A COMPREHENSIVE IDS TO DETECT BOTNET ATTACKS USING MACHINE LEARNING TECHNIQUES

Abdullah Alghamdi, Ayad Barsoum
St. Mary's University, San Antonio, TX (USA)
Emails: aalghamdi6@mail.stmarytx.edu; abarsoum@stmarytx.edu

Abstract — **In the contemporary landscape of cyber threats, Botnet attacks emerge as a pervasive and evolving menace, demanding sophisticated countermeasures. This paper presents a comprehensive development of an Intrusion Detection System (IDS) utilizing advanced machine learning techniques to thwart Botnet intrusions. Central to this IDS is an ensemble voting classifier, a synergistic integration of multiple algorithms, tailored to augment detection efficacy and adaptability. The paper delineates the systematic progression of our work, encompassing meticulous data preprocessing, strategic feature selection, rigorous model training, and the deployment of an intuitive web application. Evaluative measures are employed on real-time network traffic datasets, affirming the model's proficiency in discerning Botnet activities with notable accuracy and reliability. Our work introduces an approach to Botnet detection leveraging machine learning which increases the detection accuracy underscoring the efficacy of the proposed approach.**

Keywords — Intrusion Detection System, Botnet Attacks, Machine Learning, Ensemble Voting Classifier, Data Preprocessing, Feature Selection.

I. INTRODUCTION

In the ever-evolving domain of cybersecurity, the threat posed by Botnet attacks has become increasingly prominent, posing significant challenges to both individual privacy and organizational security. Botnets, networks of infected computers controlled by malicious actors, are utilized to carry out a variety of cyber-attacks, including but not limited to, Distributed Denial of Service (DDoS) attacks, spamming, and data theft. The complexity and stealthiest of these Botnet attacks necessitate advanced and dynamic defense mechanisms.

Traditional Intrusion Detection Systems (IDS) [24], while foundational in cybersecurity defenses, often struggle to keep pace with the sophistication of modern Botnet threats. These conventional systems typically rely on signature-based detection methods, which are less effective against new or evolving attack vectors. Consequently, there is a pressing need for an IDS that not only addresses these limitations but also adapts to the changing landscape of cyber threats.

In this paper we propose a comprehensive IDS developed to combat Botnet attacks through the application of advanced machine learning techniques. At the heart of this IDS is an ensemble voting classifier that synergizes multiple machine learning algorithms to enhance detection accuracy and adaptability. This approach not only improves the system's ability to detect known Botnet patterns but also equips it to identify new and evolving threats.

The development process of this IDS encompasses comprehensive data preprocessing, strategic feature selection, and rigorous model training. Additionally, the system is integrated into a user-friendly web application, making it accessible for both cybersecurity experts and non-experts.

In summary, this research contributes to the field of cyber defense by presenting an effective and adaptable solution to detect and counter Botnet attacks [1, 2, 3, 4, 5]. By leveraging machine learning techniques, the proposed IDS represents a step forward in the development of countermeasures, offering both theoretical and practical implications in the ongoing battle against cyber threats.

The remainder of the paper is organized as follows. Section II presents our proposed machine learning-based IDS. Section III demonstrates the used datasets. Evaluation metrics for intrusion detection systems are discussed in Section IV. Section V presents the performance analysis for individual classifiers. The enhanced performance of our approach is discussed in Section VI. Concluding remarks are given in Section VII.

II. PROPOSED MACHINE LEARNING -BASED IDS FOR BOTNET DETECTION

In the development of an IDS for Botnet attack detection, this paper utilizes the robust capabilities of Machine Learning (ML) to establish a sophisticated defense mechanism. At the forefront of the proposed approach is an ensemble voting classifier, integrating Gradient Boosting, Decision Tree, and Random Forest classifiers. This ensemble approach enhances the system's accuracy and robustness by synergizing the unique strengths of each classifier. The training and evaluation of these models are conducted with rigor, involving the segmentation of the dataset into subsets for detailed training and evaluation. This approach ensures that the models are not only effective in the current context but are also capable of generalizing to new and unseen data, as shown by performance metrics such as accuracy, precision, recall, and F1-score. Another feature of this system is its capacity for near real-time analysis of network traffic, a feature essential for the prompt detection of Botnet attacks. This real-time functionality, coupled with the continuous learning and adaptability of the ML models, guarantees the sustained efficacy of the IDS. Complementing the technical aspects of this system is an interactive web application, designed to simply present complex data for users, thereby facilitating informed decision-making in network security. This application, through its intuitive interface and utilization of visual aids like charts and graphs, significantly enhances user engagement and interaction with the IDS [6,7,8,9].

Our proposed IDS developed in this study employs an ensemble voting classifier model, integrating three distinct machine learning algorithms: the Gradient Boosting Classifier, the Decision Tree Classifier, and the Random Forest Classifier. The rationale for selecting these specific algorithms and their operational mechanisms are as follows:

1. Gradient Boosting Classifier [13, 16]:
   - Reason for Selection:

The Gradient Boosting Classifier (Figure 1) is chosen for its proficiency in forming a strong predictive model by combining multiple weak learners. This ability to sequentially improve and correct the predictions of these weak learners makes it highly effective in scenarios where accuracy is paramount.

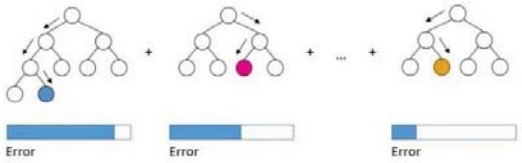

Figure 1. Gradient Boosting Classifier

- Mechanism:

It operates through an iterative process, where each subsequent model attempts to correct the errors of the previous one. This results in a cumulative improvement in prediction accuracy over iterations, making it well-suited for complex and evolving patterns in network traffic, like those in Botnet attacks.

2. Decision Tree Classifier [14]:
   - Reason for Selection:

The Decision Tree Classifier is included for its straightforward, interpretable structure, which aids in understanding the decision-making process. This transparency is crucial for validating the model's decisions and ensuring trustworthiness in a security context.

- Mechanism:

It utilizes a tree-like model of decisions, where each node represents a feature, and each branch represents a decision rule. This hierarchical structure enables the classifier to make step-by-step decisions, effectively handling diverse data with varying feature importance.

3. Random Forest Classifier: [15].
   - Reason for Selection:

The Random Forest Classifier (Figure 2) was selected due to its inherent capability to reduce overfitting, a common challenge in machine learning models. By being an ensemble of decision trees itself, it introduces randomness and diversity in the model, enhancing its generalization capabilities.

- Mechanism:

This classifier operates by constructing multiple decision trees during training and outputting the mode of their predictions for classification. The randomness in selecting features and training instances for each tree ensures a broad and generalized learning, reducing the risk of overfitting and improving overall predictive accuracy.

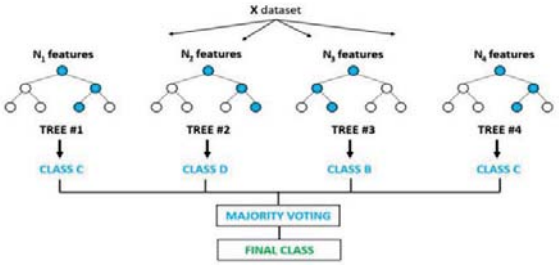

Figure 2. Random Forest Classifier

4. Ensemble Voting Classifier [14]:
   - Reason for Selection:

The ensemble model effectively combines these three algorithms, leveraging their individual strengths. This integration results in a more robust system capable of accurately detecting a wide range of Botnet activities.

- Mechanism:

The mechanism involves training each classifier on the dataset, and then using a voting system (Figure 3) where each classifier contributes to the final decision. This approach ensures that the strengths of one algorithm compensate for the weaknesses of others, leading to a balanced and effective detection system.

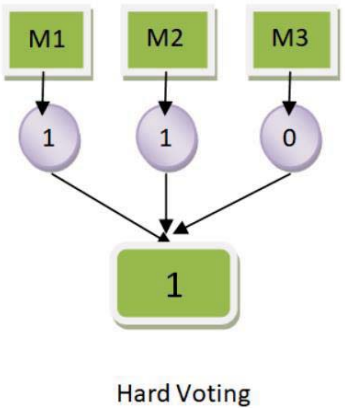

Figure 3. Ensemble Voting Classifier

III. Datasets Utilization and Preprocessing

In the development of our IDS to counter botnet attacks, a meticulous approach to dataset selection and processing was employed, which is critical in training and evaluating the machine learning models underpinning the IDS.

Our work utilized both publicly available and proprietary datasets. Among the public datasets were the NSL-KDD, CIC-IDS2018, and BoT-IoT [17, 18, 19], each offering a diverse range of network traffic scenarios, including both normal activities and botnet patterns.

The data preprocessing phase was pivotal in refining these datasets for optimal use in machine learning. This phase included several key stages:

- Data Cleaning:

Initial steps involved removing irrelevant features and addressing missing values. Features not directly impacting network security, like timestamps and payload data, were excluded to sharpen the focus on relevant attributes.

- Normalization and Scaling:

To account for the varied scales of data features, we employed Min-Max scaling, normalizing the data to a uniform scale [23]. This normalization facilitated faster

and more accurate convergence of machine learning algorithms.

- Feature Selection:

We applied techniques such as Mutual Information (MI) to identify and retain the most significant features for detecting botnet attacks [21]. This step not only boosted model performance but also reduced computational demands.

- Data Augmentation:

Particularly for the proprietary dataset, we tackled class imbalance through data augmentation, employing techniques like SMOTE to oversample minority classes [22]. This ensured a balanced representation of both normal and attack classes in the training data.

- Label Encoding:

For datasets with categorical representation of attack types, label encoding was used to transform these categories into numerical formats, making them more amenable to processing by machine learning algorithms.

- Train-Test Split:

Finally, the training dataset was divided into training and evaluation subsets in a 80:20 ratio, and the other dataset was all used for testing, allowing comprehensive model training while retaining a significant portion for unbiased evaluation

This methodical approach to dataset selection and processing was foundational in ensuring the proposed IDS's robustness, precision, and effectiveness in detecting botnet attacks. [10, 11, 12]. Figure 4 shows the general flow of the proposed IDS.

IV. Evaluation Metrics for Intrusion Detection Systems

In assessing the efficacy of Machine Learning-based IDS for Botnet attack detection, several key performance metrics are utilized. These metrics, namely Accuracy, Precision, Recall, and the F1 Score, provide a comprehensive evaluation of the system's effectiveness in identifying and differentiating between normal and malicious network activities. Each metric offers unique insights into the IDS's performance, highlighting its strengths and areas for improvement [20].

- Accuracy in IDS Evaluation

The Accuracy metric is defined as:

$$\text{Accuracy} = \frac{(\text{True Positive} + \text{True Negative})}{(\text{Total Sample})} \quad (1)$$

This metric quantifies the overall effectiveness of the IDS by measuring the proportion of correctly identified instances, both as Botnet attacks (True Positives) and normal behavior (True Negatives), out of the total instances evaluated. High accuracy indicates a system that is generally reliable across various scenarios.

- Precision: A Measure of Exactness

Precision is expressed as:

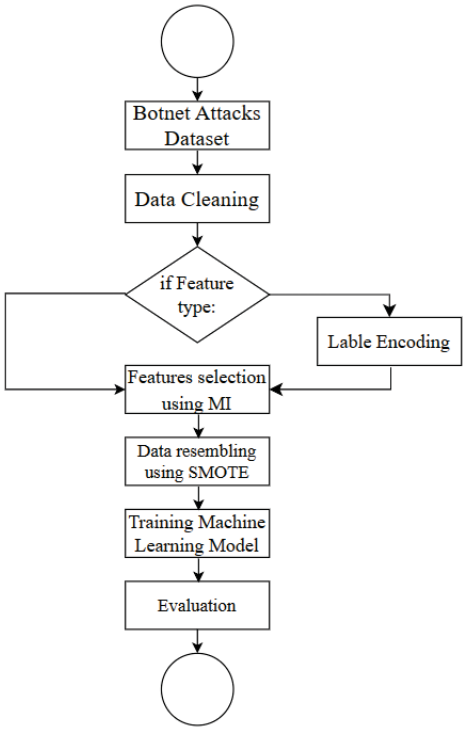

Figure 4. General Flow of IDS

$$\text{Precision} = \frac{(\text{True Positive})}{(\text{True Positive} + \text{False Positive})} \quad (2)$$

Precision assesses the system's exactness, focusing on its capability to minimize false positives – instances where normal behavior is incorrectly classified as a Botnet attack. A high precision score is indicative of a system that accurately flags attacks with minimal false alarms.

- Recall: Sensitivity of the IDS

The Recall metric is formulated as:

$$\text{Recall} = \frac{(\text{True Positive})}{(\text{True Positive} + \text{False Negative})} \quad (3)$$

Recall, or Sensitivity, evaluates the IDS's ability to correctly identify all actual Botnet attacks. This metric is crucial in determining the system's proficiency in detecting threats, with a higher recall indicating fewer missed attacks.

- F1 Score: Balancing Precision and Recall

The F1 Score is calculated using the formula:

$$\text{F1 score} = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (4)$$

As the harmonic means of Precision and Recall, the F1 Score provides a balanced measure of the system's accuracy and reliability. This metric is particularly useful in scenarios where an equilibrium between false positives and false negatives is essential. A high F1 Score denotes a

well-balanced system in terms of both detecting attacks and minimizing false alerts.

In conclusion, these metrics collectively offer a multifaceted view of an IDS's performance, essential for ensuring the reliability and effectiveness of systems designed to safeguard against Botnet attacks. Through rigorous evaluation using these metrics, IDS developers can fine-tune their systems, achieving optimal performance in real-world cybersecurity scenarios.

V. Performance Analysis of Individual Classifiers in an Ensemble Model for Botnet Intrusion Detection

In the realm of developing Botnet IDS, the effectiveness of machine learning classifiers is paramount. This study details the performance of individual classifiers within an ensemble model, focusing on their accuracy, recall, and F-score across training and testing datasets.

1. Gradient Boosting Classifier: Figures 5 and 6 show the training and testing results of the Gradient Boosting classifier, respectively.
   - Training Performance: Exhibited a notable accuracy of 79%, demonstrating its proficiency in learning complex data relationships and identifying Botnet patterns.
   - Testing Performance: Maintained an accuracy of 81%, indicating its capability to generalize and adapt to new, unseen data.
   - Evaluation Metrics: Emphasized high recall and precision, with a focus on minimizing false negatives and positives.
2. Decision Tree Classifier: Figures 7 and 8 show the training and testing results of the decision tree classifier, respectively.
   - Training Performance: Attained an accuracy of 79%, demonstrating the model's proficiency in understanding basic data correlations.
   - Testing Performance: Equaled the training accuracy with a rate of 79%, indicating the model's capability in generalizing learned patterns to unfamiliar datasets.
   - Strengths and Limitations: Shows efficiency in handling uncomplicated cases, however, it tends to reduce more intricate relationships to simpler forms.
3. Random Forest Classifier: Figures 9 and 10 show the training and testing results of the random forest classifier, respectively.
   - Training Performance: Demonstrated a high accuracy of 92%, benefiting from its structure of multiple decision trees to avoid overfitting.
   - Testing Performance: Recorded almost 90% accuracy, underlining its adaptability and versatility in handling diverse patterns.

Figure 5. Gradient Boosting Training Results

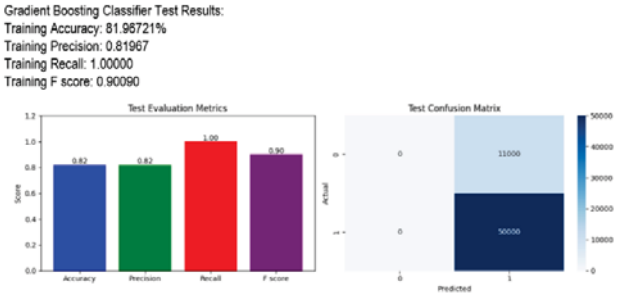

Figure 6. Gradient Boosting Testing Results

Figure 7. Decision Tree Training Results

Figure 8. Decision Tree Testing Results

- Ensemble Voting Classifier: Figures 11 and 12 show the training and testing results of the proposed ensemble classifier, respectively.

Figure 9. Random Forest Training Results

Figure 11. Ensemble Voting Training Results

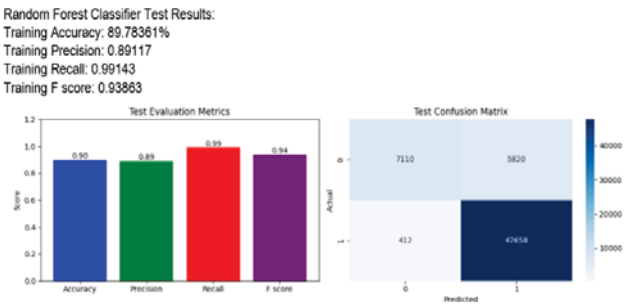

Figure 10. Random Forest Testing Results

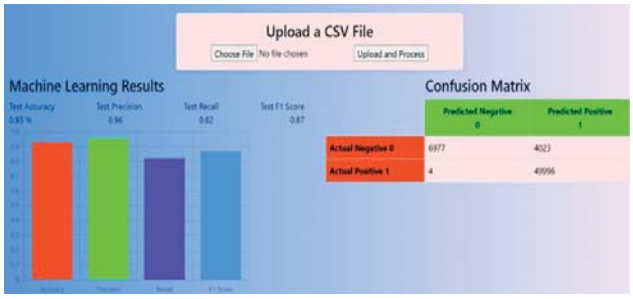

Figure 12. Ensemble Voting Testing Results

- Training Results: Remarkably achieved an 88% recall, a 95% accuracy, and a 92% F-score, illustrating the ensemble's balanced approach to Botnet attack detection.
- Comprehensive Evaluation: The ensemble model effectively combines the strengths of individual classifiers, resulting in enhanced overall performance.

The comprehensive evaluation of these classifiers underlines the importance of selecting appropriate machine learning models for IDS. The Gradient Boosting Classifier excels in generalization, the Decision Tree Classifier in interpretability and application to new data, and the Random Forest Classifier in versatility and resistance to overfitting. The Ensemble Voting Classifier, integrating these individual strengths, emerges as a robust solution for Botnet intrusion detection, balancing recall, accuracy, and F-score to effectively combat cybersecurity threats.

This analysis provides critical insights into the efficacy of machine learning classifiers in cyber defense, specifically in developing a reliable IDS against Botnet attacks. The findings underscore the potential of ensemble models in enhancing detection capabilities and adapting to the evolving landscape of cyber threats.

VI. Enhanced Performance through Integrated Machine Learning Models in Botnet Detection

The completion of our Botnet detection system was followed by a rigorous phase of testing, designed to evaluate its efficacy and real-world performance. This testing phase was critical in assessing the system's ability to identify Botnet attacks accurately and promptly, using several key performance metrics.

This evaluation focused on the system's detection capabilities, response time, and its precision in distinguishing true Botnet attacks from normal network behaviors.

The integration of three distinct machine learning models – the Gradient Boosting Classifier, the Decision Tree Classifier, and the Random Forest Classifier – along with the Ensemble Voting Classifier, led to a significant enhancement in performance.

The synergy created by combining these classifiers resulted in an overall improvement in the system's detection accuracy.

Figure 12 presents the testing results, which demonstrate the performance improvements achieved.

Notably, there was a 3% increase in detection accuracy, attributable to the combined application of the machine learning models and the ensemble technique.

The improvement in detection accuracy marks not just a technical advancement, but also a significant

enhancement in the system's practical application for robust cyber defense.

In real-world scenarios, even a small increase in detection rates can profoundly impact the overall effectiveness of cybersecurity strategies and threat mitigation.

This phase of testing the Botnet detection system underscores the benefits of integrating a variety of machine learning models and an ensemble approach. The observed 3% increase in detection accuracy is a testament to the effectiveness of this integrated machine learning strategy in confronting complex and evolving cyber threats.

VII. Conclusion

The implementation of our Botnet detection system marks a pivotal stride in bolstering network security against evolving cyber threats. By integrating advanced machine learning algorithms, including ensemble voting classifiers with Gradient Boosting, Decision Trees, and Random Forest, the system has demonstrated remarkable precision in identifying Botnet attacks. A significant achievement of this development is the 3% increase in detection accuracy, underscoring the efficacy of our approach.

The system's adaptability, coupled with user-centric design, ensures it remains agile and effective in dynamic security landscapes. This advancement not only enhances the accuracy of threat detection but also reinforces our commitment to evolving cybersecurity solutions in response to emerging challenges. The success of this study offers a robust and intelligent defense mechanism against complex cyber threats.

References

[1] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, "Machine learning-based IoT-botnet attack detection with sequential architecture," Sensors, vol. 20, no. 16, pp. 4372, 2020.

[2] S. Pokhrel, R. Abbas, and B. Aryal, "IoT security: botnet detection in IoT using machine learning," arXiv preprint arXiv:2104.02231, 2021.

[3] S. Haq and Y. Singh, "Botnet detection using machine learning," in 2018 5th International Conference on Parallel, Distributed and Grid Computing, IEEE, Dec. 2018.

[4] M. Alauthman, N. Aslam, M. Al-Kasassbeh, S. Khan, A. Al-Qerem, and K. K. R. Choo, "An efficient reinforcement learning-based Botnet detection approach," Journal of Network and Computer Applications, vol. 150, 2020.

[5] A. Juyal, B. Bhushan, A. A. Hameed, and A. Jamil, "Deep Learning Approaches for Cyber Threat Detection and Mitigation," in Proceedings of the 2023 7th International Conference on Advances in Artificial Intelligence, 2023.

[6] A. Alharbi and K. Alsubhi, "Botnet detection approach using graph-based machine learning," IEEE Access, vol. 9, pp. 99166-99180, 2021.

[7] D. Nookala Venu, A. Kumar, and M. A. S. Rao, "Botnet Attacks Detection in Internet of Things Using Machine Learning," NeuroQuantology, vol. 20, pp. 743-754, 2022.

[8] M. M. Alani, "BotStop: Packet-based efficient and explainable IoT botnet detection using machine learning," Computer Communications, vol. 193, pp. 53-62, 2022.

[9] W. Fei, H. Ohno, and S. Sampalli, "A Systematic Review of IoT Security: Research Potential, Challenges and Future Directions," ACM Computing Surveys, 2023.

[10] S. Y. Yerima and A. Bashar, "A novel Android botnet detection system using image-based and manifest file features," Electronics, vol. 11, no. 3, pp. 486, 2022.

[11] A. Alhowaide, I. Alsmadi, and J. Tang, "Towards the design of real-time autonomous IoT NIDS," Cluster Computing, pp. 1-14, 2021.

[12] B. Kaur, S. Dadkhah, F. Shoeleh, E. C. P. Neto, P. Xiong, S. Iqbal, et al., "Internet of things (IoT) security dataset evolution: Challenges and future directions," Internet of Things, p. 100780, 2023.

[13] M. Alqahtani, H. Mathkour, and M. M. Ismail, "IoT botnet attack detection based on optimized extreme gradient boosting and feature selection," Sensors, vol. 20, 2020.

[14] T. G. Dietterich, "Ensemble methods in machine learning," in International workshop on multiple classifier systems, pp. 1-15, Springer Berlin Heidelberg, Jun. 2000.

[15] D. David, "Random Forest classifier tutorial: How to use tree-based algorithms for machine learning," freeCodeCamp.org, Aug. 13, 2020. [Online]. Available: https://www.freecodecamp.org/news/how-to-use-the-tree-based-algorithm-for-machine-learning/

[16] V. Aliyev, "A hands-on explanation of gradient boosting regression," Medium, Sep. 4, 2020. [Online]. Available: https://vagifaliyev.medium.com/a-hands-on-explanation-of-gradient-boosting-regression-4cfe7cfdf9e

[17] University of New Brunswick, "University of New Brunswick est.1785," n.d. [Online]. Available: https://www.unb.ca/cic/datasets/index.html

[18] University of New Brunswick, "University of New Brunswick est.1785," n.d. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.html

[19] "The bot-IOT dataset," UNSW Research, n.d. [Online]. Available: https://research.unsw.edu.au/projects/bot-iot-dataset

[20] R. A. Disha and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique," Cybersecurity, vol. 5, no. 1.

[21] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," IEEE Transactions on Neural Networks, vol. 20, no. 2, 2009.

[22] A. Gosain and S. Sardana, "Handling class imbalance problem using oversampling techniques: A review," in 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 79-85, IEEE, Sep. 2017.

[23] S. G. O. P. A. L. Patro and K. K. Sahu, "Normalization: A preprocessing stage," arXiv preprint arXiv:1503.06462, 2015.

[24] S. Axelsson, "Intrusion Detection Systems: A Survey and Taxonomy," Technical Report 99-15, Department of Computer Engineering, Chalmers University, 2000.