Assessing the Effectiveness of Malicious Domain Prediction Using Machine Learning

by

Jinlin Bu

B.Eng, Civil Aviation University of China, Tianjin, 2018

A Report Submitted in Partial Fulfillment
of the Requirements for the Degree of

MASTER OF ENGINEERING

in the Department of Electrical and Computer Engineering

University of Victoria

Supervisory Committee

Assessing the Effectiveness of Malicious Domain Prediction
Using Machine Learning

by

Jinlin Bu

B.Eng, Civil Aviation University of China, Tianjin, 2018

Supervisory Committee

Dr. Issa Traore, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Isaac Woungang, Departmental Member
(Department of Electrical and Computer Engineering)

Abstract

Malicious domains are a serious threat to network security as they deceive users into accessing them, leading to information disclosure, identity theft, and economic losses. Despite efforts to tackle this problem, cybercriminals continue to buy and use brand-new domains to evade detection, bypassing network defenses and endangering users' security. Predicting future malicious domains in advance can greatly reduce their harm. The Domain Prediction System (DPS*)* developed by one of the industry partners of the Information Security and Object Technology (ISOT) Lab aims to predict in advance potentially malicious domains, but the effectiveness of the system needs to be tested as it is uncertain whether the predicted domains will be used for malicious purposes. This report introduces the problem's background and a description of the dataset used in the experiments. Then evaluates the effectiveness of the DPS system by comparing two sets of models: baseline and predictive models. The baseline models were obtained by training and testing different machine learning (ML) classifiers using existing (known) benign and malicious domains. The predictive models were obtained by training the ML classifiers using domains generated by the DPS that may be used for malicious purposes, and testing using the same benign domains as previously. The evaluation of the predictive models on the same test set as the baseline models yielded comparable performance measures, providing a strong indication of the utility and credibility of the predicted domains.

Table of Contents

List of Tables

List of Figures

Glossary

TLD: Top-level Domain

NOD: Newly Observed Domain

ML: Machine Learning

DPS: Domain Prediction System

ISOT: Information Security and Object Technology Research Lab at the University of Victoria

DNS: Domain Name System

NRD: Newly Registered Domain

C&C: Command and Control

DDoS: Distributed Denial of Service

ICANN: Internet Corporation for Assigned Names and Numbers

AI: Artificial Intelligence

TP: Ture Positive

FN: False Negative

FP: False Positive

TN: True Negative

DL: Deep Learning

CNN: Convolutional Neural Network

RL: Reinforcement Learning

Acknowledgments

I would like to express my heartfelt gratitude to Dr. Issa Traoré, my supervisor, for his unwavering support and generous assistance throughout my master's program. Whenever I encountered difficulties, he was always there to guide me with his expertise and provide valuable insights that helped me complete my MEng project successfully. Moreover, I appreciate Dr. Issa Traoré's encouragement and help for my future career plans in Canada. His guidance has been instrumental in shaping my academic and professional aspirations, and I feel privileged to have had him as my supervisor.

I would like to express my sincere appreciation to the friends I made at University of Victoria. Thank you, Ahmed Abouelkhaire, Ahmed Farag, Achyuth Nandikotkura, Jianfeng Liu, Haidong Wang, Zhonglin Hu and Junlin Shang, my dear friends, for your kindness, generosity, and camaraderie!

Dedication

I would like to respectfully dedicate this report to the memory of my beloved grandmother.

Chapter 1: Introduction

1.1 Context

By using the Internet, everyone today will unavoidably come across various domains. Malicious domains are concealed among them and are used by cybercriminals or hackers to disseminate malware, steal personal data, launch cyberattacks, and participate in other illegal activities. Malicious domains have grown in importance as a way of disseminating crimes with the growth of the Internet and information technology on a worldwide scale, posing severe hazards to network security and user privacy.

As cybercrime continues to evolve and become more sophisticated, the number of new malicious domains being created is rapidly increasing. From the 36 million DNS requests to malicious domains blocked by Canadian Shield, 506,974 unique malicious domains were associated with 638 top-level domains (TLDs) in 2021 [1]. As shown in Figure 1.1, Akamai researchers flagged nearly 79 million domains as malicious in the first half of 2022 alone, according to a newly observed dataset of domains, which equates to roughly 13 million malicious domains per month or 20.1% of all successfully resolved Newly Observed Domains (NODs) [2]. It might be challenging for users to identify malicious domains and avoid falling victim to their malicious activities because malicious domains often masquerade as legitimate domains.

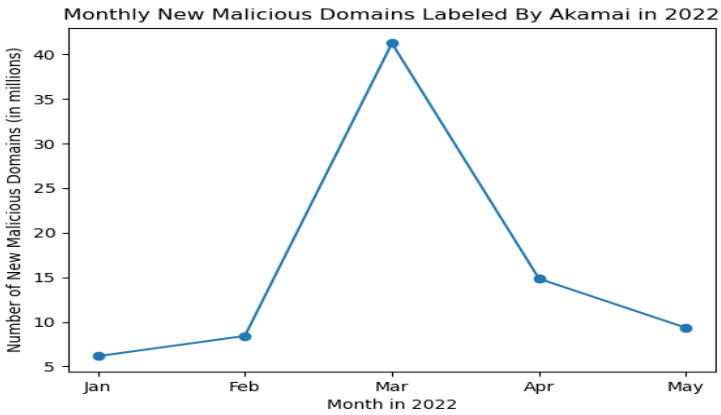

Figure 1.1: New Malicious Domains Labeled by Akamai in 2022 [2].

Conventional approaches to identifying malicious domains need the creation of manual rules and ongoing updating to stay current with new threats. However, these methods have limitations in coverage and detection effectiveness, as cybercriminals continue to adapt their tactics to evade detection. New technology must thus be put in place immediately to address this issue.

By accurately estimating and recognizing dangerous domains in advance, substantial losses in the future may be prevented. With emerging detection techniques that leverage Machine Learning (ML), it may be possible to identify and block malicious domains in real-time. These technologies make it possible to quickly discover suspicious patterns and behavior in enormous amounts of data analysis, allowing for the early identification and mitigation of dangers. In summary, it is crucial to develop and adopt new technologies that can efficiently identify and prevent these attacks since the number of malicious domains keeps growing.

1.2 Related Work

There are two widely used approaches for detecting and mitigating the impact of malicious domains currently.

Rule-based approaches detect malicious domains using manually set criteria such as domain age, registration details, and hosting location. These guidelines are based on the characteristics of known malicious domains. Although these techniques are simple to adopt and comprehend, they must be updated on a regular basis to keep up with newly developing risks.

Because of its capacity to identify previously unseen patterns and behaviors, machine learning-based techniques are growing in popularity for malicious domain detection in recent years. These techniques can be used to train models that can detect malicious domains in real-time by analyzing enormous datasets of domain properties, including DNS traffic data, WHOIS data, and SSL certificate information.

Many researchers have made progress in finding solutions to detect malicious domains. Almarzooqi used statistical knowledge and proposed a method to predict malicious domain names by extracting Features in TLD [3]. Ghafir and Prenosil conducted a study on malicious domain detection by analyzing DNS requests and matching them to blacklists. They proposed improvements to existing blacklist-based approaches for detecting malicious domains [4]. Vinayakumar et al. proposed a deep learning-based approach to detect malicious domain names at scale [5].

1.3 Objective

This project intends to assess the effectiveness (e.g., accuracy) of the domain prediction system (DPS) developed by one of the industry partners of the Information Security and Object Technology (ISOT) Research Laboratory at the University of Victoria. Existing malicious domains are known to be such because they were used in at least one instance of fraudulent activity. In contrast, while some predicted domains are clearly malicious because they are part of some blacklist, a few others may not have such a history because they have not yet been used in malicious activities. Therefore, there are some uncertainties about whether they can be categorized as potentially malicious. The approach used to evaluate the effectiveness of DPS consists of assessing the usefulness of the generated data in developing a detection model using machine learning. This was done by training and comparing two separate detection models using different machine learning techniques. The first model was trained using a dataset consisting of a mix of existing benign domains and predicted malicious domains, and then tested using a dataset consisting of existing benign and malicious domains. The second model, which served as a baseline, was trained using a dataset of existing legitimate and malicious domains and tested using the same test dataset used in testing the first model, which contains only known benign and malicious domains. The performance for the two models were then compared, and suggestions for improving the domain prediction system were made based on the results.

1.4 Report Outline

The report structure is as follows.

Chapter 1 introduced the context of the project, the related work, the project objective, and an outline of the report.

Chapter 2 provides background information on detecting malicious domains, including relevant basic knowledge, malicious domain classification, characteristics, and potential risks. The ISOT domain prediction system is also discussed.

Chapter 3 describes the datasets used in the project, including their sources, and the preprocessing steps performed to extract features.

Chapter 4 presents the experimental procedure and performance metrics, evaluates, and compares the performance of different machine learning models, and discusses the results.

Chapter 5 makes concluding remarks and suggestions for future work.

Chapter 2: Background

Understanding the fundamentals of malicious domains is necessary to detect the potential presence of malicious domains in the future or domain names that will be used for malicious purposes. In this chapter, the basics, classification, characteristics, and hazards of malicious domains will be covered, along with the introduction of DPS.

2.1 Basics

A domain, also known as a domain name, is a unique identification for a computer or a group of computers on the internet. It serves the purpose of recognizing and locating the device during data transmission. Figure 2.1 depicts the format of a domain name, which consists of a succession of names separated by dots.

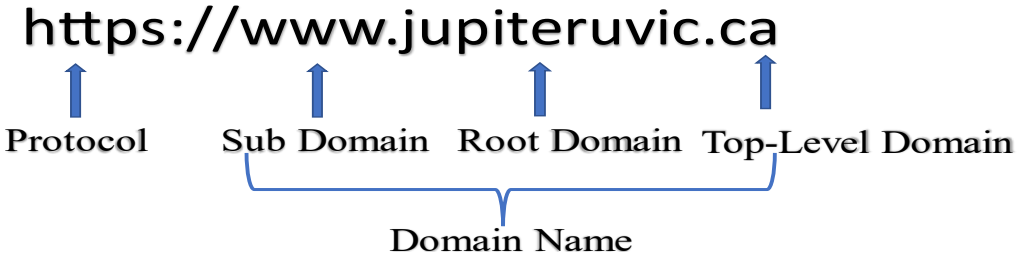

Figure 2.1: Anatomy of a Domain.

The Domain Name System (DNS) plays an important role in simplifying internet access for users by mapping domains and IP addresses together. Figure 2.2 provides a clear illustration of how DNS operates.

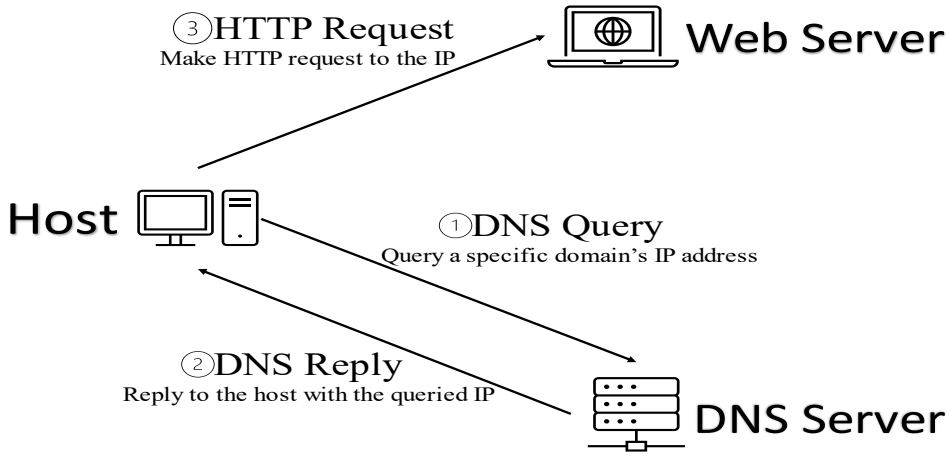

Figure 2.2: DNS Workflow.

Due to the limited availability of quality domain names, domains are distinctive. To accomplish their objectives, cybercriminals may occasionally purchase many domains in advance. They will then utilize some of them first and save the rest for later. According to statistics for the first half of 2019, 69.73% of Newly Registered Domains (NRDs) are suspicious, as shown in Figure 2.3 based on data originating from [6]. These domains might potentially be used maliciously in the future, which is highly likely.

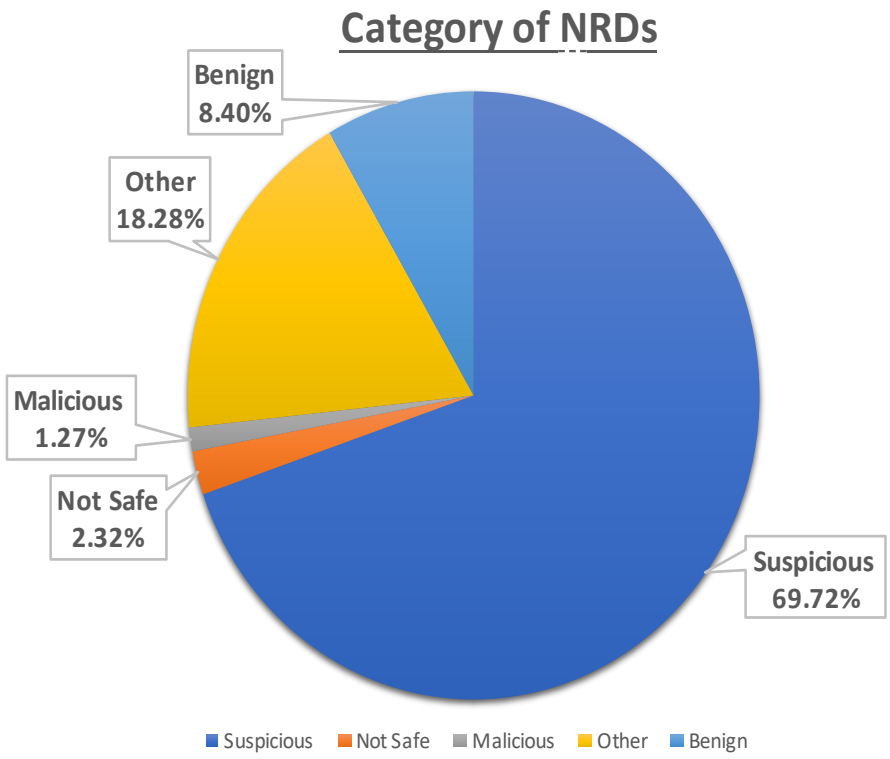

Figure 2.3: Category of NRDs in the First Half of 2019.

2.2 Classification of Malicious Domains

Based on their intended uses and the kinds of threats they pose; malicious domains can be divided into categories. Below are several examples of typical malicious domains.

Malware distribution domains: These domains distribute malicious software, such as viruses, ransomware, or trojans, through drive-by downloads or exploit kits. Malicious distribution domains are usually hidden in scenarios such as spam emails and malicious advertisements.

Phishing domains: In order to steal sensitive information like credit card numbers, personal information, or login passwords, phishing domains are designed to deceive users into thinking they are authentic domains. This may lead to dishonest behavior and frauds, as seen in Figure 2.4. To trick users, phishing domains frequently masquerade as well-known websites like banks, e-commerce sites, and social networking networks.

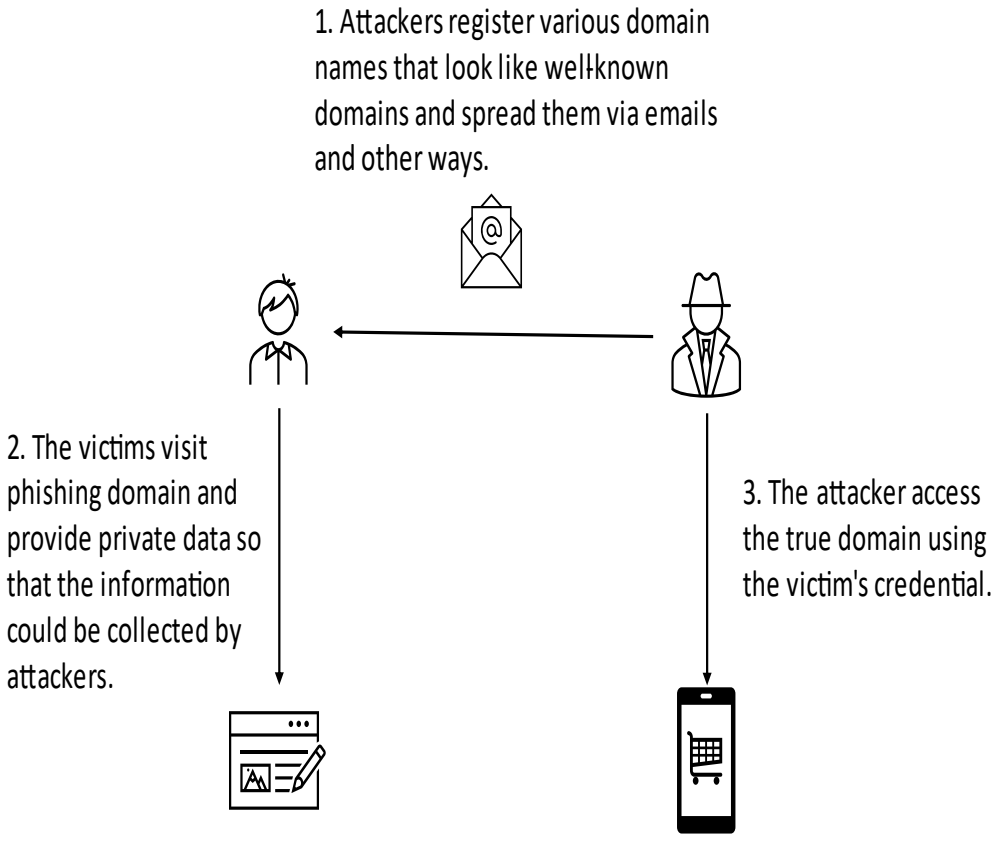

Figure 2.4: Phishing Domains Attack Process.

Command and Control (C&C) domains: These domains are used to control networks of compromised computers, or botnets, for various purposes. By controlling an infected computer, malicious behaviors such as network attacks and Distributed Denial of Service (DDoS) attacks can be carried out. C&C domains are often hidden within malware and can take remote control of infected computers [7].

It is important to keep in mind that parked domains and typosquatting may also be used maliciously [8]. Therefore, they should all be considered.

2.3 Characteristics of Malicious Domains

Malicious domains usually have some characteristics that tell them distinct from legitimate domains. These characteristics consist of the following:

1. The length of the domain tends to be either too short or excessively long: This could be used to facilitate quick access by users, or to confuse the public and make it difficult for people to distinguish the authenticity [9].
2. Obfuscated domain names: They use misspelled domain names that resemble legitimate domains. Users find it challenging to recognize them as harmful because of this.
3. Contains numbers or special characters: This is to increase the complexity of the domain name to bypass traditional malicious domain name detection methods.
4. Short lifespan: Malicious domains are often registered for short periods, typically ranging from a few days to a few weeks.
5. Frequently changing IP addresses: Malicious domains alter their IP addresses on a frequent basis to avoid being detected and blocked by defense systems.
6. Suspicious registration information: The registration information of malicious domain names is usually false or uses stolen identity information to conceal the identity and avoid tracking.
7. Use of redirection: They may use redirection techniques to redirect users to other domains that may contain malware or phishing websites.
8. High traffic volume: Malicious domains may have a high volume of traffic, particularly if they are used to distribute malware or carry out phishing attacks [10].
9. Lack of SSL certificate: Many legitimate domains use SSL certificates to secure communication between the website and the user's browser. Malicious domains may lack SSL certificates, making it easier to detect them as suspicious [11].

By identifying these characteristics, it was feasible to extract more effective features for use in machine learning during this project.

2.4 Potential Risks of Malicious Domains

The harmful impact of malicious domains can be shown in the following aspects.

Spreading malware: Hackers can spread malware to users' computers through malicious domains, thereby stealing users' personal privacy information.

Stealing personal information: Hackers lure users into entering sensitive information by using malicious domains for camouflage.

Launch network attacks: Hackers can launch DDoS attacks and other network attacks through malicious domains, resulting in serious consequences such as target system paralysis or data leakage.

Therefore, the detection and defense of malicious domains play a vital role, not only to protect user's personal privacy and network security, but also to maintain the health and stability of the entire network ecological environment.

2.5 Domain Predict System Overview

The flow chart of DPS is shown in Figure 2.5. Importing a zone file is required before utilizing the DPS. A .com zone file from the Internet Corporation for Assigned Names and Numbers (ICANN) February 2023 data was imported for this experiment.

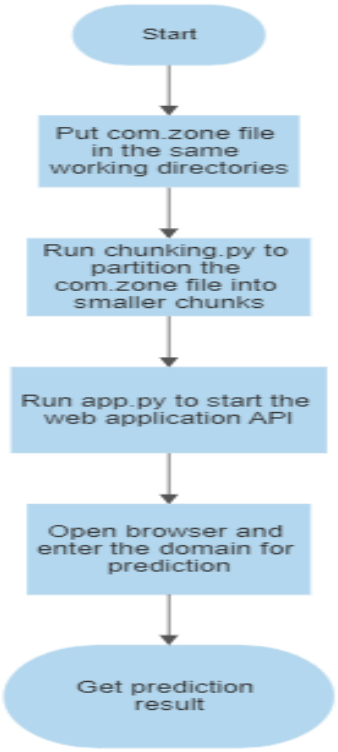

Figure 2.5: DPS Flow Chart.

A zone file is a text file that contains information about the DNS resource records for a specific domain or subdomain. This information includes the IP addresses and other resource records associated with the domain, such as MX records for email servers or NS records for specifying the authoritative name servers for the domain [12]. The information and anatomy of a com zone file are shown in Figure 2.6.

Host Label	TTL	Record Class	Record Type	Record Data
viewbergencountyhomes.com.	172800	in	ns	ns37.domaincontrol.com.
viewbergencountyhomes.com.	172800	in	ns	ns38.domaincontrol.com.
viewberkeleycountyhomes.com.	172800	in	ns	ns1.livebuyers.com.
viewberkeleycountyhomes.com.	172800	in	ns	ns2.livebuyers.com.
viewberkscountyareahomes.com.	172800	in	ns	ns1.cincpro.com.
viewberkscountyareahomes.com.	172800	in	ns	ns2.cincpro.com.
viewberkscountyareahomes.com.	172800	in	ns	ns3.cincpro.com.
viewberkscountyareahomes.com.	172800	in	ns	ns4.cincpro.com.
viewberkscountyhomes.com.	172800	in	ns	ns-1403.awsdns-47.org.
viewberkscountyhomes.com.	172800	in	ns	ns-1816.awsdns-35.co.uk.
viewberkscountyhomes.com.	172800	in	ns	ns-286.awsdns-35.com.
viewberkscountyhomes.com.	172800	in	ns	ns-680.awsdns-21.net.
viewberkshirehomes.com.	172800	in	ns	ns1.cincpro.com.

Figure 2.6: A Snippet of the .com Zone File.

Using DPS will partition the .com.zone file into small chunks in alphabetical order and store them separately into URLs directories and nameservers directories. The format of the files in these two directories is shown in Figure 2.7 and Figure 2.8.

```
xmczzx.com. 172800  in  ns  ns1.dns.com.
xmczzx.com. 172800  in  ns  ns2.dns.com.
xmd-5.com.  172800  in  ns  donald.ns.cloudflare.com.
xmd-5.com.  172800  in  ns  leia.ns.cloudflare.com.
xmd-academic.com.   172800  in  ns  jm1.dns.com.
xmd-academic.com.   172800  in  ns  jm2.dns.com.
xmd-bearing.com.    172800  in  ns  adam.dnspod.net.
xmd-bearing.com.    172800  in  ns  jim.dnspod.net.
```

Figure 2.7: A Snippet of a File in chunks_of_urls.

```
goodsportsthailand.com. 172800  in  ns  harlee.ns.cloudflare.com.
goodssarang.com.    172800  in  ns  hns1.nsgodo.com.
goodssarang.com.    172800  in  ns  hns2.nsgodo.com.
goodstady.com.  172800  in  ns  hassan.ns.cloudflare.com.
goodstorycoffeeshop.com.    172800  in  ns  helium.ns.hetzner.de.
goodstorycoffeeshop.com.    172800  in  ns  hydrogen.ns.hetzner.com.
goodstudentcarinsurance.com.    172800  in  ns  hank.ns.cloudflare.com.
goodstuffkw.com.    172800  in  ns  harmony.ns.cloudflare.com.
```

Figure 2.8: A Snippet of a File in chunks_of_ns.

After successfully importing the zone file and running chunking.py and app.py, users can open the rendered page by accessing the browser and loading the domain prediction service using the following URL format. Syntax: http://127.0.0.1:5000/domain_name

According to the user's input, DPS will identify the nameservers resolving the seed domains in the .com zone file through the prediction algorithm[1]. It will then search for the potential domains with the same nameserver and active time, followed by finding the self-

[1] The working of the domain prediction is out of the scope of this project.

resolved domains with the same active time as self-resolved bad domains. DPS will combine the above results, and format seed domain and predicted domains as a dictionary.

Figure 2.9: Example of DPS Predictions.

After a period, DPS will display the prediction results on the webpage, as shown in Figure 2.9. It is worth mentioning that one input seed domain may produce one or multiple domains as a result.

Chapter 3: Dataset and Preprocessing

Building a model for malicious domain detection requires using suitable datasets for training and testing. Our evaluation approach consisted of building two models. The first model under evaluation that we refer to as DPS-based detector, was built by training the ML models using the existing (i.e., known) benign domain names and the predicted malicious domain names generated by DPS. However, the testing of the model was done using only the existing domain names (benign and malicious). Next, we build a baseline malicious domain detection model as a standard ML-based detector trained using the same existing benign domain names as the first model and some known malicious domain names, and tested using the existing domain names (malicious and benign) used for testing the first model. In this chapter, description of the datasets utilized in the experiment, including their source, quantity, and roles will be provided. Additionally, this chapter will outline the pre-processing involved in preparing machine learning data, considering malicious domains' characteristics.

3.1 Dataset

The datasets used in the experiment are shown in the following table.

Table 3.1: Amount, Sources, Type and Role of Datasets.

Dataset	Amount	Sources	Type	Role
A	10628	Generated by DPS	Malicious	Training with dataset B to test the effectiveness of DPS
B	20700	Alexa dataset	Legitimate	Training for two models
C	10628	PhishTank and ISOT	Malicious	Training for baseline malicious domain detection model
D	7832	Alexa dataset,PhishTank and ISOT	Mixed	Testing for two models
(Sum)	49788	-	-	-

Predicted domains were generated by using DPS and inputting 40 existing malicious domains as seed, yielding dataset A, with a total of 10,628 domain names. All predicted domains (generated by DPS) are claimed to be potentially malicious and were categorized as such in the study.

The existing legitimate domain data in the experiment was compiled using a part of the Alexa Top 1 million domain name dataset [13] on Kaggle, which comprised a total of 25,875 domains. The data was provided by Alexa and contains the most popular website domains in the world. These domain names are labeled as legitimate.

The existing malicious domain data, part of which was obtained from PhishTank [14] and consists mainly of some phishing domain names, and the other part was collected previously by ISOT laboratory, including other kinds of malicious domain names. There are 13,285 such domains in total. These domain names are labeled as malicious.

Next, we selected randomly 80% of the existing malicious and legitimate domain data for training purposes. This resulted in 20,700 samples labeled as legitimate domains and 10,628 samples labeled as malicious domains, which were respectively divided into dataset B and dataset C based on their labels.

The remaining 20% of the data was reserved for testing and formed dataset D, which comprised a total of 7,832 samples. This dataset contained a mix of legitimate and malicious domains.

The class distribution of a dataset has a significant impact on the precision and effectiveness of training a machine learning model. The trained model may tend to predict more categories while disregarding a smaller number of categories, leading to model inaccuracy, if the data ratio of the two categories is out of balance. To prevent this issue, in the experiment, the ratio of datasets A, B, and C is around 1:2:1.

The datasets were used in two sets of experiments as follows:

1. Experiment 1: Merge datasets A and B and use only the merged dataset AB in the training process, and then use dataset D for testing. This allows assessing the effectiveness of the domain prediction system in generating malicious domain names.
2. Experiment 2: Combine datasets B and C and train the same classifiers as experiment 1. After testing on dataset D, the results are used as a comparison to prove that the accuracy of the domain prediction system satisfies certain requirements.

3.2 Data Preprocessing

Due to the sheer quantity and diversity of domain data, it may not be sufficient to process it without appropriate preparation. In other words, the performance of the machine learning model is closely linked to the selection of relevant features. A necessary step is to extract relevant features through preprocessing. Figure 3.1 depicts the flow chart involved in preprocessing the data.

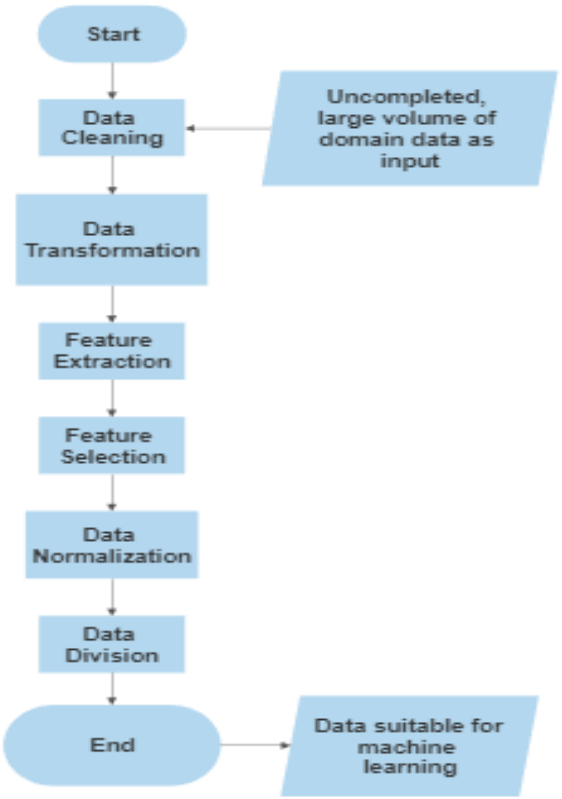

Figure 3.1: Flow Chart of Data Preprocessing.

As shown in Figure 3.1, the preprocessing steps involved in the experiments are as follows.

The first step is data cleaning, which involves recognizing and deleting redundant, inaccessible, and superfluous domains. This is a crucial stage since it assures the validity and accuracy of the data utilized for analysis.

The second phase is data transformation, which consists of converting unprocessed data into an analysis-ready format. Domains are transformed into feature vectors in this project so they may be used in further analyses.

The third step consists of feature extraction, which involves extracting useful information from unprocessed data. In this instance, pertinent features such as domain length, character type, frequency, and misspelling are extracted from the domains [15]. Subsequently, these properties are applied to train machine learning models.

The fourth stage, feature selection, seeks to discover the optimal feature subset. Feature selection may omit less important characteristics in order to reduce the number of features, boost model validity, and shorten runtime.

The fifth stage is data normalization, which entails bringing the values of several features into the same range. The objective of this phase is to remove dimensional discrepancies between features, which will make classifier training and prediction easier.

The final step is data division, which entails splitting the initial dataset into a training set and a testing set. This is done to assess the performance of the machine learning model on unseen data.

Overall, these processes are critical for assuring the quality and reliability of analysis data, as well as enhancing the performance of machine learning models used for prediction and classification.

Chapter 4: Implementation and Performance Evaluation

This chapter will describe the whole processes and explain the experiment's implementation stages, as well as analyze and compare the results.

4.1 Processes

Machine learning is a vital artificial intelligence (AI) method that enables computers to learn and train on massive volumes of data to give predicted patterns. Machine learning is used in this project to classify malicious and legitimate domains. The implementation is mainly divided into two experiments, the process of the whole experiment is depicted in Figure 4.1.

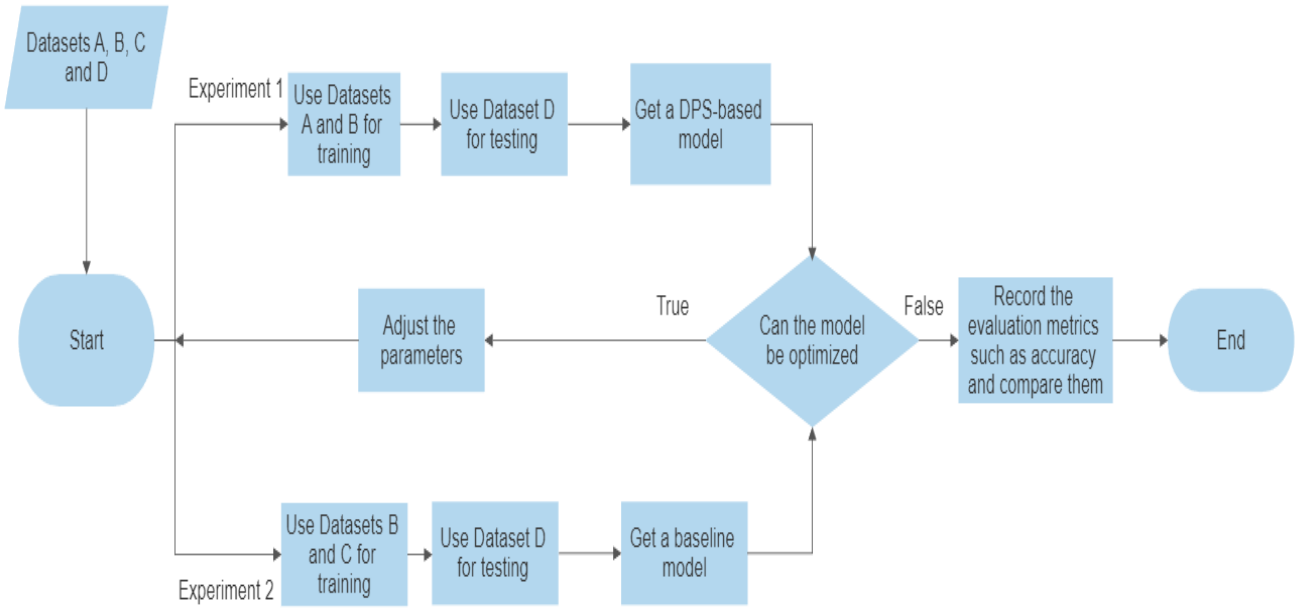

Figure 4.1: Implementation Processes.

The experiment 1 involves training the model under evaluation using dataset A obtained from the domain prediction system along with the existing benign domain dataset B. The existing domains dataset D is used just for testing. Subsequently, adjusting the parameters in each model, and recording the evaluation metrics of models. The purpose of this stage is to verify the credibility of the domain dataset obtained by the domain name prediction system as a malicious domain. Specifically, if the domain prediction system can

successfully predict malicious domains, the generated domains should have some of the same characteristics as the malicious domain. Therefore, the knowledge gained during training on dataset A and B can be used to classify dataset D.

The experiment 2 is training the baseline model using datasets B and C and testing using dataset D. Then, update the parameters of each model continuously and optimize the selected features. Train and test the models again to improve the performance. The purpose of this step is to guarantee precise evaluation of the domain prediction system. It is crucial to establish judgment criteria, which can be done by using the existing benign domain dataset B and the existing malicious domain dataset C for training and testing using the same testing dataset D in the experiments, and the resulting accuracy can serve as valuable reference points.

Finally, compare and evaluate the accuracy and other relevant evaluation metrics of the two models. The purpose of this stage is to determine the trustworthiness of the DPS by comparing the evaluation criteria. If there is a large gap between the evaluation metrics of the two models, it proves that most of the domains generated by the DPS do not have the characteristics of malicious domains, so the purpose of predicting future malicious domains cannot be achieved. On the contrary, if there is little difference in the accuracy and other evaluation metrics of the two models, it proves that the DPS can generate domains with the same characteristics as malicious domains, which are likely to be used for malicious purposes in the future.

In summary, via these two experiments, the effectiveness of DPS for generating domains that may be used for malicious purposes in the future could be assessed as a whole, and the objective of identifying malicious domains beforehand could be accomplished.

4.2 Evaluation Metrics

Machine learning includes methods such as supervised learning, unsupervised learning, and semi-supervised learning. In this experiment, the data is already labeled, and the label is a typical binary classification, so classifying such data is supervised machine learning.

To evaluate the performance of such a supervised classification model, a confusion matrix is provided.

In this experiment, the confusion matrix is a 2 × 2 analysis table, as shown in Table 4.1. Each column of the confusion matrix represents the predicted category, and the total number of each column indicates the number of samples predicted as this category. Each row represents the true category of the sample; the total number of samples in each row corresponds to the number of data instances of this category; the value in each column indicates the number of real samples predicted as this category.

Table 4.1: Confusion Matrix.

		Prediction	
	Domain Types	**Malicious**	**Legitimate**
Reference	**Malicious**	***True Positive***	***False Negative***
	Legitimate	***False Positive***	***True Negative***

- True Positive (TP): The model correctly predicted the malicious domain.
- False Negative (FN): The model classified the sample domain as a legitimate domain although its real category was a malicious domain.
- False Positive (FP): The sample really belongs to the legitimate domain category, but the model classified it as malicious.
- True Negative (TN): The sample's true class is a legitimate domain, and the model identified it as such.

The confusion matrix may provide more detailed performance evaluation information and is also an important foundation for model optimization.

In our experiment, other evaluation metrics were also used, including accuracy, precision, recall rate and F1-score. The explanations of these evaluation metrics are provided below.

Accuracy is the fundamental evaluation metric. It measures the agreement between the predicted results and the actual results. Higher accuracy indicates better precision in model prediction. However, solely focusing on accuracy fails to reflect the comprehensive performance of the model, as it may overfit or suffer from class imbalance. Accuracy is calculated as follows:

$$Accuracy = 100\% \times \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Precision is the model's capacity to detect all legitimate and malicious domains. It is the percentage of legitimate and malicious domains properly identified. The classification accuracy of the model increases with precision, but this may also lead to a rise in the false negative rate. Precision is computed as:

$$Precision = 100\% \times \frac{TP}{TP + FP} \tag{4.2}$$

Recall rate is the proportion of true malicious domains detected among all malicious domains and legitimate domains, and it refers to the model's ability to accurately predict all malicious domains. The higher the recall rate, the better the model's capacity to detect malicious domains, but it may also raise the false positive rate. Missing certain malicious domains might lead to security risks, hence the recall rate is a crucial indicator. Recall is calculated as follows:

$$Recall\ Rate = 100\% \times \frac{TP}{TP + FN} \tag{4.3}$$

F1-score is the harmonic mean of precision and recall, which can comprehensively consider the precision and recall of the model. The higher the F1-score, the better the overall performance of the model. F1-score is computed as follows:

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{4.4}$$

In the experimental process, reports containing these evaluation metrics can be generated by importing sklearn.metrics.classification_report [16].

4.3 Experiment Results

Experiment One: Train the model using datasets A and B, and test using dataset D.

A total of 7 machine learning algorithms were explored in the experiment, and their evaluation metrics are summarized in Table 4.2.

Table 4.2: The Models and Their Evaluation Metrics in Experiment One.

Models	Accuracy	Precision	Recall	F1-Score
Logistic Regression	89.87%	Malicious:0.80 Legitimate:0.99	Malicious:0.99 Legitimate:0.84	Malicious:0.88 Legitimate:0.91
Naive Bayes	90.44%	Malicious:0.80 Legitimate:0.99	Malicious:0.99 Legitimate:0.85	Malicious:0.89 Legitimate:0.92
Decision Tree	91.27%	Malicious:0.82 Legitimate:0.99	Malicious:0.99 Legitimate:0.86	Malicious:0.90 Legitimate:0.92
KNN	86.77%	Malicious:0.75 Legitimate:0.98	Malicious:0.98 Legitimate:0.80	Malicious:0.85 Legitimate:0.88
Gradient Boosting	88.67%	Malicious:0.78 Legitimate:0.99	Malicious:0.98 Legitimate:0.83	Malicious:0.87 Legitimate:0.90
Random Forest	92.08%	Malicious:0.83 Legitimate:0.99	Malicious:0.99 Legitimate:0.88	Malicious:0.91 Legitimate:0.93
SVM	91.62%	Malicious:0.82 Legitimate:0.99	Malicious:0.99 Legitimate:0.87	Malicious:0.90 Legitimate:0.93

The following are detailed results of the 7 models in Experiment 1.

Logistic Regression

Table 4.3: Confusion Matrix of Logistic Regression Model in Experiment One.

Prediction

Reference	Domain Types	Malicious	Legitimate
	Malicious	***2948***	***37***
	Legitimate	***756***	***4091***

	precision	recall	f1-score	support
malicious	0.80	0.99	0.88	2985
legitimate	0.99	0.84	0.91	4847
accuracy			0.90	7832
macro avg	0.90	0.90	0.90	7832
weighted avg	0.90	0.90	0.90	7832

Figure 4.2: Evaluation Metrics Report for Logistic Regression in Experiment One.

Naive Bayesian

Table 4.4: Confusion Matrix of Naive Bayes Model in Experiment One.

Prediction

Reference	Domain Types	Malicious	Legitimate
	Malicious	***2952***	***33***
	Legitimate	***716***	***4131***

	precision	recall	f1-score	support
malicious	0.80	0.99	0.89	2985
legitimate	0.99	0.85	0.92	4847
accuracy			0.90	7832
macro avg	0.90	0.90	0.90	7832
weighted avg	0.90	0.90	0.90	7832

Figure 4.3: Evaluation Metrics Report for Naive Bayesian in Experiment One.

Decision Tree

Table 4.5: Confusion Matrix of Decision Tree Model in Experiment One.

		Prediction	
	Domain Types	**Malicious**	**Legitimate**
Reference	**Malicious**	***2959***	***26***
	Legitimate	***658***	***4189***

	precision	recall	f1-score	support
malicious	0.82	0.99	0.90	2985
legitimate	0.99	0.86	0.92	4847
accuracy			0.91	7832
macro avg	0.91	0.91	0.91	7832
weighted avg	0.91	0.91	0.91	7832

Figure 4.4: Evaluation Metrics Report for Decision Tree in Experiment One.

K Nearest Neighbor (KNN)

Table 4.6: Confusion Matrix of KNN Model in Experiment One.

Prediction

Reference	Domain Types	Malicious	Legitimate
	Malicious	*2913*	*72*
	Legitimate	*964*	*3883*

	precision	recall	f1-score	support
malicious	0.75	0.98	0.85	2985
legitimate	0.98	0.80	0.88	4847
accuracy			0.87	7832
macro avg	0.87	0.87	0.87	7832
weighted avg	0.87	0.87	0.87	7832

Figure 4.5: Evaluation Metrics Report for KNN in Experiment One.

Gradient Boosting

Table 4.7: Confusion Matrix of Gradient Boosting Model in Experiment One.

Prediction

Reference	Domain Types	Malicious	Legitimate
	Malicious	*2932*	*53*
	Legitimate	*837*	*4010*

	precision	recall	f1-score	support
malicious	0.78	0.98	0.87	2985
legitimate	0.99	0.83	0.90	4847
accuracy			0.89	7832
macro avg	0.89	0.89	0.89	7832
weighted avg	0.89	0.89	0.89	7832

Figure 4.6: Evaluation Metrics Report for Gradient Boosting in Experiment One.

Random Forest

Table 4.8: Confusion Matrix of Random Forest Model in Experiment One.

		Prediction	
	Domain Types	**Malicious**	**Legitimate**
Reference	**Malicious**	***2963***	***22***
	Legitimate	***598***	***4249***

	precision	recall	f1-score	support
malicious	0.83	0.99	0.91	2985
legitimate	0.99	0.88	0.93	4847
accuracy			0.92	7832
macro avg	0.92	0.92	0.92	7832
weighted avg	0.92	0.92	0.92	7832

Figure 4.7: Evaluation Metrics Report for Random Forest in Experiment One.

Support Vector Machine (SVM)

Table 4.9: Confusion Matrix of SVM Model in Experiment One.

Prediction

Reference	Domain Types	Malicious	Legitimate
	Malicious	***2959***	***26***
	Legitimate	***630***	***4217***

	precision	recall	f1-score	support
malicious	0.82	0.99	0.90	2985
legitimate	0.99	0.87	0.93	4847
accuracy			0.92	7832
macro avg	0.92	0.92	0.92	7832
weighted avg	0.92	0.92	0.92	7832

Figure 4.8: Evaluation Metrics Report for SVM in Experiment One.

Experiment Two: Train the model using dataset B and C, and test using dataset D.

Apply the same 7 ML models and record their evaluation metrics in the tables below.

Table 4.10: The Models and Their Evaluation Metrics in Experiment Two.

Models	Accuracy	Precision	Recall	F1-Score
Logistic Regression	93.83%	Malicious:0.86 Legitimate:1.00	Malicious:1.00 Legitimate:0.90	Malicious:0.93 Legitimate:0.95
Naive Bayes	94.97%	Malicious:0.88 Legitimate:1.00	Malicious:1.00 Legitimate:0.92	Malicious:0.94 Legitimate:0.96
Decision Tree	96.39%	Malicious:0.91 Legitimate:1.00	Malicious:1.00 Legitimate:0.94	Malicious:0.95 Legitimate:0.97
KNN	91.46%	Malicious:0.82 Legitimate:1.00	Malicious:0.99 Legitimate:0.87	Malicious:0.90 Legitimate:0.93
Gradient Boosting	92.00%	Malicious:0.83 Legitimate:1.00	Malicious:1.00 Legitimate:0.87	Malicious:0.90 Legitimate:0.93
Random Forest	98.07%	Malicious:0.95 Legitimate:1.00	Malicious:1.00 Legitimate:0.97	Malicious:0.98 Legitimate:0.98
SVM	97.20%	Malicious:0.93 Legitimate:1.00	Malicious:1.00 Legitimate:0.96	Malicious:0.96 Legitimate:0.98

Logistic Regression

Table 4.11: Confusion Matrix of Logistic Regression Model in Experiment Two.

		Prediction	
	Domain Types	**Malicious**	**Legitimate**
Reference	**Malicious**	***2979***	***6***
	Legitimate	***477***	***4370***

	precision	recall	f1-score	support
malicious	0.86	1.00	0.93	2985
legitimate	1.00	0.90	0.95	4847
accuracy			0.94	7832
macro avg	0.94	0.94	0.94	7832
weighted avg	0.94	0.94	0.94	7832

Figure 4.9: Evaluation Metrics Report for Logistic Regression in Experiment Two.

Naive Bayesian

Table 4.12: Confusion Matrix of Naive Bayes Model in Experiment Two.

Prediction

Reference	Domain Types	Malicious	Legitimate
	Malicious	*2980*	*5*
	Legitimate	*389*	*4458*

	precision	recall	f1-score	support
malicious	0.88	1.00	0.94	2985
legitimate	1.00	0.92	0.96	4847
accuracy			0.95	7832
macro avg	0.95	0.95	0.95	7832
weighted avg	0.95	0.95	0.95	7832

Figure 4.10: Evaluation Metrics Report for Naive Bayesian in Experiment Two.

Decision Tree

Table 4.13: Confusion Matrix of Decision Tree Model in Experiment Two.

Prediction

Reference	Domain Types	Malicious	Legitimate
	Malicious	*2980*	*5*
	Legitimate	*278*	*4569*

	precision	recall	f1-score	support
malicious	0.91	1.00	0.95	2985
legitimate	1.00	0.94	0.97	4847
accuracy			0.96	7832
macro avg	0.96	0.96	0.96	7832
weighted avg	0.96	0.96	0.96	7832

Figure 4.11: Evaluation Metrics Report for Decision Tree in Experiment Two.

K Nearest Neighbor (KNN)

Table 4.14: Confusion Matrix of KNN Model in Experiment Two.

		Prediction	
	Domain Types	**Malicious**	**Legitimate**
Reference	**Malicious**	***2970***	***15***
	Legitimate	***654***	***4193***

	precision	recall	f1-score	support
malicious	0.82	0.99	0.90	2985
legitimate	1.00	0.87	0.93	4847
accuracy			0.91	7832
macro avg	0.91	0.91	0.91	7832
weighted avg	0.91	0.91	0.91	7832

Figure 4.12: Evaluation Metrics Report for KNN in Experiment Two.

Gradient Boosting

Table 4.15: Confusion Matrix of Gradient Boosting Model in Experiment Two.

Prediction

Reference	Domain Types	Malicious	Legitimate
	Malicious	*2974*	*11*
	Legitimate	*615*	*4232*

	precision	recall	f1-score	support
malicious	0.83	1.00	0.90	2985
legitimate	1.00	0.87	0.93	4847
accuracy			0.92	7832
macro avg	0.92	0.92	0.92	7832
weighted avg	0.92	0.92	0.92	7832

Figure 4.13: Evaluation Metrics Report for Gradient Boosting in Experiment Two.

Random Forest

Table 4.16: Confusion Matrix of Random Forest Model in Experiment Two.

Prediction

Reference	Domain Types	Malicious	Legitimate
	Malicious	*2985*	*0*
	Legitimate	*151*	*4696*

	precision	recall	f1-score	support
malicious	0.95	1.00	0.98	2985
legitimate	1.00	0.97	0.98	4847
accuracy			0.98	7832
macro avg	0.98	0.98	0.98	7832
weighted avg	0.98	0.98	0.98	7832

Figure 4.14: Evaluation Metrics Report for Random Forest in Experiment Two.

Support Vector Machine (SVM)

Table 4.17: Confusion Matrix of SVM Model in Experiment Two.

		Prediction	
	Domain Types	**Malicious**	**Legitimate**
Reference	**Malicious**	***2984***	***1***
	Legitimate	***218***	***4629***

	precision	recall	f1-score	support
malicious	0.93	1.00	0.96	2985
legitimate	1.00	0.96	0.98	4847
accuracy			0.97	7832
macro avg	0.97	0.97	0.97	7832
weighted avg	0.97	0.97	0.97	7832

Figure 4.15: Evaluation Metrics Report for SVM in Experiment Two.

Compare the models' evaluation metrics of experiment one and experiment two.

Using accuracy, precision, recall, and f1-score as indicators, we compare the values of the same model in the two experiments and get the following four tables.

Table 4.18: Accuracy Comparison.

Models	Accuracy in Experiment One	Accuracy in Experiment Two	Difference
Logistic Regression	89.87%	93.83%	3.96%
Naive Bayes	90.44%	94.97%	4.53%
Decision Tree	91.27%	96.39%	5.12%
KNN	86.77%	91.46%	4.69%
Gradient Boosting	88.67%	92.00%	3.33%
Random Forest	92.08%	98.07%	5.99%
SVM	91.62%	97.20%	5.58%

Table 4.19: Precision Comparison.

Models	Precision in Experiment One	Precision in Experiment Two	Difference
Logistic Regression	Malicious:0.80 Legitimate:0.99	Malicious:0.86 Legitimate:1.00	Malicious:0.06 Legitimate:0.01
Naive Bayes	Malicious:0.80 Legitimate:0.99	Malicious:0.88 Legitimate:1.00	Malicious:0.08 Legitimate:0.01
Decision Tree	Malicious:0.82 Legitimate:0.99	Malicious:0.91 Legitimate:1.00	Malicious:0.09 Legitimate:0.01
KNN	Malicious:0.75 Legitimate:0.98	Malicious:0.82 Legitimate:1.00	Malicious:0.07 Legitimate:0.02
Gradient Boosting	Malicious:0.78 Legitimate:0.99	Malicious:0.83 Legitimate:1.00	Malicious:0.05 Legitimate:0.01
Random Forest	Malicious:0.83 Legitimate:0.99	Malicious:0.95 Legitimate:1.00	Malicious:0.12 Legitimate:0.01
SVM	Malicious:0.82 Legitimate:0.99	Malicious:0.93 Legitimate:1.00	Malicious:0.11 Legitimate:0.01

Table 4.20: Recall Comparison.

Models	Recall in Experiment One	Recall in Experiment Two	Difference
Logistic Regression	Malicious:0.99 Legitimate:0.84	Malicious:1.00 Legitimate:0.90	Malicious:0.01 Legitimate:0.06
Naive Bayes	Malicious:0.99 Legitimate:0.85	Malicious:1.00 Legitimate:0.92	Malicious:0.01 Legitimate:0.07
Decision Tree	Malicious:0.99 Legitimate:0.86	Malicious:1.00 Legitimate:0.94	Malicious:0.01 Legitimate:0.08
KNN	Malicious:0.98 Legitimate:0.80	Malicious:0.99 Legitimate:0.87	Malicious:0.01 Legitimate:0.07
Gradient Boosting	Malicious:0.98 Legitimate:0.83	Malicious:1.00 Legitimate:0.87	Malicious:0.02 Legitimate:0.02
Random Forest	Malicious:0.99 Legitimate:0.88	Malicious:1.00 Legitimate:0.97	Malicious:0.01 Legitimate:0.09
SVM	Malicious:0.99 Legitimate:0.87	Malicious:1.00 Legitimate:0.96	Malicious:0.01 Legitimate:0.09

Table 4.21: F1-Score Comparison.

Models	F1-Score in Experiment One	F1-Score in Experiment Two	Difference
Logistic Regression	Malicious:0.88 Legitimate:0.91	Malicious:0.93 Legitimate:0.95	Malicious:0.05 Legitimate:0.04
Naive Bayes	Malicious:0.89 Legitimate:0.92	Malicious:0.94 Legitimate:0.96	Malicious:0.05 Legitimate:0.04
Decision Tree	Malicious:0.90 Legitimate:0.92	Malicious:0.95 Legitimate:0.97	Malicious:0.05 Legitimate:0.05
KNN	Malicious:0.85 Legitimate:0.88	Malicious:0.90 Legitimate:0.93	Malicious:0.05 Legitimate:0.05
Gradient Boosting	Malicious:0.87 Legitimate:0.90	Malicious:0.90 Legitimate:0.93	Malicious:0.03 Legitimate:0.03
Random Forest	Malicious:0.91 Legitimate:0.93	Malicious:0.98 Legitimate:0.98	Malicious:0.07 Legitimate:0.05
SVM	Malicious:0.90 Legitimate:0.93	Malicious:0.96 Legitimate:0.98	Malicious:0.06 Legitimate:0.05

4.4 Discussion

It can be seen from Table 4.2 that the accuracy of the Random Forest model in experiment one is the highest, reaching 92.08%. The accuracy of Naive Bayes, Decision Tree, and SVM all reached more than 90%, and Logistic Regression and Gradient Boosting model are slightly below 90%. The KNN model has the lowest accuracy rate, only 86.77%. In terms of precision, the precisions of legitimate data for all 7 models reached more than 98%. But the precisions of malicious data are not high enough, most models have a precision around 80%, but the lowest KNN model's precision for malicious is 75%. The recall rates show the opposite result, that is, the recall rates of legitimate data are low, and the recall rates of malicious data are high for all 7 models. For F1-score, the F1 scores of most models are around 0.9. The lowest is KNN model, with only 0.85 for malicious data and 0.88 for legitimate data. And the highest is Random Forest model, with 0.91 for malicious data and 0.93 for legitimate data. Overall, using datasets A and B for training can effectively classify the dataset D.

Table 4.10 shows that the Random Forest model has the highest accuracy in the experiment two, reaching 98.07%, while the SVM model has the second highest accuracy, 97.20%, followed by Decision Tree 96.39%, Naive Bayes 94.97%, Logistic Regression 93.83%, and Gradient Boosting 92.00%. The KNN model has the lowest accuracy rate, likewise 91.46%. All models achieved 100% precision for legitimate data and around 100% recall for malicious domain data. The f1 scores are all above 0.9, the highest f1 scores is still for the Random Forest model, with 0.98 for malicious data and 0.98 for legitimate data. And the lowest f1 scores is still for the KNN model.

Through the comparison of the same models in these two experiments, it can be seen that the accuracy differences of Logistic Regression, Naive Bayes, and KNN and Gradient Boosting are all within 5%. While the difference between the two models of Decision Tree, Random Forest and SVM models is greater than 5%. The accuracy of the two KNN models is the lowest, which may be related to the K value and the characteristics of the

experimental data. It is difficult for KNN to use distance measures to reflect the differences between data in these experiments.

Combining the results of the confusion matrix to compare and analyze the precision and recall of the models, it can be found that models in experiment 1 misclassified many domains that should belong to legitimate domains as malicious. This shows that some of the domain names generated by DPS have more similarities with existing legitimate domains than existing malicious domains, which leads to inaccurate classification. As for the difference of F1 scores, the gaps of all models' F1-score are within the acceptable range.

In summary, it can be noted that DPS can generate other domains with characteristics of malicious domains through existing malicious domains.

Chapter 5: Conclusion and Future Work

5.1 Conclusion

In today's digital landscape, the threat of malicious domains to network security has become increasingly prevalent. Many recently registered domain names unknown to blacklist maintainers are potential sources for nefarious activities in the short term. To address this issue, the DPS implements an approach for detecting these domains in advance. The purpose of this project was to evaluate the effectiveness of the DPS in predicting potentially malicious domains.

To assess the feasibility of this method, it was necessary to judge the evaluation metrics of domains generated by DPS through experiments. This was done by building and comparing baseline ML models and corresponding predictive models. First, I trained the predictive models using both existing benign domains and potentially malicious domains generated by the DPS, followed by testing with known malicious and legitimate domains to validate the system's effectiveness. Then I compared these models with baseline models developed by training and testing with existing malicious and legitimate domains. It was found that most of the predictive models achieved acceptable accuracy, at par with the baseline models. This demonstrates that utility of the domains generated by DPS in their potential to be used for malicious activities.

In summary, the DPS can generate domains with characteristics of malicious domains, and in conjunction with other methods, can help anticipate and prevent attacks before they occur. This represents a significant step towards improving network security defense.

5.2 Future Work

Although this project was successful in proving the effectiveness of DPS, some problems with the DPS were found during the experiment, and there is still room for improvement in the DPS.

1. It takes too long to produce predicted domain names. Sometimes it takes one or two days for an input to produce an output.
2. The domain name generated by DPS is highly random, and the output results will be different each time.
3.The accuracy may drop significantly when generating many domains at once.
4. The gaps in performance of some models in these two experiments are slightly larger.

In future research, the prediction algorithm of DPS should be optimized, and the speed of traversing URL and nameserver folders should be improved. Adding the user interface and local storage function allows users to save the results of each operation and access their historical records. In addition, the large amount of data generated by DPS at one time needs to be evaluated separately to determine whether there are unreliable predicted domains in these data. The slightly larger gap between the two experiments' models may be due to the higher outlier sensitivity as outliers are data points that deviate significantly from the majority of the dataset. Therefore, regularization and loss functions should be added when training models [17].

Furthermore, several other technologies can be employed to validate the domain generated by DPS, such as Deep Learning (DL), specifically Convolutional Neural Networks (CNN), or Reinforcement Learning (RL). To utilize RL, one must establish and select appropriate states, actions, rewards, and the environment to create a Markov Decision Process (MDP), then an appropriate algorithm should be chosen.

Bibliography

[1] CIRA, “Q4 2021 CIRA Canadian Shield Insights”, 2021, https://www.cira.ca/resources/cybersecurity/report/cira-canadian-shield-insights-q42021

[2] Akamai, “Flagging 13 Million Malicious Domains in 1 Month with Newly Observed Domains”, 2022, https://www.akamai.com/blog/security-research/newly-observed-domains-discovered-13-million-malicious-domains

[3] A. Almarzooqi, J. Mahmoud, B. Alzaabi, A. Ghebremichael and M. Aldwairi, "Detecting Malicious Domains Using Statistical Internationalized Domain Name Features in Top Level Domains," 2022 14th Annual Undergraduate Research Conference on Applied Computing (URC), Dubai, United Arab Emirates, 2022, pp. 1-6, doi: 10.1109/URC58160.2022.10054226. Chinneck J W. Practical Optimization: A Gentle Introduction. 2022.

[4] I. Ghafir and V. Prenosil, "DNS traffic analysis for malicious domains detection," 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2015, pp. 613-918, doi: 10.1109/SPIN.2015.7095337.

[5] V. Ravi, S. Kp, and P. Poornachandran “Detecting Malicious Domain Names Using Deep Learning Approaches at Scale”. 1 Jan. 2018: 1355 – 1367.

[6] Unit 42, “Newly Registered Domains: Malicious Abuse by Bad Actors”, 2019, https://unit42.paloaltonetworks.com/newly-registered-domains-malicious-abuse-by-bad-actors/

[7] Hara, D., Sakurai, K. and Musashi, Y. “Classification of Malicious Domains by Their LIFETIME”, Advances in Internet, Data and Web Technologies. EIDWT 2020. Lecture Notes on Data Engineering and Communications Technologies, vol 47.

[8] L. M. da Silva, M. R. Silveira, A. M. Cansian and H. K. Kobayashi, "Multiclass Classification of Malicious Domains Using Passive DNS with XGBoost: (Work in Progress)," 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 2020, pp. 1-3, doi: 10.1109/NCA51143.2020.9306705.

[9] H. Zhao, Z. Chang, W. Wang and X. Zeng, "Malicious Domain Names Detection Algorithm Based on Lexical Analysis and Feature Quantification," in IEEE Access, vol. 7, pp. 128990-128999, 2019, doi: 10.1109/ACCESS.2019.2940554.

[10] Gopinath Palaniappan, Sangeetha S, Balaji Rajendran, Sanjay, Shubham Goyal, Bindhumadhava B S, “Malicious Domain Detection Using Machine Learning On Domain Name Features, Host-Based Features and Web-Based Features”, Procedia Computer Science, Volume 171, 2020,Pages 654-661, ISSN 1877-0509.

[11] Robert Lemos, “Key Characteristics of Malicious Domains: Report”, 2021, https://www.darkreading.com/threat-intelligence/research-outs-the-providers-more-likely-to-host-malicious-content

[12] P. Mockapetris, “Domain Names - Implementation and Specification”, 1987, https://datatracker.ietf.org/doc/html/rfc1035

[13] Kaggle, “Alexa Top 1 Million Sites”, 2022, https://www.kaggle.com/datasets/cheedcheed/top1m

[14] PhishTank, https://phishtank.org/

[15] Verma, Rakesh & Das, Avisha. “What's in a URL: Fast Feature Extraction and Malicious URL Detection”. 2017, 55-63. 10.1145/3041008.3041016.

[16] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[17] Jim Frost, “Guidelines for Removing and Handling Outliers in Data”, 2022, https://statisticsbyjim.com/basics/remove-outliers