Universiteit Leiden

Master Computer Science

Ensemble Methods of Malicious Domain Detection using Whois Features and DNS Data Analysis

Name:	Yuang Yuan
Student ID:	s3115631
Date:	28/09/2023
Specialisation:	Data Science
1st supervisor:	Dr. Olga Gadyatskaya
2nd supervisor:	Dr. Yury Zhauniarovich

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Contents

1 Introduction **3**
- 1.1 Motivation . . . 3
- 1.2 Problem statement . . . 3
- 1.3 Contributions . . . 4
- 1.4 Structure of the paper . . . 4

2 Preliminaries **5**
- 2.1 Malicious and benign domains . . . 5
- 2.2 Whois protocol . . . 5
- 2.3 DNS records . . . 7
- 2.4 Malware URL Databases . . . 8
- 2.5 Language models . . . 8
- 2.6 Classifiers . . . 8
- 2.7 Ensemble learning . . . 9

3 Related work **10**

4 Datasets **12**
- 4.1 Dataset . . . 12
  - 4.1.1 Skewed dataset . . . 13
  - 4.1.2 Excessive number of categories . . . 15
- 4.2 Whois data . . . 15
- 4.3 DNS records . . . 16

5 Methodology **17**
- 5.1 Data collecting and preprocessing . . . 17
- 5.2 Category labeler . . . 18
- 5.3 Malicious Domain Classifiers . . . 19
- 5.4 Ensemble methods . . . 19

6 Experiments **21**
- 6.1 Experimental setup . . . 21
- 6.2 Performance evaluations on category labeler . . . 21
- 6.3 Performance evaluations on ensemble system . . . 23
  - 6.3.1 Preliminary experiments . . . 23
  - 6.3.2 Comparison with detection systems . . . 24

7 Discussion **26**

8 Conclusion **27**

Ensemble Methods of Malicious Domain Detection using Whois Features and DNS Data Analysis

Abstract

Cybercrimes involving the use of domains are a prevalent issue, and extensive research has been conducted on the detection of malicious domains. This research first examines the datasets utilized by previous studies and concludes that the common practice of using well-known domain lists, such as Alexa Top N domains, as the ground truth of benign domains is not a good idea because of the bais introduced by the significant differences in the DNS features of well-known domains and less-known newly registered domains. To develop a system capable of detecting malicious domains, even at the early stages of malicious activities, the datasets used in this research comprise newly registered domains with only easy-to-retrieve and publicly available information. The low detection rate in domains in under-represented categories has always been an issue with existing research. We propose an ensemble system to learn category-specific patterns of each type of domain name using different base learners. Thanks to the ensemble approach of malicious domain detection, a higher overall detection accuracy of 0.969 can be achieved compared to a detection system that consists of a single model.

1 Introduction

Among the rising cybersecurity threats, malicious domains created by cybercriminals play a crucial role by hosting fraudulent websites and facilitating malicious activities including command and control operations, phishing, and spear-phishing. Protecting against such domains is vital for safeguarding sensitive information, preventing financial losses, and preserving organizational reputation. Despite the continuous progress in detecting these attacks in general, many alarming problems remain open, such as the weak spot in detecting attacks conducted especially in less commonly spoken languages and in under-represented categories of malicious domains.

1.1 Motivation

Filter lists are commonly used tools that are used for blocking access to known harmful domains or IP addresses that can host malware or exploit kits. URLhaus, Threatfox, and Phishtank are a few of the most famous ones. They are created through crowdsourcing, where people collaborate to identify and generate rules for blocking malicious content.

However, non-English-speaking web users, especially those in languages with fewer speakers or less affluent users, have limited and less well-maintained options for content blocking by the nature of crowd-sourced blocking lists, leading to degraded web experiences and exposure to web maladies that filter lists aim to fix [1].

In the meantime, statistics show there are over 360 million registered domain names [2]. The sheer number of domains on the internet is vast, making it difficult to manually examine each one for malicious activity. This scale makes it impractical to rely solely on human efforts to identify and categorize malicious domains.

1.2 Problem statement

Given that existing detection engines have limitations in terms of accuracy, and robustness, and often rely on manual labeling or signature-based methods that cannot keep up with the rapidly evolving threat landscape, the key problem that this research aims to solve is:

- the difficulty of accurately detecting malicious domains in under-represented categories using solely easy-to-retrieve and public information

To solve this problem, the proposed methods leverage the power of ensemble models and natural language processing techniques to model the complex relationships among DNS and Whois features in and between clusters of domain names used for different purposes. The detection system collects and analyzes various types of DNS and Whois data, constructs an NLP model to cluster domain names, and uses ensemble neural networks to estimate the likelihood of a domain being malicious. The malicious domain classifier has been validated with experiments over real-world DNS and Whois information of newly registered domains and has shown superior performance in especially categories of domains that make the minor proportions in real-life data.

1.3 Contributions

Instead of fighting against malicious cyber activities in a traditional way of putting numerous computational resources and human effort into scanning and analyzing domain names after malicious cyber activities have already taken place, this paper proposed a method that is specialized in:

1. detecting malicious domains in the early stage of malicious activities, and
2. performing well even in detecting the under-represented categories of domains in the training dataset.

1.4 Structure of the paper

The rest of the paper is structured as follows: Section 2 introduces the fundamental concepts in this research. Section 3 discusses the previous work on the topics of malicious domain detection, and concludes the issues with existing research. Section 4 presents the dataset used for model training and testing as well as the data preparation steps taken to generate the datasets. Section 5 introduces our proposed methods of malicious domain detection and provides reasoning on why we built the system like this. Section 6 describes the experimental setup and results. Section 7 discusses the performance of the proposed ensemble system. Section 8 summarizes the thesis, and outlines the future work directions.

2 Preliminaries

2.1 Malicious and benign domains

The classification of domains into malicious and benign categories is essential for analyzing and identifying potential threats in the digital landscape. In this research, **malicious domains** refer to domain names that are associated with malicious activities aimed at compromising the security of Internet users. They are typically created and utilized by threat actors with the intention of deceiving users, exploiting vulnerabilities, and compromising their systems, or stealing sensitive information. On the other hand, **benign domains** are considered safe and trustworthy. They are associated with legitimate websites, businesses, organizations, or individuals who adhere to ethical practices and do not engage in any form of cybercrime. Benign domains have a relatively low chance of being misreported as malicious domains while malicious domain names are very likely to appear in phishing activity databases such as PhishTank [3] and URLhaus [4].

2.2 Whois protocol

The Whois protocol was originally designed to allow users to retrieve information about registered domain names, IP addresses, and other network resources. Three features of the Whois protocol add extra effort to exploiting Whois data for identifying malicious domains. First, the protocol does not define a specific format for the contents of the response due to historical reasons that there was an expectation that the information returned would be in a human-readable format when the protocol was developed [5]. Second, the Whois data ultimately comes from multiple sources called "registrars" and "registries" which makes the entries, format, and even languages of the Whois query output between "registrars" and "registries" varying and ever-changing [6]. For instance, an upgrade of the Whois server could lead to a different format of query results. That being said, there is no way to know all possible output formats from a Whois query. Therefore, the collected "raw Whois data" should be parsed in a way before we can exploit the information in each entry to predict if a domain is malicious. Whois features that could be used to build the malicious classifier are displayed in Table 1.

Table 1: The name, type, and description of each Whois feature. The "Data Type" Column shows the content and type of corresponding data. Detailed description regarding each feature is given in the "Description" Column.

Feature	Data Type	Description
Region	Including: Country, State, City. Unstructured, Natural language data.	Domain holders are obligated to provide address information by registrars when registering a domain. The provided information is the data source of Whois servers. Note that domain holders are free to decide the format of the input [7]. It means when referring to the United States of America, the information provided by domain holders could be all variants of the country name including US, USA, United States, the United States of America, and more. Besides, the address information could be redacted for privacy [7] by registrars upon the request of domain holders' requests. It is also worth noting that domain holders may also provide false information, so the address information may not be true.
Registrars / Registries	Integer data after being assigned with IANA Number.	Registrar refers to the company or organization that an individual or entity uses to register a domain name. It acts as an intermediary between domain registrants (owners) and the domain registry which facilitates the registration process, manages the domain settings, and handles administrative tasks related to the domain name. A registry is the authoritative entity responsible for managing the top-level domain (TLD) extensions, such as ".com," ".org," ".net," etc. The registry maintains the central database of domain names within the TLD and controls the distribution and management of these domain names to accredited registrars. The "Registrars/Registries" field in a WHOIS query response typically lists the names and contact information of the domain registrar and the domain registry associated with the queried domain name. Some registrars offer domain drop catching which also referred to as domain sniping, involves the act of swiftly registering a domain name immediately after its registration has expired.

Continued on next page

Feature	Data Type	Description
Domain Status	Text data with limited number of status.	In a WHOIS query response, the "Domain Status" field provides information about the current status of the domain name within the Domain Name System (DNS). This field indicates whether the domain is active, pending, expired, or undergoing any other specific state that might affect its functionality or availability. Each domain status is typically represented by a code or keyword that conveys a specific meaning.
DNSSEC Status	Text data with limited number of status.	DNSSEC stands for "Domain Name System Security Extensions." It is a suite of cryptographic protocols and security extensions designed to enhance the security of the Domain Name System (DNS).
Time	Including: Creation, Expiration, Modifaction time. Time data. Cyclical data.	This field of the query results provides information about the specific dates and times associated with the creation and expiration of a domain name registration. It offers essential details about the lifecycle of the domain, showing when the domain was initially registered, when its registration is set to expire and more.

2.3 DNS records

The Domain Name System (DNS) is a crucial component of the internet infrastructure that translates domain names into IP addresses, allowing users to access websites and other online resources. DNS records are the data stored in DNS servers, containing information about domain names, such as IP addresses, mail server addresses, and other important details.

DNS data, especially, DNS records, is a valuable data source for extracting informative features for malicious domain analysis. They are publicly accessible and not protected by privacy regulations, allowing researchers to extract information without constraints. DNS records contain patterns that help identify malicious domains, such as ownership details or known malicious activities. DNS records are easy to retrieve and analyze using available tools and APIs. Additionally, they provide rich contextual information like record changes and multiple records for a single domain, aiding in the detection of malicious intent.

DNS records that may be related to malicious activity detection are shown in Table 2.

Table 2: Description of DNS Record Types for Malicious Domain Analysis.

Record Type	Description
A/AAAA	Maps a domain name to an IPv4/IPv6 address, allowing identification of the server hosting the domain and potential malicious IP addresses.
CNAME	Creates an alias for a domain, allowing malicious actors to redirect traffic and hide their activities by pointing to different domains.
NS	Specifies the authoritative DNS server for a domain, assisting in identifying the infrastructure associated with malicious domains.
TXT	Stores text-based information, often used for domain ownership verification, security policies, and signatures of known malicious activities.
SRV	Specifies the location of a service within a domain, providing insights into potential malicious service configurations.
PTR	Performs a reverse DNS lookup, mapping an IP address to a domain name, assisting in identifying the domains associated with malicious IP addresses.
MX	Specifies the mail server responsible for handling emails for a domain, aiding in the detection of malicious email activities.
SOA	Stores information about the start of authority for a domain, including details such as the primary DNS server and contact information, potentially revealing malicious infrastructure.
AXFR	Allows zone transfers between DNS servers, which can be exploited by attackers to gather information about a domain's DNS configuration.
CAA	Specifies which certificate authorities (CAs) are authorized to issue certificates for a domain, helping prevent malicious certificate issuance.

2.4 Malware URL Databases

URLhaus [4], known for its focus on malware distribution, provided insights into whether the domains were associated with any malicious activities. Phishtank [3], on the other hand, specializes in detecting and reporting phishing websites, offering valuable information on potential threats related to the domains. These two databases are a good data source for domain name labeling.

2.5 Language models

BERT (Bidirectional Encoder Representations from Transformers) [8] is a pre-trained language model developed by Google that is widely used for natural language processing tasks such as question answering and sentiment analysis. Multilingual BERT (M-BERT) is a variation of BERT that has a robust ability to generalize texts cross-lingually. It is pre-trained on a large corpus of unannotated text from 104 different languages to learn a shared multilingual representation of language [9]. This model can be fine-tuned for specific tasks in different languages, allowing for effective transfer of information across languages.

2.6 Classifiers

- **MLP (Multilayer Perceptron)**
  The Multilayer Perceptron [10] is a type of artificial neural network (ANN) that

consists of multiple layers of interconnected nodes called neurons and can effectively handle complex patterns and non-linear relationships.

- **Decision Trees**
  Decision Trees are a popular machine learning algorithm used for both classification and regression tasks [11]. They recursively partition the data based on different features to create a tree-like model. Each internal node represents a feature, and each leaf node represents a class or a value. Decision trees offer interpretability and the ability to handle categorical and numerical features. They can efficiently split data such as domain names based on Whois and DNS features.
- **XGBoost**
  XGBoost (Extreme Gradient Boosting) [12] is an ensemble learning algorithm that combines the power of decision trees with gradient boosting. It can handle a large number of features and effectively capture complex relationships and interactions.
- **TabNet**
  TabNet [13] is a deep learning model specifically designed for tabular data analysis. It excels in handling structured data. TabNet's ability to learn feature interactions and select relevant features through a sparse attention mechanism makes it suitable for detecting malicious domains based on structured Whois and DNS datasets.

2.7 Ensemble learning

Ensemble learning [14] is a machine learning technique that combines the predictions of multiple individual models (known as base models or weak learners) to make more accurate and robust predictions. The idea behind ensemble learning is that by aggregating the predictions of multiple models, the weaknesses and errors of individual models can be mitigated, leading to improved overall performance.

When it comes to detecting malicious domains, ensemble learning can be effective, especially when considering domain names associated with similar topics that exhibit similarity in terms of Whois and DNS features. In the meanwhile, domain names used for different purpose introduce noise or outliers that might impact the performance of a detection model because the difference in different types of domain names. Ensemble learning helps mitigate the effect of such noise or outliers by combining the predictions of multiple models, reducing the impact of individual model errors. By using an ensemble of models, each trained on different subsets of the data, the ensemble can capture a broader range of patterns and increase the chances of accurately detecting malicious domains.

3 Related work

The field of malicious domain detection has seen significant advancements in recent years. Prominent categories of approaches include rule-based detection, machine learning-based detection, and the combinations of these two [15]. While rule-based methods rely on human expertise to identify patterns and characteristics of malicious domains, machine learning-based methods leverage computational algorithms to detect malicious domains without requiring extensive pre-defined rules. Additionally, researchers have recognized the importance of addressing challenges in deploying malicious domain detection technologies in real-life scenarios. One such challenge pertains to the selection of a representative benign dataset for training and testing models. It is crucial to ensure that the dataset accurately represents legitimate domains while avoiding potential biases or vulnerabilities. Furthermore, the availability and accessibility of data is also important. Obtaining hard-to-retrieve data, such as passive DNS datasets, may limit the scalability of the detection methods. Utilizing publicly available information for training models while reducing the reliance on hard-to-retrieve data is the way out of this situation.

- **Rule-based Malicious Domain Detection**
  Early research uses human expertise to identify patterns and characteristics of malicious domains and then uses this knowledge to detect new malicious domains [16]. It's observed that malicious domains belonging to one malware family tend to be queried simultaneously, and by measuring the degree of co-occurrences between known malicious and unknown domains, it is possible to detect new malicious domains [17, 18]. These observations can be used to develop rules or thresholds for identifying malicious domains based on their DNS query patterns. Note that adversaries have the ability to adjust their actions over time which may lead to a decline in the detection capabilities of these methods. Consequently, knowledge-based methods become outdated as attackers refine their techniques. Also, the development and upkeep of knowledge-based methods often demand a substantial amount of human expertise and effort, making it a resource-intensive endeavor [15].

- **Machine Learning-based Malicious Domain Detection**
  Machine learning-based detection methods do not require extensive knowledge of patterns of malicious domains and are able to evolve to adapt to new techniques thus are the current most popular methods [15].
  Clustering and classification are two main approaches of machine learning-based malicious domain detection.
  Deepdom [19] is a cutting-edge clustering-based detection method achieving a high accuracy of 0.9791. It clusters domains that may involved in malicious activities with a heterogeneous information network and a spatial-based graph convolutional network (GCN) method called SHetGCN. They collect DNS-related data, construct a HIN to model the DNS scene, extract meta-paths to demonstrate associations among domains and use SHetGCN to estimate whether a domain is benign or not. However, its association-based approach makes it hard to detect new types of malicious domains. Moreover, this method relies on passive DNS datasets which are not

easily accessible so more difficult to deploy in real-life applications. Similar research [20, 21] relying on hard-to-retrieve data like passive DNS datasets, DNS traffic among top-level-domain servers also suffer for the same reasons.
Instead of using private data for training clustering models, the content of a domain like the lexical features of malicious URLs and the content of the webpage are also useful features for malicious domain detection [22]. This method proposed by Saleem Raja et al. utilizes the text information to detect malicious activities and successfully reduces the computational overhead by using unsupervised learning models and still yields high accuracy in detecting malicious domains, the detecting targets of these systems are limited to mostly phishing domains which do have webpages.
These methods usually take features of a domain that are obtained from DNS query, Whois data, and more as input. The lists of domain names that are used for training and testing are therefore critical for the performance of the classifier. Those lists are often obtained either by researchers from personal knowledge or crowd-sourcing [23] which both show unreliability to some extent [24].

- **Issues in creating test dataset using well-known domain names lists**
  Combosquatting is a specific type of domain squatting in which attackers register domains that combine a popular trademark with one or more phrases [25]. While most research uses Alex Top List to make their list of domains for model training or testing, as study [25] suggested, well-known domains, especially those in the Alexa Top list may not be appropriate candidates for malicious domain analysis datasets because of combo-squatting. Creating test datasets for training a model would lead to low detection accuracy, especially in this spectrum of malicious activities.

- **Issues with training model with hard-to-retrieve data**
  A possible workaround for solving the issues of testing models with well-known domain names is making a domain list for model training based on collected real-life DNS queries received by DNS servers [19, 26]. However, DNS traffic logs of a DNS server can contain personal data, such as IP addresses, which are considered personal identifiers and subject to GDPR's privacy regulations [27]. Therefore, it couldn't be considered a universal solution since DNS logs data are hard to retrieve, especially in the EU region.

The field of malicious domain detection has witnessed significant progress encompassing rule-based, machine learning-based, and hybrid methods. In the meantime, most of the existing solutions still face issues with data retrieval, privacy regulations, and low detection rates in certain classes of domains. Our proposed method prioritizes publicly available data for training, sidestepping data retrieval issues and privacy concerns, aiming to enhance the accuracy of detecting malicious domains in under-represented categories. A more in-depth overview of the related work can be found in [28–30].

4 Datasets

One major goal of this research is to build a system specialized in detecting and even predicting potential malicious activities in the very early stage. Thus, in this research, the criteria for data used for training and testing the models are refined as:

1. must be public information that is not under privacy protection regulations, contains malicious patterns, and can be easily retrieved at any stage of the life cycle of a malicious domain.
2. preferred to be newly registered domains that are collected as inclusively as possible.

The following subsections will introduce which and how data was collected and processed in detail.

4.1 Dataset

Whois information and DNS records are valuable for analyzing malicious domains as they provide crucial details about the domain owner, registration history, and associated infrastructure. This data helps with validating the legitimacy of a domain, identifying patterns of malicious behavior, and tracking the activities of potential threat actors. By analyzing Whois and DNS records, security professionals can gain insights into the reputation, history, and potential threats associated with a domain, enabling them to take appropriate actions to protect users and strengthen cybersecurity measures. And, importantly, both are public, non-confidential, and can be easily retrieved using open-source tools.

To build the domain list that is used to collect DNS/Whois information based on, this research mainly focuses on domains with .com as TLD and especially includes newly registered domains that have been created a month or so since the experiments were conducted.

Shallalist webpage [31] is an enhanced dataset based on a widely used and comprehensive blacklist, Shallalist, for web filtering purposes. This dataset was originally designed to help organizations and individuals enforce internet usage policies and maintain a safe browsing environment by providing information to block or restrict access to these specific categories of websites. Besides human-curated categorized data on various types of websites (such as adult content, gambling, drugs, violence, and more), it also includes crawled HTML webpage data for each domain name which is an excellent data source for training an NLP-based domain category classifier.

URLhaus and Phishtank were used to label the newly registered domain list into malicious and benign categories. The list of domains was cross-referenced with the extensive databases of both tools to determine their categorizations. By combining the information provided by URLhaus and Phishtank, the domains were accurately labeled as either malicious or benign which ensured an accurate evaluation of the newly registered domains.

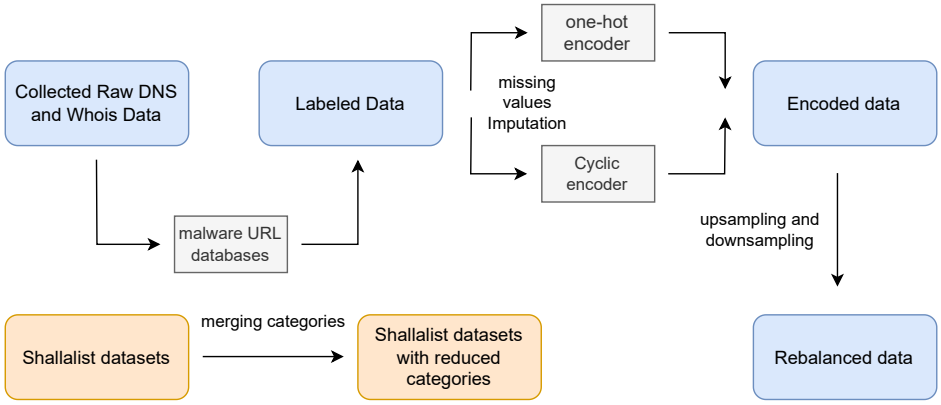
The dataset used in the research is displayed in Table 3.

Table 3: Data that are used in the research.

Data Source	Count	Benign-Malicious Ratio
newly registered .com domains created between 01/04/2023 and 30/04/2023	3,099,614	5047.23 : 1
Alexa Top 500 domains	500	$+\infty$
Shallalist Webpage	837,070	Unknown

Figure 1 presents data preprocessing pipelines that show the steps taken to prepare datasets for detection model training and testing. First, domain names will be labeled as either malicious or benign domains based on malware URL databases. Second, collected data will be coded with encoders after which rebalancing strategies will be applied to the data to yield a more balanced dataset. Categories of Shallalist datasets will be reduced to an amount that is suitable and feasible for model training with limited computational recourses. The following subsections will discuss how and why data are processed in each step.

Fig. 1: Data preprocessing pipelines.



4.1.1 Skewed dataset

Working with imbalanced datasets can lead to biased models that prioritize the majority class, resulting in poor performance when it comes to detecting the minority class. In our case, the vast majority of domains are benign, making it difficult for the model to learn and identify malicious domains effectively. Therefore, it is crucial to implement appropriate preprocessing techniques to mitigate this class imbalance. Several preprocessing strategies can be employed to tackle the class imbalance in the dataset. These strategies aim to either oversample the minority class (malicious domains) or undersample the majority class (benign domains). The most commonly used strategies include [32]:

- **Random Undersampling**: This technique involves randomly removing instances from the majority class to achieve a more balanced dataset. However, it comes with a risk of losing important information present in the majority class, potentially leading to decreased overall performance.
- **Random Oversampling**: In this technique, additional instances are randomly duplicated from the minority class to match the number of instances in the majority class. While oversampling can help in addressing the class imbalance by providing more samples for the minority class, it also runs the risk of overfitting and introducing redundant information.
- **Adaptive Downsampling**: This strategy involves intelligently selecting instances from the majority class based on specific criteria or algorithms. It aims to retain representative instances while reducing the number of majority-class samples. Adaptive downsampling helps mitigate the risk of losing crucial information while addressing the class imbalance.
- **Adaptive Upsampling**: This technique focuses on generating synthetic instances for the minority class rather than duplicating existing instances. It employs algorithms such as SMOTE (Synthetic Minority Over-sampling Technique) to create new instances that are similar to the existing minority class samples. Adaptive upsampling helps in increasing the diversity of the minority class without introducing redundancy.

Considering the number of domain names is as large as 3 million, datasets that provides a sufficient representation of both malicious and benign domains are made by combining both downsampling and upsampling strategies.

To ensure the fairness of the testing dataset, it is crucial to properly separate the original dataset into two different groups before applying any up or down sampling techniques. This separation is done prior to any modifications to the dataset to avoid introducing bias in the testing process. By separating the original dataset into a training group and a testing group beforehand, we can ensure that the testing dataset remains representative of the original distribution. Any modifications or preprocessing steps are only applied to the training dataset. This ensures that the testing dataset is independent and unbiased, allowing for a fair evaluation of the trained model's performance.

Note that the primary objective is to create a more balanced distribution of data between the classes. However, achieving an exact balance is not always necessary or even desirable. Having an equal number of data points for each category can lead to a loss of information and can introduce bias in the model. We strike a balance between providing enough data for the minority class to improve its representation, while still maintaining the overall distribution of the original dataset.

Therefore, after the rebalance of datasets, the training dataset includes 54,958 malicious records and 193,697 benign records while the testing dataset includes 84 malicious records and 534 benign records.

4.1.2 Excessive number of categories

Shallalist Webpage category domains into 74 classes which is an excessive number for making an ensemble system. To reduce the number of classes, categories are merged into larger and more general categories. We consider similarities and overlaps between the categories to ensure that the merged categories still capture the essence of the original categories. The distribution of different categories in the dataset after the merge is shown in Table 4.

Table 4: Distribution of domain names in each category in Shallalist

Category	Counts	Proportion
miscellaneous	385,786	46.1%
socialmedia	315,487	37.7%
sexuality	83,714	10.0%
finance	28,982	3.5%
technology	23,101	2.7%

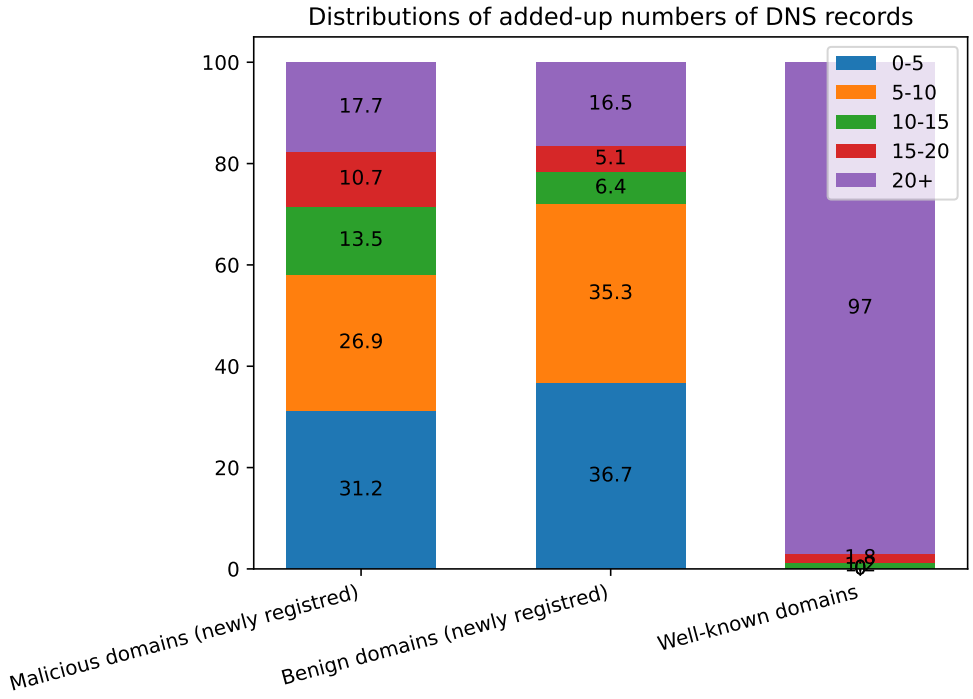
4.2 Whois data

All of the Whois data listed in Section 2.2 are collected for detection model training and testing. Collected Whois query results are parsed in a way that the most common fields in the results are extracted to make a uniform Whois database. Considering not all Whois query results contain information regarding every predefined feature, missing data are marked and categorized into a new class.

4.3 DNS records

All of the 11 types of DNS records mentioned in Section 2.3 are collected for detection model training and testing. A huge difference in the number of records of domain names can be noticed in the Alexa Top 10000 domains (benign domains), legit newly registered ".com" domains. In the meantime, much less difference between legit and malicious newly registered ".com" domains was observed. As Figure 2 shows, a large proportion (97%) of domain names in the group of well-known domains (Alexa Top 500) have more than 20 records in total while both groups of newly registered domains show more even distributions in the number of records. It indicates that training and testing a detection model using well-known domains as the ground truth of benign domains may lead to biased results.

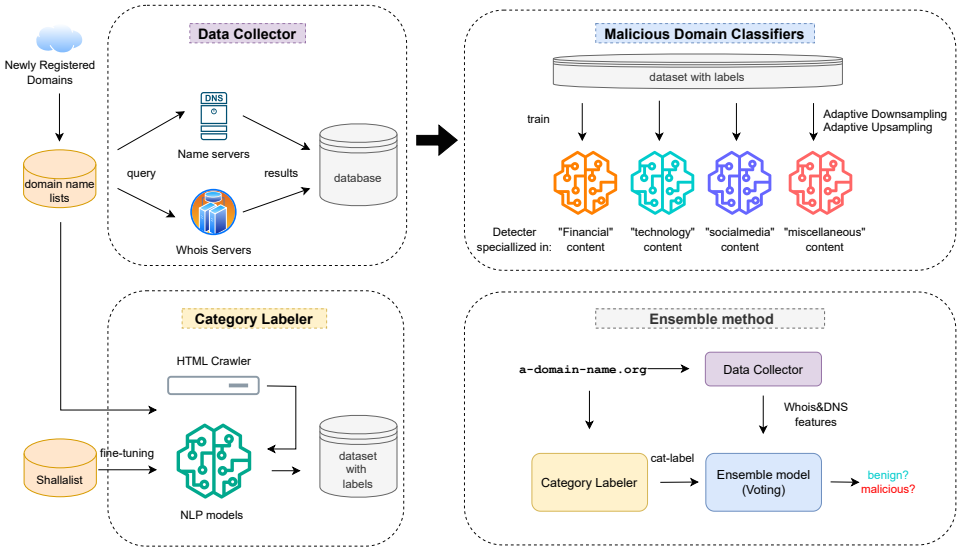
Fig. 2: Distributions of added-up numbers of DNS records. Each color in the chart indicates a range of added-up numbers of DNS records in each domain name group. The numbers shown on each bar refer to the proportion of the range on the scale of 100 in the group.



5 Methodology

Figure 3 illustrates the system of ensemble malicious domain classifiers, which consists of a data collector, category labeler, malicious domain classifiers, and an ensemble structure. Firstly, the Whois and DNS features of the domains on lists are obtained from dedicated Whois servers and Name servers respectively after which these data undergo preprocessing to ensure a consistent structure before being stored in a database. An NLP domain name category labeler is then used for labeling the domains on lists into predefined classes. When the Whois and DNS features dataset with category labels is ready, each base learner is trained with corresponding adaptive downsampled and upsampled datasets to learn the relevant information and pattern of the class. Overall, the ensemble methods are made up of two classifiers that first classify the target domain name into a class and then make a prediction using the ensemble malicious domain detection model in a voting manner. The following sections will give explanations of each component in the system in detail.

Fig. 3: System architecture of the ensemble malicious domain detection method



5.1 Data collecting and preprocessing

Whois features listed in Table 1 and DNS features listed in Table 2 are collected from Whois servers and name servers separately. Before collected Whois and DNS data are stored in the database, they need to go through a data preprocessing pipeline.

The first step is to imputing missing values for categorical variables by assigning a new category value "missing". This way missingness is explicitly accounted for, preventing any unintended bias or distortion in subsequent analyses.

Then most Whois and DNS features are encoded with a one-hot encoder to represent categorical variables. During the process, each categorical feature is transformed into a binary vector representation. This encoding technique converts each unique value in the WHOIS features into a separate binary feature. For example, a WHOIS feature that has three unique values (A, B, C) will be transformed into three binary features (is_A, is_B, is_C). If a particular sample has a specific value for a WHOIS feature, the corresponding binary feature is set to 1, indicating its presence. Conversely, if the sample does not have that value, the binary feature is set to 0. This encoding approach ensures that the WHOIS features are represented in a format that can be effectively utilized by machine learning algorithms.

When it comes to encoding time features, a cyclical encoding approach is used to preserve the cyclic nature of this feature. The time variables (time, day, month, year) are transformed into two new features - sine and cosine transformations of the time variable. These two features will have values between -1 and 1, representing the cyclical nature of time, and are computed as:

$$\text{Encoded\_Time\_sin} = \sin\left(\frac{2\pi \times \text{Time\_Feature}}{\text{Max\_Value\_of\_Time\_Feature}}\right)$$

$$\text{Encoded\_Time\_cos} = \cos\left(\frac{2\pi \times \text{Time\_Feature}}{\text{Max\_Value\_of\_Time\_Feature}}\right)$$

where "Time_Feature" represents the original time value, and "Max_Value_of_Time_Feature" represents the maximum value of the time feature. The resulting "Encoded_Time_sin" and "Encoded_Time_cos" variables are used as features in the classifier training.

5.2 Category labeler

The task of the category labeler is to classify domain names into 6 predefined classes so that each base learner of ensemble malicious domain detection models can be trained and tuned with corresponding adaptive sampled datasets as aforementioned in section 4.1.2. It is a super challenging task for several reasons:

1. While domain names are a good source of data that can be used for domain classification analysis, domain names themselves can be highly ambiguous, with multiple possible interpretations. For example, a domain name like "applestore.com" could refer to an online store selling apple products or a website dedicated to apple farming.
2. A domain name could consist of either multiple or single words in one single language, multiple languages or even non-human language at all, which adds extra complexity to the architecture of the classifier.
3. As a necessary step of domain name feature engineering, word segmentation is a hard task when dealing with certain languages. A domain name like "donga.com" could be segmented into either "don", "ga", ".com" (could be interpreted as a "dry valley" in Japanese), "do", "nga", ".com" (refers to "many crows" in Cantonese) or "dong", "a", ".com" (means "east Asia" in Korean).

Overall, this ambiguity complicates the classification task, as the classifier needs to identify categories based on very limited information. Category labeler solves the issues with sophisticated feature engineering strategies and cutting-edge NLP models.

Two textual features of a domain name are used for building a category labeler:

1. **Domain Name**: It often provides valuable insights into the nature and purpose of a website. This feature may contain keywords or phrases that indicate the content or industry associated with the domain. For example, a domain name includes words like "mitsubishi," "tesla," or "motor" can provide strong hints that the domain is associated with automobiles. By considering the domain name, we can leverage this valuable information to make accurate categorization decisions.
2. **HTML content**: The HTML content of a website can offer further context and clues about its category. The HTML structure, meta tags, page titles, headings, and textual content within the web pages can contain relevant keywords, descriptions, or specific terms that indicate the website's purpose or content focus. Even if the existence of HTML content of a domain name itself also delivers informative signals regarding the purpose of the domain name is used for. Besides, the feature is relatively easy to retrieve from a given domain.

As shown in Figure 3, domain names and HTML content are first tokenized using a multilingual BERT tokenizer. In the stage of data preprocessing, words are broken down into smaller subwords using WordPiece [33] techniques to make handling a wide range of languages possible and improve the model's ability to capture morphological variations. Next, the multilingual BERT is fine-tuned with Shallalist datasets to learn domain category-specific patterns and nuances.

5.3 Malicious Domain Classifiers

After domain names are labeled with their classes as discussed in Section 5.1, the Whois and DNS features database is further adaptively upsampled to yield new 6 different datasets in which domain names associated with each domain category make a higher proportion. The malicious domain classifiers which are later used as base learners of the ensemble model are trained with the tokenized features in different datasets separately to learn special patterns in different categories. Four commonly used classification models that are mentioned in 2.6 are trained, tested, and compared with each other to identify the most suitable model for the task.

5.4 Ensemble methods

The ensemble model is responsible for making predictions about the maliciousness of a given domain name based on the the predictions of base learner and the label of the domain name.

When a new domain name is inputted, it first goes through the classifying classifier. This classifier evaluates the domain name using the learned information and patterns and assigns it to a specific class as described in Section 5.2. The output of the classifying classifier is the predicted class label for the domain name. Next, the ensemble malicious domain detection model comes into play. It makes predictions based

on the outputs of the domain category-specific base learners to achieve higher accuracy and robustness in identifying a wide range of malicious domains. Specifically, the base learner with a matched category label to the targeted domain name is assigned a weight of 1, and the rest of base learners are assigned a weight of 0 when making a prediction.

6 Experiments

This section focuses on evaluating the effectiveness of category labelers, analyzing the performance of the ensemble system, and comparing the proposed system with other detection systems.

6.1 Experimental setup

The ensemble system is implemented in Python 3.11.3 with scikit-learn [34], an open-source machine-learning library that provides a wide range of tools for data preprocessing and model evaluation. Metrics used for evaluation of the performance of the ensemble system are listed in Table 5. Datasets fo newly registered domains created from 01/04/2023 to 30/04/2023 are used for training and testing models, and the preprocessing and train/test splitting strategies applied in the experiments as discussed in Section 4.

Table 5: Description of metrics used in the experiments.

Metric	Description
True Positive (TP)	The number of positive instances correctly classified as positive by the model.
False Positive (FP)	The number of negative instances incorrectly classified as positive by the model.
True Negative (TN)	The number of negative instances correctly classified as negative by the model.
False Negative (FN)	The number of positive instances incorrectly classified as negative by the model.
Accuracy	It is the proportion of correctly classified instances (both positive and negative) out of the total number of instances. It is calculated as $(TP+TN)/(TP+FP+TN+FN)$.
Precision	Also known as positive predictive value, it is the proportion of correctly classified positive instances out of the total instances predicted as positive. It is calculated as $TP/(TP+FP)$.
Recall	Also known as sensitivity or true positive rate, it is the proportion of correctly classified positive instances out of the total actual positive instances. It is calculated as $TP/(TP+FN)$.
F1 Score	It is the harmonic mean of precision and recall. It provides a balanced evaluation measure that considers both precision and recall. It is calculated as $2 \cdot (precision \cdot recall)/(precision + recall)$.

6.2 Performance evaluations on category labeler

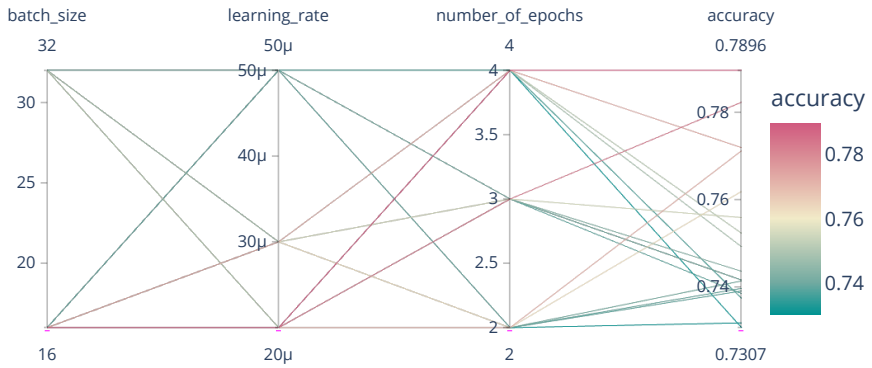
As mentioned in Section 4.1, the Shallalist dataset is used for fine-tuning the BERT-based category labeler. Specifically, `BERT-base-multilingual-uncased`, a multilingual language model that is case-insensitive and serves as a base model for a range of natural language processing tasks across multiple languages, is selected for the classification task. Grid search strategy is applied to optimize the hyperparameters configuration. We further limit the search space to the range suggested by Devlin

et al. [8] which is expected to perform well across all kinds of tasks. Combinations of the following hyperparameters are tested in the experiments:

- Batch size: 16, 32
- Learning rate (Adam): $5e-5$, $3e-5$, $2e-5$
- Number of epochs: 2, 3, 4

As far as hyperparameter sensitivities, Figure 4 displays the accuracy of fine-tuned BERT-based category labeler under each combination of hyperparameters. The figure shows that batch size and learning rate have more impact on the model performance compared with the number of epochs. Also, the combination of smaller batch size and lower learning rate yields higher accuracy. Out of the 18 combinations, the category labeler tuned with a batch size of 16, a learning rate of 2e-5, and 4 epochs produces the highest accuracy of 0.7896 which shows the category labeler can effectively classify domain names by exploiting public textural information of domains.

Fig. 4: Performance of the BERT-based category labeler fined-tuned under each combo of hyperparameters in parallel coordinates



Domain names in the datasets are labeled with this optimized category labeler to allow base learners in the ensemble system to learn the special patterns in each domain category. The distribution of each category in the labeled datasets is listed in Table 6.

Table 6: Predicted distribution of domain names in each category

Category	Counts	Proportion
miscellaneous	1,286,339	41.5%
socialmedia	827,597	26.7%
sexuality	409,149	13.0%
finance	362,654	11.7%
technology	220,072	7.1%

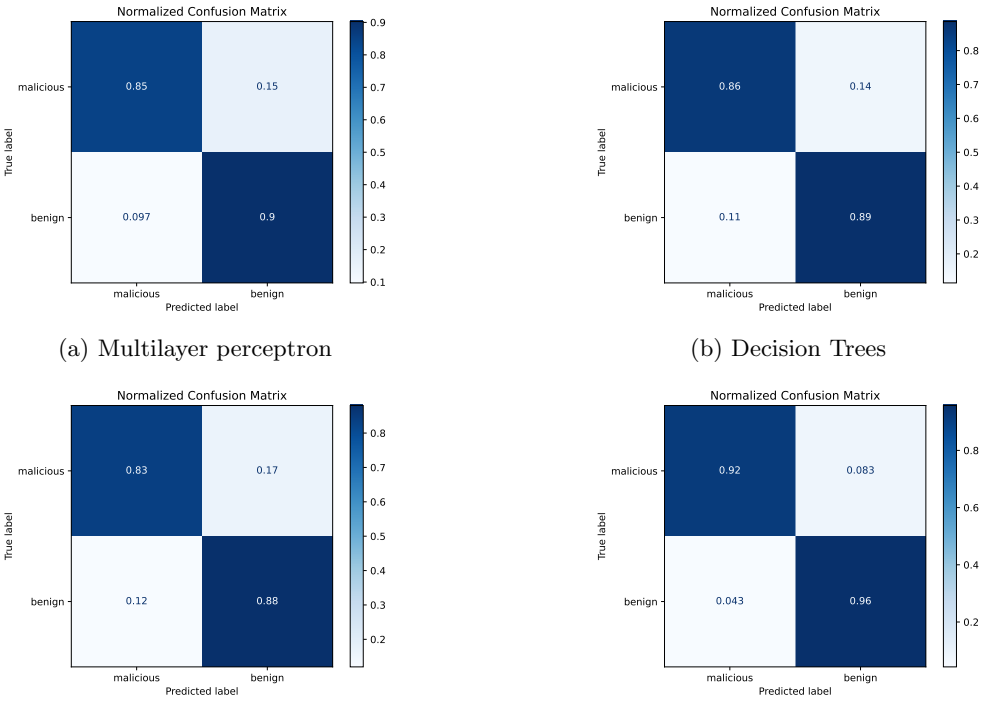
6.3 Performance evaluations on ensemble system

6.3.1 Preliminary experiments

A series of preliminary experiments are conducted to examine the performance of different machine learning models before we make choices on the cores of base learners in the ensemble system. Specifically, we test four models mentioned in Section 2.6. For a fair comparison, hyperparameter optimizations are performed for Multilayer perceptron, decision trees, XGBoost, and TabNet, and the results of the experiments are shown in Table 7. Confusion matrices of each model are displayed in Figure 5.

Table 7: Performance comparison of different models. The highest score for each metric is bolded.

Model	Accuracy	Precision	Recall	F1 Score
TabNet	**0.951**	**0.986**	**0.957**	**0.971**
MLP	0.895	0.974	0.903	0.937
XGBoost	0.874	0.971	0.880	0.923
Decision Trees	0.882	0.975	0.886	0.928



(a) Multilayer perceptron

(b) Decision Trees

(c) XGBoost

(d) TabNet

Fig. 5: Confusion Matrices for the results of four different models

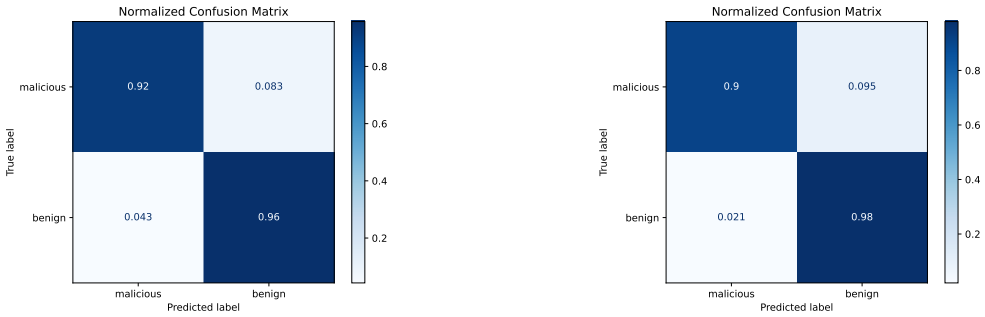
The result shows that TabNet outperformed the other three models in all of the four metrics by a large margin. Therefore, TabNet is used for comparing different detection systems.

6.3.2 Comparison with detection systems

The performance of malicious detection for the ensemble system is evaluated and compared with a "vanilla" system in which ensembling methods are not introduced and detection models are trained on datasets without category labels. As mentioned in Section 6.3.1, TabNet is used as the detection model for both systems. The confusion matrices and scores of each system are shown in Figure 6 and Table 8 separately.

System	Accuracy	Precision	Recall	F1 Score
"vanilla"	0.951	**0.986**	0.957	0.971
ensemble	**0.969**	0.985	**0.979**	**0.982**

Table 8: Performance comparison between the "vanilla" system and ensemble system. The highest score for each metric is bolded.



(a) The "vanilla" system

(b) Ensemble system

Fig. 6: Confusion Matrices for the results of the "vanilla" system and ensemble system

The results show that the ensemble system outperformed the "vanilla" system in aspects of accuracy, recall, and F1 score, but produced a slightly lower precision. The ensemble system achieves a significantly lower false positive rate on benign domain names while maintaining nearly exact accuracy on malicious domain names which indicates an overall better performance. From the experimental results, we can conclude that the ensemble malicious domain detection system is able to achieve a good detection rate on newly registered domain names.

7 Discussion

We have shown the possibility of accurately labeling domain names into predefined categories with textual information that can be easily and quickly extracted using large language models like BERT. However, the choice of large language models could be debated. The base version of BERT used in the system is a minimal BERT that has 110 million parameters, while the large version of BERT has 340 million parameters. Thus, the base version takes less computational resources but is less likely to perform as well as the large version in terms of the ability to capture more fine-grained semantic and syntactic information which may have a significant impact on tasks such as text classification. Besides, we categorize domain names based on only textual information due to the constraints of computational resources, and the lack of associated HTML content in the collection of newly registered domain names indicates that many domains are categorized on the information delivered by domain names only. This further limited the accuracy of the category labeler. Expanding the list of features that are used for categorizing domains could have a positive impact on the performance of the category labeler.

Experiments show the high reliability of the ensemble system in detecting malicious domain names without requiring any private and sensitive data. Compared with the "vanilla" system, the ensemble system has a better overall performance and is especially better at learning the pattern of benign domain names. It means that the ensemble system made a positive impact on the detection rate by learning different patterns in categorized domains. This observation suggests that the proposed method of detecting malicious domains with specialized models on categorized domain names can be applied to other existing models to further improve the detection rates. However, it is still not clear why additional label information did not help to improve the detection rate of malicious domain names. The limited size of test datasets and the difficulty of labeling malicious domains from collected newly registered domain names could be a possible reason.

As for the comparisons with similar research, Deepdom [19] achieved an overall accuracy of 0.9791 on a real-world dataset collected from DNS traffic in CERNET2 while the extreme learning methods proposed by Shi et al. [26] is able to achieve a detection rate of 0.9628 on the DNS data collected from the DNS servers of Shanghai Jiaotong University. Our proposed methods made a similar overall accuracy of 0.9691 on the DNS and Whois data of newly registered domains. However, our research has slightly different objectives from existing work. First, we focus on detecting malicious domains from newly registered malicious domains. Second, we are more aimed to propose a way to further enhance existing detection systems by improving the detection rate of underrepresented domains. Those two factors make it hard to make a fair and direct comparison with similar work.

8 Conclusion

Detecting malicious domain names in their early stages using only public information with an ensemble system that contains different category-specific base learners is a promising research direction.

The experimental results show that the category labeler can effectively label domain names based on textual information. Considering the informative data imaginary data of domain names contained, this feature could be introduced to further enhance the category labeler given sufficient computational resources in the future. Furthermore, the performance of the category labeler could be improved with other variations of BERT. Roberta, introduced in the paper by Liu et al. (2019) [35], is an enhanced variation of BERT. It is trained on a larger corpus of text data from the internet and for a longer duration. As a result, Roberta has the potential to outperform the base version of BERT in tasks involving the understanding of HTML content, as demonstrated in the experiment. With this larger language model, the category labeler could achieve a better accuracy score, especially for the domain names associated with websites used for phishing purposes. Additionally, the predefined categories of category labelers could be expanded by considering both the type of content and the language associated with the domain to improve the compactness of a category. This way base learners in the ensemble system could better learn the special pattern in different categories to make a more accurate prediction. Another potential avenue for future work is to expand the collection of Whois and DNS information, resulting in larger datasets. This approach aims to mitigate bias and enhance the confidence of experimental results to provide insights into why additional label information did not lead to improvements in the performance of the ensemble system in detecting malicious domains.

References

[1] Sjösten, A., Snyder, P., Pastor, A., Papadopoulos, P., Livshits, B.: Filter list generation for underserved regions. In: Proceedings of The Web Conference 2020, pp. 1682–1692 (2020)

[2] Verisign: Verisign domain name industry brief: 350.4 million domain name registrations in the fourth quarter of 2022. https://blog.verisign.com/domain-names/verisign-q4-2022-the-domain-name-industry-brief/#:~:text=Today%2C%20we%20released%20the%20latest,the%20third%20quarter%20of%202022. (2023 (accessed July 1, 2023))

[3] OpenDNS: Phishtank. https://doi.org/10.23721/100/1504408 Accessed 05 May 2023

[4] Abuse.ch: URLhaus. https://doi.org/10.23721/100/1504320 Accessed 02 May 2023

[5] Daigle, L.: WHOIS Protocol Specification. RFC Editor (2004). https://doi.org/10.17487/RFC3912 . https://www.rfc-editor.org/info/rfc3912

[6] ICANN: About WHOIS — ICANN WHOIS. https://whois.icann.org/en/about-whois (2017 (accessed July 7, 2023))

[7] ICANN: Registrant Contact Information and the ICANN WHOIS Data Reminder Policy (WDRP). https://www.icann.org/resources/pages/registrant-contact-information-wdrp-2017-08-31-en (2017 (accessed July 8, 2023))

[8] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

[9] Pires, T., Schlinger, E., Garrette, D.: How multilingual is multilingual bert? arXiv preprint arXiv:1906.01502 (2019)

[10] Haykin, S.: Neural Networks: a Comprehensive Foundation. Prentice Hall PTR, ??? (1994)

[11] Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Philip, S.Y., *et al.*: Top 10 algorithms in data mining. Knowledge and information systems **14**(1), 1–37 (2008)

[12] Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16, pp. 785–794. ACM, New York, NY, USA (2016). https://doi.org/10.1145/2939672.2939785 . http://doi.acm.org/10.1145/2939672.2939785

[13] Arik, S.Ö., Pfister, T.: Tabnet: Attentive interpretable tabular learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 6679–6687 (2021)

[14] Dietterich, T.G., *et al.*: Ensemble learning. The handbook of brain theory and neural networks **2**(1), 110–125 (2002)

[15] Zhauniarovich, Y., Khalil, I., Yu, T., Dacier, M.: A survey on malicious domains detection through dns data analysis. ACM Computing Surveys (CSUR) **51**(4), 1–36 (2018)

[16] Sato, K., Ishibashi, K., Toyono, T., Miyake, N.: Extending black domain name list by using co-occurrence relation between dns queries. In: Proceedings of the 3rd USENIX Conference on Large-Scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More. LEET'10, p. 8. USENIX Association, USA (2010)

[17] Krishnan, S., Taylor, T., Monrose, F., McHugh, J.: Crossing the threshold: Detecting network malfeasance via sequential hypothesis testing. In: Proceedings of the 2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). DSN '13, pp. 1–12. IEEE Computer Society, USA (2013). https://doi.org/10.1109/DSN.2013.6575364 . https://doi.org/10.1109/DSN.2013.6575364

[18] Sun, X., Wang, Z., Yang, J., Liu, X.: Deepdom: Malicious domain detection with scalable and heterogeneous graph convolutional networks. Computers Security **99**, 102057 (2020) https://doi.org/10.1016/j.cose.2020.102057

[19] Sun, X., Wang, Z., Yang, J., Liu, X.: Deepdom: Malicious domain detection with scalable and heterogeneous graph convolutional networks. Computers & Security **99**, 102057 (2020)

[20] Bilge, L., Sen, S., Balzarotti, D., Kirda, E., Kruegel, C.: Exposure: A passive dns analysis service to detect and report malicious domains. ACM Trans. Inf. Syst. Secur. **16**(4) (2014) https://doi.org/10.1145/2584679

[21] Antonakakis, M., Perdisci, R., Lee, W., Vasiloglou II, N., Dagon, D.: Detecting malware domains at the upper {DNS} hierarchy. In: 20th USENIX Security Symposium (USENIX Security 11) (2011)

[22] Saleem Raja, A., Vinodini, R., Kavitha, A.: Lexical features based malicious url detection using machine learning techniques. Materials Today: Proceedings **47**, 163–166 (2021) https://doi.org/10.1016/j.matpr.2021.04.041 . NCRABE

[23] Alrizah, M., Zhu, S., Xing, X., Wang, G.: Errors, misunderstandings, and attacks: Analyzing the crowdsourcing process of ad-blocking systems. In: Proceedings of the Internet Measurement Conference. IMC '19, pp. 230–244. Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/

3355369.3355588 . https://doi.org/10.1145/3355369.3355588

[24] Lashkari, A.H., Seo, A., Gil, G.D., Ghorbani, A.: Cic-ab: Online ad blocker for browsers. In: 2017 International Carnahan Conference on Security Technology (ICCST), pp. 1–7 (2017). https://doi.org/10.1109/CCST.2017.8167846

[25] Kintis, P., Miramirkhani, N., Lever, C., Chen, Y., Romero-Gómez, R., Pitropakis, N., Nikiforakis, N., Antonakakis, M.: Hiding in plain sight: A longitudinal study of combosquatting abuse. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 569–586 (2017)

[26] Shi, Y., Chen, G., Li, J.: Malicious domain name detection based on extreme machine learning. Neural Processing Letters **48**, 1347–1357 (2018)

[27] European Parliament, Council of the European Union: Regulation (EU) 2016/679 of the European Parliament and of the Council. https://data.europa.eu/eli/reg/2016/679/oj Accessed 2023-04-13

[28] Li, K., Yu, X., Wang, J.: A review: How to detect malicious domains. In: Advances in Artificial Intelligence and Security: 7th International Conference, ICAIS 2021, Dublin, Ireland, July 19-23, 2021, Proceedings, Part III 7, pp. 152–162 (2021). Springer

[29] Dou, Z., Khalil, I., Khreishah, A., Al-Fuqaha, A., Guizani, M.: Systematization of knowledge (sok): A systematic review of software-based web phishing detection. IEEE Communications Surveys & Tutorials **19**(4), 2797–2819 (2017)

[30] Pradeepa, G., Devi, R.: Malicious domain detection using nlp methods—a review. In: 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART), pp. 1584–1588 (2022). IEEE

[31] Sood, G., Laohaprapanon, S.: Shallalist Web Page Data. https://doi.org/10.7910/DVN/ZXTQ7V . https://doi.org/10.7910/DVN/ZXTQ7V

[32] Mohammed, R., Rawashdeh, J., Abdullah, M.: Machine learning with oversampling and undersampling techniques: overview study and experimental results. In: 2020 11th International Conference on Information and Communication Systems (ICICS), pp. 243–248 (2020). IEEE

[33] Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al.: Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016)

[34] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine

learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)

[35] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)