Review Article

Journal of Journalism and Media Management

Detecting Botnets from Dns Query Data Using Machine Learning

Odonchimeg Lkhagva[1*], Bulganmaa Togookhuu[2] and Bilguunt Sogtsaran[3]

[1]Associate Professor, Doctor (Ph.D), Department of Cyber Security, School of Information and Communication Technology (SICT), University of Science & Technology (MUST), Ulaanbaatar, Mongolia

[2] Senior Lecturer, Doctor (Ph.D), Department of Cyber Security, School of Information and Communication Technology (SICT), University of Science & Technology (MUST), Ulaanbaatar, Mongolia

[3] Student, 4th Year, School of Information and Communication Technology, MUST, Ulaanbaatar, Mongolia

[*]Corresponding author

Gor Saribekyan, Moscow State Institute of Electronics and Mathematics, Moscow, Russian Federation.

Received: July 17, 2025; **Accepted:** July 26, 2025; **Published:** July 31, 2025

ABSTRACT

In recent years, botnets have emerged as a significant cybersecurity threat, making detection increasingly complex. Botnets serve as powerful tools for cybercriminals to conduct cyberattacks such as Distributed Denial of Service (DDoS), phishing, and data theft. Therefore, implementing Domain Name System (DNS)-based detection methods to address the botnet problem has emerged as a critical research direction. DNS plays a vital role in mapping domain names to IP addresses and maintaining continuous network traffic management. Botnets exploit this system to hide their operations, leveraging techniques such as dynamic DNS (DDNS), domain generation algorithms (DGA), and fast-flux, which frequently alter domain information to evade detection.

This study delves into DNS functionality and its use in detecting botnets, aiming to enhance protection against attacks. By utilizing machine learning to detect botnets from DNS query data, the research improves the ability to identify cyber threats at an early stage. This provides a novel solution in the cybersecurity field, contributing to proactive defense against cybercriminal activities. The Random Forest machine learning method was employed to detect botnets, achieving an accuracy of over 97%, outperforming traditional methods. However, the detection process faced challenges when some botnets used domain names structurally similar to legitimate ones. To address this, the research introduces a new feature called the Consonant-Vowel Ratio (CVR) as the 25th feature to better identify domains generated by DGAs. The CVR reflects linguistic characteristics commonly observed in DGA-generated domain names, such as a lack of phonetic and structural regularity. The study demonstrates that legitimate domain names maintain consonant-vowel harmony for readability, whereas DGA-generated domains tend to include an excessive number of consonants to evade detection models and blacklists. The findings highlight the effectiveness of CVR in improving detection performance against such botnet strategies.

Keywords: Botnet, Machine Learning, DNS, Random Forest, CVR

Introduction

Cyberattacks targeting individuals and organizations pose significant threats to the internet infrastructure and users, with various types of attacks being observed globally. The rapid evolution of cyberattacks has made botnets one of the most dangerous weapons for cybercriminals. A botnet is a network of compromised devices controlled remotely by an entity known as the botmaster. Under the botmaster's control, these devices coordinate large-scale attacks, including Distributed Denial of Service (DDoS), phishing, spam distribution, and data theft (Figure 1) [1]. These malicious networks typically rely on centralized Command and Control (C&C) infrastructure, enabling the botmaster to orchestrate millions of infected devices to work in unison. Detecting botnets is particularly challenging because of their distributed nature and the adaptive techniques they employ, allowing them to circumvent traditional network security measures.

As a result, Domain Name System (DNS)-based detection has emerged as a critical research direction for uncovering botnet communications and control mechanisms. DNS plays a pivotal role in botnet operations, as it can be used to mask command and

Citation: Odonchimeg Lkhagva, Bulganmaa Togookhuu, Bilguunt Sogtsaran. Detecting Botnets from Dns Query Data Using Machine Learning. J Journalism Media Manag. 2025. 1(1): 1-8. DOI: doi.org/10.61440/JJMM.2025.v1.03

control communications, allowing botnets to dynamically adapt and evade detection [1-3].

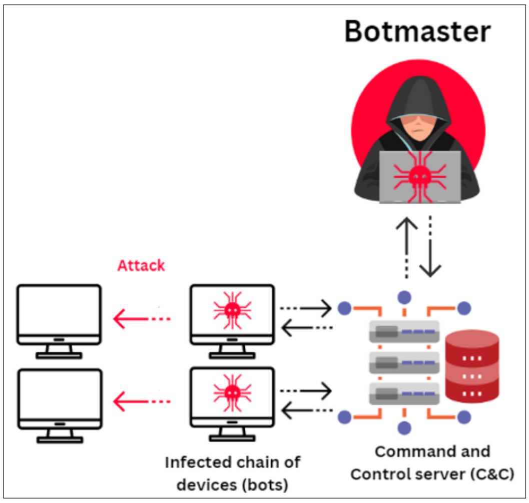
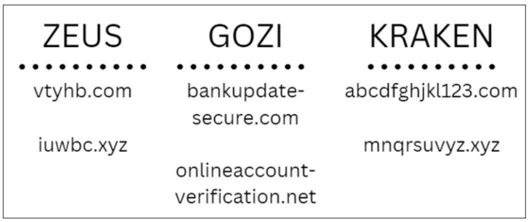

Botmaster
Attack
Infected chain of devices (bots)
Command and Control server (C&C)

Figure 1: Example of a botnet topology.

The DNS is a globally distributed system that plays a critical role in converting domain names into IP addresses and ensuring the continuous management of network traffic. Botnets exploit this system to hide and maintain their operations by using techniques such as DDNS, Domain Generation Algorithms (DGA), and fast-flux, which allow them to continuously modify domain information to evade blacklisting [1,2].

These techniques enable botmasters to dynamically change the IP addresses associated with malicious domains. Even if a specific domain or IP address is detected, the communication with infected devices can remain uninterrupted. DGA-based botnets utilize algorithms to generate numerous domain names, some of which are registered with DNS servers and used as Command and Control (C&C) server domains. Bots employ the same algorithm to sequentially generate domain names and query them from DNS servers. If a queried domain matches the C&C server's domain name, the correct name resolution occurs, and the IP address of the C&C server is returned. Subsequently, the bot sends packets to that IP address, establishing communication with the C&C server. Since DGAs often use current timestamps or random inputs, such as dictionary words or arbitrary strings, the generated domain names differ for every execution. This dynamic nature enables frequent updates of the C&C server's domain names, allowing botnets to evade blacklisting techniques. DGA-based botnets include Zeus, Conficker, Kraken, Gozi, Torpig, and Bedep (Figure 2).



ZEUS	GOZI	KRAKEN
vtyhb.com	bankupdate-secure.com	abcdfghjkl123.com
iuwbc.xyz		mnqrsuvyz.xyz
	onlineaccount-verification.net	

Figure 2: Some of the infamous DGA botnet examples

DNS-based botnet detection faces significant challenges in distinguishing malicious DNS traffic from legitimate DNS requests. The massive volume of DNS queries generated by internet networks makes it difficult to identify the subtle and concealed patterns of botnet traffic. Traditional signature-based and anomaly-based detection methods remain inadequate for addressing the rapidly evolving botnet techniques. Recent research has focused on utilizing advanced methods such as machine learning and deep learning to analyze large volumes of DNS data and detect patterns indicative of malicious activity. These approaches leverage features such as DNS query frequency, domain name age, and request intervals to identify botnet behavior more precisely. However, as detection mechanisms improve, attack trends and techniques evolve daily to bypass them. For instance, zero-day attacks often evade detection by avoiding identifiable patterns or signatures.

While DNS-based botnet detection methods have progressed, several challenges remain. Rule-based approaches tend to produce a high number of false positives when encountering legitimate traffic with anomalous patterns, while model-based approaches require large, well-labeled datasets and substantial computational resources for real-time operations. As a result, hybrid approaches that combine multiple detection strategies have gained preference due to their ability to cover diverse attack scenarios. Nevertheless, the evolving nature of botnets continues to demand more sophisticated, stable, and efficient DNS-based detection techniques with high accuracy and reduced latency.

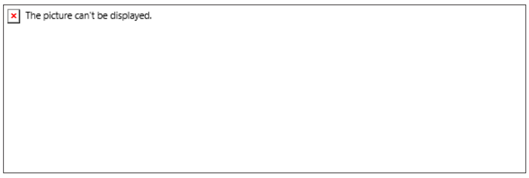


Figure 3: DNS query and DNS

Our research aims to contribute to the field of DNS query-based botnet detection by exploring how machine learning-enhanced models can improve detection accuracy and stability (Figure 3).

Overview of Botnet

The rapid growth of botnet proliferation poses significant challenges to cybersecurity, prompting research efforts to detect botnets effectively. This section provides a detailed overview of botnets, their lifecycle, and the importance of DNS-based machine learning detection methods. The lifecycle of a botnet consists of specific stages that focus on recruiting, managing, and utilizing compromised devices for attacks (Figure 4). Understanding these stages is essential for detecting botnet operations using DNS-based machine learning. The botnet lifecycle includes the following key stages [1,4,5].

Exploitation Stage

The botnet lifecycle begins by identifying vulnerabilities in internet-connected devices. These vulnerabilities are exploited through phishing attacks, malicious software downloads, or weakly secured systems, enabling the installation of malware that connects the device to the botnet network [2,6].

Command and Control (C&C) Stage

Once devices are infected, they connect to the botnet's C&C system. The C&C network serves as the central control mechanism for managing botnet operations. While older botnets relied on centralized C&C servers, modern botnets use decentralized architectures (e.g., DNS-based communication) to evade detection [7]. A decentralized C&C model prevents single points of failure, making it harder to disrupt botnet operations [3].

Attack Execution Stage

When activated, the botnet carries out various malicious activities, including DDoS attacks, data theft, spam distribution, and click fraud. For example, high-profile botnets such as Mirai exploit poorly secured IoT devices to orchestrate massive DDoS attacks [9,10]. Some botnets specifically target financial institutions, while others contribute to the spread of malware by distributing spam emails [2,5].

Update and Maintenance Stage

The final stage of the botnet lifecycle involves updating and maintaining the bots. The botmaster periodically upgrades the bots to improve their functionality and maintain control. When law enforcement or security researchers detect and disrupt botnets, botmasters may abandon the compromised devices or temporarily suspend operations to evade detection, reactivating them once conditions are safe. Some botnets can even self-destruct to prevent further detection or eliminate competition from rival botnets [11].

At the end of the update phase, the botnet reverts to the command and control establishment stage, where bots reconnect to the C&C server to receive new instructions and check for network readiness. This continuous cycle enhances the botnet's adaptability, making it easier to evade security measures.

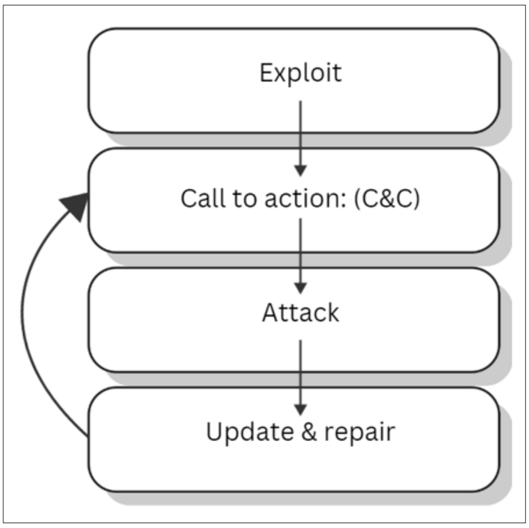


Figure 4: The botnet cycle.

Types of Botnets and DNS-Based Detection Methods

Botnets are utilized for malicious purposes such as data theft, spam distribution, and DDoS attacks, and they can be classified based on their structure and communication models. The main types are as follows:

Centralized Botnet

A centralized botnet uses a single C&C server to control the bots. It typically employs HTTP and IRC protocols for communication, making command transmission simple and effective. However, this model is vulnerable to a single point of failure, meaning that shutting down the C&C server can neutralize the botnet [1].

Peer-to-Peer (P2P) Botnet

P2P botnets operate without relying on a centralized C&C server. Each bot communicates directly with others, distributing control across the network. This decentralized structure makes it difficult to dismantle the botnet, as its functionality can persist even if some nodes are removed. However, P2P botnets are slower and face complexities in command distribution [2,3]. An example of a P2P botnet is GameOver Zeus, a sophisticated variant of the Zeus malware that emerged in 2011 and remained active until 2014. Unlike the original Zeus, GameOver Zeus utilized a P2P architecture for its C&C infrastructure, which made it more resilient to takedown operations and harder for law enforcement to disrupt. Like its predecessor, GameOver Zeus primarily targeted bank theft and personal data exfiltration, including login credentials, credit card information, and other sensitive data. In June 2014, a coordinated operation known as Operation Tovar—led by the FBI, Europol, and other law enforcement agencies—successfully dismantled the GameOver Zeus botnet. The operation targeted critical nodes within the P2P network, temporarily disrupting the botnet and enabling investigations into its creators and botmasters.

Hybrid Botnet

Hybrid botnets combine the advantages of centralized and P2P botnets. While they use a main C&C server for command dissemination, P2P nodes are integrated as an additional layer to enhance stability. This structure makes detection and neutralization of the botnet significantly more challenging [4,8].

DNS-Based Botnet Detection Methods

DNS is often used as a command transmission channel for botnets. DNS-based botnet detection methods aim to analyze DNS traffic to identify suspicious activity and can be classified as follows:

Flow-Based Analysis

This method observes the patterns in DNS traffic to detect abnormal frequencies or Time to Live (TTL) values associated with botnets. By analyzing DNS flow data, botnets utilizing synthetic DGA domains can be identified [1]. While flow-based analysis is suitable for real-time botnet detection, it requires significant capacity to process large volumes of data [5].

Signature-Based Detection

This method identifies botnets by comparing DNS queries against known signatures or malicious domain lists. Signature-based detection is effective for detecting specific botnets; however, it struggles to identify new or emerging botnets [7].

Anomaly-Based Detection

Anomaly-based detection methods identify abnormal DNS query behaviors. For example, it can detect spikes in DNS query frequency or unusual domain structures. This approach is effective for detecting new or unknown botnets but may produce a high rate of false positives, requiring careful tuning [6].

Machine Learning-Based Detection

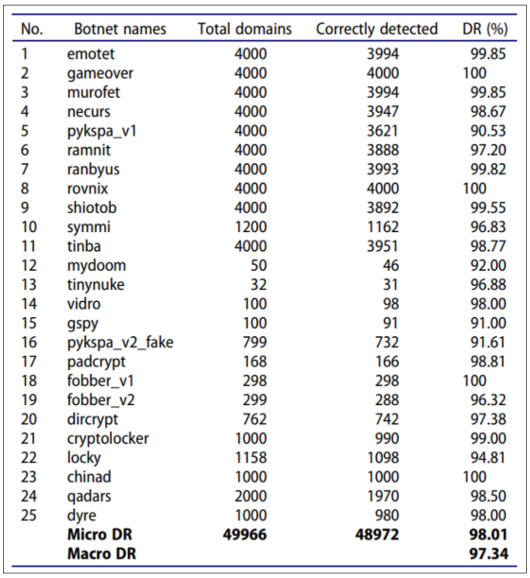
Machine learning methods enable classification of DNS traffic by leveraging features associated with botnet activity. For instance, domain name entropy, query frequency, and response times can be analyzed to identify botnet traffic. Machine learning approaches are adaptable to the evolving nature of botnets [9,10].

Hybrid Detection

Hybrid detection methods combine multiple detection techniques to improve botnet detection accuracy. This approach not only allows for the rapid detection of known botnets but also enables anomaly and behavior-based analysis to identify new botnet traffic [8].

Foundational Studies

Modern networks generate an enormous volume of DNS traffic, requiring systems capable of processing data quickly. Deep learning models, in particular, enable efficient real-time analysis of large-scale DNS queries to detect botnets in extensive networks [4]. For example, in Jiang and Zhou's (2018) research on DNS anomaly detection, deep neural networks (DNN) achieved over 95% accuracy in detecting botnet patterns, surpassing other detection methods [9].



No.	Botnet names	Total domains	Correctly detected	DR (%)
1	emotet	4000	3994	99.85
2	gameover	4000	4000	100
3	murofet	4000	3994	99.85
4	necurs	4000	3947	98.67
5	pykspa_v1	4000	3621	90.53
6	ramnit	4000	3888	97.20
7	ranbyus	4000	3993	99.82
8	rovnix	4000	4000	100
9	shiotob	4000	3892	99.55
10	symmi	1200	1162	96.83
11	tinba	4000	3951	98.77
12	mydoom	50	46	92.00
13	tinynuke	32	31	96.88
14	vidro	100	98	98.00
15	gspy	100	91	91.00
16	pykspa_v2_fake	799	732	91.61
17	padcrypt	168	166	98.81
18	fobber_v1	298	298	100
19	fobber_v2	299	288	96.32
20	dircrypt	762	742	97.38
21	cryptolocker	1000	990	99.00
22	locky	1158	1098	94.81
23	chinad	1000	1000	100
24	qadars	2000	1970	98.50
25	dyre	1000	980	98.00
	Micro DR	**49966**	**48972**	**98.01**
	Macro DR			**97.34**

Figure 5: Overvie w of prior research on different DGA botnets.

According to Jiang and Zhou (2018), decision trees demonstrated the ability to classify DGA domains with an accuracy exceeding 90% using DNS data [9]. Lee and Park's study revealed that RNN models achieved a 15% improvement in detection rates compared to non-deep learning approaches [4]. In network environments, precision and reduced false positives are critical, and ML-based approaches provide this advantage. High-accuracy detection systems ensure reliable performance without disrupting network operations. Kolini and Martin's research showed that SVM models achieved 92% accuracy, highlighting their importance in ensuring reliability within organizational networks [6]. Wang and collaborators demonstrated that clustering techniques could detect botnets in unsupervised conditions with an 87% accuracy [2,10]. Zeng et al. (2020) presented hybrid models that reduced false positives in high-DNS-query environments, achieving a 92% accuracy [10]. Xuan Dau Hoang and Xuan Hanh Vu introduced an advanced machine learning model using the Random Forest algorithm to classify DNS queries as legitimate or malicious for detecting DGA-based botnets. Their model improved detection results and reduced the false positive rate (FPR) and false negative rate (FNR) compared to previous detection systems [11].

Detecting Botnets Using Machine Learning

Machine Learning (ML)-based detection is one of the most reliable methods for identifying botnets, as it employs algorithms capable of learning patterns within DNS traffic. ML-based detection stands out due to its adaptability, accuracy, and ability to process large-scale data, surpassing other methods. By refining various data types and models, ML-based approaches effectively identify botnet threats. For example, the well-known Conficker botnet could generate up to 50,000 domain names per day, making immediate detection infeasible [6]. This limitation necessitated the use of ML techniques to recognize and detect such intricate patterns. Pattern-based detection relies on known patterns, which is insufficient for identifying new botnets. In contrast, ML algorithms detect botnet traffic based on behavior rather than predefined patterns. By analyzing DNS query frequency, domain name entropy, and unusual query patterns, ML models can detect continuously evolving botnets using Domain Generation Algorithms (DGA) [1,2]. In an ideal scenario, detecting and mitigating attacks at the network level is the most efficient way to prevent security issues before they occur. Network-level detection can block malicious packets before they reach targeted hosts, preventing their spread across internal and external networks. However, real-time network-level detection requires security devices to inspect every packet as it enters the network. Current security devices, such as next-generation firewalls and Intrusion Detection Systems (IDS), can analyze network traffic at speeds of up to 100 Gbps using signature-based methods. To process packets at this speed, the acceptable processing time per packet is minimal. Specifically, achieving a 10 Gbps throughput requires processing a 64-byte packet within 50 nanoseconds.

The primary advantage of ML is its ability to reduce false positives by refining detection criteria based on extensive training datasets. Supervised learning enables accurate classification of DNS requests, effectively distinguishing malicious queries. ML methods for DGA detection have become more effective by analyzing the characteristics of DNS queries. Different ML techniques improve the accuracy of DGA detection by classifying domains based on structural, behavioral, and network attributes.

Decision tree and Random Forest

Random Forests are more stable and reduce false positives by 15% because they combine multiple decision trees. A Random Forest consists of hundreds or thousands of decision trees, forming a "forest" or ensemble. Each tree is trained on a subset of the training data, making its conclusions independently. This technique is an example of bagging, where training data subsets are randomly selected to train each tree. Additionally, each tree analyzes different features and produces its own conclusion, reducing the likelihood of overfitting or misclassification based on specific features. For classification problems, Random Forests determine the result by taking the majority vote of all the trees' outputs. For example, if most trees classify the data into a particular category, that category will be chosen as the final result. By aggregating the conclusions of many trees, Random

Forests prevent bias and overfitting, thus improving accuracy (Figure 5).

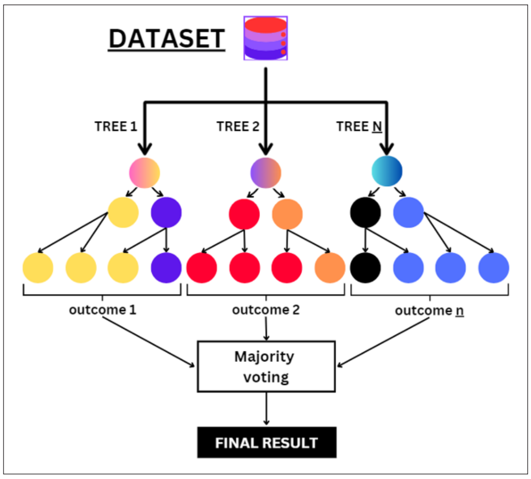


Figure 6: Random forest algorithm and bagging process.

Neural Networks and Deep Learning

Deep learning models, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), enhance DNS-based botnet detection. By processing sequences of DNS queries, RNNs learn temporal patterns, allowing the distinction between regular traffic and botnet traffic. Deep learning techniques are particularly effective in addressing challenges associated with encrypted DNS queries, such as DNS over HTTPS (DoH) and DNS over TLS (DoT) [6]. There are also unsupervised machine learning models which are effective for detecting unknown botnets. Clustering techniques, such as k-means or DBSCAN, group data based on similarity and can identify abnormal traffic patterns associated with botnets. Hybrid approaches combine multiple algorithms to improve detection accuracy. For example, anomaly detection techniques can first identify suspicious queries, and then a supervised classifier can confirm whether the queries are generated by DGAs.

Research Methodology and Results

In our study, we selected Random Forest, a machine learning ensemble method comprising multiple decision trees, as it is renowned for its robustness and accuracy. Random Forest is widely used for detecting DGA-based botnet DNS because it effectively captures distinctive patterns and highlights the differences between malicious and legitimate domains. For example, in botnets like Gozi, Random Forest leverages features such as domain length, entropy, and frequency distribution to detect the unique patterns of malicious bot domains.

Our study uses an advanced machine learning model employing the Random Forest algorithm to classify DNS queries as legitimate or malicious to detect DGA-based botnets. The model consists of two primary phases: training and detection (Figure 6). To enhance the detection of DGA-generated domains, we propose introducing the **Consonant-Vowel Ratio (CVR)** as the 25th feature. The CVR is a linguistic indicator that highlights the lack of phonetic and structural regularity commonly observed in DGA-generated domain names. Our findings reveal that while legitimate domain names maintain consonant-vowel harmony for readability, DGA domains tend to feature excessive consonants to evade detection and blacklisting.

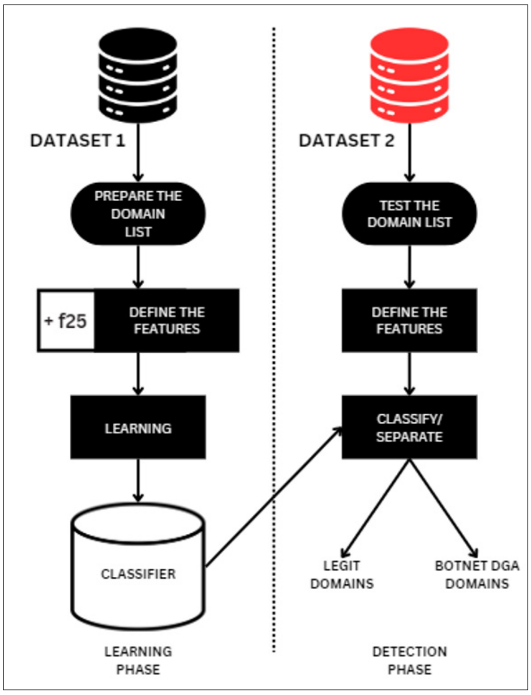


Figure 7: The proposed research method

Training and Detection Phases

The training phase involves building a classifier using a dataset of pre-labeled legitimate and botnet-generated domains. This phase comprises two key steps: feature extraction and model training. We incorporated 24 traditional DNS-based features along with Hoang and Vu's features, while adding the CVR as the 25th feature. Entropy measures the randomness in a domain name, which helps identify anomalous sequences. n-gram weighting quantifies the importance of specific character sequences to detect irregular patterns. Distribution of consonants, vowels, and digits helps identify abnormal combinations, as DGA domains often exhibit unusual patterns [11,8]. By combining CVR with other linguistic and statistical features, the model performs more comprehensive analysis. CVR complements entropy and n-gram frequency measures by capturing linguistic irregularities commonly observed in DGA domain names. Including CVR improves the classifier's accuracy, enabling it to better distinguish between legitimate and DGA-generated domains [5,6].

Table 1: The 25 features of the algorithm.

No.	Feature Name	Description
1	count(d)	Number of 2-grams in domain d.
2	m(d)	2-gram frequency distribution of d.
3	s(d)	2-gram weight of domain d.
4	ma(d)	Average 2-gram frequency of d.
5	sa(d)	Average 2-gram weight of d.
6	tan(d)	Average of popular 2-grams in d.
7	taf(d)	Avg. frequency of popular 2-grams in d.
8	ent(d)	2-gram entropy of domain d.
9	count(d)	Number of 3-grams in domain d.
10	m(d)	3-gram frequency distribution of d.
11	s(d)	3-gram weight of domain d.

12	ma(d)	Average 3-gram frequency of d.
13	sa(d)	Average 3-gram weight of d.
14	tan(d)	Average of popular 3-grams in d.
15	taf(d)	Avg. frequency of popular 3-grams in d.
16	ent(d)	3-gram entropy of domain d.
17	tanv(d)	Vowel distribution in d.
18	tanco(d)	Consonant distribution in d.
19	tandi(d)	Digit distribution in d.
20	tansc(d)	Special character distribution in d.
21	tanhe(d)	Hexadecimal character distribution in d.
22	is_digit	Whether the first character is a digit.
23	ent_char(d)	Character entropy of d.
24	eod(d)	Expected value of d.
25	cvr(d)	Consonant to vowel ratio of d

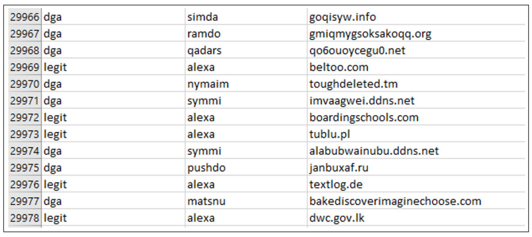
For example, the dataset we have used indicate that legitimate domains often have a lower consonant distribution due to linguistic patterns, while DGA-generated domains tend to have a higher vowel distribution (Feature 18) value because of their algorithmically generated randomness. This feature is crucial for distinguishing between legitimate and DGA domains in machine learning models like Random Forest. Our results demonstrate that integrating linguistics-based indicators such as CVR reduces false positives caused by domain names resembling legitimate traffic and enhances detection accuracy. Adding CVR to machine learning algorithms allows for detailed analysis of domain structural features, moving beyond the limitations of signature-based traditional methods. By incorporating CVR, this machine learning model aligns with emerging trends in botnet detection, improving the identification of both new and existing DGA domain names. The formula for calculating the **Consonant-Vowel Ratio (CVR)** is expressed as follows.

$$CVR = \frac{countco(d)}{countnv(d)} \tag{1}$$

Here, **countco(d)** represents the total number of consonants in the domain name, and **countnv(d)** represents the total number of vowels in the domain name. The **detection phase** follows a similar feature extraction process as the training phase but classifies incoming domains using the trained classifier. The domain names in the new DNS queries are converted into feature vectors identical to those used during training. Using the trained Random Forest classifier, the majority vote from each decision tree determines whether a domain is classified as legitimate or malicious. This phase is essential for ensuring high-efficiency real-time detection operations [11].

Test Results

The dataset used in the experiment includes 10,000 legitimate domains obtained from Alexa's top domains and 100,000 botnet domains generated by DGA botnets such as Gozi and (Figure 7), totaling 110,000 domains.



29966	dga	simda	goqisyw.info
29967	dga	ramdo	gmiqmygsoksakoqq.org
29968	dga	qadars	qo6ouoycegu0.net
29969	legit	alexa	beltoo.com
29970	dga	nymaim	toughdeleted.tm
29971	dga	symmi	imvaagwei.ddns.net
29972	legit	alexa	boardingschools.com
29973	legit	alexa	tublu.pl
29974	dga	symmi	alabubwainubu.ddns.net
29975	dga	pushdo	janbuxaf.ru
29976	legit	alexa	textlog.de
29977	dga	matsnu	bakediscoverimaginechoose.com
29978	legit	alexa	dwc.gov.lk

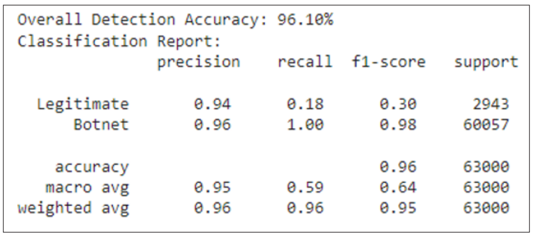
Figure 8: Sample of the dataset used for the training.

Initially, a data-cleaning procedure was performed to improve the quality of the dataset. This included:

- Removing duplicate query-response pairs.
- Eliminating incorrectly labeled query-response pairs.
- Removing query-response pairs showing errors or incomplete responses caused by network or name server issues.

Following the cleaning process, feature analysis of the dataset was conducted.

One unique aspect of this model is its ability to differentiate detection performance based on DGA botnet types. Botnets such as Emotet and Gameover, which are more distinguishable, achieved nearly perfect detection rates. However, for botnets like Banjori and Matsnu, which resemble legitimate domains, detection rates were lower, indicating areas for further improvement. Hoang and Vu's model demonstrated the efficiency of DNS-based botnet detection, proving to be effective for real-time DGA botnet detection with high accuracy and low false alarms. Incorporating our proposed 25th feature, the research achieved the following accuracy results (Figure 11):



```
Overall Detection Accuracy: 96.10%
Classification Report:
              precision    recall  f1-score   support

  Legitimate       0.94      0.18      0.30      2943
      Botnet       0.96      1.00      0.98     60057

    accuracy                           0.96     63000
   macro avg       0.95      0.59      0.64     63000
weighted avg       0.96      0.96      0.95     63000
```

Figure 9: The result #1.

The table above is a classification report for a machine learning model detecting DNS-based botnets. The legitimate label refers to DNS queries from normal, non-malicious traffic, while the Botnet label corresponds to malicious DNS queries. Key metrics provided are precision, recall, F1-score, and support. Precision measures how many instances predicted as a particular class are correct, while recall measures how many actual instances of a class were correctly identified. For the legitimate domains, precision is 0.94, meaning 94% of the predictions labeled as legitimate are correct, but recall is only 0.18, indicating that the model correctly identified only 18% of actual legitimate domains. Consequently, the F1-score for legitimate domains is 0.30, highlighting poor overall performance due to low recall. In contrast, the Botnet class demonstrates excellent results, with precision at 0.96, recall at 1.00, and an F1-score of 0.98, showing the model's near-perfect detection capabilities for botnet domains. The support column reveals the dataset's imbalance, with 2,943 legitimate instances and 60,057 botnet instances, contributing to the model's bias towards detecting botnets more effectively. The overall accuracy is 96.10%, meaning 96.1% of all predictions were correct, but accuracy alone does not reflect the poor performance for legitimate domains. The macro average provides the average precision, recall, and F1-score across both classes equally, while the weighted average adjusts these values according to class support, making it more representative of the dataset. Our report indicates that while the machine learning model excels at botnet detection, it **struggles to correctly identify legitimate domains**, a challenge caused by class imbalance and potentially feature limitations that needs to be addressed on future research. Improving recall for legitimate domains is necessary to achieve balanced and reliable detection performance when handling DNS queries that have non-legitimate domains which appear as legitimate.

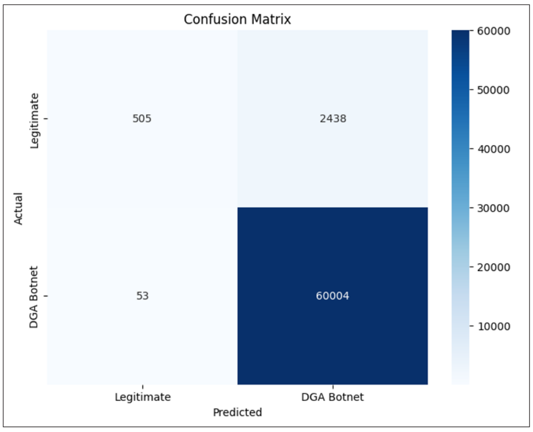


Figure 10: The confusion matrix for result #1.

The confusion matrix above shows that the model performs exceptionally well in detecting DGA botnet domains, correctly identifying 60,004 out of 60,057 (**true positives**) with only 53 false negatives, indicating a high recall for botnet detection. However, it struggles with legitimate domains, correctly classifying only 505 (true negatives) while misclassifying 2,438 as botnets (**false positives**).

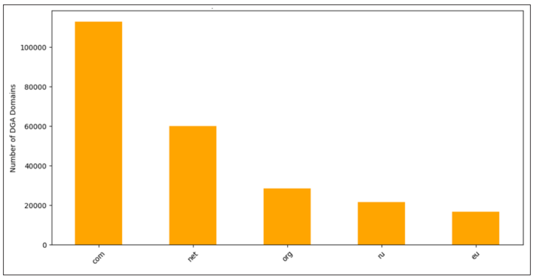
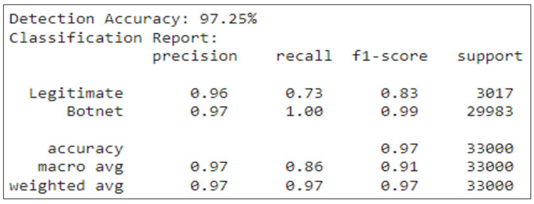


Figure 11: The most common TLD in botnets.

The bar chart above reveals that DGA botnets predominantly use **.com** domains, which are significantly more prevalent than other TLDs, likely due to their global recognition and ability to blend with legitimate traffic. The **.net** and **.org** TLDs follow as the second and third most common, highlighting their widespread use and association with legitimate organizations. Additionally, country-specific TLDs such as **.ru** (Russia) and **.eu** (European Union) also appear, suggesting regional targeting or specific origins This concentration of DGA domains in a few TLDs provides an opportunity for DNS query classification to focus monitoring efforts on these common endings, especially **.com**, to enhance detection. However, the presence of less common TLDs underscores the adaptability of botnets in exploiting less monitored or newer TLDs, making it essential to understand domain usage patterns for effective detection and mitigation strategies.

In the initial training phase, the model demonstrated near-perfect precision and recall for identifying botnet domain names, indicating high detection capability for domains generated by botnets like Cryptolocker and Tinba. However, the recall for legitimate domains was only 18%, suggesting that some legitimate domains were misclassified as botnet domains. This discrepancy may be due to the features used; while they were effective for identifying botnet domains, they may not fully represent the **characteristics of legitimate domains**.

In the subsequent training phase (Figure 12), the updated model achieved an overall accuracy of **97.25%**, which is a high result.



```
Detection Accuracy: 97.25%
Classification Report:
              precision    recall  f1-score   support

  Legitimate       0.96      0.73      0.83      3017
      Botnet       0.97      1.00      0.99     29983

    accuracy                           0.97     33000
   macro avg       0.97      0.86      0.91     33000
weighted avg       0.97      0.97      0.97     33000
```

Figure 12: The updated algorithm result #2.

The updated model shows significant improvements over the previous results, with an increase in overall detection accuracy from 96.10% to 97.25%. The detection of legitimate domains has improved dramatically, with precision increasing from 0.94 to 0.96, recall improving substantially from 0.18 to 0.73, and the F1-score rising from 0.30 to 0.83, indicating a much more balanced performance. While botnet detection remains consistent, with precision improving slightly from 0.96 to 0.97 and the F1-score increasing from 0.98 to 0.99, the recall for botnets remains perfect at 1.00. Additionally, the dataset, with a slightly higher proportion of legitimate domains (from 2,943 to 3,017) and fewer botnet domains (from 60,057 to 29,983), contributed to the model's ability to address the previous issue of poor recall for legitimate traffic, achieving a more robust and reliable detection system.

Conclusion

In recent years, botnets have become a significant cybersecurity threat, posing increasingly complex challenges for detection. DNS, which plays a fundamental role in translating domain names into IP addresses, serves as a key foundation for detecting botnet activity. This study delves into DNS-based botnet detection methods, evaluating their potential to mitigate cyber threats. The research demonstrates that DNS detection technology based on machine learning (ML) enhances the ability to identify botnet attacks and cyber threats at an early stage, thereby improving protection. The study highlights the effectiveness of DNS-based security methods in identifying cyber threats, with particular emphasis on the Random Forest algorithm, which proved effective in detecting botnet activity. Within the scope of this research, the advanced ML model proposed by Xuan Dau Hoang and Xuan Hanh Vu, which classifies DNS queries as legitimate or malicious, was examined. This model significantly reduced false positive (FPR) and false negative (FNR) rates compared to previous systems while improving detection accuracy to over 97%, outperforming traditional methods. However, the structural similarity of some botnets to legitimate domain names negatively impacted detection rates, underscoring areas for improvement. In conclusion, cybersecurity is essential today, as it safeguards the security of our data, the reliability of systems, and the safety of the cyber environment. As cybercrime grows increasingly sophisticated and potent, it is imperative to enhance detection outcomes and develop advanced approaches to effectively counteract these threats. This study demonstrates the feasibility of implementing a real-time DNS-based botnet detection model with high accuracy and low false alarm rates. While the model effectively detected various botnet attacks and enabled better threat control, challenges remain in identifying botnets that closely resemble legitimate domains. Future efforts to incorporate deep learning methods into botnet detection could expand the model's scope and improve its ability to address diverse threats more comprehensively.

References

1. Singh M, Singh M, Kaur S. Issues and challenges in DNS-based botnet detection: A survey. Computers & Security. 2019. 86: 28-52.
2. Wang Y, Zhou A, Liao S, Zheng R, Hu R, et al. A comprehensive survey on DNS tunnel detection. Computer Networks. 2021. 197: 108322.
3. Alieyan K, Almomani A, Manasrah A, Kadhum MM. A survey of botnet detection based on DNS. Neural Computing and Applications. 2017.
4. Lee J, Park Y. Botnet detection using deep learning techniques: A survey. IEEE Access. 2020. 8: 200749-200771.
5. Chen T, Yan Q, Wang Z, Liu Y. Detecting botnets with network behavior analysis using machine learning. IEEE Transactions on Dependable and Secure Computing. 2020. 17: 290-304.
6. Kolini F, Martin J. Understanding botnet behavior and network characteristics. *Future Internet. 2019. 11: 78.