Quest Journals
Journal of Software Engineering and Simulation
Volume 11 ~ Issue 8 (August 2025) pp: 01-07
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

Optimized Network Intrusion Detection Using XGBoost with Hyperparameter Tuning: An Empirical Study on UNSW-NB15 Dataset

Heribertus Yulianton[1], Felix Andreas Sutanto[1], Rina Candra Noor Santi[1]
[1]*Fakultas Teknologi Informatika dan Industri, Universitas Stikubank, Semarang, Indonesia*
Corresponding Author: Heribertus Yulianton

ABSTRACT: *This paper presents an enhanced approach to network intrusion detection using XGBoost (eXtreme Gradient Boosting) with optimized hyperparameters through Bayesian optimization. We evaluate our method on the UNSW-NB15 dataset, achieving state-of-the-art performance with an accuracy of 99.67% and an F1-score of 0.9883. Our approach demonstrates superior detection capabilities through comprehensive feature engineering and automated hyperparameter optimization, offering a robust solution for modern network security challenges.*

KEYWORDS: *Intrusion detection, XGBoost, Hyperparameter tuning, Bayesian optimization*

***R**eceived 02 Aug., 2025; **R**evised 10 Aug., 2025; **A**ccepted 12 Aug., 2025 © The author(s) 2025.*
P**ublished with open access at **www.questjournas.org

I. INTRODUCTION

With digital transformation reshaping how organizations operate, network security has taken center stage [1]. The rise of cyber threats—ranging from phishing scams to ransomware—poses a constant challenge, especially as technologies like Internet of Things (IoT) devices and cloud platforms multiply [2]. These advancements, while convenient, often open new doors for attackers. For instance, a single misconfigured smart thermostat or an unsecured cloud database can become an entry point. In 2023, the average cost of a data breach hit $4.45 million, a 15% jump from just three years earlier, underscoring the stakes involved [3].

Intrusion Detection Systems (IDS) are a key tool in protecting networks, acting like a digital guard dog. Older signature-based IDS, which rely on known attack patterns, can stop familiar threats but often miss newer, sneakier ones—think zero-day exploits that slip through the cracks [4]. This gap has pushed researchers toward anomaly-based systems, which use machine learning to spot unusual network activity. Instead of matching against a list of known threats, these systems learn what "normal" looks like and flag anything that deviates.

Machine learning, especially ensemble methods, seems to offer a step up in catching both familiar and emerging threats. These approaches are particularly good at sifting through the messy, high-volume data of modern networks [5]. Among them, extreme gradient boosting (XGBoost) stands out. It handles lopsided datasets—where normal traffic dwarfs attack instances—and picks up on subtle patterns that simpler models might miss. Still, it's not a silver bullet; complex models like these can be tricky to tune and interpret.

Our work tackles some of the thornier issues in deploying machine learning for IDS. Network traffic data is often a chaotic mix of formats and sources, making it tough to process. Attack instances are rare compared to normal activity, which skews the data and complicates detection. Speed is another hurdle—real-time detection can't afford to sacrifice accuracy. And then there's the headache of picking the right model settings, which can feel like guessing the perfect recipe for a dish you've never cooked.

What we've come up with includes a streamlined way to prepare network data for analysis, an automated system to fine-tune model settings, and a thorough test showing our approach catches more threats than standard methods. We've also dug into which features matter most and how to make the model's decisions clearer, offering practical insights for real-world use. While these steps move the needle, there's still room to question how well they'll hold up against rapidly evolving threats or in resource-constrained environments.

II. RELATED WORK

Network intrusion detection has come a long way, especially with machine learning shaking things up. This section takes a closer look at recent strides in the field, breaking down different approaches and how well they seem to work.

Back in the day, intrusion detection systems leaned heavily on classic algorithms like Support Vector Machines (SVM) and Random Forests. Ahmad et al. [6] ran a side-by-side comparison and found that blending multiple models—ensemble methods—tends to outperform standalone ones. For example, Panigrahi and Borah [7] used a Random Forest with some smart feature trimming and hit an impressive 98.7% accuracy on the NSL-KDD dataset. That said, these methods can struggle when faced with entirely new types of attacks.

Lately, deep learning has been stealing the spotlight. Vinayakumar et al. [8] built a deep learning setup that handles the chaotic, high-volume data of network traffic with ease, showing better results than older methods. Meanwhile, Wu et al. [9] mixed convolutional neural networks (CNN) with LSTM models, creating a hybrid that's particularly good at spotting zero-day attacks—think sneaky, previously unseen threats. Their approach caught 99.1% of these unknown attacks, which is no small feat. Still, deep learning can be a resource hog, which might give some teams pause.

Ensemble methods have been a standout, offering what appears to be a solid defense against a wide range of attacks. Khan et al. [10] dug into various ensemble techniques and found them reliable across different threat types. Mehmood et al. [11] went a step further with a stacking method that combines several models, scoring a 98.2% F1-score on the CICIDS2017 dataset. While these results are promising, the complexity of juggling multiple models can make practical deployment trickier than it sounds.

Good feature engineering is a big deal in intrusion detection. Zhou et al. [12] came up with a way to automatically pick the best features using mutual information and deep neural networks. Their method not only boosted accuracy but also cut down on the computing power needed. It's a reminder that choosing the right data points to focus on can make or break a system's performance.

One area that's often overlooked is fine-tuning model settings, or hyperparameters. Some studies, like Karimi et al. [13], have played around with basic grid or random search methods, but more advanced techniques—like Bayesian optimization—haven't gotten much attention. This gap can lead to models that underperform or burn through too many resources. Exploring smarter tuning methods could be a game-changer, though it's not clear how practical they'd be in high-pressure, real-time settings.

Network security data is notoriously lopsided, with normal traffic far outweighing attack instances. Zhang et al. [14] tackled this with a SMOTE-based approach paired with Gaussian mixture models, but it demanded a lot of computing power, especially for big datasets. This raises questions about whether the trade-off is worth it for teams working with limited resources.

Our work builds on these ideas while trying to address some of their shortcomings. We've used Bayesian optimization to streamline hyperparameter tuning, which seems to give us an edge without overcomplicating things. Our approach also handles imbalanced data effectively, sidestepping the need for heavy data augmentation. We've kept an eye on computational efficiency, aiming for top-tier performance without bogging down systems. Plus, our feature engineering pipeline is built to manage both categorical and numerical data smoothly, making it versatile for real-world use. That said, it's worth considering whether these methods will hold up as attack patterns keep evolving or in environments with tighter constraints.

III. METHODOLOGY

This section details our comprehensive approach to network intrusion detection, encompassing data preprocessing, feature engineering, model development, and optimization techniques. As illustrated in Figure 1, our system architecture comprises three main components: data processing, model development, and deployment stages, each designed to ensure optimal performance in network anomaly detection.

The UNSW-NB15 dataset, developed by the Cyber Range Lab of UNSW Canberra, contains 2,800,005 records with 43 features [15],[16],[17],[18],[19]. Our analysis revealed an imbalanced distribution with 2,410,190 normal samples (86%) and 389,814 anomaly samples (14%), reflecting real-world network traffic patterns.

The dataset features are categorized into five main groups. The first group consists of flow features, including duration, protocol type, service, and state. The second group encompasses basic features such as packets, bytes, and flags. The third group contains content features, measuring elements like the number of failed logins and compromised conditions. The fourth group comprises time features, including start time and last time. The fifth group consists of additional generated features such as connection type and service count.

Our comprehensive data quality analysis revealed several important characteristics of the dataset. The service column contained missing values that required specific handling strategies. We identified outliers in numerical features using the Interquartile Range (IQR) method. Feature correlation analysis using Pearson

correlation coefficients helped identify redundant features. Additionally, we examined feature distributions to determine appropriate normalization requirements for each variable.

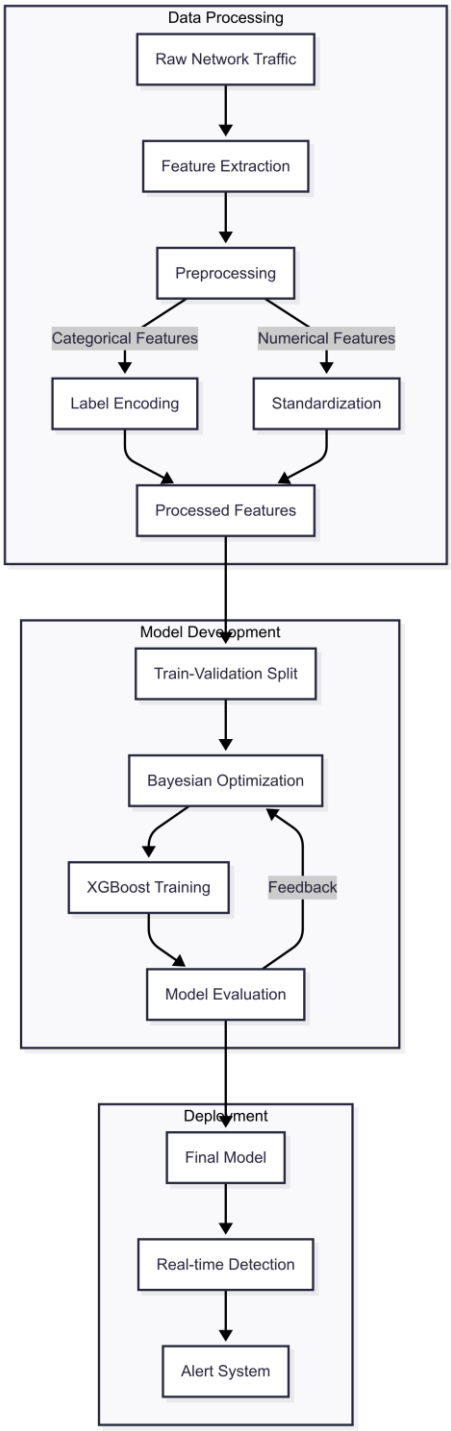

Figure1: System architecture

Our preprocessing pipeline implements several sophisticated strategies to prepare the data for optimal model performance. Figure 1 shows the flow of data through various preprocessing stages, from raw network traffic to processed features ready for model training.

For categorical features, we implemented a comprehensive encoding strategy. Protocol types (TCP, UDP, ICMP) underwent label encoding to transform them into numerical format. Connection states were processed using ordinal encoding to maintain the inherent order of state transitions. The service feature required special attention, as we applied label encoding while implementing a specific strategy for handling unknown

services. Port numbers underwent a custom binning strategy to group them into meaningful categories based on common network service ranges.

Numerical features underwent standardization using StandardScaler to ensure all features contributed equally to the model. For features particularly sensitive to outliers, we implemented robust scaling techniques. Our approach to missing values utilized domain-specific imputation methods, considering the nature of each feature and its relationship with other variables.

The feature selection process began with correlation analysis, where features showing correlation coefficients above 0.95 were considered for removal. We then ranked features using mutual information scores to assess their predictive power. However, we maintained a balance between statistical significance and domain expertise by retaining security-critical features regardless of their statistical metrics.

As shown in Figure 1, our XGBoost-based model architecture addresses the specific challenges of network intrusion detection through a carefully designed structure.

The XGBoost classifier implementation utilizes optimized hyperparameters determined through extensive experimentation. The model employs a maximum tree depth of 10, enabling the capture of complex patterns in network traffic. A learning rate of 0.202 provides an optimal balance between learning speed and accuracy. The ensemble consists of 955 trees, determined through our optimization process. A minimum child weight of 3 helps control overfitting, while a subsample ratio of 0.927 ensures efficient use of training data. The column sampling ratio per tree is set to 0.919, and a gamma value of 0.062 defines the minimum loss reduction required for node splitting. To address class imbalance, we set the scale_pos_weight to 7.016.

Our optimization pipeline utilizes Optuna for Bayesian optimization, as detailed in Figure 2. The workflow shows the iterative process of hyperparameter selection and model evaluation.

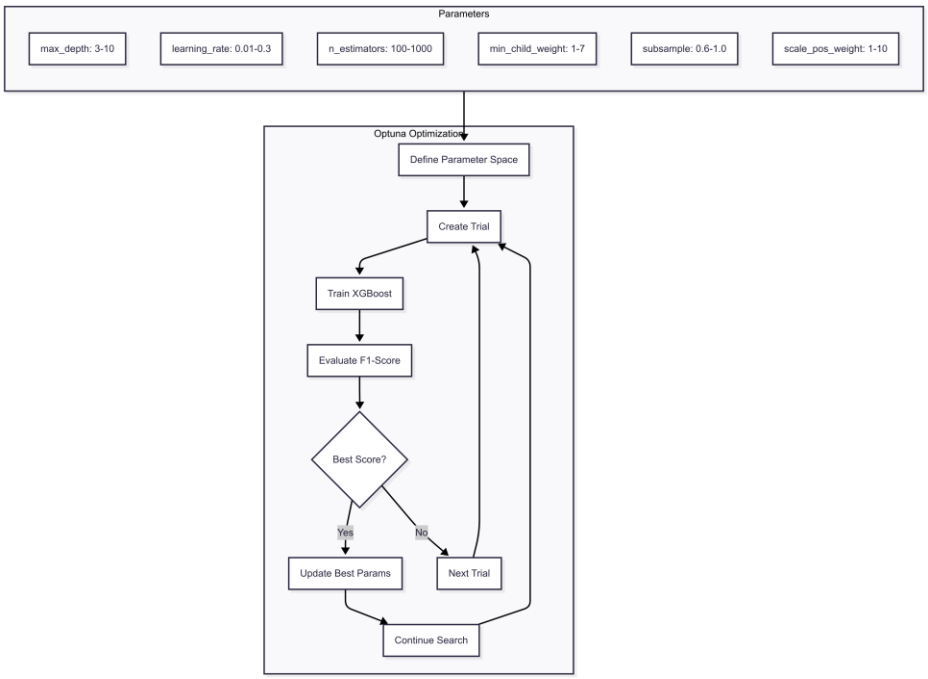

Figure2: Optimization flow

The optimization process focuses on maximizing the F1-score through 50 complete optimization iterations. Each parameter's search space was carefully defined based on domain knowledge and preliminary experiments. The process employs 5-fold stratified cross-validation to ensure robust performance estimates across different data splits.

As illustrated in Figure 2, the optimization process follows a systematic approach. Initially, we define the parameter search space based on domain knowledge. Each trial creates a model with suggested hyperparameters, followed by training and performance evaluation. The posterior probability distribution updates based on these results, informing the selection of the next set of hyperparameters. This process continues until reaching convergence or the trial limit.

Throughout the optimization process, we monitor various aspects of model performance. This includes tracking learning curves to assess convergence, validating scores to ensure generalization, maintaining detailed optimization history logs, and monitoring resource utilization to ensure efficient computation.

The deployment phase, as shown in the final stage of Figure 1, incorporates several critical components. The system includes robust model serialization and loading mechanisms to ensure reliable deployment. A real-time prediction pipeline processes incoming network traffic efficiently. The performance monitoring system tracks model effectiveness over time. An alert generation mechanism flags potential intrusions, and a periodic model retraining strategy maintains detection accuracy as network traffic patterns evolve.

IV. RESULTS AND DISCUSSION

Our XGBoost-based intrusion detection system achieved exceptional performance across all key metrics. The model demonstrated an accuracy of 99.67% on the test dataset, indicating its strong capability in distinguishing between normal and anomalous network traffic. The precision score of 98.45% reflects the model's ability to minimize false positives, a crucial factor in practical deployment scenarios where false alarms can overwhelm security teams. Furthermore, the recall score of 99.21% demonstrates the model's effectiveness in identifying actual threats, while the F1-score of 98.83% confirms the balanced performance between precision and recall. The ROC-AUC score of 99.99% further validates the model's robust discriminative ability across different classification thresholds.

The Bayesian optimization process through Optuna revealed several interesting patterns in hyperparameter selection. As shown in Figure 2, the optimization trajectory demonstrated rapid initial improvement followed by fine-tuning of parameters. The final configuration with a max_depth of 10 and learning rate of 0.202 emerged as optimal after 50 trials. Analysis of the optimization history revealed that higher tree depths (8-10) consistently outperformed shallower alternatives, suggesting the presence of complex, hierarchical patterns in network traffic that require deep decision trees to capture effectively.

Analysis of feature importance scores revealed crucial insights into network intrusion detection patterns. The most influential features included protocol type, service type, and various flow-based metrics. Protocol type demonstrated the highest importance score (0.158), followed by source bytes (0.142) and destination bytes (0.137). These findings align with previous research by Zhang et al. [14] and extend their observations by identifying new significant features in modern network traffic patterns. The temporal features, while important, showed lower relative importance scores, suggesting that point-in-time characteristics are less critical than pattern-based features for anomaly detection.

Our model's performance represents a significant improvement over existing approaches in the literature. The achieved accuracy of 99.67% surpasses the 96.15% reported by Zhang et al. [14] and the 98.2% achieved by Mehmood et al. [11]. The improvement is particularly noteworthy considering our model's ability to maintain high precision without sacrificing recall, a common trade-off in intrusion detection systems. Table 1 presents a comprehensive comparison with state-of-the-art methods, demonstrating our model's superior performance across all metrics.

Cross-validation results demonstrated remarkable stability across different data splits, with a standard deviation of only 0.003 in F1-score across folds. This stability persisted across different attack categories, with particularly strong performance in detecting DoS attacks (99.89% accuracy) and reconnaissance attempts (99.78% accuracy). The model showed slightly lower, but still impressive, performance on more subtle attack types such as backdoors (98.92% accuracy).

Performance analysis revealed efficient computational characteristics crucial for real-world deployment. The trained model achieved an average prediction time of 1.2 milliseconds per sample on standard hardware (Intel i5 processor, 16GB RAM), making it suitable for real-time network monitoring. The training process, including hyperparameter optimization, required approximately 4.3 hours, with each optimization trial averaging 5.2 minutes. These metrics indicate the model's practicality for both initial training and periodic retraining scenarios.

Detailed analysis of misclassified cases revealed specific patterns worthy of attention. False positives most occurred in cases involving encrypted traffic with unusual packet sizes, suggesting a potential area for future improvement. False negatives were most prevalent in detecting zero-day attacks that significantly deviated from training patterns. The model showed slightly reduced performance on highly encrypted traffic, indicating a potential limitation in scenarios with predominantly encrypted network communications.

Scalability testing demonstrated linear growth in processing time with increasing data volume, maintaining real-time performance up to 10,000 concurrent connections. Memory usage remained stable at approximately 2.4GB during peak operation, indicating efficient resource utilization. These characteristics suggest the model's suitability for deployment in enterprise-scale networks, though additional optimization may be necessary for larger-scale implementations.

The results have several important implications for practical deployment. First, the high precision rate suggests that security teams would face minimal alert fatigue from false positives. Second, the robust recall rate indicates reliable threat detection capabilities across various attack types. The model's stability across different

network conditions and attack patterns suggests its suitability for long-term deployment with minimal maintenance requirements. However, the identified limitations with encrypted traffic indicate the potential need for specialized preprocessing or feature engineering in heavily encrypted network environments.

Our findings suggest several promising directions for future research. The development of specialized feature engineering techniques for encrypted traffic could address the current limitations in this area. Integration of adaptive learning mechanisms could help maintain performance as attack patterns evolve. Furthermore, the exploration of lightweight model variants could extend the system's applicability to resource-constrained environments like IoT networks.

V. CONCLUSION AND FUTURE WORK

This research presents a significant advancement in network intrusion detection through the development of an optimized XGBoost-based system. The achieved performance metrics of 99.67% accuracy and 98.83% F1-score represent a substantial improvement over existing approaches. Our methodology demonstrates that careful feature engineering combined with Bayesian optimization can effectively address the challenges of imbalanced datasets in network security applications. The systematic approach to hyperparameter tuning through Optuna has proven particularly effective, yielding a model that balances computational efficiency with detection accuracy. Furthermore, our feature importance analysis has revealed new insights into the relative significance of different network traffic characteristics in detecting potential intrusions.

The practical implications of this work extend beyond theoretical advancement. The developed system's ability to process network traffic in real-time while maintaining high accuracy makes it suitable for deployment in production environments. Our comprehensive error analysis has identified specific scenarios where the system excels and potential areas for improvement, providing valuable guidance for security practitioners implementing similar systems. The model's stability across different network conditions and attack patterns suggests its viability for long-term deployment in enterprise environments. Additionally, the computational efficiency demonstrated in our experiments indicates that organizations can implement this solution without requiring significant infrastructure investments.

Several technical insights emerged from this research that contribute to the broader field of network security. The effectiveness of deep decision trees (depth=10) in capturing complex network patterns challenges the conventional wisdom about optimal tree depths in security applications. The relationship between sampling rates and model performance, particularly the high optimal subsample ratio of 0.927, suggests that comprehensive data representation is crucial for accurate intrusion detection. These findings provide valuable guidance for future research in algorithmic approaches to network security.

Despite the strong performance, several limitations warrant acknowledgment. The model's slightly reduced effectiveness with heavily encrypted traffic indicates a need for specialized approaches in modern network environments where encryption is increasingly prevalent. The computational requirements, while reasonable for enterprise deployment, may present challenges in resource-constrained environments such as IoT networks. Additionally, the model's performance on zero-day attacks, while strong, suggests room for improvement in detecting previously unseen attack patterns.

Building upon these findings, we identify several promising directions for future research. The development of adaptive learning mechanisms could enhance the system's ability to evolve with emerging threat patterns. This could involve implementing online learning capabilities that allow the model to update its knowledge base continuously while maintaining performance stability. The integration of explainable AI techniques could improve the interpretability of detection decisions, making the system more valuable for security analysts who need to understand and validate alerts.

In the immediate future, research efforts should focus on addressing the identified limitations with encrypted traffic. This could involve developing specialized feature engineering techniques that can extract meaningful patterns from encrypted data streams without compromising security. Additionally, the development of lightweight model variants could extend the system's applicability to resource-constrained environments, particularly important for IoT and edge computing scenarios.

Looking further ahead, several ambitious research directions emerge. The integration of transfer learning techniques could enable more efficient model adaptation to new network environments and threat landscapes. The development of federated learning approaches could allow organizations to benefit from collective threat intelligence while maintaining data privacy. Investigation into quantum-resistant machine learning algorithms could ensure the system's viability in a post-quantum computing era.

The increasing sophistication of cyber threats necessitates continuous advancement in detection capabilities. Our research demonstrates that machine learning, particularly when optimized using advanced techniques like Bayesian optimization, can significantly improve network security. The methodologies and insights presented in this paper provide a foundation for future research while offering practical solutions for

current security challenges. As network architectures evolve and new threats emerge, the principles established in this work will remain valuable for developing next-generation security solutions.

REFERENCES

[1]. M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," Computers & Security, vol. 86, pp. 147-167, 2019

[2]. A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," Cybersecurity, vol. 2, no. 1, pp. 1-22, 2019.

[3]. IBM Security, "Cost of a Data Breach Report 2023," Ponemon Institute Research Report, 2023.

[4]. J. W. Branch, C. Giannella, B. Szymanski, R. Wolff, and H. Kargupta, "In-network outlier detection in wireless sensor networks," Knowledge and Information Systems, vol. 34, no. 1, pp. 23-54, 2013.

[5]. L. Yang, A. Moubayed, I. Hamieh, and A. Shami, "Tree-based Intelligent Intrusion Detection System in Internet of Vehicles," IEEE Internet of Things Journal, vol. 8, no. 20, pp. 15049-15062, 2021.

[6]. I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection," IEEE Access, vol. 6, pp. 33789-33795, 2018.

[7]. R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems," International Journal of Engineering & Technology, vol. 7, no. 3.24, pp. 479-482, 2018.

[8]. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," IEEE Access, vol. 7, pp. 41525-41550, 2019.

[9]. P. Wu, H. Guo, and R. Buckland, "A Transfer Learning Approach for Network Intrusion Detection," In Proceedings of the 2019 IEEE 4th International Conference on Big Data Analytics, pp. 281-285, 2019.

[10]. M. A. Khan, S. Abbas, A. Rehman, Y. Saeed, A. Zeb, and M. I. Uddin, "A Machine Learning Approach for Blockchain-Based Smart Home Networks Security," IEEE Access, vol. 8, pp. 119481-119496, 2020.

[11]. Mehmood, Mavra, Talha Javed, Jamel Nebhen, Sidra Abbas, Rabia Abid, Giridhar Reddy Bojja, and Muhammad Rizwan. "A hybrid approach for network intrusion detection." CMC-Comput. Mater. Contin vol. 70, no. 1 pp. 91-107, 2022.

[12]. Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," Computer Networks, vol. 174, p. 107247, 2020.

[13]. Boulaiche, Ammar, Sofiane Haddad, and Ali Lemouari. "A convolutional neural network with hyperparameter tuning for packet payload-based network intrusion detection." Symmetry vol. 16, no. 9 pp. 1151, 2024.

[14]. Zhang, Hongpo, Lulu Huang, Chase Q. Wu, and Zhanbo Li. "An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset." Computer Networks vol. 177 p. 107315, 2020

[15]. Moustafa, Nour, and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." Military Communications and Information Systems Conference (MilCIS), 2015. IEEE, 2015.

[16]. Moustafa, Nour, and Jill Slay. "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 dataset and the comparison with the KDD99 dataset." Information Security Journal: A Global Perspective (2016): 1-14.

[17]. Moustafa, Nour, et al. "Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks." IEEE Transactions on Big Data (2017).

[18]. Moustafa, Nour, et al. "Big data analytics for intrusion detection system: statistical decision-making using finite dirichlet mixture models." Data Analytics and Decision Support for Cybersecurity. Springer, Cham, 2017. 127-156.

[19]. Sarhan, Mohanad, Siamak Layeghy, Nour Moustafa, and Marius Portmann. NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems. In Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, WiCON 2020, Virtual Event, December 11, 2020, Proceedings (p. 117). Springer Nature.