National College of Ireland

Anomaly Detection-Based Approach for Identifying Domain Generation Algorithm (DGA) Domains in Cybersecurity

MSc Research Project
Cyber security

MAHESH KONI
Student ID: X23146931

School of Computing
National College of Ireland

Supervisor: Khadija Hafeez

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Mahesh Koni
Student ID:	X23146931
Programme:	Cyber security **Year:** 2024
Module:	MSc practicum 2
Supervisor:	Khadija Hafeez
Submission Due Date:	12/12/2024
Project Title:	Anomaly Detection-Based Approach for Identifying Domain Generation Algorithm (DGA)Domain in Cybersecurity
Word Count:	4070 **Page Count** 17

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	MAHESH KONI
Date:	12/12/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on a computer.	□

Assignments submitted to the Program Coordinator Office must be placed into the assignment box outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Anomaly Detection-Based Approach for Identifying Domain Generation Algorithm (DGA) Domains in Cybersecurity

Mahesh Koni
X23146931

Abstract

There has been a constant innovation in cyber-attack techniques, and Domain Generation Algorithms (DGAs) appear to be one of the most effective ones. DGAs enable malware to create many domain names, making it dynamic, constantly changing, and difficult to tap on the shoulder and tell it to stop. Apart from helping malware evade detection programs, it also helps create a random and reliable connection with the C&C servers, making it even harder to detect a botnet connection. Contemporary malicious software perpetually employs DGAs in effort to prevent its C&C domains or IPs from being seized, where affected systems try to connect with as many domains as possible until a connection with the C&C server is established. Therefore, detecting DGA domains is another important factor which can be automatically solved to prevent sending malicious traffic and define compromised hosts. However, many simple DGAs create domain names that appear like English words, thus making it easy for a manual check to be overwhelmed. To this end, we integrate different domain features to improve the identification of suspicious domain names. Domain Parameters: length, presence of numbers, entropy Features like length of domain names, the ratio of unique characters, including numeric characters, and entropy give indications of Domain Generation Algorithm behavior. Subsequently, these features are used to train machine learning models for domain categorization, as legitimate or generated by DGA. Feature engineering and high-level skilled machine learning techniques will enable an effective and efficient way of differentiating DGA-generated domains accurately and efficiently.

1 Introduction

The growth of the Internet over recent decades has established cyberspace as the main platform for the exchange of information in the international context across most areas of human activity. However, this overdependence has brought various problems especially the vulnerability of key Internet infrastructure to cyber-attack. The domain name system or DNS, an Internet system that translates the domain names to IP addresses is often the object of hacker's attack. DNS helps work with Web Sites, E-mail and other distributed computing services and facilities. While conventional DNS services have been adopted, they have also the ability to penetrate firewalls and therefore become an instrument for hackers (Antonakakis et al., 2012). Cybercriminals in various way

use DNS, for instance, in controlling malware via command and control (C2) servers. As a result, the protection of DNS operations remains an important goal in the securing of cyberspace (Plohmann et al., 2016).

corebot	ep16g6gjwfixyhs8gfy.ddns.net
	ev5texifc43nebil3pk.ddns.net
	gf7bm4163fmjkje.ddns.net
cryptolocker	agryjvdaabkyt.ru
	pwitjnqgjfaqm.org
	dhhubfepcdgfv.co.uk
dircrypt	hedhryendqlss.com
	lgnggnlufbtyjpnvct.com
	tzrbdmhoumoy.com
kraken_v2	fwulvdmdytm.com
	gybuisybe.cc
	gyinkvye.net
lockyv2	btlwubflhfllshn.info
	cpgcjsysfwuwa.click
	jlbroeji.biz
pykspa	gqjgflhop.net
	gqumcwaa.org
	jpivjh.net
qakbot	fgfifkyfut.info
	flzuzsaekkipatbtet.biz
	owpbsjekk.com

Figure 1: Examples of DGA Algorithms

DNS is used in today's botnets and ransomwares to initiate communication with C2 servers for transferring files and updating the malware. For this purpose, malware uses domain names to connect to C2 servers when the names are dynamic. Earlier many malwares contained IP addresses or domain names which were easily identified and blocked easily.
Nonetheless, to avoid blocking and make malware more credible, hackers use so-called Domain Generation Algorithms (DGAs) to generate pseudo-random domains on the fly, thus remaining in touch with the C2 servers (Curtin et al., 2019). The identification of Domain Generation Algorithms (DGA) behind a domain name is equally important in dealing with such malware (Geffner, 2013; Yu et al., 2018).

The general research in identifying DGA domains has gone through growth, from the case where features are manually extracted from domains all the way up the machine learning phase, and more recently through the deep learning phase. The first machine learning models used in the identification of DGAs were based on static rule-set features and can only accommodate a fixed amount of DGA types and are thus not effective in identifying wordlist-type of DGAs (Kumar et al.). For instance, the suppobox DGA domains as middleapple.net may seem less risky than oewvdjhwxkwdr.com — the domain generated by the Locky DGA. This difference illustrates the shortcomings of traditional ad hoc mechanisms, especially in terms of identifying phony sounding domains derived from word lists or language patterns (Anderson & Weaver, 2016).

To overcome these challenges, this paper presents an enhanced Anomaly Detection-based approach that leverages machine learning algorithms to identify DGA domains especially those produced by wordlist-based-DGAs. Our model focuses on datasets that have names that seem legitimate but capture statistical features that are likely to be DGA patterns when fed into machine learning algorithms with statistical features as inputs. Further, this approach is endeared to improve the overall detection accuracy and operate against the dynamic nature of techniques they use in developing malware (Feng et al., 2017; Mohan et al., 2018).

2 Related Work

A Domain Generation Algorithm (DGA) is used by malware families to create domains for the C&C (Command and Control) server and create a pseudo-server that requires no pre-programmed server IP address. DGAs work together in the increment of an input number to generate a string attached with domain extensions such as .com or Net (Plohmann et al., 2016). Newer types of DGAs are known as the word list where domain names are being generated with reference to one or more than one word of a word list. However, as recent malware called Matsnu involves use of a list of more than 1300 words to create 24 characters long strings and Suppox constructs domain names like heavenshake.net by joining pseudo randomly selected English words, the problem is not easily solvable as they work by mimicking legitimate domain names. Recent wordlist-based DGAs have been designed to mimic legitimate domain names which hardens their detection (Zhauniarovich et al., 2018).

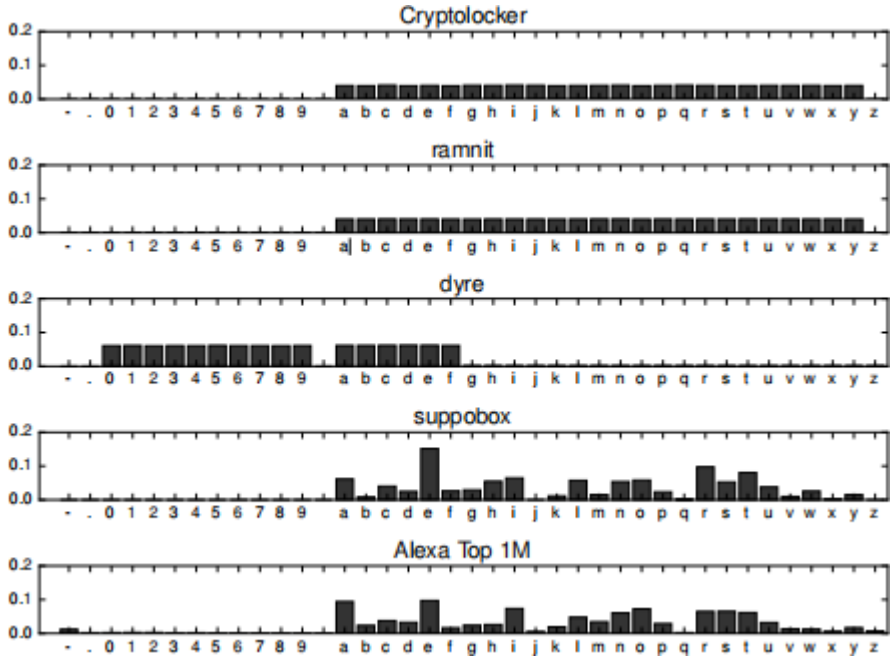


Figure 2; The unigram distributions for the Crypto locker, Ramnit, and Dyre DGAs (which are simple character-based), the Suppobox DGA (which is dictionary-based), and the Alexa top one million domains, were analyzed. These distributions provide insight into the structure of domain names generated by each of these DGA types, with the Alexa list serving as a benchmark for comparison (Yadav et al., 2012; Zhauniarovich et al., 2018).

2.1 Early Detection Methods

Domain name identification, particularly separating phony domain names created by DGAs from actual domain names, has been a focal area of analysis. Initial models employed statistical machinery to understand the geometric and dynamical properties of domain names. For example, Yadav et al. (2012) utilized time correlation and entropy features of both successful and failed domains. Similarly, Antonakakis et al. (2010) applied domain clustering that included the length of the domain, randomness, number of characters, and n-gram distributions. Another advancement was made with the employment of Hidden Markov Models (HMMs) for the estimation of a domain being created by a DGA. Nevertheless, these methods failed on more complex DGAs where domain names resembled real words (Zhauniarovich et al., 2018).

2.2 Advances with DNS and Traffic Analysis

Over the years, DNS data has been increasingly used in research related to DGA detection. Zhou et al. (2017) used DNS NXDOMAIN data from recursive DNS servers to identify domains produced by DGAs. Similarly, Jian et al. (2010) utilized DNS traffic analysis to cluster anomalous domains using lookup graph decomposition techniques. These approaches were useful in identifying threats that were invisible to other traditional security solutions but did not address the main problems with wordlist-based DGAs, which resemble legitimate domains (Curtin et al., 2018).

2.3 Machine Learning-Based Detection Approaches

Multi-level analysis of DGA detection has revealed that modern ML methods have made substantial contributions by overcoming some of the drawbacks of traditional methods. Hamed et al. (2014) used the Phoenix framework, which employs statistical features such as the pronunciation measure, blacklist information, and DNS query values to detect DGA domains. However, Phoenix relied on older methods, and there was potential for improvement. To enhance detection performance, additional features such as entropy and n-grams were incorporated (Tong & Nguyen, 2015).

Zhao et al. (2016) presented a large-scale ML-based method for DGA botnets that uses features like TTL, IPs, and WHOIS data with the J48 decision tree algorithm to compute domain reputation scores. Luo et al. (2014) developed a method that achieved 93% accuracy for identifying malicious domains using lexical patterns derived from legitimate domains and machine learning techniques. Despite such successes, challenges persist, and some of the key issues with current approaches include false positives when identifying wordlist-based DGAs from legitimate domains (Curtin et al., 2018; Zhauniarovich et al., 2018).

2.4 Deep Learning Approaches

Deep learning approaches have been receiving attention because it can diagnose DGA domains accurately. Woodbridge et al. (2016) explained that the DGA domains are easily detected by LSTM networks at character level. Mac et al. (2017) proposed an LSTM- based model for identifying botnet generated domains. Saxe and Berlin (2017) expanded CNN to DGA and discovered it enhances accuracy in detection. However, the most recent deep learning models have giant difficulties when establishing wordlist-based DGAs that generate domains that resemble true English words, leading to incomparably higher false positive rates (Anderson et al., 2016). Furthermore, these models depend on the availability

of labeled data and are computationally expensive; they are not particularly useful for small datasets or RTD applications (Zhao et al., 2016).

2.5 Recent Developments for Wordlist-Based DGAs

The most relevant recent studies are dedicated to enhancing the identification of wordlist-based DGAs. Yang et al. (2018) put forth a random forest classifier using the features like the words' frequency, POS tags, and correlation for discovering wordlist-based DGA domains. Patil and Dharmaraj (2018) introduced a multiclass classification model to demarcate between malicious URLs and categories the type of attacks. Pereira et al. (2018) presented the Word Graph method in which they enhance the identification rate of the dictionary based DGAs compared to the conventional approaches. Some extra characteristics of DGA were introduced by Curtin et al. (2018) that include "smash word", which is the ratio of the number of domains generated by a DGA and the number of English words; the second strategy is based upon an RNN model that is used to detect complicated DGA families, for instance, Matsnu and Suppox.

2.6 Limitations in DGA Detection Techniques

- **False Positives:** A lot of detection techniques, especially the machine and deep learning-based techniques, suffer from high false positives. For instance, models like LSTMs or CNN may label corresponding legitimate domains generated by DGAs due to the resemblance of applicable attributes, of which, consistency of string sequences (Anderson et al., 2016). This problem is more common in wordlist-based DGAs which create domains like real English words (Curtin et al. 2018).

- **Scalability:** Although some models, for example, decision trees or random forests, proved high efficiency, they may face difficulties with the data scaling problems in terms of application to vast datasets. These detective models need a large amount of training and as volume of DNS traffic increases time and computational resources to detect increase also (Zhao et al., 2016). Second, another limitation of most emerging DGAs is that before the model remains viable indefinitely, it needs to be continuously trained, which can create other issues regarding scalability.

- **Real-time Detection:** Most of the existing methods for DGA detection, and especially those based on deep learning, are time consuming and thus can be unfavorable to identify malicious domains in real-time detecting during live traffic. Several models such as LSTMs and CNNs take considerable time before it can perform analysis and classify the domains, which is suitable for threat detection and mitigation (Zhauniarovich et al., 2018).

- **Feature Extraction Challenges:** Compared to wordlist-based DGAs, containing strict lexical patterns is highly informative when detecting such DGAs, there is a need for more complex feature extraction. While the existing techniques focus on the domain structure or DNS patterns, the wordlist-based DGAs necessitate finer features like, Word Semantic Analysis (Yang et al., 2018). These techniques are still evolving and one of the largest challenges remains to determine the ideal balance resulting in both high accuracy of detection and reasonable computation time.

2.7 Critical Analysis

DGA detection has enhanced with statistical analysis, clustering techniques, and machine and deep learning techniques but each come with their own flaws. For example, deep learning models lack efficiency with wordlist-based DGAs, thereby increasing the FP rate. Further, most of these methods rely on the labeled data and are complex processes which make them non-real-time solutions (Curtin et al., 2018). More works on feature abstract, other information apart from registered domains data, and other methods of detection can improve detections of DGA, especially for more complex and based on wordlists (Yang et al., 2018; Shibahara et al., 2016)

3 Research Methodology

3.1 Aim

This study's research aim is to produce a well-balanced classifier for identifying new domains generated by DGAs and differentiation from normal domains. DGAs are used by malware to construct several domain names from which one connects to the active C&C server of a botnet or some other malicious thing. And such generated domains tend to contain patterns which make them different from usual, genuine domain names. To this end, we use several machine learning techniques to translate domain names and identify those generated by DGAs. It is used for feature extraction, model selection, model training, and model validation to ensure that the classifier to be developed is suitable for further deployment and use to be able to classify DGA domains on the fly immediately in live traffic flow.

3.2 Data

For this research, three datasets were used to compile an overall training and validation dataset. Benign Domains was collected from the Alexa Top 1 million Sites (the data is available at Kaggle) and includes domains that were considered safe and reputable among web users. The second and third datasets focus on DGA Domains: The sample malicious algorithmically generated domains. We used a balanced dataset that encompassed both legitimate domain names and domain names crafted by malware-infected machines using various kinds of DGA algorithms. This dataset is used to train and evaluate the proposed detection models as follows. It is our objective for this approach to create a powerful and reliable model in interpreting whether a domain is malicious, created by DGA systems.

3.3 Data Preprocessing

Pre-processing is an important process in most machine learning and is used to prepare raw data for the models. To carry out the actual preprocessing of domain names used machine learning techniques, the following preprocessing was carried out in this study: Below are the preprocessing steps applied to the dataset:

a. **Number of Characters:**

This feature translates into the total size of each domain name throughout its arch length. The length of a domain name is also useful for understanding its structure, as was mentioned

DGA-created domain names are less or more than drawn domains. From the study it was found that the domains generated by DGAs are longer and more random as compared to the real ones.

b. **Unique Character Ratio:**

This feature finds out the fraction of unique characters to the total number of characters in domain name. A high Unique character ratio might mean that the domain name was random or taken from a generator or an algorithm, while the low range of Unique character ratio could point to well-chosen and meaningful domain.

c. **Number of Vowels (Vowels) and Consonants:**

These features calculate the number of vowels and consonants in each of the domain names. It is possible to sometimes get some information with the origin of certain domain names based on the understanding of linguistic patterns associated with it. Names created by the DGA of distinct classes may contain different distributions of vowels or consonants compared to regular names.

d. **Percentage of Numeric Characters:**

This feature establishes the proportion of the numeric character string in the domain name. Numbers can be used in the domains created by DGA to make it more random. They said that a larger percentage of numeric characters than the average may be the sign of a domain generated by a DGA.

e. **Entropy:**

Entropy is a measure of how likely the string of characters comprising a given domain name is to be random in terms of meaning. This feature makes an additional column and sums up the entropy of each domain name. Entropy has higher values when the strings are more random or can be considered as potentially malicious and that is why entropy belongs to the characteristics of DGA domains.

f. **N-grams and Similarity:**

These features estimate the degree of similarity of each domain name to a group of 3-grams and 4-grams containing strings of three and four consecutive characters in a list of real domain names. This comparison makes it easier in differentiating the new registered names and related names to the legitimate domains hence helping in differentiating between safe and risky names.

g. **Number of Dots:**

This feature measures the degree of domain string density and that is done by counting the number of dots that is “.”The randomly generated domain names by DGA contain more dots compared to actual domains as most of the domain names follow certain naming hierarchy such as single level domains.

h. **Number of Consecutive Vowels:**

This feature counts the number of vowels in between two consonants within the string of each domain name. It observes extended continual sequences of vowels which may not be present in genuine domains; such constructs may assist in identifying abnormally that could point to a DGA.

i. **Longest Sequence of Consonants:**

This feature identifies the longest sequence of consecutive consonants in each domain name. Like the consecutive vowels feature, longer sequences of consonants may indicate randomness in DGA-generated domains.

j. **Label Encoding:**

This step turns categorical variables into numbers so that they are in a format acceptable by machine learning models.

3.3.1 Importance of Preprocessing Steps

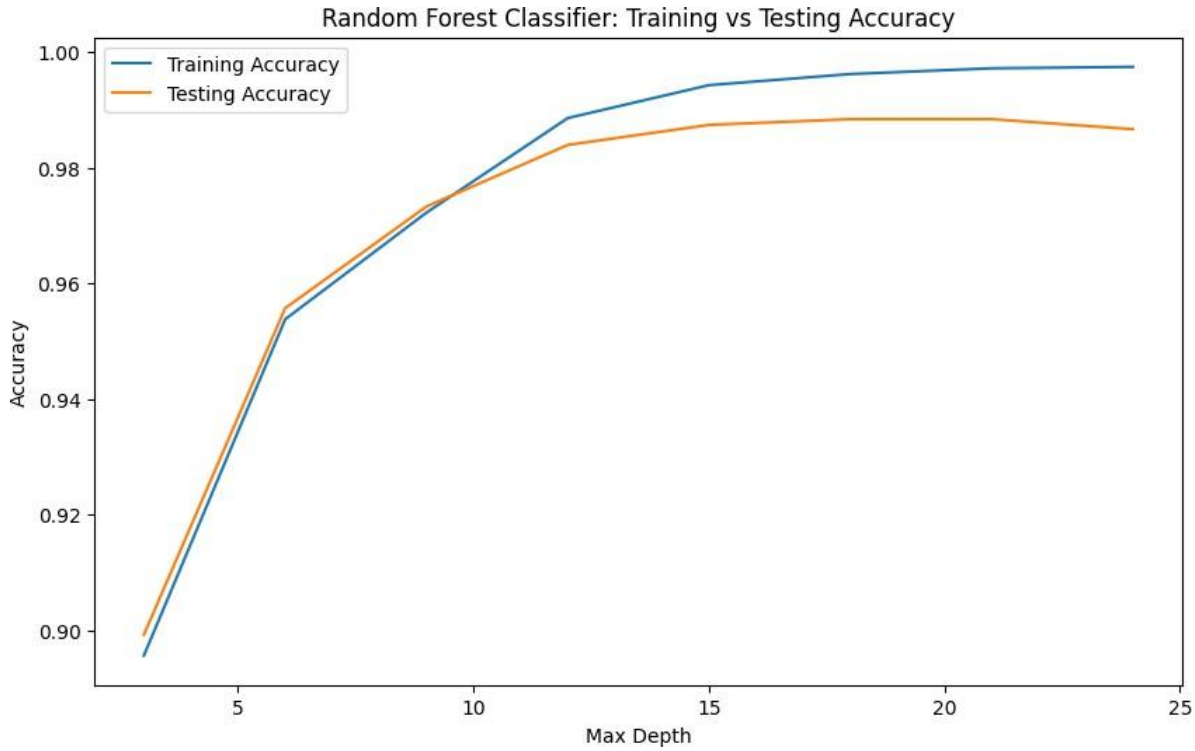
Every preprocessing is crucial for revealing patterns that are specific to domains that are created by DGA and distinguishing algorithmically created domains from the real ones. Thus, having adapted raw domain data and prepared them as features that capture randomness, structure, and non-standard patterns of language use, the data is optimally preprocessed for machine learning classification. This exhaustive dynamic feature engineering improves the DGA domains identification and classification performance of the model.

4 Design Specification

For detecting DGA (Domain Generation Algorithm) domains, three machine learning models were chosen: Random Forest, Linear Support Vector Classification, and Light Gradient Boosting Machine. Each of these models was adopted due to the suitability of the models in classification problems and the suitability to deal with the DGA domains in anomaly detection.

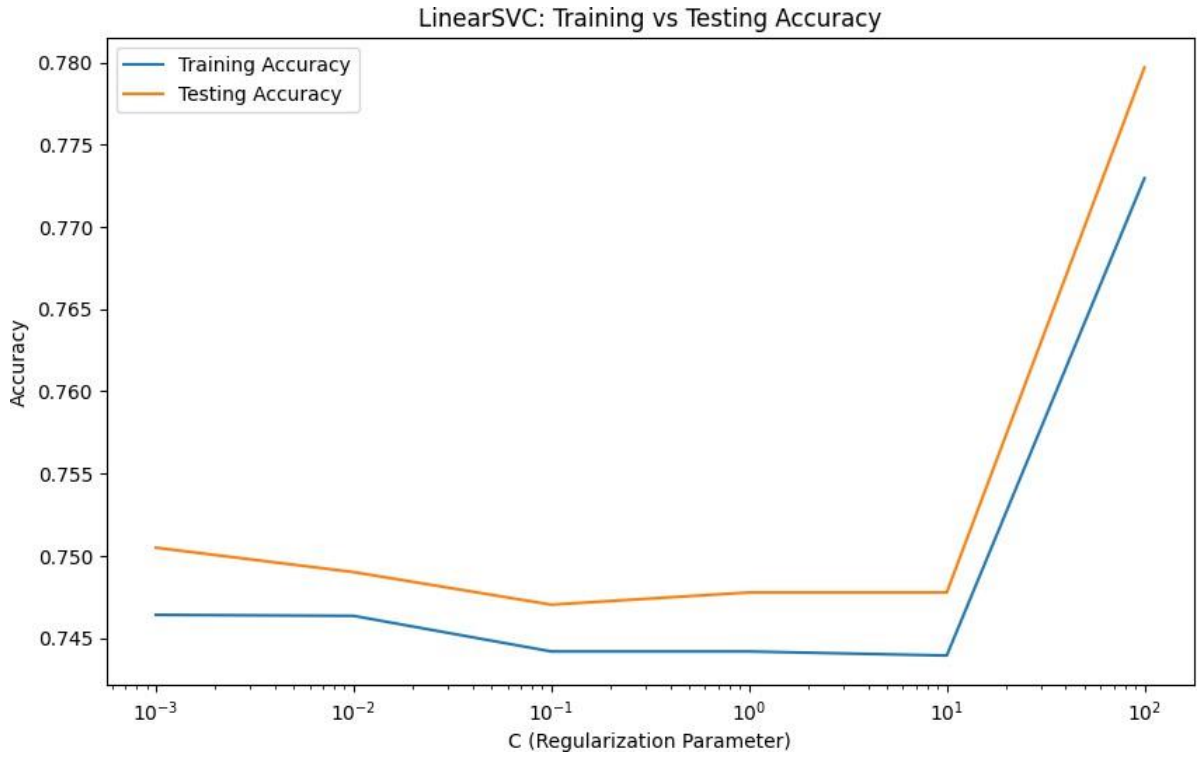
Random Forest:

Random Forest is an ensemble learning method consisting of creating multiple decision trees while training and given the mode of the classes (classification) or mean prediction (regression) of the separate trees. It is a powerful model used for modeling high dimensional data and for identifying non-linear relationships of DGA domains for classification. Also, Random Forest is good in avoiding overfitting for example, as pointed out earlier Cross-Validation and Random Forest helps in avoiding over-fitting since it includes an element of averaging.



Linear Support Vector Classifier (LinearSVC):

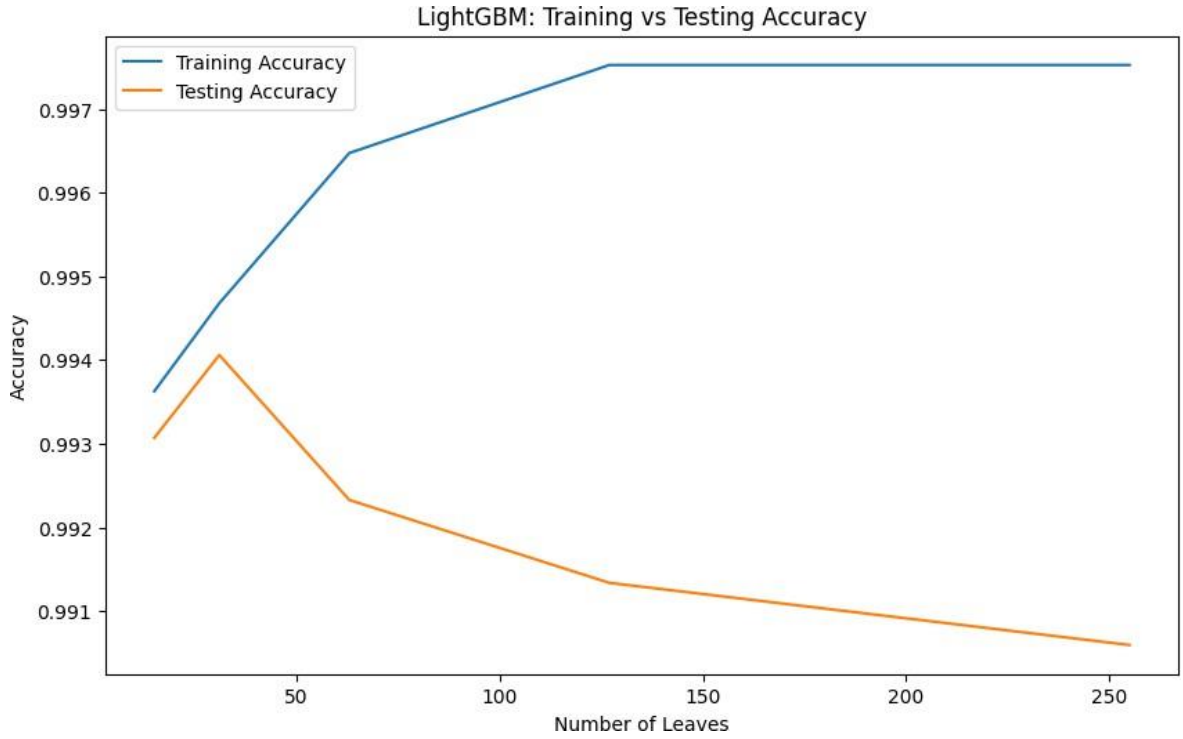
LinearSVC is a classification method under Support Vector Machine (SVM). It tries to locate a hyperplane that suits the classes as far as possible in the feature space. It is suitable for the simplest problems of distinguishing between two classes if the classes are well separated. In the case of anomaly detection LinearSVC is useful in trying to find the separation line between legitimate domains and DGA domains which makes it useful in detecting the normal and the abnormal behavior.



Light GBM :

Light GBM is a gradient boosting framework for gaining knowledge regarding trees. It can be optimized for big data and the features even have the availability of the regression and classification. It uses histogram based, which makes it better than other methods of doing gradient boosting and secondly, it is way better than all those methods when it comes to computational speed and the good part is, it does not affect the accuracy. As for anomaly detection, Light GBM is created that can solve a large number of complicated decisions making and restore interactions between features especially for the identification of the DGA domains.

Each of the models comes with its advantages to solve the problem of the DGA domain detection task, including feature selection, as well as working through high dimensionality requirements and minimizing the effects of predicting the model when there are imbalanced classes.



5. Model Training and Evaluation

The training process comprised several critical stages to enhance and move toward perfection in categorizing domains into DGA and non-DGA kinds. This entailed learning on a small test set with character statistics such as number of characters, number of unique characters, entropy of test sample, and n-gram cross correlation matrices trained on a large sample of data split into a test and training set. In this work, the process of setting the hyperparameter for each model was done through GridSearchCV, which adjusted parameters, examined different combinations of the values and selected the one with the best cross-validation. This process adjusted model-specific parameters:

- **Random Forest:** we adjusted the number of estimators, the depth of individual trees, and minimum sample size used to split data into new branches.
- **Linear SVC:** we further tuned the regularization parameter so that the size of the margins should be large and also should avoid over fitting.
- **Light GBM**: parameters adjusted were the number of leads, learning rate, and boosting iterations. These tuning efforts were aimed at fine-tuning of the model for one specific task, namely classification.

6. Results Analysis

The dataset revealed the distribution of legitimate and DGA-labeled domains, with varying frequencies across different DGA types. The visualization of DGA type counts highlighted the prevalence of certain types over others, which could potentially influence model training by introducing class imbalance.

Feature Engineering and Data Preprocessing

To enhance the model's ability to detect DGA domains, several features were engineered:

- **Character-based Features**: Character-based Features: The count of the character and the ratio of unique characters gave an understanding of how random each of the domains was and acted as a basis for determining which domains were legitimate and which were generated algorithmically.
- **Linguistic Features**: Linguistic pattern of the acronym-focused, such that many DGAs do not include actual human language words, were measured by vowel and consonant occurrence, consecutive vowels occurrence, and longest consonant sequences.
- **Entropy**: When entropy is high in domain names the strings created are random which is a feature of many DGA domains.
- **N-gram Similarity**: Using the Jaccard similarity scores on the 3-grams and 4-grams against legitimate domains allowed for the quantitative assessment of distance and detect structural abnormalities in DGA domains.

All these features together and in a way wanted to make the given dataset more appropriate for anomaly detection by converting the domain data into numerical format of values.

Discussion

The results of the experiment also reveal that machine learning models including Random Forest are accurate in differentiating between real domain and DGA domain. Due to the high entropy and random characteristic, or unpredictable value of DGA domains, these features stand out as high significance in classification tasks. Surprisingly, LinearSVC received a slightly lower accuracy than Random Forest, but it still could operate sufficiently for real-time detection where computational December 25, 2015, Introducing sophisticated techniques with a linear time complexity made it possible to achieve a high result without affecting the speed. Light GBM was even more accurate than Random Forest but slightly slower concerning precision, demonstrating the remarkable challenge in developing both complex models and fast performance.

Random Forest Classifier Performance

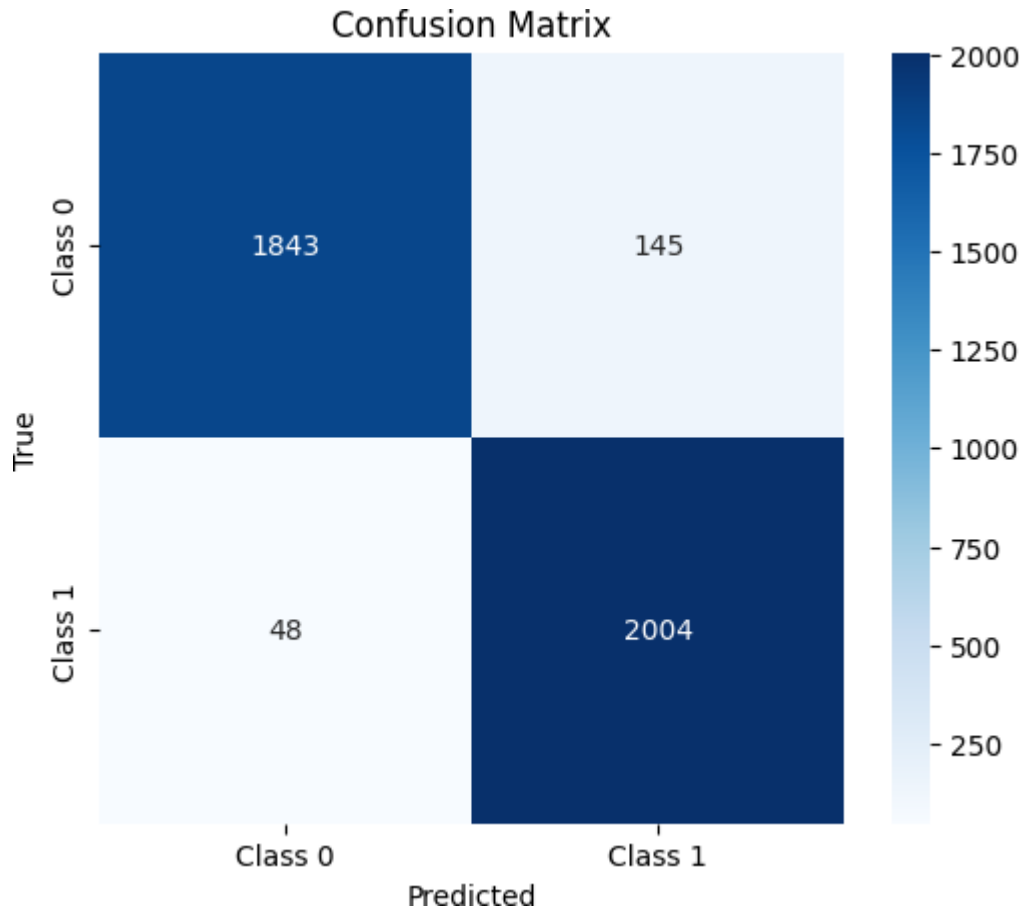
This Random Forest classifier was made optimal by GridSearchCV using parameters like number of estimators, maximum features as well as depth of the trees. Overall, the last estimated model shows the accuracy score that reached ~96.8% to test dataset. The classification report and confusion matrix provided additional insights:

- **Precision**: The model yielded accuracy of 97% on Class 0 (legit domains) and about 93% on Class 1 (DGA domains) which suggested one out of a thousand legit domains were misclassified.
- **Recall**: The proposed model has a good detection rate for both the DGA domains and the legitimate domains whereby the recall values were 0.98 in DGA domains and 0.93 in legitimate domains.
- **F1 Score**: In both classes, our method came out with an F1 score of 0.95, which indicates good balance between precision and recall.

Confusion Matrix

An analysis of the confusion matrix revealed high true positive numbers together with low false positive and false negative numbers to prove the high classification ability of the model.

- **True Positives (DGA Domains)**: The model accurately identified 2004 out of the total DGA domains.
- **True Negatives (Legit Domains)**: Similarly, 1843 legitimate domains were correctly classified.
- **False Positives** and **False Negatives** were minimal, suggesting the model's capacity to generalize well without excessive overfitting or underfitting.



Key Observations

- Entropy and N-gram Similarity as Predictors: The features of high entropy and low N-gram similarity was the most discriminative when comparing DGA domains and non-malicious domains and asserted the value of feature engineering for DGA detection.
- Imbalance in DGA Types: Some DGA types were missing, which, in turn, may affect generalization to all the DGA types, for the chosen model. More detailed balance strategies or even better, directed sampling, could positively affect the model’s quality in the case of disclosing lesser-known DGAs.

- Model Precision and Recall Balance: The model was able to keep up well with the precision and recall values ideal in cybersecurity to reduce both false positives that may lead to alert fatigue and false negatives that may leave malicious domains unnoticed.

Conclusion and Recommendations

The study proves the possible application of supervised learning methods to identify DGA domains. Out of all the models tested the Random Forest classifier is considered as the best model yielding almost equal accuracy, precision as well as recall rates. The flow diagram of dataset feature engineering shows that entropy, N-gram similarity, and linguistic features contributed significantly to the improvement of accurate detection.

The discoveries presented in the study reveal the applicability in cybersecurity, with an emphasis on the threats arising from the malware's utilization of DGAs. Future work may explore:

- Apartment from using CNN or ResNet-like architectures for image-based feature learning, developing complex LSTMs or Transformer models for sequence-based feature learning.
- Expanding the proposed model to address new patterns of DGA and using supervised learning for retraining the algorithm by feeding it current data in real time.
- Real-time use of the classifier is important within a live network environment for detection and prevention.

References

Anderson, H. S., Woodbridge, J., & Filar, B. (2016). DeepDGA: Adversarially-Tuned Domain Generation and Detection. In *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security* (pp. 13–21). ACM.

Antonakakis, M., Perdisci, R., Nadji, Y., Vasiloglou, N., Abu-Nimeh, S., Lee, W., & Dagon, D. (2012, August). From throw-away traffic to bots: Detecting the rise of DGA-based malware. *USENIX Security Symposium*.

Berman, D. S., Buczak, A. L., Chavis, J. S., & Corbeil, C. L. (2019). A survey of deep learning methods for cybersecurity. *Computer Science Review, 12*, 1–122.

Chen, G., Ye, D., Cambria, E., Chen, J., & Xing, Z. (2017). Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. *Proceedings of the 2017 International Joint Conference on Neural Networks*, 1–6.

Curtin, R. R., Gardner, A. B., Grzonkowski, S., Kleymenov, A., & Mosquera, A. (2019). Detecting DGA domains with recurrent neural networks and side information. *arXiv e-prints*, arXiv:1810.02023.

Feng, Z., Shuo, C., & Xiaochuan, W. (2017). Classification for DGA-based malicious domain names with deep learning architectures. *Proceedings of the 2017 International Conference on Network and System Security*.

Geffner, J. (2013). End-to-end analysis of a domain generating algorithm malware family. *Black Hat USA 2013*.

Gomaa, W. H., & Fahmy, A. A. (2013). A survey of text similarity approaches. *International Journal of Computer Applications*, 55(7), 13–18.

Kumar, A. D., Hodupunoori, H., Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2019). Enhanced Domain Generating Algorithm Detection Based on Deep Neural Networks. *Proceedings of the 2019 International Conference on Machine Learning and Data Science*.

Mohan, V. S., R, V., Kp, S., & Poornachandran, P. (2018). S.P.O.O.F Net: Syntactic Patterns for Identification of Ominous Online Factors. *Proceedings of the 2018 International Conference on Cybersecurity*.

Plohmann, D., Yakdan, K., Klaß, M., Bader, J., & Gerhards-Padilla, E. (2016). A comprehensive measurement study of domain generating malware. *Proceedings of the 25th USENIX Security Symposium*, 263–278.

Yu, B., Pan, J., Hu, J., Nascimento, A., & De Cock, M. (2018). Character level-based detection of DGA domain names. *International Symposium on Research in Attacks, Intrusions, and Defenses*.