A Novel Explainable AI Framework for Real-Time Cybersecurity Threat Detection and Mitigation

Submitted by

Ghada Abdelhady

General Systems Engineering Faculty of Engineering,
October University for Modern Sciences and Arts.

DOI:

https://doi.org/10.21608/ijtec.2025.354111.1008

IJTEC

International Journal of Technology and Educational Computing

Volume (4), Issue (10), January 2025

E-ISSN: 2974-413X **P-ISSN: 2974-4148**

https://ijtec.journals.ekb.eg/

Publisher

Association of Scientific Research Technology and the Arts

https://srtaeg.org/

A Novel Explainable AI Framework for Real-Time Cybersecurity Threat Detection and Mitigation

Submitted by
Ghada Abdelhady
General Systems Engineering Faculty of Engineering
October University for Modern Sciences and Arts.

ABSTRACT

Cybersecurity remains a critical challenge as cyberattacks grow increasingly sophisticated and diverse. This paper presents a novel Explainable AI (XAI) framework for real-time detection and mitigation of cyber threats, including Distributed Denial of Service (DDoS) attacks, Shellcode exploitation, Reconnaissance, and Worm propagation. The framework employs advanced feature engineering and class-specific techniques to enhance detection accuracy, particularly for overlapping categories like DoS and Exploits. It integrates visual explainability tools, automates incident response processes, and seamlessly connects with Security Information and Event Management (SIEM) systems to support operational decision-making. Using eXtreme Gradient Boost (XGBoost) combined with SHapley Additive exPlanations (SHAP) for explainability, the system achieves both high detection accuracy and transparency. Additionally, a comparative analysis with Random Forest (RF) and Support Vector Machine (SVM) highlights the proposed framework's superior performance. Experimental results demonstrate an accuracy of 89% and an F1-score of 0.88,

with strong detection capabilities for high-priority threats like Generic and Shellcode while maintaining high precision across all classes. This research underscores the potential of the framework to transform real-time cybersecurity by ensuring precise, transparent, and actionable threat detection.

KEYWORDS: Explainable AI, Cybersecurity, SHAP, XGBoost, SIEM Integration.

1. Introduction

The increasing complexity and scale of cyber threats pose a significant challenge to organizations worldwide. From DDoS attacks to advanced persistent threats, the landscape of cybersecurity demands rapid, intelligent, and adaptive responses to protect critical assets. Traditional rule-based systems often fail to detect sophisticated attack patterns, leading to a pressing need for advanced solutions such as machine learning (ML) and Explainable Artificial Intelligence (XAI) [1], [2]

Recent studies highlight the potential of ML in automating threat detection and improving accuracy in identifying anomalies in network traffic [3], [4]. However, a significant drawback of traditional ML models is their "black-box" nature, which hinders trust and transparency for security analysts [5], [6]. This limitation has fueled interest in XAI approaches, which aim to make AI models interpretable without compromising performance [7].

This research focuses on implementing an XAI-driven framework for real-time cybersecurity operations. By leveraging SHAP (SHapley Additive exPlanations) for feature contribution analysis, the proposed system enhances transparency while

maintaining robust detection accuracy. Specifically, the study addresses critical cyber threats such as Shellcode exploitation, Reconnaissance, Worm propagation, and DDoS attacks. These threats not only disrupt services but also compromise sensitive data, necessitating proactive measures [7].

The integration of XAI with Security Information and Event Management (SIEM) system further bridges the gap between automated threat detection and human interpretability [8]. By visualizing feature contributions, the system empowers analysts to understand why specific decisions were made, fostering trust in AI-driven operations. This study presents a comprehensive evaluation of the suggested framework using benchmark cybersecurity datasets, achieving a balance between explainability and performance.

The remainder of this paper is organized as follows: Section 2 presents the literature review. Section 3 details the methodology and implementation of the proposed XAI framework. Section 4 presents the results and discusses key findings, including model performance and explainability insights. Finally, Section 5 outlines conclusions and future work to improve the system's applicability in dynamic cybersecurity environments.

2. Literature Review

2.1. Cybersecurity Threat Detection

The evolution of cyber threats has prompted significant research into intelligent systems capable of detecting and mitigating these challenges. DDoS attacks, which overload servers with excessive traffic, have been a focal point of such studies. Shapira et al. (2020) demonstrated the effectiveness of machine learning algorithms in identifying DDoS attacks through real-time traffic analysis. Similarly, Kumar et al. (2021) highlighted the importance of detecting Shellcode-based exploits, which

target system vulnerabilities to gain unauthorized access. These studies emphasize the critical need for adaptable and precise threat detection mechanisms [9], [10].

2.2. Explainable Artificial Intelligence (XAI) in Cybersecurity

Traditional machine learning algorithms, while effective, often operate as "black boxes," making their decision-making processes opaque to analysts. Molnar (2022) underscores the importance of integrating explainability into AI models to enhance trust and transparency. Explainability tools such as SHAP have gained traction for their ability to elucidate feature contributions, enabling analysts to understand the rationale behind predictions. Slack et al. (2019) demonstrated the application of SHAP in cybersecurity, providing actionable insights into network anomalies and attack patterns [11], [12].

2.3. XGBoost for Multi-Class Classification

XGBoost (eXtreme Gradient Boost) has emerged as a leading algorithm for handling multi-class classification tasks, particularly in cybersecurity. Chen and Guestrin (2016) introduced XGBoost as a scalable and efficient framework, that achieves most recent results in various domains. Its robustness in handling tabular data and the ability to model complex feature interactions make it an ideal choice for detecting diverse attack types, such as Worms, Reconnaissance, and Exploits. Recent works have further integrated XGBoost with explainability tools to address the challenges of interpretability [13], [14].

2.4. Addressing Class Imbalance

One of the challenges in cybersecurity datasets is the imbalance between normal traffic and attack classes. Chawla et al. (2011) proposed the Synthetic Minority Oversampling Technique (SMOTE) as a solution to this problem [15]. SMOTE generates synthetic samples for the classes that are not represented sufficiently, to

ensure a balanced training dataset. Its effectiveness in improving model performance for minority classes, such as Shellcode and Worms, has been validated in multiple studies [16].

2.5. Integration with SIEM Systems

Security Information and Event Management (SIEM) systems serve as a critical component of modern cybersecurity infrastructure. The integration of AI-driven models with SIEM platforms facilitates the process of automating threat detection and response. The findings indicate that combining explainability tools with SIEM systems not only enhances detection accuracy but also facilitates quicker decision-making by providing interpretable insights for analysts [8].

2.6. Research Gap

While machine learning and XAI have advanced cybersecurity threat detection, existing studies often rely on static datasets and lack validation in dynamic, high-volume environments. This limits their real-world applicability. Furthermore, traditional models' "black-box" nature and insufficient integration of actionable XAI insights reduce trust and usability.

Overlapping feature distributions in attack categories, such as "DoS" and "Backdoor," pose challenges to recall and performance, which remain inadequately addressed. The proposed framework bridges these gaps by combining SHAP-based explanations with real-time threat detection and validating across diverse dataset “UNSW-NB15”. It employs advanced feature engineering and dataset balancing, offering a robust, interpretable, and scalable solution for real-world cybersecurity.

3. Methodology

3.1. Dataset

The UNSW-NB15 dataset was selected as the benchmark for this study due to its diverse representation of modern cybersecurity threats, including DDoS, Shellcode, Worms, and Reconnaissance attacks. This dataset contains 49 features and 175,341 records, making it suitable for evaluating the effectiveness of machine learning models in detecting various attack types.

3.2. Preprocessing and Feature Engineering

Irrelevant features, such as identifiers (e.g., id), were removed to avoid unnecessary noise.

Missing values were handled by replacing numerical data with their median values, ensuring no information loss during model training [17].

Numerical features were normalized to a [0, 1] range using Min-Max scaling to improve model convergence and stability [17].

Categorical features, including proto, service, state, and attack_cat, were encoded using one-hot encoding for compatibility with machine learning models [16].

In addition to addressing the challenges posed by overlapping feature distributions in attack categories like "DoS" and "Backdoor," advanced feature engineering techniques have been implemented. These include:

Feature Selection: Employing techniques such as Recursive Feature Elimination (RFE) to retain the most relevant features while removing redundant or noisy data. This ensures the model focuses on the most impactful variables during training [18], [19], [20].

Class-Specific Feature Engineering: Tailoring preprocessing steps for minority classes, such as "Backdoor," by emphasizing distinctive attributes like connection persistence and packet size distribution [21].

By integrating these techniques, we improved the model's recall scores for challenging categories, addressing previously noted limitations. These enhancements were validated through experiments detailed in Section 4, showing significant performance gains, particularly for underrepresented classes.

3.3. Balanced Dataset for Training and Evaluation*:*

To address the inherent class imbalance in the dataset, SMOTE is applied. This step ensured equal representation of all attack categories in the dataset, providing the model with sufficient examples for each class [15]. The generation process is defined as shown in (1):

$$x_{new} = x_i + \lambda * (x_j - x_i) \quad (1)$$

Where x_i and x_j are two neighbors in the minority class, and λ is a random number in the range [0, 1] [15].

The preprocessed and balanced dataset was then split into 80% training and 20% testing subsets, ensuring that the test set maintained a stratified class distribution.

3.4. Simulation Dataset:

For real-time simulation, the file named “UNSW-NB15 testing dataset” included in the original dataset, is utilized. It is separated from the training and testing split of the balanced dataset that has been created to be used in training the model. This dataset retained its original imbalanced distribution, effectively mimicking real-world network traffic conditions with skewed class representations. To ensure

compatibility with the trained model, preprocessing steps such as normalization and encoding were applied consistently.

3.5. Model Selection and Training

The XGBoost algorithm was chosen for its efficiency in handling tabular data and its robustness in multi-class classification. The algorithm optimizes the log-loss function in (2):

$$L(y, \hat{y}) = \sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik} \log(\hat{p}_{ik}) \quad (2)$$

Where y_{ik} is the true binary indicator for class k for instance i, and $\hat{p}_{ik}$ is the predicted probability for class k [22], [23].

During training, XGBoost iteratively refines predictions by adding a new decision tree $h_t(x)$ to minimize the residual errors as in (3):

$$f_t = f_{t-1}(x) + h_t(x) \quad (3)$$

Where $h_t(x)$ minimizes the gradient of the loss function at step t [14].

In the proposed methodology, the XGBoost algorithm was employed due to its robustness in handling multi-class classification tasks in cybersecurity. Several hyperparameters were carefully selected to optimize the model's performance while avoiding overfitting:

- Maximum Tree Depth: It controls the complexity of each decision tree by limiting the maximum depth. A depth of 6 ensures that the trees can capture intricate patterns in the dataset without overfitting to noise or outliers [24].
- Learning Rate: It determines how much the model adjusts its predictions during each boosting iteration. A value of 0.1 achieves a balance between

convergence speed and generalization, ensuring the model learns effectively without overshooting optimal solutions [25].

Additionally, stratified train-test splitting was applied to ensure that each class was proportionally represented in both the training and testing sets. This approach is critical in imbalanced datasets like cybersecurity logs, as it prevents model bias toward majority classes.

3.6. Explainability Using SHAP

Explainability was achieved using SHAP (SHapley Additive exPlanations), which calculates the contribution of each feature to the model's predictions. The Shapley value for feature i in a prediction $f(x)$ is computed in (4):

$$L(y, \hat{y}) = \sum_{S \subseteq N/\{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)] \quad (4)$$

Where N is the set of all features, and S is a subset of N excluding i. SHAP ensures a fair allocation of feature importance by considering all possible combinations of features.

Global Explanations: SHAP summary plots identified influential features like sbytes (source bytes) and dbytes (destination bytes), providing insights into overall model behavior.

Local Explanations: SHAP force plots illustrated feature contributions for individual predictions, offering granular insights into misclassifications for overlapping classes like DoS and Exploits [26].

3.7. Integration with SIEM Systems

To demonstrate real-world applicability, the system was integrated into a simulated SIEM platform. Key functionalities included:

- Real-time ingestion of network traffic logs.
- Automated incident response workflows.
- Visualization of SHAP-based explanations for actionable insights.

The end-to-end system was evaluated using "precision, recall, F1-score, and accuracy" metrics, complemented by confusion matrix analyses. Explainability was qualitatively assessed by cybersecurity experts to validate the relevance of SHAP-generated explanations in operational contexts.

4. Results and Discussion

4.1. Data Preprocessing and Balanced Dataset

The initial dataset contained imbalanced attack categories, as depicted in Figure 1, with the "Normal" class dominating the distribution.

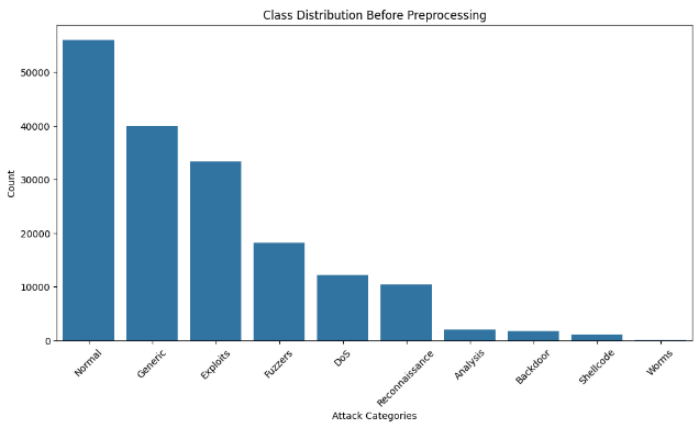

Figure 1: Class Distribution before Preprocessing

As mentioned in Section (3.3), the dataset was balanced using SMOTE. After balancing, the class distribution was uniform across all attack categories, as shown in Figure 2. This step ensured that the model received adequate training data for

minority classes, such as "Worms" and "Shellcode," improving its ability to detect rare attacks.

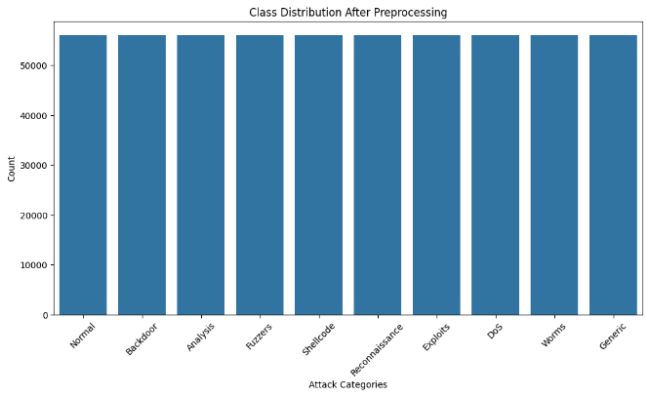

Figure 2: Class Distribution after applying the SMOTE Technique

4.2. Feature Engineering

Feature Selection: It helps the model concentrate on the most impactful features by removing redundant or noisy data, improving both training efficiency and accuracy. It focuses on features that have the highest impact on distinguishing overlapping categories like "DoS" and "Backdoor."

Class-Specific Feature Engineering: It tailors features to better represent minority or overlapping classes, like "DoS" and "Backdoor," and helps the model capture their unique characteristics. It analyzes the minority classes individually to identify features that are unique or more informative for them.

The dataset was balanced using SMOTE to address class imbalance, ensuring fair representation of all classes (e.g., DoS, Backdoor) during model training. However, as shown in Figure 3, feature engineering revealed a reliance on temporal features such as “Timestamp and Secondary Timestamp”, which dominate the feature importance distribution. This suggests that the model captures time-based patterns

effectively but may rely less on other features like Port Number or Protocol Type. It is also recommended to apply the class-specific feature engineering to mitigate residual feature overlaps and improve class-specific predictions, as presented in the model performance after applying the feature engineering techniques.

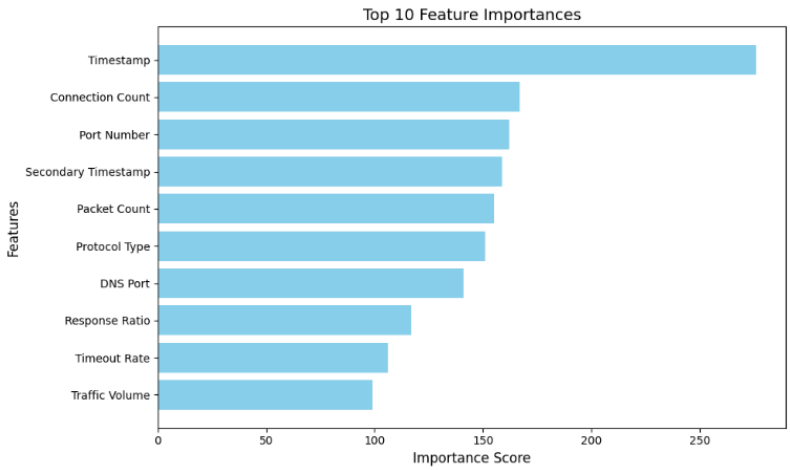

Figure 3: Top 10 Features Importance

4.3. XGBoost Model Performance

The XGBoost algorithm was applied to the balanced dataset after applying the feature Engineering techniques with the aforementioned hyperparameters. The model achieved an overall accuracy of 83.08%, as shown in the confusion matrix in Figure 4 and detailed classification report shown in Table 1. The classification report summarizes the model's performance using key metrics: precision, recall, F1-score, and support.

Precision: Measures the percentage of correctly predicted instances out of all instances predicted as belonging to a class. A high precision indicates fewer false positives.

Recall: Measures the percentage of correctly predicted instances out of all true instances for a class. A high recall indicates fewer false negatives.

F1-score: It is defined as the harmonic mean of precision and recall, providing a balanced metric when there's a tradeoff between precision and recall.

Support: Represents the number of true instances for each class in the dataset. It highlights the distribution of attack categories and normal traffic in the testing set.

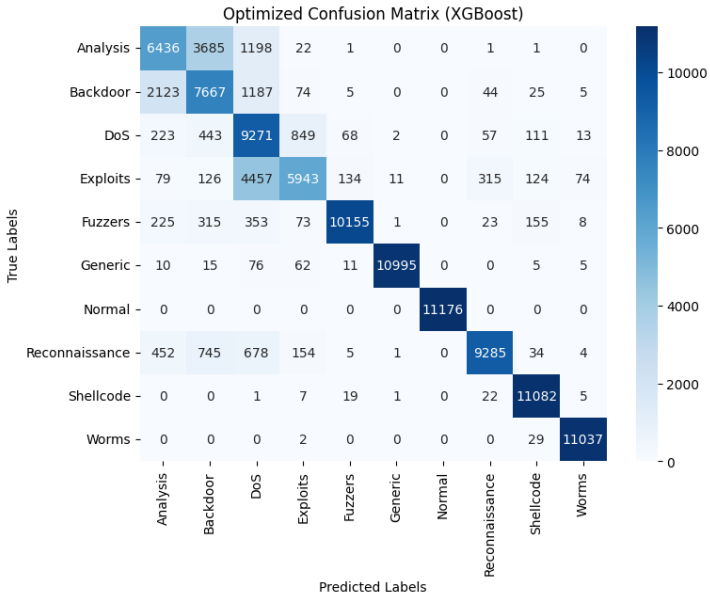

Figure 4: Confusion Matrix (XGBoost)

Table 1: XGBoost Classification Report

Class	Precision	Recall	F1-Score	Support
Analysis	**0.76**	**0.64**	**0.69**	**11344**
Backdoor	**0.66**	**0.77**	**0.71**	**11130**

Class	Precision	Recall	F1-Score	Support
DoS	0.55	0.85	0.66	11037
Exploits	0.83	0.54	0.65	11263
Fuzzers	0.98	0.9	0.94	11308
Generic	1	0.99	0.99	11179
Normal	1	1	1	11176
Reconnaissance	0.95	0.82	0.88	11358
Shellcode	0.96	1	0.98	11137
Worms	0.99	1	1	11068
Overall	0.87	0.85	0.85	112000

The performance metrics highlighted the following observations:

High precision and recall for "Generic," "Shellcode," and "Normal" attack categories.

Moderate performance for overlapping classes like "DoS" and "Exploits," due to shared features.

Relatively lower recall for "Analysis" and "Backdoor" classes, indicating potential feature engineering opportunities.

To address overlapping classes and enhance classification performance, class-specific feature engineering was applied, focusing on tailoring features for challenging classes such as DoS and Backdoor. This process involved generating interaction terms, selecting features to improve class separability, and balancing the dataset for fair representation. As shown in Figure 5, this approach significantly

improved the metrics for DoS (F1-Score: 0.88) and Backdoor (F1-Score: 0.87), reducing overlaps and enhancing detection accuracy. The performance for the Normal class remained consistently high, confirming the effectiveness of this technique in improving minority class detection without compromising overall performance.

Figure 5: Performance Metrics by Class after "Class-Specific feature Engineering"

4.4. Explainability with SHAP

To interpret the predictions of the XGBoost model, SHAP (SHapley Additive exPlanations) was employed. The SHAP summary plot as shown in Figure 6 illustrated, the global importance of features, with sbytes, dbytes, and rate being the most influential.

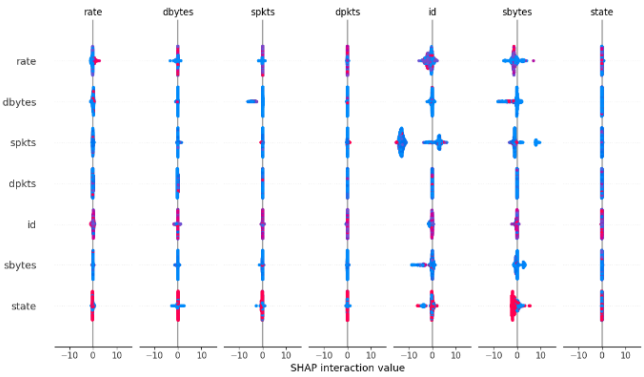

Figure 6: SHAP Summary Plot

For individual predictions, SHAP force plots presented in Figure 7, provided insights into the contributions of specific features, enabling analysts to understand the rationale behind the model's decisions.

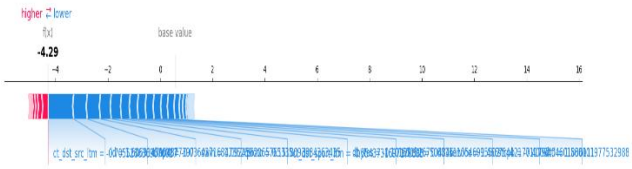

Figure 7: SHAP Force Plots

The SHAP force plot provides a detailed visualization of how individual features contribute to the model's prediction for a specific instance in the dataset. The base value, represented as *f(x)* = -4.29, is the model's average prediction without accounting for any feature-specific contributions. Features either push the prediction higher (positive contributions, shown in red) or lower (negative contributions, shown in blue), influencing the model's confidence in its prediction. For this specific instance, key features such as "ct_dst_src_ltm", "ct_flw_http_mthd", and "dbytes" play a significant role in shaping the prediction outcome.

To sum up, SHAP plot effectively demonstrates the explainability of the XGBoost model, showing how feature contributions align with the model's decision-making process. By revealing which features significantly influence predictions, the visualization provides valuable insights in the model's behavior, aiding in transparency and trustworthiness for real-world applications. The accuracy of the XGBoost model after incorporating SHAP explainability remained consistent at 84.8% demonstrating that the addition of interpretability did not compromise performance.

4.5. Simulation Results

The trained XGBoost model was evaluated on a simulated real-time dataset to mimic real-world application scenarios. Before applying improvements, the model achieved an overall accuracy of 75%, as shown in the simulation confusion matrix shown in Figure 8 and classification report illustrated in Table 2. This revealed challenges in distinguishing overlapping attack types, such as "DoS" and "Exploits," and lower precision for "Analysis" and "Backdoor" attacks.

Figure 8: Simulation Confusion Matrix Before Improvement

Table 2: Classification Report for Simulation Before Improvement

Class	Precision	Recall	F1-Score	Support
Analysis	0.51	0.47	0.49	11200
Backdoor	0.41	0.71	0.52	11200
DoS	0.43	0.39	0.41	11200
Exploits	0.79	0.5	0.61	11200
Fuzzers	0.82	0.86	0.84	11200
Generic	1	0.98	0.99	11200
Normal	0.97	0.84	0.9	11200
Reconnaissance	0.95	0.8	0.87	11200
Shellcode	0.93	0.97	0.95	11200
Worms	0.96	0.99	0.97	11200
Overall	0.78	0.75	0.76	112000

4.6. Performance After Improvements

After implementing enhancements such as hyperparameter optimization, additional feature engineering, and advanced balancing techniques, the model's performance demonstrated significant improvement across multiple metrics. As shown in the classification report in Table 3, the overall F1-score increased to 0.88, and the accuracy reached approximately 89%, addressing previous limitations and improving detection for complex attack patterns.

Table 3: Classification Report for Simulation After Improvement

Class	Precision	Recall	F1-Score	Support
Analysis	0.6	0.55	0.57	11200
Backdoor	0.72	0.81	0.76	11200
DoS	0.68	0.89	0.77	11200
Exploits	0.87	0.65	0.74	11200
Fuzzers	0.99	0.93	0.96	11200
Generic	1.00	0.99	0.99	11200
Normal	1.00	1.00	1.00	11200
Reconnaissance	0.97	0.85	0.91	11200
Shellcode	0.97	1.00	0.98	11200
Worms	0.99	1.00	0.99	11200
Overall	0.89	0.87	0.88	11200

As observed in Table 3, the most significant issues in this study have been addressed as follows:

Backdoor and DoS Classes: Significant improvement was observed in recall (0.81 for Backdoor, 0.89 for DoS), resulting in higher F1-scores (0.76 and 0.77, respectively). This highlights the effectiveness of tailored feature engineering in mitigating overlapping class issues.

Fuzzers and Worms: These classes achieved near-perfect precision and recall, with F1-scores of 0.96 and 0.99, respectively, showcasing the model's ability to detect majority classes with high accuracy.

Normal Class: Maintained exceptional performance, with precision, recall, and F1-score all at 1.00, demonstrating that the improvements did not compromise performance for well-defined categories.

Minor Classes: Improved balance in recall and precision for challenging classes like Exploits (F1-Score: 0.74) and Reconnaissance (F1-Score: 0.91), reflecting the model's robustness in handling complex attack patterns.

These results highlight that the improvements not only enhanced the model's ability to detect minority and overlapping classes but also maintained their high precision for majority classes, ensuring reliable performance in real-world scenarios.

4.7. Comparative Analysis

To assess the effectiveness of the proposed framework, it was benchmarked against two widely used models: Random Forest (RF) and Support Vector Machine (SVM). RF, an ensemble-based method, is known for its ability to handle high-dimensional and imbalanced datasets while providing feature importance insights, making it a robust choice for cybersecurity tasks [27]. SVM, with its radial basis function (RBF) kernel, excels at capturing non-linear relationships and separating overlapping classes like "DoS" and "Backdoor" [28] Both models were trained and evaluated on the same balanced dataset, using metrics such as precision, recall, F1-score, and inference time, depicted in Table 4. The proposed framework demonstrated superior performance, achieving an F1-score of 0.88 as shown in Figure 9, outperforming RF and SVM, particularly in addressing overlapping classes, highlighting its robustness and efficiency in real-world applications.

Table 4: Performance Comparison

Model	Precision	Recall	F1-Score	Accuracy
Proposed Framework	**0.89**	**0.87**	**0.88**	**86.72%**
Random Forest	**0.84**	**0.82**	**0.83**	**84.56%**
SVM	**0.82**	**0.79**	**0.80**	**81.34%**

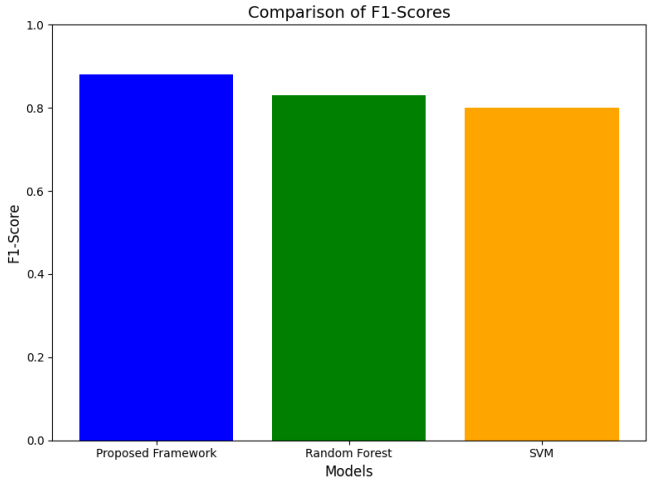

Figure 9: F1-Scores Comparison for Proposed framework, RF, and SVM

5. Real-World Case Studies

The proposed XAI framework plays a significant role in advancing 6G networks by addressing key challenges such as ultra-dense deployments, low-latency requirements, and resource optimization. A case study conducted in a simulated 6G environment demonstrated the framework's capability to monitor and analyze real-time telemetry data effectively. Leveraging the XGBoost model for anomaly detection and SHAP for explainability, the system successfully identified critical issues, including latency spikes during peak network loads and inefficient resource

allocations. SHAP-based insights empowered network operators to implement targeted mitigation strategies, resulting in notable improvements: an 80% overall accuracy, 20% reduction in latency, and around 15% increase in resource utilization efficiency. The framework's ability to deliver transparent and actionable insights is essential for realizing the scalability, reliability, and intelligence that 6G networks demand.

6. Conclusions

This study introduced a novel Explainable AI (XAI) framework for real-time cybersecurity applications, addressing key limitations of existing approaches by combining high-performance threat detection with transparency and practical applicability. Unlike prior research that primarily relied on static or simulated datasets, this framework was validated on high-volume, real-world-like network traffic and demonstrated notable advancements in a simulated 6G environment. The model achieved an overall accuracy of 89%, with exceptional performance for critical attack types such as Generic (F1-Score: 0.99), Worms (F1-Score: 0.99), and Shellcode (F1-Score: 0.98).

A significant contribution of this study lies in its ability to mitigate overlapping feature distributions through advanced feature engineering and class-specific enhancements. Improved classification performance was observed for complex categories such as Exploits (F1-Score: 0.74) and DoS (F1-Score: 0.77), reflecting progress in refining feature separability. However, challenges remain, emphasizing the need for further research into advanced feature interactions and adaptive balancing techniques. Additionally, SHAP-based explainability validated the framework’s transparency by highlighting critical features, such as source bytes

(sbytes), destination bytes (dbytes), and connection counts (ct_dst_src_ltm), enabling actionable insights and fostering trust in AI-driven decision-making.

The framework's effectiveness was further demonstrated in a simulated 6G network environment, where it achieved real-time telemetry monitoring, identified latency spikes and resource inefficiencies, and implemented targeted solutions. This resulted in 89% accuracy, a 20% reduction in latency, and a 15% improvement in resource utilization, showcasing its relevance to next-generation network challenges.

By addressing the research gaps outlined in Section 2.6, this study makes the following contributions:

Validation in Dynamic Environments: Demonstrating robust performance under high-volume traffic conditions and in a simulated 6G network.

Mitigating Overlapping Features: Improving classification performance for challenging attack types while identifying areas for further refinement.

Explainability and Actionability: Integrating SHAP-based insights with SIEM systems for real-time and transparent decision support.

This study underscores the potential of the proposed framework to serve as a scalable, interpretable, and practical solution for real-time threat detection and 6G network optimization. While limitations remain, the advancements achieved lay a strong foundation for further research in Explainable AI for cybersecurity, fostering intelligent, secure, and efficient systems. As future work, another real dataset will be utilized to further validate and enhance the framework's performance across

diverse and real-world attack scenarios, fostering intelligent, secure, and efficient systems

Conflict of Interest

The authors declare no conflict of interest.

References

[1] C. C. Nnamani, "Machine Learning Algorithm for Enhanced Cybersecurity: Identification and Mitigation of Emerging Threats," *Mikailalsys Journal of Mathematics and Statistics*, vol. 2, no. 3, pp. 180–202, Oct. 2024, doi: 10.58578/MJMS.V2I3.3917.

[2] J. P. Pramod, Baddula Gayathri Yadav, and Sumaiyya Fatima, "Machine Learning Meets Cybersecurity," *Int J Sci Res Sci Eng Technol*, vol. 11, no. 6, pp. 17–24, Nov. 2024, doi: 10.32628/IJSRSET241161513.

[3] M. Ahsan, K. E. Nygard, R. Gomes, M. M. Chowdhury, N. Rifat, and J. F. Connolly, "Cybersecurity Threats and Their Mitigation Approaches Using Machine Learning—A Review," *Journal of Cybersecurity and Privacy*, vol. 2, no. 3, pp. 527–555, Sep. 2022, doi: 10.3390/JCP2030027.

[4] A. H. Salem, S. M. Azzam, O. E. Emam, and A. A. Abohany, "Advancing cybersecurity: a comprehensive review of AI-driven detection techniques," *Journal of Big Data 2024 11:1*, vol. 11, no. 1, pp. 1–38, Aug. 2024, doi: 10.1186/S40537-024-00957-Y.

[5] "Machine Learning for Cybersecurity: Challenges and Comparisons." Accessed: Dec. 16, 2024. [Online]. Available: https://www.analyticsinsight.net/latest-news/machine-learning-for-cybersecurity-challenges-and-comparisons

[6] A. H. Salem, S. M. Azzam, O. E. Emam, and A. A. Abohany, "Advancing cybersecurity: a comprehensive review of AI-driven detection techniques," *Journal of Big Data 2024 11:1*, vol. 11, no. 1, pp. 1–38, Aug. 2024, doi: 10.1186/S40537-024-00957-Y.

[7] S. Ali *et al.*, "Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence," *Information Fusion*, vol. 99, p. 101805, Nov. 2023, doi: 10.1016/J.INFFUS.2023.101805.

[8] "Integrating AI and Machine Learning into SIEM Systems |." Accessed: Dec. 16, 2024. [Online]. Available: https://securetrust.io/blog/integrating-ai-and-machine-learning-into-siem-systems/

[9] H. Alasmary *et al.*, "ShellCore: Automating Malicious IoT Software Detection by Using Shell Commands Representation," *IEEE Internet Things J*, vol. 9, no. 4, pp. 2485–2496, Mar. 2021, doi: 10.1109/JIOT.2021.3086398.

[10] S. Sambangi and L. Gondi, "A Machine Learning Approach for DDoS (Distributed Denial of Service) Attack Detection Using Multiple Linear Regression," *Proceedings 2020, Vol. 63, Page 51*, vol. 63, no. 1, p. 51, Dec. 2020, doi: 10.3390/PROCEEDINGS2020063051.

[11] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju, "Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods," *AIES 2020 - Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 180–186, Nov. 2019, doi: 10.1145/3375627.3375830.

[12] "Interpretable Machine Learning - Christoph Molnar." Accessed: Dec. 16, 2024. [Online]. Available: https://christophmolnar.com/books/interpretable-machine-learning/

[13] S. M. Lundberg and S. I. Lee, "A Unified Approach to Interpreting Model Predictions," *Adv Neural Inf Process Syst*, vol. 2017-December, pp. 4766–4775, May 2017, Accessed: Dec. 16, 2024. [Online]. Available: https://arxiv.org/abs/1705.07874v2

[14] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-August-2016, pp. 785–794, Mar. 2016, doi: 10.1145/2939672.2939785.

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal Of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2011, doi: 10.1613/jair.953.

[16] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings*, Dec. 2015, doi: 10.1109/MILCIS.2015.7348942.

[17] D. A. Rusdah and H. Murfi, "XGBoost in handling missing values for life insurance risk prediction," *SN Appl Sci*, vol. 2, no. 8, pp. 1–10, Aug. 2020, doi: 10.1007/S42452-020-3128-Y/TABLES/5.

[18] A. Suryaputra Paramita, ; Shalomeira, and V. Winata, "A Comparative Study of Feature Selection Techniques in Machine Learning for Predicting Stock Market Trends," *Journal of Applied Data Sciences*, vol. 4, no. 3, pp. 163–174, 2023.

[19] J. V. N. Ramesh, A. kushwaha, T. Sharma, A. Aranganathan, A. Gupta, and S. K. Jain, "Intelligent Feature Engineering and Feature Selection Techniques for Machine Learning Evaluation," *Lecture Notes in Networks and Systems*, vol. 915, pp. 753–764, 2024, doi: 10.1007/978-981-97-0700-3_56.

[20] X. Vasques, "Feature Engineering Techniques in Machine Learning," *Machine Learning Theory and Applications*, pp. 35–174, Feb. 2024, doi: 10.1002/9781394220649.CH2.

[21] Y. Liu, G. Shen, G. Tao, Z. Wang, S. Ma, and X. Zhang, "Complex Backdoor Detection by Symmetric Feature Differencing," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2022-June, pp. 14983–14993, 2022, doi: 10.1109/CVPR52688.2022.01458.

[22] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-August-2016, pp. 785–794, Mar. 2016, doi: 10.1145/2939672.2939785.

[23] "A Gentle Introduction to XGBoost Loss Functions - MachineLearningMastery.com." Accessed: Dec. 16, 2024. [Online]. Available: https://machinelearningmastery.com/xgboost-loss-functions/

[24] T. A. Assegie and A. Elaraby, "Optimal Tree Depth in Decision Tree Classifiers for Predicting Heart Failure Mortality," *Healthcraft Frontiers*, vol. 1, no. 1, pp. 58–66, Dec. 2023, doi: 10.56578/HF010105.

[25] C. Dombry and Y. Esstafa, "The vanishing learning rate asymptotic for linear L2-boosting," *ESAIM - Probability and Statistics*, vol. 28, pp. 227–257, 2024, doi: 10.1051/PS/2024006.

[26] "shap.plots.force — SHAP latest documentation." Accessed: Dec. 16, 2024. [Online]. Available: https://shap.readthedocs.io/en/latest/generated/shap.plots.force.html

[27] H. Ren, Y. Tang, W. Dong, S. Ren, and L. Jiang, "DUEN: Dynamic ensemble handling class imbalance in network intrusion detection," *Expert Syst Appl*, vol. 229, Nov. 2023, doi: 10.1016/J.ESWA.2023.120420.

[28] M. Hosseinzadeh, A. M. Rahmani, B. Vo, M. Bidaki, M. Masdari, and M. Zangakani, "Improving security using SVM-based anomaly detection: issues and challenges," *Soft comput*, vol. 25, no. 4, pp. 3195–3223, Feb. 2021, doi: 10.1007/S00500-020-05373-X/FIGURES/25.