Detecting Algorithmically Generated Domains using Entropy and Lexical Features

Jinsu Ann Mathew
Department of Physics
Newman College (Affiliated to Mahatma Gandhi University)
Thodupuzha, Kerala, India.

Ninan Sajeeth Philip
Artificial Intelligence Research and Intelligent Systems (airis4D)
Thelliyoor, Kerala, India.

Joe Jacob
Department of Physics
Newman College (Affiliated to Mahatma Gandhi University)
Thodupuzha, Kerala, India

ABSTRACT

Detecting domain names generated by Domain Generation Algorithms (DGAs) is a key challenge in cybersecurity, as these domains are designed to appear unpredictable and evade standard filtering methods. This work proposes a lightweight and interpretable detection method that relies on lexical properties and entropy-based features derived from domain names. By analyzing character patterns and measuring randomness through Shannon entropy and relative entropy across bigrams, trigrams, and fourgrams, the method captures both structural and statistical differences between legitimate and algorithmic domains. Multiple machine learning classifiers were trained and evaluated, with the best results achieved using XGBoost and Random Forest. Entropy-based features were found to be highly influential in the classification process, highlighting their effectiveness in distinguishing algorithmically generated domains. The findings support the use of entropy as a practical and theoretically grounded feature for DGA detection.

General Terms

Algorithms; Security; Experimentation; Performance

Keywords

Domain Generation Algorithm (DGA), Entropy-based features, Lexical features, N-gram analysis.

1. INTRODUCTION

The Domain Name System (DNS) is a foundational component of the internet, translating human-friendly domain names into IP addresses. It allows users to access websites using easy-to-remember names, such as ‘example.com’, rather than numerical addresses. [1–3]. While DNS enables ease of use, it is also frequently exploited by malware to evade detection. One of the most common techniques used by malware authors is the implementation of Domain Generation Algorithms (DGAs) [2,4–6].

A DGA is a program that creates many domain names automatically. These domains are used by malware to communicate with a command-and-control (C&C) server, which controls infected computers (called bots) [2,5,7]. By using different domains every day, attackers make it hard for defenders to block communication using blacklists. This technique, called domain fluxing, enables malware to avoid disruption by regularly altering its domain names [5,8,9].

Because DGA domains change frequently and can be generated in large numbers, detecting them is an important task in cybersecurity [10–12]. Traditional methods such as blacklists or reverse engineering are not fast or flexible enough to deal with new and unknown DGA domains [13,14]. As a result, many researchers have started using machine learning to detect these domains based on patterns found in the domain names themselves [3,8,13,15–19].

DGA-generated domain names often differ structurally from legitimate, human-generated domains [1,4]. While legitimate domains are typically influenced by linguistic norms, branding considerations, and ease of memorization, DGA domains—especially those that are randomly generated—tend to lack these characteristics. They often appear as meaningless sequences of characters with higher randomness, making them detectable through statistical analysis [2,5,11,20]. This study focuses on identifying such differences to classify domain names as either malicious (DGA) or legitimate (Non-DGA) using machine learning techniques.

In this study, two primary categories of features are employed for DGA detection: lexical features and entropy-based features. Lexical features capture structural properties of a domain name, including its length, the number of digits, vowels, consonants, and unique characters [11,21,22]. Entropy-based features, on the other hand, quantify the degree of randomness or unpredictability in character sequences [3,12,23–25]. Shannon entropy is used to measure the overall randomness within a domain, while relative entropy (Kullback–Leibler divergence) is applied to compare the character distribution of DGA-generated domains with that of legitimate domains [7,8].

The primary objective of this study is to determine whether a given domain name has been generated by a DGA using only the characters contained within the domain string, without reliance on external sources such as DNS logs or WHOIS records. To achieve this, a balanced dataset was constructed by combining legitimate domains from Alexa’s top list with malicious domains obtained from a publicly available DGA dataset. From these domains, a set of lexical and entropy-based features was extracted, which served as input for multiple machine learning classifiers. The models evaluated include Random Forest, Support Vector Machine (SVM), Logistic Regression, XGBoost, and K-Nearest Neighbors (KNN).

This paper is organized as follows: Section 2 explains how the dataset was created and what features were used. Section 3 describes the machine learning models and outlines the procedure followed for their evaluation. Section 4 presents the results and analysis. Section 5 gives the conclusion and future work.

2. DATASET AND FEATURE EXTRACTION

This section describes how the dataset was prepared and the features that were extracted from domain names for use in the machine learning models.

2.1 Dataset Collection and Preparation

A labeled dataset consisting of both legitimate (Non-DGA) and malicious (DGA) domain names was prepared for developing and evaluating the DGA detection system.

For the malicious class (DGA), a publicly available dataset compiled by J. Selvi (2019) [13] was employed, containing approximately 32,000 domain names generated by Domain Generation Algorithms. These domains encompass a wide range of character patterns and are commonly used in academic research for evaluating DGA detection methods.

To construct a balanced dataset, an equal number of legitimate (Non-DGA) samples were drawn from the Alexa Top 1 Million list [26]. Specifically, the first 32,000 domains were selected, as Alexa ranks websites based on popularity and traffic, and domains appearing at the top of the list are generally considered trusted and representative of real-world, human-generated domain names. These domains were therefore regarded as safe and suitable for use as negative samples in the binary classification task.

Before model training, the combined dataset was pre-processed to ensure consistency and quality. All domain names were converted to lowercase, and any extraneous components (e.g., URLs, subdomains, or paths) were removed. Duplicate entries were also eliminated to avoid bias in the training and testing phases.

In addition to the creation of labelled samples, the full Alexa Top 1 Million list was utilized to construct reference n-gram distributions. These reference distributions are later used to compute relative entropy features, which measure how much a domain name's character patterns deviate from those found in legitimate domains. The actual calculation and role of relative entropy are described in Section 2.2.

2.2 Feature Extraction

The system extract features solely from the domain name string, without relying on DNS responses, WHOIS data, or network traffic. These features are divided into two broad categories: lexical/statistical and entropy-based features. These features aim to capture both human-like patterns typical of legitimate domain names and the algorithmic structure often found in DGA-generated ones.

2.2.1 Lexical and Statistical Features

Lexical and statistical features are derived directly from the domain name strings and aim to capture patterns that differentiate legitimate domains from those generated algorithmically. These features are simple yet effective, as many DGAs produce domains with unnatural structures and statistical properties.

One of the most basic features is the length of the domain. As shown in Figure 1(a), the average length of DGA domains is noticeably higher (mean ≈ 14.81) compared to Non-DGA domains (mean ≈ 8.46). DGA domains also display a wider range and higher variability in length due to the way many DGA algorithms generate long, random-looking strings to increase uniqueness and avoid collisions. In contrast, legitimate domains tend to be shorter and more consistent in length, often optimized for user readability and brand recognition.

Other lexical features include the number of digits, vowels, and consonants, which are also normalized by computing their ratios over the total length. Typically, legitimate domains contain more vowels and linguistically valid character sequences, whereas DGA domains show a different pattern depending on the generation logic (e.g., dictionary-based or random).

The number of unique characters, the number of repeated characters, and the longest consecutive sequences of vowels and consonants were also computed to reflect the diversity and structural characteristics of each domain. Legitimate domains often reuse characters and follow pronounceable sequences, while DGA domains may contain irregular repetition or unusual sequences.

Character-level variations between DGA and Non-DGA domains are visualized in Figure 1(b), which depicts their respective character probability distributions. Legitimate domains show a non-uniform distribution dominated by vowels and commonly used consonants, whereas DGA domains often have a flatter distribution due to random character selection. These lexical features lay the foundation for understanding structural differences before moving to more advanced sequence-based and entropy-based analysis.

Finally, two binary features were included: whether a domain starts with a digit and whether it ends with a digit. Such characteristics are rare in user-friendly domain names but more common in automatically generated ones.

2.2.2 N-Gram and Entropy-Based Features

While character-level features indicate the frequency of individual symbols within a domain name, they do not account for the order or sequential arrangement of characters. To capture such structural information, n-gram-based features are employed, which consider consecutive character combinations, in conjunction with entropy measures to quantify randomness and deviations from typical patterns.

For each domain name, overlapping bigrams, trigrams, and fourgrams—representing sequences of 2, 3, and 4 characters, respectively—are extracted. Relative entropy (Kullback–Leibler divergence) is then calculated by comparing the n-gram frequency distributions of each domain against a reference distribution derived from the full Alexa Top 1 Million legitimate domains. This analysis allows determination of whether a domain conforms to natural character patterns or exhibits structural deviations indicative of algorithmic generation.

To illustrate this, Figure 2 shows the top 20 most frequent bigrams found in both DGA and Non-DGA domain names. As seen, Non-DGA domains often contain common, readable sequences such as "in", "er", and "an", which are frequently used in natural language. In contrast, DGA domains feature more irregular and less meaningful combinations, highlighting the lack of linguistic structure in algorithmically generated names.

In addition to n-gram analysis, Shannon entropy was computed at the unigram (character) level to quantify the overall randomness of characters within each domain name. Higher entropy values typically suggest less predictable, more disordered character sequences — a common trait in many DGA-generated domains. In contrast, legitimate domain names, influenced by linguistic patterns and branding, tend to exhibit lower entropy and more structured distributions.

To visualize this, Figure 3 shows the normal distribution fit of Shannon entropy for both DGA and Non-DGA domains. The curve for DGA domains is shifted toward higher entropy values and displays greater variance, while Non-DGA domains are more tightly clustered around lower entropy. This clear separation demonstrates that Shannon entropy is a strong and interpretable feature for distinguishing between algorithmically and human-generated domain names.

Entropy-based features, when combined with lexical statistics, provide a rich representation of both the randomness and structural characteristics of domain names. This enables machine learning models to distinguish DGA domains more effectively.

3. METHODOLOGY

The methodology adopted for this study builds upon the dataset construction and feature extraction process described in Section 2. The overall approach involves partitioning the dataset into training and testing subsets, training multiple machine learning classifiers on the extracted features, and evaluating their performance using standard classification metrics. In addition, model interpretability and comparative performance are examined to validate the robustness of the approach.

3.1 Dataset Partitioning

To enable supervised learning, the dataset was first divided into training and testing sets. A split ratio of 80% for training and 20% for testing was employed. Stratified sampling was applied to maintain proportional representation of both classes (DGA and Non-DGA) within each subset. This ensured that the models were trained and evaluated on balanced distributions, thereby reducing the risk of bias toward either legitimate or malicious domains.

3.2 Model Selection and Training

Five machine learning algorithms were selected to represent a diverse range of classification strategies: Logistic Regression, Support Vector Machine (SVM), Random Forest, K-Nearest Neighbors (KNN), and Extreme Gradient Boosting (XGBoost**).** These models were chosen because they cover both linear and non-linear approaches, ensemble-based methods, and instance-based learning, thereby providing a comprehensive assessment of the discriminative power of the selected features.

All models were trained on the same standardized feature set to ensure fairness in evaluation. Features were normalized to a uniform scale prior to training, which is particularly important for distance-based algorithms such as KNN and margin-based algorithms such as SVM. Hyperparameter tuning was performed using grid search with cross-validation where computationally feasible, while default settings were retained in cases where parameter optimization did not yield substantial performance gains. By considering a diverse set of classifiers, the study ensured that the results were not dependent on a single model but instead reflected the strength of the feature set itself.

3.3 Performance Evaluation and Feature Analysis

The trained classifiers were evaluated using widely accepted performance measures, namely accuracy, precision, recall, and F1-score.

- **Accuracy** provides an overall measure of correct classifications across both classes.
- **Precision** emphasizes the proportion of domains classified as DGA that were indeed malicious, thus quantifying the system's ability to avoid false alarms.
- **Recall** captures the proportion of malicious domains correctly identified, reflecting the sensitivity of the system to DGA detection.
- **F1-score** combines precision and recall into a single harmonic mean, offering a balanced metric when trade-offs exist between false positives and false negatives.

This set of metrics was selected to provide a holistic understanding of classifier behavior, particularly in security-related tasks where both detection sensitivity and minimization of false alarms are critical. In addition to these quantitative measures, feature importance values were extracted from tree-based models (Random Forest and XGBoost), while coefficient weights were examined in Logistic Regression. This interpretability analysis allowed identification of the most influential lexical and entropy-based attributes, highlighting whether the observed detection performance was driven by specific structural or randomness-based characteristics of the domains.

4. RESULTS

The performance of the five classifiers—Random Forest, Logistic Regression, Support Vector Machine (SVM), XGBoost, and K-Nearest Neighbors (KNN)—was systematically evaluated on the same standardized feature set. Model outputs were assessed using accuracy, precision, recall, and F1-score, with the comparative results summarized in Figure 4.

4.1 Classification Performance and Precision–Recall Analysis

All classifiers achieved consistently strong performance, with accuracies exceeding 90%. This high baseline indicates that the discriminative power is primarily attributable to the chosen feature set rather than the specific learning algorithm. Among the models, XGBoost achieved the highest overall performance**,** with a precision of 0.95, recall of 0.98, and F1-score of 0.96. The elevated recall value is particularly noteworthy, as it demonstrates the ability of XGBoost to minimize false negatives—an essential factor in cybersecurity contexts where undetected threats can have severe consequences.

Random Forest followed closely, showing balanced precision and recall values with only marginally lower scores than XGBoost. This reinforces the reliability of ensemble-based classifiers for the detection of DGA domains. Logistic Regression and SVM achieved slightly lower but still competitive results, illustrating that even linear models can effectively separate DGA and Non-DGA domains when guided by well-constructed features. KNN recorded the lowest performance, yet it still surpassed baseline expectations, reflecting its capacity to capture local similarities in the feature space.

A closer inspection of the precision–recall trade-offs further illustrate classifier behavior. XGBoost achieved the best balance by simultaneously maximizing recall and maintaining high precision, ensuring both comprehensive detection and minimal false alarms. Random Forest exhibited a similar trend but with a slightly lower recall, implying a marginally greater risk of missed detections. Conversely, Logistic Regression and SVM tended toward higher precision relative to recall, adopting a more conservative classification boundary that reduces false positives at the expense of underdetecting some malicious domains. KNN showed moderate trade-offs but lacked the stability demonstrated by ensemble methods. These results highlight that the selection of a classifier may depend on application-specific requirements, such as prioritizing high recall for security-critical systems or high precision for reducing unnecessary alerts.

4.2 Feature Importance Analysis

To better understand the basis for classification, feature importance scores were extracted from Random Forest and XGBoost, while coefficient weights were examined for Logistic Regression. The results, shown in Figure 5, consistently emphasized the dominance of entropy-based features. In particular, trigram entropy and fourgram entropy emerged as the most influential predictors across models. These features effectively capture deviations from natural character sequences, which are characteristic of domains generated algorithmically.

Other lexical attributes, such as domain length, number of consonants, and the longest consonant sequence, also contributed meaningfully to classification, though their relative importance was consistently lower. By contrast, features such as whether a domain begins or ends with a digit had minimal influence, reinforcing the idea that simple heuristics are insufficient for robust detection. This analysis underscores the critical role of entropy measures in identifying algorithmically generated randomness in domain structures.

4.3 Robustness and Key Findings

An important aspect of the evaluation lies in the relative consistency of classifier performance. Although XGBoost and Random Forest achieved the highest scores, the margin of improvement over Logistic Regression, SVM, and KNN was not substantial. This pattern suggests that the predictive strength resides primarily in the chosen feature set, rather than being heavily dependent on the modeling technique. In other words, the entropy- and lexical-based features provide a stable representation of domain characteristics that can be effectively leveraged by both linear and non-linear classifiers.

Feature importance analysis further supports this observation, as entropy measures—particularly trigram and fourgram entropy—consistently ranked highest across models, while lexical features such as domain length and consonant counts contributed supplementary but secondary value. The alignment of importance rankings across different classifiers highlights the robustness and generalizability of entropy as a discriminative signal.

Collectively, these findings indicate that the proposed feature set offers a strong and transferable foundation for DGA detection. This robustness reduces reliance on any single algorithm, making the approach adaptable to different operational environments and computational constraints.

5. CONCLUSION

This study investigated the detection of domain names generated by Domain Generation Algorithms (DGAs) through a combination of lexical and entropy-based features. The primary objective was to assess the effectiveness of entropy—a concept from information theory—in identifying structural irregularities that are characteristic of algorithmically generated domains.

A balanced dataset was curated, consisting of 32,000 legitimate domain names from the Alexa repository and 32,000 DGA-generated domain names from the J. Selvi dataset. Each domain was analyzed using a carefully selected set of features, including character-level statistics, structural indicators, and multiple entropy-based measures. Specifically, Shannon entropy was computed at the unigram level, while relative entropy (Kullback–Leibler divergence) was calculated across bigram, trigram, and fourgram sequences to capture deviations from natural character patterns. Five machine learning models were trained and evaluated, with XGBoost and Random Forest achieving the highest performance across the evaluated metrics. Feature importance analysis further revealed that entropy-based features—particularly trigram and fourgram entropy—were among the most informative. These findings indicate that entropy can serve as a strong and interpretable signal for detecting algorithmic randomness in domain names.

While the results are promising, the study has certain limitations. The dataset primarily consists of randomly generated DGA domains, which generally exhibit high entropy and lack natural linguistic structure. Consequently, the models demonstrate strong effectiveness in detecting such domains. However, dictionary-based DGA domains, which are constructed using real words or word-like sequences, often display lower entropy and resemble legitimate domains more closely. The selected feature set may not adequately capture these subtle patterns, which could limit detection performance against such advanced DGAs. Future research could address this challenge by incorporating contextual or semantic features that better reflect the linguistic characteristics of dictionary-based domain names.

Overall, the findings demonstrate that entropy-based features, when combined with lexical attributes, constitute a robust and interpretable framework for detecting DGA-generated domains. This approach provides practical value for strengthening the capabilities of threat detection systems in real-world cybersecurity applications.

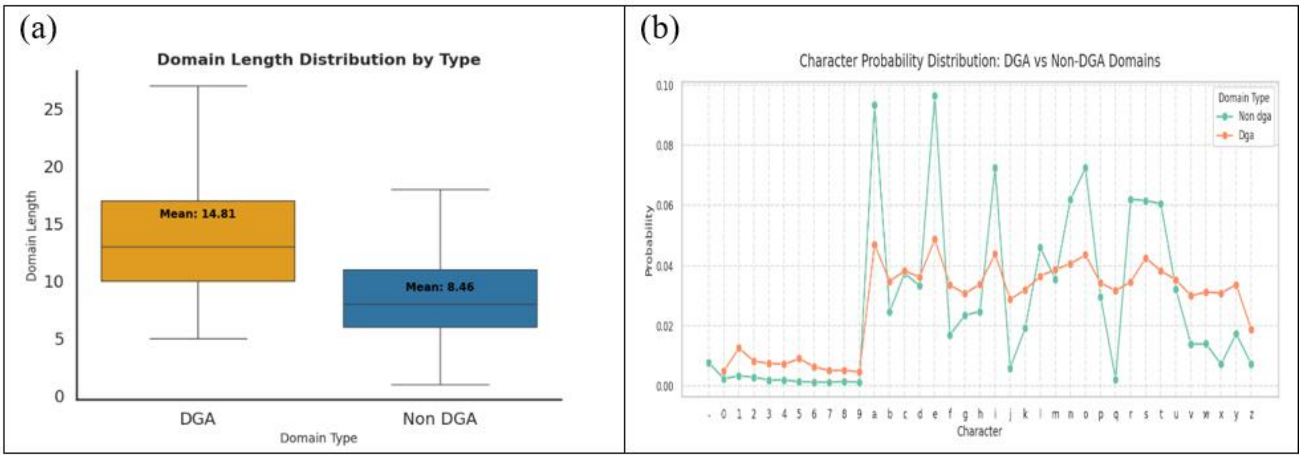

Fig 1: (a) Domain length distribution for DGA and Non-DGA domains. DGA domains tend to be longer and more variable in length, while Non-DGA domains are shorter and more consistent. (b) Character distribution comparison between DGA and Non-DGA domains. DGA domains exhibit irregular character usage, while Non-DGA domains align more with typical linguistic patterns.

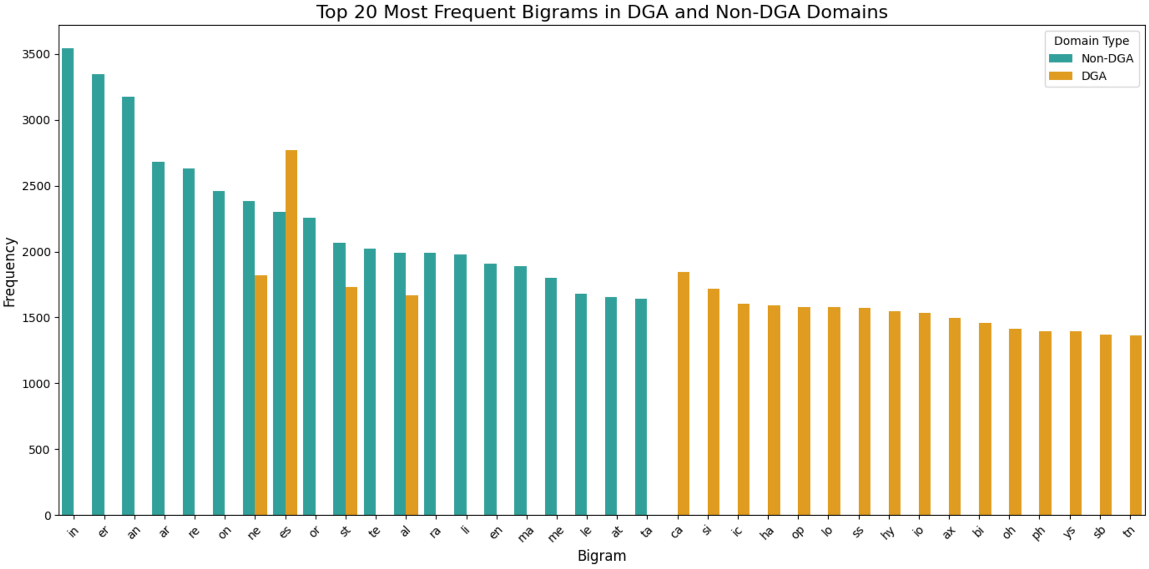

Fig 2: Top 20 most frequent bigrams in DGA and Non-DGA domain names. Non-DGA domains exhibit common, linguistically meaningful bigrams, while DGA domains contain less frequent and more irregular combinations.

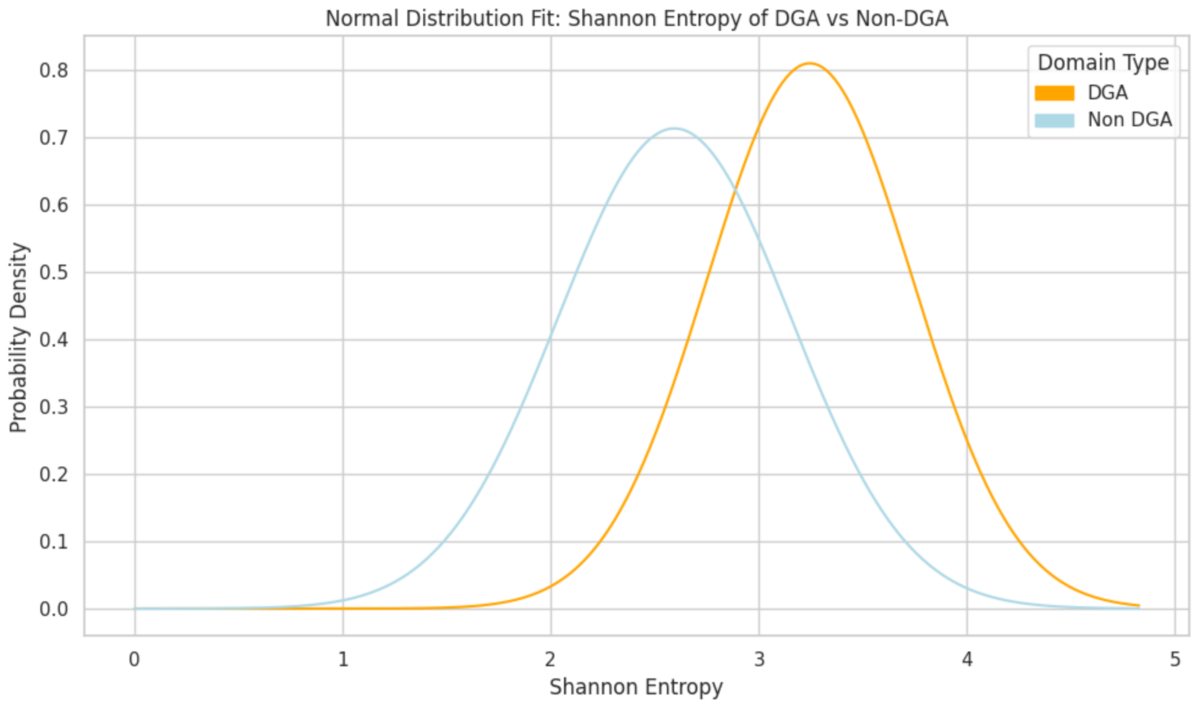

Fig 3 : Normal distribution fit of Shannon entropy for DGA and Non-DGA domains. DGA domains show a higher average entropy than Non-DGA domains, reflecting their more random character composition.

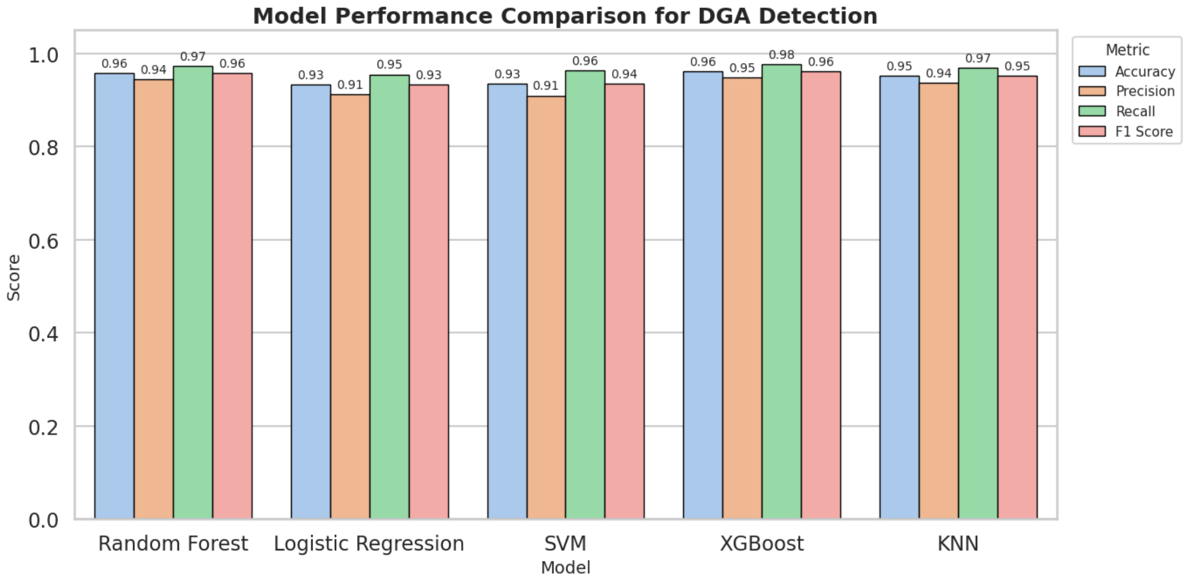

Fig 4: Model Performance Comparison for DGA Detection. XGBoost and Random Forest outperform other classifiers across all metrics.

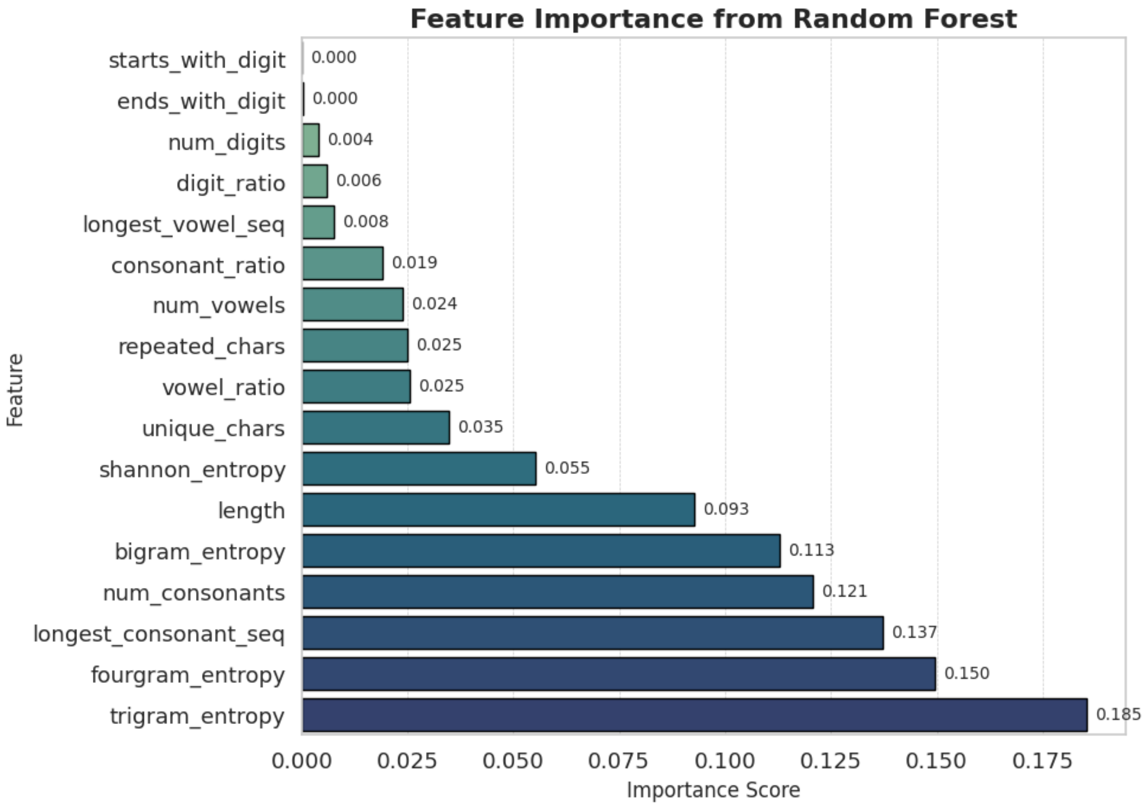

Fig 5: Feature Importance from Random Forest. Entropy-based features (trigram and fourgram entropy) are the most influential in distinguishing DGA from Non-DGA domains.

6. ACKNOWLEDGMENTS

The author Jinsu Ann Mathew gratefully acknowledges Mahatma Gandhi University, Kottayam, Kerala, India for providing fellowship support that contributed to this research.

7. REFERENCES

[1] Zhao H, Chang Z, Bao G, Zeng X. Malicious Domain Names Detection Algorithm Based on *N*-Gram. J Comput Netw Commun. 2019 Feb 3;2019:1–9.

[2] Chen S, Lang B, Chen Y, Xie C. Detection of Algorithmically Generated Malicious Domain Names with Feature Fusion of Meaningful Word Segmentation and N-Gram Sequences. Appl Sci. 2023 Mar 30;13(7):4406.

[3] Zhang Y. A Ensemble Learning method for Domain Generation Algorithm Detection. 3(4).

[4] Ren F, Jiang Z, Wang X, Liu J. A DGA domain names detection modeling method based on integrating an attention mechanism and deep neural network. Cybersecurity. 2020 Dec;3(1):4.

[5] Satoh A, Fukuda Y, Kitagata G, Nakamura Y. A Word-Level Analytical Approach for Identifying Malicious Domain Names Caused by Dictionary-Based DGA Malware. Electronics. 2021 Apr 28;10(9):1039.

[6] Zhang Y, Zhang Y, Xiao J. Detecting the DGA-Based Malicious Domain Names. In: Yuan Y, Wu X, Lu Y, editors. Trustworthy Computing and Services [Internet]. Berlin, Heidelberg: Springer Berlin Heidelberg; 2014 [cited 2025 Aug 8]. p. 130–7. (Communications in Computer and Information Science; vol. 426). Available from: https://link.springer.com/10.1007/978-3-662-43908-1_17

[7] Huynh KH, Visser M. Detecting Botnets Communicating with Command and Control Servers with DNS and NetFlow Data.

[8] Wang T, Chen LC, Genc Y. A dictionary-based method for detecting machine-generated domains. Inf Secur J Glob Perspect. 2021 July 4;30(4):205–18.

[9] Yadav S, Reddy AKK, Reddy ALN, Ranjan S. Detecting Algorithmically Generated Domain-Flux Attacks With DNS Traffic Analysis. IEEEACM Trans Netw. 2012 Oct;20(5):1663–77.

[10] Zhang P, Liu T, Zhang Y, Ya J, Shi J, Wang Y. Domain Watcher: Detecting Malicious Domains Based on Local and Global Textual Features. Procedia Comput Sci. 2017;108:2408–12.

[11] Zhang W wei, Gong J, Liu Q. Detecting Machine Generated Domain Names Based on Morpheme Features: In Shanghai, China; 2013 [cited 2025 Aug 8]. Available from: https://www.atlantis-press.com/article/9952

[12] Liang Z, Zang T, Zeng Y. MalPortrait: Sketch Malicious Domain Portraits Based on Passive DNS Data. In: 2020 IEEE Wireless Communications and Networking Conference (WCNC) [Internet]. Seoul, Korea (South): IEEE; 2020 [cited 2025 Aug 8]. p. 1–8. Available from: https://ieeexplore.ieee.org/document/9120488/

[13] Selvi J, Rodríguez RJ, Soria-Olivas E. Detection of algorithmically generated malicious domain names using masked N-grams. Expert Syst Appl. 2019 June;124:156–

63.

[14] Yadav S, Reddy AKK, Reddy ALN, Ranjan S. Detecting algorithmically generated malicious domain names. In: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement [Internet]. Melbourne Australia: ACM; 2010 [cited 2025 Aug 8]. p. 48–61. Available from: https://dl.acm.org/doi/10.1145/1879141.1879148

[15] Casino F, Lykousas N, Homoliak I, Patsakis C, Hernandez-Castro J. Intercepting Hail Hydra: Real-time detection of Algorithmically Generated Domains. J Netw Comput Appl. 2021 Sept;190:103135.

[16] Cucchiarelli A, Morbidoni C, Spalazzi L, Baldi M. Algorithmically generated malicious domain names detection based on n-grams features. Expert Syst Appl. 2021 May;170:114551.

[17] Palaniappan G, S S, Rajendran B, Sanjay, Goyal S, B S B. Malicious Domain Detection Using Machine Learning On Domain Name Features, Host-Based Features and Web-Based Features. Procedia Comput Sci. 2020;171:654–61.

[18] Anderson HS, Woodbridge J, Filar B. DeepDGA: Adversarially-Tuned Domain Generation and Detection. In: Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security [Internet]. Vienna Austria: ACM; 2016 [cited 2025 Aug 8]. p. 13–21. Available from: https://dl.acm.org/doi/10.1145/2996758.2996767

[19] G. P. A, R. G, S. K, Gladston A. A machine learning framework for domain generating algorithm based malware detection. Secur Priv. 2020 Nov;3(6):e127.

[20] Ma J, Saul LK, Savage S, Voelker GM. Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining [Internet]. Paris France: ACM; 2009 [cited 2025 Aug 8]. p. 1245–54. Available from: https://dl.acm.org/doi/10.1145/1557019.1557153

[21] Sivaguru R, Choudhary C, Yu B, Tymchenko V, Nascimento A, Cock MD. An Evaluation of DGA Classifiers. In: 2018 IEEE International Conference on Big Data (Big Data) [Internet]. Seattle, WA, USA: IEEE; 2018 [cited 2025 Aug 8]. p. 5058–67. Available from: https://ieeexplore.ieee.org/document/8621875/

[22] Sivaguru R, Peck J, Olumofin F, Nascimento A, De Cock M. Inline Detection of DGA Domains Using Side Information. IEEE Access. 2020;8:141910–22.

[23] Tong V, Nguyen G. A method for detecting DGA botnet based on semantic and cluster analysis. In: Proceedings of the Seventh Symposium on Information and Communication Technology [Internet]. Ho Chi Minh City Vietnam: ACM; 2016 [cited 2025 Aug 8]. p. 272–7. Available from: https://dl.acm.org/doi/10.1145/3011077.3011112

[24] Almashhadani AO, Kaiiali M, Carlin D, Sezer S. MaldomDetector: A system for detecting algorithmically generated domain names with machine learning. Comput Secur. 2020 June;93:101787.

[25] Hwang C, Kim H, Lee H, Lee T. Effective DGA-Domain Detection and Classification with TextCNN and Additional Features. Electronics. 2020 June 30;9(7):1070.

[26] Alexa Top 1 Million Sites [Internet]. [cited 2025 Aug 19]. Available from: https://www.kaggle.com/datasets/cheedcheed/top1m