Received: 21 March 2024 | Revised: 6 May 2024 | Accepted: 21 May 2024 | Published online: 31 May 2024

RESEARCH ARTICLE

Journal of Computational and Cognitive Engineering
2024, Vol. 3(3) 213–225
DOI: 10.47852/bonviewJCCE42022875

BON VIEW PUBLISHING

Data Science in Cybersecurity to Detect Malware-Based Domain Generation Algorithm: Improvement, Challenges, and Prospects

Mohamed Hassaoui[1], **Mohamed Hanini**[1] and **Said El Kafhali**[1,*]

[1]*Computer, Networks, Modeling, and Mobility Laboratory (IR2M), Hassan First University of Settat, Morocco*

Abstract: Nowadays, the malware communicates with command and control servers using domains generated algorithmically. Domain generation algorithms (DGAs) are continually evolving, which degrades the accuracy of the existing methods calls for the continuous tracking of how DGAs develop and their detection methods and calls for a good evaluation of the stage to open horizons for new detection methods. Data science plays a key role in cybersecurity by providing methods for detecting and analyzing network traffic data, including DGAs, and helping to improve the overall security of computer systems and networks. It can also be used to analyze large datasets of domain names and to develop and optimize solutions for DGA detection, by applying techniques such as machine learning, deep learning, and genetic algorithms, which have shown their effectiveness in detecting new and unknown DGAs. This paper reviews the role of data science in cybersecurity systems to detect DGAs. Hence, it also brings together publicly available domain name datasets and data science techniques utilized in recent DGA detection systems to highlight current issues and potential directions. This article additionally explains issues related to DGA detection. This will assist researchers in improving the current DGA detection algorithms as well as creating new powerful models.

Keywords: domain generation algorithms, data science, malware, cybersecurity, machine learning

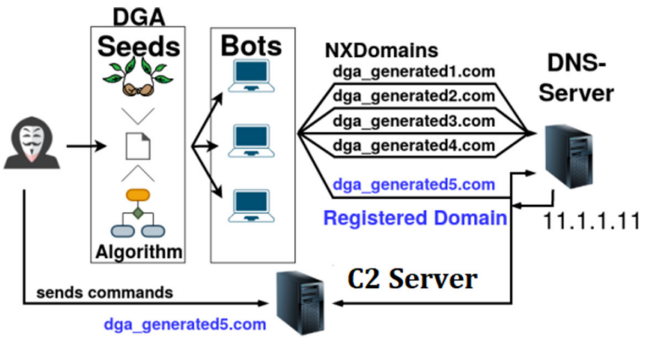
1. Introduction

It is important to keep in mind that technology is not inherently good or bad, but its use and application can be. As technology continues to advance, an attacker can use advanced technology to find new attack methods or identify vulnerabilities in a system. Therefore, it is also important for organizations to stay informed and adapt their cybersecurity strategies to mitigate the risks of attacks [1].

Domain generation algorithms (DGAs) have become popular tools for cybercriminals to evade detection and maintain control over their malware. Figure 1 represents a schematic illustration of DGA attacks. DGA is a technique used by malware to generate a large number of domain names that can be used as command and control (C2) servers [2]. The malware uses a DGA seed value, which is a starting point, and an algorithm to generate the domain names. The infected system will then periodically check the generated domain names to see if any of them are active [3]. If a C2 server is found, the infected system will connect to it and receive instructions. All these techniques are used by malware to evade detection and maintain persistent control of the infected system.

The random or pseudorandom nature of DGA-generated domain names makes it difficult to detect and block them using traditional methods. Furthermore, the fact that DGAs are always changing makes it necessary to continuously monitor how they evolve and their detection methods; it also necessitates a good review of the stage at which new detection methods can be developed.

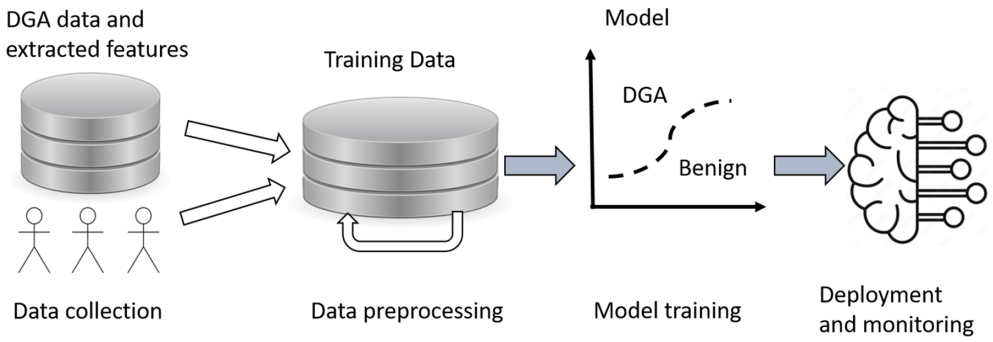
Figure 1
Schematic illustration of DGA attacks



Data science provides a powerful approach to detect DGA-generated domain names by analyzing patterns and features in the domain names and classifying them as malicious or benign. The process of DGA detection can be broken down into several steps such as presented in Figure 2, including data collection, data

*Corresponding author: Said El Kafhali, Computer, Networks, Modeling, and Mobility Laboratory (IR2M), Hassan First University of Settat, Morocco. Email: said.elkafhali@uhp.ac.ma

Figure 2
Data science process to detect DGAs



preprocessing, feature extraction by humans or automatically, model training and evaluation, and deployment and monitoring. The performance of DGA detection methods can be improved by using advanced machine learning (ML) techniques such as deep learning, which can automatically extract features from the data and improve the performance of the model. Additionally, assembling multiple models can also improve the performance and robustness of the DGA detection system.

If computer networks do not have a security plan that includes a thorough assessment of the steps to use innovative detection techniques, they are susceptible to attack by DGAs.

We have developed a number of important questions as part of our research methodology. As a result, this study has considerable significance since it explores the following important subjects:

1) What are the essential components or parameters that an attacker might take into account while creating a DGA?
2) What are the latest tactics employed by attackers to create DGAs, and how do they modify their approaches to keep one step ahead of security experts' detection and mitigation strategies?
3) What recent and essential features should be looked for while examining DGAs in order to identify and lessen malware?
4) What factors are most important to take into account when choosing a technique for gathering DGA data?
5) What effect does data preprocessing have when building a model that accurately identifies DGAs?
6) Which data science methods and tools may be used for the study and identification of DGA-generated domain names, and how can we leverage DGA data to enhance our capacity to recognize and avoid harmful communications?

With these questions in mind, this research aims to highlight current obstacles to DGA detection by consolidating current DGA construction methods and exploring how attackers adapt to circumvent detection. Additionally, the study reviews over 90 existing papers on publicly available labeled DGA detection, encompassing data sources, features, and ML models. Finally, the research examines the limits and strengths of existing methodologies to pave the way for new detection approaches.

This study employs a spectrum of data science techniques to delve into the role of cybersecurity systems in identifying DGAs. The methodology endeavors to evaluate the efficacy of diverse data science methodologies in fortifying cybersecurity measures, specifically in uncovering and mitigating threats originating from DGAs. By conducting a comprehensive analysis of various algorithms, methodologies, and their utilization in cybersecurity systems, the aim is to contribute to the comprehension of how data science can be utilized to reinforce the defenses against malicious activities, including DGA-based attacks.

The rest of this article is organized as follows. Section 2 gives an overview of algorithms used in DGA construction. Section 3 discusses the DGA detection methods using data science; their processes are detailed, and their challenges and prospects are summarized. Some selected works are studied and compared in Section 4. A summary of the main research issues and future directions in the studied field are given in Section 5. Section 6 gives a conclusion of this work.

2. Construction of Domain Generation Algorithms

The use of advanced methods of computation such as artificial intelligence (AI), machine learning (ML), and genetic algorithms can be both a powerful tool and a significant threat in cybersecurity and intrusion detection [4]. On one hand, these methods can help to improve cybersecurity by automating tasks such as intrusion detection, incident response, and threat intelligence. They can analyze large amounts of data from various sources, such as network traffic and log files, to identify patterns and anomalies that may indicate a security incident. They can also learn from past incidents to improve their ability to detect and respond to future incidents. On the other hand, these advanced computational methods can also be used by attackers to evade detection and launch sophisticated attacks. Adversaries can use AI and ML techniques to develop malware that can evade traditional security measures and impersonate legitimate users [5]. They can also use AI and ML to automate the process of launching and managing attacks, making it more difficult for defenders to identify and respond to them.

In the context of malware based on DGAs, the seed value plays a critical role in the construction DGA process; it is often hard-coded into the malware, and it is the starting point of the algorithm. The use of advanced methods of computation makes it difficult to identify the seed of DGAs, and if the seed value is difficult to identify, researchers cannot predict the next set of domain names generated by the DGA, which cannot help with blocking the communication between the malware and the command and control infrastructure.

There are many methods used to generate domain names, but the common goal of these methods is to make DGAs hard to be detected by security systems and take them down; some of these methods are complex and use advanced methods of computation. The authors of DGAs also use a top-level domain (TLD) list which is a program that generates domain names by using a list of TLDs such as .Com, .Net, .Org, etc. The program may use a specific rule to combine the TLDs with the domain names, such as appending them or prepending them.

The following are the methods used to construct DGAs:

1) **Using a date** [6]: An algorithm generates domain names using a specific date. The program may use a specific rule to combine the date with the domain names, such as appending the date in the form of YYYYMMDD. This technique has been used for several years as a way for malicious actors to evade detection and maintain access to their infrastructure.
2) **Combining words from a dictionary** [7, 8]: A program generates domain names by combining words from a dictionary with a set of characters or numbers. The program may use a specific rule to combine the words, such as concatenating them or inserting characters between them.
3) **Pseudorandom number generators** [9–11]: A program generates domain names using a pseudorandom number generator, which creates a sequence of numbers that appears random but is determined by a seed value. The generated domain names may be based on a specific pattern, such as combining a set of random words with a set of random characters.
4) **Hash-based** [12–14]: A program generates domain names by applying a hash function to a seed value. The generated domain names are determined by the input to the hash function and the specific hash algorithm used.
5) **Markov chain** [15–17]: A program generates domain names by using a Markov chain model, which is a mathematical model that describes a sequence of events in which the probability of each event depends on the state of the system in the previous event. The program generates domain names by simulating a sequence of characters based on the probabilities of the previous characters.
6) **Deep neural networks** [18–20]: Neural networks are designed to capture long-term dependencies in sequential data and to learn the underlying patterns and distributions in the data, such as Long Short-Term Memory (LSTM) or Generative Adversarial Networks (GANs). It can be used to generate domain names by training the network on a large dataset of existing domain names and using it to generate new domain names that are similar in structure.
7) **Genetic algorithm** [21, 22]: A program generates domain names by using a genetic algorithm, which is a method that mimics the process of natural selection to evolve solutions to a problem. The program generates a population of domain names, and through a process of selection, crossover, and mutation, it generates new domain names that fit more according to some fitness function.

In the context of DGA detection against construction methods of DGAs, it is a constant battle between the adversary and defender, with neither having a permanent advantage nor superiority. The adversary continually evolves his techniques to evade detection by using DGAs to dynamically construct domain names using construction methods, while the defender continually improves his methods for detecting and classifying DGAs through feature extraction methods to block malicious activities and protect the system. As technology and techniques on both sides continue to advance, the defender needs to stay up-to-date and proactive in his efforts to detect and prevent malicious activities. At the same time, the adversary will likely continue to find new ways to evade detection and carry out his malicious activities, so the battle will likely continue indefinitely.

We should never underestimate the intelligence of our adversaries, assuming that they are capable of understanding our plans, anticipating our actions, and coming up with countermeasures. When we often assume the intelligence of our adversaries, we improve our capacity to anticipate possible obstacles, which lowers risks and increases our chances of success.

By consolidating the techniques used in constructing DGAs, we've contributed to heightening awareness of the risks posed by DGA-based malware. This effort aids in recognizing recurring patterns and similarities in DGA creation, facilitating their detection and blocking, thus fortifying overall security against malicious domains and cyber assaults. Moreover, summarizing these methods facilitates a deeper comprehension and analysis of DGA evolution over time, enabling swifter and more effective responses to emerging threats.

3. DGA Detection Methods Using Data Science

Examining the domain names' length, entropy, and character frequency distribution are some of the features that are typically examined in order to identify DGAs using data science. These features are used as inputs to train ML models, which allow DGA-generated domains to be distinguished from real ones. Examples of these models are decision trees (DT) and random forests (RF).

3.1. Data science methods

The quality and the quantity of the data are crucial for the DGA detection. The more data you have, the better the detection rate will be [23]. Additionally, it is important to preprocess the data and remove duplicates and irrelevant data before using them for DGA detection. Therefore, this process should be done consistently and systematically. This section presents a brief overview of the techniques used to gather information about DGA-generated domains. Its role is to provide a concise overview of the methods used, including the tools and techniques used to collect data and to highlight the key findings and insights gained from the data collection process.

3.1.1. Summary of collection methods

The summary of data collection methods helps to provide context and an understanding of the data collection methods. There are several ways to collect data from domains for DGA detection:

1) **Passive data collection** [24, 25]: This involves observing Domain Name System (DNS) traffic to accumulate a significant dataset of domains. Achieving this can be accomplished through tools like tcpdump for capturing and examining network traffic or by configuring a DNS resolver to log all DNS queries and responses.
2) **Active data collection** [26]: This entails actively querying domain name registrars or WHOIS [27] databases to collect information about registered domains. This can be done using scripting languages such as Python or R to automate the process of collecting and parsing the data.
3) **Open-source intelligence (OSINT)** [28, 29]: This involves using publicly available information, such as social media to gather information about domains. This can be done using web scraping, data mining, or other techniques to extract information from online sources.
4) **Honeypots** [30–33]: This method involves setting up a server or a service that looks like a legitimate service but is used to trap DGA-generated domains. When a DGA-generated domain is accessed, it can be logged and added to the dataset.
5) **Crowdsourcing** [34]: This involves using a community of people to collect and label data. For example, a website could be set up where users can submit domains they suspect are DGA-generated, and then other users verify whether the domains are legitimate or not.
6) **Using Application Programming Interfaces (APIs)** [35]: Some companies, such as malware research companies, provide APIs to access their data. These APIs can be used to get information about domains and IP addresses and use them to improve the dataset.
7) **Using existing datasets**: There are also existing datasets of DGA-generated domains that are publicly available, such as the Alexa

Table 1
Challenges of DGA data collection

Challenges	Description	Prospects
Bias	DGAs data collection methods can be biased in various ways; for example, the sources used by OSINT method may be biased because it relies on publicly available information.	Collect large DGA datasets from a diverse range of sources over a long period.
Verification and validation	Verifying the accuracy and reliability of information collected through OSINT can be challenging and time-consuming.	Create a test environment of the DGAs collected, statistical analysis, and test.
Cost	Data collection methods can be expensive and time-consuming, especially when large amounts of data are being collected.	Use of parallel computing and GPU.
Data storage and management	Storing and managing large amounts of data can be a challenge, especially when data is being collected from multiple sources.	Develop a storage strategy, and implement data compression.
Inadequate data	Open-source information may not provide all the data needed for a specific analysis or research project.	Collaborate with OSINT and other sources. Utilize analytical tools such as machine learning algorithms.
Integration with other security tools	Integrating honeypots with other security tools, such as intrusion detection systems and firewalls, can be complex and require specialized knowledge.	Develop strategies to integrate honeypots with other security tools.
Limited participation	Crowdsourcing efforts may not reach a large enough audience, limiting the amount and diversity of data collected.	Develop strategies for crowdsourcing such as identifying the target audience and providing clear guidelines.
Data updates	Keeping existing datasets up-to-date and relevant can be challenging, as DGA patterns and behaviors change rapidly.	Automation of the data collection and web scraping process.

[36] dataset, which contains a list of the most popular websites on the Internet, or the Common Crawl [37] dataset, which contains a large number of web pages collected from the Internet, or UMUDGA dataset [38], which contains both benign and malicious domain names and can be used for training and evaluating ML algorithms for DGA detection. These datasets can be used as a starting point for DGA detection.

8) **Combining different sources**: It is also possible to combine data from different sources to create a more comprehensive dataset.

3.1.2. Discussion and challenges

In general, data collection methods can result in inaccurate or incomplete information due to human error, measurement error, or problems with the data collection instruments.

In addition to providing context, the summary of DGA data collection methods can also play an important role in evaluating the effectiveness and reliability of the methods used to detect DGAs. It can highlight potential limitations and biases in the data collection process and provide suggestions for improvements or alternative methods that could be used in the future. The summary given in Table 1 can also serve as a reference for future research and analysis, providing a historical record of the methods used and their outcomes. Overall, the summary of DGA data collection methods is a valuable tool for understanding and interpreting DGA data and for improving the quality and reliability of DGA research and analysis.

3.2. Features extracted from DGAs

It is important to note that not all DGAs use the same features and some DGAs are more sophisticated than others. Therefore, it is important to use multiple features to achieve the best detection rate. Additionally, it is also important to keep updating the extraction of features. Several types of features can be used to detect and classify DGAs such as those presented in Table 2. In general, we can categorize features extracted from domain names into two broad categories.

The first one contains features that depend on the specific execution of malware samples, which are performed in an efficient environment with a specific configuration and within a specific time frame, such as temporal feature, behavioral network feature, and DNS feature, and this is what we call them aware features.

Another category of features that are only linked to the domain name and which are therefore independent of contextual knowledge, such as statistical features, structural features, lexical features, etc., is what we call free features.

3.3. Data preprocessing

Data preparation is an essential step in getting raw data ready for analysis [57]. Data must be processed and transformed into a format that can be used for analysis and modeling in this process.

It is critical to understand that finding DGAs is a challenging task that requires technical skill in data science, including data pretreatment, as well as domain-specific understanding. An introduction to data preparation techniques is given in this part, along with an examination of the potential and difficulties that come with them.

3.3.1. Summary of data preprocessing

Data preprocessing, as shown in Figure 3, involves getting data ready for ML by cleaning it, combining it, making it consistent, and changing its format. These steps, like cleaning and organizing data, help make it more accurate and reliable. They also make it easier to spot important patterns and improve the quality of models. Plus, preprocessing can make data smaller and faster to work with, which saves time and makes analysis and decision-making more efficient.

Table 3 represents a summary of data preprocessing methods with descriptions and references.

3.3.2. Discussion: Challenges and prospects in data preprocessing

Preparing the data for analysis and modeling is a critical step that comes with many difficulties and restrictions. The following are some typical challenges in data preprocessing:

1) **Data quality:** The accuracy and dependability of the preprocessed data can be significantly impacted by the quality

Table 2
DGA features

Category	Type of features	Description and examples	References
Aware feature	Temporal features	These include the time at which the domain was registered or accessed and the frequency of domain name generation.	[39]
	Domain Name System (DNS)	These include the use of DNS data, such as the time of resolution, the IP address of the resolving server, and the response status code, to identify patterns or anomalies in the generated domain names.	[40, 41]
	Behavioral network features	These include the network behavior of the generated domains, such as the IP addresses they resolve to, the domains they connect to, and the type of traffic they generate.	[42]
Free feature	Lexical features	These include the use of specific words or phrases within the domain name, such as common dictionary words, or the use of specific letter or number patterns.	[43, 44]
	Linguistic features	These include the use of specific language in the domain name, such as words or phrases in a foreign language, or the use of homoglyphs (characters that look similar to other characters).	[44]
	Randomness features	These include the use of random characters or sequences within the domain name and the measurement of randomness of the domain name using statistical tests such as chi-square or Kolmogorov–Smirnov.	[43, 45]
	Graph-based features	These include the analysis of the relationship between domains, such as the topology of the domain name graph and the clustering coefficient of the domain names.	[46]
	Phonetic features	These include the use of phonetic similarity or phonetic distance measures to compare domain names and identify patterns or similarities between them.	[47, 48]
	Levenshtein distance	This measures the difference between two strings, this is a way to measure the similarity between two domain names.	[44, 49]
	Neural network-based features	These include using neural networks to extract features from the domain names, such as the use of convolutional neural networks or recurrent neural networks to analyze the domain names.	[50, 51]
	Statistical features	Include measures such as entropy, n-gram, and word frequency.	[44, 52, 53]
	Structural features	These include the length of the domain name, the presence of specific characters or character sequences, and the presence of specific TLDs.	[54, 55]
	Markov chain features	These include the use of Markov chain models to generate domain names and the analysis of the generated names to detect patterns or anomalies in the generated names.	[44, 56]
	Web-scrapped features	These can be used to extract information from domain names, such as the following: the domain names' authority, popularity, and TLD extensions and keywords.	[44]

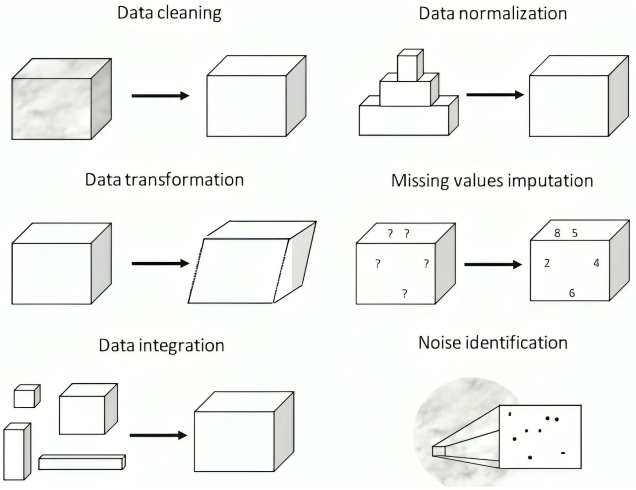
of the underlying data. Outliers, inconsistent measurement units, and incorrect data types can be challenging to find and fix when there are data quality problems.

2) **Missing values:** One of the biggest challenges in data preprocessing is handling missing values. Depending on the amount of missing values and the type of data, various methods, such as advanced imputation techniques or dropping the missing observations, may be needed to solve the problem.
3) **Data transformation:** Converting data into a format that can be analyzed can be difficult as well. To do this, data may need to be combined, new variables created, or categorical data turned into numerical data. For accurate results, choosing the right data transformation techniques is essential.

Although there are some difficulties in preprocessing data, there are many opportunities to increase the precision and dependability of data analysis and modeling results. Organizations can gain better insights and make better decisions based on their data by overcoming these obstacles and utilizing data preprocessing to the fullest extent possible. Here are the prospects for using data preprocessing in DGA detection.

Preprocessing can make it easier to extract more useful features from raw data, which can then be used to train machine learning models for DGA detection.

Figure 3
Data preprocessing methods



By ensuring that the data is consistent and well-formed, preprocessing can help to make the DGA detection system more resilient to new, unseen data.

Table 3
Data prepossessing methods

Methods	Description and examples	References
Data cleaning	This entails removing the dataset's inconsistent, redundant, or missing values.	[57–59]
Data normalization	This entails converting the data into a common scale or format. Normalization aids in minimizing the influence of variables with various scales on the analysis.	[57, 60, 61]
Data transformation	This entails transforming the data into a format that is better suited for analysis, such as changing category data into numerical data.	[57]
Data reduction	Aiming to reduce the size of the data and improve the performance of the models without sacrificing its information content, it contributes to reducing the dimensionality of the data and improving the results' interpretability.	[57, 62]
Feature selection	Because numerous features overlap and are frequently connected, it is necessary to choose a subset of pertinent features from the data for analysis. It is a specific case of data reduction.	[57, 63, 64]
Data resampling	To balance the class distribution, the data must either be over- or under-sampled. Resampling can enhance the effectiveness of machine learning models and help to avoid bias in the analysis of imbalanced data.	[57, 65]
Data augmentation	The goal of data augmentation is to increase the size of the training dataset by creating variations of existing samples, which can help to reduce overfitting and improve the generalization performance.	[57, 66]
Data integration	The goal of data integration is to provide a single, coherent, and consistent view of data, which is necessary for informed decision-making and efficient operations.	[67]

Automated data preprocessing can speed up DGA detection by decreasing the time and effort needed for manual data preparation.

By preparing the data for analysis, preprocessing can result in improved model performance, such as higher accuracy, precision, and recall in DGA detection.

3.4. Summary of DGA detection methods

This subsection provides an overview of the methodologies employed for DGA detection, including natural language processing (NLP), graph-based techniques, time series analysis, supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, and ensemble learning.

Overall, DGA detection using data science involves a combination of different techniques, depending on the nature of the DGA and the characteristics of the dataset, such as ML, network analysis, and statistical methods. The choice of techniques depends on various factors, including the specific DGA being used, the size and nature of the dataset, and the desired level of accuracy. In some cases, a combination of techniques may be necessary to accurately detect DGAs in a dataset.

3.4.1. Natural language processing (NLP)

NLP techniques [43, 68, 69] are utilized to analyze the patterns of characters in domain names and generate a probability score indicating the likelihood that a domain is DGA-generated. This involves processing textual data to extract features and train ML models.

NLP techniques can also be used to extract relevant features from domain names. This may include the following:

1) **Character-level features**: Length of the domain, frequency of certain characters, or character sequences.
2) **Word-level features**: Presence of specific keywords or patterns.
3) **Morphological features**: Prefixes, suffixes, or other linguistic elements within the domain name.

Although NLP techniques can be used in detecting DGA activity, they face significant challenges and weaknesses. One major difficulty is the rapidly evolving nature of DGAs, which can generate domain names that mimic legitimate ones, making it hard for NLP models to distinguish between malicious and benign domains. Additionally, DGAs can generate large volumes of domain names, overwhelming NLP-based detection systems and increasing the likelihood of false positives. Moreover, DGAs can adapt to evade detection by subtly altering their language patterns, further complicating NLP-based detection efforts. Limited applicability is another concern, as NLP techniques may not be well-suited for detecting DGAs in non-textual data sources such as network traffic. Furthermore, the computational complexity of NLP algorithms can pose challenges in real-time detection scenarios, where rapid analysis of large datasets is required. These challenges underscore the need for a multifaceted approach to DGA detection that combines NLP with other techniques such as ML and network analysis.

3.4.2. Graph-based techniques

Methods based on graphs [46, 70–73] entail building a graph with nodes representing domains and edges representing their similarity in order to analyze the structure of domains. Next, groupings of domains that were probably created by the same DGA are identified using clustering or community identification techniques.

While graph-based techniques offer the potential to detect DGA activity, they encounter significant difficulty. Graph-based approaches often involve constructing and analyzing large-scale graphs of domain names and their relationships, which can become computationally intensive and difficult to scale as the size of the graph increases. Additionally, ensuring the quality and reliability of the data used to construct these graphs is essential for accurate detection.

However, obtaining high-quality data, especially in dynamic environments where DGAs rapidly generate new domain names, can be challenging. Inaccurate or incomplete data can lead to false positives or negatives, undermining the effectiveness of graph-based detection methods. Addressing these challenges requires the development of efficient algorithms capable of handling large-scale graphs while also improving the quality and freshness of the data used for analysis. Moreover, integrating graph-based techniques with other detection approaches, such as ML and

behavioral analysis, can enhance the overall effectiveness of DGA detection systems.

3.4.3. Time series analysis

By analyzing patterns and fluctuations in domain name registration and usage over time, time series analysis [74–77] techniques can help identify anomalous behavior indicative of DGA-generated domains. These techniques can detect sudden spikes in domain registrations, unusual patterns in registration frequency, and other temporal anomalies that may indicate malicious activity. Additionally, time series analysis can be combined with other detection methods, such as ML algorithms, to enhance the accuracy and effectiveness of DGA detection systems.

Time series analysis also faces several challenges. One difficulty is the presence of noise and variability in the data, which can obscure the signals indicative of DGA-generated domains. Filtering out this noise while preserving relevant patterns requires robust preprocessing techniques and sophisticated analysis methods. Additionally, DGAs may exhibit subtle or evolving temporal patterns, making them harder to detect using traditional time series analysis approaches. Adapting algorithms to capture and respond to these dynamic behaviors is essential for accurate detection. Scaling up time series analysis to manage extensive datasets efficiently can present computational hurdles, particularly in real-time contexts. Furthermore, DGAs might display intricate or changing temporal patterns, which can complicate their detection using traditional time series analysis methods. It's crucial to adjust algorithms to effectively capture and react to these dynamic behaviors for precise detection. Additionally, in scenarios where timely analysis is crucial, such as real-time detection, handling large data volumes efficiently becomes even more challenging. Overcoming these obstacles demands ongoing enhancement of techniques and algorithms to guarantee successful DGA detection through time series analysis.

3.4.4. Supervised learning

Supervised learning techniques [78–80] where the model learns from labeled examples of both legitimate and malicious domain names to classify unseen domains involve training ML models on labeled data consisting of both benign and malicious domains. Models such as DT, RF, and hidden Markov models (HMM) are then used to predict the label (malicious or benign) of new, unseen domains.

However, several difficulties exist in using supervised learning for DGA detection. One challenge is the availability and quality of labeled data, as obtaining a comprehensive and accurate dataset of labeled malicious domains can be difficult due to the constantly evolving nature of DGAs and the dynamic nature of malicious activity. Additionally, DGAs can generate vast numbers of unique domain names, leading to class imbalance issues where there are far more legitimate domains than malicious ones. This imbalance can bias the model toward classifying most domains as legitimate, reducing its effectiveness in detecting DGAs. Moreover, DGAs can exhibit complex patterns and variations that may not be adequately captured by simple supervised learning models, requiring.

3.4.5. Unsupervised learning

Instead of requiring labeled data, unsupervised learning approaches [74, 81, 82] try to uncover the underlying structure of the data without any prior information. Patterns in domain data are found using algorithms like restricted Boltzmann machines (RBM) and K-means clustering.

Unsupervised learning for DGA detection faces several challenges, including dependence on data quantity and quality, finding the ideal number of clusters for domain grouping, poor compatibility between DGA families, and the challenge of assessing model performance in the absence of labeled data. These challenges impede the creation of reliable unsupervised techniques, calling for creative solutions to successfully resolve these problems.

3.4.6. Semi-supervised learning

Semi-supervised learning [73, 83, 84] includes using both labeled and unlabeled data to train models, which can be useful for detecting DGAs. This approach allows models to learn from a combination of labeled and unlabeled data, potentially improving detection accuracy.

This approach addresses the challenge of limited labeled data availability while also benefiting from the broader context provided by unlabeled data. However, designing effective semi-supervised learning algorithms for DGA detection requires careful consideration of the labeling process, selection of appropriate features, and model architecture to ensure optimal performance. Additionally, evaluating model performance in the absence of comprehensive labeled data remains a challenge in semi-supervised learning scenarios. Despite these challenges, semi-supervised learning holds promise for enhancing DGA detection capabilities by leveraging the combined strengths of labeled and unlabeled data.

3.4.7. Reinforcement learning

Reinforcement learning [85–88] techniques involve an agent learning to make decisions by taking actions in an environment to maximize a reward signal. Algorithms such as Q-learning and Deep Q Networks (DQN) can be applied to detect DGA activity. This approach addresses the challenge of the complexity of implementation, difficulty in defining the reward function, sensitivity to the choice of algorithm, and not enough labeled data available.

3.4.8. Ensemble learning

Ensemble learning [50, 89–92] techniques combine the predictions of multiple individual models to produce a more accurate overall prediction. Methods like RF, bagging, and boosting are employed to enhance the performance of DGA detection models. This approach addresses the challenges of computational complexity, overfitting, model interpretability, and the difficulty in selecting individual models.

At this stage of our investigation, we provide an overview of the methodologies employed for detecting DGAs, while also highlighting some of the challenges faced in this area. In the following, we will delve into the detailed challenges encountered in detecting DGA activities, along with an exploration of selected works that contribute to enhancing our understanding and methodologies for combating this threat landscape.

Table 4 presents the summary of the DGA detection methods using data science with descriptions, examples, references, and the challenges discussed.

3.5. Discussion: Challenges and prospects in DGA detection

The use of data science techniques holds great promise for DGA-based malware detection. However, it is a challenging task due to the dynamic nature of the threat and the complexity of the

Table 4
DGA detection methods

Methods	DGA detection methods	References	Challenges
Natural language processing (NLP)	To analyze the patterns of characters in the domain names and generate a probability score indicating the likelihood that a domain is DGA-generated.	[43, 68, 69]	Limited applicability, computational complexity
Graph-based techniques	To analyze the structure of the domains. This can be done by constructing a graph where each node represents a domain and edges represent similarity between them. Then, clustering or community detection algorithms can be applied to identify groups of domains that are likely to be generated by the same DGA.	[46, 70–73]	Scalability, data quality
Time series analysis	The idea is to monitor the domains generated over time and look for patterns such as sudden spikes in the number of domains generated in a short time, or patterns in the way the domains change over time.	[74–77]	Collecting time series data, preprocessing the data
Supervised learning	The basic idea is to train a machine learning model on labeled data that consists of both benign and malicious domains. The model is then used to predict the label (malicious or benign) of new, unseen domains, for example, DT, RF, and HMM.	[78–80]	No enough labeled data available, Evolving nature of DGAs, Feature engineering
Unsupervised learning	A machine learning technique that does not require labeled data. Instead, the algorithm tries to learn the underlying structure of the data without any prior knowledge, for example, K-means, and RBM.	[74, 81, 82]	Reliance on patterns in the data, number of clusters, limited interoperability, difficulty of evaluation
Semi-supervised learning	A type of machine learning that involves training a model on both labeled and unlabeled data. This approach can be useful for detecting DGAs, as it allows the model to learn from a combination of labeled data as well as unlabeled data that may contain similar patterns or characteristics.	[73, 83, 84]	Reliance on patterns in the data, number of clusters, limited interoperability, difficulty of evaluation
Reinforcement learning	A type of machine learning technique where an agent learns to make decisions by taking actions in an environment to maximize a reward signal, for example, Q-learning, DQN, and Actor-Critic (AC).	[85–88]	The complexity of implementation, difficulty in defining the reward function, sensitivity to choice of algorithm, not enough labeled data available
Ensemble learning	A machine learning technique that combines the predictions of multiple individual models to produce a more accurate overall prediction, for example, RF, bagging, and boosting.	[50, 89–92]	Computational complexity, overfitting, model interpretability, difficulty in selecting individual models

algorithms used. Table 4 presents some of the most important challenges in this context. The interpretation of these challenges is explained below:

1) **Limited applicability**: NLP is typically used for processing human language and may not be directly applicable to detecting DGAs, which are often generated using algorithms that are not based on natural language. However, the potential for further development in NLP-based DGA detection is high due to the constantly improving state of the NLP field, as well as the need to keep up with the evolving threats posed by DGA-based malware. New techniques and methods are developed such as domain adaptation in NLP with transfer learning and fine-tuning [93, 94]. These techniques allow models to quickly adapt to new data, which is particularly important in fields where new data is constantly being generated, and they can be applied to DGA detection to improve its accuracy and effectiveness, making it an exciting area of ongoing research and development.
2) **Computational complexity**: Most data science techniques can require a lot of computational resources to train, especially when dealing with large amounts of data or complex models. This can make it difficult to apply ML techniques in real-time or in resource-constrained environments. Additionally, some methods can have millions of parameters that need to be trained, which can be computationally expensive and time-consuming. To overcome these challenges, researchers and practitioners may need to use more efficient ML algorithms or employ techniques such as dimensionality reduction or model compression to reduce the computational complexity of models. Additionally, cloud-based computing and distributed processing can also be used to provide the necessary resources for data science applications.
3) **Scalability**: As the magnitude of the graph increases, the intricacy of employing graph-based methodologies can emerge as a bottleneck, particularly concerning extensive datasets. It is imperative for researchers and practitioners to proactively

Table 5
Comparison of selected works

References	Algorithms used	Related methods	RTD	Features	Data collection	Classification	Results
[90]	DBSCA, HMM, DNN	Ensemble learning, time series analysis	Yes	Aware and free	DNS records	Binary	Excellent
[89]	CNN-LSM-ANN	Ensemble leaning	Yes	No	Existing datasets	Binary	Excellent
[75]	DT	Supervised learning, time series analysis	Yes	Aware	DNS records	Multiclass	Good
[95]	RF, SVM, NB, XGB, DNN	Supervised leaning, NLP	Yes	Free	Existing datasets	Multiclass	Excellent
[96]	LSTM, neural embeddings	Unsupervised learning, NLP	No	No	DNS records	Binary and multiclass	Excellent
[73]	LSTM, neural embeddings	Semi-supervised learning, graph analysis, NLP	Yes	Free	DNS records	Multiclass	Excellent
[72]	Word graph	Graph analysis, NLP	Yes	Free	Existing datasets	Binary	Excellent
[78]	ResNets	Supervised leaning	Yes	No	Existing datasets	Binary and multiclass	Excellent
[50]	LSTM-CNN- SVM-RF	Ensemble leaning, NLP	Yes	Aware and free	Existing datasets	Binary	Excellent
[91]	RF-LSTM	Ensemble leaning	Yes	Aware and free	Passive data collection	Binary	Excellent
[82]	Auto encoder	Unsupervised leaning	Yes	No	OSINT, existing datasets	Binary	Excellent
[85]	Monte Carlo search, LSTM, CNN	Reinforcement learning	Yes	No	Existing datasets	Multiclass	Excellent
[97]	HMM	Supervised leaning	No	Free	DNS records	Multiclass	Good

address these scalability challenges and engineer graph-based techniques for DGA detection that are more adept and scalable.

4) **Data quality and availability**: The quality and accessibility of data wield substantial influence over its accuracy, presenting a notable challenge. Continuous and meticulous data collection regarding DGAs is essential, necessitating researchers to perpetually refine and update the dataset to remain at the forefront of the evolving DGA landscape.
5) **Feature engineering**: Feature engineering poses a significant challenge as it involves extracting pertinent features from domains. The selection of features holds considerable sway over the model's performance.
6) **Reliance on patterns in the data**: Unsupervised learning relies on the discovery of patterns within data without the need for prior knowledge or guidance, which can be difficult in the context of DGAs as the algorithms used to generate the domains are often dynamic and evolving.
7) **Model interpretability**: Because it is more challenging to pinpoint the causes of individual predictions when using multiple models in an ensemble, the overall solution may be more challenging to interpret and comprehend.
8) **Overfitting**: It is where the model performs poorly on unseen data because it is too closely fitted to the training data. Researchers and practitioners need to address these overfitting challenges and develop more efficient techniques for DGA detection.
9) **Sensitivity to choice of algorithm**: Ensemble learning and reinforcement learning are sensitive to the choice of algorithm and may require significant tuning to achieve good performance.

4. Detailed Comparison of Some Selected DGA Detection Works

It is not possible to give a thorough and detailed summary of all works in the field due to the vast amount. However, in Section 3.4, we have summarized the detection work of DGAs by methods and not by works without indicating details concerning the algorithm, data, features, type of classification, real-time detection (RTD), and outcome of each work.

The approach in this section aims to select publications from the methods presented in Section 3.4, provide a detailed analysis of their key aspects, and then compare their respective methodologies. In all the comparisons we make, we deem weak results whenever the findings presented in the works are with an accuracy below 75%, average if accuracy below 85%, accuracy good below 95%, and excellent above 95%. The results of this comparison are presented in Table 5.

5. Summary of the Main Research Issues and Future Direction

Our research has pinpointed various challenges and research domains in the realm of malware-based DGA detection using data science, aimed at extracting invaluable insights from pertinent data. These endeavors are instrumental in fostering intelligent decision-making for robust cybersecurity solutions. Identified challenges span data collection, preprocessing, feature extraction, and machine and deep learning. The most pivotal directions for exploration by researchers are succinctly outlined below:

1) **Direction 1**: Source datasets play a critical role. However, most existing domain name datasets are outdated and may not provide a complete understanding of the current behavioral patterns of various malware-based DGAs. Although the existing DGAs data can be processed to a meaningful level, there are still problems in the data as we have presented in the subsection 3.1.2.
2) **Direction 2**: DGAs data could be inconsistent with a certain DGA-based malware, noisy, incomplete, insignificant, unbalanced, or incomplete. Such flaws in a DGAs dataset could harm the learning process and hinder the effectiveness of models as we presented in subsection 3.3.2. As a result, it is

critical to comprehend these issues with cyber data and successfully address them using both current algorithms and recently developed algorithms for particular malware based on DGAs. This might be yet another crucial area for research in data science for cybersecurity.

3) **Direction 3**: As malware authors adapt and evolve their techniques, the features that are effective at detecting DGAs can also change; therefore, it is vital to employ multiple features and keep updating the extraction of DGAs features presented in Section 3.2 because not all DGAs use the same features and some DGAs are more advanced than others.
4) **Direction 4**: Detecting DGA using ML and deep learning methods poses several challenges such as presented in Section 3.4, and researchers must address these issues to improve the accuracy and reliability of the detection process.

6. Conclusion

In this article, we highlighted the mutually beneficial relationship between data science and malware-based DGA detection and talked about a few problems that call for the cooperation of various research communities. It has become essential to combine data science and cybersecurity to identify malware based on DGAs. This cutting-edge method has demonstrated impressive potential for spotting malicious activities, even those that are complex and well-hidden. The complexity of the algorithms, the requirement for frequent updates, and the need for enormous amounts of data present significant development and implementation challenges, but the advantages are obvious. The cybersecurity sector must keep innovating and evolving in order to keep up with society's growing reliance on technology and the threat landscape. The accuracy and efficiency of malware detection systems can be further increased by utilizing the most recent developments in data science and ML, providing enhanced protection to businesses, organizations, and individuals alike. In the end, this survey emphasizes the need for ongoing research and development in this area to keep up with the malware and cyberattacks constant evolution. Furthermore, the methodologies and techniques presented in this study can be adapted and integrated into existing cybersecurity frameworks to strengthen defenses against emerging threats. Overall, our research underscores the importance of leveraging data science in cybersecurity to stay ahead of cyber adversaries and safeguard digital assets and infrastructure in an increasingly interconnected world.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

References

[1] Tissir, N., El Kafhali, S., & Aboutabit, N. (2021). Cybersecurity management in cloud computing: Semantic literature review and conceptual framework proposal. *Journal of Reliable Intelligent Environments*, *7*(2), 69–84. https://doi.org/10.1007/s40860-020-00115-0

[2] Moşolea, V., & Oprişa, C. (2023). Detecting domain generation algorithms in malware traffic using constrained resources. In *2023 IEEE 19th International Conference on Intelligent Computer Communication and Processing*, 195–202. https://doi.org/10.1109/ICCP60212.2023.10398684

[3] Zhao, D., Li, H., Sun, X., & Tang, Y. (2023). Detecting DGA-based botnets through effective phonics-based features. *Future Generation Computer Systems*, *143*, 105–117. https://doi.org/10.1016/j.future.2023.01.027

[4] Hassaoui, M., Hanini, M., & El Kafhali, S. (2024). Unsupervised clustering for a comparative methodology of machine learning models to detect domain-generated algorithms based on an alphanumeric features analysis. *Journal of Network and Systems Management*, *32*(1), 18. https://doi.org/10.1007/s10922-023-09793-6

[5] Hassaoui, M., Hanini, M., & El Kafhali, S. (2023). A comparative study of neural networks algorithms in cyber-security to detect domain generation algorithms based on mixed classes of data. In *International Conference on Advanced Intelligent Systems for Sustainable Development*, 240–250. https://doi.org/10.1007/978-3-031-35251-5_23

[6] Yadav, V. K., Agarwal, S., Uprety, J., & Batham, S. (2014). SRTS: A novel technique to generate random text. In *2014 International Conference on Computational Intelligence and Communication Networks*, 268–272. https://doi.org/10.1109/CICN.2014.68

[7] Frankenberg-Garcia, A. (2020). Combining user needs, lexicographic data and digital writing environments. *Language Teaching*, *53*(1), 29–43. https://doi.org/10.1017/S0261444818000277

[8] Jiao, H., Wang, Q., Fan, Z., Liu, J., Du, D., Li, N., & Liu, Y. (2022). DGGCN: Dictionary based DGA detection method based on DomainGraph and GCN. In *2022 International Conference on Computer Communications and Networks*, 1–10. https://doi.org/10.1109/ICCCN54977.2022.9868932

[9] Naik, R. B., & Singh, U. (2024). A review on applications of chaotic maps in pseudo-random number generators and encryption. *Annals of Data Science*, *11*(1), 25–50. https://doi.org/10.1007/s40745-021-00364-7

[10] Ryan, C., Kshirsagar, M., Vaidya, G., Cunningham, A., & Sivaraman, R. (2022). Design of a cryptographically secure pseudo random number generator with grammatical evolution. *Scientific Reports*, *12*(1), 8602. https://doi.org/10.1038%2Fs41598-022-11613-x

[11] Yang, C., Taralova, I., El Assad, S., & Loiseau, J. J. (2022). Image encryption based on fractional chaotic pseudo-random number generator and DNA encryption method. *Nonlinear Dynamics*, *109*(3), 2103–2127. https://doi.org/10.1007/s11071-022-07534-z

[12] Hasan, H. A., Al-Layla, H. F., & Ibraheem, F. N. (2022). A review of hash function types and their applications. *Wasit Journal of Computer and Mathematics Science*, *1*(3), 75–88. https://doi.org/10.31185/wjcm.52

[13] Plohmann, D., Yakdan, K., Klatt, M., Bader, J., & Gerhards-Padilla, E. (2016). A comprehensive measurement study of domain generating malware. In *Proceedings of the 25th USENIX Conference on Security Symposium*, 263–278. https://dl.acm.org/doi/10.5555/3241094.3241115

[14] Wang, S., Zhao, H., Wang, Y., Huang, J., & Li, K. (2022). Cross-modal image–text search via efficient discrete class alignment hashing. *Information Processing & Management*, *59*(3), 102886. https://doi.org/10.1016/j.ipm.2022.102886

[15] Almutiri, T., & Nadeem, F. (2022). Markov models applications in natural language processing: A survey. *International Journal of Information Technology and Computer Science*, *14*(2), 1–16. https://doi.org/10.5815/ijitcs.2022.02.01

[16] Chandra, A., Bongulwar, A., Jadhav, A., Ahire, R., Dumbre, A., Ali, S., . . . , & Bhatti, S. (2022). Survey on randomly generating English sentences. *EasyChair Preprint: 7655*.

[17] Chen, J., Wu, Y., Jia, C., Zheng, H., & Huang, G. (2020). Customizable text generation via conditional text generative adversarial network. *Neurocomputing*, *416*, 125–135. https://doi.org/10.1016/j.neucom.2018.12.092

[18] Assael, Y., Sommerschield, T., Shillingford, B., Bordbar, M., Pavlopoulos, J., Chatzipanagiotou, M., . . . , & de Freitas, N. (2022). Restoring and attributing ancient texts using deep neural networks. *Nature*, *603*(7900), 280–283. https://doi.org/10.1038/s41586-022-04448-z

[19] Fatima, N., Imran, A. S., Kastrati, Z., Daudpota, S. M., & Soomro, A. (2022). A systematic literature review on text generation using deep neural network models. *IEEE Access*, *10*, 53490–53503. https://doi.org/10.1109/ACCESS.2022.3174108

[20] Kwon, H., & Lee, S. (2022). Ensemble transfer attack targeting text classification systems. *Computers & Security*, *117*, 102695. https://doi.org/10.1016/j.cose.2022.102695

[21] El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2021). Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, *165*, 113679. https://doi.org/10.1016/j.eswa.2020.113679

[22] Jain, A., Arora, A., Morato, J., Yadav, D., & Kumar, K. V. (2022). Automatic text summarization for Hindi using real coded genetic algorithm. *Applied Sciences*, *12*(13), 6584. https://doi.org/10.3390/app12136584

[23] Sordo, M., & Zeng, Q. (2005). On sample size and classification accuracy: A performance comparison. In *Biological and Medical Data Analysis: 6th International Symposium*, 193–201. https://doi.org/10.1007/11573067_20

[24] Maher, N. A., Senders, J. T., Hulsbergen, A. F., Lamba, N., Parker, M., Onnela, J. P., . . . , & Broekman, M. L. (2019). Passive data collection and use in healthcare: A systematic review of ethical issues. *International Journal of Medical Informatics*, *129*, 242–247. https://doi.org/10.1016/j.ijmedinf.2019.06.015

[25] Xiao, C., Lee, I., Dai, B., Schuurmans, D., & Szepesvari, C. (2022). The curse of passive data collection in batch reinforcement learning. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, *151*, 8413–8438.

[26] Roh, Y., Heo, G., & Whang, S. E. (2021). A survey on data collection for machine learning: A big data-AI integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, *33*(4), 1328–1347. https://doi.org/10.1109/TKDE.2019.2946162

[27] Elliott, K. (2009). The who, what, where, when, and why of WHOIS: Privacy and accuracy concerns of the WHOIS database. *SMU Science and Technology Law Review*, *12*(2), 141–172.

[28] Kanta, A., Coisel, I., & Scanlon, M. (2020). A survey exploring open source intelligence for smarter password cracking. *Forensic Science International: Digital Investigation*, *35*, 301075. https://doi.org/10.1016/j.fsidi.2020.301075

[29] Suryotrisongko, H., Musashi, Y., Tsuneda, A., & Sugitani, K. (2022). Robust botnet DGA detection: Blending XAI and OSINT for cyber threat intelligence sharing. *IEEE Access*, *10*, 34613–34624. https://doi.org/10.1109/ACCESS.2022.3162588

[30] Akiyama, M., Yagi, T., Yada, T., Mori, T., & Kadobayashi, Y. (2017). Analyzing the ecosystem of malicious URL redirection through longitudinal observation from honeypots. *Computers & Security*, *69*, 155–173. https://doi.org/10.1016/j.cose.2017.01.003

[31] Divya, T., Amritha, P. P., & Viswanathan, S. (2022). A model to detect domain names generated by DGA malware. *Procedia Computer Science*, *215*, 403–412. https://doi.org/10.1016/j.procs.2022.12.042

[32] Lee, S., Abdullah, A., & Jhanjhi, N. Z. (2020). A review on honeypot-based botnet detection models for smart factory. *International Journal of Advanced Computer Science and Applications*, *11*(6), 418–435.

[33] Sanjeev, K., Janet, B., & Eswari, R. (2020). Automated cyber threat intelligence generation from honeypot data. In *Inventive Communication and Computational Technologies: Proceedings of ICICCT 2019,* 591–598. https://doi.org/10.1007/978-981-15-0146-3_56

[34] Tong, Y., Zhou, Z., Zeng, Y., Chen, L., & Shahabi, C. (2020). Spatial crowdsourcing: A survey. *The VLDB Journal*, *29*, 217–250. https://doi.org/10.1007/s00778-019-00568-7

[35] Lomborg, S., & Bechmann, A. (2014). Using APIs for data collection on social media. *The Information Society*, *30*(4), 256–265. https://doi.org/10.1080/01972243.2014.915276

[36] Englehardt, S., & Narayanan, A. (2016). Online tracking: A 1-million-site measurement and analysis. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 1388–1401. https://doi.org/10.1145/2976749.2978313

[37] Patel, J. M. (2020). *Getting structured data from the internet: Running web crawlers/scrapers on a big data production scale*. USA: Apress.

[38] Zago, M., Pérez, M. G., & Pérez, G. M. (2020). UMUDGA: A dataset for profiling DGA-based botnet. *Computers & Security*, *92*, 101719. https://doi.org/10.1016/j.cose.2020.101719

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

[40] Schiavoni, S., Maggi, F., Cavallaro, L., & Zanero, S. (2014). Phoenix: DGA-based botnet tracking and intelligence. In *Detection of Intrusions and Malware, and Vulnerability Assessment: 11th International Conference,* 192–211. https://doi.org/10.1007/978-3-319-08509-8_11

[41] Schüppen, S., Teubert, D., Herrmann, P., & Meyer, U. (2018). *Feature-based automated NXDomain classification and intelligence.* Retrieved from: https://www.usenix.org/sites/default/files/conference/protected-files/security18_slides_schuppen.pdf

[42] Bisio, F., Saeli, S., Lombardo, P., Bernardi, D., Perotti, A., & Massa, D. (2017). Real-time behavioral DGA detection through machine learning. In *International Carnahan Conference on Security Technology*, 1–6. https://doi.org/10.1109/CCST.2017.8167790

[43] Hamroun, C., Amamou, A., Haddadou, K., Haroun, H., & Pujolle, G. (2022). A review on lexical based malicious domain name detection methods. In *2022 6th Cyber Security in Networking Conference,* 1–7. https://doi.org/10.1109/CSNet56116.2022.9955618

[44] Upadhyay, S., & Ghorbani, A. (2020). Feature extraction approach to unearth domain generating algorithms (DGAS). In *2020 IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International*

Conference on Cloud and Big Data Computing, and International Conference on Cyber Science and Technology Congress, 399–405. https://doi.org/10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00077

[45] Satoh, A., Nakamura, Y., Nobayashi, D., & Ikenaga, T. (2018). Estimating the randomness of domain names for DGA bot callbacks. *IEEE Communications Letters*, *22*(7), 1378–1381. https://doi.org/10.1109/LCOMM.2018.2828800

[46] Dong, Z., Chen, X., Zhao, J., Zhao, S., & Wu, J. (2022). Malicious domain name detection based on knowledge graph. In *2022 IEEE International Conference on Sensing, Diagnostics, Prognostics, and Control,* 251–256. https://doi.org/10.1109/SDPC55702.2022.9915824

[47] Peng, H., Ma, Y., Poria, S., Li, Y., & Cambria, E. (2021). Phonetic-enriched text representation for Chinese sentiment analysis with reinforcement learning. *Information Fusion*, *70*, 88–99. https://doi.org/10.1016/j.inffus.2021.01.005

[48] Manikandan, N., Ruby, D., Murali, S., & Sharma, V. (2022). Performance analysis of DGA-driven botnets using artificial neural networks. In *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions),* 1–6. https://doi.org/10.1109/ICRITO56286.2022.9965044

[49] Coates, P., & Breitinger, F. (2023). Identifying document similarity using a fast estimation of the Levenshtein Distance based on compression and signatures. *arXiv Preprint:2307.11496*. https://doi.org/10.48550/arXiv.2307.11496

[50] Hassaoui, M., Hanini, M., & El Kafhali, S. (2023). Domain generated algorithms detection applying a combination of a deep feature selection and traditional machine learning models. *Journal of Computer Security*, *31*(1), 85–105. https://doi.org/10.3233/JCS-210139

[51] Wang, Z., & Guo, Y. (2021). Neural networks based domain name generation. *Journal of Information Security and Applications*, *61*, 102948. https://doi.org/10.1016/j.jisa.2021.102948

[52] Cucchiarelli, A., Morbidoni, C., Spalazzi, L., & Baldi, M. (2021). Algorithmically generated malicious domain names detection based on n-grams features. *Expert Systems with Applications*, *170*, 114551. https://doi.org/10.1016/j.eswa.2020.114551

[53] Li, X., Li, Z., Xie, H., & Li, Q. (2021). Merging statistical feature via adaptive gate for improved text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(15), 13288–13296. https://doi.org/10.1609/aaai.v35i15.17569

[54] Ahluwalia, A., Traore, I., Ganame, K., & Agarwal, N. (2017). Detecting broad length algorithmically generated domains. In *Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments: First International Conference*, 19–34. https://doi.org/10.1007/978-3-319-69155-8_2

[55] Patsakis, C., & Casino, F. (2021). Exploiting statistical and structural features for the detection of domain generation algorithms. *Journal of Information Security and Applications*, *58*, 102725. https://doi.org/10.1016/j.jisa.2020.102725

[56] Yang, J., & Wang, H. H. (2010). Text classification algorithm based on hidden Markov model. *Journal of Computer Applications, 30*(9), 2348–2350.

[57] Maharana, K., Mondal, S., & Nemade, B. (2022). A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, *3*(1), 91–99. https://doi.org/10.1016/j.gltp.2022.04.020

[58] Chu, X., Ilyas, I. F., Krishnan, S., & Wang, J. (2016). Data cleaning: Overview and emerging challenges. In *Proceedings of the 2016 International Conference on Management of Data*, 2201–2206. https://doi.org/10.1145/2882903.2912574

[59] Ilyas, I. F., & Rekatsinas, T. (2022). Machine learning and data cleaning: Which serves the other? *Journal of Data and Information Quality*, *14*(3), 13. https://doi.org/10.1145/3506712

[60] Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, *97*, 105524. https://doi.org/10.1016/j.asoc.2019.105524

[61] Tang, L., Li, J., Du, H., Li, L., Wu, J., & Wang, S. (2022). Big data in forecasting research: A literature review. *Big Data Research*, *27*, 100289. https://doi.org/10.1016/j.bdr.2021.100289

[62] Namey, E., Guest, G., Thairu, L., & Johnson, L. (2008). Data reduction techniques for large qualitative data sets. In G. Guest & K. M. MacQueen (Eds.), *Handbook for team-based qualitative research* (pp. 137–161). Rowman & Littlefield Publishing Group.

[63] Dhal, P., & Azad, C. (2022). A comprehensive survey on feature selection in the various fields of machine learning. *Applied Intelligence*, *52*(4), 4543–4581. https://doi.org/10.1007/s10489-021-02550-9

[64] Venkatesh, B., & Anuradha, J. (2019). A review of feature selection and its methods. *Cybernetics and Information Technologies*, *19*(1), 3–26. https://doi.org/10.2478/cait-2019-0001

[65] Bagui, S., & Li, K. (2021). Resampling imbalanced data for network intrusion detection datasets. *Journal of Big Data*, *8*(1), 6. https://doi.org/10.1186/s40537-020-00390-x

[66] Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., & Hovy, E. (2021). A survey of data augmentation approaches for NLP. *arXiv Preprint:2105.03075*. https://doi.org/10.48550/arXiv.2105.03075

[67] Isaac, N. J., Jarzyna, M. A., Keil, P., Dambly, L. I., Boersch-Supan, P. H., Browning, E., . . . , & O'Hara, R. B. (2020). Data integration for large-scale models of species distributions. *Trends in Ecology & Evolution*, *35*(1), 56–67. https://doi.org/10.1016/j.tree.2019.08.006

[68] Balyan, R., McCarthy, K. S., & McNamara, D. S. (2020). Applying natural language processing and hierarchical machine learning approaches to text difficulty classification. *International Journal of Artificial Intelligence in Education*, *30*(3), 337–370. https://doi.org/10.1007/s40593-020-00201-7

[69] Kang, Y., Cai, Z., Tan, C. W., Huang, Q., & Liu, H. (2020). Natural language processing (NLP) in management research: A literature review. *Journal of Management Analytics, 7*(2), 139–172. https://doi.org/10.1080/23270012.2020.1756939

[70] Aravind, M., Sujadevi, V. G., Krishnan, M. R., AU, P. S., Pal, S., Vazhayil, A., . . . , & Poornachandran, P. (2022). Malicious node identification for DNS data using graph convolutional networks. In *2022 IEEE 7th International Conference on Recent Advances and Innovations in Engineering*, 104–109. https://doi.org/10.1109/ICRAIE56454.2022.10054347

[71] Deng, Z., Sun, C., Zhong, G., & Mao, Y. (2022). Text classification with attention gated graph neural network. *Cognitive Computation*, *14*(4), 1464–1473. https://doi.org/10.1007/s12559-022-10017-3

[72] Pereira, M., Coleman, S., Yu, B., DeCock, M., & Nascimento, A. (2018). Dictionary extraction and detection of algorithmically generated domain names in passive DNS traffic. In *Research*

in Attacks, Intrusions, and Defenses: 21st International Symposium, 295–314. https://doi.org/10.1007/978-3-030-00470-5_14

[73] Yan, F., Liu, J., Gu, L., & Chen, Z. (2020). A semi-supervised learning scheme to detect unknown DGA domain names based on graph analysis. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications,* 1578–1583. https://doi.org/10.1109/TrustCom50675.2020.00218

[74] Bäßler, D., Kortus, T., & Gühring, G. (2022). Unsupervised anomaly detection in multivariate time series with online evolving spiking neural networks. *Machine Learning*, *111*(4), 1377–1408. https://doi.org/10.1007/s10994-022-06129-4

[75] Bonneton, A., Migault, D., Senecal, S., & Kheir, N. (2015). DGA bot detection with time series decision trees. In *2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, 42–53. https://doi.org/10.1109/BADGERS.2015.016

[76] Cook, A. A., Mısırlı, G., & Fan, Z. (2020). Anomaly detection for IoT time-series data: A survey. *IEEE Internet of Things Journal*, *7*(7), 6481–6494. https://doi.org/10.1109/JIOT.2019.2958185

[77] Schmidl, S., Wenig, P., & Papenbrock, T. (2022). Anomaly detection in time series: A comprehensive evaluation. *Proceedings of the VLDB Endowment*, *15*(9), 1779–1797. https://doi.org/10.14778/3538598.3538602

[78] Drichel, A., Meyer, U., Schüppen, S., & Teubert, D. (2020). Analyzing the real-world applicability of DGA classifiers. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*, 15. https://doi.org/10.1145/3407023.3407030

[79] Jyothsna, P. V., Prabha, G., Shahina, K. K., & Vazhayil, A. (2019). Detecting DGA using deep neural networks (DNNs). In *Security in Computing and Communications: 6th International Symposium*, 695–706. https://doi.org/10.1007/978-981-13-5826-5_55

[80] Tuan, T. A., Long, H. V., & Taniar, D. (2022). On detecting and classifying DGA botnets and their families. *Computers & Security*, *113*, 102549. https://doi.org/10.1016/j.cose.2021.102549

[81] Gong, D., Liu, L., Le, V., Saha, B., Mansour, M. R., Venkatesh, S., & van den Hengel, A. (2019). Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1705–1714.

[82] Park, K. H., Song, H. M., Do Yoo, J., Hong, S. Y., Cho, B., Kim, K., & Kim, H. K. (2022). Unsupervised malicious domain detection with less labeling effort. *Computers & Security*, *116*, 102662. https://doi.org/10.1016/j.cose.2022.102662

[83] Cruciani, F., Moore, S., Quigley, B., Nugent, C. D., & Sani, S. (2021). Semi-supervised detection of algorithmically generated domains using neural network-based autoencoders. In *AI-CyberSec Workshop 2021: Workshop on Artificial Intelligence and Cyber Security*, 1–9.

[84] Yu, B., Pan, J., Gray, D., Hu, J., Choudhary, C., Nascimento, A. C., & de Cock, M. (2019). Weakly supervised deep learning for the detection of domain generation algorithms. *IEEE Access*, *7*, 51542–51556. https://doi.org/10.1109/ACCESS.2019.2911522

[85] Cheng, H., Fang, Y., Chen, L., & Cai, J. (2019). Detecting domain generation algorithms based on reinforcement learning. In *2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery,* 261–264. https://doi.org/10.1109/CyberC.2019.00051

[86] dos Santos, R. R., Viegas, E. K., Santin, A. O., & Cogo, V. V. (2023). Reinforcement learning for intrusion detection: More model longness and fewer updates. *IEEE Transactions on Network and Service Management*, *20*(2), 2040–2055. https://doi.org/10.1109/TNSM.2022.3207094

[87] Landen, M., Chung, K., Ike, M., Mackay, S., Watson, J. P., & Lee, W. (2022). DRAGON: Deep reinforcement learning for autonomous grid operation and attack detection. In *Proceedings of the 38th Annual Computer Security Applications Conference*, 13–27. https://doi.org/10.1145/3564625.3567969

[88] Nguyen, T. T., & Reddi, V. J. (2023). Deep reinforcement learning for cyber security. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(8), 3779–3795. https://doi.org/10.1109/TNNLS.2021.3121870

[89] Highnam, K., Puzio, D., Luo, S., & Jennings, N. R. (2021). Real-time detection of dictionary DGA network traffic using deep learning. *SN Computer Science*, *2*(2), 110. https://doi.org/10.1007/s42979-021-00507-w

[90] Li, Y., Xiong, K., Chin, T., & Hu, C. (2019). A machine learning framework for domain generation algorithm-based malware detection. *IEEE Access*, *7*, 32765–32782. https://doi.org/10.1109/ACCESS.2019.2891588

[91] Sivaguru, R., Peck, J., Olumofin, F., Nascimento, A., & de Cock, M. (2020). Inline detection of DGA domains using side information. *IEEE Access*, *8*, 141910–141922. https://doi.org/10.1109/ACCESS.2020.3013494

[92] Zhou, Y., Yang, L., Wang, Z., Li, G., & Ning, X. (2022). DNS attack detection based on multi-dimensional fusion model. In *2022 International Conference on Networking and Network Applications,* 74–81. https://doi.org/10.1109/NaNA56854.2022.00021

[93] Guo, Y., Shi, H., Kumar, A., Grauman, K., Rosing, T., & Feris, R. (2019). SpotTune: Transfer learning through adaptive fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 4805–4814.

[94] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., . . . , & He, Q. (2021). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, *109*(1), 43–76. https://doi.org/10.1109/JPROC.2020.3004555

[95] Wang, T., Chen, L. C., & Genc, Y. (2021). A dictionary-based method for detecting machine-generated domains. *Information Security Journal: A Global Perspective*, *30*(4), 205–218. https://doi.org/10.1080/19393555.2020.1834650

[96] Morbidoni, C., Spalazzi, L., Teti, A., & Cucchiarelli, A. (2022). Leveraging n-gram neural embeddings to improve deep learning DGA detection. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, 995–1004. https://doi.org/10.1145/3477314.3507269

[97] Fu, Y., Yu, L., Hambolu, O., Ozcelik, I., Husain, B., Sun, J., . . . , & Brooks, R. R. (2017). Stealthy domain generation algorithms. *IEEE Transactions on Information Forensics and Security*, *12*(6), 1430–1443. https://doi.org/10.1109/TIFS.2017.2668361

How to Cite: Hassaoui, M., Hanini, M., & El Kafhali, S. (2024). Data Science in Cybersecurity to Detect Malware-Based Domain Generation Algorithm: Improvement, Challenges, and Prospects. *Journal of Computational and Cognitive Engineering*, *3*(3), 213–225. https://doi.org/10.47852/bonviewJCCE42022875