ITC 2/51

Information Technology and Control
Vol. 51 / No. 2 / 2022
pp. 313-331
DOI 10.5755/j01.itc.51.2.30380

Improved Detection of Malicious Domain Names Using Gradient Boosted Machines and Feature Engineering

Received 2021/12/23 | Accepted after revision 2022/03/10

http://dx.doi.org/10.5755/j01.itc.51.2.30380

HOW TO CITE: Alhogail, A., Al-Turaiki, I. (2022). Improved Detection of Malicious Domain Names using Gradient Boosted Machines and Feature Engineering. *Information Technology and Control*, 51(2), 313-331. http://dx.doi.org/10.5755/j01.itc.51.2.30380

Improved Detection of Malicious Domain Names Using Gradient Boosted Machines and Feature Engineering

Areej Alhogail

STCs Artificial Intelligence Chair, Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia; e-mail: aalhogail@ksu.edu.sa

Isra Al-Turaiki

Information Technology Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia; e-mail: ialturaiki@ksu.edu.sa

Corresponding author: ialturaiki@ksu.edu.sa

Malicious domain names have been commonly used in recent years to launch different cyber-attacks. There area large number of malicious domains that are registered every day and some of which are only active for brief periods of time. Therefore, the automated malicious domain names detection is needed to provide security for individuals and organisations. As new technologies continue to emerge, the detection of malicious domain namesremains a challenging task. In this study, we propose a model to effectively detect malicious domain names. Thisis done by evaluating the performance of several machine learning algorithms and feature importance measuresusing a recent DNS dataset. Based on the empirical evaluation, the *gradient boosted machines* GBM classification with a combination of lexical and host-based features produce the most accurate detection rates of 98.8% accuracy and a low false positive rate of 0.003. In terms of feature importance, measures used in this study agree on the importance of six features, five of which are lexical in nature. Furthermore, to make the best out of these relevantfeatures, we apply automatic feature engineering. Our results show that preprocessing the dataset using deep feature synthesis and then reducing the dimensionality improves the classifications performance as compared tousing raw features. The results of this study are then verified using a challenging category of domain names, the*domain generation algorithm* dataset, and consistent results are obtained.

KEYWORDS: malicious domain detection, cyber security, machine learning, host features, lexical features, gradient boosted machines (GBM) , deep feature synthesis (DFS).

1. Introduction

Threats to Internet and network security are continuously evolving and increasing as attackers develop newtechniques to avoid detection and prevention tools. One common threat to Internet security is malicious websites or domain name links that are used for cyber-attacks. Usually, attackers attempt to trick users into visiting them by several common means using malicious domain names that may contain phishing scams, drive-by downloads, malware, botnet commands and control, or other malicious content.

The *Domain Name Service* (DNS) is the backbone of the Internet. It is a decentralised, well-distributed, hierarchical naming system that is used to allow users to access computational services through translating human-readable domain names, *uniform resource locators* (URLs), into computer-readable *Internet Protocol* (IP) addresses, and vice versa. Some web vulnerabilities of Internet settings and policies allow for the registration of malicious domain names with DNS servers [29]. According to the Google Safe Browsing report of 2021 [13], Google issued a monthly average of 3 million warnings during the months of 2020 to warn users about unsafe sites. Daily, an average of 40,000 malicious domain names are created, causing average losses of $ 17,700 per minute [33]. Distinguishing among *non-malicious*, or *benign*, and *malicious* domain names is critical for allowing or restricting access to external services, thus ensuring network and infrastructural security and preserving the privacy and security of users and organisations. Nevertheless, leaving the decision to users to evaluate the safety of a domain name link can be a difficult and costly judgment, and relying on conventional methodsis not sufficient. Therefore, automating the detection ofmalicious domains is a must to protect the security of individuals or organisations.

Several techniques for detecting malicious domains have recently been developed to identify malicious domains through an analysis of domain name data without the need for a costly and time-consuming anal ysis of page content. Most conventional DNS security monitoring solutions are based on constantly updated malicious domain lists, i.e., *blacklists*. Nevertheless, blacklisting lacks effectiveness in detecting new unknown malicious communications, especially with therapidity with which domain names are generated and registered using current tools, such as DNS fastflux and *Domain generation algorithms*, which generate many distinct domain names. An average of 0.01%of domain names are malicious and short-lived in order to prevent blacklist blocking [21]. The enormous quantity of existing domain names is difficult to track in real time. Moreover, they is vulnerable to distributed*denial of service* attacks, since they cannot handle the traffic of thousands of domain names generated by cyber-attackers. These reasons make blacklisting not enough to secure organisations from these threats and attacks [6, 35].

Therefore, it is necessary to make use of technologies that enable effective automatic detection. The detection of malicious domains based on their distinct behavior has been the subject of extensive investigation. There are mainly two types of detection: *knowledge-based* and *machine-learning-based* approaches [42]. In the first approach, external expertise and various heuristics are used to distinguish between malicious and non-malicious domains, such as the heuristic-based technique that uses the payload signatures of known attacks. Nevertheless, this method falls short of detecting novel attacks that lead to zero-day exploits [25]. The second approach depends on data-driven algorithms to automate the discrimination process. Machine learning has been applied to effectively to classify unknown domains as being malicious or not [42, 29]. The enormous amount of knowledge that needs to be considered in knowledge-based models urges investigation in machine-learning-based methods that can automatically derive knowledge from highdimensional data [42]. DNS data became an ideal option for various machine learning techniques due to the availability of a extensive number of features and a huge volume of traffic data [42]. Supervised approaches are common in this regard due to their efficiency and their ability to automatically select the most relevant features from the raw data [38, 35, 39, 29, 41, 16, 32]. Machine-learning-based detectors can identify new domains based on various specified features after being trained by a set of labeled domain names. They achieve the goal of minimizing the large amount of domain name filtration in real-time while maintain a high detection rate and a low false positive rate [21].

Different types of features have been used to classify domain names. The most common set of features used are the following: lexical, host-based, content-based, reputation, descriptive, and so forth. Lexical features are vocabulary attributes that describe the syntax of a domain name, such as length, vowel ratio, consonant ratio, and the inclusion of special characters. Host-based features are features that describe host attributes, such as the attributes related to the IP address, WHOIS, and geographical location. Some research has used passive and active features that represent features that can be directly extracted from DNS queries and record data. Passive features are collected passively, and active features are computed with additional external information. When determining the maliciousness of a URL, the majority of work, whether content-based or noncontent-based, ignores the domain name and DNS data resulting in erroneous results. As a result, a solid mechanism for detecting malicious domains will help improve the precision with which hazardous URLs are detected.

In this work, we aim to propose an effective malicious domain name detection model which is constructed by studying the most influential features and best accuracy classifiers. In order to develop this model, we empirically evaluate state-of-art machine learning algorithms for the detection of malicious domain names. This is to answer the question of *what is the most effective machine learning algorithm for malicious domain name detection among the compared algorithms*. Friedmans statistical significance test is applied to confirm the study observations. The effective classification usually depends on feeding the most influential features. Many studies use lexical features, host-based features, or a combination of both. We investigate which type of features leads to more accurate classification results (i.e., *does the feature category affect the classification accuracy of domain names?*). Using a recent DNS dataset [26], five machine learning algorithms are trained separately using one of the three feature categories: host-based, lexicalbased, and a combination of both. Then, we explore feature importance using four feature importance measures to answer the question of *what are the most influential features in the automatic detection of malicious domain names.* We further ask the question of *can the most relevant features be exploited using automatic feature engineering techniques to improve classification accuracy?* To answer this question, machine learning algorithms are trained using reduced feature space of engineered features. The results are compared with classification using the raw features. Based on the experimental results, we propose a malicious domain name detection model that uses the GBM classifier and the identified influential features to classify malicious and benign domain names with a high accuracy rate. Finally, our proposed model is verified using a specific category of malicious domains, *domain generation algorithm* (DGA) dataset [10]. This is conducted in order to see whether the same observations would be valid if we narrow down the scope of malicious domains to more challenging categories of domain names. The proposed solution can be implemented on web browsers as an add-on to enable users to automatically detect malicious domains to protect their security while surfing the web, as it can protect them from malicious domains that are not blacklisted.

The rest of this paper is organized as in the following. In Section 2, previous literature related to domain name detection is presented. The dataset and the methodology used to conduct the analysis are described in Section 3. In Section 3.5, we discuss the evaluation of the model performance, followed by a discussion of the results in Section 4. Finally, Section 5 presents the conclusions and suggestions for future work.

2. Related Work

Different technologies have been proposed in the literature and in practice to detect malicious domain names. For years, blacklisting and knowledge-based methods have been used to block malicious domain names; nevertheless, it is not effective for detecting newly generated malicious domains. To resolve these shortcomings, researchers have suggested analysing the domainname data and extracting interesting features to predict whether it is legitimate, without even examining the content of the website [23]. Several studies have suggested using machine learning approaches that havedemonstrated remarkable abilities for the detection of malicious domains; consequently, they have gained great popularity. Machine learning approaches analysethe data of domain names and extract good feature representations. They are based on training data of malicious and non-malicious domain names to learn how topredict and classify domain names as malicious or benign. In a supervised learning mode, data are labelled as malicious or not malicious. Most of the studies treatedthe detection of malicious domain names as a binary classification problem. Classification is based on domain name features that are divided into lexical, hostbased, reputation, and other classes of features. Several features are extracted from domain names, such asthe length of the domain name, the inclusion of hexadecimal characters, the @ symbol, the IP address, andthe form tag in the domain name, all of which could be an indication of a malicious domain. Other factors such as the number of dots, the domain name count, and the HTTP status could be used to distinguish malicious and benign domains. A good selection of appropriate features that balances accuracy and robustness highly affects the success of classification and detection [42]. It is also important to note that features can be forged by attackers to deceive detection tools. Therefore, real-time training sometimes yields better accuracy [16]. Different studies have proposed the use of one feature or a combination of features with machine learning techniques to accurately detect malicious domain names. Mamun et al. [25] claim that a domain names lexical analysis is effective and efficient for detecting malicious domain names. Their proposed model used obfuscation techniques with lexical analysis and accurately detected malicious domains out of 110,000 URLs, showing an accuracy rate of more than 97% and the ability to classify various URL attack as benign, defacement, spam, phishing and malware. Lexical analysis is effective because it is believed that certain red flag words and characters tend to appear in malicious domain names [23]. Nevertheless, the use of multiclass classification in their approach usually degrades the overall performance. For the proactive detection ofdomain names, Liang and Yan [20] proposed a model based on *deep bidirectional long short-term memory* (DBLSTM) to detect malicious domain names as a binary classification problem using lexical features. Their model accuracy was 98.6%, showing that the DBLSTMclassifier was superior to the other conventional machine learning approaches used in the evaluation. Theyused a public dataset with 2.4 million domain names including 3.2 million features for analysis. Nevertheless, they were challenged with class imbalances, which pointed towards the need for a larger dataset. Liu et al. [22] used character features that combine the lexical features and structural features of malicious domain names with a Random Forest classification algorithm to detect malicious domain names, with an accuracy that reached 99%. Descriptive static features to complement lexical features were used by [21] to detect malicious domain names, with a detection accuracy of 91% for2 million tested domain names resulting in a 75% reduction in workload size and the ability to detect shortlifetime characteristics of malicious URLs. However, the model based on lexical features requires a continues update to keep its performance. Saleem Raja etal. [33] examinedlexical features using a light weighedmethod that reduce time and storage requirements andfound that it was effective in classifying domain namesusing a random forest classifier and KNN in. They suggested that not all extracted features were suitable for classification and suggested utilizing the feature reduction approach to determine the fitness of the features before reducing them using correlation analysis.

On the other hand, some research based their classification on host-based features. Using a number of host-based features such as ASNs, Khalil et al. [19] created a graph-based inference technique over relateddomains. Their method assumes that a domain with a strong associations with known malicious domains is likely to be malicious. Using a modest range of previously known malicious domains and carefully constructed associations, a huge number of new maliciousdomains can be discovered. Their approach exhibited high true positive rates of 95% and low false positive rates of less than 0.5% when using a public passive DNS database. However, their model could misclassifysome URLs hosted by same IP address if one associated is malicious or share public IPs. Nevertheless, some research combined both lexical and host-based features.

To safeguard users from phishing attacks, Rupa et al. [32] presented a machine learning based classifier thatuses a Random Forest classifier to detect hidden malicious domain names on the web using a combination ofdomain names' host-based and lexical features. Their model aimed to detect authenticity and achieved an accuracy of 98.2% eliminating the probabilities of overfitting experienced in traditional decision trees. Nevertheless, their model compares new URLs to an existing database which requires continuous maintenance. Ma et al. [23] also analysed domain names lexical and host-based features rather than investigating the web page content. They employed online algorithms that proved to be more efficient at processing large numbers of URLs and adapting to new features in the constantlyexpanding distribution of malicious URLs. However, in order to get accurate results, the model needs continual retraining in the face of new features. Shi etal. [35] proposed a malware domain name detection methodology based on Extreme Machine Learning anda neural network with high accuracy, better throughput and a quick learning speed, to avoid Advanced Persistent Threat (APT) attacks. They were able to classify malicious domain names using attributes gathered froma variety of sources, combining lexical and host-basedfeatures with a high detection rate with a 95% accuracy. The main features used were the length of the domain, IP address, the number of consecutive characters, the average TTL value average, the standard deviation, the entropy of the domain, the number of countries, the active time, and the lifetime of the domain. These features were classified under four categories: construction-based, IP-based, TTL-based, and WHOIS-based. Iwahana et al. [16] used Extreme Machine Learning in conjunction with a set of optimised features that mix host-based, lexical, and web-based data to deliver higher accuracy and throughput in detecting previously undetected malicious domains using permutation importance and real-time training. Nevertheless, the permutation importance affect feature importance due to variance on features, limiting their important value to results. Moreover, real time data collection leave out several features that cannot be collected for the real-time training. In addition, Iwahana et al. [16] model runs in client-server approach which requires a continuous server management. Several machine learning classifiers have been proposed for detecting malicious DNS. Nearly all of the existing research focused on the effective malicious domain names detection in terms of improving accuracy and decreasing the false positive rate. The performance of ML algorithms is usually affected by dataset and the features used. Hostbased, content-based, and popularity-based features incur more processing time and resources to extract the desired features, whereas lexical features involve less computation and do not require access to any external sources [33]. With the advancements in technology, attackers improve their tactics to bypass known detection methods. At the same time, more datasets become available encouraging researcher to investigatenew methods to improve the automatic malicious domain name detection. In this study, we utilize recent datasets to evaluate state-of-art machine learning algorithms to identify the most effective classifier and the most influential features that will yield better results.

3. Materials and Methods

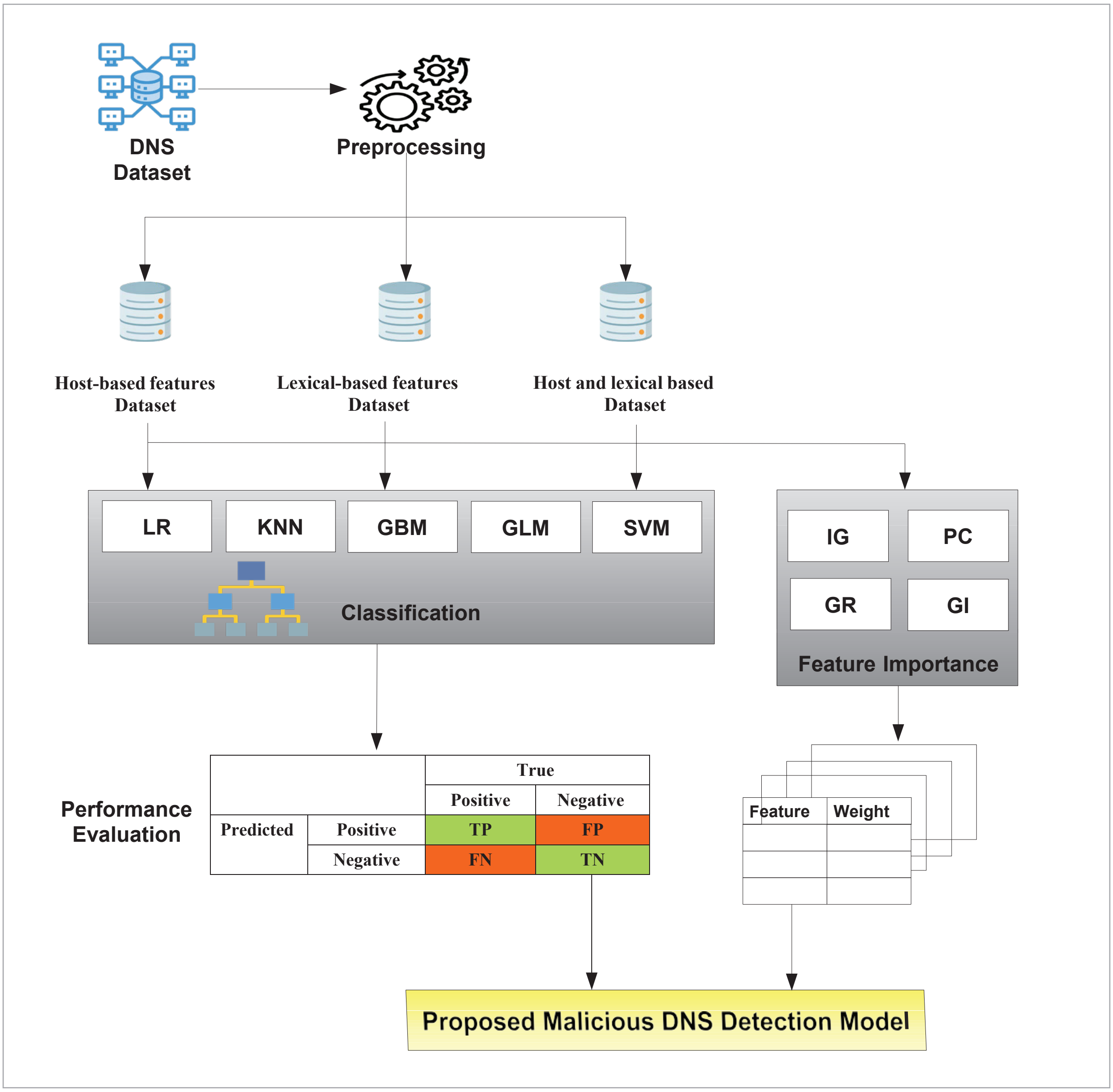
Malicious DNS name detection is a binary classification problem. Assume D, a dataset of domain names, where domain d_i is defined using a set of n features, $F = \{f_1, f_2, ..., f_n\}$, and each domain $d_i \in D$ is either *malicious* or *benign*. We need to train a supervised machine learning algorithm using D, such that the resulting model M is able to classify a new domain d_{new} that has not been seen before by M. This section gives a synopsis of the DNS dataset used in this study. Additionally, we describe the investigated classification algorithms, feature importance methods,feature engineering, and dimensionality reduction approaches. This empirical study conduct different stepsof evaluation of features and classifiers to construct the most effective model of malicious domain name detection.

A schematic view of our study is shown in Figure1.

3.1. Dataset Description

We conduct our study using two recent DNS datasets, one is used for investigation and the other for validation of the observations. The first dataset for malicious domain detection is collected and prepared by

Figure 1
A schematic view of our study


DNS Dataset
Preprocessing
Host-based features Dataset
Lexical-based features Dataset
Host and lexical based Dataset
LR
KNN
GBM
GLM
SVM
Classification
IG
PC
GR
GI
Feature Importance
Performance Evaluation
Feature
Weight
Proposed Malicious DNS Detection Model

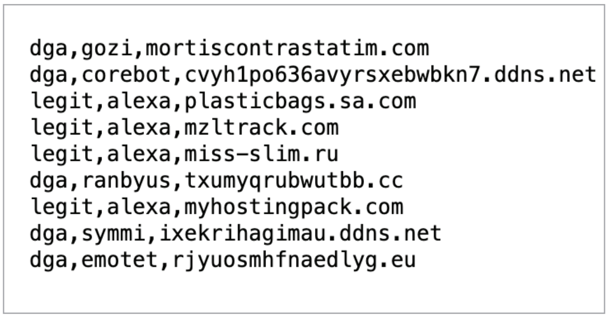
		True	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

Marques et al. [26] between September and November 2020. This dataset was intended to address the shortage of malicious and benign domain datasets based on DNS logs, which are essential in the field of cybersecurity.

The data was generated from scratch using malicious and non-malicious domain name DNS logs that were made available to the public. Data was collected utilising the Rapid7 Labs [30] open data repository as well as a well-known malicious list supplied by SANS [34] and was collected during the months of September, October and November 2020. Thirty-four features were derivedfrom the domain name. A number of domain name attributes were gleaned directly from the name, including the entropy, strange character count, and length. In addition, information such as the date on

Figure 2

A sample of the DGA dataset



```
dga,gozi,mortiscontrastatim.com
dga,corebot,cvyh1po636avyrsxebwbkn7.ddns.net
legit,alexa,plasticbags.sa.com
legit,alexa,mzltrack.com
legit,alexa,miss-slim.ru
dga,ranbyus,txumyqrubwutbb.cc
legit,alexa,myhostingpack.com
dga,symmi,ixekrihagimau.ddns.net
dga,emotet,rjyuosmhfnaedlyg.eu
```

which a domainname was created, IP addresses, open ports, and geolocation was collected from data enrichment processes. Dataset feature description, data types, and feature categories are summarised in Table 1. There are approximately 90,000 domain names in the dataset, with 50%of them being *benign* and 50% of them being *malicious*. For pre-processing, all nominal values are converted to numerical values, such that they can be handled by the machine learning algorithms. This is done using *dummy coding* to indicate whether or not a category is present or absent. A value of *zero* implies the absence of the category and a value of *one* denotes its presence. A two-dimensional binary matrix is created by converting categorical variables to dummy variables, where each column represents a separate category.

The second dataset is a balanced dataset of 50,000 records for *domain generation algorithm* (DGA) detection [10]. The DGA dataset contains DGA families extracted from the Netlab Opendata Project repository [28] and *benign* domain names retrieved from Alexa. A sample of the DGA dataset is shown in Figure 2. Weuse this dataset to verify the model and observations obtained using the first dataset. This dataset represents a specific category of malicious domains which is more challenging to detect.

3.2. Classification Models

Five well-known classifiers, namely, *logistic regression* (LR), *gradient boosted machines* (GBMs) [11], *generalised linear models* (GLMs) [27], *K-nearest neighbor* (KNN), and *support vector machines* (SVMs) [9], have been utilised to model domain name service data. This subsection provides a brief description of each of these classifiers.

- **Logistic regression:** Given a set of data points, logistic regression models the likelihood of each data point belonging to a particular class based on the values of other characteristics (independent). It then makes use of the model to predict the likelihood that a given data point belongs to a particular class in question. When constructing the regression model, the sigmoid function is utilised. It is assumed that the data points are distributed according to a linear function.
- **Gradient boosted trees:** This is used for regression and classification problems. GBMs are an ensemble method that builds a series of trees in a sequential manner. In each iteration, a tree's performance is evaluated and enhanced in accordance with the results of the preceding iteration. It consists of three components: a loss function (for example, the mean square error), a weak learner (for example, decision trees), and an additive model (for example, a random forest). GBM algorithms search for a final model that minimizes the loss function and returns it to the user.
- **Generalised linear model:** This is a generalisation of conventional linear models. By maximising the log-likelihood, this approach fits extended linear models to the data. The regularisation of parameters can be accomplished using the elastic net penalty. Model fitting is performed in parallel, is highly fast, and scales effectively for models with a small number of predictors with non-zero coefficients.
- **K-nearest neighbor:** This is a classification algorithm that is based on similarity. In KNN, a new data point is classified based on the classification of the data points that are immediately adjacent to it. KNN stores the training datasets. Each time an unlabelled data point is received, a majority vote among the neighbors in the training dataset is taken into consideration. Thus, KNN models are described as *lazy classifiers*. It is possible that the classification results will be skewed in the case of imbalanced datasets.
- **Support vector machines:** These are machine learning algorithms that transform the training dataset into a higher-dimensional representation of the training dataset. It then finds the best

Table 1. Dataset features with description, data types, and feature categories

Feature	Description	Data Type	Feature category
Domain	Baseline DNS utilized to enhance data (derive features)	Text	
DNS Record Type	Retrieved DNS record type	Text	host-based
MX DNS Response	The result of a DNS lookup for the MX record type	Boolean	host-based
TXT DNS Response	DNS response for the TXT record type	Boolean	host-based
Has SenderPolicyFramework Info	If the Sender Policy Framework attribute is included in the DNS response	Boolean	host-based
Has DKIM Info	If Domain-Based Message Authentication is included in the DNS response	Boolean	host-based
Has DMA Info	Domain-Based Message Authentication is available in the DNS response	Boolean	host-based
IP address	The domain IP address	Text	
Domain In Alexa DB	If the domain is registered in Alexa database	Boolean	host-based
Common Ports	If the domain is available for common ports (80, 443, 21, 22, 23, 25, 53, 110, 143, 161, 445, 465, 587, 993, 995, 3306, 3389, 7547, 8080, 8888)	Boolean	host-based
Country Code	The country code associated with the domain's IP address	Text	host-based
Registered Country	The country code assigned to a domain during registration process(WHOIS)	Text	host-based
Creation Date	The domain's creation date (WHOIS)	Enumerate	host-based
Last Update Date	The domain's most recent update date (WHOIS)	Enumerate	host-based
ASN	The domain's Autonomous System Number	Integer	host-based
Http Response Code	The domain's HTTP/HTTPS response code	Enumerate	host-based
Registered organisation	The organisation name associated with the domain (WHOIS)	Text	host-based
Sub-domain Number	The sub-domains total number	Integer	lexical
Entropy	The domain Shannon Entropy value	Integer	lexical
Entropy Of Sub-Domains	The average entropy value for the sub-domains	Integer	lexical
Strange Characters	The maximum number of characters hat differ from [a-z A-Z] while taking into account the existence of two numeric integer values	Integer	lexical
TLD	The domain's Top Level Domain	Text	lexical
IP Reputation	The result of the IP's blocklist search	Boolean	reputation feature
Domain Reputation	The result of the domain's blocklist search	Boolean	reputation feature
Consoant Ratio	The domain's consonant character ratio	Decimal	lexical
Numeric Ratio	The domain's numeric characters ratio	Decimal	lexical
Special Char Ratio	The domain's special characters ratio	Decimal	lexical
Vowel Ratio	The domain's vowel characters ratio	Decimal	lexical
Consonant Sequence	The domain's maximum number of consecutive consonants	Integer	lexical
Vowel Sequence	The domain's maximum number of consecutive vowels	Integer	lexical
Numeric Sequence	The domain's maximum number of consecutive numeric	Integer	lexical
Special Char Sequence	The domain's maximum number of consecutive special characters	Integer	lexical
Domain Length	The domain's total length	Integer	lexical
Class	The class of the domain (malicious = 0 and non-malicious = 1)	Integer	label

hyperplane that divides data points belonging to one class from data points belonging to another class. When choosing an optimal hyperplane, it searchesfor the one that has the greatest margin, whichis defined as the distance between the data points from each class and the hyperplane itself.

3.3. Feature Importance

Feature importance indicates strategies for valuing input features depending on on their predictive power for a target variable. There are numerous forms of feature importance scores. Statistical correlation scores, coefficients created as part of a linear model, decision trees, and permutation importance scores are just a few examples. Feature importance scores are critical components in predictive modelling since they provide enlightenment of the data and the model. In this study, four measures are used: information gain, gain ratio, Gini index, and Pearson Product-Moment Correlation Coefficient. The choice of these measures is based on their simplicity, light computational requirements, and their abilityto minimize overfitting. In addition, they are independent of the learning algorithm used for classification and are frequently employed with success for various datasets [5]. Let D be a dataset of m classes, let a be afeature that takes V possible values $\{a^1, a^2, ..., a^V\}$ in D, let D^v be the subset of samples from D that takes the value of a^v for feature a, let p_i be the probability that a sample belongs to class i. A brief description of the four measures used in this study is given below:

- **Information Gain:** This metric is based on Claude Shannon's seminal work in information theory, which investigated the value or "information content" of messages. The information gain from separating the data set D by feature is determined as follows:

$$Gain(D, a) = Ent(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v). \quad (1)$$

where $Ent(D)$ is the entropy. High information gain values indicate more purity archived by dividing D based on feature a.

- **Gain Ratio:** The information gain appears to be biased towards features having more values. When choosing features, the gain ratio is utilized instead of information gain to decrease bias. The features gain ratio is calculated as:

$$Gain\_Ration(D, a) = \frac{Gain(D, a)}{IV(a)}. \quad (2)$$

where $IV(a)$ denotes the *intrinsic value of feature a* and is calculated as follows:

$$IV(a) = -\sum_{v=1}^{V} \frac{|D^v|}{|D|} log_2 \frac{|D^v|}{|D|}. \quad (3)$$

- **Gini Index:** This is a measure of dataset impurity.It calculates the weight of the feature with respectto the class label by computing Gini index of the class distribution. The lower the Gini index, the higher the dataset's purity. The Gini index for a feature a in a dataset D is calculated as follows:

$$Gini\_Index(D, a) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} Gini(D^v). \quad (4)$$

where

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2. \quad (5)$$

- **Pearson Product-Moment Correlation Coefficient:** The correlation coefficient indicates how strong the relationship between the relative movements of two variables. The values range from negative to positive. A positive correlation is shown by a linear correlation coefficient larger than zero. A relationship is said to be negative if the value is less than zero. Finally, a score of 0 implies that there is no correlation between the two variables x and y. Three quantities need to be determined in order to calculate the Pearson product– moment correlation: $Cov(x, y)$, the covariance of the two variables under investigation, σ_x, the standard deviation for variable x, and σ_y, the standard deviation for variable x. Considering that, the correlation coefficient is determined as follows:

$$\rho_{xy} = \frac{Cov(x, y)}{\sigma_x \sigma_y}. \quad (6)$$

3.4. Automatic Feature Engineering

In order to exploit the information carried by the top features identified by the feature importance measures (Subsection 3.3), we utilize feature engineering techniques. *Feature engineering* refers to the process of building new features from existing ones. This process improves model performance by utilizing the power of the most relevant features [2]. In this research, we use *Deep Feature Synthesis* (DFS) [18] to generate new deep features. The new deep features are built by applying mathematical functions to the data in different columns and rows. This results in a set of new features, where the *depth* of a feature is defined as number of primitive operations necessary to generate it. Using the top features agreed upon by all the feature importance measures, a total of 37 features are generated using addition and multiplication primitives. Since the resulting features could be redundant or strongly correlated, the performance of classification algorithms may be compromised. Thus, we apply *dimensionality reduction using Principal Component Analysis* (PCA) with 95% of the variance is retained. PCA reduces the feature space into a smaller one that retains much of the information present in the larger feature space [17]. Compared to other dimensionality reduction, research shows that PCA is less sensitive to noise [37, 1].

3.5. The Proposed Model

The goal of this study is to propose an improved malicious domain name detection model. The problem at hand is formulated as a binary classification problem where we distinguish between two classes: *benign* and*malicious*. Based on influential features and the best performing classifier found, we propose a malicious domain name detection model. Figure 3 shows the architecture of our proposed model. The classification method begins with domain names being fed into the classifier, which are then processed to extract the most influential features. Then, DFS is applied in order to generate deep features. This results in a larger feature space that is reduced using PCA. Finally, the subset of features produced by PCA is fed into the classification algorithm

3.6. Experimental Settings

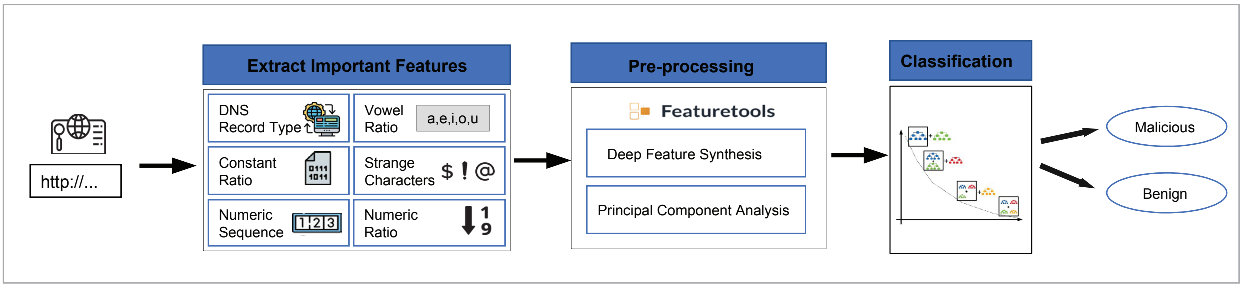
The proposed models were implemented using *Python*and *Scikit-Learn library*.All experiments were run using a MacBook Pro, with the macOS Catalina operating system, version 10.15.7, and a 2.9 GHz Quad-Core Intel Core i7 with 16 GB RAM.

Depending on the parameter settings, the performance of various algorithms can vary. In this study, the algorithms were run using the following parameters. For SVM, the regularization parameter is set to 1, linear kernel, no class weights, and using shrinking heuristic. Tolerance for the early stopping is 0.001. For LR, regularization of 1, no class weights, fit intercept is set to true, maximum iterations set to 100, L2 penalty term, Tolerance for the early stopping is 0.0001. For GBM, the learning rate is set to 0.1,' tolerance for the early stopping to 0.0001, the quality of split is measured using Friedman mean squared error, and the loss function to be optimized is set to *deviance*, which refers to logistic regression. For KNN, k = 5 and all points in each neighbourhood are weighted equally. The leaf size is set to 30 and the distance metric to Minkowski with p=2 (Euclidean Distance). For GLM, model family is binomial, number of iterations is 100, and with iteratively reweighted least squares method.

3.7. Performance Measures

In order to assess the trained classifiers' performance, 10-fold cross validation was performed, and

Figure 3
The architecture of the proposed model



the following performance measures were calculated: *accuracy,precision*, the *false positive rate*, the *detection rate*, and the *F-measure*.

The percentage of correctly identified domain names is called *Accuracy*, and it is computed using the following formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{7}$$

Precision is the percentage of malicious records accurately classified out of the total number of malicious records. The following formula is used to calculate precision:

$$Precision = \frac{TP}{TP + FP}. \tag{8}$$

The *false positive rate (FPR)* is the percentage of benign records that have been wrongly identified as malicious. It is calculated as:

$$FPR = \frac{FP}{FP + TN}. \tag{9}$$

The *detection rate (DR)*, also known as the *True Positive Rate* or *Recall*, is the percentage of malicious records appropriately labeled out of all maliciousrecords. The following formula is used to compute the detection rate:

$$DR = \frac{TP}{TP + FN}. \tag{10}$$

The *F-measure (F)* is a combined measure of the precision and detection rate that is calculated as follows:

$$F - measure = \frac{2 * P * DR}{P + DR} \tag{11}$$

where TP (true positive) refers to the number of malicious records correctly identified as malicious, FP (false positive) refers to the number of benign records inaccurately assigned as malicious, TN (true negative) refers to the number of benign records properly assigned as benign, and FN (false negative) refers to the number of malicious records incorrectly identified as benign.

In addition, we report the area under the *ROC curve* (AUC). The *receiver operating characteristic* (ROC) is a graph of the *true positive rate* versus the *false positive rate*. AUC quantifies the full two-dimensional area beneath the entire ROC curve from (0,0) to (1,1). AUC ranges in value from 0 to 1.

In this study, several classifiers were trained using the dataset with all features and then using two feature subsets: *host* and *lexical*. Thus, a statistical significance test was critical for determining the feature subset that yields better classifier performance. We used Friedmans test [12], a non-parametric test for detecting treatment differences across numerous attempts. We measured the value of Friedmans ranks for accuracy and AUC, where α = 0.5. Our hypothesis in this paper, H0, is that there is no difference between classifier performances across the various feature subsets.

3.8. Results

The majority of the currently available research focuses on improving DNS classification accuracy by experimenting with alternative feature sets and machine learning algorithms. Focusing on influential feature sets helps reduce much of the additional processing time and resources. DNS data reveal patterns for each class, allowing for the comparison and analysis in a multi-feature investigation. Each of the five classification models utilized in this study is trained separately using three sets of features: host-based features, lexically based features, and a combination of both. Our experiments were designed to study classification performance from three aspects: perfeature category (host versus lexical), per classification model, and overall feature categories and models. In addition, we look into feature importance in order to improve the effectiveness and accuracy of recognising malicious domain names.

3.8.1. Machine Learning Algorithms

We look at the classification performance of models trained using host-based features, as compared to lexically based features. Tables 2-3 show the evaluation measures for the five classification models trained using host-based features and lexically based features, respectively. We observe that all the models trained using lexically based features outperform host-based trained models in terms of all measures, regardless of the classification algorithm.

Table 4 Presents the evaluation measures of the five classification models trained applying a combination of features. Compared to the classifiers in Tables 2-3,

Table 2
The performance measures for the five classification models trained using host-based features only

Model	Accuracy	Precision	DR	F-measure	AUC	FPR	Running Time
LR	85.26%	87.47%	82.76%	85.05%	0.931	2.75%	00:08:41
KNN	67.18%	62.05%	88.54%	72.97%	0.911	4.67%	12:21:01
GBM	87.66%	97.17%	77.60%	86.29%	0.939	1.03%	00:01:19
GLM	84.46%	90.27%	77.26%	83.26%	0.917	8.33%	00:00:43
SVM	84.25%	85.91%	81.94%	83.88%	0.925	7.87%	00:18:47

Table 3
The performance measures for the five classification models trained using lexically based features only

Model	Accuracy	Precision	DR	F-measure	AUC	FPR	Running Time
LR	89.52%	90.17%	89.42%	89.79%	0.949	11.69%	00:6:25
KNN	95.45%	95.37%	95.56%	95.46%	0.995	3.13%	01:20:49
GBM	95.60%	93.78%	97.70%	95.70%	0.997	5.79%	00:00:14
GLM	91.66%	93.83%	89.20%	91.46%	0.978	3.61%	00:00:10
SVM	93.40%	97.12%	89.46%	93.13%	0.968	4.50%	00:02:27

Table 4
The performance measures for the five classification models trained using a combination of host-based andlexical features

Model	Accuracy	Precision	DR	F-measure	AUC	FPR	Running Time
LR	97.91%	99.85%	95.96%	97.87%	99.90	16.00%	00:19:46
KNN	93.87%	96.02%	91.54%	93.73%	98.90	2.67%	17:00:01
GBM	98.98%	99.70%	98.26%	98.97%	99.90	0.30%	00:01:52
GLM	98.08%	99.22%	96.92%	98.06%	99.70	0.93%	00:01:03
SVM	98.57%	99.43%	97.70%	98.56%	99.70	0.22%	00:7:43

the trained classifiers using a combination of features yield better accuracy measures.

For the host-based trained models, KNN achieves the lowest accuracy values, and GBM achieves the highest. For lexically based trained models, LR achieves the lowest accuracy values, and GBM again achieves the highest. For the trained models using a combination of features, KNN achieves the lowest accuracy values, and GBM achieves the highest.

Among all 15 trained models in this study, our results indicate that the worst performance is achieved by the KNN model trained using host-based features, while the best performing model is the GBM trained using the combination.

Friedmans test shows that our results are statistically significant, with Friedmans test values of 8.4 for both accuracy and AUC measures. As for running time, theclassification time of new domains is measured. Results show that models trained using lexical features took the least time to train. In particular, the GBM and GLM trained using lexical features were the fastest among all15 models. In second place comes models trained using the combination. The KNN models were very slowcompared to the rest of the models.

3.8.2. Feature Importance

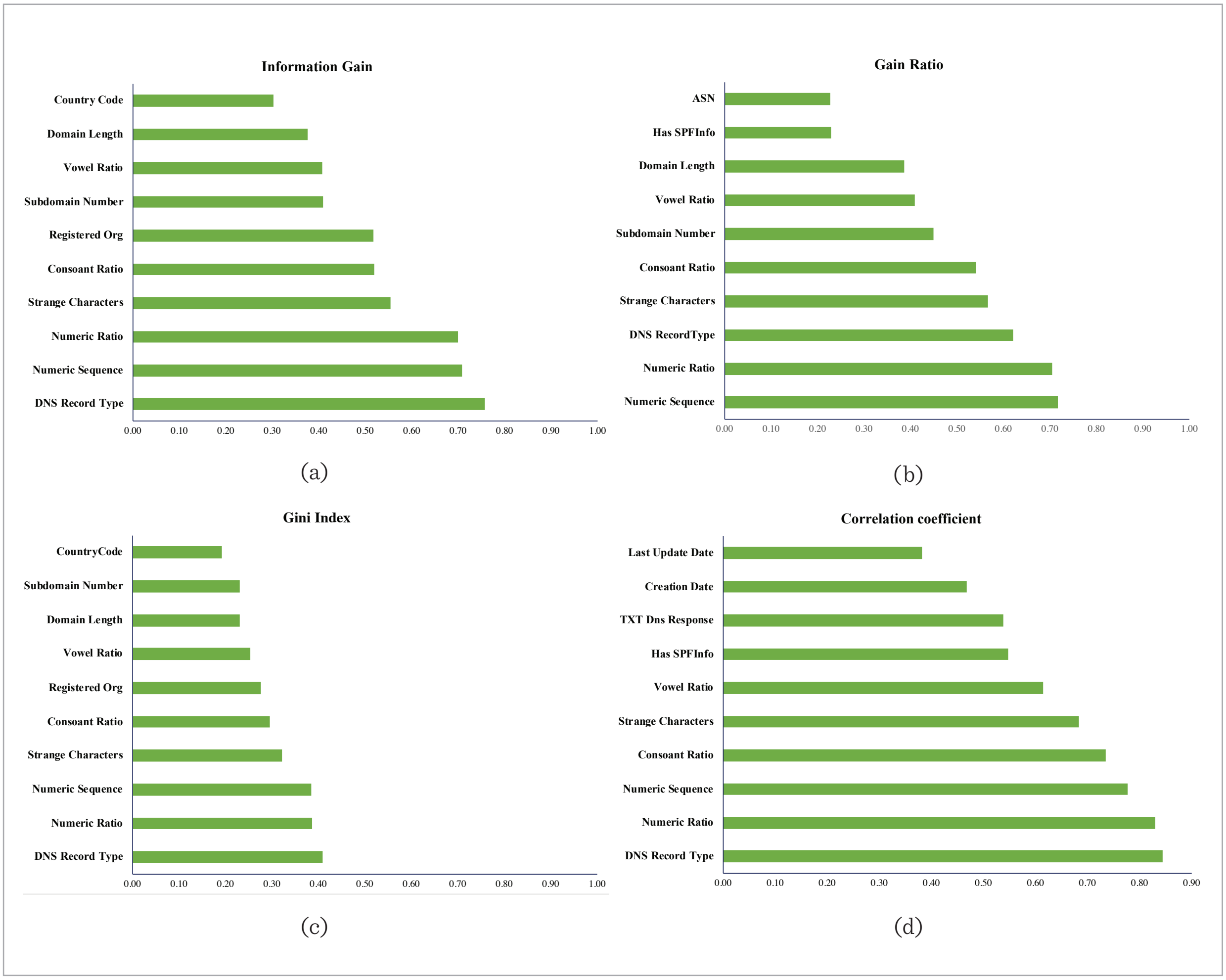
Next, we discuss the importance of features, such as which features are essential and significantly related toprediction accuracy. Figure 4 shows the top 10 features and their corresponding weights using the four feature importance measures used in this study. The results show that there is a consensus among the four measures on the importance of six features, five of which are lexically based features. The six features are DNS Record Type, Numeric Sequence, Numeric Ratio, Strange Characters, Consonant Ratio, and Vowel Ratio. Our results are consistent with results in the literature on using lexical features. According to Blum et al. [4], using lexical features to train phishing detection models produces robust classifiers.

3.8.3. DSF and PCA

Finally, we explore whether engineering the most relevant features would improve the classification performance. In Tables 5 and 6, we show the performance of the machine learning algorithms when trained using the top six best features (raw) and the deep features synthesised from them, respectively. The results show that applying deep feature synthesis generally improves the classification performance. However, only in GBM the synthesised features produce less FPR.

Figure 4

Top ten features and their corresponding weights using a) information gain, b) gain ratio, c) Gini index,and d) Pearson Product-moment Correlation coefficient



(a)

(b)

(c)

(d)

Table 5

The performance measures for the five classification models trained using the top six best features

Model	Accuracy	Precision	DR	F-measure	AUC	FPR
LR	95.74%	97.79%	93.65%	95.75%	95.67%	2.13%
KNN	97.95%	98.66%	97.25%	97.95%	97.95%	1.33%
GBM	98.23%	97.80%	98.70%	98.23%	98.25%	2.24%
GLM	95.41%	96.95%	93.81%	95.41%	95.36%	2.98%
SVM	95.86%	98.67%	93.02%	95.87%	95.76%	1.26%

Table 6

The performance measures for the five classification models trained using feature space generated usingDFS and PCA

Model	Accuracy	Precision	DR	F-measure	AUC	FPR
LR	96.67%	97.32%	96.01%	96.67%	96.66%	2.67%
KNN	98.37%	98.04%	98.73%	98.37%	98.38%	1.98%
GBM	98.29%	98.14%	98.46%	98.29%	98.30%	1.87%
GLM	96.66%	97.57%	95.74%	96.67%	96.65%	2.40%
SVM	96.98%	97.31%	96.66%	96.98%	96.99%	2.70%

3.8.4. Model Verification

In order to verify our proposed model we further investigate a publicly available DGA dataset [10]. The detection of DGA domains has gotten a lot of interest in recent years. This is a challenging task due to the ability of DGA domains to overcome blacklist filtration [31]. The features identified as influential in detection malicious domains are extracted from DGA domains. Then, the machine learning algorithms are trained using the extracted features.

Table 7 shows how the models perform for this dataset. Results show consistency with the previous obtained results using Marques et al. dataset [26]. Similarly, GBM show the best performance in terms of all measures and in particular,the lowest false positive rate of 15%. Better performance is obtained when the features are subjected to deep synthesis followed by PCA for dimensionality reduction. As shown in Table 8, the detection of DGA domains improves for the majority of machine learning algorithms across all measures.

4. Discussion

In many cases, malicious domain names are are linked to a range of activities that put the privacy and safety of individuals and organisations at risk. Analysing domain names data has been identified as one of the most significant and promising approaches to combat such attacks [35]. DNS traffic has become a prime option for experimenting with variety of machine-learning techniques in the security context due to the large number of attributes and large volume of traffic data available. The detection of malicious domain names using machine learning has many aspects, including: feature choice, feature representation, and the learning algorithm. The main questions that this research was designed to answer are: *(1) Does feature category affect the classification accuracy of domain names?; (2) what are the most influential features for the detection of malicious domain names ?;* and (3) *can the most relevant features be exploited using automatic feature engineering techniques to improve classification accuracy?*

Table 7

The performance measures for the five classification models trained using the DGA dataset with lexicalfeatures

Model	Accuracy	Precision	DR	F-measure	AUC	FPR
LR	63.98%	64.00%	64.00%	64.00%	64.03%	30.94%
KNN	66.27%	66.00%	66.00%	66.00%	66.25%	35.77%
GBM	68.92%	71.00%	69.00%	68.00%	69.07%	15.10%
GLM	63.98%	66.08%	58.81%	62.23%	64.03%	30.75%
SVM	65.20%	66.00%	65.00%	65.00%	65.35%	22.02%

Table 8

The performance measures for the five classification models trained using the DGA feature space generatedby DFS and PCA of lexical features

Model	Accuracy	Precision	DR	F-measure	AUC	FPR
LR	65.36%	67.00%	65.00%	65.00%	65.48%	21.87%
KNN	68.87%	69.00%	69.00%	69.00%	68.93%	25.52%
GBM	69.84%	72.00%	70.00%	69.00%	69.98%	15.95%
GLM	65.45%	72.04%	51.53%	60.08%	65.58%	20.38%
SVM	65.73%	68.00%	66.00%	65.00%	65.89%	16.81%

Although the use of traditional machine learning to detect malicious domain names is not new, the research questions addressed by this study have not been the primary focus of earlier research. Identifying significant features is difficult in other fields of research, however,it is extremely difficult in the field of malicious domainidentification [42]. New types of features continue to be available, while the influence of other features degradeover time [23]. Thus, researchers continue to study effective features that can be useful for detection of malicious domain names [15]. Traditional machine learningalgorithms continue to be used for this task [3].

In this study, several state-of-the-art machine learning algorithms are trained using three feature categories, namely, lexical features, host-based features, and a combination of both. We found that, regardless of classification algorithm, all models trained withlexically based features outperform host-based trainedmodels on all measures. This comes in agreement with the literature focusing on domain name features [25]. Ithas been demonstrated that training classification models using lexical features produce effective and efficientdetection models that are ideal for the proactive identification of harmful domains [25]. Lexical features are easier to collect than other types of feature and usually result in good performance [4]. Such classifiers can be applied in any context that contains domain names,such as: websites, email, chat, calendars, games or others, rather than being tied to a specific application [23].Selective lexical features appear to have a higher accuracy for identifying distinct forms of URL attacks. However, although successful in producing better accuracy, lexically based features can only operate for a limited time [8, 7]. According to Choi et al. [7], the useof lexical features results in decreased accuracy for thespam and malware URL dataset.

The trained classifiers using a combination of lexical and host-based features yield better accuracy measures. We believe that using a combination of features is more effective in detecting malicious domain names. This come in line with the findings of the literature where combining the host-based and lexical features into a single feature set yields the lowest classification error of any of the feature sets tested [14, 36, 24, 32, 23, 35]. Although this is not purely a novel finding, it has not been highlighted as the main topic of investigation in previous studies. We believe it is worth investigation as new datasets collected by new technologies become available. The dataset utilized in this study is a recent dataset that is intended to address the lack of malicious and non-malacious datasets based on DNS logs, which is critical in the field of cybersecurity. There are a large number of malicious domains that are registered every day and some of which are only active for brief periods of time. Thus, this investigation is a step towards the other contribution, in which we focus on the most influential features and harness them to improve classification results. Our findings demonstrate the advantages of ensemble learning, GBM in this study, over standalone classifiers. The GBM has been taught to distinguish between benign and malicious behaviour. The trained model was found to be effective in recognising malicious domain names with a high accuracy rate of 98.98% and a very low false positive rate of 0.003. The total execution time of the model is 112 s. GBM could be used to enhancing the accuracy of detection of harmful domain names in a timely manner and to develop a real-time DNS firewall.

Models that used lexical features required the least training time. According to [33], extra processing time and resources are required to extract the desired features from host-based, content-based, and popularity-based features, whereas lexical features require less computation and do not require access to any other sources. In comparison to the other models evaluated in this study, the KNN models were very slow. KNN is known as a *lazy classifier*, as it does not generalise across data in advance. KNN is slow when there are many observations. It works by reading the historical database each time a prediction is needed.

As regards the top 10 important features among the dataset that combines both categories, results show that the four measures agree on the importance of six features, five of which are lexical in nature. The six features are DNS Record Type, Numeric Sequence, Numeric Ratio, Strange Characters, Consonant Ratio, and Vowel Ratio. This supports our observation that all models trained with lexically based features outperform host-based trained models on all measures. It also confirms that combining both feature categories produces better results. Furthermore, using the six most relevant features to generate deep features showed improved performance. The comes in agreement with results recently reported in the literature that demonstrates the merits of DFS and PCA for cybersecurity research [2, 33]. Although the detection of DGA domains is challenging, the results of this study suggest that DFS and PCA could improve the performance of machine learning algorithms for the identification of domain names generated by a DGA or its variants. Because DGA domain names are unpredictable, research has used machine learning algorithms based on feature extraction to detect domain names [40].

5. Conclusions

DNS data analysis proved to be an effective and significant technique in detecting malicious domain names. This paper proposed a machine learning based model to detect malicious domain names. In order to build the model, an empirical study to evaluate various machine learning algorithms and feature categories was conducted. The goal was to improve the accuracy of malicious domain name detection. Five state-oftheart machine learning algorithms, namely, LR, KNN, GBM, GLM, and SVM, were trained independently on a recent DNS dataset using one of three feature categories: host-based, lexically based, or a combination of the two. We investigated the most influential features that enhance the detection of malicious domain name by weighing the value of feature importance. The GBM classifier showed better performance among other classifiers. With a supervised learning approach and a short training time, the model demonstrates a low false positive rate, a high detection rate, high precision, a high F-measure, and a high overall accuracy. We believe this will lead to more research in the field of ensemble learning, using GBM and similar algorithms, for the detection of various types of malicious domain name. Results also indicated that in comparison to employing lexical or host-based features alone, combining them yields the best accuracy results. The top identified features were DNS Record Type, Numeric Sequence, Numeric Ratio, Strange Characters, Consonant Ratio, and Vowel Ratio. Five out of the six identified features were lexical in nature, which is in line with the finding of the experiments that lexical features outperform host-based features. Prepossessing the dataset using deep feature synthesis showed classifications performance improvement. The model was verified by using a DGA dataset for the confirmation of the effectiveness of the proposed model in detecting unseen malicious domain names. Although, the detection accuracy varied between the two datasets used in this study, experimental results indicated that DFS and PCA could improve the performance of machine learning algorithms. A solution based on these findings can be implemented as a plugin in web browsers to protect users from previously unrecorded malicious domain names.

In the future, we plan to develop a real-time tool that learns the stream information of malicious domain names on a continual basis. We believe it should be capable of identifying zero-day attacks. The identified features will facilitate future research and the application development of malicious domain name detectionto improve web security. In addition, narrowing down the scope of malicious domains to focus on others kindsof malicious domain is also an important research direction. Finally, this research can be extended to address the explainability of machine learning algorithmsin malicious domain name detection, where there is a need to understand malicious activities. Explainable artificial intelligence is a new field of study that aims to help users and developers understand and interpret how machine learning methods make their predictions.

Acknowledgments

The authors are grateful to the Deanship of Scientific Research, King Saud University, for funding through Vice Deanship of Scientific Research Chairs.

References

1. Akhbardeh, A. and Jacobs, M. A. Comparative analysis of nonlinear dimensionality reduction techniques for breast MRI segmentation. Medical Physics, 2012, 39(4), 2275-2289. https://doi.org/10.1118/1.3682173
2. Al-Turaiki, I. and Altwaijry, N. A Convolutional Neural Network for Improved Anomaly-Based Network Intrusion Detection. Big Data, 2021, 9(3), 233-252. https://doi.org/10.1089/big.2020.0263
3. Almashhadani, A. O., Kaiiali, M., Carlin, D., and Sezer, S. MaldomDetector: A system for detecting algorithmically generated domain names with machine learning. Computers & Security, 2020, 93, 101787. https://doi.org/10.1016/j.cose.2020.101787
4. Blum, A., Wardman, B., Solorio, T., and Warner, G. Lexical feature based phishing URL detection using online learning. Proceedings of the 3rd ACM workshop on Artificial intelligence and security, 2010, Chicago, Illinois, USA, 54-60.
5. Chandrashekar, G. and Sahin, F. A survey on feature selection methods. Computers & Electrical Engineering, 2014, 40(1), 16-28. https://doi.org/10.1016/j.compeleceng.2013.11.024
6. Chiba, D., Yagi, T., Akiyama, M., Shibahara, T., Yada, T., Mori, T., and Goto, S. Domainprofiler: Discovering domain names abused in future. Proceedings 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2016, 2016, Toulouse, France, 491-502. https://doi.org/10.1109/DSN.2016.51
7. Choi, H., Zhu, B. B., and Lee, H. Detecting malicious web links and identifying their attack types. Proceedings of the 2nd USENIX conference on Web application development, 2011, Portland, Oregon, USA, 11.
8. Chu, W., Zhu, B. B., Xue, F., Guan, X., and Cai, Z. Protect sensitive sites from phishing attacks using features extractable from inaccessible phishing URLs. 2013 IEEE International Conference on Communications (ICC), 2013, Budapest, Hungar, 1990-1994.https://doi.org/10.1109/ICC.2013.6654816
9. Cortes, C. and Vapnik, V. Support-vector networks. Machine Learning, 1995, 20(3), 273- 297. https://doi.org/10.1007/BF00994018
10. DGA Domains Dataset. https://github.com/chrmor/DGAdomainsdataset.
11. Friedman, J. H. Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics, 2001, 29(5), 1189-1232. https://doi.org/10.1214/aos/1013203451
12. Friedman, M. A Comparison of Alternative Tests of Significance for the Problem of m Rankings. The Annals of Mathematical Statistics, 1940, 11(1), 86-92. https://doi.org/10.1214/aoms/1177731944

13. Google Safe Browsing Transparency Report. https://transparencyreport.google.com/safebrowsing/overview. 2021.

14. Hajaj, C., Hason, N., Harel, N., and Dvir, A. Less is More: Robust and Novel Features for Malicious Domain Detection. arXiv preprint arXiv:2006.01449, 2020.

15. Hara, D., Sakurai, K., and Musashi, Y. Classification of Malicious Domains by Their LIFETIME. Lecture Notes on Data Engineering and Communications Technologies, 2020, 334-341. https://doi.org/10.1007/978-3-030-39746-3_35

16. Iwahana, K., Takemura, T., Cheng, J. C., Ashizawa, N., Umeda, N., Sato, K., Kawakami, R., Shimizu, R., Chinen, Y., and Yanai, N. MADMAX: Browser-Based Malicious Domain Detection through Extreme Learning Machine. IEEE Access, 2021, 9, 78293-78314. https://doi.org/10.1109/ACCESS.2021.3080456

17. Jolliffe, I. T. and Cadima, J. Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 2016, 374(2065), 20150202. https://doi.org/10.1098/rsta.2015.0202

18. Kanter, J. M. and Veeramachaneni, K. Deep feature synthesis: Towards automating data science endeavors. 2015 IEEE international conference on data science and advanced analytics (DSAA), 2015, Paris, France, 1-10. https://doi.org/10.1109/DSAA.2015.7344858

19. Khalil, I., Yu, T., and Guan, B. Discovering malicious domains through passive DNS data graph analysis. Proceedings of the 11th ACM Asia Conference on Computer and Communications Security, 2016, Xi′an China, 663-674.

20. Liang, Y. and Yan, X. Using deep learning to detect malicious URLs. Proceedings IEEE International Conference on Energy Internet, ICEI 2019, 2019, 487-492. https://doi.org/10.1109/ICEI.2019.00092

21. Lin, M. S., Chiu, C. Y., Lee, Y. J., and Pao, H. K. Malicious URL filtering A big data application. Proceedings 2013 IEEE International Conference on Big Data, 2013, Silicon Valley, CA, USA, 589-596.

22. Liu, C., Wang, L., Lang, B., and Zhou, Y. Finding effective classifier for malicious URL detection. ACM 2nd International Conference on Management Engineering, Software Engineering and Service Sciences, 2018, Wuhan, China, 240-244. https://doi.org/10.1145/3180374.3181352

23. Ma, J., Saul, L., Savage, S., and Voelker, G. Learning to detect malicious URLs. ACM Transactions on Intelligent Systems and Technology, 2011, 2(3), 124. https://doi.org/10.1145/1961189.1961202

24. Ma, J., Saul, L. K., Savage, S., and Voelker, G. M. Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, Paris, France, 12451254.

25. Mamun, M., Islam, S., Rathore, M., Lashkari, A., Stakhanova, N., and Ghorbani, A. Detecting malicious URLs using lexical analysis. International Conference on Network and System Security, 2016, Taipei, Taiwan, 467-482. https://doi.org/10.1007/978-3-319-46298-1_30

26. Marques, C., Malta, S., and Magalhães, J. P. DNS dataset for malicious domains detection. Data in Brief, 2021, 38, 107342. https://doi.org/10.1016/j.dib.2021.107342

27. Nelder, J. A. and Wedderburn, R. W. M. Generalized Linear Models. Journal of the Royal Statistical Society. Series A (General), 1972, 135(3), 370-384. https://doi.org/10.2307/2344614

28. Netlab Opendata Project. http://data.netlab.360.com/dga/.

29. Palaniappan, G., Sangeetha, S., Rajendran, B., Sanjay, Goyal, S., and Bindhumadhava, B. S. Malicious Domain Detection Using Machine Learning on Domain Name Features, HostBased Features and Web-Based Features. Procedia Computer Science, 2020, 171(2019), 654- 661. https://doi.org/10.1016/j.procs.2020.04.071

30. Rapid7 Open Data | Forward DNS (FDNS). https://opendata.rapid7.com/.

31. Ren, F., Jiang, Z., Wang, X., and Liu, J. A DGA domain names detection modeling method based on integrating an attention mechanism and deep neural network. Cybersecurity, 2020, 3(1), 4. https://doi.org/10.1186/s42400-020-00046-6

32. Rupa, C., Srivastava, G., Bhattacharya, S., Reddy, P., and Gadekallu, T. A Machine Learning Driven Threat Intelligence System for Malicious URL Detection. The 16th International Conference on Availability, Reliability and Security (ARES 2021), 2021, Vienna, Austria, Article 154, 1-7. https://doi.org/10.1145/3465481.3470029

33. Saleem Raja, A., Vinodini, R., and Kavitha, A. Lexical features based malicious URL detection using machine learning techniques. Materials Today: Proceedings, 2021, 47(1), 163-166. https://doi.org/10.1016/j.matpr.2021.04.041

34. SANS Internet Storm Center. https://isc.sans.edu/index_dyn.html.

35. Shi, Y., Chen, G., and Li, J. Malicious Domain Name Detection Based on Extreme Machine Learning. Neural Processing Letters, 2018, 48(3), 1347-1357. https://doi.org/10.1007/s11063-017-9666-7

36. Sun, X., Tong, M., Yang, J., Xinran, L., and Heng, L. HinDom: A Robust Malicious Domain Detection System based on Heterogeneous Information Network with Transductive Classification. 22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019), 2019, Beijing, China, 399-412.

37. Tsai, F. Comparative study of dimensionality reduction techniques for data visualization. Journal of Artificial Intelligence, 2010, 3(3), 119- 134. https://doi.org/10.3923/jai.2010.119.134

38. Vanhoenshoven, F., Nápoles, G., Falcon, R., Vanhoof, K., and Köppen, M. Detecting Malicious URLs using Machine Learning Techniques. IEEE Symposium Series on Computational Intelligence (SSCI), 2016, Athens, Greece, 1-18. https://doi.org/10.1109/SSCI.2016.7850079

39. Vinayakumar, R., Soman, K. P., and Poornachandran, P. Detecting malicious domain names using deep learning approaches at scale. Journal of Intelligent and Fuzzy Systems, 2018, 34(3), 1355- 1367.https://doi.org/10.3233/JIFS-169431

40. Vranken, H. and Alizadeh, H. Detection of DGA-Generated Domain Names with TF-IDF. Electronics, 2022, 11(3), 414. https://doi.org/10.3390/electronics11030414

41. Vundavalli, V., Barsha, F., Masum, M., Shahriar, H., and Haddad, H. Malicious URL Detection Using Supervised Machine Learning Techniques. ACM International Conference Proceeding Series, 2020, Merkez, Turkey, Article 21, 1- 6. https://doi.org/10.1145/3433174.3433592

42. Zhauniarovich, Y., Khalil, I., Yu, T., and Dacier,M. A survey on malicious domains detection through DNS data analysis. ACM Computing Surveys, 2018, 51(4). https://doi.org/10.1145/3191329

This article is an Open Access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 (CC BY 4.0) License (http://creativecommons.org/licenses/by/4.0/).