Botnet detection: a system for identifying DGA-based botnets using LightGBM

Mumtazimah Mohamad[1,2], Nazirah Abd Hamid[1], Sanaa A. A. Ghaleb[1,2]
Siti Dhalila Mohd Satar[1], Suhailan Safei[1], Wan Mohd Amir Fazamin Wan Hamzah[1,2], Lim En En[1]
[1]Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Besut, Malaysia
[2]Artificial Intelligence Research Centre for Islam and Sustainability, Universiti Sultan Zainal Abidin, Malaysia

Article Info

Article history:

Received Apr 1, 2024
Revised Nov 3, 2025
Accepted Dec 13, 2025

Keywords:

Anomaly detection
Botnets
Domain generation algorithm
Light gradient boosting model
Machine learning

ABSTRACT

Botnets present a major challenge to detecting anomalies in domain generation algorithms (DGAs). Botmasters use DGAs to create numerous domain names to communicate with command-and-control servers, complicating the detection process. Traditional blacklisting methods struggle to effectively identify anomalous DGA domain names amid the vast number of randomly generated domains, leading to a greater risk of detection being evaded. The proliferation of DGA-based botnets has created an urgent need for robust detection methods. Various techniques and attributes have been utilised to categorise different DGA families, yet the dynamic nature of DGA domain names renders the current blacklisting algorithms ineffective. Additionally, the dynamic characteristics of DGAs further complicate classification, emphasising the need for machine learning models to improve detection accuracy and enhance cyber defence. This study proposes a robust solution to address the challenges posed by DGA-based botnets by developing an innovative machine learning-based model for domain name classification. The model leverages the light gradient boosting algorithm (LightGBM) and integrates n-gram features to enhance the detection of malicious DGA domains. This approach offers superior accuracy, adaptability, and efficiency in identifying and classifying anomalous domain names, achieving 96% precision when detecting true DGA domains. This system represents a significant advancement in cybersecurity and anomaly detection.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Mumtazimah Mohamad
Faculty of Informatics and Computing
Universiti Sultan Zainal Abidin (UniSZA)
Besut, Malaysia
Email: mumtaz@unisza.edu.my

1. INTRODUCTION

Cybercriminals aim to exploit vulnerabilities across various environments, targeting devices, data, software, individuals, and locations to gain unauthorised access or cause harm [1], [2]. Botnets pose a significant threat and consist of three key elements: the botmaster, compromised devices, and the command-and-control (C&C) server [3]. Communication within a botnet occurs in two stages: initially, the botmaster transmits commands to the network of infected machines, either through remote access or directly to the bots [4]. Through this communication, the compromised bots are capable of executing harmful activities upon receiving malicious instructions. The growing threat of botnets is becoming more evident as they endanger the core tenets of network security: confidentiality, integrity, and availability [5]. A particularly concerning aspect is the ability of botnets to initiate distributed denial-of-service (DDoS) attacks, which can severely disrupt network availability and performance [6], [7]. Botnet detection is generally approached from two perspectives: host-based and network-based [8]. The former method identifies the unusual consumption of system resources, such as spikes in CPU usage or memory demand, which can indicate malicious activity [9]. Meanwhile, the network-based approach focuses on analyzing network traffic patterns to detect suspicious behaviour. A key benefit of the latter method is its ability to work even when the traffic is encrypted, although it typically requires constant resource monitoring and can be more time-intensive [10]. Network-based detection methods can be further categorized into signature-based and anomaly-based techniques. Signature-based detection relies on deep packet inspection (DPI) of internet protocol (IP) packets, offering the advantages of low false-positive rates and particular effectiveness at identifying known botnets. A key challenge in identifying new attack patterns is the need for constant updates to signature-based detection systems. Moreover, encryption methods can obscure these signatures, making detection more difficult. An alternative is offered by anomaly-based detection, which analyses factors like packet payload size and unusual bot activity [11]. However, botnet attacks are becoming harder to detect as botnet behaviours evolve [12]. Machine learning techniques have gained popularity for their ability to identify anomalous traffic patterns, although they are often affected by high false-positive rates. Traditional machine learning methods also face challenges such as the requirement for extensive manual feature engineering, which is both labour-intensive and time-consuming. To overcome these drawbacks, machine learning approaches have emerged that offer more efficient feature selection and better adaptability to complex network security challenges [13]-[15]. These methods excel at handling diverse datasets and are increasingly favoured in the network security field.

This study investigates domain name classification using the light gradient boosting machine (LightGBM) algorithm and incorporating *n*-gram features. While previous studies have explored the application of machine learning techniques for domain classification, they have not explicitly addressed the effectiveness with which these techniques can distinguish between normal domains and anomalous DGA-based botnet domains. In this paper, two datasets were utilised: the Alexa Top 1 Million Popular Domains dataset created by Alexa Internet, a subsidiary of Amazon, which ranks websites based on global traffic and represents normal domain names; and the 360 NetLab DGA dataset developed by 360 NetLab, a network security research laboratory under Qihoo 360, which contains anomalous DGA domain names. These datasets were combined into one and pre-processed to extract domain name features. Performance was evaluated using accuracy, precision, recall, and the F1-score as key metrics, the aim being to develop a robust machine learning-based model to address the challenges posed by DGA-based botnets.

2. RELATED WORKS

In addressing the issue of botnet detection using machine learning (ML) techniques, researchers have explored various methods of enhancing security in cloud infrastructure and applications. For example, [16] introduced Elasticsearch as a way to improve cloud security by integrating ML techniques into intrusion detection systems (IDS). This fusion demonstrated improved threat-detection capabilities, particularly in cloud environments. However, challenges remain in scaling ML models for real-time, large-scale data processing. The core issue is the need to improve IDS accuracy and efficiency using ML techniques. Despite the significant advancements, handling large-scale datasets and achieving high precision in detecting sophisticated cyber threats remain problematic.

Our study demonstrates that adaptive, real-time machine learning models are more resilient than static or predefined-feature-based models in detecting evolving botnet activities. Future studies could explore dynamic online learning algorithms to suggest feasible ways of producing real-time threat-detection systems that continuously adapt to emerging botnet strategies.

Aziz and Abdulazeez [17], a comparative study was undertaken to evaluate the performance of three prominent algorithms - the support vector machine (SVM), J48, and naive bayes (NB) - using the KDD Cup dataset, with assessments conducted through the WEKA software. J48 achieved the most accurate intrusion detection, but unresolved issues persist, such as handling high-dimensional data, reducing false-positive rates, and improving real-time detection speed. Further research has focused on enhancing network intrusion detection systems (NIDS).

For instance, Mahfouz *et al.* [18], a novel NIDS model leveraged the one-class support vector machine (OCSVM) algorithm, which was trained using normal traffic data to detect deviations that would indicate abnormal behaviour. The model showed promise in detecting network anomalies but struggled in dynamic or complex network environments. Similarly, Hassan *et al.* [19] conducted an extensive evaluation of various ML techniques for IDS, proposing a new approach that utilised unlabelled domain name system (DNS) traffic logs and Packetbeat to feed data into an Elasticsearch database. While this method demonstrated potential, integrating diverse data types like DNS traffic logs remains a challenge. Botnet detection methodologies have also been explored extensively.

In Hoang and Nguyen [20], DNS query data was combined with ML techniques to extract domain names and train classifiers, revealing the importance of domain name patterns in botnet identification. However, botnet evasion tactics continue to challenge detection frameworks. Additionally, Can *et al.* [21] introduced a benign domain classification model and a detection method based on the novel classification model (NCM) algorithm, achieving an accuracy exceeding 81%. Nevertheless, managing noise and handling exceptional cases remain problematic. A more advanced ML framework was introduced in [22], with a truth-labelled dataset utilised to feed a botnet detection system. This approach achieved 95.3% accuracy and a false-positive rate (FPR) of 5.4%. Challenges persist despite these promising results, like reducing FPR and adapting to evolving botnet strategies.

Li *et al.* [23], another advanced ML framework was proposed, featuring a flexible blacklist, feature extraction modules, and a dual-level ML model for both classification and clustering. This deep learning model outperformed traditional ML algorithms in scalability and accuracy but still faced challenges in handling large-scale real-time data streams. Several other contributions include domain name classification methods using machine learning. For example, Segurola-Gil *et al.* [24] proposed a method structured into three phases: domain name encoding, the application of long short-term memory (LSTM) and convolutional Siamese embedders, and testing five ML algorithms. This method achieved an F1-score and accuracy of approximately 91%. In Yang *et al.* [25], a framework was introduced that featured an improved parallel CNN (IPCNN) architecture with multi-size convolution kernels and a self-attention-based bidirectional LSTM (SA-Bi-LSTM), which enhanced global feature extraction. However, traditional deep learning models still struggle to detect sophisticated domain generation algorithms (DGAs).

Cebere *et al.* [26], the authors reviewed network-level DGA detection, surveying 38 papers and highlighting critical assumptions for real-world applications. Many approaches were found to be based on fragile assumptions, limiting their practicality. Finally, Matin *et al.* [27] introduced an innovative malware detection architecture that combined honeypot techniques with advanced ML algorithms. This dynamically trapped malicious traffic using honeypots and classified malware in real-time using decision tree (DT) and SVM algorithms. While this approach improved the detection of evolving threats, it still encountered limitations in handling real-time threat detection using static datasets.

While previous studies such as [20]-[27] have demonstrated the potential of machine learning in detecting and preventing botnet activities, they often rely on static datasets or predefined features like domain-flux techniques and DNS traffic mining. These methods may not adapt effectively to real-time or evolving threats, particularly with botnets that leverage DGAs, which continuously modify their behaviour and domain names. To address these limitations, the approach presented here incorporates dynamic and adaptive models capable of evolving alongside botnets. Instead of relying solely on static features, the focus is on developing a real-time detection system using continuous learning mechanisms and real-time traffic analysis. This approach will enable a model to adjust to emerging botnet patterns, enhancing detection accuracy even as botnets evolve. More specifically, experiments were conducted with advanced machine learning techniques, such as online learning algorithms, to enable adaptation to shifting botnet communication behaviours. By integrating these dynamic models with real-time data analysis and continuously updating the detection framework, the aim of this new approach was to significantly improve upon previous methods. This would make systems more resilient to changing botnet strategies and mitigate the limitations of static detection techniques.

The paper is organised into the following structure: Section 2 provides an in-depth review of the existing literature, exploring various studies and methodologies previously employed in the field. Section 3 offers a comprehensive overview of the materials and methods used in this investigation, detailing the datasets, machine learning algorithms, and feature extraction techniques implemented. Section 4 covers performance evaluation, presenting the results of the implemented models and thoroughly examining them using relevant metrics such as accuracy, precision, recall, and the F1-score. Section 5 contains a detailed analysis and discussion of the experimental results, offering insights into the findings, their implications, and comparisons with previous research. Section 6 provides a summary and conclusion, encapsulating the key contributions of the research, highlighting its significance, and suggesting potential avenues for future work. The focus on developing dynamic and adaptive models capable of evolving alongside botnets indicates a promising direction for enhancing the accuracy and resilience of detection systems. By integrating real-time traffic analysis and continuous learning mechanisms, future researchers could address the current limitations by, for example, adapting to domain generation algorithms (DGAs) and mitigating high false-positive rates, thereby making detection frameworks more robust in combatting advanced threats.

3. METHODS

In this study, a step-by-step methodological approach was employed to address the challenges associated with DGAs. The methods were designed to ensure replicability and accuracy by providing the details necessary to verify and reproduce the findings. The methodology was divided into three distinct phases, as shown in Figure 1. The first began with identifying the challenges posed by the rapid generation of domain names by DGAs, which varies significantly across different algorithms. A critical issue is the inefficiency with which blacklist approaches detect anomalous DGA domain names, especially when linked to dynamic command-and-control (C2) servers that change frequently. These dynamic characteristics allow malicious domains to evade detection through traditional methods. Additionally, the laborious process of collecting a dedicated dataset and analyzing large amounts of data presents further obstacles. The phase involved the collection and preparation of datasets. The normal domain names were taken from the Alexa Top-1m.csv dataset, while anomalous domain names were sourced from the 360 NetLab dga.txt dataset. These datasets were combined, and relevant features such as the domain length, entropy, and *n*-grams were extracted to assist in classification. The machine learning model, utilising the LightGBM algorithm, was then trained on these features. The choice of LightGBM is justified because it is a gradient-boosting method that efficiently handles large-scale data, accelerates the training process, and uses a leaf-wise splitting approach to optimize performance. The model was tested on a separate dataset, and its performance was evaluated using metrics like precision, recall, and accuracy. The simulation phase involved dividing the data into two subsets, one for training and one for testing, in ratios such as 80/20 or 70/30. This split ensured that the model would be exposed to sufficient data during training and could generalize well to unseen data during testing. Techniques like cross-validation were employed to prevent overfitting, and the model's final performance was measured based on its ability to accurately classify normal and anomalous domain names. The choice of LightGBM was found to correlate with higher classification performance, especially in terms of precision and recall. The method proposed in this study tended to have an inordinately higher proportion of correctly classified anomalous domain names compared to traditional methods like blacklist approaches. This highlights the effectiveness with which LightGBM can address the challenges posed by dynamic DGAs and demonstrates its potential for enhancing domain classification accuracy.

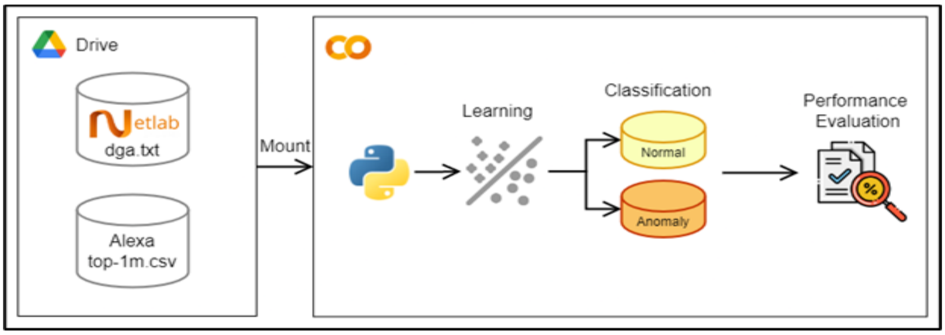

Figure 1. Methodology for the study

3.1. Data pre-processing

The datasets represent publicly established normal and DGA domain names. Alexa’s Top 1 million popular domains dataset was selected as the dataset of normal domain names, and the 360 NetLab [28] DGA dataset from network security research lab at 360 was selected as the dataset of anomalous DGA domain names [29]. These datasets were prepared in a cloud space with a .csv extension. The Alexa’s Top 1 million sites dataset was retrieved as top-1m.csv, and the 360 NetLab DGA dataset was retrieved as dga.txt. Both datasets were then concatenated into one. The combined set of domain names was further pre-processed to extract the domain name features.

3.2. Classification

In this phase, one ML algorithm used for classification was the LightGBM algorithm. LightGBM is a popular gradient-boosting technique that builds a predictive model using a stage-wise method. LightGBM utilises the labelled patterns and builds the classifier to distinguish between normal and anomalous domain names. The LightGBM algorithm optimises parallel learning by combining sophisticated network connectivity, a histogram-based algorithm [30], and a fast training process. This is known as the parallel voting DT method. Additionally, LightGBM grows trees using the leaf-wise technique, identifying which leaf to split based on its highest gain of variance. The base algorithm of LightGBM is presented in the form of pseudocode (Algorith 1), as depicted in [31]. The LightGBM algorithm comprises various parameters known as hyperparameters, which play a crucial role in influencing the performance of the LightGBM algorithm. Typically, these hyperparameters are manually set at the beginning and fine-tuned through an ongoing process of trial and error.

Algorithm 1. LightGBM algorithm

```
Start
1. Initialisation log loss and leaf value:
2. Combine features that are mutually exclusive
3. For each iteration M data sampling
Calculate gradient absolute values
Resampling data using gradient-based one-side sampling
Calculate information gain
Update leaf values and regulation
Repeat until convergence
Stop
```

Figure 2 shows the strategy framework used in the experimental setting. The combined set of domain names was further pre-processed to extract the features of the names. The two phases in ML classification are training and testing. The database was split to enable these two phases.

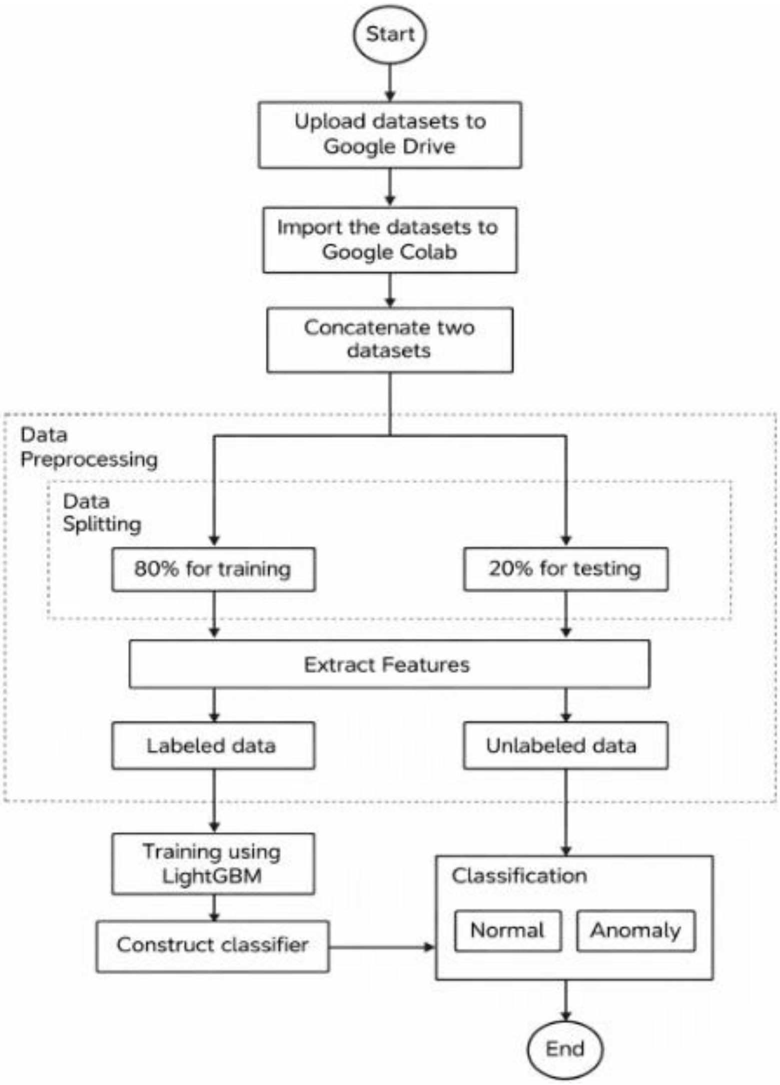

Figure 2. The flow of data pre-processing with data training and testing

4. PERFORMANCE EVALUATION

In the performance evaluation stage, precision, recall, the F1-score, and overall classification accuracy were recorded based on the classification results. Four possible outcomes would be obtained from the classification. Both normal and DGA domain names can either be classified correctly or misclassified. Table 1 shows a sample confusion matrix showing the number of predicted labels that matched the actual labels.

Table 1. Confusion matrix

Predicted\Actual	Normal	Anomaly	Total
Normal	TN	FP	TN/FP
Anomaly	FN	TP	FN/TP
Total	TN/FN	FP/TP	

TP = Number of correctly predicted anomaly class; TN = Number of correctly predicted normal class; FP = Number of normal class predicted as anomaly; FN = Number of anomaly class predicted as normal

In this study, considering anomaly DGA domain names as positive results meant that the true positives (TP) would be the proportion of correctly classified DGA domain names, while the false positives (FP) would be the proportion of normal domain names that had been misclassified as DGA domain names. Alternatively, the true negatives (TN) would be the proportion of correctly classified normal domain names, and the false negatives (FN) would be the proportion of DGA domain names misclassified as normal domain names. These results are based on the performance evaluation using the metrics shown in Table 2.

Table 2. Performance evaluated by the metrics

Measure	Definition	
Accuracy	$ACC = \frac{TP + TN}{TP + TN + FP + FN}$	(1)
Precision	$\frac{TP}{TP + FP}$	(2)
Recall	$\frac{TP}{TP + FN}$	(3)
F1-Score	$2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$	(4)

5. RESULTS AND DISCUSSION

This section presents the evaluation results to verify the performance of the classifier. The analysis focuses on two key sets of features: entropy and length, followed by the added *n*-grams features. Through this structured approach, the aim was to demonstrate the effectiveness of this new classification model.

5.1. Entropy and length features

The results of this study provide crucial insights into the ability of the LightGBM-based model to effectively classify DGA and legitimate domains. As demonstrated in Table 3, the model achieved an overall accuracy of 81%, indicating its reliability in distinguishing between DGA-generated and normal domain names. The model successfully classified 42.24% of the DGA domains and 38.54% of the legitimate domains, demonstrating its potential for improving botnet detection. However, some misclassification occurred, with 7.76% of the DGA domains wrongly flagged as legitimate and 11.46% of the legitimate domains marked as DGA, reflecting the complexity of DGA patterns. The precision for legitimate domains (0.83) was slightly higher than for DGA domains (0.79), indicating fewer false positives. Meanwhile, the recall for DGA domains was 0.84, illustrating the model's effectiveness in detecting the most malicious domains. The F1-scores of 0.81 for the DGA domains and 0.80 for the legitimate domains highlighted a balanced performance between precision and recall.

According to the confusion matrix shown in Figure 3, the model correctly classified 168,964 DGA domains (42.24%) and 154,158 legitimate domains (38.54%) in the testing set. However, 31,036 DGA domains (7.76%) were misclassified as legitimate, and 45,842 legitimate domains (11.46%) were incorrectly predicted as DGA domains. These results highlight the model's capacity to correctly identify most DGA domains, which is critical in improving the detection of malicious botnet activities.

Table 3. Domain classification results using the LightGBM-based detection model

Metric	DGA domains	Legitimate domains	Overall
Total domains in test set	200,000	200,000	400,000
Correctly classified	168,964 (42.24%)	154,158 (38.54%)	-
Misclassified	31,036 (7.76%)	45,842 (11.46%)	-
Precision	0.79	0.83	-
Recall (detection rate)	0.84	0.77	-
F1-score	0.81	0.80	-
Accuracy	-	-	0.81

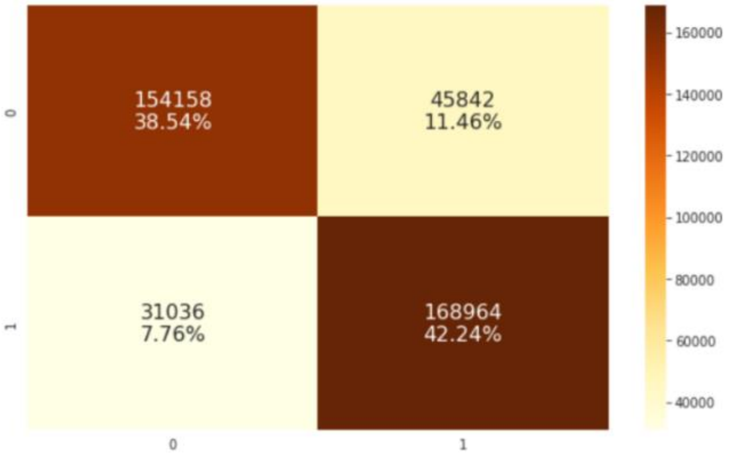

Figure 3. Confusion matrix (entropy and length features)

5.2. Added *n*-grams features

Table 4 provides a comparative analysis of the performance metrics used for the LightGBM model when trained with and without *n*-gram features. The table highlights key metrics such as TP, TN, FP, FN, precision, recall, F1-score, and accuracy. The model with *n*-gram features achieved 191,632 true positives, compared to the 168,964 achieved without *n*-gram features. This increase indicates that adding *n*-gram features significantly enhanced the model's ability to correctly identify DGA domains. Similarly, the true negatives rose from 154,158 to 192,171 with *n*-gram features. This improvement suggests the increased effectiveness of the model in correctly classifying legitimate domains as benign. The model with *n*-gram features had 7,829 false positives, a marked reduction from the 45,842 obtained using the previous model. This reduction in false positives indicates that the inclusion of *n*-gram features improved the model's precision. The false negatives decreased from 31,036 to 8,368, showing that the model with *n*-gram features was more effective at catching DGA domains that had been previously misclassified as legitimate. The precision of both models was recorded, with the model using *n*-gram features achieving 0.96, compared to 0.83 for the model without them. This high precision indicates fewer false positives, demonstrating the model’s enhanced reliability. The recall for both models was also 0.96 with *n*-gram features, an increase from the 0.84 obtained using the previous model. This high recall signifies the model's effectiveness in identifying the majority of DGA domains. Both models achieved an F1-score of 0.96 with *n*-gram features, while the previous model recorded 0.81. The F1-score reflects a balanced measure of precision and recall, demonstrating that the new model performed exceptionally well in both aspects. Finally, the accuracy of the model with *n*-gram features was 0.96, indicating that 96% of the total classifications were correct. This was a significant improvement over the 0.81 accuracy of the model without *n*-gram features.

Table 4. Performance comparison of LightGBM model with and without *n*-gram features

Metric	With *n*-gram features	Without *n*-gram features
True positive (TP)	191,632	168,964
True negative (TN)	192,171	154,158
False positive (FP)	7,829	45,842
False negative (FN)	8,368	31,036
Precision	0.96	0.83
Recall (Detection Rate)	0.96	0.84
F1-Score	0.96	0.81
Accuracy	0.96	0.81

With reference to the confusion matrix shown in Figure 4, the model demonstrated commendable performance in classifying domains. Specifically, it correctly identified 191,632 DGA domains (47.91%) from the total of 400,000 domains in the testing set, while only 8,368 DGA domains (2.09%) were misclassified as legitimate. Conversely, the model accurately classified 192,171 normal domains (48.04%) as legitimate, with just 7,829 domains (1.96%) incorrectly predicted as DGA domains. These results highlight the model's potential effectiveness in distinguishing between DGA-generated and legitimate domains, which is crucial for enhancing cybersecurity measures. The high rates of correct classification suggest that the methodology employed, particularly the integration of advanced features such as *n*-grams, was effective in capturing the nuances of domain behaviour.

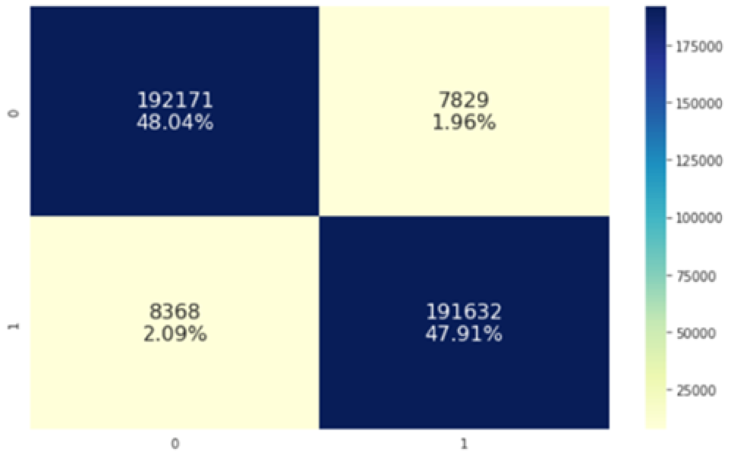

Figure 4. Confusion matrix (entropy, length and n-grams features)

This study explored a comprehensive approach involving the integration of entropy, length, and *n*-gram features for domain classification. However, further and more in-depth studies may be needed to confirm its robustness, especially regarding the ways that additional features impact model performance under different attack scenarios. Additionally, testing the model with a wider range of datasets and considering the evolution of DGA patterns could provide deeper insights into its generalization capabilities. Table 5 provides a comprehensive comparative analysis of the findings of this study and those obtained in relevant state-of-the-art research. By examining various aspects such as the methodologies employed, experimental outcomes, and theoretical contributions, the table highlights the distinctive insights and advancements introduced in the current study.

Table 5. Comparison between results of relevant work and this work

Algorithm	[20]	[21]	[22]	[23]	[24]		[25]	[26]	[27]	Our work
	kNN	NCM	SVM	DNN	MLP	RF	RNN	MCC	SVM+DT	LightGBM
Precision	-	86	75	-	91	91	-	95	-	96
Recall	-	76	84	-	87	86	-	94	-	96
F1-score	90.30	81	79	-	89	88	-	94	-	96
Accuracy	90.20	81	78	95.89	89	89	92	-	0.95	96

k-nearest neighbour → kNN; Novel classification model → NCM; Support vector machines → SVM; Deep neural network →DNN; Random Forest →RF; Recurrent neural network →RNN; Multiclass classification→ MCC; Decision tree→ DT, Light Gradient Boosting Machine → LightGBM. The dash ("-") indicates missing data.

LightGBM clearly performed the best in this analysis, excelling in all the relevant metrics, particularly accuracy, precision, recall, and the F1-score. Its consistent overall performance makes it the most reliable model for domain classification tasks. As presented in [27], [23], SVM + DT and DNN achieved very high accuracy values (95% and 95.89%, respectively). However, the lack of other critical metrics like precision and recall limited a full understanding of their effectiveness. These models could be strong competitors, but they may lack the balance evident in LightGBM. As presented in [21], [22], NCM and SVM showed weaker results, particularly in terms of recall and accuracy, suggesting they struggle to maintain a balance between identifying all positive cases and correctly classifying instances.

As presented in [24], MLP and RF are reliable models that have high precision, recall, and F1-scores, but they are outclassed by LightGBM in every major metric. As presented in [25], [20], RNN and kNN offer reasonable performance but lack detailed metric comparisons. While RNN reached 92% accuracy and kNN achieved 90.2%, they fall short of the top-tier models, especially because their capacity for precision and recall is unknown. As presented in [26], the MCC model is another strongly performing algorithm that exhibited a strong balance across precision, recall, and F1-score. Although no accuracy was provided, its MCC value of 95 suggests a very strongly performing and balanced model.

In summary, LightGBM emerged as the most effective model in terms of overall classification accuracy and balanced performance across the key metrics. Other models, such as SVM + DT, MLP, and RF, showed promise but ultimately fall short in comparison. Moreover, our study suggests that higher computational complexity is not associated with poor performance in accuracy or efficiency. The proposed method may benefit from enhanced feature selection without its ability to generalize across diverse datasets being adversely impacted, further highlighting its potential for robust and balanced performance in domain classification tasks.

6. CONCLUSION

Recent observations suggest that incorporating *n*-gram features into the LightGBM algorithm significantly enhances the detection of malicious DGA domains. Our findings provide conclusive evidence that this improvement is associated with enhanced classification performance and more robust identification of botnet-related traffic flows, rather than merely resulting from larger labelled datasets. This paper presents a model trained on Alexa's Top 1 million Popular Domains dataset and the 360 NetLab DGA dataset, which achieved a 96% accuracy rate in predicting botnet traffic flows. In addition to traffic classification, the impact of inconsistent data was analyzed, and the potential risks posed by harmful botnet attacks were highlighted. The aim of future work is to adapt the model to evolving DGA patterns through continuous learning mechanisms such as online learning, which would enable real-time updates and improved detection of emerging DGA strategies.

ACKNOWLEDGEMENTS

This research was supported by the Center of Research Excellence and Incubation Management (CREIM) and the Artificial Intelligence Research Centre for Islam and Sustainability of Universiti Sultan Zainal Abidin, Terengganu, Malaysia.

FUNDING INFORMATION

The authors state no funding is involved.

CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest.

DATA AVAILABILITY

Data availability does not apply to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] Sharipuddin, R. S. Putra, M. F. Aulia, S. A. Maulana, and P. A. Jusia, “Android security: malware detection with convolutional neural network and feature analysis,” *Media Journal of General Computer Science*, vol. 1, no. 1, pp. 7-13, Dec. 2023, doi: 10.62205/mjgcs.v1i1.7.

[2] H. A. Talib, R. B. Alothman, and M. S. Mohammed, “Malicious attacks modelling: a prevention approach for ad hoc network security,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 30, no. 3, pp. 1856-1865, Jun. 2023, doi: 10.11591/ijeecs.v30.i3.pp1856-1865.

[3] S. Ahmadi, “Challenges and solutions in network security for serverless computing,” *International Journal of Current Science Research and Review*, vol. 07, no. 01, Jan. 2024, doi: 10.47191/ijcsrr/v7-i1-23.

[4] K. Alieyan, A. Almomani, A. Manasrah, and M. M. Kadhum, “A survey of botnet detection based on DNS,” *Neural Computing and Applications*, vol. 28, no. 7, pp. 1541-1558, Jul. 2017, doi: 10.1007/s00521-015-2128-0.

[5] H. Choi and H. Lee, “Identifying botnets by capturing group activities in DNS traffic,” *Computer Networks*, vol. 56, no. 1, pp. 20-33, Jan. 2012, doi: 10.1016/j.comnet.2011.07.018.

[6] X. Li, J. Wang, and X. Zhang, “Botnet detection technology based on DNS,” *Future Internet*, vol. 9, no. 4, p. 55, Sep. 2017, doi: 10.3390/fi9040055.

[7] Y. Zhauniarovich, I. Khalil, T. Yu, and M. Dacier, “A survey on malicious domains detection through DNS data analysis,” *ACM Computing Surveys*, vol. 51, no. 4, pp. 1-36, Jul. 2019, doi: 10.1145/3191329.

[8] X. Sun and Z. Liu, “Domain generation algorithms detection with feature extraction and domain center construction,” *PLoS ONE*, vol. 18, no. 1 January, p. e0279866, Jan. 2023, doi: 10.1371/journal.pone.0279866.

[9] A. M. Manasrah, T. Khdour, and R. Freehat, “DGA-based botnets detection using DNS traffic mining,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 5, pp. 2045-2061, May 2022, doi: 10.1016/j.jksuci.2022.03.001.

[10] R. U. Khan, X. Zhang, R. Kumar, A. Sharif, N. A. Golilarz, and M. Alazab, “An adaptive multi-layer botnet detection technique using machine learning classifiers,” *Applied Sciences*, vol. 9, no. 11, p. 2375, Jun. 2019, doi: 10.3390/app9112375.

[11] T. A. Tuan, H. V. Long, and D. Taniar, “On detecting and classifying DGA botnets and their families,” *Computers and Security*, vol. 113, p. 102549, Feb. 2022, doi: 10.1016/j.cose.2021.102549.

[12] S. Yadav, A. K. K. Reddy, A. L. Narasimha Reddy, and S. Ranjan, “Detecting algorithmically generated domain-flux attacks with DNS traffic analysis,” *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1663-1677, Oct. 2012, doi: 10.1109/TNET.2012.2184552.

[13] J. Namgung, S. Son, and Y. S. Moon, “Efficient deep learning models for DGA domain detection,” *Security and Communication Networks*, vol. 2021, pp. 1-15, Jan. 2021, doi: 10.1155/2021/8887881.

[14] H. Yao, P. Gao, P. Zhang, J. Wang, C. Jiang, and L. Lu, “Hybrid intrusion detection system for edge-based IIoT relying on machine-learning-aided detection,” *IEEE Network*, vol. 33, no. 5, pp. 75-81, Sep. 2019, doi: 10.1109/MNET.001.1800479.

[15] D. Tran, H. Mac, V. Tong, H. A. Tran, and L. G. Nguyen, “A LSTM based framework for handling multiclass imbalance in DGA botnet detection,” *Neurocomputing*, vol. 275, pp. 2401-2413, Jan. 2018, doi: 10.1016/j.neucom.2017.11.018.

[16] O. Negoita and M. Carabas, “Enhanced security using elasticsearch and machine learning,” in *Advances in Intelligent Systems and Computing*, vol. 1230 AISC, 2020, pp. 244-254. doi: 10.1007/978-3-030-52243-8_19.

[17] Z. A. Aziz and A. M. Abdulazeez, “Application of machine learning approaches in intrusion detection system,” *Journal of Soft Computing and Data Mining*, vol. 2, no. 2, pp. 1-13, Oct. 2021, doi: 10.30880/jscdm.2021.02.02.001.

[18] A. M. Mahfouz, A. Abuhussein, D. Venugopal, and S. G. Shiva, “Network intrusion detection model using one-class support vector machine,” in *Advances in Machine Learning and Computational Intelligence: Proceedings of ICMLCI 2019*, 2021, pp. 79-86. doi: 10.1007/978-981-15-5243-4_7.

[19] A. Hassan, S. Tahir, and A. I. Baig, “Unsupervised machine learning for malicious network activities,” in *2019 International Conference on Applied and Engineering Mathematics (ICAEM)*, IEEE, Aug. 2019, pp. 151-156. doi: 10.1109/ICAEM.2019.8853788.

[20] X. D. Hoang and Q. C. Nguyen, “Botnet detection based on machine learning techniques using DNS query data,” *Future Internet*, vol. 10, no. 5, p. 43, May 2018, doi: 10.3390/FI10050043.

[21] N. V. Can, D. N. Tu, T. A. Tuan, H. V. Long, L. H. Son, and N. T. K. Son, “A new method to classify malicious domain name using neutrosophic sets in DGA botnet detection,” *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 4, pp. 4223-4236, Apr. 2020, doi: 10.3233/JIFS-190681.

[22] W. Fang, “Real-time botnet detection system based on machine learning algorithms,” in *2022 2nd Conference on High Performance Computing and Communication Engineering (HPCCE 2022)*, 2023, pp. 226–234.

[23] Y. Li, K. Xiong, T. Chin, and C. Hu, “A machine learning framework for domain generation algorithm-based malware detection,” *IEEE Access*, vol. 7, pp. 32765-32782, 2019, doi: 10.1109/ACCESS.2019.2891588.

[24] L. Segurola-Gil, T. Egues, F. Zola, and R. Orduna-Urrutia, “Siamese neural network and machine learning for DGA classification,” *European Conference on Information Warfare and Security, ECCWS*, vol. 2022-June, no. 1, pp. 271-279, Jun. 2022, doi: 10.34190/eccws.21.1.205.

[25] L. Yang, G. Liu, Y. Dai, J. Wang, and J. Zhai, “Detecting stealthy domain generation algorithms using heterogeneous deep neural network framework,” *IEEE Access*, vol. 8, pp. 82876-82889, 2020, doi: 10.1109/ACCESS.2020.2988877.

[26] B. Cebere, J. Flueren, S. Sebastián, D. Plohmann, and C. Rossow, “Down to earth! Guidelines for DGA-based malware detection,” in *ACM International Conference Proceeding Series*, New York, NY, USA: ACM, Sep. 2024, pp. 147-165. doi: 10.1145/3678890.3678913.

[27] I. M. M. Matin and B. Rahardjo, “Malware detection using honeypot and machine learning,” in *2019 7th International Conference on Cyber and IT Service Management, CITSM 2019*, IEEE, Nov. 2019, pp. 1-4. doi: 10.1109/CITSM47753.2019.8965419.

[28] “Alexa,” Top Site on the Web. Accessed: Nov. 06, 2025. [Online]. Available: https://www.alexa.com/topsites

[29] “360netlab,” DGA. Accessed: Nov. 06, 2025. [Online]. Available: https://data.netlab.360.com/dga/

[30] Q. Zhu, W. Ding, M. Xiang, M. Hu, and N. Zhang, “Loan default prediction based on convolutional neural network and LightGBM,” *International Journal of Data Warehousing and Mining*, vol. 19, no. 1, pp. 1-16, Dec. 2022, doi: 10.4018/IJDWM.315823.

[31] A. A. Taha and S. J. Malebary, “An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine,” *IEEE Access*, vol. 8, pp. 25579-25587, 2020, doi: 10.1109/ACCESS.2020.2971354.

BIOGRAPHIES OF AUTHORS

Mumtazimah Mohamad was born in Terengganu, Malaysia. She received the bachelor's degree in information technology from Universiti Kebangsaan Malaysia, in 2000, the M.Sc. degree in computer science from Universiti Putra Malaysia, and the Ph.D. degree in computer science from Universiti Malaysia Terengganu, in 2014. She was a Junior Lecturer, in 2000. Currently, she is an Associate Professor with the Department of Computer Science, Faculty of Informatics and Computing (FIK), Universiti Sultan Zainal Abidin, Terengganu, Malaysia. She has published over 50 research articles in peer-reviewed journals, book chapters, and proceeding. She has appointed a reviewer and technical committee for many conferences and journals and worked as a researcher in several national funded Research and Development projects. Her research interests include pattern recognition, machine learning, artificial intelligence, and parallel processing. She can be contacted at email: mumtaz@unisza.edu.my.

Nazirah Abd Hamid is a senior lecturer in University Sultan Zainal Abidin, Terengganu, Malaysia. She holds a degree in Bachelor of Information Technology from University Utara Malaysia (UUM), in 2004, M.Sc.Com. (Information Security) from University Teknologi Malaysia (UTM), in 2011 and Ph.D. in Computer Science from Universiti Teknikal Malaysia Melaka (UTeM), in 2023. Her research interests are Information Security, Cyber Security, Pattern Recognition and Data Mining. She can be contacted at email: nazirah@unisza.edu.my.

Sanaa A. A. Ghaleb received the bachelor's degree from the University of Aden, Yemen, in 2011, and the M. Sc degree from Universiti Sains Malaysia, Malaysia, in 2017. She received the Ph.D. degree from the Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Malaysia. Her research interests include technology-enhanced learning, instructional design and technology, computer networks and information security, cybersecurity, machine learning, artificial intelligence, swarm intelligence, and metaheuristic. She can be contacted at email: sanaaabduljabbar@unisza.edu.my.

Siti Dhalila Mohd Satar received her M.Sc. degree from Universiti Teknologi Malaysia (UTM) in 2012. She is a Lecturer at Universiti Sultan Zainal Abidin and her research interest are access control Security System, Security Services (Including Digital Forensic, Steganography, Network Security, and Biometrics) and Data Security. Currently, she is pursuing her PhD study in Universiti Putra Malaysia (UPM), Malaysia. She can be contacted at email: sitidhalila@unisza.edu.my.

Suhailan Safei has 7 years' experience as a programmer and currently working as a senior lecturer since 2007. He received Bachelor's and Master's degrees from Universiti Teknologi Malaysia and his Ph.D. from the Universiti Teknikal Malaysia Melaka. His research interest is in the e-learning data analytic covering clustering and decision support system specially to aid programming language learning. He can be contacted at email: suhailan@unisza.edu.my.

Wan Mohd Amir Fazamin Wan Hamzah received a Bachelor's in Information Technology (Software Engineering), M.Sc. and Ph.D. in Computer Science from the Universiti Malaysia Terengganu in 2003, 2010 and 2016, respectively. He is currently a Senior Lecturer with the Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin (UniSZA), Terengganu, Malaysia. His research interests include learning analytics, machine learning, gamification, and e-learning. He can be contacted at email: amirfazamin@unisza.edu.my.

Lim En En he is affiliated with the Department of Computer Science, Faculty of Informatics and Computing (FIK), Universiti Sultan Zainal Abidin, Terengganu, Malaysia. He can be contacted at email: 056099@putra.unisza.edu.my.