EXPLORING A ROBUST MACHINE LEARNING CLASSIFIER FOR DETECTING PHISHING DOMAINS USING SSL CERTIFICATES

by

Akanchha Akanchha

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
April 2020

To My Mummy(Uma Sinha) and Papa(Late Suresh Prasad Sinha)

Table of Contents

List of Tables

List of Figures

Abstract

Due to the phishing sites appearing genuine to users, detecting them is a challenging task. The SSL certificate that is generally used to secure and encrypt the communication, can also be generated for the phishing sites. The consequences of using the HTTPS phishing sites could be harmful to users. Attackers can easily trap users and steal sensitive information using the cloned website that looks legitimate along with its SSL certificate. This may also result in the degradation of the user's trust and belief in "green padlock" and "lock icon" shown on the browser after connecting to a web site through HTTPS. In this thesis, I have studied the important attributes of how attackers use SSL certificates in sites with fake domains. As a result, I have explored the robustness of a system to auto-detect a phishing site using the critical attributes of an SSL certificate. The proposed system uses different Machine Learning algorithms that utilize extracted SSL certificate features studied. Considering good performance and transparency in the resulting model, I have chosen the decision tree algorithm for decision making of site category. The algorithm defines a set of decision rules to classify whether the site is a legitimate site or a phishing site. The proposed classifier has achieved around 97% of correctly classified instances in comparison with other machine learning classifiers. In order to connect users to the system, a Web API is created which provides the user interface of the proposed system through HTTP service. The API verifies single domain as legitimate or phishing domain by using the decision rules of decision tree algorithm. Evaluation results show the promising effectiveness and efficiency of the Web API system designed and developed.

List of Abbreviations Used

SSL	Secure Socket Layer
TLS	Transport Layer Security
HTTP	HyperText Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
API	Application program interface
TLD	Top Level Domain
APWG	Anti-phishing Working Group
DNN	Deep Neural Network
KNN	k-Nearest Neighbours
URL	Uniform Resource Locator
SVM	Support-vector machines
IP	Internet Protocol
AOL	America Online
LMT	Logistic Model Tree
PART	Partial decision tree
Jrip	J-Repeated Incremental Pruning
AdaBoost	Adaptive Boosting
1Dconv	1D Convolution Neural Network
LSTM	Long Short Term Memory
TF-IDF	Term Frequency-Inverse Document Frequency
EMD	Earth mover's distance

C2	Command and Control
HTML	Hypertext Markup Language
VIF	Variance Inflation Factor
MCC	Matthews correlation coefficient
ACC	Accuracy
TPR	True Positive Rate
FPR	False Positive Rate
TNR	True Negative Rate
FNR	False Negative Rate
DOM	Document Object Model
CA	Certificate Authority
CO	Company Organization
ST	State
DT	Decision Tree
CERT	Certificate format
L	Location
RSA	Rivest Shamir Adleman algorithm
PKI	Public Key Infrastructure
Weka	Waikato Environment for Knowledge Analysis
LIBSVM	Library for Support Vector Machines
IPv4	Internet Protocol version 4
TCP	Transmission Control Protocol
SEO	Search Engine Optimization

Moz	Marketing Software
CSV	Comma-Separated Values
DER	Distinguished Encoding Rules
PEM	Privacy-Enhanced Mail
CN	Common Name
AFINET	Address-family
SOCK	Socket Connection Type
DDOS	Distributed Denial of Service
SaaS	Software as a Service
CSS	Cascading Style Sheets

Acknowledgements

First, I would like to express my sincere gratitude to my supervisor Dr. Nur Zincir-Heywood and co-supervisor Dr. Raghav Sampangi for providing me opportunity to do research with them. They constantly assisted me with suggestions and feedback and provided invaluable guidance throughout the thesis. I am extremely thankful to them for their patience, motivation and helping me with their ideas and technical knowledge in all the time of research.

I am heartfelt thanks to my brothers, Ankit and Aman for their love and cheering me all the time. My special thanks to my mom for her prayers and support specially when I was coming to Canada to study.

Last, I want to say thanks to my friend Alok Ranjan for his encouraging words and continuing support to complete this thesis. Many thanks for always motivating me.

Chapter 1

Introduction

With the exponential growth of the internet and digital marketing, more people rely on online services for their day to day needs. Several sectors such as payment, search engines, social media, email, banking, and different business marketing websites are designed to connect digitally with prospective users. These web services have easier internet infrastructure set up with low-cost and less cyberspace. Digitalization improves efficiency and also ease of access. It has been recorded that around 1.74 billion websites [47] are present on the internet and 351.8 million domain names were registered across all top-level domains (TLD) in 2019 [14]. Along with many advantages, digitalization also brought issues of privacy and security [52] of data. It has been observed that the threat of vulnerabilities and attacks in online services are more as compared with the offline services due to the availability of free attacking tools such as Netstat, Nikto, Wix etc. Attackers use these tools to look for weakness or flaws within a security system of organization to compromise a secure network. For an example, if a organization has broken authentication, attackers can hijack user sessions to get a user's credential and then pose as the original user. They could gain access to unauthorized data or open paths for money laundering [8]. Many cases of identity theft, data misuse, fraud etc., came into the picture [7]. In these cases, attackers created clone sites of popular legitimate sites or tried to hijack communication through man-in-the-middle attacks for stealing personal information and sensitive data of the users. In order to prevent these attacks, Secure Hypertext Transfer Protocol (HTTPS) has been applied over the transport layer protocol. HTTPS encrypts traffic between two networks. This secure web service uses Secure Socket Layer (SSL) certificate to create a secure communication between the browser and the server. On connecting through HTTPS to any legitimate website, the lock icon shown generates trust for the internet users by keeping their data or personal information protected. Nearly one million most visited websites switched from HTTP to HTTPS in 2019 [12]. It is

almost impossible to decrypt and read communication by any third party unless the third party has the private key. By using this key, the third party can decrypt the session and interrupt or read the communication between two networks.

In recent years, it has been analyzed that the attackers are able to compromise the encrypted traffic. SSL encryption hides the transmission of many attacks that are continuously rising to 400% since 2017 [62]. By using forged or compromised keys, and certificates, attackers create malicious tunnels into the network where they hide to conduct surveillance, install malware and ultimately exfiltrate valuable data [38]. The inspection of attacks under encrypted traffic is quite challenging because of lack of effective strategies. The idea of decrypting and then re-encrypting the traffic[30] in search of threats creates new vulnerabilities. In addition, the trusted lock icon shown on the address bar made attackers' way easier to deceive the users.

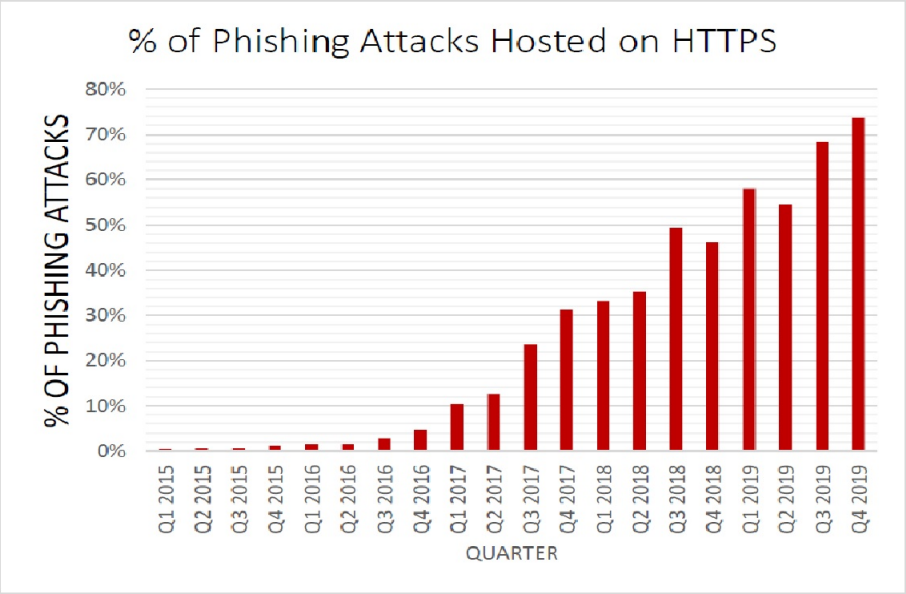

Figure 1.1: Rise of Phishing attacks using SSL certificates [31]

The percentage of phishing attacks using SSL certificates in different years is shown in Fig 1.1. According to the Anti-phishing Working Group track report (APWG), there was an increase in the phishing attacks from 2015 to 2019. In the year 2015, the attacks were less than 2%. But the phishing attacks using SSL certificates were kept on increasing since the third quarter of 2016. In that quarter, there was 2.5% of HTTPS phishing attacks which later increased to 74% in the fourth quarter of 2019.

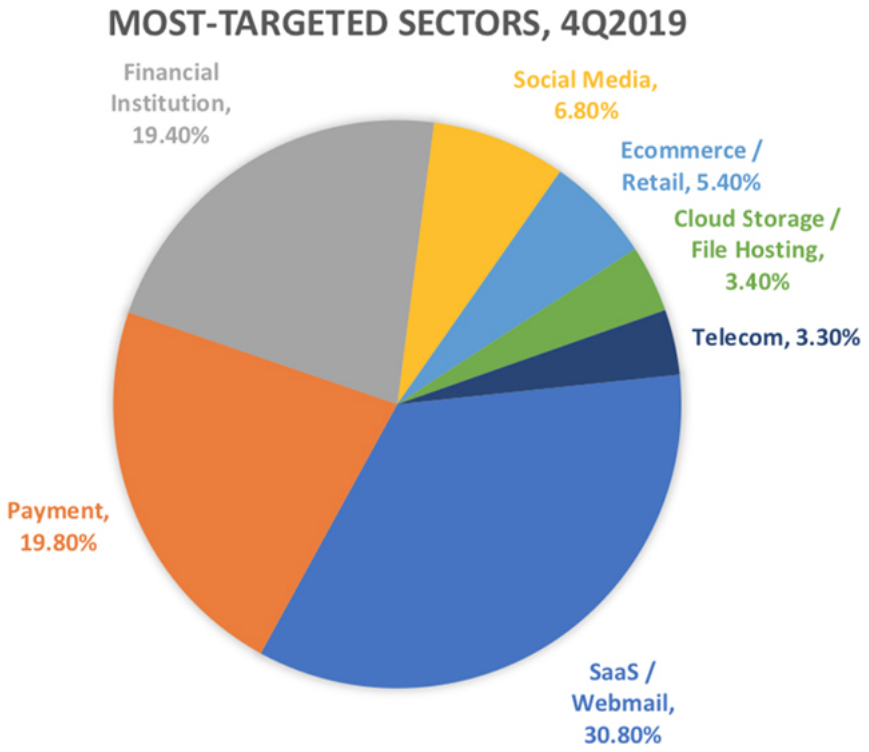

Figure 1.2: The Most Targeted Industries in 2019 [31]

In the year of 2019 only, the percentage increased from 58% to 74% in just three quarters. There was an increment of a total of 16% from second quarter to fourth quarter of 2019. This indicates that a high percentage of phishing sites are using the SSL certificate and HTTPS protocol for attacking purposes.Along with the tracking of HTTPS phishing attacks, APWG also inspected carefully about the effected industries in the last quarter of 2019 that is shown in Fig 1.2. According to the figure, users of the Software as a service (SaaS) and Webmail services were the targeted groups in 2019 and had 31% of phishing attacks. As per the report [31], first time SaaS and Webmail sectors became the highest category of phishing attacks as compared against the payment sector. The attackers paid more attention towards these sectors since these provided online services to the organization and easily accessible on the internet [31]. Financial institution and Payment services still had a high percentage of phishing attacks. Other sectors such as social media, telecom, e-commerce, and also cloud and hosting sites had less than 10% of the attacks. This implies these sectors were the least targeted compared to the other four sectors. Hence, the analysis of the APWG' 19 reports shows that attackers are utilizing SSL certificate, a security

measure for the wrong purposes in different sectors and they are targeting more to business sites as well as the accounts related to payment and finance for stealing or manipulating the sensitive information.

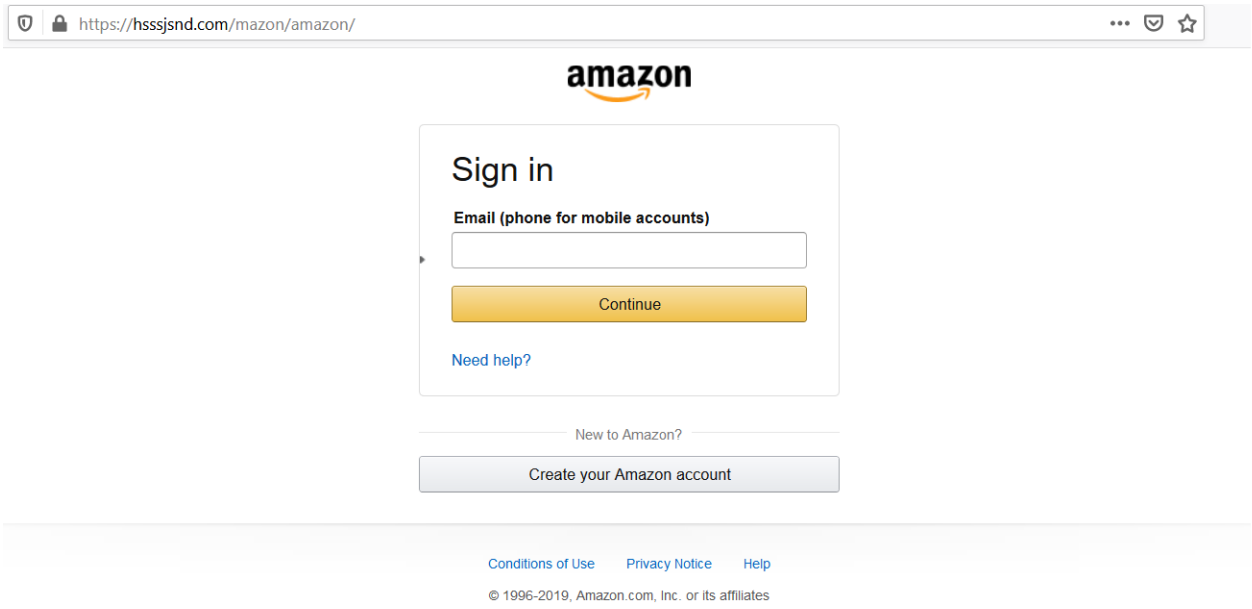

Figure 1.3: Phishing website of Amazon

Fig 1.3 shows a phishing website acting like Amazon that is from the list of phishing sites from Phishtank[1] and Fig 1.4 shows the original website of the Amazon from Alexa[2]. An unskilled user would not be able to recognize the original website that is present in one of the two figures, Fig 1.3 and Fig 1.4 because the user interface design with a lock icon at the address bar is making the phishing website look real. This shows that a SSL certificate on the website presents no assurance in securing the user's data anymore. There are more than 1000 cloned phishing websites of popular brands present on the internet such as the aforementioned Amazon phishing website. Zscaler Cloud Security Insights Threat report analyzed the SSL traffic and estimated the top phished brand over HTTPS during 2019 [62]. As per the analysis, Microsoft was on top of the list with 58% of phishing websites in 2019. The brands such as

[1] `https://www.phishtank.com/`
[2] `https://www.alexa.com/`

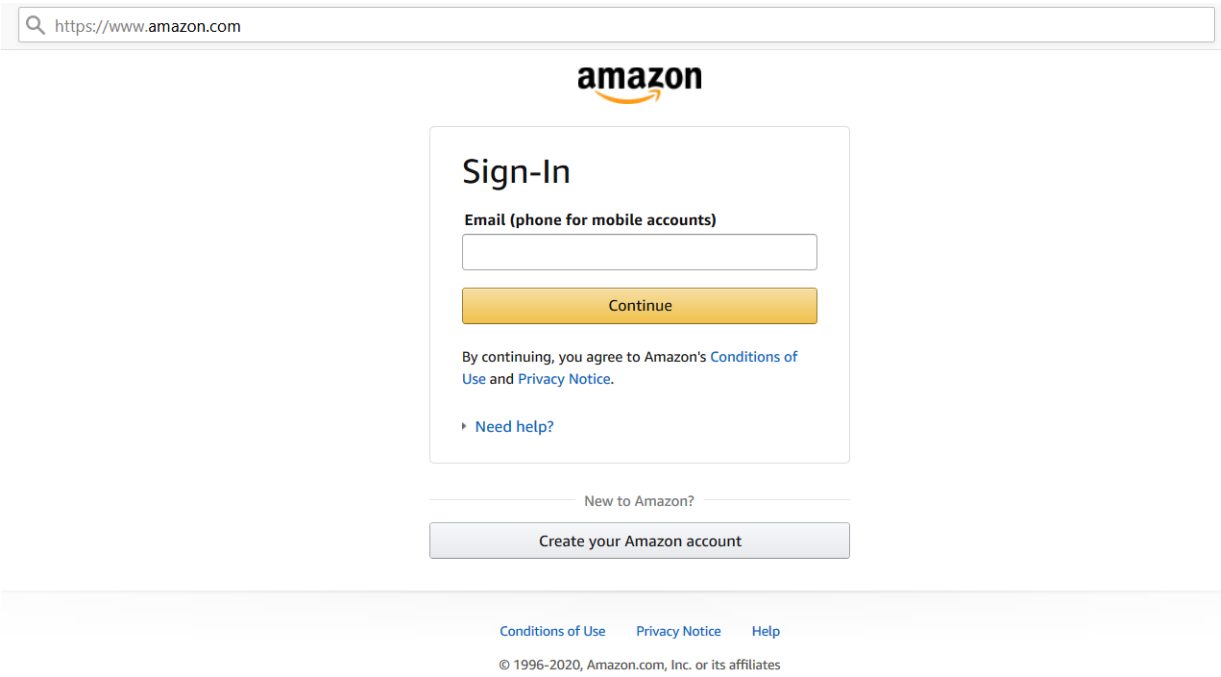

Figure 1.4: Original website of Amazon

Facebook are 12%, Amazon and Apple both 10% and other brands were having 4% of the phishing websites present on the internet. The report observed that the presence of lock icon and HTTPS protocol on the phishing websites were creating the threat of security risks while visiting the targeted brands which was decreasing the popularity of the websites.

In a recent study, Torroledo et al.[64] focused on classifying the legitimate SSL certificates and phishing SSL certificates. Their system employed feature extraction techniques and a deep neural network (DNN) classifier. The irrelevant features can create huge impact on the performance of machine learning, therefore they used a feature selection process to remove the misleading data. The feature selection process was used by Kabilan et. al as well [54]. They also adopted the same feature extraction technique and deep neural network to automate the detection process. But they did not elaborate on the feature selection methods used.

The study done by Zheng et al [33] aimed to classify the phishing and legitimate Uniform Resource Locator (URLs). The system was engineered with features from

SSL certificate and employed six machine learning algorithms such as KNN, Random Forest, C4.5, Decision Table, Naive Bayes and Simple Logistics. At the end, result of each classifier was combined and averaged to get the final decision of the URL category. Due to differences between classifiers, the time taken for classification was more in case of KNN classifier which made the system slower in making the final decision.

The current works [64, 54, 53, 33] did not analyze and test the robustness of the designed models. Robustness is vital to test the effectiveness and consistency of the final design of the system. However, previous literature did provide good recommendations on the assumptions of training data collection for the phishing and legitimate categories.

In this thesis, *the goal is to explore a robust detection system which could detect whether a domain is legitimate or phishing by utilizing details in SSL certificate.* I explore and evaluate the robustness of the detection system by training it with different sets of features and then testing with different assumptions.

The objectives of the thesis are:

- To analyze the importance of features: Which features in the model contribute to the detection system with good accuracy.
- To explore the robustness of the (classification) detection system: The purpose is to test effectiveness of the final design of the system
- To investigate the following issues raised by APWG and Zscalar:
  - How attackers generate an SSL certificate for the spoofed website?
  - How to differentiate between legitimate and phishing websites over secure service?

In order to achieve these objectives, the feature creation, feature selection and normalization techniques, and machine learning classifiers were employed for classifying the category (legitimate vs phishing) of the domains. These methods made easier the designing process of detection system by reducing the complexity of data samples, extracting the hidden indicators, and providing the predictive analysis of datasets. In the first step, HTTPS domains for legitimate and phishing classes were collected

from the internet. Then two separate automated scripts were implemented in Python programming language for collecting domain's SSL certificate and creating features from these collected certificates. These two scripts made both the processes faster and saved time in extracting 42 features and also downloading SSL certificates from domains. These features were classified into three types: Boolean, Integer, and Text. After generating datasets with 42 features, I normalized the numeric attributes and then removed irrelevant attributes from the dataset using Weka tool sets. These decreased the training time and increased the validating accuracy of the model. Thereafter, several machine-learning classifiers were built using Weka and then compared with each other based on their training performances and ten-fold cross validation evaluation results.

After selecting the best performing machine learning classifier, the selected classifier was examined with different sets of features. The purpose was to understand the importance of the statistical text-features in the dataset. There were two sets of features which were obtained: With text features and Without text features with good performances. The text-features included the statistically learning of text data features along with boolean and integer features whereas Without text-features had only boolean and integer features.

Four sets of datasets were prepared to test the effectiveness of both With text-feature system and Without text-features system. Each test dataset had different combinations of sample data. The combination consisted of same, different and mixed union of data characteristics used in the training data samples. The robustness testing indicated the systems' ability to perform under new datasets (unseen during training). The test results provided variation in test scores for each test-dataset and showed how the test-accuracy changed along with the characteristics of test data.

Finally, Web API and proof-of-concept web application was created to automate all the processes of the proposed detection system. This application allows a user to identify whether a domain visited is legitimate or phishing on frontend. As the backend classification part of the Web API, the decision tree of the chosen machine learning classifier is implemented using the Python programming language.

The main contributions of this research are:

- To demonstrate how the collected data with different patterns in certificates

and domains increased the performance of the trained model.

- To describe the different characteristics observed in the collected data.
- To extract features using scripts from the certificate data.
- To examine the system's performance with different sets of features and understand the importance of features.
- To explore and evaluate the robustness of the detection classifiers.

The thesis is organized as follows: Chapter 2 reviews previous works on different approaches for phishing attacks, provides background knowledge of SSL certificates and their working along with definitions of types of certificates. Chapter 3 discusses the proposed approach, methods used and the datasets employed. Chapter 5 presents the experiments and results of the proposed system. Chapter 6 summarizes the conclusions and the future research plans.

Chapter 2

Related Works

The chapter presents the history of phishing and methods used in previous works to identify the phishing attacks. It also covers background knowledge of a SSL certificate, its working, and types of certificates.

2.1 Phishing and Detection Methods

The first phishing attack was observed in 1996. In this case, attackers accessed private account information and passwords of users of America Online (AOL). [42, 41]. They used fake credit card details to open American Online accounts. Fake credit card numbers were generated randomly by using a script. When these incidents were reported, AOL deactivated all the fake accounts and started to validate the credit card and identity of billing accounts for any new registered users. Apart from opening accounts, attackers also used to spam other legitimate users of AOL by conducting attacks. They sent emails or messages to users specifying that users' personal information was needed to verify their account and billing details. Once they had this information, they used it to access accounts as legitimate user for fraudulent purposes. It was quite easy for attackers to trap because of lack of awareness in users regarding these scenarios. Although, the attack stopped in the same year by AOL implementing some security measures like sending warning message "no one working at AOL will ask for your password or billing information" to users [18]. Back then, the attacker's tricks such as email and instant message spoofing were pretty common. These messages or emails had familiar or attractive words which grabbed the user's attention. Later in the 2000s, they shifted their target more towards online sites [43]. They started stealing user's information by cloning the legit sites and sending the cloned website links through email or messages to users. The design and layout of the cloned website looked very similar to legit website. Once the user clicked on the link, the attacker launched attacks. At present, the attackers still follow the same

old techniques of phishing attack with the addition of new ones such as presence of phishing websites in pop-windows or advertisements, domain spoofing and phishing with HTTPS are now popular these days. Alongside, several research solutions have also been proposed for years to detect phishing techniques used by attackers.

In this section, previous researches are reviewed relating to phishing detection methods. The discussed methodologies and results by these researches were interesting since they showed different effective solutions to prevent the phishing attacks. Through their applied approaches, they demonstrated that there were possible ways to distinguish legitimate and phishing sites. Their ideas concentrated on recognizing the common patterns and characteristics followed by attackers. The phishing URLs, web contents on phishing sites, SSL/TLS certificates used by phishing sites and other techniques were involved in researcher studies. A summary of these studies are presented in following sections:

2.1.1 URL-based Detection Method

This section presents summary of researches that focused on URLs of phishing sites. The idea behind this approach was to analyze the text contents present in the website URLs to detect phishing. Samuel Marchal et al. [49] proposed an automated detection system named "PhishSorm". The system provided a score of 0 (Legitimate) and 1 (Phishing) for a URL based on machine learning analysis methods. The methods were feature-extraction and semantic analysis techniques. The system implementation included URL word extraction, feature creation, and prediction by measuring similarities between registered domain and remaining characters that were present in a URL. The query, low-level domain, main-level domain, upper-level domain, and path were parts of URL. Search engine query data were used to compute the relationship between these parts. This approach was assumed that all the words used in different parts of URLs were related to each other for legitimate URLs whereas, in phishing, the terms used in the query, path and low-level domains had a relation with the targeted brands but not with main-level domain that was used in the URL for phishing purpose. They considered four characteristics of phishing URLs to create the dataset:

- URLs with IP addresses

- URLs with long domains or incorrect words/letters
- URL obfuscation with main domain's name
- URLs obfuscation with mismatched keywords

Twelve features were created by considering intra-relatedness of URLs and its popularity. They employed supervised machine models which were built to distinguish between phishing and legitimate sites. There were seven classifiers Random Tree, Random Forest, C4.5, Logistic Model Tree (LMT), Partial Decision Tree (PART), J-Repeated Incremental Pruning (JRip) and Support Vector Machine (SVM) used for classifying phishing and legitimate sites. Among all classifiers, Random Forest achieved good performance of 95.22%.

Recently Jiwon Hong et al.[39] proposed another approach based on lexical features of URLs. They created 18 strings-based features from the URLs such as length of URLs, Occurrence of "@" etc. The system learnt common patterns by attackers in URLs to detect the phishing website. The architecture of system summarized below:

- Data collection: The Bank of America, eBay, and PayPal, three most targeted sites were selected and collected their respective URLs. Other than these, two more URLs string-dataset of previous works were also included.
- Label data: The collected data were labelled and merged five datasets into one dataset.
- Feature extraction: The method used to extract features from URLs. Word embedding technique of feature extraction used to create vectors value of text-features.
- Unsampling method: 16 methods used to make the balance vector sampling classes.
- Classifiers: Adaptive Boosting (AdaBoost), SVM, Random Forest, 1D Convolutional Neural Network (1Dconv), Long Short Term Memory (LSTM), 1Dconv + LSTM classifiers were employed. The grid search technique was also applied for each classification model to get the best combination performance. In all, Random Forest performed well with good F1 score.

Routhu Srinivasa et al.[58] also proposed a technique named "CatchPhish" to detect the phishing website at client side using the URL-based features. The features classified as Hand-crafted features and Term Frequency-Inverse Document Frequency (TD-IDF) features. The scikit-learn package and python programming were used to create the model.

- Hand-crafted features: These features were manually created based on elements, full URL, and the domain name of the URLs. It was implemented by using python programming.

- TD-IDF features: The method TD-IDF applied on the URLs to analyze the text-content of the URLs. It extracted important keywords from URLs and then used in the detection process.

Finally, both features were combined to build Random Forest model for classifying the legitimate and phishing website. Thereafter classifier was deployed as web application at client side.

These works demonstrated different techniques to understand the patterns of attacks using URLs as an indicator. I used some of these techniques such as feature extraction and feature creation methods in my research since these methods reduced the training speed as well as increased the accuracy of a classifier in previous works.

2.1.2 Web content-based Detection Method

This study was on content of web pages and source code used in creation of phishing website. The idea behind research approaches was to do comprehensive study for determining the difference between legitimate and phishing website. Haijun et al.[68] proposed a phishing detection system based on the visual similarities and text content of the website. In text content, suspicious words or terms were included and excluded stopping words such as "a", "an", "the" present on the website. For visual contents, they were added image content information. Below is the architecture of the proposed system:

- The text content features were extracted from web page and then used in Naïve Bayes classifier to determine the classification result.

- The visual content features were extracted from the web page to generate the visual signature. Then, visual features were used to calculate the Earth mover's distance (EMD) to find the visual similarities of the image of legitimate and phishing website.
- The Bayes classifier was utilized to evaluate the classifier results and determine threshold value for the Naïve Bayes classifier and visual similarities result.
- The condition was if the result values were greater than the estimated threshold value, the page was categorized as phishing.
- At last, the fusion algorithm was employed to combine the results of the classifier with Bayes threshold value to make the final decision for a new incoming page.

There was another approach[50] based on the HTTP headers and web page content. In this approach, the phishing websites were collected from phishing campaigns, drive-by downloads, and command and control (C2) infrastructure whereas legit website were collected from Alexa traffic. This approach employed feature extraction, feature selection, and analysis techniques. The web page content included HTML and JavaScript features. They extracted the below features:

- JavaScript features from prior work. It mainly consisted of common JavaScript methods such as create elements(), eval() and others. Each method had own functionality.
- Mozilla Developer Network methods for HTTP Header features.
- Hypertext Markup Language (HTML) features were included tags, attributes, links. It contained entire HTML code snippet elements.

The total of 1,865 features were analyzed that were later reduced to 26 features after applying the gradient boosting algorithm, computing Variance Inflation Factor (VIF) and, removing irrelevant and dependent features. They used two sets of features, one of 26 features, other of 50 features from prior work and built eight supervised machine learning classification models. The algorithms were Bagging Classifier, Ensemble Method, Logistic Regression, a Generalized Linear Model, K-Nearest Neighbors (KNN), Nearest Neighbor Method, and Neural Network. The comparison was done

between the results of 26 features built model with 50 features built model. The average Matthews correlation coefficient (MCC) and Accuracy (ACC) for the 26 features were higher than 50 features.

Similarly, Rosiello et al.[59] presented a detection approach called "DOMAntiPhish", based on layout similarities of the websites. The system used Document Object Model-Tree (DOM) of web page to detect the phishing website. The proposed system warned users when the same information such as same credentials were used in multiple websites. The system compared the DOM-tree of the current page and previously visited page and alerted the user in case of similarities found in the pages. The assumption was made if the user put the same password having different layout than the user was only reusing the password on another legitimate website. The system maintained a database of user's information with the visited web pages. The conditions for determining DOM-trees equivalence were:

- Classes of the corresponding vertices were the same.
- Indices of corresponding edges were the same.
- Both DOM-trees were isomorphic [59]

At the end, layout similarities were computed by ratio of weighted number of matched vertices of DOM-Trees to total vertices of legitimate web page. In the case of phishing, similarities value should be greater than threshold value of 0.5.

I used similar data collection and selection processes in my research since these improved the quality of collected data in their research work. Explanation of these processes are presented in chapter 3

2.1.3 SSL Certificates-based Detection Method

The idea behind this detection method was to learn the encrypted traffic and content of the SSL certificates. Torroledo et al.[64] proposed a system to detect maliciously encrypted traffic. They were collected three types of SSL certificates-legitimate, malware and phishing certificates and prepared two datasets, one for legitimate and malware and others for legitimate and phishing. The approach used feature extraction and feature analysis techniques. There were 40 features which categorized between

Boolean, Integer and Text features. These features were created by analyzing the important indicators present in the SSL certificates. The Deep Neural Network algorithm with LSTM was employed for classification purposes. The architecture of the system is summarized below:

- Converted all the features into a matrix form by using one hot-encoding method.
- Word embedding method was used to create vector form of features.
- Passed the vector values to LSTM layer and also additional layer of neural network was added to get single probability of phishing.
- Calculated the scores for classification of the classes

At last, the deep neural network classifier result was compared with prior work model using the SVM algorithm. The accuracy of deep neural network classifier was greater by 7% for malware certificates and 5% for phishing certificates. Kabilan et al.[54] presented a system to analyze the real network data. The Rest API employed framework and distinguished the SSL certificate as malicious, phishing or legitimate. The system had three components.

- Apache Metron: It sent individual data packets to the Rest API
- Neural Network classifier: It trained and tested datasets.
- Rest API: It deployed the classifier and provides classification result.

The user made network requests through the command-line interface on a machine that was monitored by Apache Metron to classify their datasets.

Mishari et al.[42] proposed a technique to identify fraud domains over HTTPS protocol. They included random, and malicious domains in phishing and typosquatting data samples and popular domains in legitimate data samples The nine features-Boolean and Non-Boolean features extracted from SSL certificates. These features were used by the machine-learning algorithms for classification purposes.The used classifiers were Random Forest, Decision Tree, k-Nearest Neighbour and two optimization techniques for Decision Trees: Bagging and Boosting built to train models. Then models were compared based on precision-recall performance metrics. At the

end, the performance of classifiers was evaluated using the ten-fold cross-validation method.

Similarly, Zheng et al.[43] proposed a certificate-based anti-phishing technique to detect phishing attacks. They considered the components and optional fields of SSL certificate to create features. The 42 features were extracted from the SSL certificate and built machine-learning system to classify phishing and legitimate domains. The architecture of the system is summarized below:

- Downloaded the certificates from the domains.
- Extracted features from the downloaded certificates.
- Created the classification models using Random Forest, K-Nearest Neighbors, C4.5, Decision Table, Naive Bayes Tree, and Simple Logistic algorithms
- Used the computed average probability of classifier results or decision threshold for Random Forest to make the final decision of classification.
- Evaluated average performance of classification by using ten-fold cross-validation method.

I used the same features and classifiers in my research for analysis. These made it easier for them to understand the common behaviour followed by attackers while generating SSL certificates for phishing websites. All features and classifiers which were used in this thesis are presented in chapter 3.

2.1.4 Other Detection Methods

The approaches were based on usable studies. With these studies, researchers evaluated the effectiveness of security tools. Mohamed et al.[24] conducted a user study about browsers' security indicators and phishing awareness in the users. They analyzed user's eye-catching and self-reporting observations on the websites. As per the eye-catching data results, the observation found that user paid attention towards security indicators only 6% of the time and 9% of the time on other browser elements. Remaining 85% of the time, they just looked over body of the page. Also, the self-reporting results showed that only 53% of phishing websites were correctly

detected by users. Vincent et al.[51] provided knowledge of phishing and legitimate certificates. Based on evidences, they demonstrated that the differentiation between legitimate and phishing websites was not possible by only analyzing SSL certificates' information and brands names because most phishing sites were now protected by legitimate SSL certificates. Tara et al.[66] studied the collected eye-tracking and questionnaire data. Based on the collected data, they observed that how the users utilized security tool. The results showed that the users only saw the lock icon on the address bar and completely ignored the security and certificate information on the browsers.

Along with user studies, the blacklisting approach[34] is also used to identify phishing threats. Google safe browsing API [1] uses blacklist to do safe browsing. The blacklist maintains a list of malicious/phishing sites and store at client side. API queries blacklist to determine whether visited URLs was on the list or not. If the loaded URL is present in the list, then it shows warning notification on the browser. Otherwise, the page is considered as legitimate [34].

Two approaches which were discussed in this section helped me in understanding how I could implement an effective indicator for users. These approaches were considered while implementing Web API and proof-of-concept web application that is detailed in chapter 3.

2.1.5 Summary of Phishing and Detection Methods

Different methods and features have been employed in the researches. Each detection methods have some advantages and disadvantages.

- With URL-based detection method, the advantage is less evaluation time because only a limited portion of the text is analyzed[64]. On the other hand, the flaw is the phishing methods are constantly evolving, the features used in the proposed researches may not be useful to detect the phishing websites. The features are created based on characteristics of phishing URLs which can be changed or compromised in future by the attackers.

[1] `https://safebrowsing.google.com/`

- The disadvantage of web content-based detection method is the maximum analysis time. It takes a long time to study the whole content of the website pages. Also, there is always a risk of attacks such as installing malware or ransomware in the device while navigating to the phishing web pages. For extracting the features, it is necessary to download the phishing web pages. Alongside, this method could also provide several features to examine. With the more number of features, the confidence of detection system for predicting the phishing website increases.

- With the SSL-certificate based detection method, the advantage is that the mandatory fields of certificate are taken as features to detect the websites. The chances to compromise these fields are lesser by the attackers. But the problem of this method is that data samples of the web-certificate are not easily accessible.

- Although blacklisting approach successfully warns user before visiting phishing websites but the disadvantage is phishing websites are activated for short span of time. So blacklist usually cannot cover all phishing websites because it deactivates before adding to the list. [34]

2.2 SSL Certificate

SSL certificates [3] also refers to digital certificate or public key certificate that is used over HTTPS protocol. It provides the digital identity to an organization and shows the organization and its website are trustworthy. SSL certificate follows Public Key Infrastructure (PKI) standards, that allows web server and a web browser to authenticate each other using certificate. The trusted certificate authority (CA) issues SSL certificate to the organization [3] and ensures safety of private and public keys. SSL certificate takes private and public key and use it while creating communication between a web server and a web browser for authentication purposes. The private key is always private, used only by the owner of the certificate whereas the public key is publicly available for everyone and can be used by those who need to validate the owner's SSL signature [22]. Both private and public keys are generated using the Rivest Shamir Adleman (RSA) algorithm. The keys should be safe to avoid any

type of security risks. As browsers store some trusted Certificate Authority (CA)'s certificates. In case if the incoming website's certificate is not present, the warning message will be shown on the browser before directing to the website. Figure 2.1 shows the format of the SSL certificate. Below is the general process for encrypting the traffic:

- The web server sends the public key with the certificate to the browser.
- With the public key, the web browser creates and encrypts the session for communication. The browser replies to server with the public key.
- The web server decrypts the session with its private key to enter the created session.
- In that session, secure communication starts between the browser and the server. The session is created for a short span of time to avoid security risks.

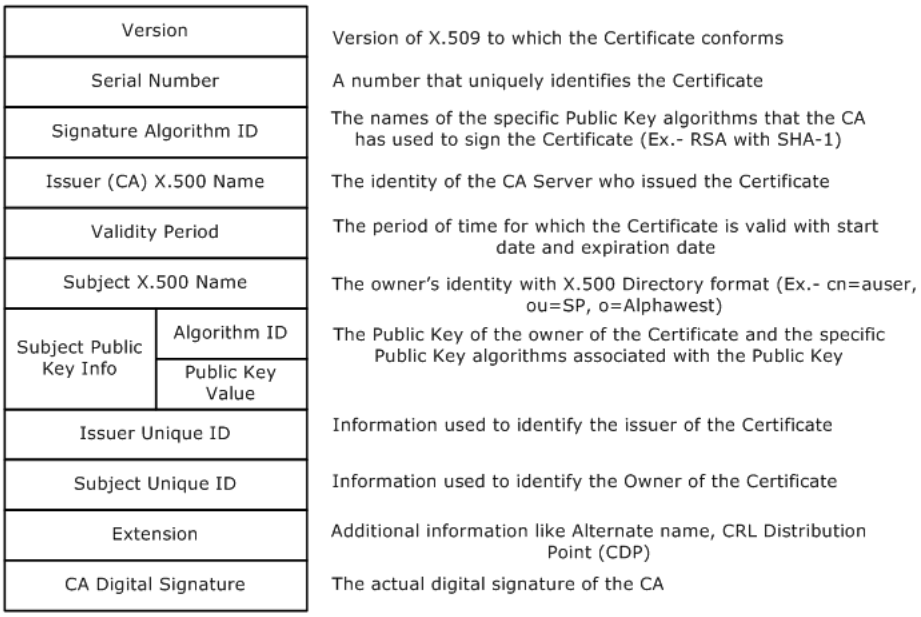

Figure 2.1: SSL Certificate Format [3]

2.3 Types of Certificate Validation

To obtain the SSL certificate [3], the organization requests CA server and sends a certificate signing request with all the organization details including email address,

organization unit etc and public key. The CA first verifies the identity of an organization with the provided detail information and then issues SSL certificate to the organization. After verifying, CA digitally signed the SSL certificate and send it to the organization. The signature of CA verifies the authenticity of SSL certificate[3]. However, validation of the organization depends on which category of SSL certificate is requested to CA for web domain. The description of SSL certificate types are provided below:

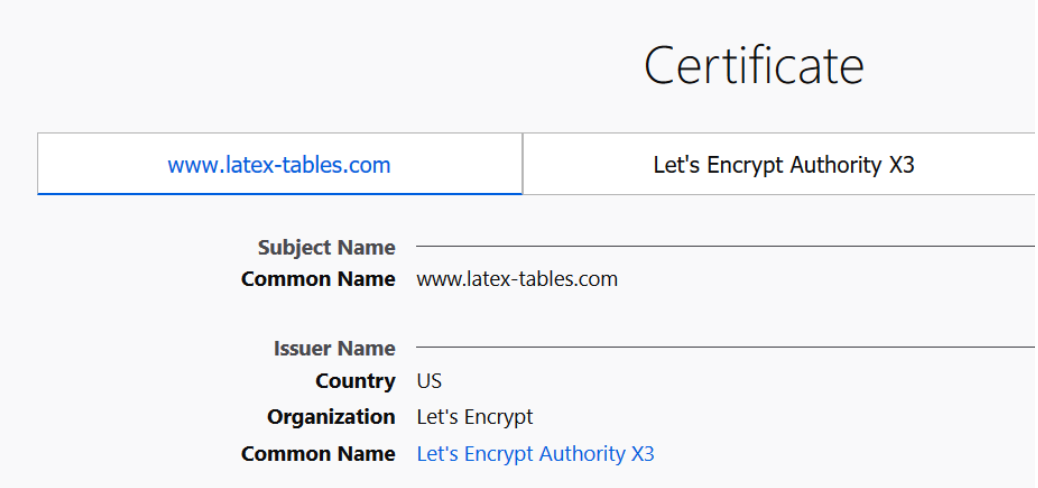

Figure 2.2: Domain Validated Certificate

- **Domain Validated Certificate**: It is a standard SSL certificate. The organization only provides domain name with the certificate signing request to the CA. There is no additional step required. CA validates domain name of the organization to issue SSL certificate. However, these certificates can give assurance that data is being sent and received by the holder of the certificate, there is no guarantee about owner of the certificate. [60]. Fig 2.2 shows the example of domain validated certificate.

- **Organization Validated Certificate**: In this standard SSL certificate, along with the domain name, the organization must provide some additional details such as organization name, locality name, country name and organization unit name. The CA strictly verifies these details and approves the SSL certificate request of the organization. The certificate provides the assurance of the organization. Fig 2.3 shows the example of organization validated certificate.

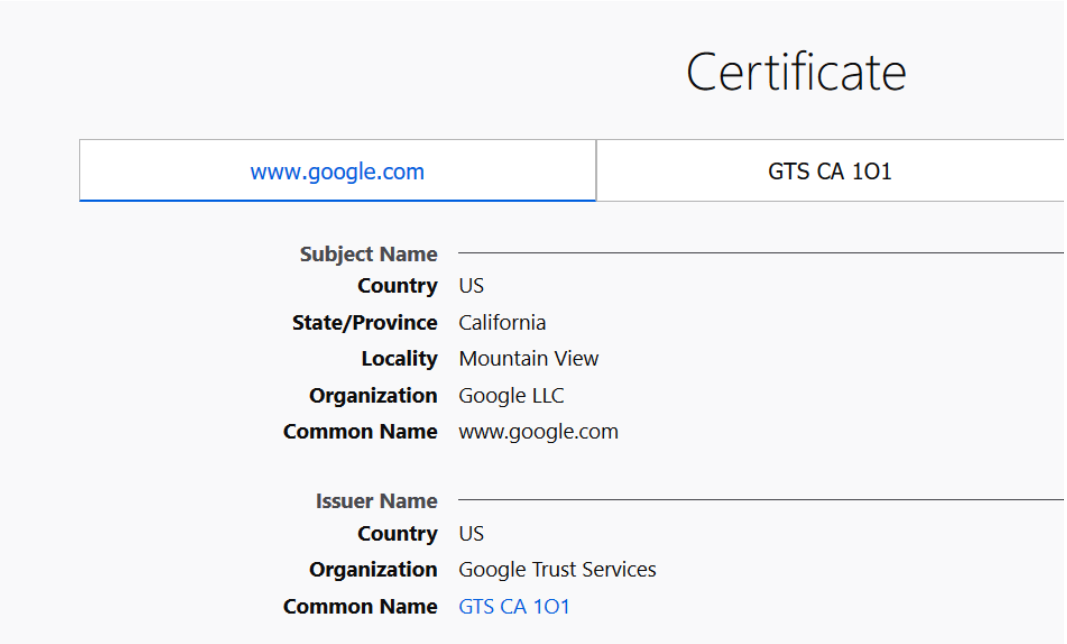

Figure 2.3: Organization Validated Certificate

- **Extended Validated Certificate**: In this standard SSL certificate, the organization must include some additional details. The details include the category of the business and organization registration number. The CA verifies the existence, official records of the organization and also validates that the organization is authorized to use the domain[60]. For extended validated certificates, the green bar is displayed on the address bar which shows the sign of reliability of the website and owner. Fig 2.4 shows the example of extended validated certificate.

- **Wildcard Certificate**: It [21] is not a standard SSL certificate. With wildcard certificate, organization can secure the same level of sub-domains with a single certificate and save money in purchasing certificates for multiple domains. The certificate can be requested by providing the details of sub-domains to CA. The type can be a domain validation wildcard certificate, organisation validation wildcard certificate and extended validation wildcard certificate based on the categorization of validation level. Figure 2.5 shows the example of wildcard domain validated certificate.

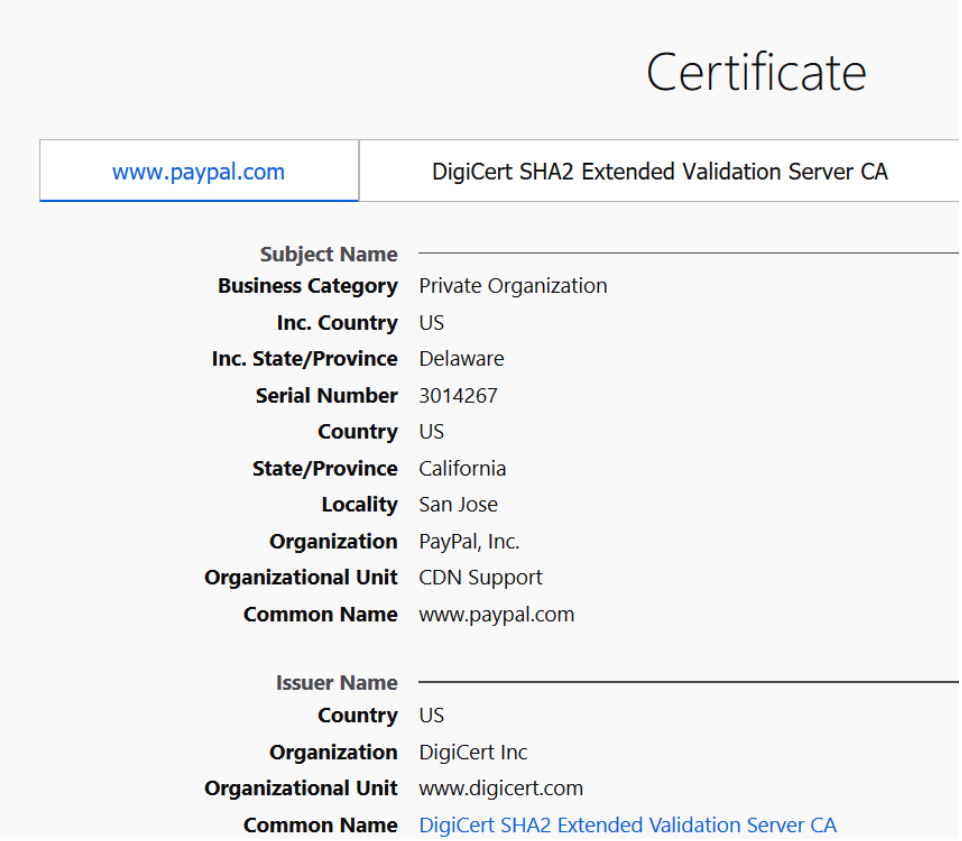

Figure 2.4: Extended Validated Certificate

To summarize, validation level increases along with the cost and trust where domain validated comes at lowest level and extended validated certificate at highest level. The organization validated certificate and extended validated certificate are expensive and reliable compared to domain validated certificate. Since SSL certificate shows more detail information of the organization for extended validated and organization validated certificates which assures that the organization are credible. Moreover, wildcard certificate category is insecure in use. If attacker infiltrate organization's web domain by any chance, they can gain privilege that allows them to create domains. The created domains will be encrypted and protected by wildcard certificate of organization and these domains can be used for phishing purposes by attackers [26].

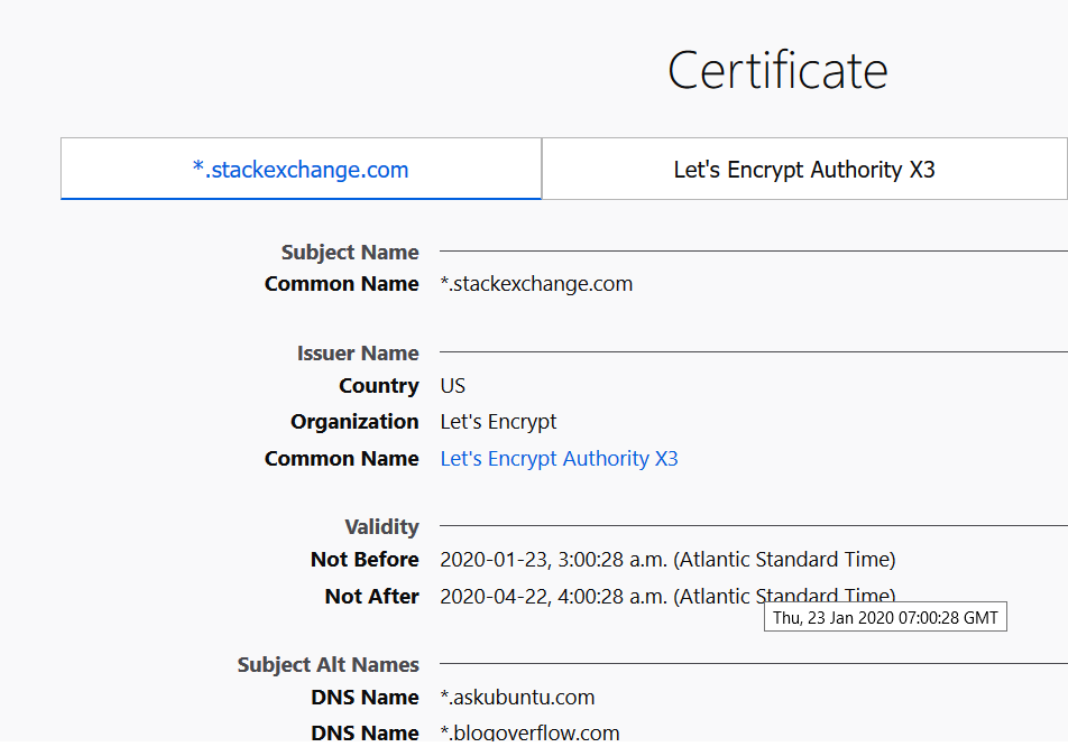

Figure 2.5: Wildcard Certificate

2.4 Chapter Summary and Research Motivation

In chapter, summary of prior works with different techniques were presented. Although all approaches were achieved good performances, but there was no effective technique till now employed to detect HTTPS phishing attacks. It was possibly because:

- Several studies were conducted based on phishing features. These features engineered after recognizing the common patterns of URLs or Web-page used to create phishing websites. These features would not be useful in detecting process if the attackers changed their patterns.
- These studies lacked in defining precise rules to differentiate the phishing and legitimate website.
- The prior works included unnecessary features which was increasing the training as well final decision making classification time.

These limitation of prior works motivated this study. Instead of Web-page or URLs,

SSL certificate of domains were studied to detect HTTPS phishing attacks. As these web certificate had mandatory fields that necessary to provide CA in issuing SSL certificate for registered domain. These fields were analyzed and created as features for identifying attacks. Apart from that, training dataset was generated based on common characteristics of phishing and legitimate classes which later tested based on alternative assumptions. Motivated by the research gaps observed in previous approaches, I focused on the following specific aspects in my research:

- To analyze the feature importance based on sets of features used to understand which features contribute more towards the detection.
- To explore the robustness of detection systems.
- To develop a set of decision rules (using machine learning classifiers) for detecting the category of a domain to implement a Web API.

Chapter 3

Methodology

In this chapter, the approach and techniques used to achieve research objectives are described. The data collection process, feature creation and feature extraction methods along with data normalization and feature selection techniques are discussed. Finally, a web API (Application Program Interface) was built over HyperText Transfer Protocol (HTTP) service, provided a user interface of the detection system.

3.1 Approach

Finding HTTPS phishing sites manually is a bit challenging because of its "legit" appearance and reliability of "lock icon" shown on browser. Attackers create phishing sites with similar look and feel of legitimate sites. This look and feel includes similar layout, colour, font of the web contents on site. The "legit" appearance along with "lock icon", sign of trust at address bar make confuse any user in identifying phishing sites over the internet. So, there is need for a method which could detect the phishing attacks over secure web service effortlessly. Although web browsers contain anti-phishing tools to block phishing sites, attackers still find a new way to compromise these detection methods by changing their attacking patterns. In this circumstance, a robust method should be applied which perform effectively and able to deal with uncertain situations without failure. To create a robust detection method for detecting HTTPS phishing attacks for this study, SSL certificate was taken as an indicator since it has many components such as issuer name, subject name, validity etc which could contain critical information that would assist in recognizing the common patterns of attackers. All the components along with optional fields of domain's certificate were analyzed and identified for creating the useful features. To examine phishing patterns in domains, machine learning classifiers were utilized on collected dataset. Generally, machine learning algorithm quickly analyze and learn the structure of the large set

of data samples. Based on the learnt patterns, it provides a reliable output and classifies the instances as per the label. The performance of machine learning algorithm increases with increase in the size of data samples. This pattern was observed when the data samples of different domain and certificate characteristics were added to categories while generating dataset for this study. The machine learning algorithms-SVM, Random Forest, J48 Decision Tree, Bagging Decision Tree, Boosting Decision Tree, KNN, Naïve Bayes, Deep Neural Network with LSTM were used to build the classification models. These algorithms are trained on a dataset to learn the examples presented in the training data for both legitimate and phishing domain data samples. The reason for using these specific supervised machine learning algorithms was that these algorithms were already used in prior research works that were presented in chapter 2. As the classifiers performs differently with different type of data, these classifiers applied to test which classifier would give the best predictive model for my generated data. The classifiers were compared on True Positive Rate (TPR), False Positive Rate (FPR) and F1 score metrics on classified instances since these gives visualization of errors being made by the classifiers. High FPR means that lots of data samples are unclassified where as high TPR means that most of the data samples are correctly classified. F1 score represents both (FPR and FNR) and measures incorrectly classified cases. By using these classification metrics, I observed number of unclassified and correctly classified instances and then evaluated the performance of classification models. Before building the classification models, the other processes such as data selection, data prepossessing and data transformation place an important role in designing the predictable model. These processes reduce the complexity and improved the quality of data.

To make these processes easier, tool, inbuilt methods, and libraries of Python programming language were used in this study. The Waikato Environment for Knowledge Analysis (Weka) tool[37] is a free software and provides graphical user interface to users. It contains several machine learning algorithms, different functions for data analysis and data pre-processing. Also, it provides graphical visualization of features used in machine learning algorithm. I utilized this tool for attribute selection process, normalization technique, creating classification models, and applying ten-cross validation method. With Weka, it was easy to get and use all the functions at same

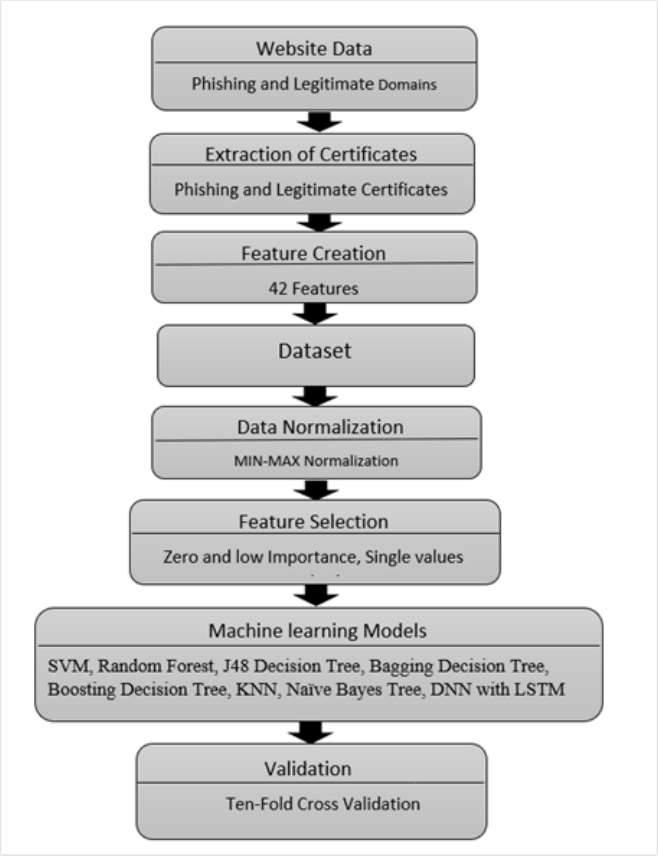

Figure 3.1: Overview of Classification Systems

place. In addition, the Python programming language and its standard libraries were used to fetch the components of SSL certificate and create 42 features that were used to prepared dataset for this study. The advantages of using Python programming language are that it is free and quite easier than other programming languages. Because of having vast libraries support, the demand of Python is increased in several fields. It can use for any application development.

There were three categories:Text, Integer and Boolean features in the dataset used in this study. For text data category, Counter Vectorization feature extraction method was employed because it was not possible to apply text data directly to any machine learning algorithm. It was necessary to first convert text data to vector form or real number. Since the text data content was small, a simple and traditional approach for vector representation was selected in my research. Python inbuilt methods were utilized for implementing boolean and integer data category. The ten-cross validation

method was used to evaluate the performance of trained models. Generally, this ten-cross validation method first randomly partitions the training dataset into ten equal sets of data. Of the ten sets of data samples, a single set remains as the validation data for testing the model, and the remaining nine sets are used as training data. Then the process is repeated ten folds, with each of the 10 sets used exactly once as the validation data [65]. The ten results from the folds are then be averaged to produce a single estimation. The importance of this method is that all observations are used for both training and testing, each observation are used for testing exactly once and at end it provided more accurate estimated accuracy[65].

In the following sections, details about classifiers used are provided along with the feature creation and feature extraction methods. In data section, details about datasets used are presented. These include descriptions of characteristics of collected domains, certificates, features and other techniques i.e. data normalization and feature selection methods. Fig 3.1 shows the overview of the classification system. Steps are given below:

- Different characteristics of phishing and legitimate domains were collected to create website domain dataset.
- Certificates were extracted from collected raw dataset
- The extracted certificates were manually labeled as phishing certificate and legitimate certificate.
- Then 42 features were created which prepared the final certificate dataset.
- Data normalization technique was applied on attributes of certificate dataset for machine learning algorithm
- Feature selection methods were used to remove irreverent features
- Remaining features were used to create the learning models and compared with each other.
- At last, the ten-fold cross validation used to evaluate the model and again compared to finalize model.

3.2 Algorithms

In this section, the general functionality of employed machine learning algorithms for classification models, python libraries, and feature extraction method for creating features is explained. As shown in Fig 3.1, eight classifiers, six libraries, and one feature extraction method were involved to design the detection system.

3.2.1 Definition of Classifiers

The supervised machine learning algorithms[23] were used in my dataset to create the classification models. The predictions were made about instances on the labeled data. The training data used in this study consisted of the pair of input data and output data. The input object was called labeled trained data and output was called desired class. The supervised learning technique learnt the output with examples presented in the training data for both the legitimate and phishing domain input data. The algorithm should be able to generalize from the training data to any unknown data. The importance of supervised learner is that it provides best solution for classification problems and it is simple to understand even having the larger data. In following sections, short description and functionality of each classifier are presented.

Support Vector Machine

The Support Vector Machine which is abbreviated as SVM[36, 63], is commonly used for regression, classification, and detection tasks. The approach of classifier is based on statistical learning theory and kernel method. The method puts the input data into high dimensional feature space. Thereafter, classifier separates the data points of two classes by constructing the hyperplane in the feature space. The hyperplane is decision boundary and data points are support-vectors. As there are many hyperplanes present which separates the data points but only one should be considered to maximize distance of two classes from the separating hyperplane. The hyperplane that maximizes the distance is also called an optimal hyperplane. Based on optimal hyperplane, the test dataset is classified. Figure 3.2 shows the hyperplane, data points for both classes. For correctly classified data points satisfy the inequality

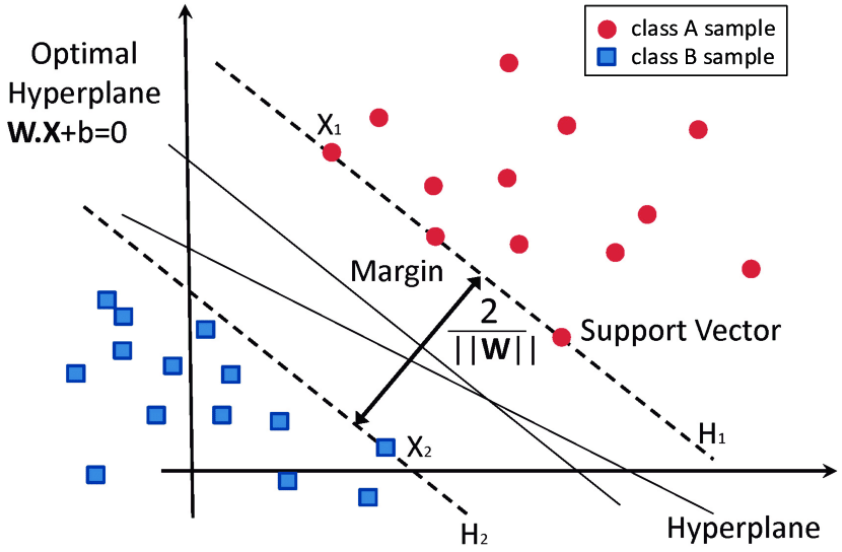

Figure 3.2: Classification of data by support vector machine [36]

Equation 3.1:

$$y^i(w.x^i + b) \geq 1 \tag{3.1}$$

where,

y^i is class A

x^i class B, combining both classes represent training data.

w is weight vector, feature weight gives the relevance information for the discrimination of the two classes.

b is bias, the distance to the origin of hyperplane.

Library for Support Vector Machines (LIBSVM) [29] was installed in Weka tool to implement the model using SVM.

Random Forest

Random Forest[48] is also used for both classification and regression tasks. The classifier is a forest of multiple decision trees. The randomness is added to the model for creating decision trees. It selects a random subset of feature to split nodes and provides a random threshold for the features. Based on the prediction of each decision tree, each tree casts a unit vote to the most popular class at input data. The final result is decided by maximum vote of the classification result. Figure 3.3 shows the semantic diagram of Random Forest. The final result of this classifier is drown by

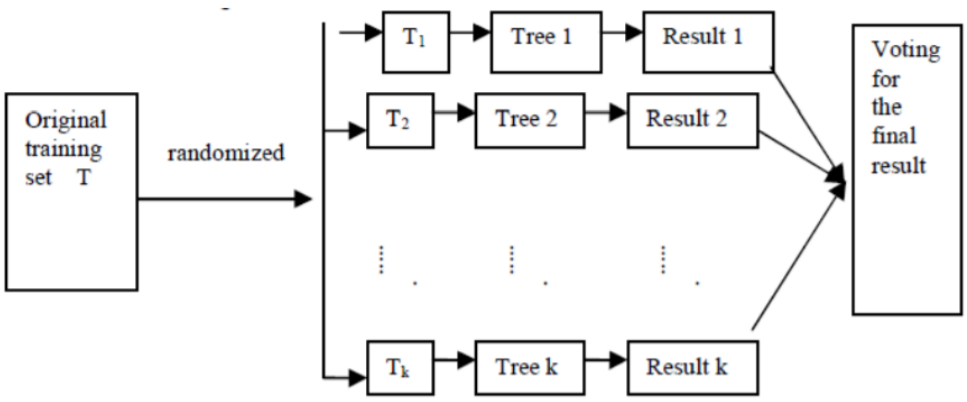

Figure 3.3: Random Forest schematic [36]

ordinary majority vote, the decision function is 3.2:

$$H(x) = argmax_Y \sum_{i=1}^{k} I(h_i(x) = Y) \tag{3.2}$$

where,

H(x) is combination of classification model

h_i is a single decision tree model

Y is the output variable

I( )· is the indicator function

"RandomForest" under the "trees" group was selected in Weka tool to implement the model using Random Forest

J48 Decision Tree

The J48 Decision tree[9, 40] is a C4.5 decision tree classifier, generally used for classification problems. It uses top-down and divides and conquer strategy to build a tree-type structure model. The classifier splits the training data sets based on the chosen nodes that provide highest normalized information gain at each level. It is recursive in nature, so the splitting process continues until it provides a terminal node. The chosen node i.e. attribute indicates the possible outcomes and terminal node i.e. class label indicates final action to be taken. The advantages[45] of C4.5 decision tree are that it handles both continuous and discrete features, missing values and solves the problem of over-fitting by using pruning method (technique to reduce the

complexity and prediction error rate). The classifier 3.3, 3.4 computes information gain ratio for the chosen node:

$$E(S) = \sum_{i=1}^{n} -P_r(C_i) * log_2 P_r(C_i) \tag{3.3}$$

$$G(S, A) = E(S) - \sum_{i=1}^{m} P_r(A_i) E(S_{Ai}) \tag{3.4}$$

where,

E(S) – information entropy of S

G(S,A) – gain of S after a split on attribute A

n – n^{th} class in S

$P_r(C_i)$ – frequency of class Ci in S

m – n^{th} of value of attribute A in S

$P_r(A_i)$ – frequency of cases that have Ai value in S

E(S_{Ai}) – subset of S with items that have Ai value.

"J48" under the "trees" group was selected in Weka tool to implement the model using J48 Decision Tree

Bagging Decision Tree

Bagging Decision Tree[16] is a machine learning ensemble decision tree. The classifier selects subsets of random data from the dataset and uses the subsets to train multiple decision trees. The average prediction is computed by combining all the tree-classifiers. The classifier provides stable performance and reduces variance. It is very similar to Random forest but the difference is Random Forest takes random subset of features along with random subsets of data whereas Bagging Decision Tree takes full set of features with random subsets of data for each tree.

"Bagging" under the "meta" group was selected in Weka tool to implement model using Bagging Decision Tree

Boosting Decision Tree

Boosting Decision Tree[32] is also a machine learning ensemble decision tree. The classifiers train multiple decision tree models by selecting a random set of features where subsequent models try to fix the prediction errors made by prior models[28].

The method continues till no improvement is required. It is the average of every model prediction but more weighted on better performance on the training dataset. With the averaged prediction, it tests new data. The equation 3.5 used to improve error of the classifier. Each weak learner produces an output hypothesis, $h(x_i)$, for each sample in the training set. At each iteration t, a weak learner is selected and assigned a coefficient α_t such that the sum training error E_t of the resulting t-stage boost classifier is minimized[15].

$$E_t = \sum_i E[F_{t-1}(x_i) + \alpha_t h(x_i)] \tag{3.5}$$

Here, $F_{t-1}(x)$ is the boosted classifier that has been built up to the previous stage of training and $\alpha_t h(x)$ is the weak learner that is being considered for addition to the final classifier[15].

"AdaBoostM1" under the "meta" group was selected in Weka tool to implement the model using Boosting Decision Tree

KNN

KNN[10, 69] stands for k-Nearest Neighbors, uses for classification tasks. The classifier is a non-parametric and instance-based learner. For training the model, the classifier observed k training-instances called optimal k-values. By using an optimal k value, the category of new instance has predicted i.e. it compares the nearest points and classifies the class based on closest similarities between them. Fig 3.4 shows the KNN classification where the new data needs to categorize. Here, if k=3, then it categorize to class 2 because in circle there are two triangles and one square. By majority voting, the final decision has taken. In equation 3.6, the formula for computing the largest probability for x

$$P(y = j|X = x) = \frac{1}{K} \sum_{i \in \mathcal{A}} I(y^{(i)} = j) \tag{3.6}$$

where,

$\mathcal{A}$ is set of points in the training data that are closest to x.

K is optimal k values

y is class label

I(x) is the indicator function "IBk" under the "lazy" group is selected in Weka tool to implement the model using k-Nearest Neighbors, where k=3.

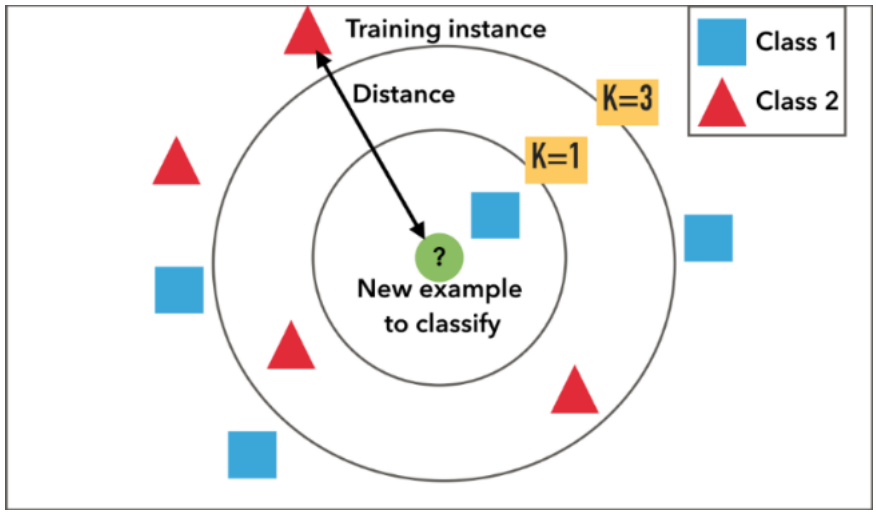

Figure 3.4: KNN Classification [27]

Naïve Bayes

Naïve Bayes[35, 11] is a probabilistic machine learning classifier, based on the Bayes theorem for classification purposes. The classifier assumes that each feature is equal and independent to each other. The presence or absence of any feature does not affect the other feature. The equation 3.7 is the Bayes theorem, where X is defined as $x_1, x_2 \ldots\ldots x_n$ and X is number of features where y is class label. The classifier computes the probability of each class based on the probability where evidences belongs to particular class.

$$P(y/X) = \frac{P(X/y)P(y)}{P(X)} \tag{3.7}$$

where,

P(y) is the probability of hypothesis y being true.

P(X) is the probability of the evidence.

P(X/y) is the probability of the evidence given that hypothesis is true.

P(y/X) is the probability of the hypothesis given that the evidence is there. "NaïveBayes" under the "Bayes" group was selected in Weka tool to implement the model using Naïve Bayes

DNN with LSTM

DNN stand for Deep Neural Network [55, 17], based on neural network concept. It uses a mathematical model to process the data in depth. Just like the neural network

model, each layer connected to one another, but it uses more complex architecture with multiple layers for feature learning and estimating the output. Here, the LSTM technique of DNN is used to create the classification model. LSTM stands for Long-short term Memory, is a class of artificial recurrent neural networks. LSTM has chain like structure which consists of four layers: a memory cell, a input gate layer, a output gate layer and a forget gate layer. The Fig 3.5 shows the architecture of the LSTM. The process repeats in every module, where each individual module passes the message to next. The memory cell is responsible for memorizing the previous task and keep track of the dependencies between the elements in the input sequence. The input gate controls the extend of adding information to the cell, the forget gate is responsible for removing the less important or no longer required information and the output gate puts filter on value in the cell, value is used to compute the output activation of the LSTM unit. The advantage of LSTM is that it solves long-short dependencies, problem of Recurrent neural network (RNN) (the model is not able to use the previous learn word in present task) by modifying the processing unit in the architecture.

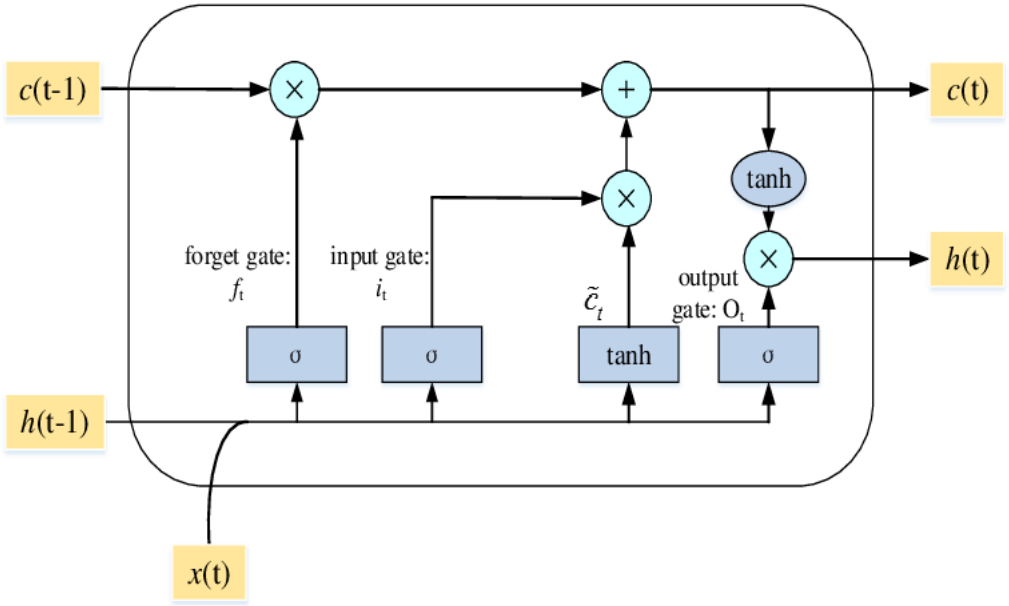

Figure 3.5: LSTM architecture [67]

WekaDeeplearning4j[46] package was installed in Weka to implement the classifier under the "Dl4jMlpClassifier" model. In the configuration window of the

Dl4jMlpClassifier provided the layer specification option to create network architectures by stacking listed neural network layer types.

3.2.2 Python libraries

Python libraries carry several inbuilt functions and provides access to install the modules or packages in python framework for specific tasks. These functions provide the standard solution for any problem. The advantage of these libraries is that along with installation of framework, they also provide the whole package for the operating system. In following section, the name and brief description of used libraries for the creation of feature is explained.

OpenSSL Cryptographic

The OpenSSL Cryptographic library[2] implements a wide range of cryptographic functions and sub-libraries. The sub-libraries are used to implement individual algorithms that include symmetric encryption, public key cryptography and key agreement, certificate handling, cryptographic hash functions and a cryptographic pseudo-random number generator. The OpenSSL's certificate functionality contains X509 object to call certificate display and signing utility. Because of cryptography, the SSL certificates are encrypted in X509 Distinguished Encoding Rules (DER) format. It is not possible to retrieve the SSL certificate details without decrypting X509 DER. The X509 object called to fetch SSL certificate information such as issuer components, validity, subject components, optional field etc.

SSL and Socket

SSL library[6] is programming library, supports a socket-like wrapper that encrypts and decrypts the data by going over the socket with SSL protocol. It also identifies the validity of the server. This library supports method called getpeercert(), to retrieve the certificate of other side of the connection.

Socket library[6] provides a socket object and is used to create server and client connection on a given address family, socket type and protocol number. To create communication, `AF_INET` and `SOCK_STREAM` parameters pass. `AF_INET` refers to the

address-family internet protocol version 4 (ipv4) and `SOCK_STREAM` means socket connection type such as connection-oriented transmission control protocol (TCP). The connection was made by passing these parameters over the host address on given port.

Publicsuffix2

Publicsuffix2[4] basically builds public suffix, registered domains and top-level domains list and supports method to obtain the lists. The method called "get_tld()" to fetch out the top-level domains of the phishing and legitimate domains. Before fetching out, it is necessary to ensure that the domain includes a valid TLD. The boolean parameter strictly provides a solution and checks whether the domain is valid or invalid TLD. In case of an invalid TLD, function returns a None value.

Seolib

Seolib[5] provides the search engine optimization (SEO) metrics of any domain. The library supports the method called "get_alexa" to get the Alexa ranking of the domains.

Scikit-learn

Scikit-learn[19] is a python machine learning library and is also called as sklearn. The library supports several functions for clustering, classification, regression including machine learning algorithms. The python libraries are integrated with sklearn to provide solutions. Sklearn library was imported for text analysis and calculation of Euclidean and Kullback-Leiber distance features.

Scipy

The scipy library[20] is used for mathematical computing. The library is built on the NumPy array object and contains methods for optimization, linear algebra, integration, interpolation, special functions etc. This library imported to call `ks_2samp` method for computing Kolmogorov-Smirnov statistic feature.

3.2.3 Count Vectorization

Count vectorization[57] is a simple approach to do text analysis. It extracts features from the text for the machine learning algorithm. Basically, it converts text into words as features and each word is known as "gram". It could be "bigram" or "trigram" as well, but it depends on how many words tokenize at a time. It counts the occurrences of the words in the text and provides the vector value. It treats every text as separate document, then covert each text into words and make a list of all unique vocabulary. For example, we have four texts

"It was the great day", "It was the worst day", "It was the best feeling", "It was the worst feeling"

list of vocabulary = "It", "was", "the", "great", "day", "worst", "best", "feeling". By counting the number of occurrences in the text with the list, it provides the vector value of each line.

"It was the great day" = [1, 1, 1, 1, 1, 0, 0, 0]

"It was the worst day" = [1, 1, 1, 0, 1, 1, 0, 0]

"It was the best feeling" = [1, 1, 1, 0, 0, 0, 1, 1]

"It was the worst feeling" = [1, 1, 1, 0, 0, 1, 0, 1]

It converts text into a matrix of word count and encodes any new text by using present list of vocabulary. If there is an occurrence of any new words, it adds to list and transform it as the vector with equal shape the vocabulary list. For this study, only the array of vectors to calculate text-features was used.

3.3 Importance of Classifiers

Machine learning classifiers [23] provides an ability to understand the different types of complex data present in the dataset. Even the demand of these classifiers highly increase in network and security since learning approaches successfully identify the malicious contents without any human involvement. The benefits of classifiers is listed below[25]:

- Machine learning methods help identify and extract features from any possible important data among all data.

- The classifier automates the whole processes to provide the predictive analysis of large volume of data. It reduces the cost and save time to carry out these heavy tasks.

- It quickly learns the trends and patterns of huge data. Based on the learnt patterns, it could easily categorize new incoming data.

- There are some tasks which might have complex data and the relationship between input and output are not defined. In this circumstances, the machine learning algorithms are able to adjust internally and produce output for large sample of data by implicitly creating relation between input and output function.

The discussed classifiers in above section were also employed in previous works. Apart from benefits, the reason of selecting these specific classifiers was to examine the performance on my generated dataset. In all, J48 decision tree was used in generating dataset and designing the final detection system since it made easier to understand importance of involved features by visualizing in tree structure. The J48 decision tree classifier result provided clarity and transparency to the decision-making process with assigning specific values to each involved features.

3.4 Data

In this section, the data along with features used in the dataset are described. Only one dataset was used to build the classification models. The purpose was to generate the datasets (raw dataset and certificate dataset) and then used the certificate dataset to create features and trained model. All the domains data, certificate data and main dataset are elaborated in the below sub sections.

3.4.1 Data Description

Selecting and collecting data samples were crucial phase of this research study. First, I determined the appropriate data sources to collect and select domain data on the internet. After identifying these sources, an automated scripts using Python were written to download SSL certificates from the shortlisted domains, and to extract

features from the certificates. In this section, details about data and important indicators in generating datasets are explained. The description of each process used in this thesis is given below:

- Data selection and data collection: The selecting process was based on the characteristics of domains and certificates. These characteristics were observed while generating the dataset and provided patterns about phishing/legitimate categories. These patterns made it easier to analyze how attackers generate SSL certificates for spoofed websites [31, 70]. In this research, total 160 certificates were used. As there is no ideal number exist for machine learning classifiers to learn data patterns, I tried to use all the freely available data in this study which is elaborated more in chapter 4.

  Characteristics of Domains:

  - Top-level domains: These top-level domains ".com", ".net", ".org", ".uk", ".ca" were found commonly in legitimate domains whereas ".site", ".xyz", ".icu", "tk", ".online", ".live" were observed in phishing domains. These top-level phishing domains are confirmed by Proofpoint Domain Fraud Report. Although these TLDs of phishing domains are also used in legitimate domains but frequency is quite less [61].
  - Length: Long length was found for phishing domains. For example, "appleid-support-update-account-supportupdate42299codeanyapp.com" whereas for legitimate domains, short length was observed.
  - Sub-domains were observed in phishing domains. For example, "ativar-net.sslblindado.site".

  Characteristics of Certificates:

  - Certificate issuer: All top-level Certificate Authority issuer such as "GTS CA 1O1", "Go Daddy Secure Certificate Authority - G2", "Sectigo Secure Server CA" were observed for legitimate domains whereas "Let's Encrypt Authority X3", "CloudFlare Inc ECC CA-2", "cPanel, Inc. Certification Authority" were found commonly in phishing domains.

  - Types of certificate: Domain validated and Wildcard certificate were found commonly in phishing domains whereas Organization and Extended validated certificate were observed for legitimate domains.

  The legitimate domains were collected from Alexa Rank[1] and Moz's Domain Authority[2]. Both Alexa Rank and Domain Authority completely evaluates listed domains. But the difference is that Alexa lists the domains by estimating internet traffic whereas Marketing Software(Moz)'s Domain Authority measures the links of pages in Domain Authority score[1]. On the other hand, phishing domains were assembled from PhishTank[3]. The PhishTank contains verified phishing domains. The verification is done by third parties such as team experts and volunteers before adding to blacklists. At the end, total 60 legitimate and 60 phishing confirmed domains were collected. These confirmed domains helped in analyzing the important indicators used in generating SSL certificates for registered domains.

- Certificate Collection: The SSL certificates were fetched in X509 DER format from collected domains. The SSL certificate used by legitimate domain was a legitimate certificate and SSL certificate used by phishing domain was phishing certificate. The count of domains was more than 100. It would take a lot of time and effort if the SSL certificates extracted one by one from collected domains. To make the process more efficient, the script was implemented in Python which would download all SSL certificates and saved in CSV file at once. In the script, the input was collected domains and the output was fetched SSL certificates in X509 format. The script contains the socket programming and ssl libraries for implementation process. The fetched SSL certificates was in DER extension. I converted DER to Privacy-Enhanced Mail (PEM) extension to save SSL certificates in Comma-Separated Values (CSV) file. Although X509 object could be encoded in both DER and PEM, PEM extension is simple to convert into any readable data compared to DER extension. There were two reasons for saving SSL certificates. First, a connection between server and client

[1] `https://www.alexa.com/topdomains`
[2] `https://moz.com/top500`
[3] `https://www.phishtank.com/`

to get a certificate was not required anymore. Second, while doing the socket connection for phishing domains, it might give an error. The phishing domains were activated for a very short span of time because blacklisting of phishing led the attackers to frequently change the domains.

3.4.2 Feature Creation

After analyzing important and useful information in the SSL certificate, total of 42 features were considered. The belief was that these features must have some contents which indicated how attackers generated SSL certificate for phishing domains. The features were categorized as text, boolean and integer. For some features, the components were directly taken from the certificate and others were computed by components but not directly used. The separate script was implemented to create features for all the phishing and legitimate certificates and saved in CSV file to prepare my final dataset. Out of 42, 40 features were based on features used in [64]. The importance was to inspect the attacker's behavior and pattern by using these below features.

- **Boolean features**: Based on components present in the SSL certificate, conditional statements were applied to execute the boolean features. The condition implied the rule for each feature
  - SubjectCommonNameIp: Checks if subjectCN is an IP address instead of domain.
  - Is_extended_validated: Checks if certificate is extended validated certificate. Identify based on the business category or registration number
  - Is_organization_validated: Checks if certificate is organization validated certificate. Identify based on the subject Organization name
  - Is_domian_validated: Checks if certificate is domain validated. Identify based on the business category or registration number and organization name should not be present.
  - SubjectHasOrganization: Checks if subject component has Organization (O) field

  - IssuerHasOrganization: Checks if issuer component has O field
  - SubjectHasCompany: Checks if subject component has Company (CO) field
  - IssuerHasCompany: Checks if issuer component has CO field
  - SubjectHasState: Checks if subject component has State (ST) field
  - IssuerHasState: Checks if issuer component has ST field
  - SubjectHasLocation: Checks if subject component has Location (L) field
  - IssuerHasLocation: Checks if issuer component has L field
  - Subject_onlyCN: Checks if subject component has only Common Name (CN) field
  - Subject_is_com: Checks if subject component CN is a '.com' domain.
  - Issuer_is_com: Checks if issuer component CN is a ".com" domain.
  - HasSubjectCommonName: Checks if CN is present in subject component
  - HasIssuerCommonName: Checks if CN is present in issuer component
  - Subject_eq_Issuer: Checks if subject component = issuer component
  - Selfsigned: Checks if certificate is self-signed
  - Is_free: Checks if the certificate is free generated
  - Is_WildCard_Validated: Checks if certificate is wildcard certificate
  - Is_domainMatchCN: Checks if domain name matches with SubjectCN

- **Integer features**: This category was also called as calculative features. In order to get the features, the calculations were done by using the components of certificate.

  - SubjectElements: Number of details present in subject component
  - IssuerElements: Number of details present in issuer component
  - SubjectLength: Number of characters of whole subject string
  - IssuerLength: Number of characters of whole issuer string
  - ExtensionNumber: Number of extensions contain in the certificate

  - DaysValidity: Calculates days between not before and not after days
  - Domain Ranking: Calculates ranking of domain by estimating traffic of the domain

- **Text features**: These features were taken subject or issuer's CN component of the certificate as one parameter. Both were string data types and the required components to validate any types of certificate. To calculate the distances, the considered two parameters would be "google.com" certificate and an incoming certificates' CN because the Alexa rank of "google.com" is 1, "google.com" certificate was kept as constant parameter for all the incoming certificates. Moreover, text data must be converted into vector or real number, so the machine learning algorithms could understand. The count vectorization, a feature extraction method was used to get vector values which is explained already in previous section of this chapter.

  - SubjectCommonName: Shannon entropy (Minimum number of bits per character needed to encode the string) of subject CN
  - Euclidian_Subject_Subjects: Euclidean distance (Distance between two points in euclidean space) of subject CN among all subjects
  - Euclidian_Subject_English: Euclidean distance of subject CN characters among English characters
  - Euclidian_Issuer_Issuers: Euclidean distance of issuer CN among all issuers
  - Euclidian_Issuer_English: Euclidean distance of issuer CN characters among English characters
  - Ks_stats_Subject_Subjects: Kolmogorov-Smirnov statistics (Distance between the empirical distribution functions of two samples) for subject CN in subjects
  - Ks_stats_Subject_English: Kolmogorov-Smirnov statistics for subject CN minimal number of bits per symbol required to encode the string in English characters
  - Ks_stats_Issuer_Issuers: Kolmogorov-Smirnov statistics for issuers CN in issuers

- Ks_stats_Issuer_English: Kolmogorov-Smirnov statistic for issuer CN minimal number of bits per symbol required to encode the string in English characters
- Kl_dist_Subject_Subjects: Kullback-Leiber distance (Compute mutual information between two data samples) for subject CN in subjects
- Kl_dist_Subject_English: Kullback-Leiber distance for subject CN characters in English characters
- Kl_dist_Issuer_Issuers: Kullback-Leiber distance for issuer CN in issuers
- Kl_dist_Issuer_English: Kullback-Leiber distance for issuer CN characters in English characters

3.4.3 Data Normalization

Normalization[70] is the process to make the values of numeric features in the dataset to a common scale. In this thesis, datasets containing text and integer features were in very different ranges. In order to normalize these two categories of features, I utilized Weka MIN-MAX normalization method, Fig 3.1. This normalization method re-scaled the range of features between [0,1]. The main advantage of using data normalization was to increase numerical stability, reduce the training time and also increase the accuracy of classification models [70]. The equation 3.8 is used to compute MIN-MAX Normalization:

$$N = \frac{n - min(n)}{max(n) - min(n)} \tag{3.8}$$

Where,

N= Normalized value for n

n=The value to be normalized

min(n)=The minimum value of n

max(n)= The maximum value of n

To explain in detail, I have provided an example. Before normalization, Euclidian_Subject_Subjects(text feature) value was 1.414213562 and SubjectElement (integer feature) was 5. I observed, the categories were in different ranges where integer feature ranged between 0-10 while text feature ranged between 0-1.0. Also, the validating accuracy was 55% taken 9.21 seconds of training time for deep neutral network

with LSTM. Thereafter, I applied MIN-MAX normalization method using Weka tool on my dataset. The value of Euclidian_Subject_Subjects(text feature) was changed 1.414213562 to 0.649311 and for SubjectElements(integer feature), it changed 5 to 0.444444. The accuracy of deep neutral network with LSTM using normalized data was also increased to 75% taken 5.17 seconds of training time. This example shows how this method increased the performance of classifiers in this research.

3.4.4 Feature Selection

Feature Selection[44] is a process of selecting the most useful features in a dataset in order to decrease the training speed with increase accuracy of the model. I manually removed some unnecessary features from the data-set by using Weka tool. The Weka provided an option to remove and add attribute or instance. Below methods have taken for feature selection.

- Features with a single unique value: With this method, the attributes observe which give the single value for all the instances. These attributes having zero variance would not be useful for machine learning classifiers.

- Features with zero importance: With this method, those features analyze which have zero participation in classification. In short, by removing these features, the performance of the model would be affected.

3.5 Web API and Proof-of-Concept Web Application

Web API provides application interface to user over the internet. It handles user request and returns output through web service. As there are many popular web frameworks that supports an environment to develop a Web API using Python programming language. In this study, Flask web framework[13] was used. The advantages are easy to install and contains several tools, libraries, and technologies to build a web application. The purpose was to implement the trained model in the form of Web API by using the Flask framework. The Web API automated the detection processes of the model where a user could verify the domain whether it is legitimate or phishing. The framework deployed the trained system functionalities in back-end and

provided identified result in front-end. HTML and Cascading Style Sheets(CSS) were used for creating page structure and styling web page that included colors,layout, and fonts. Moreover, I added dynamic functionality in web page by using JavaScript method. Based on the output of extract certificate, feature creation and decision maker functions, JavaScript action method call to response the user input request. The scripts were used in preparing training dataset, implemented in Web API for extracting certificate and creating features functions. For decision maker function, the decision rules were coded as control statements in Python programming language. These decision rules were obtained from J48 decision tree classification model. This model was chosen for developing the detection system based on the evaluations discussed in chapter 4.

3.5.1 Implementation

The functionalities and implementation flow chart are included in this section. Main functionalities of framework are:

- It must identify phishing and legitimate domains based on machine learning classifier rules.
- It must return an error message on wrong input data.
- It must successfully create features using different Python library method.
- It must extract the certificate from the website without any error because phishing domains are active for very short span of time.
- It would not detect any types of web attack such as Distributed Denial of Service (DDOS) etc.
- Input data should be a website domain name.

Fig 3.6 shows the flow chart of the framework. The flow chart of the framework is explained below:

- Input is domain of the single website.
- Extraction of certificate from the input domain.

- Features are created for single certificate.
- Machine learning classifier rules are employed to classify category of the domain.
- The result shows that a domain was legitimate or phishing domain.

3.5.2 Testing

To test my Web API, three types of testing were employed to make sure that API functionalities were working as expected.

- Unit Testing: In Unit testing, all methods and functions used for features and API were checked individually. There were three functions that were called-certification extraction, feature creation and decision maker and tested independently.
- Integration Testing: All called functions were integrated and tested such as extraction of certificate and features creation methods were combined to test. They were checked whether they worked well together or not.
- Functionality Testing: In this testing, the main requirement examined whether the application was able to do domain verification or not. To make sure, I used the domains(training and testing domains of classification model) in functionality testing of Web API.

3.6 Chapter Summary

In this chapter, the methodology used for the proposed system was discussed. An overview of different techniques used for creating classification models were provided. These include:

- List of algorithms used in the model.
- List of python inbuilt libraries and methods used in creating features.
- Different characteristics and 42 features used.
- Discussions on data normalization and feature selection.

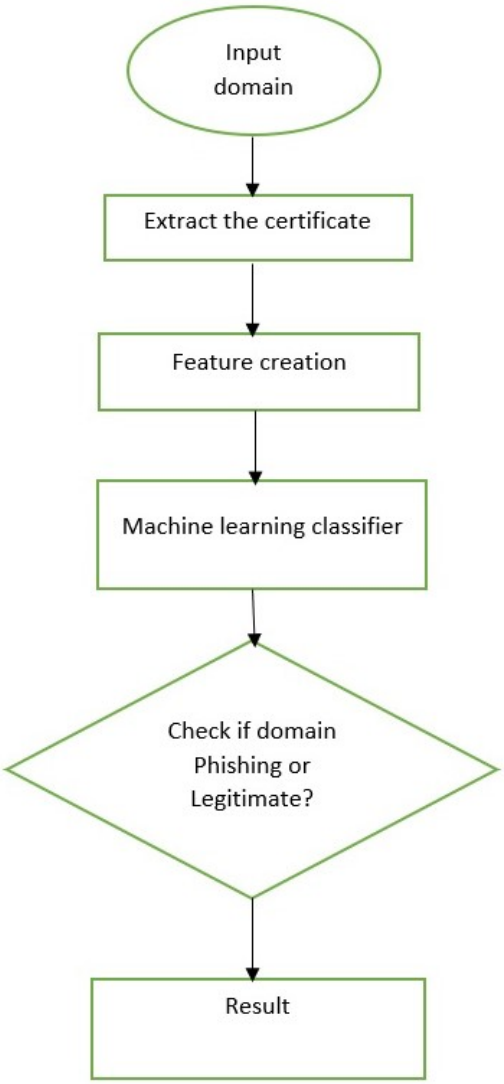

Figure 3.6: Flow Diagram of Web API

- Details on web API that is developed..

All the results in different phases of this research are explained in chapter 4. The results in these phases started with the generation of final dataset, then training performance of different classification models, followed by validating classification models results using ten-fold validation method, and at the end robustness testing of the classifier with different sets of features.

Chapter 4

Experiments And Results

In this chapter, all the evaluations and results are described in detail. These include techniques, characteristics, and feature analysis of data as well as different test cases and respective results for checking robustness of the detection system.

4.1 Classifiers Result

The main objective was to explore the robust machine classifier with set of features for distinguishing between legitimate and phishing domains. To achieve this, the collected domains and 42 features were considered to create final certificate dataset. Initially small dataset of 60 domains (30 legitimate and 30 phishing) with J48 decision tree classifier were used. Later, more domains were added till there was no more change in performance and decision tree structure of the classifier. Though the domains were added based on the domains and certificates characteristics in both legitimate and phishing category after analyzing the inconsistent behaviour of data samples. The example of inconsistent behaviour of data sample was a phishing data sample behaved as a legitimate. Then that data sample examined and based on its characteristic behaviour, more data samples added in both phishing and legitimate category. Along side, the classifier performance was also analyzed until it showed constant accuracy

No. of Domains	No. of Involved Features	Accuracy
60 (Result 1)	3	95%
70 (Result 2)	4	95.5%
80 (Result 2)	4	95%
90 (Result 4)	4	95.7%
100 (Result 5)	5	96%
110 (Result 6)	10	96.6%
120 (Result 7)	10	96.6%

Table 4.1: Results with different set sizes

by using Weka. As the examined characteristics were already discussed in section of chapter 3. At the same time, the balancing of both legitimate and phishing instances was also been taken care. The table 4.1 shows results with different set of size and their accuracy and involved features. The purpose was to generate the dataset with data specifications for each category. This dataset was later used to build the detection models by using other classifiers. There were total of seven results obtained. Each result (Result 1 to Result 7) referred to J48 classification model performance with different set of data samples and features. In the last two results, Result 6 and Result 7 had same accuracy and number of involved features with different domain sizes. This indicated that there was no further change in performance and tree structure that confirmed final dataset. From result table, I observed that as the data size increased, the performance of the classifier also increased.

Once the training dataset finalized, the feature selection and data normalization methods processed with the help of Weka tool. By using feature selection methods, I found below irrevalent features:

- SubjectCommonNameIp, HasSubjectCommonName, HasIssuerCommonName: Boolean features with a single unique value. The weight of the label were either all True or all False.

- IssuerHasCompany, Subject_eq_Issuer, IssuerHasOrganization, Issuer_is_com: Boolean features with a zero importance. The weight label was very less for one label. For example, IssuerHasCompany feature had True=1 and False= 119 instances. This showed that the features had zero participation in the classification model.

- SubjectHasLocation, SubjectHasOrganization, SubjectHasState, IssuerHasLocation: If the labels' weight of two features were same, one of the features could be removed to increase model interpretability. For an example, as the labels, True=72 and False=48 were same for boolean features-SubjectHasLocation, SubjectHasOrganization and Is_domian_validated. SubjectHasLocation, SubjectHasOrganization attributes were removed from dataset.

Thereafter, final normalized dataset with 31 features was fed into the classifiers i.e. SVM, Random Forest, J48 Decision Tree, Bagging Decision Tree, Boosting Decision

Classifier	Accuracy	Legitimate			Phishing		
		TPR	FPR	F1	TPR	FPR	F1
J48	96.6 %	98.3%	5%	96.7%	95%	1.7%	96.6%
Random Forest	95%	93.3%	1.7%	95%	98.3%	6.7%	95.9%
KNN	92.5%	98.3%	13.3%	92.9%	86.7%	1.7%	92.0%
Bagging DT	97.5%	98.3%	3%	97.5%	96.7%	1.7%	97.5%
Boosting DT	100%	100%	0%	100%	100%	0%	100%
Naïve Bayes	84.1%	80%	11.7%	83.5%	88.3%	20%	84.8%
SVM	84.2%	78.2%	10.1%	84%	87%	20%	84%
LSTM	83.3%	71.7%	5%	81.1%	95%	28.3%	85%

Table 4.2: Classification Results

Tree, KNN, Naïve Bayes Tree, DNN with LSTM for training purpose. The Weka tool was again utilized to train the classification models. Table 4.2 shows the classification results of the classifiers. According to the table, almost all the classifiers performed well in terms of TPR and FPR and F1 scores. The classifiers such as Boosting Decision Tree achieved 100% of F1 score for both legitimate and phishing certificates. Other classifiers KNN, J48 Decision Tree, Bagging Decision Trees, Random Forest have achieved above 92% score for both categories whereas Naive Bayes, SVM, DNN with LSTM were given lower than 85% score as compared to other classifiers. It was important to note the classification result for J48 Decision Tree was still same by comparing the Table 4.1 and Table 4.2. It meant that the used processes did not effect the performance of J48 Decision Tree. Apart from training scores, the ten-fold cross validation method was also applied to evaluate the classification models. Table 4.3 shows the ten-fold cross validation results for all the classifiers. As per the table 4.3, KNN, DNN with LSTM, Naive Bayes, SVM scored lower than 85%. On the other hand, all the decision tree based classifiers performed well. After analyzing the performances of each classifier in both the tables 4.2, 4.3, J48 Decision tree classifier was chosen to design the detection system. Even though other classifiers, Bagging Decision Tree, Boosting Decision Tree gave slightly higher scores than J48 Decision Tree, J48 Decision Tree provided a visualization of involvement of each feature as well as the transparency in the result. Also, the classifiers other than J48 Decision Tree acted as a black box. It was hard to recognize the individual weight of each feature at the end of the result.

Classifier	Accuracy	Legitimate			Phishing		
		TPR	FPR	F1	TPR	FPR	F1
J48	92.2%	91.1%	6.7%	92.1%	93.3%	8.9%	92.3%
Random Forest	91.1%	91.1%	8.9%	91.1%	91.1%	8.9%	91.1%
KNN	85.8%	81.7%	10%	85%	90%	18.3%	90%
Bagging DT	91.5%	93.8%	10%	92%	90%	6.2%	91.7%
Boosting DT	95%	96.7%	6.7%	95.1%	93.3%	3.3%	94.9%
Naïve Bayes	81.6%	80%	16.7%	81.4%	83.3%	20%	82%
SVM	80.2%	82.9%	22.9%	80.6%	77.1%	17.1%	79.4%
LSTM	75%	80%	30%	76.2%	70%	20%	73.7%

Table 4.3: Ten-Fold Cross Validation Results

Classifier	Accuracy	Legitimate			Phishing		
J48		TPR	FPR	F1	TPR	FPR	F1
With Text-features	96.6 %	98.3%	5%	96.7%	95%	1.7%	96.6%
Without Text-features	94.16%	95%	6.7%	94.2%	93.3%	5%	94.1%

Table 4.4: Classification results of Decision Trees

4.2 Decision Tree Systems

As discussed, a total of 31 features were considered to make the prediction. Out of 31, there were 13 text-features which created using the statistical learning of CNs of subject and issuer name. To understand the importance of the text-feature, the dataset was processed by removing text features using the filter option in the Weka. It revealed the involvement and threshold estimation of features in different decision trees. The performance of the algorithm was analyzed each time. The two decision trees are obtained i.e With text features and Without text features with two different sets of features. Table 4.4 shows the classification results of decision trees. As the result shown, the classifier With text features achieved higher scores than classifier Without text feature in both the legitimate and phishing category. Eventually, the participation of new features also observed in the Without text feature decision tree which could not be ignored. Since there was not much difference in performance and addition of new features in the decision tree, both the decision trees classifiers were considered for the system design. Along side, I built two separate Web API with both With text-feature and Without text-feature decision tree rules.

4.3 Models Uncertainty

In this section, the different test cases are demonstrated. The purpose of preparing these test cases was to do a model variation test [56], where model specific assumptions were changed or replaced with alternative assumptions. As discussed earlier, the dataset was generated by analyzing domains and certificate characteristics for both legitimate and phishing categories. Here, by keeping all these characteristics in mind, the four sets of test cases were prepared. Each test case had 72 data with equal balance of phishing and legitimate domains. The test cases would analyze the trained models uncertainties and test whether their performances changes with change in characteristics. For creation of testing datasets, again the same implemented scripts were used for extracting certificate and creating features. But only involved features in both the classification decision tree models were included. For With text-feature, there were 10 features involved whereas Without text-feature, 11 features were participated in tree structure. Table 4.5 shows the scenarios of four test cases where similar characteristics (represented as 0's) between test and training data samples and dissimilar (different) characteristics (represented as 1's), are given. All domains used in the four test cases are listed in Figure B.1 in Appendix B. The similar characteristics are already presented in chapter 3 and the dissimilar characteristics are described below:

Characteristics of Domains:

- Top-level domains: The top-level domains, ".com", ".net", ".org", ".uk",".ca" were observed in legitimate domains were considered for phishing domains whereas ".site", ".xyz", ".icu", "tk", ".online", ".live" which were found commonly in phishing domains were treated for legitimate domains.

- Length: Long length was observed in phishing were taken for legitimate domains where as short length were found commonly in legitimate domains were considered for phishing domains. For example, "www.washingtonpost.com" for legitimate domain and "it.com" for phishing domain.

- Sub-domains were found commonly for phishing domains were considered for legitimate domains. For example, "cs.dal.ca" is the sub-domain of "www.dal.ca". Alexa provides same rank to both the domains.

Case	Legitimate	Phishing
1	1(Dissimilar)	0 (Similar)
2	0 (Similar)	0 (Similar)
3	0 (Similar)	1 (Dissimilar)
4	1 (Dissimilar)	1 (Dissimilar)

Table 4.5: Test Case Scenarios

Characteristics of Certificates:

- Certificate issuer: All the top-level Certificate Authority issuer such as "GTS CA 1O1", "Go Daddy Secure Certificate Authority - G2", "Sectigo Secure Server CA" were observed for legitimate domains were considered for phishing domains whereas "Let's Encrypt Authority X3", "CloudFlare IncECC CA-2","cPanel, Inc. Certification Authority" which were common for phishing domains were taken for legitimate domains.

- Types of certificate: Domain validated and Wildcard certificate types were found commonly for phishing domains were considered for legitimate domains whereas Organization and Extended validated certificate types were observed for legitimate domains were considered for phishing domains.

- If top-level domain matches, I verified that the certificate issuer should not be matched with training data for both legitimate and phishing domains. For example, "cpanel.net" was legitimate domain and issuer name was "COMODO RSA Domain Validation Secure Server CA" for "cpanel.net" certificate. Here, top level domain ".net" was present in training data but also verified that "COMODO RSA Domain Validation Secure Server CA" was not issuer of any other legitimate domain's certificate.

Case 1: Table 4.6 shows result of different characteristics for legitimate data samples and similar characteristics for phishing data samples.

Case 2: Table 4.7 shows result of similar characteristics for both legitimate and phishing data.

Case 3: Table 4.8 shows result of similar characteristics for legitimate data samples and different characteristics for phishing data samples. Here, the same legitimate data samples were taken that used in case 2.

Decision Tree Classifier	Accuracy	Legitimate					Phishing				
		TPR	FNR	FPR	TNR	F1	TPR	FNR	FPR	TNR	F1
With Text-feature	81.9 %	72.2%	27.8%	8.3%	91.7%	80.1%	91.7%	8.3%	27.8%	72.2%	83.5%
Without Text-feature	86.1%	83.3%	16.7%	11.1%	88.9%	85.7%	88.9%	11.1%	16.7%	83.3%	86.5%

Table 4.6: Test Result of Case 1

Decision Tree Classifier	Accuracy	Legitimate					Phishing				
		TPR	FNR	FPR	TNR	F1	TPR	FNR	FPR	TNR	F1
With Text-feature	91.6%	91.7%	8.3%	8.3%	91.7%	91.7%	91.7%	8.3%	8.3%	91.7%	91.7%
Without Text-feature	87.7%	86.1%	13.9%	11.1%	88.9%	87.2%	88.9%	11.1%	13.9%	86.1%	87.7%

Table 4.7: Test Result of Case 2

Case 4: Table 4.9 shows result of different characteristics for both legitimate and phishing data. Here, the same legitimate data samples were taken that used in case 1 and the same phishing data samples were taken that used in case 3.

After analyzing all the test cases results, I found that case 2 and case 3 were given the same level of variations as training results. But there was a huge difference in the performance in case 1 and case 4. The scores of With text-feature decision tree classifier was lower in case 4 against Without text-feature decision tree for both categories. Although the case 2 achieved good scores of 91.7% in both categories for With text feature and 87% for Without text feature which matched the hypothesis of the training data. From the results, the observation was that the cases having dissimilar characteristics with training data performed well for Without text feature. Even in case 1, the score for legitimate category was quite good and in case 3, the score of the phishing category was above 85% where the model specific assumptions were changed. Therefore, Without text-feature performed better than With Text-feature in all the uncertain scenarios.

While testing both Web API, I found that there were many legitimate domains that were classified as phishing because these domains were sub-domains, in-spite of having good Alexa rank i.e. ranks were lower than 1000. This point was contradicting as per mentioned in analyzed characteristics, the sub-domains were observed in phishing domains. So, the text features such as Euclidean distances where CNs

Decision Tree Classifier	Accuracy	Legitimate					Phishing				
		TPR	FNR	FPR	TNR	F1	TPR	FNR	FPR	TNR	F1
With Text-feature	83.3%	88.9%	11.1%	22.2%	77.8%	88.9%	77.8%	22.2%	11.1%	88.9%	77.8%
Without Text-feature	86.1%	86.1%	13.9%	13.9%	86.1%	86.1%	86.1%	13.9%	13.9%	86.1%	86.1%

Table 4.8: Test Result of Case 3

Decision Tree Classifier	Accuracy	Legitimate					Phishing				
		TPR	FNR	FPR	TNR	F1	TPR	FNR	FPR	TNR	F1
With Text-feature	75%	72.2%	27.8%	22.2%	77.8%	74.3%	77.8%	22.2%	27.8%	72.2%	77.8%
Without Text-feature	84%	83.3%	16.7%	13.9%	86.1%	84%	86.1%	13.9%	16.7%	83.3%	84%

Table 4.9: Test Result of Case 4

Classifier	Accuracy	Legitimate			Phishing		
J48		TPR	FPR	F1	TPR	FPR	F1
With Text-features	90.62 %	95.3%	13.8%	91.0%	86.3%	5%	90.2%
Without Text-features	84.37%	83.8%	15%	83.5%	85%	16%	84.5%

Table 4.10: Classification results of new decision trees

strings computed the distance were predicting as wrong and classified in the phishing category because the length was long. To solve this issue, 20 sub-domains legitimate and 20 phishing domains were added in the final dataset and trained again using Weka. The reason of retraining was to remove wrong perception of test scenarios where the similar characteristics were taken for both training and testing data samples. Table 4.10 shows the new classification result for both decision tree systems. As per the new classification result of the decision trees, the accuracy for both decision trees changed as compared to previous classification results. Also, there was a change in the decision trees structure. The 11 features involved in text-features decision tree where Without text-features, 12 features were participated in decision tree. Later, I added 10 more domains in both legitimate and phishing domains of final training dataset to check for new changes. Similar results and tree structure were also found for both the decision trees classifiers with added 10 domains.

Case 1: Table 4.11 shows new result of different characteristics for legitimate data samples and similar characteristics for phishing data samples.

Case 2: Table 4.12 shows new result of similar characteristics for both legitimate samples and phishing data samples.

Case 3: Table 4.13 shows new result of similar characteristics for legitimate data

Decision Tree Classifier	Accuracy	Legitimate					Phishing				
		TPR	FNR	FPR	TNR	F1	TPR	FNR	FPR	TNR	F1
With Text-feature	79.5 %	73.8%	26.2%	11.9%	88.1%	79.0%	88.1%	11.9%	26.2%	73.8%	82.2%
Without Text-feature	72%	73.8%	26.2%	28.6%	71.4%	72.9%	71.4%	28.6%	26.2%	73.8%	72.3%

Table 4.11: New Test Result of Case 1

Decision Tree Classifier	Accuracy	Legitimate					Phishing				
		TPR	FNR	FPR	TNR	F1	TPR	FNR	FPR	TNR	F1
With Text-feature	85.7%	83.3%	16.7%	11.9%	88.1%	85.4%	88.1%	11.9%	16.7%	83.3%	86%
Without Text-feature	75%	78.6%	21.4%	28.6%	71.4%	75.9%	71.4%	28.6%	21.4%	78.6%	74.1%

Table 4.12: New Test Result of Case 2

Decision Tree Classifier	Accuracy	Legitimate					Phishing				
		TPR	FNR	FPR	TNR	F1	TPR	FNR	FPR	TNR	F1
With Text-feature	67%	88.3%	11.1%	50.0%	50%	71.4%	50%	50%	16.7%	88.3%	60.0%
Without Text-feature	72.6%	78.6%	21.4%	33.3%	66.7%	74.2%	66.7%	33.3%	21.4%	78.6%	70.9%

Table 4.13: New Test Result of Case 3

samples and different characteristics for phishing data samples. Again, the same legitimate data samples were taken that used in case 2.

Case 4: Table 4.14 shows new result of different characteristics for both legitimate and phishing data. Again, the same legitimate data samples were taken that used in case 1 and the same phishing data samples were taken that used in case 3.

To test the uncertainties of new classification results, a new characteristic was added in domains' characteristics with other four to induce similarities between testing and training data sample.

- Sub-domain with good Alexa ranking in legitimate data whereas sub-domain with poor Alexa ranking in phishing data.

For ensuring differences,

- Sub-domain with good Alexa ranking in phishing data whereas sub-domain with poor Alexa ranking in legitimate data.

The reason for considering this characteristics was because Alexa provided same ranking to full-domain and sub-domain. For example, Alexa ranking of "www.google.com" and "developers.google.com" was 1. As "developers.google.com" was sub-domain of "www.google.com". I found that the Alexa ranking of sub-domains in phishing domains was more than 100000 or None. Along with domain characteristics, certificates

Decision Tree Classifier	Accuracy	Legitimate					Phishing				
		TPR	FNR	FPR	TNR	F1	TPR	FNR	FPR	TNR	F1
With Text-feature	64%	76.2%	23.8%	50%	50%	67.4%	50%	50%	23.8%	76.2%	57.5%
Without Text-feature	70%	71.4%	28.6%	33.3%	66.7%	70%	66.7%	33.3%	28.6%	71.4%	68.3%

Table 4.14: New Test Result of Case 4

characteristics which was mentioned earlier were also considered to evaluate the robustness of new classification results. Same as the previous test-data, I again prepared four sets of test cases same as Table 4.5 for new decision tress and 20 more domains (10 phishing and 10 legitimate) were added in each test case, now each test case had 92 data samples. Following are new test results for each case. The new results had many variations in contrast to the previous results. In case 1, With text-feature was given average performance with 79% of F1 score in legitimate category but in case 3, it scored only 60% in phishing category where assumptions were replaced. Although in case 2, both the decision trees performed well, With text-feature decision tree the F1 score was 85.4% for the legitimate category and 86% for the phishing category whereas F1 score of 75.9% for the legitimate category and 74.1% for the phishing category by Without text-feature decision trees. In case 4, Without text-feature, decision trees provided better performance than the With text-feature. After analyzing these new results, the observation was similar to previous results, the testing results of the Without text-feature decision tree model showed good scores in the categories where the dissimilar behaviours of training and testing data were considered. Hence, the new results also supported that Without text-feature decision tree model was insensitive to changes in base model specifications.

After testing classification models with new test data samples, both Web API's decision maker function updated with new decision rules. Thereafter, the Web API's tested with new training data samples to check its response when sub-domain taken as an input request.

4.4 Chapter Summary

In this chapter, all the experiments along with their results were discussed in detail. Here, I have highlighted some key points to summarize this chapter:

- The final dataset was generated by analyzing the performance and decision tree structure of J48 decision tree classifier.

- After applying feature selection methods on attributes on the final dataset, out of 42 features, 31 features were chosen for the Web API detection system.

- Among all classifiers, J48 decision tree classification model was chosen to design the detection system.
- J48 decision tree classifier was trained using two feature sets: With text-features and Without text-features.
- With text-feature classification model achieved higher scores than Without text-features classification model in training the J48 decision tree classifier.
- But Without text-features classification model performed well in almost all the test case scenarios where different assumptions were taken.
- Both trained and retrained testing results showed that Without text-features classification model was more robust than With text-features classification model. Hence, Web API with Without text-features decision tree rules was implemented as the proposed system in this thesis.

Chapter 5

Conclusion

5.1 Summary of Research

In this research, a robust machine learning classifier with different features was explored to identify legitimate and phishing domains. The identification was based on the features of SSL certificates generated for registered domains. By use of Python scripts, the connections were made with more than 100 domains on TCP port 443 and their SSL certificates were retrieved along with creating 42 features-boolean, integer and text features. Two scripts (certificate extraction and feature creation) were used for collecting training and testing datasets. The different characteristics of domains and certificates for both legitimate and phishing classes were analyzed with collected data samples. The final dataset was formed with J48 decision tree after observing the behaviour and performance of the decision tree. It was analyzed that the performance of the trained model increased as the number of data samples increased with different characteristics. In addition, by using feature selection methods, the training speed and performance of the models were improved after removing the unnecessary features. Out of 42, only 31 features were considered to prepare the final training dataset. After selecting the suitable machine learning classifier based on the performance results of the ten-fold cross validation, the performance of the selected model was examined with different sets of features. One set consists of text-features and the other consists of without text features. The robustness of these decision tree classification models were analyzed by four sets of test cases where similar and dissimilar characteristics of (compared to the training) of data were included. The results of each case provided the performance difference of With text features and Without text features. The obtained results for Without text features decision model performed well in almost every case for both classes as compared to With text features. Thus, Without text features decision model was proposed to classify legitimate and phishing domains.

As per my knowledge, this is first effort towards analyzing the classifier robustness of the trained models with different features. With these evaluations, it was easy to estimate the effectiveness and consistency of the classifiers if the attackers changed their patterns in phishing. With decision tree classifier, the hidden patterns and indicators were recognized in the SSL certificate. The decision tree classifier made it easier for me to interpret the decision rules because of its human readable rules and visualization property. By comparing and analyzing the differences in each test case performances of both decision trees, the robust classifier was explored with respective features. As the dataset had only 160 data samples (80 legitimate and 80 phishing domains), these results could be taken as preliminary but promising results in detecting the phishing domains.

5.2 Key Findings

Finally, on the basis of analyzed domains and certificates patterns between legitimate and phishing classes, I observe the following:

- Attackers use free certificates issued by CA such as "Let's Encrypt Authority X3", "cPanel, Inc. Certification Authority". However, these free web certificates are available mainly for start-up companies who can not afford expensive SSL certificate but attackers use these methods for phishing purposes.

- Attackers use domain validated and wild card certificates. They do not seem to purchase organization or extended validated certificates as it may disclose their actual purposes and personal details. CA strictly verifies provided information before issuing organization or extended validated certificates.

- Attackers seem to use SSL certificates that are valid for only 90 days. These provide enough time to fool internet users by showing encrypted traffic.

- Attackers tend to use the fake domains with self-signed certificates where they issue the SSL certificate to themselves by employing free SSL certificate tools.

Moreover, the Web API and proof-of-concept web application that I developed in this research provides a user interface to detect between legitimate and phishing domains over HTTPS protocol. Since the implementation of decision rules as conditional

control statements was simple, the process was automated through the proposed framework where SSL certificate was extracted from the input domain and classified as phishing or legitimate using Without text-feature decision tree rules. These were coded in Python programming language. The properties of the proposed Web API are given below-

- Web API categorizes the input domain within seconds.
- Web API is extremely easy to use and flexible.

5.3 Future Work

The possible future research directions are as the following:

- **Addition of more data samples in training dataset**: Since only 160 data samples were used as the training dataset in this research, for future studies, researchers can increase data size by adding more data samples in legitimate and phishing categories. More data will encourage them to learn different domain and certificate characteristics in both categories.
- **Improvement in Web API designing and functionality with users' perspective**: With user studies, researchers can learn possible ways to make more interactive design and functionality for the Web API. This will help in increasing effectiveness and proper utilization of the Web API.
- **Browser plugin implementation**: In this thesis, the implemented proof-of-concept web application allows users to do domain verification. In future, Web API's functionalities can also be implemented as a browser plugin.

Bibliography

[1] Alexa rank different from moz domain authority. https://reliableacorn.com/web-metrics/alexarankvsmozdomainauthority/.

[2] Cryptographic library. http://openssl.cs.utah.edu/docs/crypto/crypto.html.

[3] Digital certificates explained - knowledge base. https://sites.google.com/site/amitsciscozone/home/security/digital-certificates-explained.

[4] Publicsuffix2. https://pypi.org/project/publicsuffix2/.

[5] Seolib. https://pypi.org/project/seolib/.

[6] Ssl - python wiki. https://wiki.python.org/moin/SSL.

[7] There's just no consensus on the size of ad fraud right now. https://www.emarketer.com/content/the-size-of-the-ad-fraud-problem-in-digital-marketing-is-varying.

[8] Vulnerabilities, exploits, and threats: Definitions and examples. https://www.rapid7.com/fundamentals/vulnerabilities-exploits-threats/.

[9] Decisiontrees, Nov 2011. https://octaviansima.wordpress.com/2011/03/25/decision-trees-c4-5/.

[10] A complete guide to k-nearest-neighbors, Jul 2016. https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/.

[11] How the naive bayes classifier works in machine learning, Feb 2017. https://dataaspirant.com/2017/02/06/naive-bayes-classifier-machine-learning/.

[12] Majority of the world's top million websites now use https, Sep 2018. https://www.welivesecurity.com/2018/09/03/majority-worlds-top-websites-https/.

[13] Explain what flask is and its benefits?, Oct 2019. https://www.i2tutorials.com/technology/explainwhatflaskisanditsbenefits/.

[14] Verisign q1 2019 domain name industry brief: Million domain name registrations in the first quarter of 2019, May 2019. https://blog.verisign.com/domain-names/verisign-q1-2019-domain-name-industry-brief-internet-grows-to-351-8-million-domain-name-registrations-in-the-first-quarter-of-2019/.

[15] Adaboost, Jan 2020. https://en.wikipedia.org/wiki/AdaBoost.

[16] Bagging decision tree in ensemble models?, Mar 2020. https://sebastianraschka.com/faq/docs/bagging-boosting-rf.html.

[17] Long short-term memory, Mar 2020. https://en.wikipedia.org/wiki/Long_short-term_memory.

[18] Phishing, Mar 2020. https://en.wikipedia.org/wiki/Phishing.

[19] Scikit-learn, Jan 2020. https://en.wikipedia.org/wiki/Scikit-learn.

[20] Scipy, Feb 2020. https://en.wikipedia.org/wiki/SciPy.

[21] Wildcard certificate, Feb 2020. https://en.wikipedia.org/wiki/Wildcard_certificate.

[22] Alex.Arnaut. Digital signature, Aug 2019. https://www.docusign.com/how-it-works/electronic-signature/digital-signature/digital-signature-faqpki.

[23] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

[24] Mohamed Alsharnouby, Furkan Alaca, and Sonia Chiasson. Why phishing still works: User strategies for combating phishing attacks. *International Journal of Human-Computer Studies*, 82:69–82, 2015.

[25] Taiwo Oladipupo Ayodele. Types of machine learning algorithms. *New advances in machine learning*, pages 19–48, 2010.

[26] David Bisson. Wildcard certificates make encryption easier, but less secure. https://www.venafi.com/blog/wildcard-certificates-make-encryption-easier-but-less-secure.

[27] Adi Bronshtein. A quick introduction to k-nearest neighbors algorithm, May 2019. https://blog.usejournal.com/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7.

[28] Jason Brownlee. How to use ensemble machine learning algorithms in weka, Aug 2019. https://machinelearningmastery.com/use-ensemble-machine-learning-algorithms-weka/.

[29] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[30] Thawatchai Chomsiri. Https hacking protection. In *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, volume 1, pages 590–594. IEEE, 2007.

[31] M A Descalle. Quarterly progress report q4 fy2019. 2019. https://docs.apwg.org/reports/apwg_trends_report_q4_2019.pdf.

[32] Thomas G Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 40(2):139–157, 2000.

[33] Zheng Dong, Apu Kapadia, Jim Blythe, and L Jean Camp. Beyond the lock icon: real-time detection of phishing websites using public key certificates. In *2015 APWG Symposium on Electronic Crime Research (eCrime)*, pages 1–12. IEEE, 2015.

[34] Pallavi D Dudhe and PL Ramteke. A review on phishing detection approaches. *Journal of Computer Science and Mobile Computing*, 4(2):166–170, 2015.

[35] Rohith Gandhi. Naive bayes classifier, May 2018. https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c.

[36] Esperanza García-Gonzalo, Zulima Fernández-Muñiz, Paulino José García Nieto, Antonio Bernardo Sánchez, and Marta Menéndez Fernández. Hard-rock stability analysis for span design in entry-type excavations with learning classifiers. *Materials*, 9(7):531, 2016.

[37] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[38] Eva Hanscom. Shadow brokers: Are insider threats hiding on your network? https://www.venafi.com/blog/shadow-brokers-and-beyond-what-insider-threats-are-hiding-your-network.

[39] Jiwon Hong, Taeri Kim, Jing Liu, Noseong Park, and Sang-Wook Kim. Phishing url detection with lexical features and blacklisted domains. In *Adaptive Autonomous Secure Cyber Systems*, pages 253–267. Springer, 2020.

[40] Badr Hssina, Abdelkarim Merbouha, Hanane Ezzikouri, and Mohammed Erritali. A comparative study of decision tree id3 and c4. 5. *International Journal of Advanced Computer Science and Applications*, 4(2):13–19, 2014.

[41] Ankit Kumar Jain and Brij B Gupta. Phishing detection: analysis of visual similarity based approaches. *Security and Communication Networks*, 2017, 2017.

[42] Russell Kay. Sidebar: The origins of phishing, Jan 2004. https://www.computerworld.com/article/2575094/sidebar–the-origins-of-phishing.html.

[43] KnowBe4. History of phishing. https://www.phishing.org/history-of-phishing.

[44] Will Koehrsen. A feature selection tool for machine learning in python, Jun 2018. https://towardsdatascience.com/a-feature-selection-tool-for-machine-learning-in-python-b64dd23710f0.

[45] Dennis Korotyaev, Oct 2014. https://www.quora.com/What-are-the-differences-between-ID3-C4-5-and-CART.

[46] Steven Lang, Felipe Bravo-Marquez, Christopher Beckham, Mark Hall, and Eibe Frank. Wekadeeplearning4j: A deep learning package for weka based on deeplearning4j. *Knowledge-Based Systems*, 178:48 – 50, 2019.

[47] Lindsay Liedke. 100 internet statistics facts for 2020, Jan 2020. https://www.websitehostingrating.com/internet-statistics-facts/.

[48] Yanli Liu, Yourong Wang, and Jian Zhang. New machine learning algorithm: Random forest. In *International Conference on Information Computing and Applications*, pages 246–252. Springer, 2012.

[49] Samuel Marchal, Jérôme François, Radu State, and Thomas Engel. Phishstorm: Detecting phishing with streaming analytics. *IEEE Transactions on Network and Service Management*, 11(4):458–471, 2014.

[50] John McGahagan, Darshan Bhansali, Ciro Pinto-Coelho, and Michel Cukier. A comprehensive evaluation of webpage content features for detecting malicious websites. In *2019 9th Latin-American Symposium on Dependable Computing (LADC)*, pages 1–10. IEEE, 2019.

[51] Ulrike Meyer and Vincent Drury. Certified phishing: Taking a look at public key certificates of phishing websites. In *Fifteenth Symposium on Usable Privacy and Security ({SOUPS} 2019)*, 2019.

[52] Migrator. Advantages and disadvantages of digital marketing, Aug 2019. https://www.nibusinessinfo.co.uk/content/advantages-and-disadvantages-digital-marketing.

[53] Mishari Al Mishari, Emiliano De Cristofaro, Karim El Defrawy, and Gene Tsudik. Harvesting ssl certificate data to identify web-fraud. *arXiv preprint arXiv:0909.3688*, 2009.

[54] Pravesh Moelchand, Kabilan Gnanavarothayan, Jim Verheijde, and Just van Stam. Real time threat detection through network analysis. 2019.

[55] Arun Mohan. Recurrent neural network and long term dependencies, Feb 2020. https://www.infolks.info/blog/recurrent-neural-network/.

[56] Eric Neumayer and Thomas Plümper. *Robustness tests for quantitative research*. Cambridge University Press, 2017.

[57] Paritosh Pantola. Natural language processing: Text data vectorization, Jun 2018. https://medium.com/@paritosh_30025/naturallanguageprocessing-textdatavectorizationaf2520529cf7.

[58] Routhu Srinivasa Rao, Tatti Vaishnavi, and Alwyn Roshan Pais. Catchphish: detection of phishing websites by inspecting urls. *Journal of Ambient Intelligence and Humanized Computing*, 11(2):813–825, 2020.

[59] Angelo PE Rosiello, Engin Kirda, Fabrizio Ferrandi, et al. A layout-similarity-based approach for detecting phishing pages. In *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops-SecureComm 2007*, pages 454–463. IEEE, 2007.

[60] Margaret Rouse. What is digital certificate? - definition from whatis.com, Aug 2018. https://searchsecurity.techtarget.com/definition/digital-certificate.

[61] James Sanders. How fraudulent domain names are powering phishing attacks, Jun 2019. https://www.techrepublic.com/article/how-fraudulent-domain-names-are-powering-phishing-attacks/.

[62] Stu Sjouwerman. Phishing attack use of encryption increases. https://blog.knowbe4.com/phishing-attack-use-of-encryption-increases-400-for-malware-delivery-communications-and-data-exfiltration.

[63] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[64] Ivan Torroledo, Luis David Camacho, and Alejandro Correa Bahnsen. Hunting malicious tls certificates with deep neural networks. In *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*, pages 64–73, 2018.

[65] Joaquin Vanschoren. 10-fold crossvalidation. https://www.openml.org/a/estimation-procedures/1.

[66] Tara Whalen and Kori M Inkpen. Gathering evidence: use of visual security cues in web browsers. In *Proceedings of Graphics Interface 2005*, pages 137–144. Canadian Human-Computer Communications Society, 2005.

[67] Xiaofeng Yuan, Lin Li, and Yalin Wang. Nonlinear dynamic soft sensor modeling with supervised long short-term memory network. *IEEE Transactions on Industrial Informatics*, 2019.

[68] Haijun Zhang, Gang Liu, Tommy WS Chow, and Wenyin Liu. Textual and visual content-based anti-phishing: a bayesian approach. *IEEE Transactions on Neural Networks*, 22(10):1532–1546, 2011.

[69] Shichao Zhang, Xuelong Li, Ming Zong, Xiaofeng Zhu, and Ruili Wang. Efficient knn classification with different numbers of nearest neighbors. *IEEE transactions on neural networks and learning systems*, 29(5):1774–1785, 2017.

[70] Zixuan Zhang. Understand data normalization in machine learning, Aug 2019. https://towardsdatascience.com/understand-data-normalization-in-machine-learning-8ff3062101f0.

Appendix A

Decision Tree Classifiers with Feature Sets

In this chapter, decision tree classifier with different set of features are demonstrated that includes both old and new decision tree structures and involved features.

A.1 Old Decision Tree Classifier With Text Features

Decision Tree classifier With text features involved three category of features:Boolean, Text, Integer. The tree structure of this decision tree classifier is shown in Figure A.1. The involved features definition are presented below:

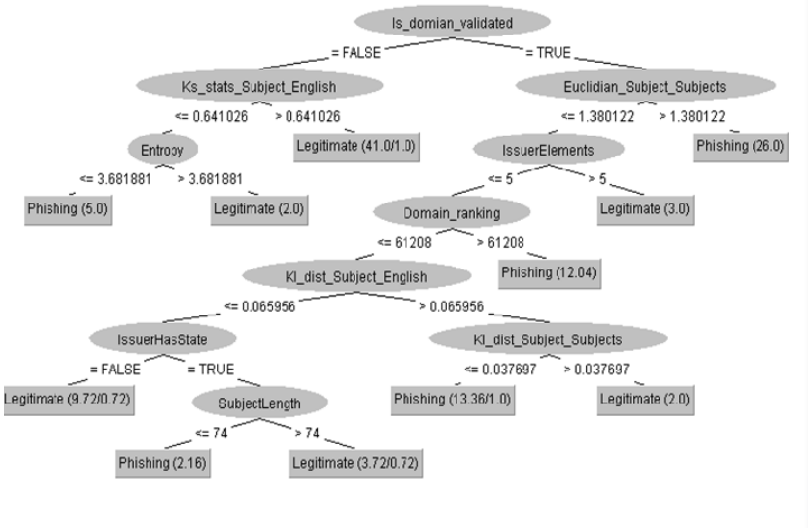

Figure A.1: Old decision tree structure With text features

- Is_domian_validated(Boolean feature): It checks whether the certificate is domain validated or not. In any domain validated certificate, the subject name must not have a business category and organization name fields.
- IssuerHasState(Boolean feature): It only checks whether the issuer component had state field or not in the certificate

- IssuerElements(Integer feature): It counts a number of details present in the Issuer component.

- SubjectLength(Integer feature): It counts a number of characters of the whole subject string.

- Domain_Ranking(Integer feature): It provides a rank to a domain name by estimating traffic of the domain.

- Entropy(Text feature): It calculates Shannon entropy of subject name of CN component. Shannon entropy is the minimal number of bits per symbol required to encode the string. First, calculated the frequency of alphabet in CN component and then found minimum number of bits required to encode each symbol.

- Euclidian_Subject_Subjects(Text feature): It calculates euclidean distance between two different certificate subjects. Each certificate subject has its own CN field. The feature produces the distance between two CN's. I computed feature vectors from the CN's using Count Vectorization feature extraction method. Because the Alexa Traffic Rank of "google.com" is 1, I kept "google.com" certificate as constant parameter for all the incoming certificates. Hence, the considered two parameters would be "google.com" and an incoming certificate.

- Ks_stats_Subject_English(Text feature): It calculates Kolmogorov-Smirnov statistic between CN and English characters(both upper and lower case characters were considered). Kolmogorov-Smirnov statistic determines if two samples are drawn from the same continuous distribution. ks_2samp library function was imported to get the result. Same as previous feature, I initialized document for two samples where one of the sample is incoming subject CN and other would be English characters. I computed feature vectors from both samples using count vectorization feature extraction method. Based on the obtained feature vectors, I calculated the statistics. If the statistic is small, then the hypothesis is that the distributions of the two samples are same.

- Kl_dist_Subject_Subject(Text feature): It calculates Kullback-Leiber distance between two different certificate subjects. Kullback-Leiber distance also known

as the mutual information that provides common information shared between two labels with same type of data. Same as Euclidian_Subject_Subjects, I processed rest and used sklearn.metrics.mutual_info_score library function to calculate distances.

- Kl_dist_Subject_English(Text feature): It calculates Kullback-Leiber Distance between CN and English characters (both upper and lower case characters were considered). Same as Kolmogorov-Smirnov statistic, I processed rest and used sklearn.metrics.mutual_info_score library function to calculate distances.

A.2 Old Decision Tree Classifier Without Text Features

Decision Tree classifier Without text features involved two category of features- Boolean and Integer. The tree structure of this decision tree classifier is shown in Figure A.2. The involved features definition are presented below:

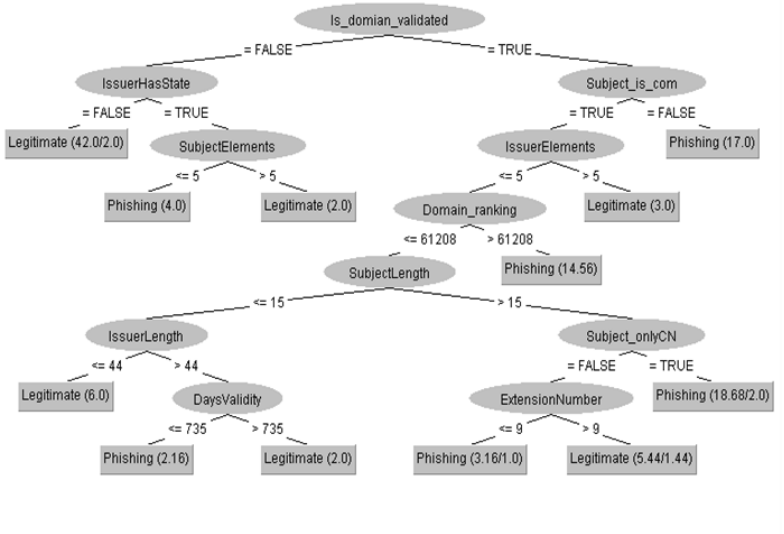

Figure A.2: Old decision tree structure Without text features

- Is_domian_validated(Boolean feature: It checks whether the certificate is domain validated or not.

- IssuerHasState(Boolean feature): It only checks whether the issuer component has state field or not in the certificate.

- IssuerElements(Integer feature): It counts number of details present in the issuer component of certificate.
- SubjectLength(Integer feature): It counts number of characters of the whole subject string of certificate.
- Domain_ranking(Integer feature): It provides ranking to the domain name by estimating traffic of the domain.
- DaysValidity(Integer feature): It calculates difference between Not Before and Not After fields of Validity in the certificate.
- SubjectElements(Integer feature): It counts number of details present in the subject component.
- Extension number(Integer feature): It counts a number of extensions contained in the certificate.
- Subject_onlyCN(Boolean feature): It checks whether a subject component has only CN field.
- Subject_is_com(Boolean feature): It checks whether subject CN field is a '.com' doman.
- IssuerLength(Integer feature): It counts number of characters of the whole issuer string in certificate.

A.3 New Decision Tree Classifier With Text Features

New decision Tree classifier With text features also involved three category of features- Boolean, Text, Integer. The tree structure of this decision tree classifier is shown in Figure A.3. There were some common features in both new decision tree structure and old decision tree structure of With text feature decision tree classifiers. I have defined below only new features and rest repeated features are already explained in Old Decision Tree Classifier With Text Feature section.

- Extension number(Integer feature): It counts a number of extensions contained in the certificate.

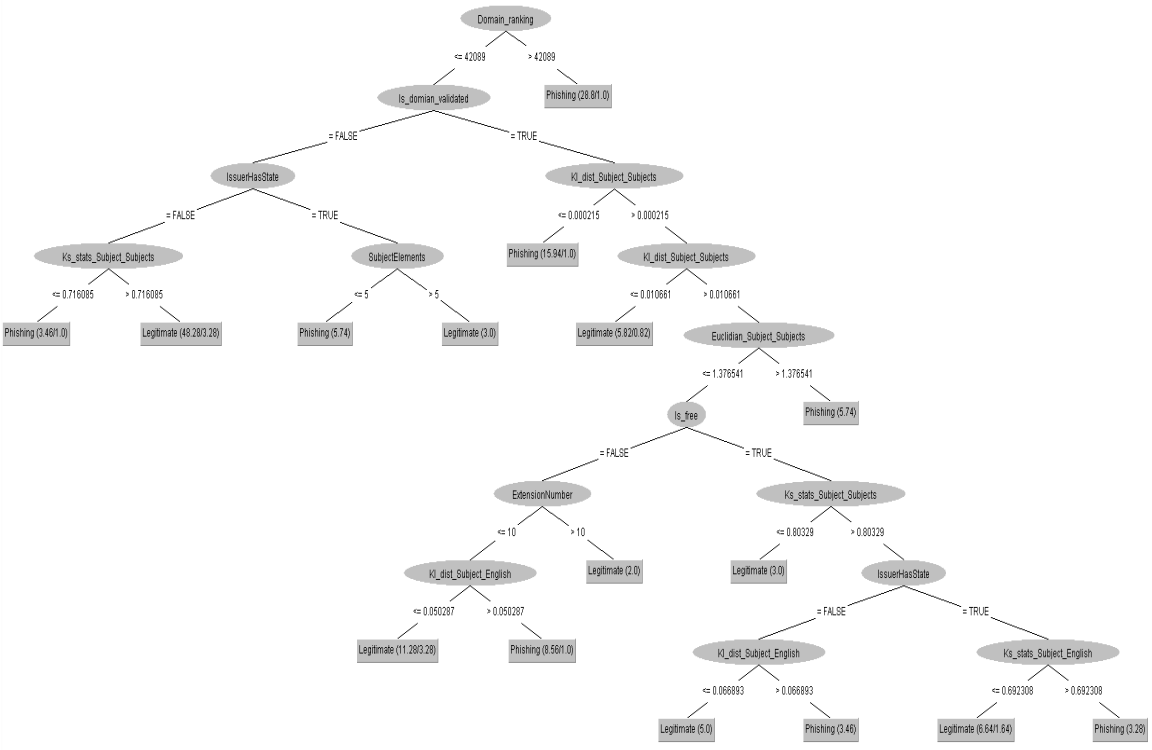

Figure A.3: New decision tree structure Without text features

- Is_free(Boolean feature): It indicates if the certificate is freely generated. For any free certificate, the type should be domain validated and have only 90 days of validity.

- Ks_stats_Subject_Subject(Text feature): It calculates Kolmogorov-Smirnov statistic between incoming certificate and google' certificate since google's certificate was kept as one constant parameter for calculating all the distances in this study.

A.4 New Decision Tree Classifier Without Text Features

New decision Tree classifier Without text features also involved two category of features-Boolean, Integer. The tree structure of this decision tree classifier is shown in Figure A.4 Same as previous section, there were addition of new features in this new decision tree structure of Without text feature decision tree classifiers. I have presented below only new features and rest repeated features are already explained

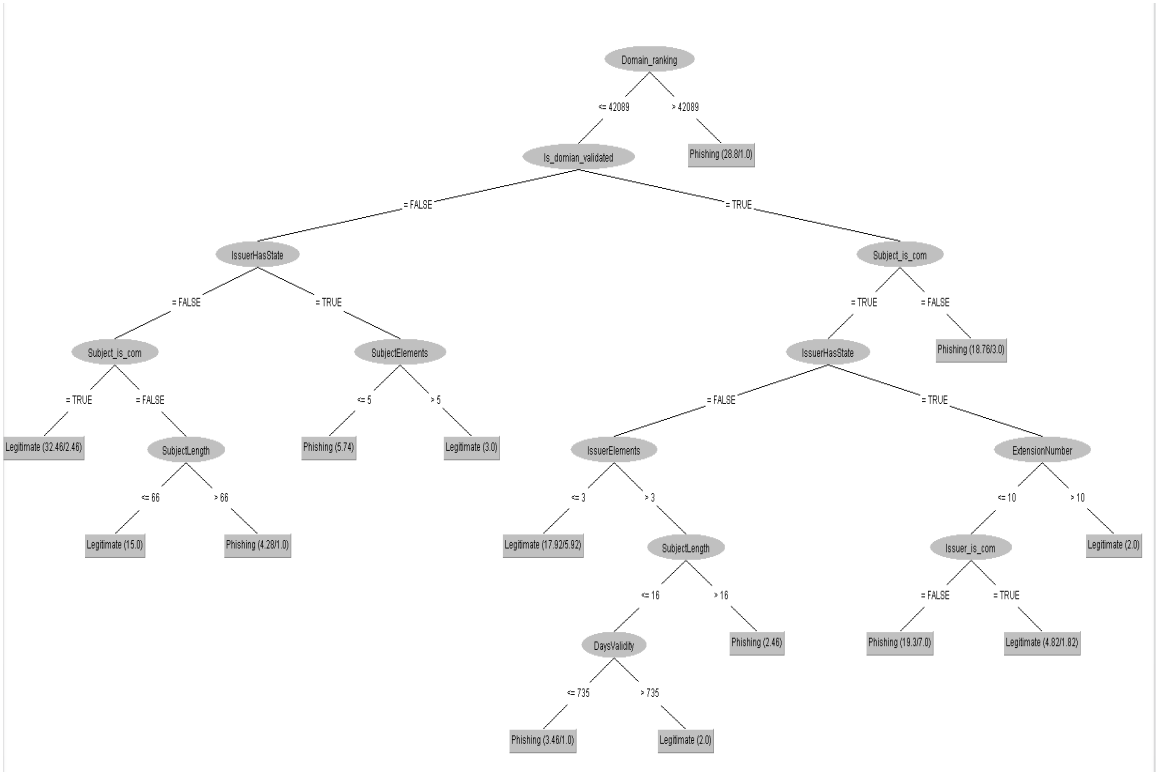

Figure A.4: New decision tree structure Without text features

in Old Decision Tree Classifier Without Text Feature section.

- Issuer_is_com(Boolean feature): It checks whether issuer CN field is a '.com' domain.
- IssuerElements(Integer feature): It counts number of details present in the issuer component.

Appendix B

Test Cases

In this chapter, the domains used in test cases are presented. The total four test case scenarios were created that is showed in Table 4.5. In these test cases, similar and dissimilar characteristics of domains and certificates were included for testing the robustness of With text features and Without text-features classification models. I have detailed the similar characteristics in chapter 3 in data section and dissimilar characteristics in chapter 4 in model uncertainty section.

Figure B.1 shows lists of domains used in different test cases. The description of cases is presented below:

Case 1: Domains of different characteristics for legitimate data and similar characteristics for phishing data .

Case 2: Domains of similar characteristics for both legitimate data and phishing data.

Case 3: Domains of similar characteristics for legitimate data and different characteristics for phishing data. Here, the same legitimate data are taken that used in case 2.

Case 4: Domains of different characteristics for both legitimate and phishing data. Here, the same legitimate data are taken that used in case 1 and the same phishing data are taken that used in case 3.

	Domain	Case 1	Case 2	Case 3	Case 4
2	www.mozilla.org		✓	✓	
3	www.linkedin.com		✓	✓	
4	vimeo.com		✓	✓	
5	www.cloudflare.com		✓	✓	
6	www.bbc.com		✓	✓	
7	www.istockphoto.com		✓	✓	
8	www.dailymotion.com		✓	✓	
9	medium.com		✓	✓	
10	www.cnn.com		✓	✓	
11	www.nytimes.com		✓	✓	
12	www.slideshare.net		✓	✓	
13	www.forbes.com		✓	✓	
14	www.imdb.com		✓	✓	
15	www.reuters.com		✓	✓	
16	www.nih.gov		✓	✓	
17	www.whatsapp.com		✓	✓	
18	www.dropbox.com		✓	✓	
19	www.uol.com.br		✓	✓	
20	www.washingtonpost.com		✓	✓	
21	www.pinterest.ca		✓	✓	
22	www.oracle.com		✓	✓	
23	www.lefigaro.fr		✓	✓	
24	rt.com		✓	✓	
25	www.usatoday.com		✓	✓	
26	archive.org		✓	✓	
27	www.samsung.com		✓	✓	
28	www.ig.com.br		✓	✓	
29	www.fandom.com		✓	✓	
30	issuu.com		✓	✓	
31	www.aliexpress.com		✓	✓	
32	www.jimdo.com		✓	✓	
33	www.ft.com		✓	✓	
34	www.rakuten.co.jp		✓	✓	
35	www.change.org		✓	✓	
36	www.wired.com		✓	✓	
37	nasa.gov		✓	✓	
38	newgetrecoveri.000webhostapp.com	✓	✓		
39	app.exemetrics.com	✓	✓		
40	zastudio.ca	✓	✓		
41	www.promodamagazine.com	✓	✓		
42	pamelakurier.com	✓	✓		
43	kiandkaa.com	✓	✓		
44	qpayonline.live	✓	✓		
45	on-potsdam.de	✓	✓		
46	www.promocaodamagazine.com	✓	✓		
47	solicitar.personalitecadastro.com	✓	✓		
48	info-credem.ca	✓	✓		
49	secure.runescape.com-gg.ru	✓	✓		
50	4k-magazineluiza.ddns.net	✓	✓		
51	timeoutlastoff.com	✓	✓		
52	bcplzonasegurabeta.viaqbcp.com	✓	✓		
53	amazon.co.jp.15cd3e49653687574bf2668832cf384a.xyz	✓	✓		

54	portalsemprepresenteprogama.com	✓	✓		
55	souktabule.com	✓	✓		
56	bholabhaichowk.com	✓	✓		
57	staging.hospitalitychain.com	✓	✓		
58	drphilhaggerty.com	✓	✓		
59	santos-schulz.com	✓	✓		
60	felkart.in	✓	✓		
61	myanalytics.com.my	✓	✓		
62	atualizacao-app-caixa-gov.com	✓	✓		
63	solicitacaoneucard-com.umbler.net	✓	✓		
64	hhe.net.ua	✓	✓		
65	bsemep.com	✓	✓		
66	www.albanopoulou.gr	✓	✓		
67	itaclass-para-voce.000webhostapp.com	✓	✓		
68	peleken.com	✓	✓		
69	educ.rec.br	✓	✓		
70	outlookowa23.wixsite.com	✓	✓		
71	suportebanco.com	✓	✓		
72	telebang.com	✓	✓		
73	travelpura.com	✓	✓		
74	wordpress.org	✓			✓
75	www.theguardian.com	✓			✓
76	www.linkedin.com	✓			✓
77	www.bestbuy.ca	✓			✓
78	www.lemonde.fr	✓			✓
79	line.me	✓			✓
80	www.skyrock.com	✓			✓
81	primevideo.com	✓			✓
82	www.globo.com	✓			✓

83	www.hugedomains.com	✓			✓
84	www.last.fm	✓			✓
85	t-online.de	✓			✓
86	narod.ru	✓			✓
87	www.w3.org	✓			✓
88	www.instagram.com	✓			✓
89	www.interia.pl	✓			✓
90	cpanel.net	✓			✓
91	www.washingtonpost.com	✓			✓
92	telegram.org	✓			✓
93	bitly.ly	✓			✓
94	home.www.upenn.edu	✓			✓
95	www.huffpost.com	✓			✓
96	themeforest.net	✓			✓
97	techcrunch.com	✓			✓
98	www.usatoday.com	✓			✓
99	www.scribd.com	✓			✓
100	www.buydomains.com	✓			✓
101	corriere.it	✓			✓
102	www.fandom.com	✓			✓
103	wix.com	✓			✓
104	www.skyrock.com	✓			✓
105	www.bloomberg.com	✓			✓
106	gen.xyz	✓			✓
107	www.mediafire.com	✓			✓
108	ask.fm	✓			✓
109	www.quora.com	✓			✓
110	tc-leopoldshafen.de			✓	✓
111	azperacount.000webhostapp.com			✓	✓

112	tokokainbandung.com			✓	✓
113	verifywall.cravenhill.site			✓	✓
114	luvmp.com			✓	✓
115	jtadeo.com			✓	✓
116	stalkercup.info			✓	✓
117	joomelink.com			✓	✓
118	yourfitnesscorner.com			✓	✓
119	lonestarcommissary.com			✓	✓
120	magalu.promocoesnoapp.tech			✓	✓
121	bizarro.surf			✓	✓
122	www.superliquidamagazine.com			✓	✓
123	docesessencias.com.br			✓	✓
124	vantagens-itau.ml			✓	✓
125	serenafragancia.xyz			✓	✓
126	defib.uk			✓	✓
127	itauapp.cf			✓	✓
128	stormgristly.co.kr			✓	✓
129	it-suporte2020.tk			✓	✓
130	rxj58.weblium.site			✓	✓
131	superliquidamagazine.com			✓	✓
132	liquida-magazine.com			✓	✓
133	www.topofertasbr.com			✓	✓
134	allegro.host			✓	✓
135	xdizi1.com			✓	✓
136	imma.vn			✓	✓
137	icloud.com.us-fmi.live			✓	✓
138	gleaners.org			✓	✓
139	shikharinsurance.com			✓	✓
140	t.com			✓	✓
141	institutocriativo.org.br			✓	✓
142	verlimites.me			✓	✓
143	psychotherapiepraktijkteriele.nl			✓	✓
144	bcpzonasegurabeta-viabcp-com.qatarriders.qa			✓	✓
145	apple.com.diztk.xyz			✓	✓

Figure B.1: Lists of Domains (Green(Similar characteristics for legitimate data), Red(Similar characteristics for phishing data), Yellow(Dissimilar characteristics for legitimate data), Brown(Dissimilar characteristics for phishing data))

Appendix C

Web API and Proof-of-Concept Web Application

HTML and CSS were used for creating page structure and styling of web page. In addition, JavaScript provided dynamic functionality to web page by handling HTML elements. Based on the output of backend functions, JavaScript action method call on check box to response the user input request and show the result on screen.

In text box, users can enter the domain name of website to verify. By clicking on check box, it shows whether the entered domain is legitimate or phishing. Fig C.1 shows an example of legitimate domain verification and Fig C.2 shows an example of phishing domain verification.

The web application also returns an error message on wrong input data. It shows error messages on screen when input request is black or input data is other than domain name of the website. Fig C.3 and Fig C.4 display examples of error messages. On Fig C.3, the enter input is full website address of a web page. In order to do domain verification, the user should enter only domain name of a website. A domain is the name of website whereas full website address is used to fetch web pages of the website. For classify the domain category, I have used domain name since an organization uses a domain name of website to generate the SSL certificate. On Fig C.4 shows error on black input request.

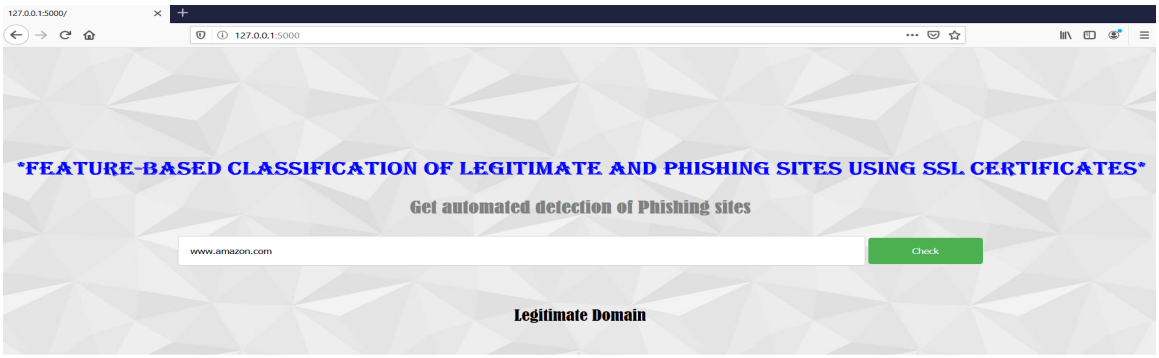

Figure C.1: Example of a legitimate domain

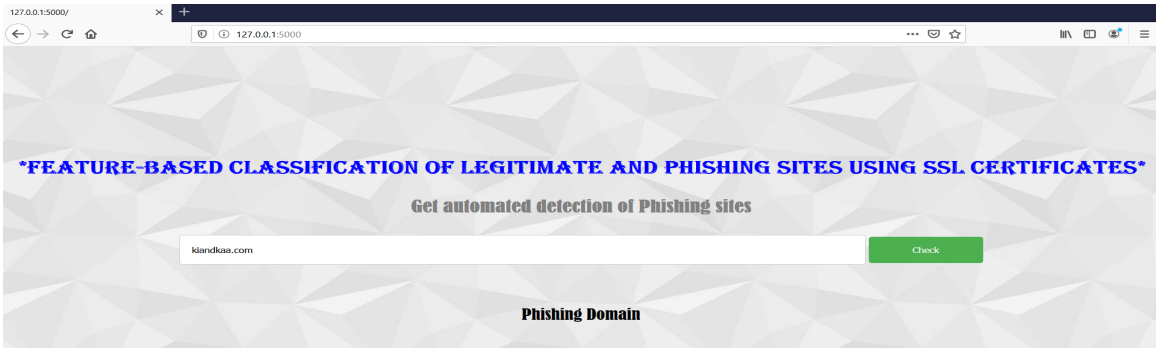

Figure C.2: Example of a phishing domain

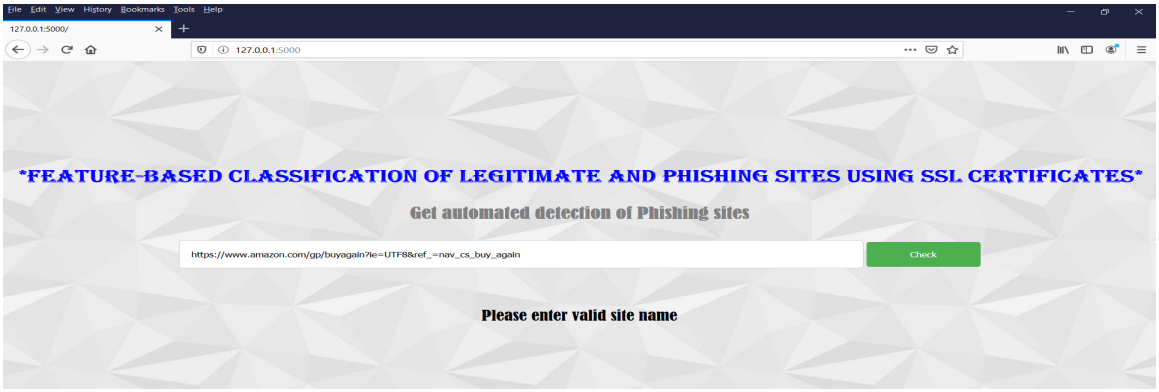

Figure C.3: Example of error case 1

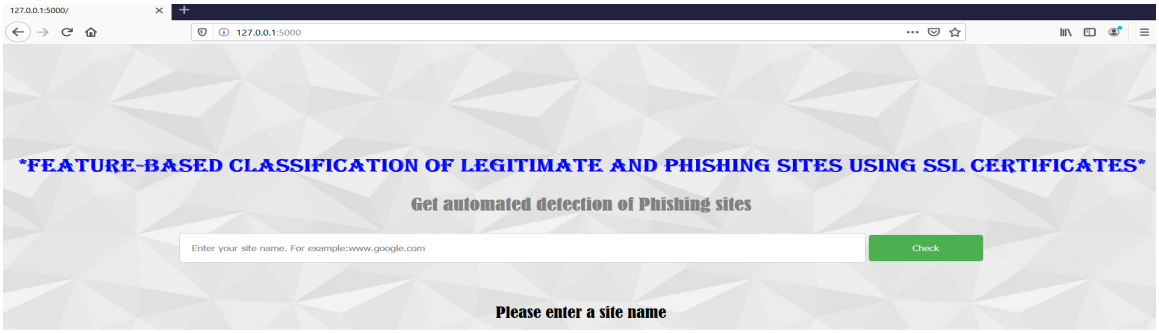

Figure C.4: Example of error case 2