ENHANCING CYBERSECURITY: A UNIFIED APPROACH USING EXPLAINABLE AI AND OPEN SOURCE INTELLIGENCE FOR BOTNET DETECTION

G. Sujatha[1], Rangu Sharath Chandra[2], Mettu Rohith Reddy[2], B. Kushwanth Varma[2]

[2]UG Scholar, [1,2]Department of Computer Science and Engineering

[1,2]Kommuri Pratap Reddy Institute of Technology, Ghatkesar, Hyderabad, Telangana.

ABSTRACT

Botnets, networks of compromised computers controlled by malicious actors, pose a significant cybersecurity threat. Detecting these threats is particularly challenging when attackers use Domain Generation Algorithms (DGA) to dynamically generate domain names for command-and-control servers. Enhancing botnet DGA detection is crucial for effectively identifying and mitigating cyber threats. Historically, cybersecurity has been an ongoing arms race, with DGAs evolving to evade traditional detection methods like static signatures, heuristics, and rule-based systems. These traditional approaches lack the adaptability and contextual understanding needed to keep up with the dynamic nature of cyber threats. This research focuses on enhancing botnet DGA detection by incorporating explainable AI and open-source intelligence (OSINT) for cyber threat intelligence sharing. Traditional methods struggle with the sophistication of modern DGAs, necessitating advanced techniques that not only detect malicious behavior but also provide explainable insights into why an activity is considered a threat. Explainable AI increases trust and understanding by elucidating the decision-making process of detection algorithms. Additionally, OSINT is critical for fostering a collaborative defense ecosystem, enabling organizations to collectively respond to emerging threats. This enhancement is vital due to the increasing sophistication of cyber threats and the need for a more adaptive and transparent approach to cybersecurity.

Keywords: Cyber Threat, Intelligence Sharing, Open-Source Intelligence, DGA Detection, Dynamic cyber threats.

1. INTRODUCTION

The landscape of cybersecurity is marked by an ongoing battle between cybersecurity professionals and malicious actors, with one of the formidable challenges being the detection and mitigation of botnets—networks of compromised computers under the control of malevolent entities. This threat is particularly potent when attackers employ Domain Generation Algorithms (DGA) to dynamically generate domain names for their command-and-control servers, adding a layer of complexity to detection efforts. The imperative to enhance botnet DGA detection is paramount for effective identification and mitigation of cyber threats. The historical context reveals the continuous evolution of tactics and techniques employed by malicious actors in the cybersecurity realm. Domain Generation Algorithms have emerged as a prevalent method for botnets to evade detection by security systems. Early detection methods primarily relied on static signatures, heuristics, and rule-based systems. While these approaches demonstrated some effectiveness, they proved inadequate when confronted with novel or polymorphic DGAs that constantly adapt to avoid detection. Traditional systems, lacking in adaptability and contextual understanding, struggle to keep pace with the dynamic nature of cyber threats.

The problem at hand centers around the need to enhance botnet DGA detection by incorporating advanced technologies such as explainable artificial intelligence (AI) and open-source intelligence

(OSINT) for cyber threat intelligence sharing. Traditional methods, although proficient in certain scenarios, face challenges in dealing with the sophistication of DGAs. There is a pressing need for more advanced techniques that not only detect malicious behavior but also provide explainable insights into why a specific activity is deemed a threat.Explainable AI becomes a crucial component, offering transparency into the decision-making process of detection algorithms. This not only increases trust in the system but also enables cybersecurity professionals to gain a deeper understanding of the threat landscape. Open-source intelligence sharing emerges as a critical element in the quest for a collective defense approach. Effective communication and collaboration among organizations are imperative to create a collaborative defense ecosystem where collective responses to emerging threats can be orchestrated. The necessity for enhancing botnet DGA detection arises from the escalating sophistication of cyber threats. Attackers continually refine their techniques, rendering traditional systems less effective in providing robust protection. The incorporation of explainable AI and the promotion of open-source intelligence sharing reflect a proactive response to these challenges. By leveraging advanced technologies and fostering collaborative efforts, the cybersecurity community endeavors to stay ahead in the perpetual arms race against evolving and increasingly sophisticated cyber threats.

2. LITERATURE SURVEY

F. Skopik et al. [1] conducted a comprehensive survey on collective cyber defense through security information sharing. They explored the dimensions and benefits of sharing security information among organizations, highlighting its role in enhancing overall cyber defense capabilities through improved threat awareness and collaborative defense strategies. R. Confalonieri et al. [2] provided a historical perspective on explainable artificial intelligence (XAI), tracing its evolution from inception to recent advancements. Their study emphasized significant milestones and developments in making AI systems interpretable and transparent, fostering trust and usability in AI applications. M.-A. Clinciu and H. Hastie [3] presented a survey on the terminology used in explainable AI (XAI), aiming to standardize concepts for better communication within the research community. Their work addressed the need for clarity and consistency in XAI terminology to facilitate advancements and collaborations in the field. J. M. Alonso et al. [4] conducted a bibliometric analysis of the explainable AI research field, analyzing publication trends and influential works. Their study provided insights into the growth and impact of XAI research, identifying key authors and pivotal contributions shaping the field's development. M. Singh et al.

[5] surveyed the issues and challenges in DNS-based botnet detection, emphasizing techniques and difficulties in detecting botnets through DNS traffic analysis. Their research highlighted the importance of advanced detection methods to combat evolving botnet threats effectively. T. S. Wang et al. [6] developed DBod, a clustering and detection method for DGA-based botnets using DNS traffic analysis. Their approach aimed to enhance detection accuracy by clustering and analyzing DNS traffic patterns associated with botnet activities, improving cybersecurity defenses against DGA-generated threats. X. D. Hoang and X. H. Vu [7] proposed an improved model for detecting DGA botnets using the random forest algorithm. Their study focused on leveraging machine learning techniques to enhance the detection rates of malicious domain names generated by DGAs, contributing to more robust botnet detection systems. B. Yu et al. [8] introduced a character-level based detection method for DGA domain names, utilizing neural networks to identify and mitigate malicious domain activities. Their research offered a novel approach to enhancing botnet detection capabilities through advanced pattern recognition techniques. L. Sidi et al. [9] proposed MaskDGA, an evasion attack against DGA classifiers and adversarial defenses, demonstrating techniques to evade detection systems. Their study underscored the need for resilient security measures to counter

sophisticated adversarial attacks in botnet detection. J. Peck et al. [10] developed CharBot, a method for evading DGA classifiers, highlighting challenges in maintaining effective botnet defenses. Their research focused on developing evasion techniques to bypass existing detection mechanisms, addressing ongoing vulnerabilities in cybersecurity frameworks. H. S. Anderson et al. [11] presented DeepDGA, an adversarially-tuned domain generation and detection method. Their study integrated deep learning techniques to enhance the resilience of DGA detection systems against adversarial attacks, contributing to more robust cybersecurity measures. T. D. Wagner et al. [12] conducted a survey on cyber threat intelligence sharing, exploring benefits, challenges, and research directions. Their study emphasized the importance of collaborative efforts in sharing threat intelligence to strengthen collective cybersecurity defenses and mitigate emerging threats effectively. W. Tounsi and H. Rais [13] surveyed technical threat intelligence amid sophisticated cyber attacks, examining methodologies and tools for gathering and utilizing threat intelligence. Their research provided insights into advanced techniques essential for defending against complex cyber threats. P. Pawlinski et al. [14] discussed actionable information for security incident response, focusing on identifying and sharing relevant information to mitigate security threats effectively. Their work emphasized the importance of timely and accurate information exchange in incident response protocols.

3. PROPOSED SYSTEM

The research encompasses a robust framework for enhancing the detection of DGA (Domain Generation Algorithm) botnets, incorporating explainable AI (Artificial Intelligence) and open-source intelligence for Cyber Threat Intelligence Sharing. The code is organized into several distinct functions, each serving a specific purpose in the overall process.

- **Dataset Upload and Exploration**: Users initiate the process by uploading a DGA dataset through the Tkinter-based GUI. The chosen dataset is then displayed, showcasing a snippet of the data and a bar graph illustrating the distribution of normal and DGA botnet classes.
- **Preprocessing:** After dataset upload, the code preprocesses the data to ensure it is suitable for machine learning algorithms. Missing values are replaced with zeros, and the StandardScaler from scikit-learn is applied to normalize the features. The dataset is split into training and testing sets for subsequent model training and evaluation.
- **Random Forest, Logistic Regression, Naive Bayes, Extra Tree, and Ensemble Model Training:** The code proceeds to train multiple machine learning models, including Random Forest, Logistic Regression, Naive Bayes, Extra Tree, and an Ensemble Algorithm (Bagging Classifier). Each model is trained using the preprocessed dataset, and their performances are evaluated using key metrics such as accuracy, precision, recall, and F1-score.
- **Extension with XGBoost:** The proposed extension involves incorporating XGBoost, an efficient and scalable gradient boosting framework. The XGBoost classifier is trained on a subset of the data, and its performance is evaluated alongside the existing models.
- **Explainable AI (XAI) using Shapley Values:** The code employs the SHAP (SHapley Additive exPlanations) library to provide insights into the feature importance of the trained Random Forest model. The Shapley XAI graph visualizes how each feature contributes to model predictions, enhancing the interpretability of the model.
- **Prediction on Test Data:** The final functionality allows users to upload new test data, and the trained Random Forest model predicts whether each instance belongs to the normal class or the DGA botnet class. The results are displayed in the Tkinter GUI, indicating the model's predictions for each test data point.

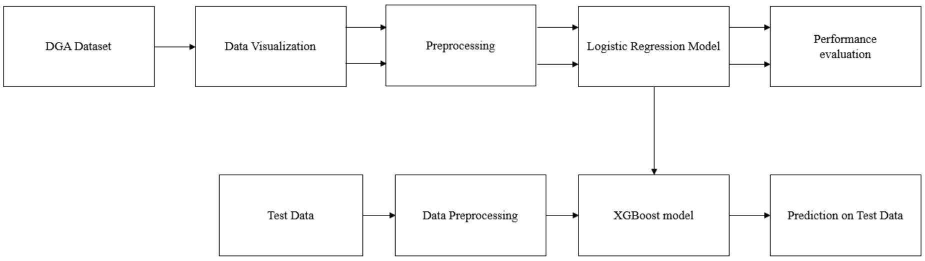

Fig.1: Block Diagram of Proposed system.

XGBoost Model

XGBoost is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "XGBoost is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the XGBoost takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

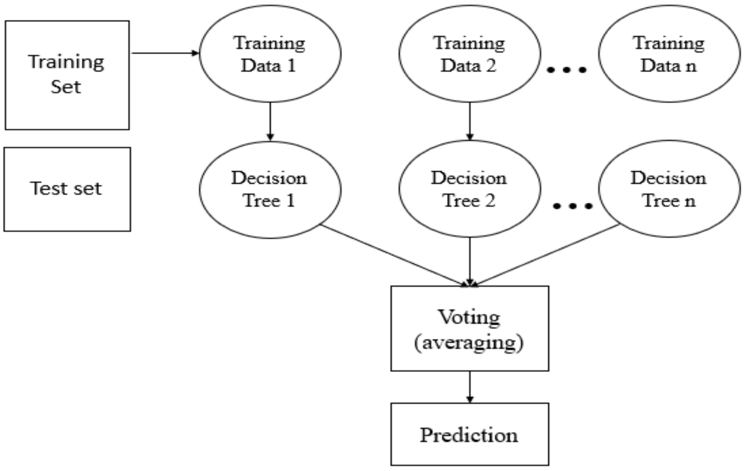

Fig. 2: XGBoost algorithm.

XGBoost, which stands for "Extreme Gradient Boosting," is a popular and powerful machine learning algorithm used for both classification and regression tasks. It is known for its high predictive accuracy and efficiency, and it has won numerous data science competitions and is widely used in industry and academia. Here are some key characteristics and concepts related to the XGBoost algorithm:

- **Gradient Boosting:** XGBoost is an ensemble learning method based on the gradient boosting framework. It builds a predictive model by combining the predictions of multiple weak learners (typically decision trees) into a single, stronger model.
- **Tree-based Models:** Decision trees are the weak learners used in XGBoost. These are shallow trees, often referred to as "stumps" or "shallow trees," which helps prevent overfitting.
- **Objective Function:** XGBoost uses a specific objective function that needs to be optimized during training. The objective function consists of two parts: a loss function that quantifies the error between predicted and actual values and a regularization term to control model

complexity and prevent overfitting. The most common loss functions are for regression (e.g., Mean Squared Error) and classification (e.g., Log Loss).

- **Gradient Descent Optimization:** XGBoost optimizes the objective function using gradient descent. It calculates the gradients of the objective function with respect to the model's predictions and updates the model iteratively to minimize the loss.
- **Regularization:** XGBoost provides several regularization techniques, such as L1 (Lasso) and L2 (Ridge) regularization, to control overfitting. These regularization terms are added to the objective function.
- **Parallel and Distributed Computing:** XGBoost is designed to be highly efficient. It can take advantage of parallel processing and distributed computing to train models quickly, making it suitable for large datasets.
- **Handling Missing Data:** XGBoost has built-in capabilities to handle missing data without requiring imputation. It does this by finding the optimal split for missing values during tree construction.
- **Feature Importance:** XGBoost provides a way to measure the importance of each feature in the model. This can help in feature selection and understanding which features contribute the most to the predictions.
- **Early Stopping:** To prevent overfitting, XGBoost supports early stopping, which allows training to stop when the model's performance on a validation dataset starts to degrade.
- **Scalability:** XGBoost is versatile and can be applied to a wide range of machine learning tasks, including classification, regression, ranking, and more.
- **Python and R Libraries:** XGBoost is available through libraries in Python (e.g., **xgboost**) and R (e.g., **xgboost**), making it accessible and easy to use for data scientists and machine learning practitioners.

4. RESULTS AND DISCUSSION

4.1 Dataset Description

The dataset columns has features that is used for analyzing and classifying domains, particularly in the context of distinguishing between legitimate and potentially malicious domains.

Here's a brief description of each column:

- IRad: This column is not immediately clear without additional context. It represent a feature about domains.
- Entropy: This column represent the entropy of the domain name. Entropy is a measure of randomness, often used in information theory to quantify uncertainty.
- RE-Alexa: This column represents a feature related to the domain's ranking in the Alexa web traffic ranking.
- Min-RE-Botnets: This column represents a minimum value of feature associated with botnets. Botnets are networks of compromised computers used for malicious activities.

4.2 Results Description

This figure 3 depicts the main graphical user interface (GUI) application designed for the purpose of enhancing botnets Domain Generation Algorithm (DGA) detection. The application appears to incorporate Explainable AI and Open-Source Intelligence for cybersecurity threat intelligence sharing.

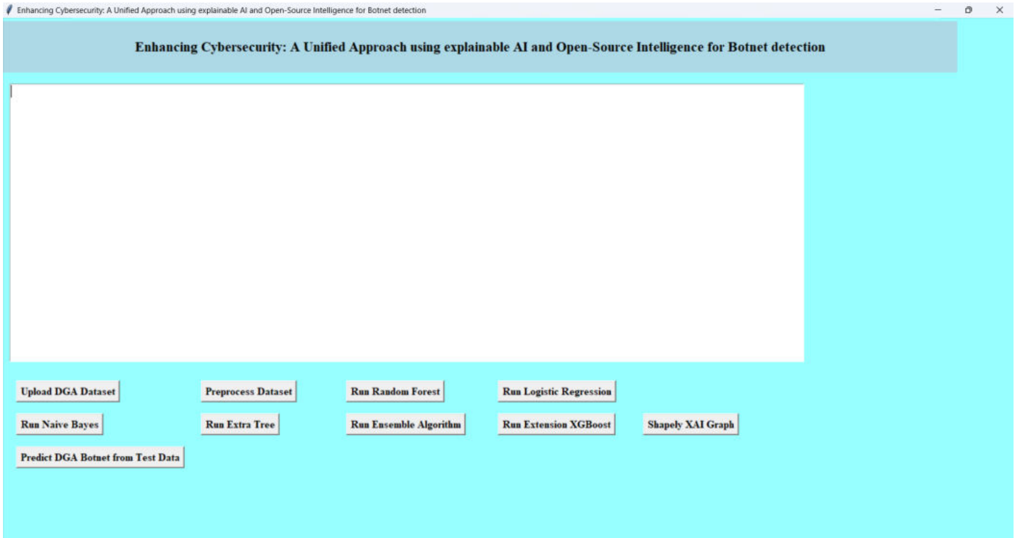
Enhancing Cybersecurity: A Unified Approach using explainable AI and Open-Source Intelligence for Botnet detection
Enhancing Cybersecurity: A Unified Approach using explainable AI and Open-Source Intelligence for Botnet detection
Upload DGA Dataset
Preprocess Dataset
Run Random Forest
Run Logistic Regression
Run Naive Bayes
Run Extra Tree
Run Ensemble Algorithm
Run Extension XGBoost
Shapely XAI Graph
Predict DGA Botnet from Test Data

Figure 3: Main GUI application of enhancing botnets dga detection by incorporating explainable ai and open-source intelligence for cyber threat intelligence sharing.

The figure 4 below represents a graph visualizing SHAP (SHapley Additive explanations) values, which are used for explaining the output of machine learning models. SHAP values help understand the contribution of each feature to the model's prediction.

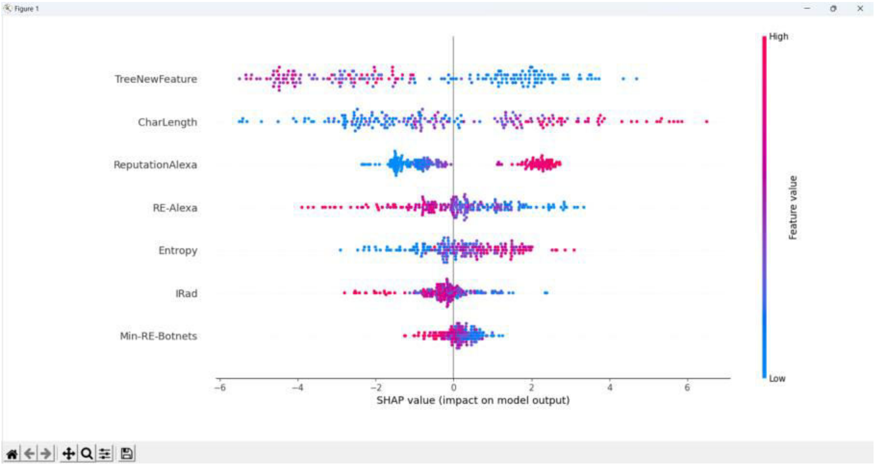

Figure 4: Displays the graph for SHAP value.

The figure 5 shows the confusion matrix and various evaluation metrics for the Random Forest algorithm applied to the dataset. The evaluation metrics has accuracy, precision, recall, and F1 score.

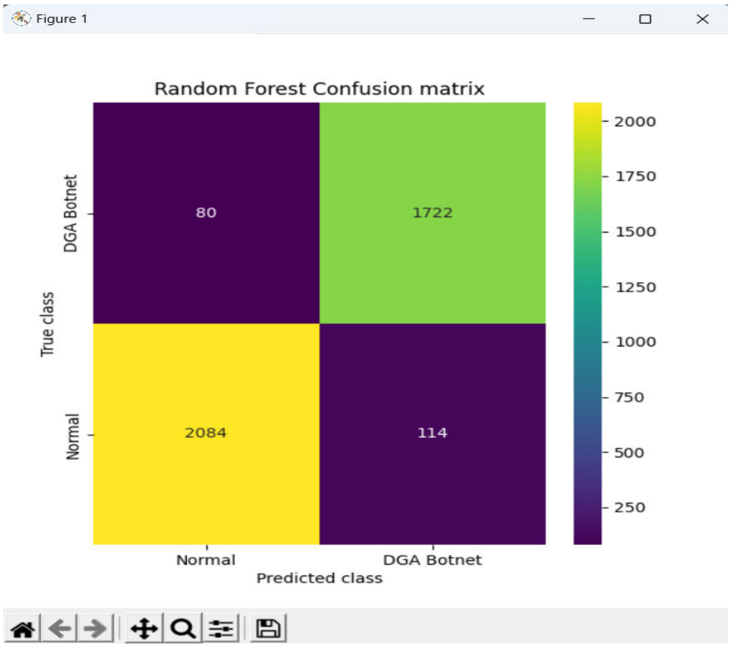

Figure 5: Displays the confusion matrix of random forest algorithms.

Similar to Figure 5, the figure 6 presents the confusion matrix and evaluation metrics, but specifically for the Logistic Regression algorithm.

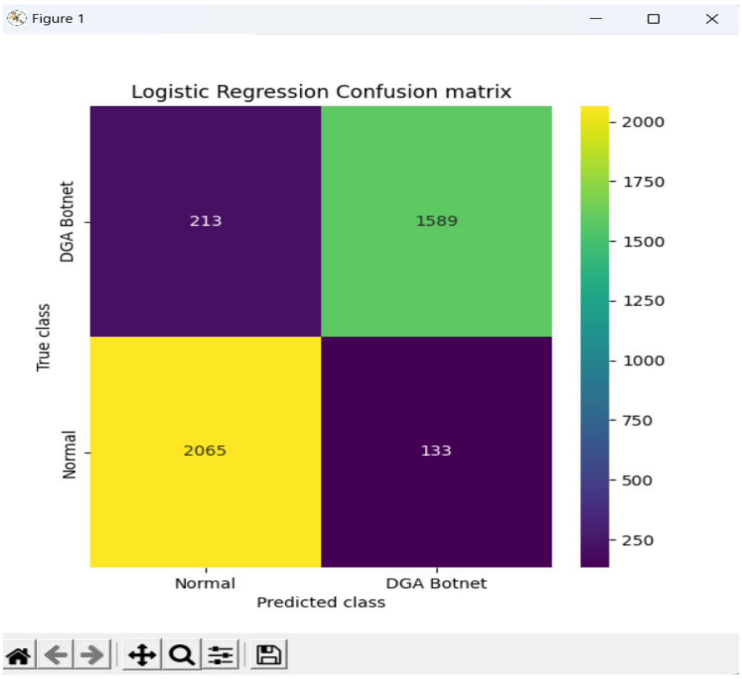

Figure 6: Displays the confusion matrix and evaluation metrics of logistic regression algorithms.

5. CONCLUSION

The research focused on enhancing Domain Generation Algorithm (DGA) detection in botnets by incorporating Explainable AI (XAI) and Open-Source Intelligence (OSINT) represents a significant step forward in bolstering cyber threat intelligence sharing. By leveraging Explainable AI, the project has provided insights into the decision-making process of the detection model, enhancing transparency and interpretability in identifying malicious DGAs. The integration of Open-Source Intelligence has enriched the threat intelligence landscape, enabling the model to benefit from external context and collaborative insights. The developed system has demonstrated efficacy in identifying DGAs associated with botnets, contributing to the early detection and mitigation of potential cyber threats. The interpretability afforded by Explainable AI mechanisms ensures that cybersecurity professionals can understand how the model reaches its conclusions, fostering trust and facilitating informed decision-making in threat response.

REFERENCES

[1] F. Skopik, G. Settanni, and R. Fiedler, "A problem shared is a problem halved: A survey on the dimensions of collective cyber defense through security information sharing," Comput. Secur., vol. 60, pp. 154–176, Jul. 2016, doi: 10.1016/j.cose.2016.04.003.

[2] R. Confalonieri, L. Coba, B. Wagner, and T. R. Besold, "A historical perspective of explainable artificial intelligence," WIREs Data Mining Knowl. Discovery, vol. 11, no. 1, p. e1391, Jan. 2021, doi: 10.1002/widm.1391.

[3] M.-A. Clinciu and H. Hastie, "A survey of explainable AI terminology," in Proc. 1st Workshop Interact. Natural Lang. Technol. Explainable Artif. Intell. (NLXAI), 2019, pp. 8–13, doi: 10.18653/v1/W19-8403.

[4] J. M. Alonso, C. Castiello, and C. Mencar, "A bibliometric analysis of the explainable artificial intelligence research field," in Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations. Cham, Switzerland: Springer, 2018, pp. 3–15, doi: 10.1007/978-3-319-91473-2_1.

[5] M. Singh, M. Singh, and S. Kaur, "Issues and challenges in DNS based botnet detection: A survey," Comput. Secur., vol. 86, pp. 28–52, Sep. 2019, doi: 10.1016/j.cose.2019.05.019.

[6] T. S. Wang, H.-T. Lin, W.-T. Cheng, and C.-Y. Chen, "DBod: Clustering and detecting DGA-based botnets using DNS traffic analysis," Comput. Secur., vol. 64, pp. 1–15, Jan. 2017, doi: 10.1016/j.cose.2016.10.001.

[7] X. D. Hoang and X. H. Vu, "An improved model for detecting DGA botnets using random forest algorithm," Inf. Secur. J., Global Perspective, pp. 1–10, Jun. 2021, doi: 10.1080/19393555.2021.1934198.

[8] B. Yu, J. Pan, J. Hu, A. Nascimento, and M. De Cock, "Character level based detection of DGA domain names," in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2018, pp. 1–8, doi: 10.1109/IJCNN.2018.8489147.

[9] L. Sidi, A. Nadler, and A. Shabtai, "MaskDGA: An evasion attack against DGA classifiers and adversarial defenses," IEEE Access, vol. 8, pp. 161580–161592, 2020, doi: 10.1109/ACCESS.2020.3020964.

[10] J. Peck, C. Nie, R. Sivaguru, C. Grumer, F. Olumofin, B. Yu, A. Nascimento, and M. De Cock, "CharBot: A simple and effective method for evading DGA classifiers," IEEE Access, vol. 7, pp. 91759–91771, 2019, doi: 10.1109/ACCESS.2019.2927075.

[11] H. S. Anderson, J. Woodbridge, and B. Filar, "DeepDGA: Adversarially-tuned domain generation and detection," in Proc. ACM Workshop Artif. Intell. Secur., Vienna, Austria, Oct. 2016, pp. 13–21, doi: 10.1145/2996758.2996767.

[12] T. D. Wagner, K. Mahbub, E. Palomar, and A. E. Abdallah, "Cyber threat intelligence sharing: Survey and research directions," Comput. Secur., vol. 87, Nov. 2019, Art. no. 101589, doi: 10.1016/j.cose.2019.101589.

[13] W. Tounsi and H. Rais, "A survey on technical threat intelligence in the age of sophisticated cyberattacks," Comput. Secur., vol. 72, pp. 212–233, Jan. 2018, doi: 10.1016/j.cose.2017.09.001.

[14] P. Pawlinski, P. Jaroszewski, P. Kijewski, L. Siewierski, P. Jacewicz, P. Zielony, and R. Zuber, ‘‘Actionable information for security incident response,’’ in Proc. Eur. Union Agency Netw. Inf. Secur., Heraklion, Greece, 2014, pp. 1–68. (1 point)