ARTIFICIAL INTELLIGENCE ADOPTED BOTNET ATTACK DETECTION THROUGH EXTREME GRADIANT BOOSTING MODEL

Venkata ramana[1], R. Sravanthi[2], R. Abhignya[2], R. Rushika[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Science Engineering
[1,2]Malla Reddy Engineering College for Women, Maisammaguda, Dhulapally, Kompally, Secunderabad-500100, Telangana, India

ABSTRACT

Botnets, powered by Domain Generation Algorithms, have become a pervasive and sophisticated threat in the cybersecurity landscape. These algorithms enable malicious actors to dynamically generate a large number of domain names, making it challenging for conventional detection systems to keep pace. Existing systems often rely on single models, such as machine learning classifiers, which may struggle with the adaptability required to combat the constantly evolving strategies employed by botnets. While these methods have demonstrated some effectiveness, they often fall short in accurately identifying newly generated domains and may produce false positives or negatives. Machine learning models, for instance, may struggle with overfitting or underfitting due to the dynamic nature of DGA attacks. Rule-based systems, on the other hand, might lack the adaptability needed to handle the diverse patterns of malicious domain generation. These limitations underscore the necessity for a more sophisticated and adaptable solution.The "proposed Synergetic Shield" proposes an ensemble of diverse models, each designed to contribute unique perspectives to the detection process. The ensemble comprises machine learning classifiers, anomaly detection models, and heuristic approaches, working collaboratively to identify Botnet attack. The machine learning models can adapt to evolving attack patterns, the anomaly detection mechanisms can identify deviations from normal behavior, and heuristics can capture subtle nuances indicative of malicious intent. The synergy between these components enhances the overall robustness and accuracy of the detection system.

Keywords: Gradiant Boosting Model, botnet Attack Detection, Artificial Inteligence.

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

The Internet threat landscape is fundamentally changing. A major shift away from hobby hacking toward well-organized cyber crime can be observed. These attacks are typically carried out for commercial reasons in a sophisticated and targeted manner, and specifically in a way to circumvent common security measures. Additionally, networks have grown to a scale and complexity, and have reached a degree of interconnectedness, that their protection can often only be guaranteed and financed as shared efforts. Consequently, new paradigms are required for detecting contemporary attacks and mitigating their effects. Today, many attack detection tasks are performed within individual organizations, and there is little crossorganizational information sharing. However, information sharing is a crucial step to acquiring a thorough understanding of large-scale cyber-attack situations, and is therefore seen as one of the key concepts to protect future networks. A historical perspective of explainable artificial intelligence: Explainability in Artificial Intelligence (AI) has been revived as a topic of active research by the need of conveying safety and trust to users in the “how” and “why” of automated decision-making in different applications such as autonomous driving, medical diagnosis, or banking and finance. While explainability in AI has recently received significant attention, the origins of this line of work go back several decades to when AI systems were mainly developed as (knowledge-based) expert systems. Since then, the definition, understanding, and implementation of explainability have been picked up in several lines of research work, namely, expert systems, machine learning, recommender systems, and in approaches to neural-symbolic learning and reasoning, mostly happening during different periods of AI history. A survey of explainable AI terminology: The field of Explainable Artificial Intelligence attempts to solve the problem of algorithmic opacity. Many terms and notions have been introduced recently to define Explainable AI, however, these terms seem to be used interchangeably, which is leading to confusion in this rapidly expanding field. As a solution to overcome this problem, we present an analysis of the existing research literature and examine how key terms, such as transparency, intelligibility, interpretability, and explainability are referred to and in what context. This paper, thus, moves towards a standard terminology for Explainable AI. Explainable AI, black-box, NLG, Theoretical Issues, Transparency, Intelligibility, Interpretability, Explainability, XAI. A bibliometric analysis of the explainable artificial intelligence research field: This paper presents the results of a bibliometric study of the recent research on explainable Artificial Intelligence (XAI) systems. We took a global look at

the contributions of scholars in XAI as well as in the subfields of AI that are mostly involved in the development of XAI systems. It is worthy to remark that we found out that about one third of contributions in XAI come from the fuzzy logic community. Accordingly, we went in depth with the actual connections of fuzzy logic contributions with AI to promote and improve XAI systems in the broad sense. Finally, we outlined new research directions aimed at strengthening the integration of different fields of AI, including fuzzy logic, toward the common objective of making AI accessible to people. Issues and challenges in DNS based botnet detection: A survey: Cybercrimes are evolving on a regular basis and as such these crimes are becoming a greater threat day by day. Earlier these threats were very general and unorganized. In the last decade, these attacks have become highly sophisticated in nature. This higher level of coordination is possible mainly due to botnets, which are clusters of infected hosts controlled remotely by an attacker (botmaster). The number of infected machines is continuously rising, thereby resulting in botnets with over a million infected machines. This powerful capability gives the botmaster a lethal weapon to launch various security attacks.

3. PROPOSED SYSTEM

The provided source code 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.
- .

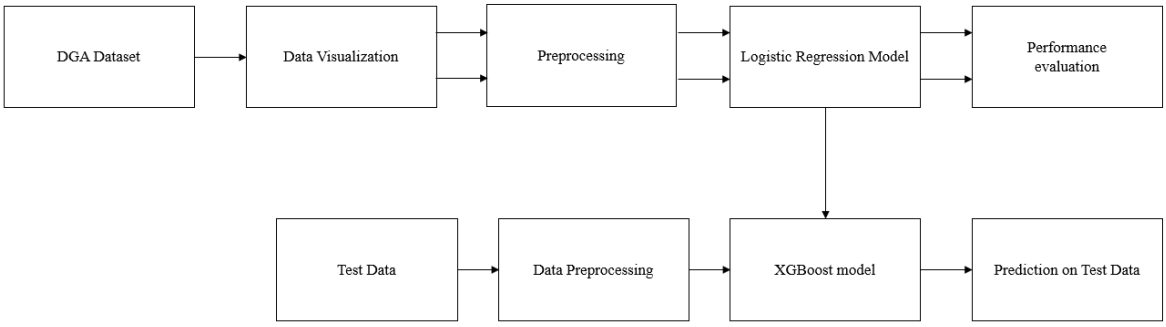

Fig.1: Block Diagram of Proposed system.

Data Preprocessing

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data pre-processing task. A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

Step 1: Data Description: Generates summary statistics for all columns in the dataset. This step provides statistics such as count, mean, standard deviation, minimum, and maximum values for both numeric and categorical columns, giving an overview of the dataset's characteristics.

Step 2: Missing Value Analysis: Visualizes missing values using a matrix plot and displays the total count of missing values for each column. These steps help identify and visualize missing data in the dataset, which is essential for data cleaning and imputation.

Step 3: Data Types: Inspects the data types of columns and converts object-type columns to the category data type. Converting object-type columns to the category data type can reduce memory usage and improve analysis efficiency.

Step 4: Data Preprocessing: Preprocesses the dataset by extracting and transforming date components, removing '$' from ticker values, and converting data types. These steps prepare the data for analysis by making it more suitable for visualization and modeling.

Step 5: Feature Engineering: Removes the 'date' column from the dataset. Feature engineering involves selecting, transforming, or removing features to prepare the dataset for modeling.

Step 6: Final Data Check: Checks for missing values and inspects unique values in specific columns. These final checks ensure that the data is in a suitable state for further analysis and modeling.

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.

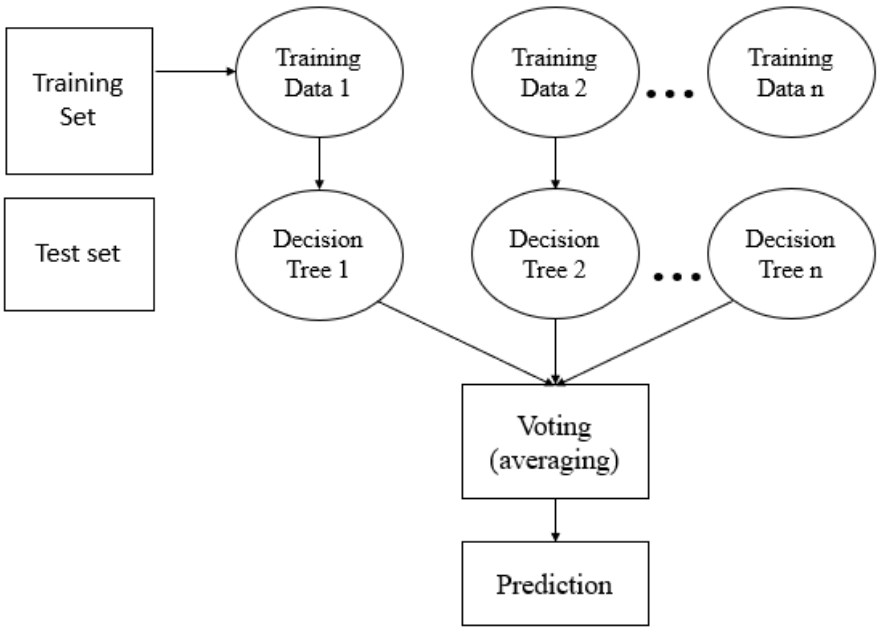

Fig. 2: XGBoost algorithm.

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

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.

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.

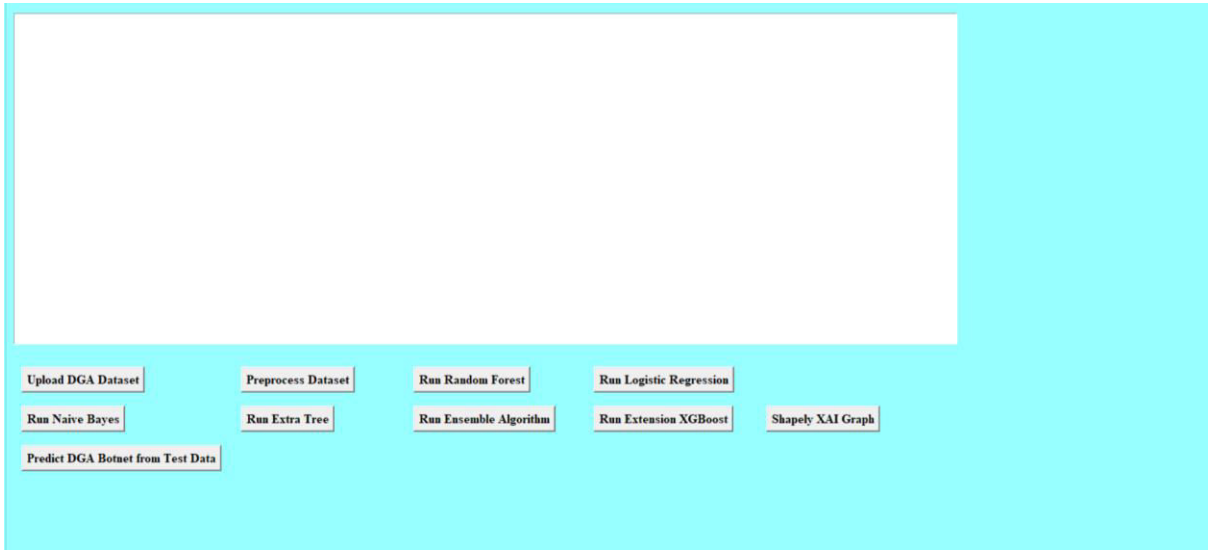

Figure. 3: Main GUI application of enhancing botnets dga detection by incorporating explainable ai and open-source intelligence for cyber threat intelligence sharing.

The figure 4 shows the user interface or a section of the application where the user can choose or load the DGA (Domain Generation Algorithm) dataset for analysis.

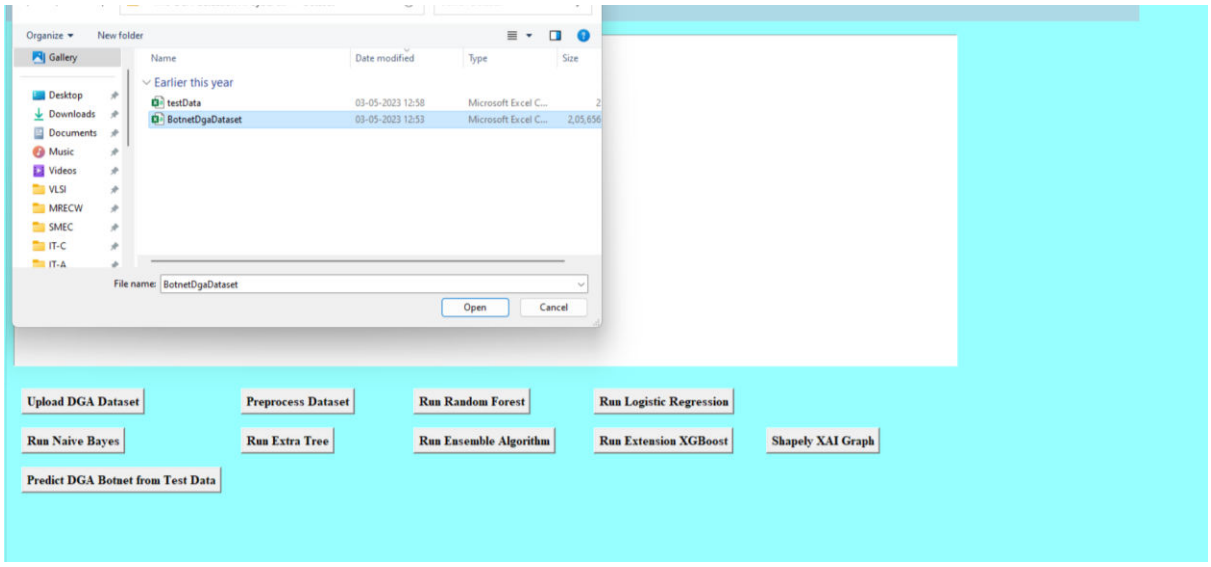

Figure. 4: Displays the selection of DGADataset.

The figure 5 presents the confusion matrix and evaluation metrics for the XG Boost algorithm, showcasing its performance on the dataset.

Figure. 5: Displays the confusion matrix and evaluation metrics of XG Boost algorithms.

The figure 6 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.

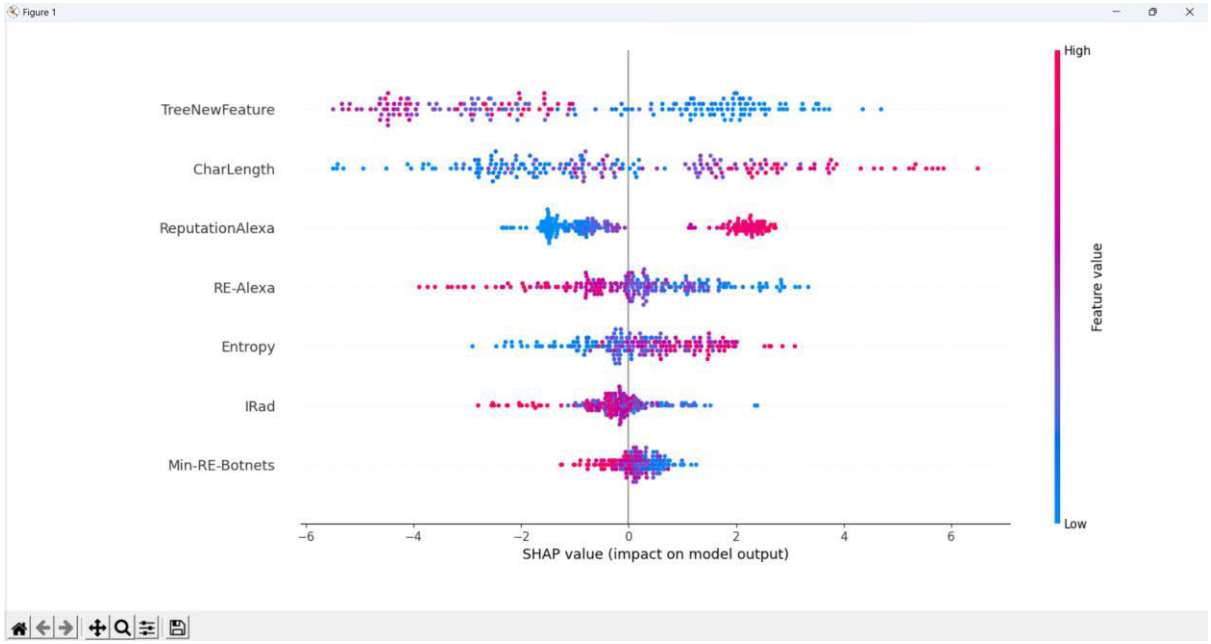

Figure 11: Displays the graph for SHAP value.

The figure 6 illustrates the predictions generated by the chosen machine learning models on the test dataset, providing insights into how well the models generalize to new, unseen data.

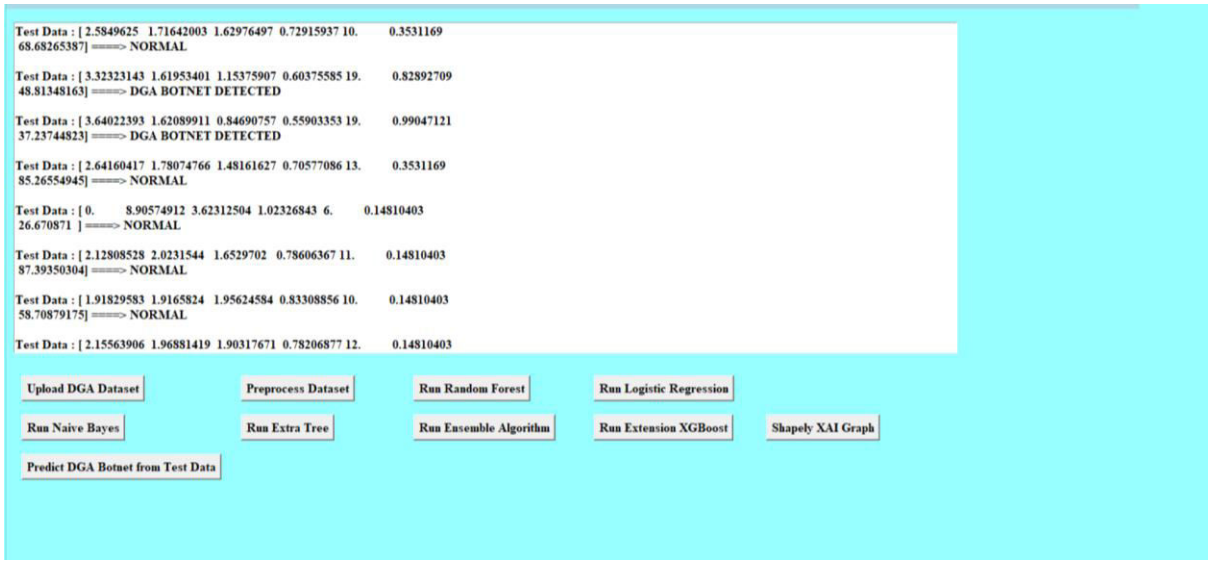

Figure. 6: Displays the prediction of the test data.

5. CONCLUSION AND FUTURE SCOPE

The project 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: Adversariallytuned 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.