënyrë që të mund të bëjë parashikim për faqe të cilat nuk i ka parë asnjëherë.

Disa nga sfidat që do të diskutohen në këtë punim lidhur me metodën ML janë : formimi i një dataset-i të besueshëm, gjetja e algoritmit optimal për trajnimin e klasifikuesit, optimizimi i algoritmit për rritjen e saktësisë, formimi i urës lidhëse në mes përdoruesit dhe klasifikuesit përmes “Chrome Extension”.

1.2. Motivimi

Sulmet phishing janë sulme të cilat nuk shënjestërojn vetëm një sistem apo vetem një sistem operative. Ato mund të i gjejmë kudo në botën virtuale në shumë forma të ndryshme. Ndonëse sot ekzistojnë sisteme të ndryshme për mbrojtje ndaj sulmeve phishing qoftë si aplikacione kundër phishing, qoftë si organizime për trajnimin e përdoruesve, çdo kemi numër të madh të viktimave.

Si pasojë e kësaj dëmtohen shumë persona, kompani, organizata. Disa statistika që ndërlidhen me rritjen e numrit të faqeve phishing nga marsi i vitit 2020 deri në shtatorin e vitit të njejtë janë paraqitur në Fig.2.

Figura 2. Statistika në periudhë 1 vjeçare (shihet rritja në muajt e fundit) [3]

Andaj vazhdimisht ka hapësirë për përmirësim në aspektin e mbrojtjes ndaj këtyre sulmeve dhe çdo hulumtim në këtë drejtim do të thotë më pak viktima. Sulmet e këtij lloji nuk mund të zgjidhen një herë përgjithmonë, vazhdimisht duhet bërë përpjekje të studimit të metodave të reja të sulmit dhe adaptimi ndaj tyre. Pikërisht kjo është edhe ideja e këtij punimi, që nëpërmjet një hulumtimi dhe analize të detajuar të ofrohen zgjidhje të reja, më të thjeshta për aplikim dhe me përdorim të teknologjive aktuale. Përveç pjesës së zhvillimit dhe trajnimit të veglës, ky punim i jep

rëndësi edhe thjetësis së përdorimit të veglës. Qëllimi është që çdo përdorues pavarësisht përgatitijes profesionale në këtë fushë të jetë në gjendje të marrë informacionet e duhura rreth faqes që po e viziton.

1.3. Përshkrimi i problemit

Një ndër sfidat e para gjatë zbatimit të Machine Learning[1] në identifikimin e faqeve phishing është posedimi i një dataset-i i cili është i përditësuar me të dhënat e fundit dhe me vlera informative. Fatëmirësisht ka shumë databaza online me dataset të shumtë. Një ndër faqet më të mëdha anti-phishing, ku çdo ditë paraqiten dhe validohen me mijëra faqe është “PhishTank”. Ndër tjera posedon edhe API përmes së cilit zhvilluesit mund të marrin të dhënat.

Meqenëse klasifikimi se a është faqe e dyshimt apo jo do të bëhet nga vegla e trajnuar me Machine Learning, paraqitet pyetja se në çka do të bazohet vegla për parashikimin e saj. Elementi kryesor që duhet të shikohet për detektimin e faqeve të dyshimta për phishing është URL i faqes. Në ditët e sotme nuk mjafton vetëm të shikojmë se a përdor faqja protokollin e sigurt HTTPS për komunikim. Sipas raportit të fundit nga APWG, rreth 80% të faqeve phishing përdorin protokollin HTTPS (Fig.3).

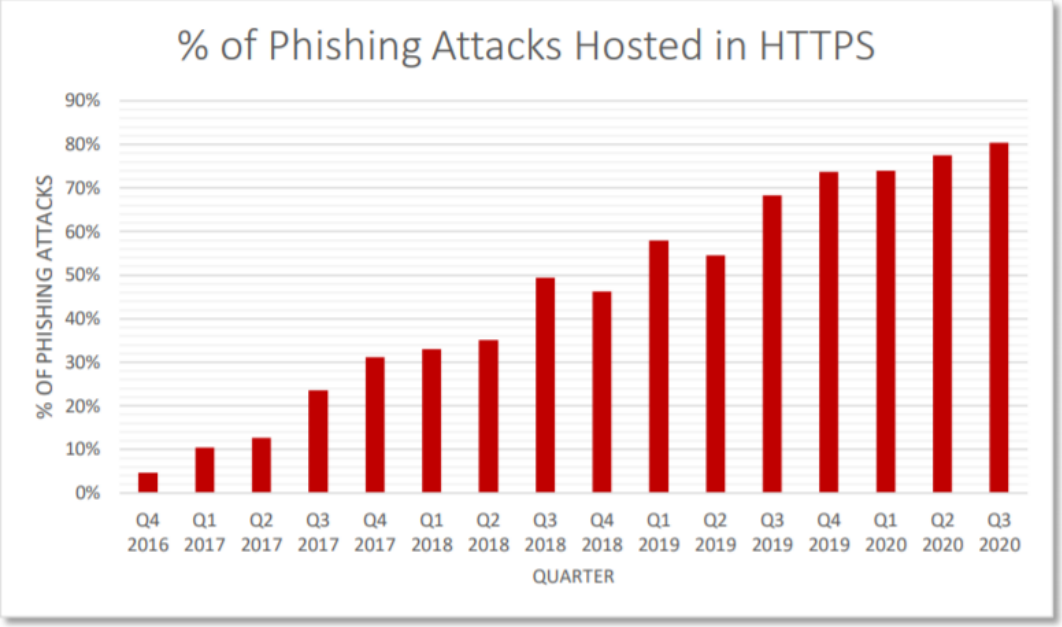

Figura 3 Përdorimi i protokollit HTTPS nga faqet phishing [3]

Prandaj nga URL nxerren disa veti në bazë të cilave një faqe mund të cilësohet e dyshimtë. Pjesa tjetër e vetive nxerret nga përmbatja e faqes, ku autorët e këtyre faqeve shpesh fokusohen vetëm të bëjn kopje pamjen e parë të faqes së vërtet dhe lënë shumë mangësi të cilat mund të identifikohen. Dhe kategoria e fundit e vetive të faqes janë kërkimi i kësaj faqeje në databazat më të njohura të statistikave rreth faqeve ne web, si “Alexa Database”.

[1] Machine Learning e përkthyer do të ishte mësimi i makinës , në këtë punim do të përdoret me kuptimin internacional në fushën e kibernetikës në gjuhen angleze.

Pjesa më e rëndësishme e veglës është padyshim algoritmi i trajnimit, në bazë së cilit vegla merr vendimet se si të i cilësoj faqet që i trajton. Në kuader të Machine Learning kemi një diversitet të madh të algoritmeve të cilët janë adekuat për situata të ndryshme. Për këtë punim rrethi është ngushtuar në 3 algoritma më adekuat, të cilët do të testohen dhe krahasohen. Ata janë : Random Forest, SVM (Support Vector Machine) , kNN (k-Nearest Neighbor).

Rolin e ndërmjetësuesit në mes përdoruesit dhe klasifikuesit do ta luaj Chrome Extension. Google Chrome ngjajshëm si dhe disa shfletues të tjerë, ofrojnë mundësinë e shtimit të një vegle shtesë e cila ka mundësi përdorimi në çdo faqe që shfleton, kjo i bie që vegla të jetë në pikën më të afërt të mundshme me faqen e dyshimtë. Paraqitet pyetja , pse pikërisht Google Chrome ? Sipas statistikave të Nëntorit 2020 të paraqitura në Fig.4, shfletuesi Chrome dominon me 66.1% të përdorimit në të gjithë botën.

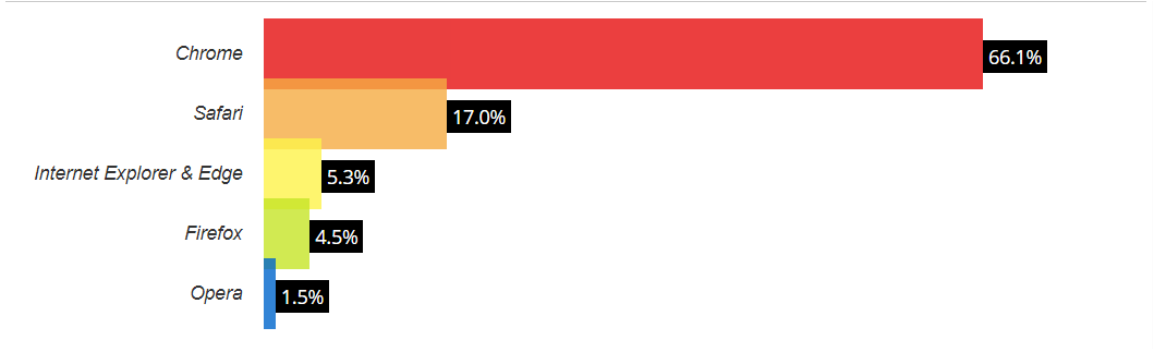

Figura 4 Përdorimi i shfletuesve (browsers) në nivel global. [4]

2. Gjendja e tanishme

Lufta kundër sulmeve phishing vazhdon që nga paraqitja e sulmit të parë e deri më sot. Fatkeqësisht në këtë betejë, rasti më i mirë që pala mbrojtëse mund të shpresoj është një barazim, pasi që pala sulmuese është gjithmonë një hap para. Sofistikimi i veglave dhe teknikave për të mbrojtur përdoruesit nga sulmet phishing po i detyron sulmuesit të adaptohen dhe të evoluojn teknikat e tyre. Sot sulmet phishing po bëhen gjithmonë e më polimorfike. Gjendja bëhet edhe më shqetësuese kur dihet se sulmet më të përsosura ndodhin brenda disa minutave dhe nuk përsëriten dy herë nga sulmuesi pa u përmirësuar për herën tjetër.

Një nga teknologjitë e fundit për phishing janë *trickbots*. Trickbots janë trojan bankar modular që targetojnë informacione të ndjeshme dhe vepron si një ndërmjetësues për malware të tjerë si ransomware[2]. Trickbots fillimisht kontrollojnë mjedisin ku ndodhen në mënyrë që të sigurohen se nuk janë në një mjedis sandbox[3], dhe pastaj tentojn të e ndalojnë antivirusin. Sulmet “zero-day attack” që nga definicioni janë sulmet që nuk kanë ndodhur më parë, sot nuk janë të rrallë. Këto sulme lancohen gjithmonë me një set të ri të IP adresave dhe variant të ri të virusit.

Mbrojtja nga phishing është një koncept i gjerë i cili mund të ndahet në dy kategori kryesore : trajnimi i njerëzëve me qëllim të rritjes së ndërgjegjësimit ndaj sulmeve phishing dhe mbrojtja kibernetike përmes softuerëve të ndryshëm. Ky punim diplome do të fokusohet në gjetjen e një zgjidhje që hynë në kategorinë e dytë të cekur më lartë. Softuerët kundër sulmeve phishing mund të ndahen në tre kategori: sisteme që përdorin blacklists, sisteme me rregulla të paracaktuara dhe sisteme që përdorin machine learning.

Metoda më e zakonshme për të zbuluar faqet phishing është duke azhurnuar URL-të dhe IP adresat e listuara si phishing në bazën e të dhënave të antivirusëve, kjo metodë njihet si metoda “blacklist”. Për të shmangur listat e zeza sulmuesit përdorin teknika kreative për të mashtruar përdoruesit duke modifikuar URL-në që të duket e ligjshme përmes obfuskimit dhe shumë teknika të tjera të thjeshta duke përfshirë edhe metodën “fast-flux”. Përmes kësaj metode proxies[4] gjenerohen automatikisht për të hostuar faqen në Internet dhe gjenerohen URL të reja përmes algoritmeve. E meta kryesore e kësaj metode është se, ajo nuk mund të zbulojë sulmin phishing “zero-day attack”. Pra duhet të ketë të paktën një viktimë i cili do të raportoj këtë faqe dhe do të listoj atë faqe në listën e zezë. Një ndër veglat më të njohura të këtij lloji është “Netcraft Extension”. Kjo vegël përbëhet nga një komunitet i madh, dhe pret që viktima e parë të raportoj një faqe në mënyzë që të e bllokoj atë për përdoruesit e tjerë.

Metodë tjetër është edhe sistemi me rregulla të paracaktuara. Ku vegla e analizon përmbajtjen e një faqe dhe në bazë të rregullave që i ka të paracakatuara vendos nëse ajo faqe konsiderohet si

[2] Ransomware paraqet një lloj virusi i cili enkripton të dhënat e viktimës dhe kërcënon se ato do të fshihen nëse nuk paguan një shumë (haraçin) të caktuar.
[3] Mjedisi sandbox paraqet një mjedis të izoluar ku ekzekutohen virusët e ndryshëm me qëllim studimi të tyre.
[4] Proxy serverat janë serverat që luajnë rolin e ndërmjetësuesit në mes kërkesës së klientit dhe serverit ku po dërgohet ajo kërkesë.

phishing apo jo. Problemi i këtij lloj sistemi ndonëse në parim mund të ndaloj sulmet “zero-day”, karakteristikat e paracaktuara nuk janë të garantuara të gjenden në çdo sulm kështu numri i parashikimeve false pozitive është shumë i lartë dhe përdorimi mbetet shumë i ngushtë. Një ndër veglat më të njohura të këtij lloj sistemi është “Phish Detector”. Vegël e cila parandalon sulmet phishing në online banking. Është një sistem “rule-based” dhe pretendon zero false negativ parashikime. E meta është se është e limituar vetëm në domainin e online banking.

Metoda e tretë, detektimi i faqeve phishing përmes ML, tenton të ndaloj sulmet “zero-day” njëkohësisht duke përfshirë një sa më shumë domene të sulmit. Pra një vegël e gjeneralizuar. Në ditët e sotme ka mjaft vegla me këtë metodë, mirëpo gjithmon ka hapësirë për përmirësime. Një ndër veglat më të njohura është “Cascaded Phish Detector” e cila ngjajshëm me këtë punim përbëhet nga një komponent në anën e klientit dhe një komponentë në anën e serverit. Hapësira ku mendohet se mund të përmirësohet nga kjo vegël është se ajo merr parasysh vetëm përmbajten HTML të faqes se jo parametra të tjerë.

3. Phishing

3.1. Çka është phishing?

Phishing është një sulm kibernetik në të cilin mashtruesit përpiqen të vjedhin informacione ose të dhëna të ndjeshme nga viktimat, duke u maskuar si një burim i besueshëm. Përveç kësaj phishing është bërë edhe si urë lidhëse për sulmet të tjera, duke mashtruar përdoruesit për të shkarkuar dokumente të rrezikshëm si virus, malware apo ransomware.

Sado i përhapur dhe i njohur sa është phishing tani, ai nuk ka ekzistuar përgjithmonë. Megjithëse praktika filloi diku rreth vitit 1995, këto lloje mashtrimesh nuk njiheshin zakonisht nga njerëzit e zakonshëm deri gati dhjetë vjet më vonë. Kjo nuk do të thotë që phishing nuk ishte një forcë për tu llogaritur që nga fillimi. Fjala phishing u përdor për herë të parë rreth vitit 1996 nga hakerat që vidhnin llogaritë dhe fjalëkalimet e America Online. Për analogji me sportin e peshkimit, këta mashtrues në Internet dërgoin email të rrezikshëm, duke vendosur grepa për të "peshkuar" për fjalëkalime dhe të dhëna financiare nga "deti" i përdoruesve të Internetit. Ata e dinin që edhe pse shumica e përdoruesve nuk do të binin pre e këtyre sulmeve, disa të tjerë do të bëheshin viktima. Termi u përmend në grupin e lajmeve të hakerave alt.2600 në janar 1996, por mund të jetë përdorur më parë në revistën e shtypur 2600, The Hacker Quarterly. Një rregull tipik nga hakerët është që shkronja "f" të shndërrohet në "ph". Origjina e fjalës phishing konsiderohet të jetë një shtrirje e fjalës "phreaking". Phreaking u shpik nga John Draper, ose ndryshe Captain Crunch, i cili krijoi Kutinë famëkeqe “Box Blue” që emetonte tinguj të dëgjueshëm për të hakuar sistemet telefonike në fillim të viteve 1970 [6].

Në ditët e sotit phishing është bërë një koncept i gjerë të cilin e gjejmë në shumë forma të ndryshme (për të cilat do të diskutohet më gjerë në Kapitullin 2.2). Mirëpo pa marrë parasysh se cilën metodë përdorin mashtruesit, qëllimi mbetet i njejtë, vjedhja e informacioneve të ndjeshme si të dhënat e llogarive bankare, fjalëkalimet etj. Disa nga tentimet e sulmit shpesh duken jo mirë të planifikuara, duke përdorur gramatikë të dobët ose duke kërkuar të klikojmë në linqe me pamje të çuditshme. Mirëpo çëllimi kryesor i sulmuesit është që sulmi i tij qoftë në formë të faqes në Internet apo si emal, të shpërndahet sa më shumë dhe të arrij te sa më shumë viktima potencial. Mjafton që të pakten një përdorues të bëhet viktimë e atij sulmit, që të cilësohet si i suksesshëm.

Viktimat e sulmeve mund të jenë nga kategori të ndryshme, që nga kompanitë e mëdha deri tek individët si përdorues të zakonshëm të Internetit. Sulmi më i shpeshtë ndaj kompanive të mëdha është targetimi i punëtorëve të asaj kompanie përmes email-ve phishing. Kurse sa i përket sulmeve të cilat nuk kanë cak një person apo kompani të caktuar, mirëpo bëhen për masa të gjera, lloji më i shpeshtë sulmit është imitimi i kompanive të mëdha. Sipas statistikave të nxerra nga “Vade Secure” për çerekun e tretë të vitit 2020, kompania e cila imitohet më së shpeshti vazhdon të jetë “Microsoft” e cila është larg vendit të dytë, siç shihet në Fig.5.

#		Brand	Unique Phishing URLs
1	-	Microsoft Category: Cloud	13,617
2	-	Facebook Category: Social Media	2,868
3	-	PayPal Category: Financial Services	2,512
4	↑3	eBay Category: E-Commerce/Logistics	1,358
5	↓1	Chase Category: Financial Services	932
6	-	Amazon Category: E-Commerce/Logistics	901
7	↑5	Netflix Category: Cloud	889
8	↓3	WhatsApp Category: Social Media	839
9	↑4	DHL Category: E-Commerce/Logistics	748
10	↑4	Google Category: Cloud	735

Figura 5. 10 kompanitë më të imituara nga sulmuesit phishing. [5]

3.2. Llojet e sulmit phishing.

Që nga paraqitja e parë e sulmit phishing në vitin 1995 e deri më sot ky lloj sulmi ka evoluar dhe marrë forma nga më të ndyshme. Sot mund të themi se numri i llojeve të phishing është mbi dhjetë. Disa nga llojet janë paraqitur në Fig. 6.

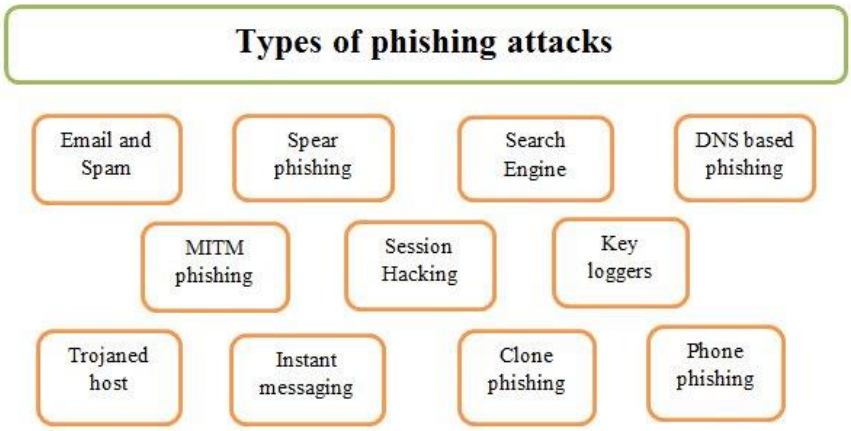

Figura 6 Disa nga llojet e sulmit phishing [7].

Llojet më të rendësishme të phishing:

Spear Phishing

Spear Phishing paraqet llojin e sulmit phishing ku targetohet një individ apo një kompani e caktuar. Për dallim nga phishing masiv, sulmuesit para se të kryejnë sulmin mledhin informacione për viktimën. Studimi i parë që u bë për “social phishing” i cili është një lloj i spear phishing që shfrytëzon informacionin e miqësisë nga rrjetet sociale, rezultoi mbi 70 përqind suksesishëm në eksperimente [8].

Disa nga rastet më të njohura të spear phishing janë : Targetimi i llogarive të gmail lidhur me kampanjën presidenciale të Hillary Clinton nga grupi “Threat Group-4127 (Fancy Bear) ” , Midis 2013 dhe 2015, Facebook dhe Google u dëmtuan nga 100 milion dollarë për shkak të një fushate të zgjatur phishing. Sulmuesi përfitoi nga fakti që të dy kompanitë përdorën Quanta, një kompani me bazë në Tajvan, si një shitës. Sulmuesi dërgoi një seri faturash të rreme tek kompania që imitonte Quanta, të cilat u paguan si nga Facebook ashtu edhe nga Google.

Whaling

Viktimat e sulmit Whaling janë njerëzit me pozita të larta në kompani të mëdha. Një email Whaling mund të thotë që kompania e tyre po paditet dhe ata duhet të klikojnë në linkun për të marrë më shumë informacion. Linku pastaj i çon në një faqe ku ata fusin të gjitha të dhënat kritike për kompaninë e tyre si Id e taksave dhe numrin e llogarisë bankare .

Email Phishing

Phishing me email është ndoshta lloji më i zakonshëm i phishing që është parë që nga vitet '90. Këto janë email që dërgohen në çdo adresë emaili që një haker mund të marrë. Emaili zakonisht i thotë marrësit se ka pasur një kompromis në llogarinë e tyre dhe ata duhet të përgjigjen menjëherë duke klikuar në linkun e dërguar. Këto sulme zakonisht janë të lehta për tu dalluar pasi që në ditët e sotme kryhen nga vende jo të zhvilluara dhe mund të vrehen që nga gramatika e dobët e deri tek formulimi jo i mirë.

Clone Phishing

Për këtë lloj sulmi, sulmuesi imiton një nga faqet më të vizituara në Internet. Kopjon pamjen e parë, zakonisht pamjen ku përdoruesin kyçen në atë sistem. Nëse përdoruesi nuk është i vëmendshëm atëher mund të jap informacione të ndjeshme duke menduar se po kyçet në faqen reale.

4. Machine Learning

4.1. Çka është Machine Learning?

Machine Learning (ML) është studimi i algoritmeve kompjuterikë që përmirësohen automatikisht përmes përvojës. Shihet si një nëngrup i inteligjencës artificiale. Algoritmet e Machine Learning ndërtojnë një model të bazuar në të dhënat e mostrës, të njohur si "të dhëna trajnimi", në mënyrë që të bëjnë parashikime ose vendime pa u programuar në mënyrë të qartë për ta bërë këtë. Algoritmet e të Machine Learning përdoren në një larmi të gjerë aplikimesh, të tilla si filtrimi i postës elektronike[5] dhe vizioni kompjuterik[6], ku është e vështirë ose e parealizueshme të zhvillohen algoritme konvencionale për të kryer detyrat e nevojshme. ML është bërë një pjesë integrale e shumë aplikacioneve tregtare dhe projekteve kërkimore, por kjo fushë nuk është ekskluzive për kompanitë e mëdha me ekipe të gjera kërkimore. Nëse përdorim Python, edhe si fillestar mund të ndërtojmë vetë zgjidhjet tona për të ML. Me të gjitha të dhënat e disponueshme sot, aplikacionet që në thelb kanë ML kufizohen vetëm nga imagjinata jonë.

ML është koncept i vjetër, që është krijuar si term qysh prej kohës së kompjuterëve të parë. Ndonëse për një kohë shumë shkencëtarë u morën me këtë problem kjo fushë u neglizhua për një kohë të gjatë për shkak të dështimeve të asaj kohe dhe pamundësive teknologjike për të realizuar diçka konkrete. Mirëpo në dekadat e fundit ky koncept është ringjallur dhe avancimet e mundësitë janë shtuar shumë. Ndoshta prej përdorimeve të para më interesante të kësaj fushe ishin lojërat e shahut që u organizua mes makinës dhe shahistëve të njohur. Kurse prej përdorimeve të sotme kemi shumë e padyshim që kompania gjigande Google është lider në këtë fushë, duke filluar nga organizimi i mesazheve në spam/not spam në mënyrë automatike që është rast tipik i Machine Learning dhe sistemin më të avancuar e posedon kompania në fjalë, pastaj organizimi i fotove në Google Photos ku mjaft të etiketojmë në një foto personat dhe pastaj në të gjitha të tjerat do të kemi etiketim automatik nga makina. Padyshim që një prej implementimeve më të njohura të Machine Learning është edhe aplikacioni i njohur për përdoruesit e Apple, Siri, i cili është aplikacion që u ofron asistencë përdoruesve duke i dëgjuar ata, funksion të ngjashëm ka edhe Cortana në sistemin operativ Windows. Ekzistojnë edhe chatbot të tjerë, të cilët dita e ditës po rriten dhe po bëhen më intelegjent falë Machine Learning dhe koncepteve të inteligjencës artificiale në përgjithësi.

Në Machine Learning janë tre paradigma kryesore [23], duke filluar nga mësimi i mbikëqyrur (supervised learning) që realizohet duke e trajnuar modelin me të dhëna të etiketuara (labeled data) paraprakisht, proces ky që shpesh është mjaft i kushtueshëm në të gjitha aspektet. Paradigma e dytë është mësimi i pa-mbikëqyrur (unsupervised learning) që në fakt është i kundërt me atë paraprak, pra këtu kemi të dhëna për të cilat nuk kemi informata rreth etiketave të tyre por që duke i analizuar ato si tërësi mundohemi të gjejmë mostra brenda tyre pa pasur nevojë për mbikëqyrje shtesë. Kurse

[5] Filtrimi i postës elektronike është përpunimi i postës elektronike për ta organizuar atë sipas kritereve të specifikuara

[6] Vizioni kompjuterik është një fushë shkencore ndërdisiplinore që merret me mënyrën se si kompjuterët mund të fitojnë kuptim të nivelit të lartë nga imazhet ose videot digjitale.

si paradigmë e tretë është edhe mësimi përforcues (reinforcment learning) që është edhe një prej metodave më të aplikuara tani, por edhe më i ndërlikuar sesa alternativat e tjera.

Në Fig.7 janë paraqitur tre paradigmat e ML dhe aplikimet e tyre sipas [9].

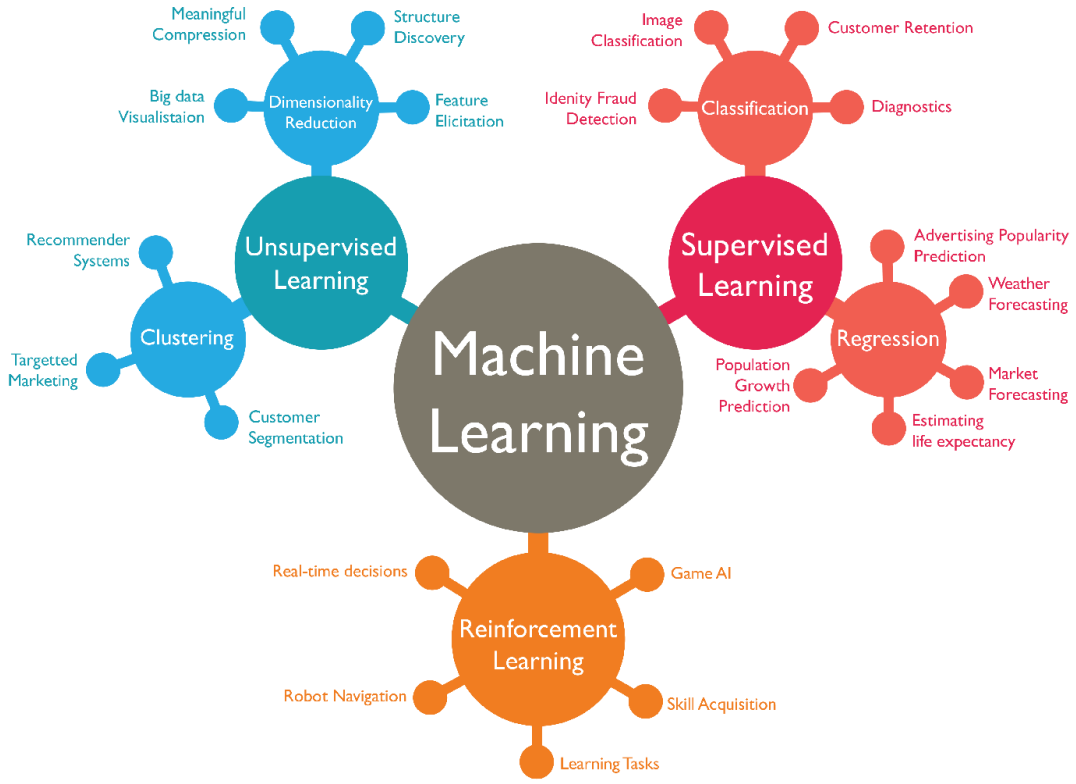

Figura 7 Tre paradigmat e ML, dhe aplikimet e tyre. [9]

Dy përkufizimet më të njohura të Machine Learning janë:

- Machine Learning është fusha e studimit që u jep mundësinë makinave të mësojnë pa pasur nevojë të programohen në mënyre eksplicite (përkufizim i Arthur Samuel - 1959)
- Një program kompjuterik thuhet se mëson nga eksperienca E në relacion me një detyrë T dhe një performancë P, nëse performanca e tij në T e matur me P rritet me eksperiencën E (përkufizimi i Tom Mitchell – 1998) [13].

Si nënpjesë e Machine Learning është edhe një fushë e re e quajtur Deep Learning, e cila realizohet përmes neural networks dhe është prej fushave më të avancuara në konceptin e inteligjencës artificiale [10]. Si shihet në Fig.8 , Deep Learning shkon edhe një hap para duke kryer edhe punën e nxerrjes së vetive.

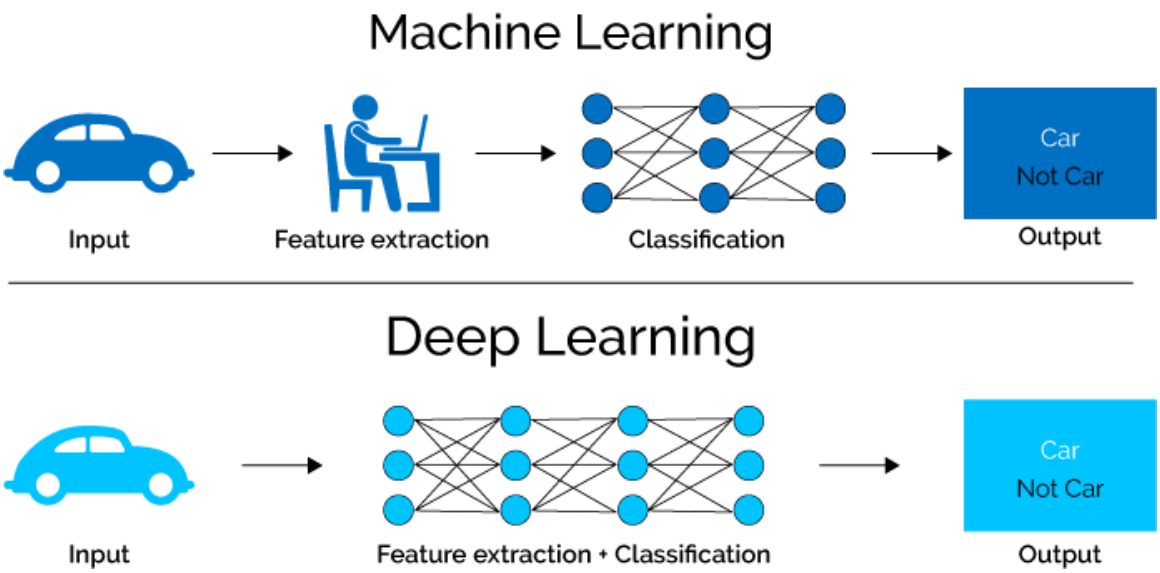

Figura 8 Dallimi në mes Deep Learning dhe Machine Learning [11]

4.2. Algoritmet në Machine Learning

Pasi e tërë dega e Machine Learning ndahen në paradigmat e përmendura në pikën 3.1. edhe algoritmet që do të diskutohen do u takojnë paradigmave të ndryshme. Algoritmet të cilët duhet për punimin e kësaj teme diplome janë algoritma për klasifikim. Zakonisht këta algoritma i takojn paradigmës së parë, pra supervised.

4.2.1. Support Vector Machine (SVM)

Parimi i SVM linear është paraqitur në Fig.9. Siç shihet në figurë kufirin klasifikues e përcaktojnë të ashtuquajturit support vectors. Ky është rast i përkryer kur të dhënat janë mjaftueshëm të diferencueshme dhe nuk ka përjashtime (outliers), përkatësisht instanca të cilat ndodhen jashtë kufijve të përcaktuar. Nëse modeli trajnohet duke përdorur margjina të forta (hard margin classification) atëherë modeli funksionon vetëm nëse të dhënat janë linearisht të ndara. Për këtë arsye duhet të përcaktohet parametra më elastik të margjinave, përkatësisht të kufijve klasifikues (soft margin classification) [10].

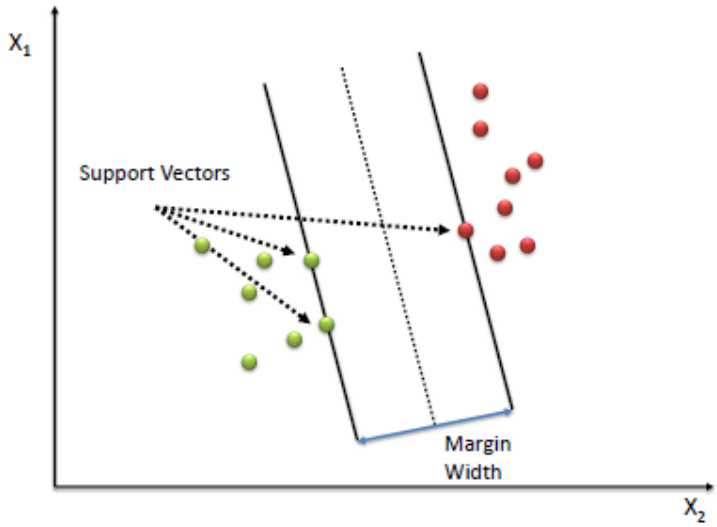

Figura 9 Parimi i punës së SVM Linear [12]

Në anën tjetër shumë dataset-e nuk janë linearisht të ndara andaj për këtë është e mundur që të shtohen më shumë veti(features) sikurse është zbatimi i algoritmit PolynomialFeatures. Në këtë rast SVM klasifikuesi mund të klasifikojë të dhënat me lakore jo-lineare sikurse është paraqitur në Fig.10.

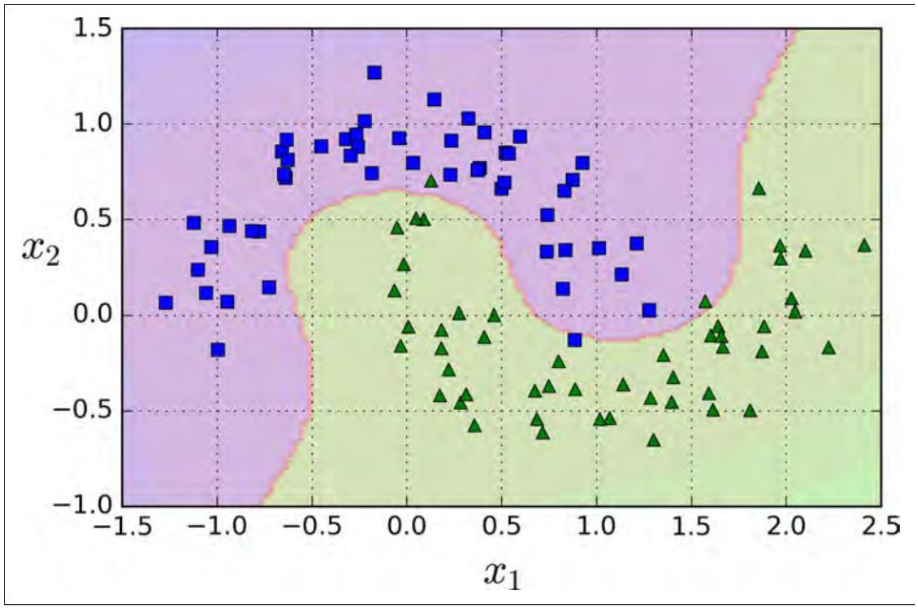

Figura 10 Trajnimi i modelit duke përdorur më shumë fusha [10]

4.2.2. kNN – Nearest Neighbor

Parimi që qëndron pas kNN është të gjesh një numër të paracaktuar të mostrave të trajnimit më të afërt në distancë nga pika e re dhe të parashikosh nga këto. Numri i mostrave mund të jetë një konstante e përcaktuar nga përdoruesi (k-nearest neighbor learning), ose e ndryshueshme bazuar në dendësinë lokale të pikave (radius-based neighbor learning). Distanca, në përgjithësi, mund të jetë çdo masë metrike: distanca standarde Euklidiane është zgjedhja më e zakonshme. Metodat e bazuara në fqinjët janë të njohura si metoda jo-përgjithësuese të ML, pasi ato thjesht "kujtojnë" të gjitha të dhënat e tyre të trajnimit.

Pavarësisht nga thjeshtësia e saj, kNN ka qenë i suksesshëm në një numër të madh të klasifikimit dhe problemeve të regresionit, duke përfshirë shifrat e shkruara me dorë dhe skenat e imazheve satelitore. Duke qenë një metodë jo-parametrike, ajo shpesh është e suksesshme në situatat e klasifikimit ku kufiri i vendimit është shumë i parregullt.

Klasifikimi themelor i kNN përdor pesha uniforme: domethënë, vlera e caktuar për një pikë pyetëse llogaritet nga një shumicë e thjeshtë e votave të fqinjëve më të afërt. Në disa rrethana, është më mirë të peshosh fqinjët ashtu që fqinjët më të afërt të kontribuojnë më shumë në përshtatjen. Kjo mund të realizohet përmes fjalës kyçe të peshave. Vlera e paracaktuar, weights = 'uniforme', cakton pesha uniforme për secilin fqinj. weights = “distance” cakton pesha proporcionale me distancën nga pika e pyetjes. Përndryshe, një funksion i përcaktuar nga përdoruesi i distancës mund të ofrohet për të llogaritur peshat.

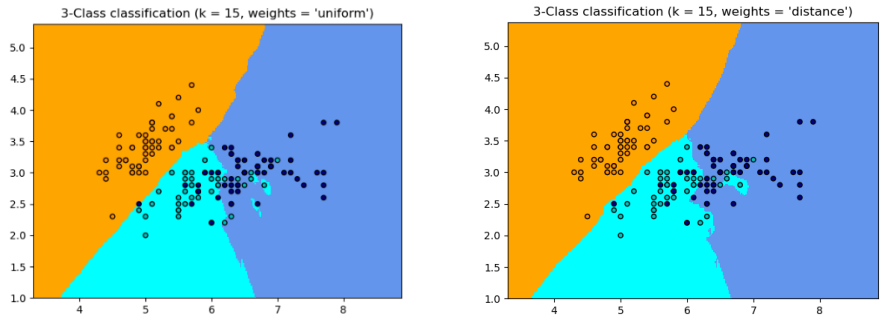

Figura 11 Dallimi mes llojeve të ndryshme të peshës së fqinjëve

4.2.3. Random Forest

Random Forest është një algoritëm fleksibil, i lehtë për të u përdorur që prodhon, edhe pa rregullimin e parametrave për raste specifike, një rezultat të shkëlqyeshëm në shumicën e kohës. Random Forest është një nga algoritmet më të përdorura, për shkak të thjeshtësisë dhe diversitetit të tij (mund të përdoret si për klasifikimin ashtu edhe për detyrat e regresionit).

Random Forest është një version i avancuar i algoritmit Decision Trees, në thelb e përdor algoritmin Decision Trees në nëngrupe të ndryshme të dataset-it dhe përdorë vlerën mesatare për të rritur performancen dhe për të evituar overfitting[7]. Kjo metodë njihet si “bagging” ose “bootstrap aggregating”. Formula më posht e tregon mënyrën se si funksionin metoda “bagging” [24].

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x) \qquad (1)$$

Një tjetër cilësi e shkëlqyeshme e algoritmit të Random Forest është se është shumë e lehtë të matësh rëndësinë relative të secilës veçori në parashikim. Sklearn ofron një mjet të shkëlqyeshëm për këtë që mat rëndësinë e një tipari duke parë se sa nyjet e pemëve që përdorin atë veçori zvogëlojnë papastërtinë në të gjithë pemët në pyll. Ai llogarit këtë rezultat automatikisht për secilën veçori pas trajnimit dhe shkallëzon rezultatet, kështu që shuma e të gjithë rëndësisë është e barabartë me një. Në Fig.12 është paraqitur diagrami i plotë i algoritmit Random Forest.

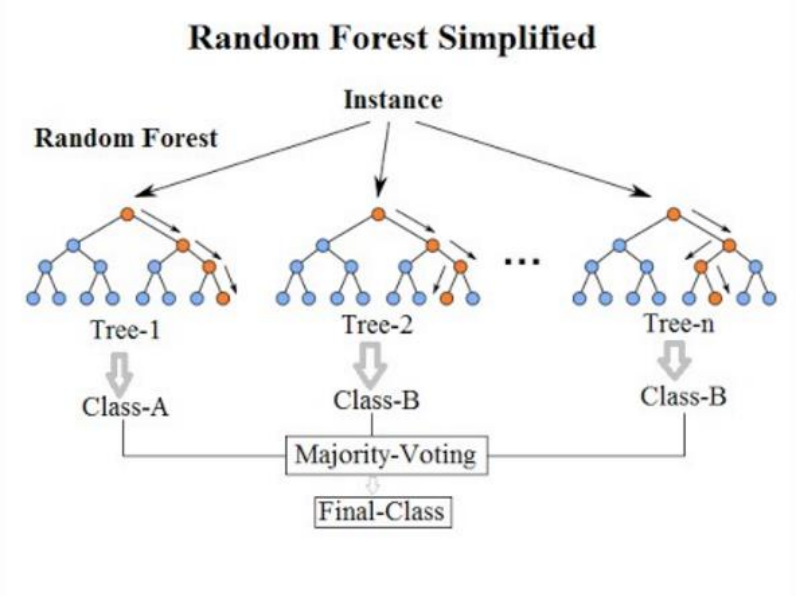

Figura 12 Diagrami i Random Forest [14]

[7] Overfitting do të thotë kur modelit të trajnimit i japim aq shumë të dhëna sa që fillon të ndikoj negativisht në saktësinë e parashikimit të rasteve të reja.

5. Rasti i studimit - zhvilimi i “NoPhish” për chrome

5.1. Metodologjia dhe veglat e përdorura

Vegla përbëhet nga dy komponenta kryesore : komponenta “client-side” me të cilën do të ketë përdoruesi interaksion , komponenta “server-side” e cila do të jetë klasifikuesi që merr vendimet se a cilësohet një faqe si phishing dhe webserveri i cili do të luaj rolin e ndërmjetësuesit në mes dy komponentave dhe do të krijoj ambientin për ekzekutimin e klasifikuesit. Në Figurën 13 është paraqitur arkitektura e veglës.

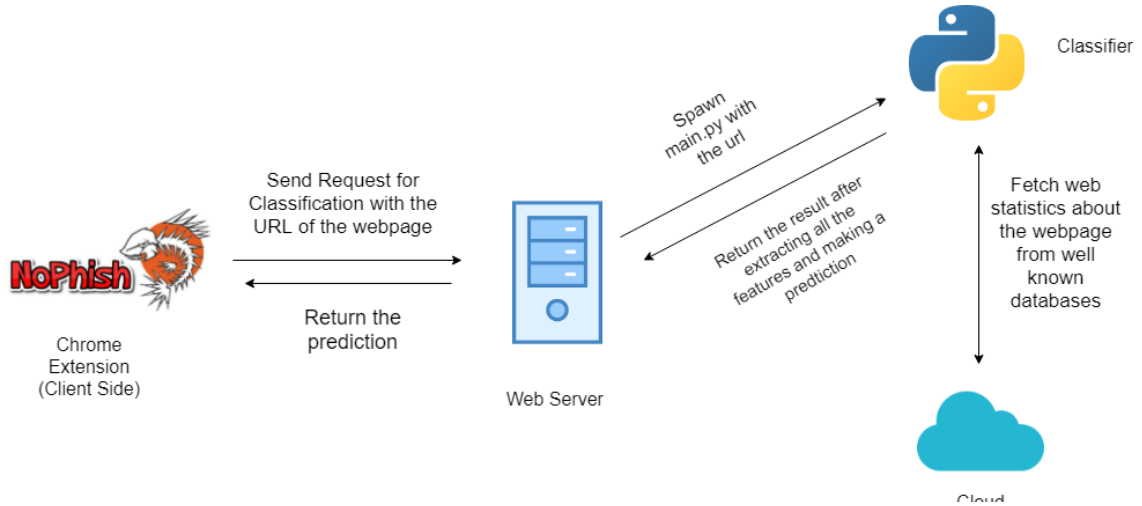

Figura 13 Arkitektura e veglës NoPhish

Në bazë të analizës së punës së nevojshme, rrjedha që do të ndiqet në këtë punim do të jetë si vijon:

1 Përzgjedhja e një dataset-i informativ.
2 Përcaktimi se cilat veçori të të dhënave do të merren parasysh.
3 Përzgjedhja e algoritmit optimal për rastin konkret.
4 Zhvillimi i komponentës “server-side” për klasifikimin e faqeve.
5 Zhvillimi i komponentës “client-side”.
6 Zhvillimi i webserverit për ndërlidhjen e dy komponentave.

Meqenëse pike kyqe e këtij punimi është ML, është vendosur që për implementimin e komponentës së klasifikuesit të përdoret gjuha Python (versioni 3.9). Një ndër avantazhet e kësaj gjuhe është padyshim numri i madh i librarive. Gjuha Python sot njihet si lidere në fushën e ML, përmes librarisë *scikit-learn* implementimi i algoritmave për ML është më i lehtë se asnjëherë tjetër. Përveq kësaj Python po ashtu lehtëson procedurën e manipulimit me të dhëna për nxjerrjen e veçorive të faqes. . Libraria numpy është përdorur për manipulimet me array të mëdha

shumëdimensionale, ndërsa matplotlib është përdorur për paraqitjet grafike të rezultateve dhe të dhënave statistikore të nevojshme për vizualizim. Komponenta “client-side” do të zhvillohet si “Chrome Extension” për arsye të qasjes shumë të lehtë të përdoruesit. Arsyeja e përzgjedhjes së shfletuesit Google Chrome është përqindja e madhe e përdoruesve si është spjeguar edhe në kapitullin 1.3. Webserveri i cili do të bëjë ndërlidhjen në mes dy komponentave do të zhvillohet në gjuhen Node.Js, për arsye të performancës së mirë dhe thjeshtësis së përdorimit.

5.2. Implementimi

5.2.1. Përzgjedhja e dataset-it

Fatmirësisht sot bota e Internetit është e pasur me të dhëna të shumta. Shumë faqe të ndryshme kontribojnë në luftën kundër phishing me dataset të ndryshëm. Dataset-i i zgjedhur për trajnimin e veglës për këtë punim i është dhuruar Universitetit të Kalifornisë Irvine [15]. Fillimisht është dhuruar me të dhëna për rreth 2400 faqe në vitin 2015 mirëpo është përditësuar vazhdimisht dhe sot ky numër ka arritur deri ne rreth 11000. Dataset-i përmban të dhëna për tridhjetë veçori të faqeve.

5.2.2. Përcaktimi i veçorive në bazë së cilave do të klasifikohen faqet

Meqenëse dataset-i përmban veçori që ndihmojnë të detektohen faqet jo vetëm për phishing por edhe për rreziqe tjera. Nga tridhjetë veçori janë përzgjedhur njëzet e dy të cilat cilësohen si informative për phishing. Ato janë :

Përdor Ip adresë në vend të domain

Nëse një adresë IP përdoret për t`u qasur në faqe në vend të emrit të domain-it në URL, siç është "http://125.98.3.123/fake.html", kjo nuk është profesionale dhe llogaritet si faqe e dyshimtë. Ndonjëherë, IP adresa madje shndërrohet në kod heksadecimal siç tregohet në linkun vijues “http: //0x58.0xCC.0xCA.0x62/2/paypal.ca/index.html”.

$$\textit{Rregulla}: \begin{cases} \text{Nëse ka IP në vend të domainit} \rightarrow \text{ Phishing} \\ \text{Përndryshe} \rightarrow \text{ Legjitime} \end{cases} \quad (1)$$

URL i gjatë për të fshehur pjesët e dyshimta

Sulmuesit mund të përdorin URL të gjatë në mënyrë që të fshehin pjesët e dyshimta. Si për shembull:

http://federmacedoadv.com.br/3f/aze/ab51e2e319e51502f416dbe46b773a5e/?cmd=_home&dispatch=11004d58f5b74f8dc1e7c2e8dd4105e811004d58f5b74f8dc1e7c2e8dd4105e8@phishing.website.html

Duke analizuar dataset-in është formuar rregulla në vijim.

$$\textit{Rregulla:}\begin{cases} \text{Gjatësia e URL } < 54 \rightarrow \text{ Legjitime} \\ \text{Gjatësia e URL } \geq 54 \text{ dhe } \leq 75 \rightarrow \text{ E dyshimtë} \\ \text{Përndryshe} \rightarrow \text{ Phishing} \end{cases} \quad (2)$$

Ka përdorur servise të shkurtimit të URL

Shkurtimi i URL-së është një metodë në "World Wide Web" në të cilën një URL mund të bëhet shumë më e vogël në gjatësi dhe prapë të çojë në faqen e kërkuar të Internetit. Kjo është arritur me anë të një "HTTP Redirect" në një emër domain që është i shkurtër, i cili lidhet me faqen në Internet që ka një URL të gjatë. Për shembull, URL " https://fiek.uni-pr.edu/Shpallje-dhe-rezultate-(1)/Bachelor/Literatura.aspx" mund të shkurtohet në " https://rb.gy/x7ox7d".

$$\underline{\textit{Rregulla}}: \begin{cases} \text{TinyURL} \rightarrow \text{ Phishing} \\ \text{Përndyshe} \rightarrow \text{ Legjitime} \end{cases} \quad (3)$$

URL përmban simbolin “@”

Nëse URL përmban simbolin “@” atëher shfletuesi injoron gjithçka para këtij simboli dhe adresa reale gjindet pas simbolit “@”.

$$\text{Rregulla: } \begin{cases} \text{Url përmban simbolin @} \rightarrow \text{ Phishing} \\ \text{Përndryshe} \rightarrow \text{ Legjitime} \end{cases} \quad (4)$$

Ridirektim duke përdorur “//”

Ekzistenca e "//" brenda shtegut URL do të thotë që përdoruesi mund të ridirektohet në një faqe tjetër në Internet si p.sh: "http://www.legitimate.com//http://www.phishing.com". Ne do të shikojmë nëse hera e fundit e paraqitje së “//” është pas http eventualisht https.

$$\text{Rregulla: } \begin{cases} \text{Pozicioni i paraqitjes i fundit për "//" në URL} > 7 \rightarrow \text{ Phishing} \\ \text{Përndryshe} \rightarrow \text{ Legjitime} \end{cases} \quad (5)$$

URL përmban simbolin “-”

Sulmuesit kanë tendenca të shtojn prefike domainit që të veçoj emrin e ndonjë kompanie të madhe me qëllim mashtrimi. Si p.sh http://www.confirm-paypal.com/.

$$\text{Rregulla: } \begin{cases} \text{Emri i Domain përmban simbolin "} - \text{"} \rightarrow \text{ Phishing} \\ \text{Përndryshe} \rightarrow \text{ Legjitime} \end{cases} \quad (6)$$

Ka shumë subdomains

Nëse një faqe ka shumë subdomains atëherë mund të llogaritet se sulmuesi po mundohet të maskoj domainin e faqes.

$$\text{Rregulla: } \begin{cases} \text{Subdomain} = 1 \rightarrow \text{Legjitime} \\ \text{Subdomains} = 2 \rightarrow \text{E dyshimtë} \\ \text{Përndryshe} \rightarrow \text{Phishing} \end{cases} \quad (7)$$

Gjatësia e regjistrimit të domainit

Bazuar në faktin që faqet legjitime regjistrojn domainin për disa vite në avancë. Atëherë çdo domain i cili do skadoj në më pak se një vit, llogaritet si i dyshimtë.

$$\text{Rregulla: } \begin{cases} \text{Domain do skadoj në} \leq 1 \text{ vit} \rightarrow \text{Phishing} \\ \text{Përndryshe} \rightarrow \text{Legjitime} \end{cases} \quad (8)$$

Ka Favicon

Favicona është ikona e vogël që shfletuesi zakonisht e shfaqë në address bar për të na rikujtuar se në cilën faqe jemi, nëse ikona merret nga domain i jashtëm atëher konsiderohet si e dyshimtë.

$$\text{Rregulla: } \begin{cases} \text{Favicon merret nga domain i jashtëm} \rightarrow \text{Phishing} \\ \text{Përndryshe} \rightarrow \text{Legjitime} \end{cases} \quad (9)$$

Ekzistenca e tokenit HTTPS në pjesën e domain-it

Sulmuesit mund të tentojn të shtojn tokenin HTTPS në pjesën e domain-it për të mashtruar përdoruesit me mendimin se faqja është e siguruar me SSL. Si për shembull http://https-www-paypal-it-webapps-mpp-home.soft-hair.com/.

$$\text{Rregulla: } \begin{cases} \text{Domain përmban tokenin HTTPS} \rightarrow \text{Phishing} \\ \text{Përndryshe} \rightarrow \text{Legjitime} \end{cases} \quad (10)$$

Ka shumë Request URL

Nëse shumica e objekteve të një faqe janë të ngarkuara nga domain të jashtëm, atëherë kjo faqe llogaritet si e dyshimtë pasi që faqet legjitime e ngarkojn përmbajtjen e tyre.

$$\text{Rregulla: } \begin{cases} \%\text{ e Request URL} < 22\% \rightarrow \text{Legjitime} \\ \%\text{ e Request URL} \geq 22\% \text{ and } 61\% \rightarrow \text{E dyshimtë} \\ \text{Përndryshe} \rightarrow \text{Phishing} \end{cases} \quad (11)$$

URL i tagut anchor “<a>”

Anchor është elementi që definohet nga tagu <a>. Për këtë veçori ne do të analizojmë :

1. Nëse URL brenda tagut ka domain të ndryshëm nga faqja atëherë është rasi i njejtë me veçorinë paraprake.
2. Nëse URL nuk lidhet me ndonjë faqe tjetër, p.sh.:
   A. <a href=“#”>
   B. <a href=“#content”>
   C. <a href=“#skip”>
   D. <a href=“JavaScript ::void(0)”>

$$\underline{\textit{Rregulla}}: \begin{cases} \%\ \text{URL e Anchor} < 31\% \ \rightarrow \ \textit{Legjitime} \\ \%\ \text{URL e Anchor} \geq 31\% \text{ And} \leq 67\% \ \rightarrow \ \text{E dyshimtë} \\ \text{Përndryshe} \rightarrow \ \text{Phishing} \end{cases} \quad (12)$$

Linqet në tagjet <Meta>, <Script> and <Link>

Bazuar në faqe legjitime është e natyrshme që faqet të përdorin tagun <Meta> për të ofruar metadata për dokumentin HTML, tagun <Script> për të krijuar një skriptë në anën e klientit dhe tagun <Link> për të marr web resurse nga serveri. Nëse në këto tagje ka linka për domain të jashtëm atëherë llogaritet si tag I dyshimtë.

Rregulla:

$$\begin{cases} \%\ \text{e linqeve të jashtëm në } " < \text{Meta} > ", " < \text{Script} > " \text{ and } " < \text{Link}>" < 17\% \ \rightarrow \ \text{Legjitime} \\ \%\ \text{e linqeve të jashtëm në } < \text{Meta} >,\ < \text{Script} > \text{ and } < \text{Link}>" \geq 17\% \text{ And} \leq 81\% \ \rightarrow \ \text{E dyshimtë} \\ \text{Përndryshe} \rightarrow \ \text{Phishing} \end{cases}$$

(13)

Server Form Handler (SFH)

Nëse format në një faqe funksionet e formave janë një string i zbrazët, ose përmbajn “about:blank” konsiderohen të dyshimta sepse një veprim duhet të ndërmerret kur bëhet submit një informacion. Po ashtu nëse funksioni i formës është në një domain tjetër edhe kjo llogaritet si e dyshimtë.

$$\text{Rregulla:} \begin{cases} \text{SFH është "about: blank" ose e zbrazët} \ \rightarrow \ \text{Phishing} \\ \text{SFH referohet te një domain i jashtëm} \rightarrow \ \text{E dyshimtë} \\ \text{Përndryshe} \ \rightarrow \ \text{Legjitime} \end{cases} \quad (14)$$

Dërgimi i informacioneve në email

Format në faqe lejojnë përdoruesin të jap informacione të cilat do të shkojnë në server për të u procesuar, nëse një form dërgon këto të dhëna me email atëherë kjo është e dyshimtë, pasi që sulmuesi mund të jetë duke dërguar informacionet e ndjeshme në emailin e tij.

Rregulla:

$$\begin{cases} \text{Përdor funksionon "mailto:" për të bërë submit të dhënat e përdoruesit} \rightarrow \text{Phishing} \\ \text{Përndryshe} \rightarrow \text{Legjitime} \end{cases} \quad (15)$$

URL e pazakontë

Kjo veçori mund të nxirret nga databaza WHOIS [16]. Për një faqe legjitime pritet që emri i hostit të jetë pjesë e URL.

Rregulla:

$$\begin{cases} \text{Emri i hostit nuk përfshihet në URL} \rightarrow \text{Phishing} \\ \text{Përndryshe} \rightarrow \text{Legjitime} \end{cases} \quad (16)$$

Përmban tagun iframe

Iframe është një HTML tag i cili përdoret për të shfaqur një faqe tjetër brenda faqes ku gjindet përdoruesi. Sulmuesit shfrytëzojnë atributin “frameBorder” për të larguar kornizat dhe për të mos e vërejtur që je duke dhënë të dhënat personale në webfaqe tjetër.

Rregulla:

$$\begin{cases} \text{Përdor iframe} \rightarrow \text{Phishing} \\ \text{Përndryshe} \rightarrow \text{Legjitime} \end{cases} \quad (17)$$

Jetëgjatësia e domain-it

Kjo veçori mund të merret nga databaza WHOIS. Shumica e faqeve phishing kanë jetëgjatësi të shkurtër. Faqet e reja konsiderohen të dyshimta.

Rregulla:

$$\begin{cases} \text{Jetëgjatësia e domain} - \text{it} \geq 6 \text{ muaj} \rightarrow \text{Legjitime} \\ \text{Përndryshe} \rightarrow \text{Phishing} \end{cases} \quad (18)$$

Regjistri DNS

Nëse në databazën WHOIS nuk gjindet domain-i i faqes ose është i zbrazët, atëherë faqja konsiderohet si phishing.

Rregulla:

$$\begin{cases} \text{Nuk ka regjister DNS për domainin} \rightarrow \text{Phishing} \\ \text{Përndryshe} \rightarrow \text{Legjitime} \end{cases} \quad (19)$$

Trafiku i faqes

Kjo veçori kalkulon popularitetin e faqes, duke shikuar se sa vizitor ka pasur ajo. Kjo shifer nxirret nga databaza Alexa (Alexa the Web Information Company., 1996). Nëse faqja rangohet në top 100,000 atëherë konsiderohet si legjitime , nëse është nën 100,000 atëherë konsiderohet si e dyshimtë, kurse nëse nuk njihet nga Alexa konsiderohet si phishing.

$$\text{Rregulla:} \begin{cases} \text{Rangimi i faqes} < 100{,}000 \rightarrow \text{Legjitime} \\ \text{Rangimi i faqes} > 100{,}000 \rightarrow \text{E dyshimtë} \\ \text{Përndryshe} \rightarrow \text{Phishing} \end{cases} \tag{20}$$

Google Index

Kjo veçori kontrollon nëse faqja është në indeksin e Google apo jo. Kjo do të thot që faqja ka popullaritet dhe nuk e ka jetëgjatësinë e shkurtër siç e kanë faqet phishing.

$$\text{Rregulla:} \begin{cases} \text{Gjendet në indeksin e google} \rightarrow \text{Legjitime} \\ \text{Përndryshe} \rightarrow \text{Phishing} \end{cases} \tag{21}$$

Veçoria e bazuar në statistika dhe raporte

Kompani të ndryshme si PhishTank apo StopBadware formojnë raportime statistikore duke ekspozuar IP dhe domain-et më të përdorura për phishing. Kjo veçori kontrollon nëse hosti bën pjesë në këto raportime.

$$\text{Rregulla:} \begin{cases} \text{Hosti bën pjesë në raportimet e kompanive anti} - \text{phishing} \rightarrow \text{Phishing} \\ \text{Përndryshe} \rightarrow \text{Legjitime} \end{cases} \tag{22}$$

5.2.3. Përzgjedhja e algoritmit optimal për rastin konkret

Për këtë punim baçelor të diplomës në shqyrtim janë marrë tre algoritma për klasifikim, ata në: SVM , kNN dhe Random Forest. Të gjithë do të testohen dhe do të përzgjedhet algortimi i cili performon më mirë. Në Fig. 14 janë paraqitur rezultatet e saktësisë së algoritmeve duke e ndarë dataset-in në proporcione të ndryshme : 50/50 , 70/30, 90/10.

	50/50	70/30	90/10
SVM	87.39	88.13	93.08
kNN	78.47	81.05	90.26
Random Forest	87.92	93.81	96.88

Figura 14 Statistikat e testimeve të tre algoritmeve.

Në bazë të statistikave të nxerra shihet se algoritmi Random Forest është në vendin e parë për nga saktësia. Efiçienca e algortimeve në ML matet edhe me disa parametra tjerë siç janë: precision score, recall score dhe matrica e konfuzionit (confusion matrix).

Matrica e konfuzionit paraqet performancën e algoritmit duke treguar numrin rasteve TN, FP, FN, TP. True Negative (TN) është rasti kur parashikimi është negativ dhe vlera e vërtet është negative. False Positive (FP) është rasti kur parashikimi është pozitiv mirëpo vlera e vërtet është negative. False Negative (FN) është rasti kur parashikimi është negativ kurse vlear e vërtet është pozitive. Dhe rasti i fundit True Positive (TP) është kur parashikimi është pozitiv dhe vlera e vërtet është pozitive. [18].

Precision score paraqet proporcionin në mes të parashikimeve pozitive të sakta dhe numrit total të parashikimeve pozitive. $Precision = TP/TP+FP$. [18].

Recall score paraqet proporcionin në mes parashikimeve pozitive të sakta dhe numrit se sa raste pozitive në total. *Recall = TP/TP+FN*. [18].

Në Tabelën 1 janë paraqitur rezultatet e testimit të tre algoritmeve me ndarje të dataset-it 90/10, sipas parametrave të lartë cekur.

	Precision	Recall	TN	FP	FN	TP
SVM	0.913	0.969	422	57	19	601
kNN	0.901	0.904	431	48	59	561
Random Forest	0.921	0.970	435	44	17	603

Tabela 1 Rezultatet e testimit të algoritmeve për precision, recall, confusion matrix.

Në bazë të rezultateve të nxjerra nga testimi shihet se Random Forest është algoritmi më i saktë për këtë rast. Dhe duke marrë parasysh veçorinë “bagging” të këtij algoritmi e cila e mitigon problemin e overfitting, vegla do të zhvillohet me algoritmin Random Forest.

5.2.4. Zhvillimi i komponentës “server-side” për klasifikimin e faqeve

5.2.4.1. Trajnimi i modelit klasifikues

Si është cekur edhe në kapitullin 4.1 trajnimi i modelit do të bëhet me gjuhen Python përmes librarisë scikit-learn. Për arsye të leximit më të lehtë të dataset-it nga Python, fillimisht dataset-in i cili është në formatin .csv, do ta konvertojmë në formatin .ARFF. ARFF është shkurtesë e cila do të thotë “Attribute-Relation File Format”, është një text file i formatit ASCII i cili u zhvillua nga Universiteti i Waikatos. Ky format ka 2 pjesë kokën ose header-in dhe të dhënat ose data. Të dhënat duhet të transformohen në array 2d për librarinë sklearn, këtë proces na ndihmon ta kryejm libraria numpy.

```
labels = []
data_file = open('dataset/dataset.arff').read()
data_list = data_file.split('\n')
data = np.array(data_list)
data_array = [i.split(',') for i in data]
data_array = data_array[0:-1]
for i in data_array:
    labels.append(i[30])
data_array = np.array(data_array)
features = data_array[:, :-1]
```

Kodi 1 Përgatitja e të dhënave për sklearn.

Një avantazh tjetër i algortmit Random Forest është se nuk ka nevoj për hyperparameter tuning[8]. Sipas testimeve me parametrave të lojeve të ndryshëm është nxerrur konkludimi që me parametrat e parazgjedhur ose default, algoritmi jep rezultatet më të mira.

```
Clf4 = RandomForestClassifier(verbose=2, oob_score=True)
clf4.fit(features_train, labels_train)
```

Kodi 2 Deklarimi i klasifikuesit dhe parametrave të tij.

Si shihet nga Copëza e kodit 2, parametrat e vetëm janë “verbose=2” dhe “oob_score=True” , të dy këta parametra ndikojnë vetëm në dhënien e informacioneve gjatë dhe pas përfundimit të trajnimit. Megjithëse për trajnim janë përdorur parametrat e parazgjedhur, do të diskutohen disa nga më kryesorët në vazhdim.

n_estimators *int, default=100* :

tregon numrin e pemëve të konstruktuara, në versionin e kaluar të librarisë është rritur numri i paracaktuar nga 10 në 100. Sipas testimeve çdo numër më i madh se 100 vetëm merr më shumë kohë për trajnim kurse nuk rritet saktësia.

[8] Hyperparameter tuning nënkupton zgjedhjen e parametrave për një algoritëm të ML, me qëllim rritjen e saktësisë.

min_samples_split *int or float, default=2* :

tregon numrin minimal të mostrave që kërkohen për të ndarë një nyje të brendshme.

criterion*{“gini”, “entropy”}, default=”gini”* :

tregon funksionin me të cilin kalkulohet kualiteti i ndarjes së një nyje. Më i suksesshëm për rastin tonë është treguar funksioni i parackatuar Gini.

Pas përfundimit të testimit në Fig.15 shihen rezultatet e testimit. Ndër to shihet precision, recall dhe matrica e konfuzionit

```
Accuracy is :96.88828584350972
========================================
Precision Score: 0.9673580961569385
========================================
Recall Score: 0.9770992366412213
========================================
Confusion Matrix: [[4695  203]
 [ 141 6016]]
========================================

Process finished with exit code 0
```

Figura 15 Rezultatet e trajnimit të klasifikuesit me algoritmin Random Forest

Algoritmi Random Forest na ofron edhe rëndësinë e secilës veçori nga trajnimi. Përmes objektit *feature_importance_* rangohen veçorit në bazë se sa kanë qenë informative gjatë trajnimit.

Këto përqindje algoritmi i kalkulon nga MDI (Mean Decrease in Impurity) pra sa e zvogëlon papastërtinë e dataset-it ajo veçori. Në Fig.16 mund të shihet rangimi i veçorive sipas MDI.

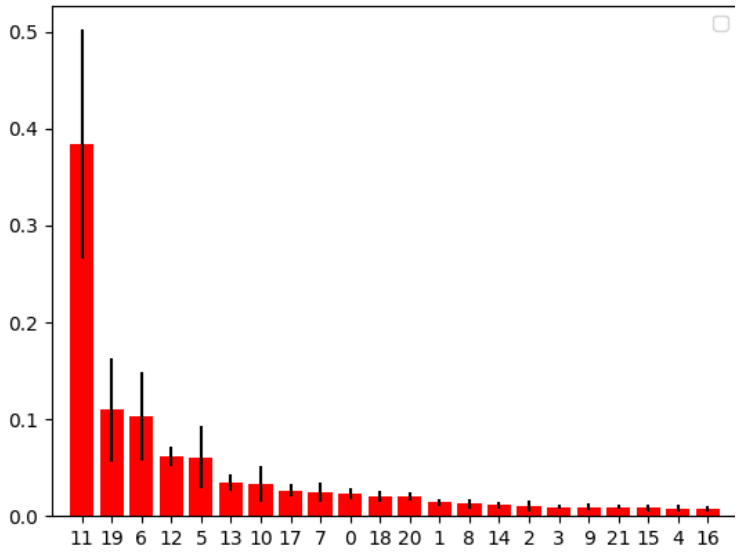

Figura 16 Rëndësia e vetive bazuar në MDI

Legjenda:

0-Përdor Ip adresë në vend të

1-URL i gjatë

2-Ka përdorur servise të shkurtimit të URL

3-URL përmban simbolin “@”

4-Ridirektim duke përdorur “//”

5-URL përmban simbolin “-”

6-Ka shumë subdomains

7-Gjatësia e regjistrimit të domainit

8-Ka Favicon

9-Ekzistenca e tokenit HTTPS në pjesën e domain-it

10-Ka shumë Request URL

11-URL i tagut anchor “<a>”

12-Linqet në tagjet <Meta>, <Script> and <Link

13-Server Form Handler (SFH)

14-Dërgimi i informacioneve në email

15-URL e pazakon

16-Përmban tagun iframe

17-Jetëgjatësia e domain-it

18-Regjistri DNS

19-Trafiku i faqes

20-Google Index

21-Veçoria e bazuar në statistika dhe raporte

Një problem i paraqitur kohëve të fundit është se rëndësia e veçorive bazuar në papastërti mund të fryjë rëndësinë e veçorive numerike. Për më tepër, këto rëndësi vuajnë nga llogaritja në statistikat e nxjerra nga të dhënat e trajnimit, përqindjet mund të jenë të larta edhe për karakteristikat

që nuk janë parashikuese të ndryshores së synuar, për sa kohë që modeli ka aftësinë të kaloj në overfit. Përveq kësaj këto statistika nxirren nga të dhënat e trajnimit. Si alternativë mund të llogariten rëndësia e permutacioneve (permutation importances) të cilat kalkulohen nga të dhënat e testimit.

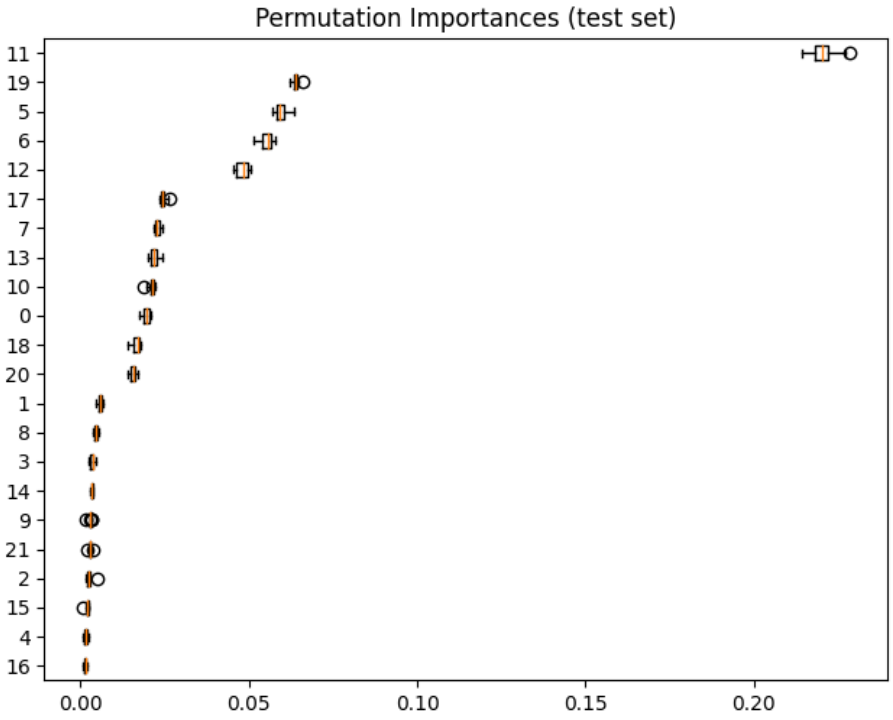

Figura 17 Rëndësia e veçorive bazuar në permutacione

5.2.4.2. Detektimi i faqeve për phishing

Pas tajnimit të modelit për klasifikim mund të bëhet testimi i faqeve për phishing. Skripta kryesore që ekzekutohet është *main.py* e cila merr 1 argument, URL e faqes. Në copëzën e kodit 3 mund të shihet funksioni main i cili bën klasifikimin e faqeve duke marr URL si argument dhe përmbajtjen e faqes nga fajlli *innerHTML.txt* të cilin i`a ofron webserveri. Kjo skriptë thërret skriptën *features_extraction.py* e cila nxjerr veçoritë e faqes duke shfrytëzuar URL dhe përmbajtjen e faqes.

```
def main():
    url = sys.argv[1]
    features_test = features_extraction.main(url)
    # 2d array per scikit-learn
    features_test = np.array(features_test).reshape((1, -1))
    clf = joblib.load(LOCALHOST_PATH + DIRECTORY_NAME +
'/classifier/random_forest_new_4.pkl')
    prediction = clf.predict(features_test)
    prediction_int = int(prediction[0])
    probability = clf.predict_proba(features_test)
    respond_data = {"features": features_test.tolist(), "probability":
probability.tolist(),
                    "prediction": prediction_int, "url": url}
    json_respond_data = json.dumps(respond_data)
    print(json_respond_data)
```

Kodi 3 Funksioni main i cili bën klasifikimin e faqeve

5.2.5. Zhvillimi i komponentës "client-side"

Komponenta me të cilën përdoruesi do të ketë interaksion do të zhvillohet si Chrome Extension. Zhvillimi i Chrome Extension është jashtëzakonisht i thjeshtë, tre elementet që duhen për një extension funksional janë : popup.html ku definohen elementet HTML , popup.js ku shkruhen funksionet në javascript dhe manifest.json ku shkruhen detajet e extesion.

```
{
    "manifest_version": 2,

    "name": "NoPhish",
    "description": "This extension will analyze a page and predict if it is a
phishing site or not , by using machine learning!",
    "version": "0.8",

    "browser_action": {
```

```
    "default_icon": "icon.png",
    "default_popup": "popup.html"
   },
   "icons": { "16": "icon.png",
       "48": "icon48.png",
      "128": "icon128.png" },
   "permissions": [
    "activeTab"
    ]
}
```

Kodi 4 Fajlli manifest.json i detyrueshëm për funksionimin e Chrome Extension

Qëllimi kryesor i Chrome Extension është të jetë sa më i thjeshtë në përdorim. Dizajnimi është bërë në mënyrë që me një klimin të butonit të merren të gjitha informacionet e nevojshme. Në Fig.18 shihet pamja e parë e veglës.

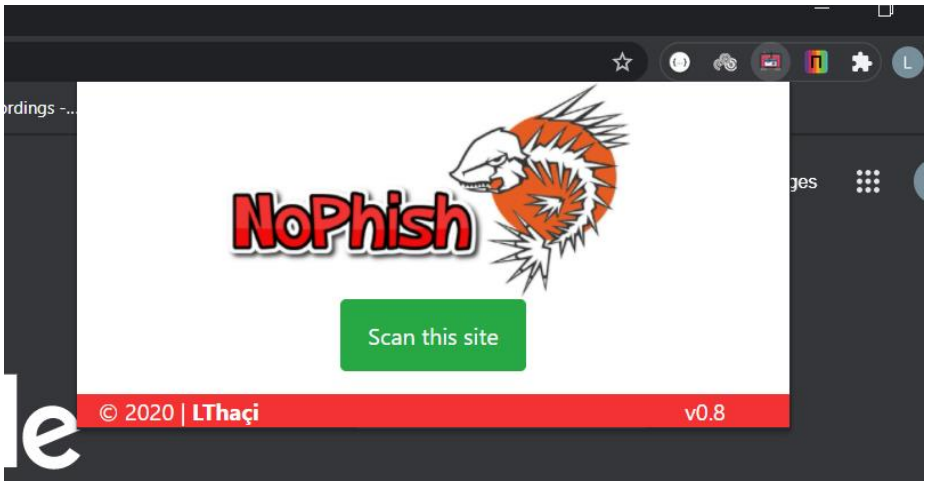

Figura 18 Pamja e parë e veglës

Butoni “Scan this site” dërgon kërkesë në webserver me URL e faqes ku gjindemi aktualisht dhe pret përgjigje. Pas kthimit të përgjigjes nga webserveri, si shihet në Fig.19 filimisht shfaqet parashikimi i cili lajmëron përdoruesin se a e sheh vegla faqen si phishing apo jo. Shfaqet probabiliteti i llogaritur se sa i dyshimtë është faqja dhe po ashtu për secilën veçori të faqes tregohet se cila ka dështuar gjatë testimit.

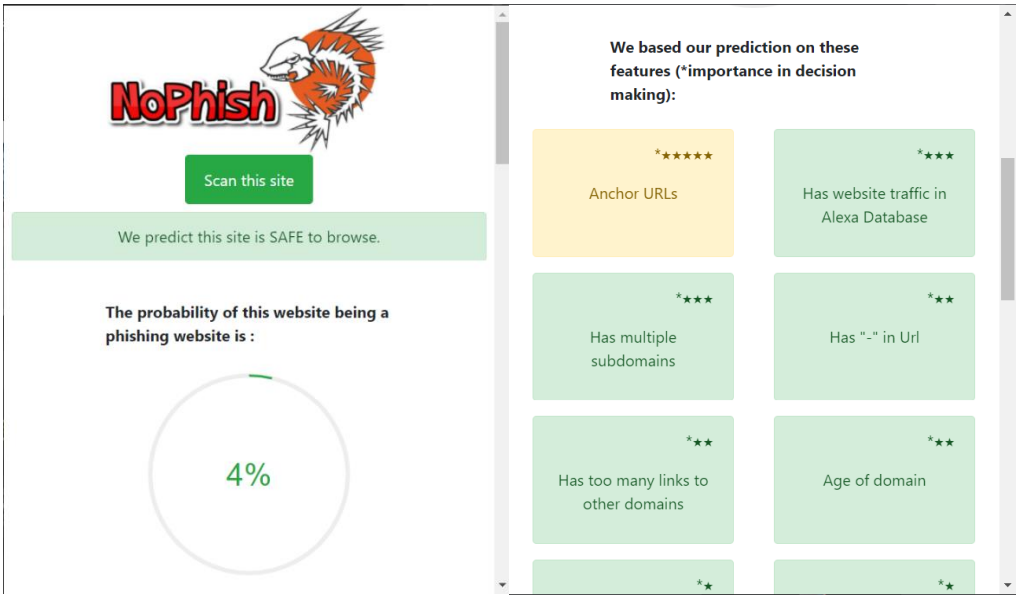

Figura 19 Pamja e veglës pas parashikimit

Duke u bazuar në parimin se më pak i dëmshëm është një parashikim false positive se sa një parashikim false negative, përveq parashikimit se a është phishing apo jo, është shtuar edhe një zonë e paralajmërimit. Kjo zonë tregon se faqja parashikohet si e sigurtë mirëpo ka shenja të dyshimta në disa veçori, prandaj kërkohet vëmendje nga përdoruesi të rishikoj se në cilën faqe është. Përveç rastit kur parashikimi kthehet si safe që shihet në Fig.19 , në Fig.20 mund të shihen edhe dy rastet tjera.

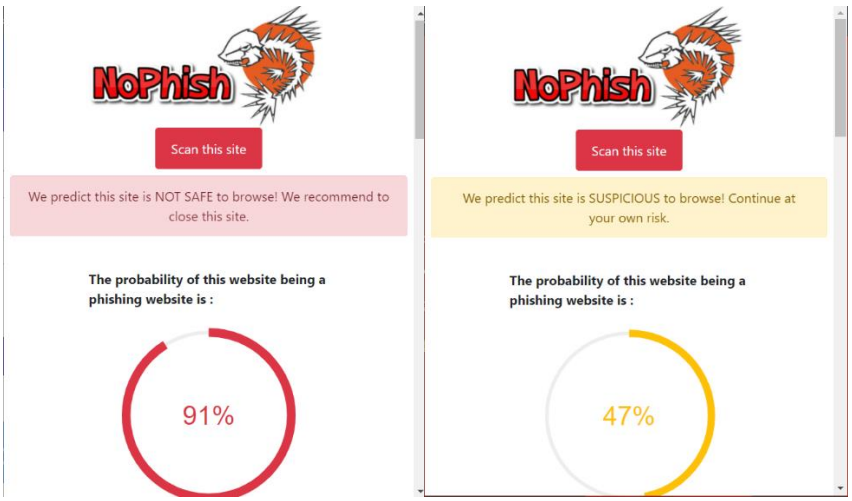

Figura 20 Pamjet e veglës kur parashikimi është si phishing apo si faqe e dyshimtë

5.2.6. Zhvillimi i webserverit për ndërlidhjen e dy komponentave.

Webserveri do të shërbej si një urë lidhëse në mes dy komponentave, dhe njëkohësisht në të krijohet ambienti i duhur për ekzekutimin e skriptave të klasifikuesit në Python, pasi që një gjë e tillë nuk lejohet në ambientin e Chrome Extension. Webserveri është i zhvilluar në gjuhën Node.Js dhe është mjaftë i thjeshtë në parim. Ai pranon një URL nga kërkesa, përmes librarisë *get-content* [19] merr përmbajtjen e faqes dhe e ruan në innerHTML.txt, pastaj krijon një proces të pythoni me skriptën main.py nga e cila merr rezultatet.

```
get(url).then((pageContent) => {
          fs.writeFileSync('../innerHTML.txt',pageContent)
          const pythonProcess = spawn('python',["../main.py",url]);
          pythonProcess.stdout.on('data', (data) => {
            response.writeHead(200, {'Content-Type': 'application/json'});
            response.end(data.toString('utf8'));
          });
        })
```

Kodi 5 Ekzekutimi i klasifikuesit në webserver dhe kthimi i rezultateve

5.3. Punë eksperimentale

Përveç testimeve dhe nxjerrjes së rezultateve së saktësisë nga dataset-i, është e nevojshme që vegla të testohet se si do të performojë në një skenar real. Për të krijuar skenarin e përdorimit nga një përdorues i zakonshëm, janë përzgjedhur njëzet e shtatë (27) webfaqe. Katërmbëdhjetë (14) nga ata janë phishing të verifikuar nga PhishTank, kurse trembëdhjetë (13) janë webfaqe të besuara. Përveç testimit të veglës NoPhish janë përzgjedhur edhe dy vegla tjera në mënyrë që të krahasohet performanca me veglat të cilat janë aktualisht në përdorim. Këto dy vegla janë : Google Safe Browsing dhe Web Of Trust (WoT). Në figurën në vijim janë paraqitur rezultatet e testimit.

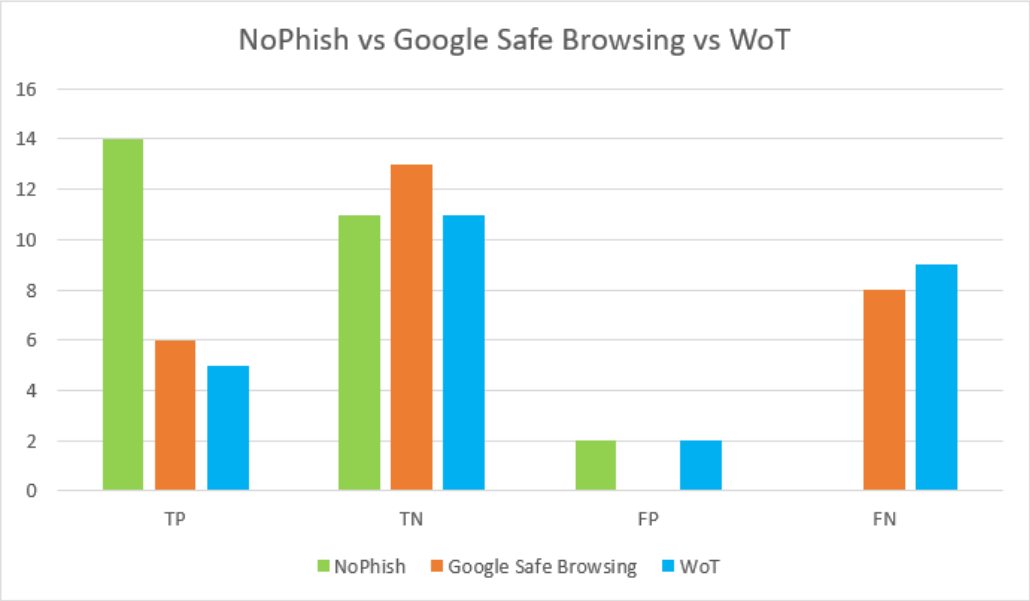

Figura 21 Testimi i performancës së NoPhish dhe krahasimi me vegla si Google Safe Browsing dhe WoT

Nga rezultatet e nxerra nga testimit shihet se NoPhish performon shum më mirë se dy veglat e tjera. Nga grafi mund të konkludojmë se NoPhish në numrin e rasteve FP (False Positive) nuk performon më mirë se dy veglat tjera, mirëpo realisht dy rastet FP kanë qenë vetëm warnings, pra nuk kanë qenë parashikimet false positive nga vegla. Kjo është pikërisht pjesa konservative e veglës e cila mundohet ta vendos përdoruesin në anën e sigurt, dhe po ashtu kjo lejon që vegla të ketë **0 parashikime FN** të cilat janë parashikimet më të rrezikshme nga veglat anti-phishing.

6. Diskutime dhe Konklukzione

6.1. Reflektim kritik

Duke marrë parasysh të gjitha statistikat për sulmet phishing të cekura në këtë punim, shihet se është një ndër sulmet më të rrezikshme kibernetike sot. Meqenëse cak të sulmit e ka mungesën e vetëdijes së përdoruesit atëherë patjetër duhet të zhvillohen softuer të cilët futen në mes sulmuesit dhe përdoruesit.

Numri jashtëzakonisht i madh i të dhënave që na ofrohen sot në Internet dhe gjuhët programuese në kombinim me numrin e madh të librarive të cilat i lehtësojn shum punët e vështirat, na krijojnë një ambient motivues për hulumtim më të gjerë në këtë fushë. Si është cekur dhe më lartë në këtë punim sulmuesit janë gjithmonë një hap para, me drejtimin e Machine Learning mund të zvogëlojmë këtë largësi në rastin më të mirë të jemi në hap të njejtë. Hulumtimi në këtë fushë qoftë në zbulimin e metodave të reja për detektimin e faqeve phishing apo përmirësimin e metodave aktuale nuk është vetëm e mirëpritur por e nevojshme! Zhvillimi i një softueri i cili ka aftësi të parandaloj sulmet “zero-day”, nuk limitohet në domain specifik dhe është lehtë i përdorueshëm si nga kompanitë e mëdha ashtu edhe nga individët do të ishte ideal.

Nga rezultatet e nxerra nga ky punim baçelor i diplomës shihet se rruga e zhvillimit të një klasifikuesi me Machine Learning jep rezultate të kënaqshme. Përveç rezultateve të mira po ashtu shihet se ka një potencial të madh për hulumtim më të gjerë. Numri i madh i algoritmeve për Machine Learning dhe mundësitë e gjera për optimizimin e tyre lënë një hapësirë të zbrazët për një punë të madhe në këtë aspekt.

6.2. Përmirësimet e mundshme

Për kundër që rezultatet e këtij punimi janë të kënaqshme duke arritur saktësi të lartë me një numër të vogël të parashikimeve false negative, shihet se kjo temë ka shumë potencial për hulumtim më të gjerë dhe për përmirësime. Duke marrë parasysh që sulmet gjithmon evoluojn dhe janë nga më të ndryshmet duhet bërë përpjekje që vegla mos të mbetet mbrapa sulmeve më të reja.

Ideja e parë për një ndryshim potencial në të ardhmen është kombinimi i dy metodave të detektimit të faqeve phishing në një. Pra krijimi i një modeli hibrid të veglës duke kombinuar metodën blacklist me metodën Machine Learning. Në këtë mënyrë vegla fillimisht do të shikonte se nëse URL i caktuar ndodhet në listën e zezë të antivirusëve, nëse po atëherë do të lajmëronte përdoruesin për rrezikun, nëse jo atëhere do të kryente një skanim të faqes me klasifikuesin përmes Machine Learning në mënyrë që të tentoj të parandaloj ndonjë sulm “zero-day” potencial. Avantazhet e këtij lloji hibrid të veglës do të ishin rritja e sigurisë përmes shtimit të një shtrese shtesë të sigurisë në kontekstin e metodës blacklist. Kurse disavantazhet do të ishin ulja e shpejtësisë së parashikimit pasiqë kërkimi në databazat e mëdha të antiphishing kërkon kohë dhe resurse.

Sa i përket përmirësimeve në metodën Machine Learning padyshim se ka hapësirë të majftueshme. Megjithëse algoritmi Random Forest rezultoi më i suksesshëm gjatë këtij punimi, nuk do të thotë që në të ardhmen në rrethana tjera me evoluim të madh të sulmeve nuk mund të jetë një algoritëm tjetër më i suksesshëm. Veçoritë në bazë të cilave kryhet parashikimi duhet të jenë

gjithmonë në hap me kohën. Si për shembull veçoria me numër katër ku nëse URL përmban “//” mirret si i dyshimt mund të largohet në të ardhmen. Për arsye se disa nga shfletuesit në ditët e sotme veq se e kanë ndaluar ridirektimin në faqe tjera nëse URL përmban “//”, mirëpo derisa të jetë kjo siguri në të gjithë shfletuesit nuk mund të anashkalohet.

Për përfundim, ky punim i temës baçelor ishte shum atraktivë për të u punuar dhe mendoj se ka mundësi shkallëzimi në diçka shum më të madhe se kaq.

Bibliografia dhe referencat

[1] Business Email Compromise The $26 Billion Scam 2019. Gjendet në: https://www.ic3.gov/Media/Y2019/PSA190910 .

[2] Wandera,"Mobile Threat Landscape Report 2020" 2020. Gjendet në: https://www.wandera.com/mobile-threat-landscape/.

[3] APWG Phishing Activity Trends Report Q3 2020. Gjendet në: https://docs.apwg.org/reports/apwg_trends_report_q3_2020.pdf.

[4] W3Counter Browser & Platform Market Share November 2020. Gjendet në: https://www.w3counter.com/globalstats.php.

[5] Vade Secure Infographic Phishers Favorite Top 25 Q3 2020. Gjendet në: https://www.vadesecure.com/wp-content/uploads/VS_Infographic_Phishers_Favorites_Q3_2020_EN.pdf.

[6] Kay, R (2004) Sidebar: The Origins of Phishing. [Online]. Gjendet në: http://www.computerworld.com/s/article/89097/Sidebar_The_Origins_of_Phishing.

[7] B, Prabadevi & Nagamalai, Jeyanthi & Udzir, Nur & Nagamalai, Dhinaharan. (2019). Lattice Structural Analysis on Sniffing to Denial of Service Attacks.

[8] Jagatic, Tom; Nathaniel Johnson; Markus Jakobsson; Filippo Menczer (October 2007). "Social Phishing". Communications of the ACM.

[9] Business Intelligence and its relationship with the Big Data, Data Analytics and Data Science , Armando GeekStyle. Gjendet në: https://www.linkedin.com/pulse/business-intelligence-its-relationship-big-data-geekstyle

[10] A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow : Concepts, Tools, and Techniques to Build Intelligent Systems*. Sebastopol: O'Reilly Media, Incorporated, 2017.

[11] Automatic Log Analysis using Deep Learning and AI , Jagreet Kaur Gill. Gjendet në: https://www.xenonstack.com/blog/log-analytics-deep-machine-learning/.

[12] Support Vector Machine - Classification (SVM). Gjendet në: https://www.saedsayad.com/support_vector_machine.html.

[13] A. Ng. What is Machine Learning? Gjendet në: https://www.coursera.org/learn/machine-learning/lecture/Ujm7v/what-is-machine-learning.

[14] Random Forest Template for TIBCO Spotfire, Venkata Jagannath. Gjendet në: https://community.tibco.com/wiki/random-forest-template-tibco-spotfire

[15] Phishing Websites Data Set , UCI. Gjendet në: https://archive.ics.uci.edu/ml/datasets/phishing+websites

[16] WHOIS , Databaza e cila ofron informacione mbi domain-in . Gjendet në: https://who.is/

[17] Databaza Alexa (Alexa the Web Information Company., 1996). Gjendet në: https://www.alexa.com/

[18] Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures . Gjendet në : https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures

[19] Libraria get-content gjendet në: https://www.npmjs.com/package/get-content.

[20] LEO BREIMAN, "Random Forest" , Machine Learning, 45, 5–32, 2001

[21] Andreas C. Müller and Sarah Guido, "Introduction to Machine Learning with Python"

[22] Giuseppe Bonaccorso, "Mastering Machine Learning Algorithms".

[23] Giuseppe Bonaccorso, "Machine Learning Algorithms" , Një linjë : faqe 19.

[24] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The Elements of Statistical Learning. Springer, 2st edition, 2017.