Детекција малициозних *Phishing e-mail* порука

Бранислав Рољић
Факултет техничких наука
Универзитет у Новом Саду
Трг Доситеја Обрадовића 6
21000 Нови Сад
е-пошта: roljic.r26.2023@uns.ac.rs

Сажетак---Phishing **представља једну од најпопуларнијих врста сајбер криминала, док је најчешћа техника спровођења овог напада путем *e-mail* порука. У овом раду поређене су методе детекције *phishing e-mail* порука користећи комбинације техника обраде природног језика(TF-IDF, Word2Vec, BERT и LSTM) и алгоритама машинског учења(*Random Forest, Decision Tree, Logistic Regression, Gradient Boosting Trees* и *Naive Bayes*) са циљем откривања правилности у подацима које нису могле бити детектоване традиционалним *blacklist* приступима. Посебна пажња посвећена је ручном одређивању скупа атрибута, те имплементацији прикупљања и селекције најзначајнијих атрибута. Урађена је евалуација добијених модела којом се показује да BERT *Longformer* и TF-IDF (у комбинацији са алгоритмима машинског учења) дају најбоље резултате у свим метрикама, али је показан и велики потенцијал у тренирању LSTM-а.**

Кључне речи: Natural Language Processing, Machine Learning, Phishing, Email, TF-IDF, Word2Vec, BERT, LSTM, Random Forest, Logistic Regression, Gradient Boosting Trees, Naive Bayes

1 Увод

Развојем информационо-комуникационих технологија присутан је и софистицирани развој променљивих претњи који као платформу користе рачунари и мреже. Оваква врста криминала позната је под именом сајбер криминал (енг. cyber criminal).

Међу најпопуларнијим врстама сајбер криминала издваја се *Phishing* који представља врсту сајбер криминала у којој се путем различитих канала комуникације краду осетљиве и поверљиве информације људи (бројеви картица, лозинке или банковне информације). Далеко најпопуларнија техника спровођења *phishing* напада јесте путем *e-mail* порука.

Упркос огромним напорима академске заједнице и индустрије последњих година, истраживање против *phishing*-а се и даље суочава са разним изазовима.

Phishing напад представља скалабилан чин обмане при чему се за добијање информација од мете користи лажно представљање [1]. *Phishing* представља форму социјалног инжењеринга који покушава да дође до осетљивих или поверљивих података "маскирањем" у ентитет од поверења, на тај начин обмањујући корисника да отвори *e-mail*, инстант или текстуалну поруку. Прималац затим бива "намамљен" да кликне на малициозни линк, преузме датотеку или попуни обрасце, што може проузроковати инсталацију малвера, "замрзавање" система као део *ransomware* напада или откривање осетљивих информација.

Како би извршили *phishing* напад, малициозни корисници морају разумети основне начине размишљања и понашања људи да би употребом психологије навели

њихову жртву да открије осетљиве информације. Неки нападачи покушавају изазвати осећај хитности код жртве претећи да ће прекинути приступ налогу или услузи, као што је банковни рачун, осим ако не открију одређене (осетљиве) информације. Алтернативно, нападачи могу најавити приход новца или повраћај средстава уколико корисник открије свој број рачуна, лозинку и сл.

2 Претходна решења

У раду [2] се систематски прегледају и синтетизују истраживања о употреби NLP-а за откривање *phishing e-mail* порука. На основу унапред дефинисаних критеријума, идентификовано је и анализирано укупно 100 истраживачких чланака објављених између 2006. и 2022. године. Проучаване су кључне области истраживања у откривању *phishing e-mail* порука употребом NLP-а. Разматрани су алгоритми машинског учења и NLP технике које се користе у откривању *phishing e-mail* порука, текстуалне функције у *phishing e-mail* порукама, скупови података и ресурси који су коришћени у *phishing e-mail* порукама као и критеријуми евалуације. Значај овог рада лежи у чињеници да представља својеврсни скуп научних радова на тему детекције *phishing e-mail* порука. Кроз детаљну анализу извршену од стране аутора, добијамо јасну слику о водећим трендовима по питању употребе NLP техника, алгоритама машинског учења, избора корпуса и метода евалуације.

Рад [3] истражује ефикасност различитих комбинација NLP техника (TF-IDF, Word2Vec и BERT) и алгоритама машинског учења (RF, DT, LR, GBT, NB) у откривању *phishing* e-mail порука. Употребом TF-IDF, Word2Vec и BERT вршена је екстракција карактеристика из тела *e-mail* порука а потом су креирана три различита скупа карактеристика. Затим је сваки скуп карактеристика обрађен помоћу хи-квадрат теста како би се идентификовале најинформативније карактеристике које ће бити коришћене за RF, DT, LR, GBT и NB у сврху класификације *e-mail* порука као малициозних или бенигних. Коришћени скупови података су *Enron* (бенигни) и *Nazario* (малициозни). Евалуирани су NLP/ML комбинације кроз два експеримента са балансираним и не балансираним односом малициозни и бенигних порука. Мере перформанси укључују Recall, Precision, Accuracy, F1-Score, FPR, FNR и AUC. Коришћени алгоритми су LR, DT, RF, GBT и NB.

Резултати показују да су сви ML алгоритми са TF-IDF постигли тачност изнад 90%. Word2Vec је донео боље резултате, са тачношћу изнад 95%. BERT је показао најслабије резултате у поређењу са TF-IDF и Word2Vec. Иако се очекивало да ће BERT бити доминантан у откривању *phishing e-mail* порука, из овог рада видимо да то не мора бити случај. Битно је нагласити да су аутори истраживали само тело *e-mail* порука, остављајући простор за проширење рада на друге карактеристике као што су заглавља, прилози и URL-ови.

У радовима [4] и [5] вршена је класификација *phishing e-mail* порука употребом скупа предефинисаних карактеристика које су се показале најзначајнијим у самом процесу детекције. Ови радови уводе концепт пондерисања појмова *phishing*-а, који процењује тежину појмова *phishing*-а у свакој *e-mail* поруци. У раду [5] се врши класично „пречишћавање“ карактеристика с циљем доласка до оних које носе највише информација, док се у раду [4] класификација *phishing e-mail* порука ослања се на модел откривања знања (KD) и рударења података за изградњу интелигентног класификатора *e-mail* порука који може класификовати нову *e-mail* поруку као легитимну или *phishing*. Предложени модел се гради применом итеративних корака KD-а ради идентификације и екстракције корисних карактеристика из скупа података тренирања *e-mail* порука. Осим конкретних резултата класификације, резултати од интереса за наш рад тичу се одабраних карактеристика које представљају резултат фазе екстракције карактеристика. У првом раду, коришћењем KD модела, издвојено је 16 најзначајнијих карактеристика, док је у другом раду, посматрањем фреквенција појављивања карактеристика у "*ham*" односно "*phishing*" *e-mail* порукама, пронађено 17 најзначајнијих

карактеристика.

Ова истраживања имају значај јер укључују анализу не само тела *e-mail* поруке као обичног текста, већ такође узимају у обзир значај информација из заглавља *e-mail* порука и садржаја у URL-овима који често чине део phishing e-mail порука. Издвојене карактеристике представљају својеврсно проширење рада [3].

3 Метод

Када је реч о NLP техникама, кориштени су:

TF-IDF - метод који се често користи у области претраге информација, рударења текста, а у последњих неколико година примењује се и у детекцији *phishing* е-порука [2].

Word2Vec - метод за обраду природног језика који захвата контекст речи, њене везе са другим речима, семантичку и синтаксичку сличност. Користи неуронску мрежу за учење веза из велике колекције текстова, као што је колекција е-порука у овом случају. Word2Vec ствара векторизоване репрезентације где су сличне речи блиске у *embedded* простору. Word2Vec има две имплементације: *Continuous Bag of Words* (CBOW) и *skip-gram*. CBOW предвиђа циљну реч на основу контекста, док skip-gram користи суседне речи за предвиђање циљне речи.

BERT је модел дубоког учења дизајниран за различите задатке обраде природног језика, укључујући разумевање језика, превођење и класификацију текста. Заснива се на трансформер архитектури и користи *self-attention* механизме како би се фокусирао на различите делове улазних података, у циљу разумевања реченице у којој се реч налази, посматрајући речи које долазе и пре и после дате речи. На пример, традиционални модели као што су TF-IDF и Word2Vec могу представити реч "account" исто и у реченици "account for my actions" и у "bank account" [3]. Међутим, BERT генерише различите репрезентације за сваку реч, узимајући у обзир околне речи у реченици. Ова способност подстиче боље разумевање семантике и веза између речи.

За разлику од трансформера, BERT користи само енкодер дио, док се декодер дио одбацује, па се BERT сматра *language-based* моделом, а не *sequence-to-sequence* моделом. Модел је претрениран на великом корпусу текста како би научио опште репрезентације језика, које се затим *fine-tune*-ју за специфичне задатке попут класификације текста, додавањем слојева специфичних за задатак. Ово *fine-tune*-овање омогућава BERT-у да ефикасно класификује нове, непознате текстове на основу разумевања језика стеченог током предтренирања.

LSTM је специјализована врста рекурентне неуронске мреже (RNN) дизајнирана да превазиђе ограничења традиционалних RNN-ова, нарочито проблем *vanishing gradient*-а. LSTM ово постиже укључивањем меморијских ћелија које могу задржати или заборавити информације током времена. Свака LSTM ћелија опремљена је са неколико врата (енг. *gate)*—улазна(енг. *input),* заборавна(енг. *forget)* и излазна(енг. *output)*—која регулишу проток информација. Улазна врата контролишу да ли ће нове информације бити додате у ћелију, заборавна врата одређују које информације из претходног стања треба задржати или одбацити, а излазна врата управљају информацијама које се преносе на следећи слој. Врата омогућавају LSTM мрежама да ефикасно ухвате дугорочне зависности у секвенцијалним подацима, што их чини посебно погодним за задатке као што су моделовање језика, препознавање говора и класификација текста. Задржавањем стања ћелије, LSTM-ови могу да запамте важне информације током дужих секвенци, пружајући значајно побољшање у односу на традиционалне RNN-ове.

У раду су обучавани различити алгоритми и обучени алгоритми су тестирани методама евалуације како би се добили оптимални резултати.

Алоритми који су коришћени:

Decision Tree:

- *Decision Tree* користи своје структуралне одлуке како би класификовао податке у складу са задатим параметрима.

Gradient Boosting Tree:

- *Gradient Boosting Tree* комбинује више слабих модела како би побољшао укупну предиктивну моћ.

Logistic Regression:

- *Logistic Regression* је линеаран

модел који се користи за бинарну класификацију.

Naive Bayes:

- *Naive Bayes* се ослања на Бајесову теорему са претпоставком о независности из међу карактеристика.

Random Forest:

- *Random Forest* је ансамбл метода која користи више стабала одлучивања како би побољшала стабилност и тачност.

Сву модели су фино подешавани кроз *GridSearchCV* како би се пронашли оптимални хипепараметри за побољшање перформанси алгоритама.

У овом пројекту су коришћене различите методе евалуације како би се добили увиди у перформансе коришћених модела машинског учења. Методе евалуације које су коришћене:

Accuracy:

Мера која оцењује колико тачно модел предвиђа класе. Иако је једноставна за разумевање, коришћена је како би се добио општи увид у тачност модела.

Precision:

Precision мери колико је модел тачан када предвиђа позитивне инстанце. Коришћена је да би се добили увиди у тачност предвиђања позитивних класа.

Recall:

Recall мери колико добро модел препознаје све позитивне инстанце. Употребљена је да би се добио увид у способност модела да открије све позитивне случајеве.

F1-score:

F1-score је хармонијска средина између прецизности и одзива. Коришћена је као компромисна мера која узима у обзир и *precision* и *recall*.

AUC-ROC:

Мера која оцењује способност модела да раздвоји класе. Коришћена је како би се добила комплетна слика о способности модела.

False Positive Rate и False Negative Rate:

Ове стопе су истакнуте како би се разумела различита ограничења и предности модела у контексту лажних предвиђања.

Све ове методе су коришћене због њихове комбинације и различитих аспеката које мере. Они омогућавају комплетну анализу перформанси модела и пружају различите угледе у његове јачине и слабости. Комбиновани приступ овим методама обезбеђује целокупну слику ефикасности различитих алгоритама у раду са конкретним скупом података.

Циљ овог истраживања је спровођење свеобухватне анализе различитих метода представљања и обраде података, као што су TF-IDF, Word2Vec, и скупова података са ручно извученим карактеристикама. Истраживање има за циљ поређење перформанси ових метода у оквиру различитих модела машинског учења са моделима дубоког учења – BERT и LSTM, те разумевање на који начин различите технике представљања података доприносе учинку модела.

3.1 Скуп података

Први корак у изградњи предложеног класификатора *phishing e-mail* порука је одабир одговарајућег скупа података који је прави узорак постојећих е-порука који се састоји и од малициозних и легитимних е-порука (познате и као *phishing e-mail* и *ham e-mail*).

Један скуп података представља Nazario [6] док је други SpamAssassin [7]. Мотивација за употребу ова два скупа лежи у чињеници да ова два скупа представљају два најчешће кориштена и најажурнија скупа у области детекције *phishing e-mail* порука.

Иницијална идеја је била да један скуп буде кориштен за тренинг и валидацију, а други за тестирање. Након детаљне анализе скупова података, и издвајања релевантних карактеристика, закључено је да овакав приступ даје лоше резултате. Разлог томе је значајна разлика у подацима, поготово у садржају тела поруке, што потиче од чињенице да и сами скупови не користе исте базе *e-mail* порука.

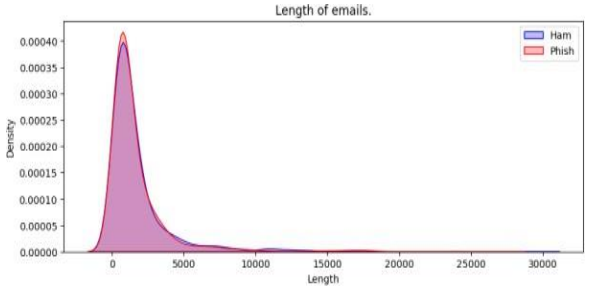
Као алтернатива овом приступу, изабрано је спајање ова два скупа, те накнадна подјела новодобијеног скупа на тренинг и тест. Да би спајање било могуће, било је неопходно довести скуп података *SpamAssassin* на исту структуру као и скуп *Nazario*. У вези са тим, вршена је екстакција URL-ова из тела поруке. Такође, скуп *Nazario* је балансиран, док *SpamAssassin* није, па је извршено његово балансирање *Random undersampling* методом. Потом је извршено спајање ова два скупа, чиме је добијен скуп података у коме се сваки податак у скупу састоји од пошаљиоца поруке, примаоца поруке, датума, наслова, тела *e-mail* поруке, URL-а и лабеле.

3.2 Претпроцесирање и ЕДА

Предобрада подразумева уклањање нерелевантних и сувишних информација из текстова наслова и тела е-поште, те даљу конверзију у униформан формат. Сврха овог задатка је олакшати екстракцију текстуалних карактеристика идентификујући најинформативније речи у корпусу е-поште.

Посматрањем дужина *e-mail* порука примећено је да постоје како *phishing* тако и *ham e-mail* поруке чија дужина значајно одудара од остатка података. Иако, доменски гледано, термин *outlier* по дужини поруке нема значаја, уклоњене су *phishing* и *ham* поруке чија дужина прелази 30000 карактера, чиме је убрзан процес претпроцесирања, а очувано је 97% података.

Иако важи правило да је дужина *ham e-mail* порука типично већа од дужине *phishing* порука, приметимо да у нашем скупу то није случај.



Слика 1. Приказ расподеле дужина *phishing* и *ham e-mail* порука

Следећи корак представља уклањање дупликата. Иако постоји значајан број порука које су дупликати по телу поруке, због значаја података у колонама “пошиљалац” и “наслов”, дупликатом су се сматрали само подаци који имају исте вредности у све три колоне, те су такви уклоњени. Такође, доменски гледано, колоне “прималац” и “датум” нису од значаја за детекцију *phishing* порука, те су наведене колоне уклоњене из скупа.

Након уклањања дупликата, вршено је издвајање HTML, URL, *e-mail* тагова и бројева из тела порука. Док традиционалан начин рада спроведен у радовима подразумева замену ових тагова празним стрингом, у овом раду је примењен приступ замене тагова генеричким називима “htmltag”, “urltab”, “emailaddresstag” и “digittag” чиме се очувавају информације о постојању наведених тагова, што доводи до бољих резултата.

Први корак претпроцесирања подразумева трансформацију садржаја у *lowercase*, експанзију контракција, те уклањање специјалних и *whitespace* карактера.

Наредни корак представља уклањање стоп речи и токенизацију. Токенизација подразумева процес раздвајања речи на основу делимитера и њихову конверзију у листу речи(токена). Из узетих токена се уклањају изузетно честе речи које немају велику инфорациону вредност. Уобичајене стоп речи укључују токене "the", "then", "he", ... итд. Овај корак помаже у смањењу сличности између е-порука и повећава перформансе предложеног модела, посебно у извршавању каснијих корака.

Касније, у фази претпроцесирања уводимо лематизацију. Лематизација подразумева смањивање инфлективних облика речи на њихове коренске облике, што доводи до мањег скупа речи који ће бити обрађен алгоритмима за обраду природног језика (NLP). Ова смањења догађају се јер се сви инфлективни облици речи конвертују у коренски облик. Иако је *stemming* алтернативни метод са сличним циљем, изабрана је лематизација због њене основе у морфолошкој анализи, што обезбеђује смислене облике речи.

За извршење лематизације, неопходан је вокабулар за идентификацију изворних форми речи, познатих као леме. За тај циљ, кориштена је WordNet база података. WordNet је широко коришћен лексички репозиторијум који успоставља семантичке везе између речи на преко 200 језика. У WordNet-у, речи су повезане кроз семантичке везе и организоване у концептуалне синонимне скупове (synsets) који изражавају значење концепта. Коришћење WordNet-а помаже у смањивању семантичке сложености текстова е-порука путем процеса лематизације.

Суштински, лематизација укључује замену група речи које су синоними и имају исту лему, чиме се смањује разноврсност текста. На пример, лема речи "better" је реч "good". Додатно, користимо обележавање делова говора (Part-of-Speech - POS) за указивање граматичке категорије сваке речи (именица, глагол, придев, прилог), што олакшава процес лематизације и додатно “усавршава” инстанце речи.

3.3 Екстракција карактеристика

Проблем детекције *phishing* е-порука је посматран као задатак класификације текста, што представља суштински део обраде природног језика (NLP). Стога, задатак екстракције текстуалних карактеристика примењује NLP методе за претварање резултата претпроцесирања текста наслова и тела е-поруке у карактеристике које ће обрадити ML алгоритми. На овај начин, наш приступ може бити примењен и на податке који не садрже приватне и осетљиве информације, као што су *e-mail* адресе, домени итд.

За извршење овог задатка, користили смо два метода: TF-IDF, Word2Vec. Ови методи играју кључну улогу у претварању текстуалног садржаја у формате прилагођене ML алгоритмима.

На текст е-поруке је примењен TF-IDF у циљу издвајања карактеристика заснованих на тексту. TF-IDF генерише тежину која указује на значај речи у колекцији текстова е-поште.

Након добијања резултата TF-IDF метода, употребом *chi-square* алгоритма је вршено издвајање најзначајнијих карактеристика, при чему је за постотак очуваних карактерисика узето 50%.

3.4 Мануелно изабране карактеристике

Осим “аутоматског“ издвајања карактеристика, вршено је и мануелно издвајање. Употребом резултата добијених у [4] и [5] те релевантних радова везаних за доменско знање, мануелно је издвојено 48 најзначајнијих карактеристика. У табели 1 су наведене издвојене карактеристике и њихово значење.

Карактеристика	**Опис**
body_html	Провера да ли тело садржи HTML
body_forms	Провера да ли тело садржи HTML форме
body_iframe	Провера да ли тело садржи *iframe*
body_scripts	Провера да ли тело садржи *script* таг
body_general_salutation	Провера да ли тело садржи термине попут: *Dear user/customer/buyer/account holder, Good, Greetings…*
body_popups	Провера да ли тело садржи *popup*-е
body_num_of_attachments	Број прилога у поруци
body_images_as_links	Број фотографија које се

	користе као линкови
body_Account	Провера да ли тело садржи термин *account*
body_Access	Да ли тело садржи термин *access*
body_Verify	Провера да ли тело садржи термин *verify*
body_Click	Провера да ли тело садржи термин *click*
body_Open	Провера да ли тело садржи термин *open*
body_Confirm	Провера да ли тело садржи термин *confirm*
body_Attachment	Провера да ли тело садржи термин *attachment*
body_Password	Провера да ли тело садржи термин *password*
body_Risk	Провера да ли тело садржи термин *risk*
body_Login	Провера да ли тело садржи термин *login*
body_Security	Провера да ли тело садржи термин *security*
body_Bank	Провера да ли тело садржи термин *bank*
body_Paypal	Провера да ли тело садржи термин *paypal*
body_Win	Провера да ли тело садржи термин *win*
body_richness	Однос броја речи и броја карактера тела
url_is_valid_url	Провера валидности URL-а
url_hexadecimal	Број хексдецималних карактера
url_domains_count	Број домена у URL-у
url_dots_count	Број тачака у домену URL-а
url_num_with_ip	Број URL-ова који садрже IP адресу
url_count_mismatched_urls	Број URL-ова код којих

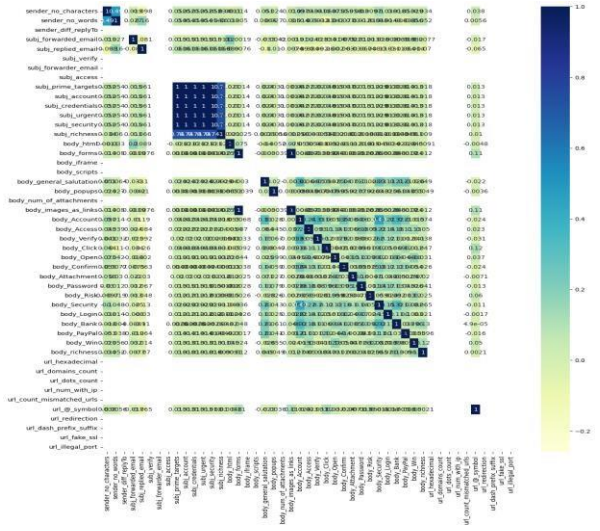
	линк унутар *href* тага није исти као URL
url_@_symbol	Провера да ли URL садржи @ симбол
url_redirection	Провера да ли је URL редирекција (садржи “//” у путањи)
url_dash_prefix_suffix	Провера да ли URL садржи “-” у домену
url_fake_ssl	Провера да ли је URL “лажни” *ssl* (*https* има редирекцију на *http*)
url_illegal_port	Провера да ли порт унутар URL не припада стандардним портовима (80, 443)
subj_forwarded_email	Провера да ли је *e-mail* прослеђен
subj_replied_email	Провера да ли је *e-mail reply*-ован
subj_verify	Провера да ли наслов садржи термин *verify*
subj_update	Провера да ли наслов садржи термин *update*
subj_access	Провера да ли наслов садржи термин *access*
subj_prime_targets	Провера да ли наслов садржи неки од термина: *"debit", "paypal", "bitcoin", "payoneer", "ebay", "amazon", "bank"*
subj_account	Провера да ли наслов садржи неки од термина: *"profile", "handle", "account", "deactivation"*
subj_credentials	Провера да ли наслов садржи неки од термина: *"credential", "password"*
subj_urgent	Провера да ли наслов садржи неки од термина: *"important","warning", "trouble", "urgent", "immediate"*

subj_security	Провера да ли наслов садржи неки од термина: *"security", "win", "update", "upgrade", "service"*
subj_richness	Однос броја речи и броја карактера наслова
sender_no_characters	Број карактера пошиљаоца
sender_no_words	Број речи пошиљаоца
sender_diff_replyTo	Провера да ли је *ReplyTo* адреса различита од домена пошиљаоца

Табела 1. Приказ издвојених карактеристика

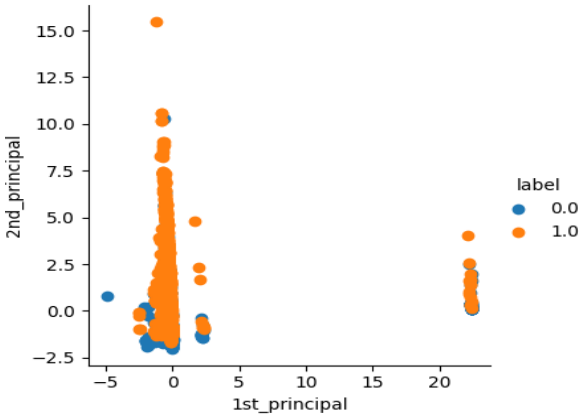
Након издвајања карактеристика, вршена је конструкција скупа података. Употребом *heatmap*-а се закључује да не постоји значајна корелација између карактеристика, али постоји одређена корелација између:

- Осетљивих речи унутар колоне "subject"
- Осетљивих речи унутар колоне "body"
- Popup-а, URL-ова са @ знаком и речи “click”
- *General salutation*-а и осетљивих речи



Слика 2. Корелација међу карактеристикама приказана употребом *heatmap*-а

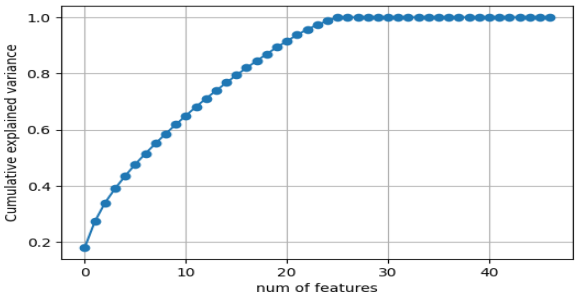
Такође, са слике 3 добијене путем PCA методе, видимо да је сепарација међу класама прилично слаба, што је и стварно стање, јер *phishing e-mail* поруке теже да буду што сличније *ham* порукама.



Слика 3. PCA визуализација

У циљу редукције димензионалности добијеног скупа података употребљене су *PCA, Variance Treshold* и *RandomForestClassifier* metode.

Са графа на слици 4 видимо да уколико изаберемо 19 карактеристика, биће очувано 90% варијансе.



Слика 4. Кумулативна варијанса

Након примене наведених метода, вршено је спајање њихових резултата, чиме је добијен скуп од 25 карактеристика које су прослеђене у наредну фазу.

3.5 Тренирање модела

Пошто су скупови података обрађени они се користе за обучавање модела који су описани у поглављу методе:

Decision Tree, Gradient Boosting Tree, Logistic Regression, Naive Bayes, Random Forest.
Тренинг скуп се дели на тренинг и тест скуп у 80:20 размери коришћењем train_test_split методе.

Узлаз су извучене карактеристике а излаз је лабела која говори да ли је мејл пхисхинг или хам.

Скуп података са ручно извученим карактеристикама се скалира користећи методу *MinMaxScaler* која врши нормализацију података који нису између 0 и 1.

Хиперпараметри модела су подешени коришћењем методе *GridSearchCV* која пролази кроз задати скуп хиперпараметара и бира ону комбинацију која даје највећу тачност. Ова метода је изабрана јер је време обучавања модела веома мало (1-10 секунди) па и тражење најбољих хиперпараметара не траје дуго (на доста слабој машини на којој је рађен пројекат 10-60 минута по алгоритму, у зависности од комплекности алгоритма).
Такође добро је што се хиперпараметри из скупа евалуирају у односу на метрику која се проследи као параметар GridSearchCV методи.

Методе евалуације су наведене у поглављу методе и оне су: *Accuracy, Precision, Recall,*
F1-*score*, AUC-ROC, FPR, FNR.

BERT

Поред различитих техника машинског учења, у циљу изградње високо-перформантног класификатора *phishing*-а у овом раду је коришћен и *Bidirectional Encoder Representations from Transformers* "BERT". Постоје два главна типа BERT-а, и то BERT *base* и BERT *large*. BERT *base* се састоји од стека од 12 енкодера из *Transformer* модела један изнад другог и има вектор величине 768, док BERT *large* има стек од 24 енкодера и величину вектора од 1024.

У овом раду је коришћен BERT *base* модел. Такође, због чињенице да је просечна дужина текста (значајно) већа од 512, а модел трансформера, због свог *self-attention* механизма који скалира квадратно са дужином секвенце, не може да обради дуге секвенце, коришћен је *Longformer*. *Attention* механизам *Longformer*-а скалира линеарно са дужином секвенце, што омогућава обраду значајно дужих докумената. *Attention* механизам *Longformer*-а је замена за стандардни *self-attention* и комбинује локални "прозорски" *attention* са глобалним "задатком мотивисаним" *attention*-оном.

У циљу развоја ефикасног класификатора *phishing e-mail* порука, вршен је *fine-tuning* BERT модела додавањем класификационог слоја „на врх" претходно обученог BERT модела. За ову сврху је кориштен *BertForSequenceClassification*, односно *LongformerForSequenceClassification*(у случају *Longformer*-а).

Горе наведени модели су у суштини стандардни BERT модели са додатним линеарним слојем дизајнираним за класификацију. Оваква конфигурација нам омогућава да користимо BERT као класификатор текста. Овај процес, познат као *fine-tuning* BERT-а, користи скривено стање (вектор величине 768) као улаз у класификациони слој, који затим производи резултат класификације, одређујући да ли је улаз *phishing e-mail* или не. Током обуке, и претходно обучени BERT модел и новододати класификациони слој се *fine-tune*-ују за класификацију *phishing*-а.

Први корак у процесу *fine-tuning*-а представља токенизација заглавља/тела *e-mail* порука употребом BERT токенизатора. За максималну дужину секвенце узимамо 512 за BERT *base*, односно 1024 у случају *Longformer*-а(због ограничених ресурса). Такође, додати су специјални токени: [CLS] који је потребан за класификацију и [SEP] који означава крај секвенце.

„Први токен сваке секвенце је увек специјални класификациони токен ([CLS]). Коначна скривена стања која одговарају овом токену користе се као агрегатна репрезентација секвенце за класификационе задатке." [8]. У случају да дужина улазне секвенце не одговара задатим вредностима, вршен је *padding* (коришћењем [PAD] токена) односно тримовање. Излаз енкодера представља секвенцу ID-јева и *attention_mask* токена који указују да ли је токен „прави" улаз или се ради о [PAD] токену.

Затим је вршена подела података на тренинг, валидационе и тест податке. За

тренинг је коришћено 70% података, за валидацију 10%, а за тестирање је коришћено 20% података.

Следећи корак представља дефинисање модела, оптимизационе и *loss* функције. За оптимизацију је коришћен *AdamW* оптимизатор са *learning rate*-ом иницијално подешеним на *2e−5*, као што је препоручено у раду [9]. У циљу „прилагођавања“ *learning rate*-а током процеса тренирања, употребљен је *learning rate scheduler*. Кориштен је линеарни *scheduler* који постепено смањује брзину учења линеарно од почетне вредности до нуле током обуке.

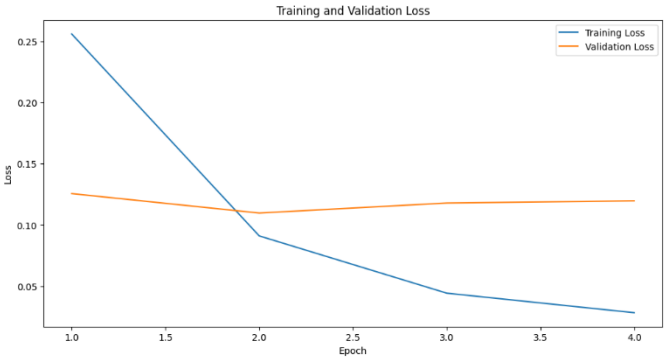
Због ограничених ресурса, величина *batch size*-а је узета као 16, док је број епоха 4. Као функција грешке изабрана је *Cross-entropy loss* функција која је погодна за класификационе задатке. Ова функција мери разлику између предвиђених вероватноћа и стварних лабела, подстичући модел да додели веће вероватноће исправној класи.

Процес обуке је подељен на два кључна подпроцеса: *feed-forward* и *backpropagation*.

У *feed-forward* подпроцесу, сваки улазни *e-mail* се обрађује кроз BERT модел, који распоређује тежине кроз своје слојеве како би произвео излаз. Разлика између овог излаза и стварне лабеле се рачуна као губитак(*енг. loss)*.

Током *backpropagation*-а, градијенти се рачунају на основу овог губитка, а тежине модела се ажурирају у циљу минимизације грешке, почињући од излазног слоја и пропагирајући назад до улазних слојева. Битно је нагласити да *BertForSequenceClassification*, и *LongformerForSequenceClassification* модели аутоматкси користе излазе [CLS] токена, те није потребно њихово мануелно “издвајање” из излаза BERT модела.

У току обуке, вршена је итерација у више епоха, ресетујући акумулирани губитак на почетку сваке епохе. За сваки *batch* података, улазни тензори су пребачени на GPU, градијенти су ресетовани, а затим је вршено рачунање излаза. *Cross-entropy loss* је коришћен за процену тачности предвиђања. Затим је вршен *backpropagation* у циљу ажурирања параметара модела користећи *AdamW* оптимизатор. *Learning rate scheduler* је динамички прилагођавао брзину учења како би побољшао конвергенцију.



Слика 5. BERT тренинг и валидациони губици

Уз тренирање модела, вршен је и процес валидације. Разлог за то је чињеница да је *validation loss* прецизнији показатељ него тачност, јер са тачношћу не гледамо тачну вредност излаза, већ само “на коју страну прага” пада. Са слике примећујемо да, иако *Training loss* опада, *Validation loss* расте након друге/треће епохе, што нам сугерише да модел може бити подложан *overfitting*-у. Слична ситуација је и код *Longformer* модела.

Након тренирања модела, вршена је евалуација перформанси модела коришћењем тестног скупа података. У процесу евалуације, моделу се предаје евалуациони(тестни) скуп података који садржи *e-mail*-ове и одговарајуће лабеле. Подразумевано, у овом случају се не врши ажурирање тежина. *E-mail* поруке се уносе у модел, а када модел класификује улазни *e-mail*, упоређујемо класификацију модела са одговарајућом улазном лабелом како бисмо израчунали перформансе модела.

LSTM

Рекурентне неуронске мреже (RNNs) су се показале као ефикасне у моделовању секвенцијалних података, као што је текст. Међу RNN архитектурама, *Long Short-Term Memory* (LSTM) мреже су стекле посебну важност због своје способности да “ухвате” дугорочне зависности и ублаже проблем *vanishing* градијената, који је карактеристичан за стандардне RNNs. LSTM мреже су посебно погодне за детекцију *phishing e-mail* порука,

јер могу ефикасно анализирати секвенцијалну природу текста, хватајући контекстуалне информације које су кључне за разликовање *phishing e-mail* порука од легитимних.

Бидирекциони LSTM (Bi-LSTM) представља неуронску мрежу састављену од LSTM јединица које раде у оба смера инкорпорирајући контекстуалне информације из прошлости и будућности. За разлику од LSTM мреже, Bi-LSTM мрежа има два паралелна слоја који врше пропагацију у два смера - узапред и уназад - да би ухватили зависности у оба контекста.

У овом раду су имплементирани LSTM и Bi-LSTM, у циљу детекције *phishing e-mail* порука.

LSTM модел обрађује текст *e-mail* порука унидирекционо (од почетка до краја секвенце), што му омогућава да учи зависности унутар "природног" тока текста. Међутим, тактике *phishing* напада често укључују комплексне обрасце који можда нису лако уочљиви када се текст обрађује само у једном правцу.

Узимајући у обзир овакву природу *e-mail* порука, у овом раду је кориштен и Bi-LSTM модел са циљем поређења перформанси. Двосмерна обрада текста код Bi-LSTM модела је нарочито предност када су кључни индикатори *phishing*-а присутни и на почетку и на крају *e-mail* порука, или када је контекст око одређене речи од суштинске важности за тачну класификацију.

Иницијална фаза изградње модела подразумева претпроцесирање, токенизацију и изградњу речника употребом *torchtext* библиотеке. Токенизација је извршена употребом *spaCy* токенизатора.

Подела података је извршена тако да је 70% података употребљено за тренинг, 10% података је резервисано за валидацију, док је 20% података узето за тестирање. Овакво стратификовано раздвајање је осигурало да је модел обучен на репрезентативном подскупу података, док су валидација и тестирање извршени на одвојеним деловима података.

Речник је креиран од тренинг података и за величину речника је узето 10000 јединствених токена. На овај начин је обезбеђено да речи које се појављују ретко(мање од пет пута) буду искључене, чиме смањујемо димензионалност улаза и ублажавамо могућност *overfitting*-а.

Следећи корак представља дефинисање модела.

Први слој модела представља *Embedding* слој. Разлог постојања *Embedding* слоја лежи у чињеници да токенизовани подаци губе своје значење. Код обраде слике, величина магнитуде одговара осветљености пиксела, па већа вредност значи светлији пиксел. Међутим, магнитуда података у векторизованом сету података нема такво значење – тачка са вредношћу 5000 нема нужно више значења од тачке са вредношћу 0. Ако текст представимо као категоријске податке са *one-hot* низовима, долазимо до проблема реткости података(енг. *sparse*): са речником од 50000 речи, где се у документу користи само 400 речи, ова метода би била изузетно неефикасна. [10] *Embedding* слој конвертује токене у густе(енг. *dense*) векторе одређене димензије омогућавајући моделу да научи репрезентације речи које обухватају њихово семантичко значење.

Излаз LSTM слоја се прослеђује на два потпуно повезана (FC) слоја у циљу мапирања високо-димензионалних скривених стања из LSTM-а у коначни простор излаза. Први потпуно повезани слој редукује димензионалност LSTM излаза на интермедијарну димензију величине 20 [2]. Овај слој омогућава компактнију репрезентацију података при чему очувава најважније карактеристике извучене из LSTM-а. Други потпуно повезани слој мапира излаз из првог слоја на циљну димензију која одговара броју класа излаза.

Последњи слој модела садржи сигмоидну активациону функцију. Ова функција се примењује на излаз другог потпуо повезаног слоја. Сигмоидна функција је нелинеарна активациона функција која мапира улаз у вероватноћу између 0 и 1. Ова вероватноћа може се тумачити као вероватноћа да улазна секвенца припада позитивној класи у бинарном класификационом задатку.

Dropout слојеви су укључени како би се спречио *overfitting* насумичним деактивирањем дела неурона током тренинга, те како би се смањила зависност неуронске мреже од одређених карактеристика(енг. *features*).

Осим овакве архитектуре, разматрана је могућност додавања других слојева, првенствено слојева са ReLu активационом функцијом у циљу уношења нелинеарности у систем. Међутим, овакве архитектуре су се показале лошијим од горе описане, те нису узете у разматрање.

Такође, водећи се радом [10], први потпуно повезани слој је замењен *Max-Pooling* техником у циљу екстакције карактеристика и редукције димензионалности. У оваквој архитектури, излаз секвенце се агрерира бирајући максималне вредности кроз све временске кораке, на тај начин истичући најзначајније карактеристике. Овакав приступ, осим саме редукције димензионалности, потенцијално побољшава генерализацију фокусирајући се на најистакнутије карактеристике у секвенци.

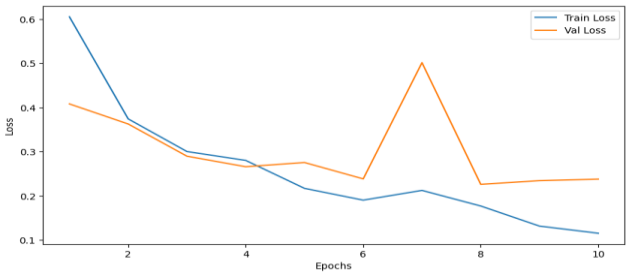
Током обуке модела је вршена оптимизација хиперпараметара употребом *grid search* приступа. У процесу оптимизације је вршено тестирање различитих комбинација хиперпараметара, а сваки модел је обучен за фиксан број епоха. Процес тренирања модела је укључује *backpropagation* у циљу ажурирања параметара модела користећи *AdamW* оптимизатор, док *learning rate scheduler* динамички прилагођава брзину учења како би побољшао конвергенцију.

У циљу спречавања *overfitting*-а модела над тренажним подацима, примењен је механизам раног заустављања(енг. *early stopping*). Тренирање модела се прекида уколико се *validation loss* није побољшао за предефинисани број епоха.

Након завшетка тренирања модела у свим конфигурацијама, модел са најмањим *validation loss*-ом изабран је као најефикаснији.

Хиперпараметри којима се постиже најбоља ефикасност LSTM модела су:

embedding: ***64***
hidden: ***128***
learning rate: ***0.01***
batch size: ***64***
dropout првог слоја: **0.3**
dropout другог слоја: **0.3**



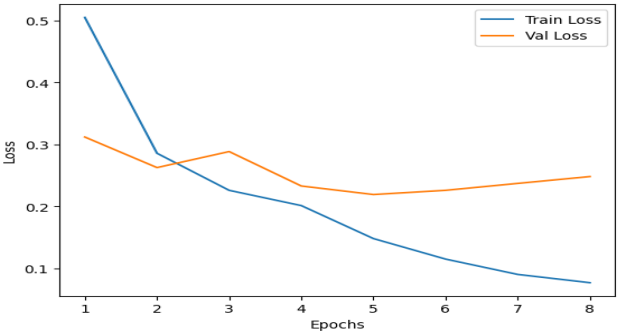
Слика 6. LSTM тренинг и валидациони губици

Са слике примећујемо да, иако *Training loss* опада, *Validation loss* расте након шесте и десете епохе, што нам сугерише да модел може бити подложан *overfitting*-у.

Осим LSTM модела, вршено је обучавање и Bi-LSTM модела. Како је сам процес дефинисања модела и избора хиперпараметара идентичан, у наредној секцији ће бити приложени резултати оба приступа, као и резулатати архитектуре са *Max-Pooling* слојем.

Хиперпараметри којима се постиже најбоља ефикасност код Bi-LSTM модела су:

embedding: ***64***
hidden: ***128***
learning rate: ***0.01***
batch size: ***64***
dropout првог слоја: **0.3**
dropout другог слоја: **0.4**



Слика 7. Bi-LSTM тренинг и валидациони губици

Са слике видимо да је у случају Bi-LSTM модела примењено рано заустављање на осмој епохи у циљу превенције *overfitting*-а, јер валидациони губици континуално расту.

4 Резултати и дискусија

У следећем сегменту биће графички, кроз табеле, приказани резултати сваког алгоритма кроз сваки скуп података и сваку методу евалуације.

Logistic Regression	TF-IDF	Word2Vec	Custom features
Accuracy	97%	93%	81%
Precision	97%	92%	82%
Recall	98%	93%	78%
F1-score	97%	93%	80%
AUC-ROC	99%	98%	89%
FPR	3.42%	8.05%	17.12%
FNR	1.58%	6.69%	21.65%

Random Forest	TF-IDF	Word2Vec	Custom features
Accuracy	97%	95%	89%
Precision	95%	94%	88%
Recall	99%	96%	90%
F1-score	97%	95%	89%
AUC-ROC	99%	99%	79%
FPR	5.14%	5.82%	11.64%
FNR	1.23%	3.87%	9.86%

Gradient Boosting Tree	TF-IDF	Word2Vec	Custom features
Accuracy	97%	96%	88%
Precision	96%	96%	87%
Recall	98%	96%	89%
F1-score	97%	96%	88%
AUC-ROC	99%	99%	95%
FPR	4.25%	3.94%	13.01%
FNR	2.29%	3.52%	11.27%

BERT	Fine-tuned	Longformer
Accuracy	96.2%	99.04%
Precision	96.02%	98.8%
Recall	96.7%	99.3%
F1-score	96.4%	99.1%
AUC-ROC	96.2%	99.03
FPR	4.2%	1.24%
FNR	3.3%	0.7%

Naive Bayes	TF-IDF	Word2Vec	Custom features
Accuracy	95%	X	77%
Precision	95%	X	80%
Recall	95%	X	72%
F1-score	95%	X	76%
AUC-ROC	99%	X	88%
FPR	4.45%	X	17.29%
FNR	5.28%	X	28.17%

Decision Tree	TF-IDF	Word2Vec	Custom features
Accuracy	95%	94%	86%
Precision	94%	92%	85%
Recall	96%	95%	87%
F1-score	95%	94%	86%
AUC-ROC	95%	94%	86%
FPR	5.50%	7.77%	15.53%
FNR	4.29%	4.95%	13.20%

LSTM	Base	Max-Pooling	Bi-LSTM
Accuracy	92.57%	92.75%	94.1%
Precision	93.47%	93.17%	93.08%
Recall	92.01%	92.36%	93.4%
F1-score	92.74%	92.76%	94.1%
AUC-ROC	97.75%	98.11%	98.55
FPR	6.5%	6.85%	7.03%
FNR	7.99%	7.64%	6.60%

Из табеле са резултатима можемо закључити да су BERT *Longformer* и *TF-IDF* најефикаснији за све испитиване методе, обезбеђујући најбоље резултате у свим метрикама. *Word2Vec* се такође показао као изузетно добар, са резултатима који су блиски *TF-IDF*-у. Супротно томе, ручно извучене карактеристике показале су се мање успешним, нарочито код *Logistic Regression* и *Naive Bayes* метода.

Када је реч о BERT *base* моделу, увиђамо да фино подешен модел достиже резултате у рангу *Word2Vec* и TF-IDF.

LSTM, LSTM са *Max-Pooling* слојем и *Bi-LSTM* су показали нешто слабије резултате. Очекивано, *Bi-LSTM* даје боље резултате због начина обучавања модела.

Уочено је да се резултати већим делом слажу са већ публикованим резултатима, те ово истраживање углавном потврђује тенденције пронађене у релевантној литератури.

Закључак

У овом раду описан је предлог решења за детекцију малициозних *phishing e-mail* порука.

Идеја истраживања је била поређење ефикасности детекције употребом ручно извучених карактеристика са карактеристикама извученим помоћу *TF-IDF, Word2Vec,* BERT и LSTM метода. Значај овог рада се првенствено огледа у чињеници да је за разлику од већине радова, осим тела поруке посматран и пошиљалац поруке као и URL-ови садржани у телу поруке.

Посебна пажња посвећена је одабиру и имплементацији решења базираног на ручном одабиру карактеристика. Број колона у скупу података са ручно извученим карактеристикама знатно је мањи, па је самим тим време обуке модела знатно мање.

Међутим, за разлику од пријашњих радова, из добијених резултата увиђамо да је ефикасност модела са ручно издвојеним карактеристикама значајно лошија, што произилази из чињенице да и малициозни корисници прате трендове детекције, те откривају нове начине за обману корисника.

Време обуке је мање, али и време обуке *TF-IDF* и *Word2Vec* је прилично мало. Када је реч о BERT и LSTM методама, време обучавања је дуже, а резултати се већим делом слажу са већ публикованим резултатима.

Уколико је скуп података веома велики (стотине хиљада или милиона редова) постоји могућност да је вредно ручно извлачити карактеристике.

5.1 Практичне примене и правци даљег истраживања

Основа овог рада представља детекција *phishing e-mail* порука, те у тој области рад и проналази своју примену. Такође, како је посебна пажња посвећена детекцији малициозних URL-ова и анализи тела поруке, рад би се(уз потенцијално прилагођавање) могао употребити и за детекцију *smishing*-а и детекцију малициозних URL-ova.

Могући правци даљег истраживања обухватају додавање нових ручно одабраних карактеристика, првенствено URL карактеристика. Такође, како је у раду фино подешени BERT дао изузетно добре резултате, очигледно је да се оставља простор за даље обучавање датог модела у циљу повећања ефикасности.

Поред тога, један од праваца даљег истраживања јесте употреба CNN као и LSTM-CNN комбинације. Конволутивне неуронске мреже су се показале корисним у "хватању" хијерархијских односа присутних у подацима кроз операције конволуције. Ова особина може бити веома корисна у представљању одређених образаца у тексту *e-mail* порука, као што су секвенце карактера или речи које могу бити знаци *phishing*-а.

Хибридни модел LSTM-CNN би користио предности обе архитектуре - конволутивне неуронске мреже би ефикасно екстраховале локалне карактеристике, док би рекурентне неуронске мреже "хватале" дугорочне зависности у секвенцијалним подацима. [11]

Библиографија

[1] E. Lastdrager, „Achieving a consensual definition of phishing based on a systematic review of the literature," *Crime Science,* t. 3, br. 1, pp. 1-10, 2014.

[2] T. G. S. V. i. K. S. S. Salloum, "A Systematic Literature Review on Phishing Email Detection Using Natural Language Processing Techniques," *IEEE,* vol. 10, pp. 65703-65727.

[3] P. Bountakas, K. Koutroumpouchos and C. Xenakis, "A Comparison of Natural Language Processing and Machine Learning Methods for Phishing Email Detection," *Association for Computing,* vol. 127, pp. 1-12, 2021.

[4] A. Yasin and A. Abuhasan, "An intelligent classification model for phishing email detection," *Journal paper, International Journal of Network Security & Its Applications (IJNSA),* vol. 8, no. 4, 2016.

[5] A. A. Abdullah, L. E. George and I. J. Mohammed, "Email Phishing Detection System Using Neural Network," *Research Journal of Information Technology,* vol. 6, no. 3, pp. 39-43, 2015.

[6] "A. Phishing Email Curated Datasets[Nazario]2023," [Online].

[7] "A. Phishing Email Curated Datasets [SpamAssassin]," [Online].

[8] J. Devlin, C. Ming-Wei, L. Kenton and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for," *Proceedings of NAACL-HLT*, p. 4171–4186, 2019.

[9] S. Chi, Q. Xipeng , X. Yige and H. Xuanjing , "How to Fine-Tune BERT for Text Classification".

[10] P. Zhou, Q. Zhenyu, Z. Suncong and X. Jiaming, "Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling," 2016.

[11] Z. Alshingiti, R. Alaqel, J. Al-Muhtadi, Q. Haq, K. Saleem and M. Faheem, "A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN," *Electronics,* vol. 12, no. 1, 2023.