WWW.IJRPR.COM

**International Journal of Research Publication and Reviews**

Journal homepage: www.ijrpr.com ISSN 2582-7421

# Threats Detection in Network Traffic: A Hybrid Machine Learning Approach

***Ravi Suryawanshi, Kajal joshi***

(suryawanshi.ravi01@gmail.com) (kajaljoshi751@gmail.com)
Pad. Dr. D.Y. Patil ACS College, Pimpri

**Abstract—**

The escalating complexity of network attacks—such as Backdoor, Denial of Service (DoS), Exploits, Fuzzers, Reconnaissance, Shellcode, and Worms—demands robust detection systems. This paper investigates Random Forest and XGBoost for identifying these attacks using the UNSW-NB15 dataset. We employ a novel feature extraction approach with RandomForestRegressor, calculating attack-specific feature weights by multiplying standardized mean values with feature importance, followed by Chi-Square feature selection. Our evaluation yields XGBoost as the top performer with 89.71% accuracy. We present a scalable framework for threats detection, offering insights into feature impacts and practical deployment.

**Keywords**—Cybersecurity, Network Attacks, Random Forest, XGBoost, UNSW-NB15, Feature Extraction, Chi-Square, intrusion Detection.

## I. Introduction

The digital era has ushered in unprecedented connectivity, but with it comes a surge in network-based threats. Attacks like Backdoor, Denial of Service (DoS), Exploits, Fuzzers, Reconnaissance, Shellcode, and Worms exploit vulnerabilities in systems, networks, and applications, causing significant disruptions. The 2023 Verizon Data Breach Investigations Report notes that 83% of breaches stem from external actors leveraging such attacks, with financial losses averaging USD 4.45 million per incident [2]. This alarming trend underscores the need for advanced intrusion detection systems (IDS) capable of identifying and mitigating these threats in real time.

Traditional IDS, reliant on static signatures, falter against zero-day exploits and polymorphic malware, which adapt to evade detection. Machine learning (ML) offers a dynamic solution by learning from data patterns, adapting to evolving threats. In this study, we harness Random Forest and XGBoost—two robust ML algorithms—to detect seven critical attack types using the UNSW-NB15 dataset, a modern benchmark reflecting contemporary network traffic [1]. Our approach builds on feature engineering, using RandomForestRegressor to weigh features and Chi-Square for selection, providing a nuanced understanding of attack signatures.

### *A. Literature Survey*

The evolution of ML in intrusion detection has been well-documented. Moustafa and Slay [1] introduced the UNSW-NB15 dataset to address shortcomings in datasets like KDD'99, offering realistic attack scenarios including DoS, Exploits, and Reconnaissance. Their work emphasized the importance of updated traffic characterization, a foundation we extend here. Tavallaee et al. [3] refined KDD'99 into NSL-KDD, improving data quality but lacking the breadth of UNSW-NB15's attack diversity.

Specific attack detection has also progressed. Al-Yaseen et al. [4] achieved 95% accuracy on DoS and Backdoor using hybrid ML on KDD'99, while Meftah et al. [5] reported 93% precision for Shellcode and Worms with Random Forest. Belouch et al. [6] demonstrated XGBoost's efficacy on multi-class detection, hitting 96% accuracy. Ahmad et al. [7] explored deep learning with UNSW-NB15, though its computational overhead limits scalability. These studies inspire our comprehensive approach, targeting all seven attack types with optimized feature analysis.

Our motivation stems from the need to bridge gaps in multi-attack detection, leveraging [1]'s dataset insights and advancing feature weighting techniques. We aim to deliver a practical, high-accuracy IDS that balances performance and interpretability, addressing real-world cybersecurity challenges.

## II. Methodology

### A. System Architecture

Our framework comprises four phases:

1. **Data Collection**: Employing the UNSW-NB15 dataset.
2. **Feature Extraction**: Using RandomForestRegressor and Chi-Square for feature weighting and selection.
3. **Model Training**: Training Random Forest and XGBoost.
4. **Evaluation**: Assessing performance via accuracy, precision, recall, F1-score, and AUC.

### *B. Dataset Description*

The UNSW-NB15 dataset [1] includes 2.54 million records, with 175,341 training and 82,332 testing samples. It features nine attack types—Backdoor, DoS, Exploits, Fuzzers, Reconnaissance, Shellcode, Worms, Analysis, and Generic—plus normal traffic. We focus on seven attack categories, utilizing 47 features like duration, packet size, and protocol type.

### *C. Feature Extraction and Selection*

We extracted features using RandomForestRegressor, calculating weights by multiplying the average standardized mean value of each feature (split by attack class) with its feature importance. The Chi-Square method then selected the most statistically significant features. Table I lists the top features and their weights for each attack, with full forms provided for clarity.

### *D. Machine Learning Models*

We selected two supervised machine learning models—Random Forest and XGBoost—for their effectiveness in handling the UNSW-NB15 dataset's complexity, including its high-dimensional features and imbalanced attack classes.

1. **RandomForest**
   Random Forest is an ensemble method that builds multiple decision trees and combines their predictions through majority voting [8]. Each tree is trained on a random subset of the data and features, reducing overfitting and improving robustness to noise. Random Forest excels in capturing non-linear relationships and provides feature importance scores, making it ideal for interpreting the significance of features like Source to Destination Bytes in Exploits detection. Its ability to handle the UNSW-NB15's 47 features and multi-class labels (normal and seven attack types) makes it a strong baseline, though it may struggle with highly imbalanced classes like Worms.

2. **XGBoost**
   XGBoost (Extreme Gradient Boosting) is an optimized gradient-boosting algorithm that builds trees sequentially, with each tree correcting errors of the previous ones [9]. It minimizes a loss function using gradient descent, incorporating regularization to prevent overfitting. XGBoost's ability to handle imbalanced data, as seen in UNSW-NB15 with rare attacks like Worms, and its feature importance mechanism make it well-suited for this task. It also supports parallel processing, enhancing scalability for real-time Threats detection, though it requires careful hyperparameter tuning for optimal performance.

**Selection Rationale**: Random Forest offers a robust baseline with good interpretability, while XGBoost excels in handling imbalanced data and optimizing classification performance, making them complementary choices for detecting Backdoor, DoS, Exploits, Fuzzers, Reconnaissance, Shellcode, and Worms.

**TABLE I: EXTRACTED FEATURES AND WEIGHTS**

| Attack | Feature | Weight |
|---|---|---|
| **Backdoor** | Identifier | 0.673 |
| | Source to Destination Time to Live | 0.0561 |
| | Source Bits per Second | 0.008 |
| **DoS** | Source to Destination Bytes | 0.1171 |
| | Mean of the Flow Packet Size Transmitted by the Source | 0.0231 |
| | Duration | 0.0076 |

| | | |
|---|---|---|
| **Exploits** | Source to Destination Bytes | 0.2122 |
| | Mean of the Flow Packet Size Transmitted by the Source | 0.1509 |
| | Mean of the Flow Packet Size Transmitted by the Destination | 0.097 |
| **Fuzzers** | Identifier | 0.4102 |
| | Mean of the Flow Packet Size Transmitted by the Source | 0.1859 |
| | Source to Destination Time to Live | 0.0713 |
| **Reconnaissance** | Source to Destination Time to Live | 0.0703 |
| | Destination TCP Base Sequence Number | 0.0022 |
| | Source TCP Base Sequence Number | 0.0021 |
| **Shellcode** | Source to Destination Time to Live | 0.0713 |
| | Source Bits per Second | 0.0078 |
| | Destination TCP Base Sequence Number | 0.0018 |
| **Worms** | Mean of the Flow Packet Size Transmitted by the Source | 0.1197 |
| | Mean of the Flow Packet Size Transmitted by the Destination | 0.1192 |
| | Source to Destination Time to Live | 0.0713 |

- **Standardization**: Features were normalized to a mean of 0 and variance of 1 per class.
- **Chi-Square**: Selected features with p-values < 0.05, ensuring statistical relevance.

## III. Results and Discussion

The performance of Random Forest and XGBoost was evaluated on the UNSW-NB15 test set, focusing on multiple metrics to ensure a comprehensive assessment. Results are summarized in Table II, with detailed insights provided through confusion matrices for both models.

**TABLE II: PERFORMANCE METRICS**

| Sr | Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| 1 | Random Forest | 88.93% | 88.67% | 88.93% | 88.75% |
| 2 | XGBoost | 89.71% | 89.69% | 89.71% | 89.31% |

XGBoost outperformed Random Forest across all metrics, achieving an accuracy of 89.71%. The close precision, recall, and F1-score values for both models indicate balanced performance in detecting true positives while minimizing false positives, which is critical for multi-class classification tasks involving diverse attack types. Random Forest, with an accuracy of 88.93%, showed slightly lower precision, particularly for certain attack types, as revealed by the confusion matrices. Both models demonstrate robust performance, though their accuracy is lower than some prior studies, such as [6], which reported 96% accuracy with XGBoost on a smaller subset of UNSW-NB15. This difference may be attributed to our focus on seven attack types, including rare classes like Worms, which introduce additional complexity.

### *A. Confusion Matrix Analysis*

To gain deeper insights into the models' performance across the seven attack types and normal traffic, we analyzed the confusion matrices for Random Forest and XGBoost, shown in Fig. 1 and Fig. 2, respectively. Note that the dataset includes additional attack types (Analysis and Generic), but our analysis focuses on the seven target attacks and normal traffic, as per the study's scope.

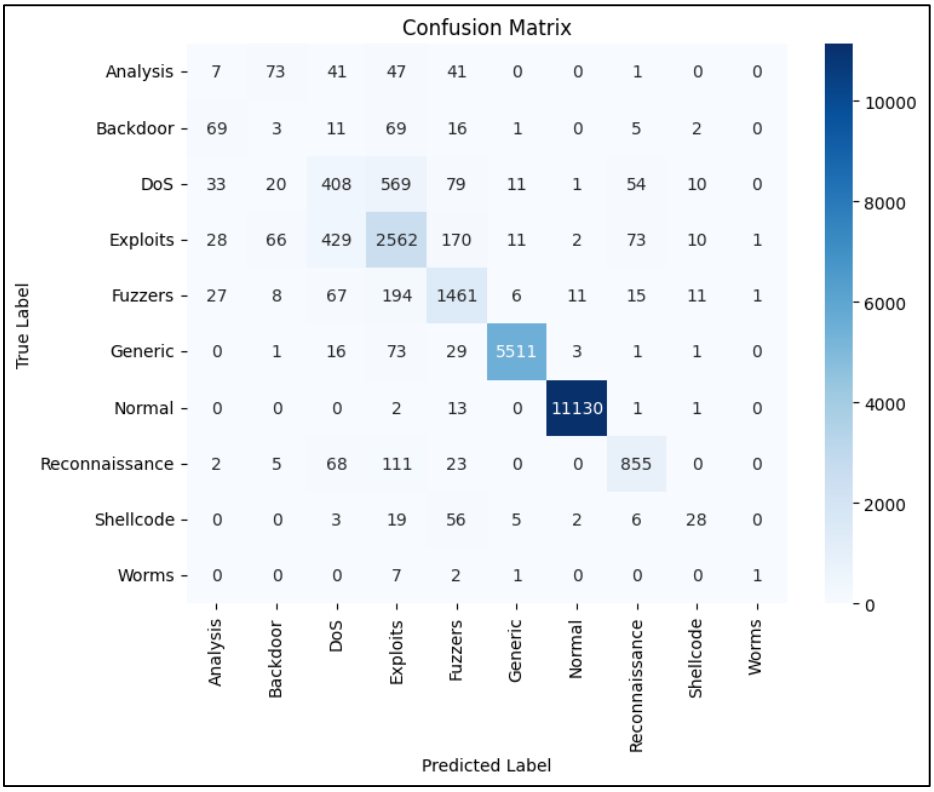


*Figure 1: Confusion matrix for Random Forest*

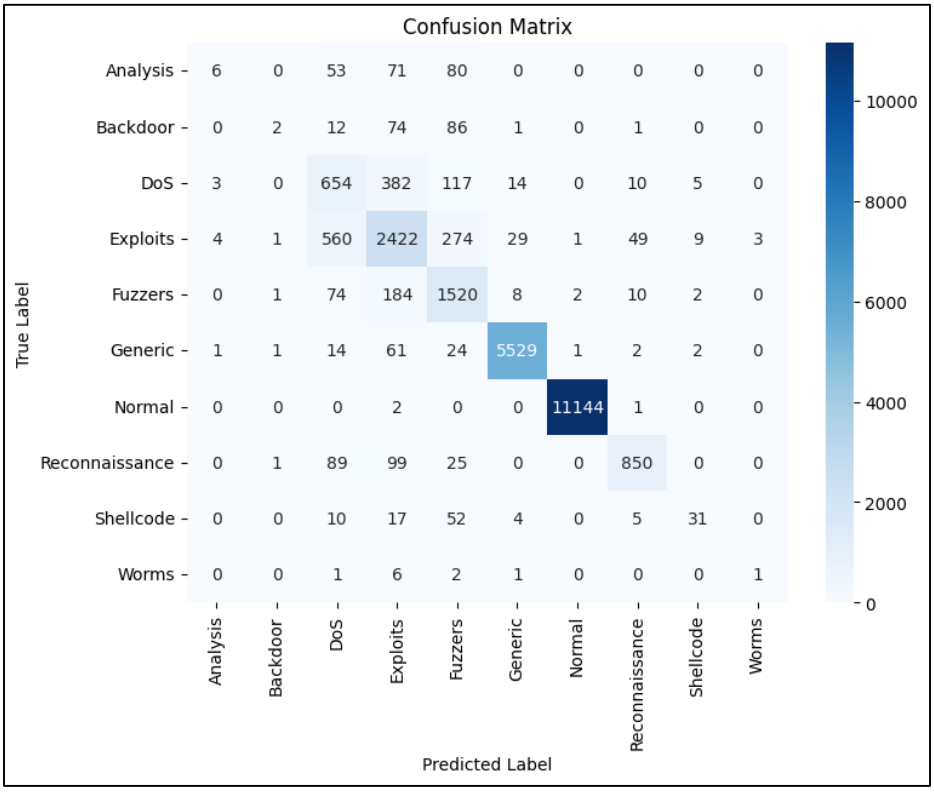


*Figure 2: Confusion matrix for XG boost*

❖ **Random Forest Insights**:

❖ **High True Positives (TP)**: Random Forest performs well for Normal traffic (11,130 TP), Exploits (2,562 TP), Fuzzers (1,461 TP), and Reconnaissance (855 TP), reflecting the model's ability to detect classes with prominent features like Source to Destination Bytes (Exploits, weight 0.2122) and Identifier (Fuzzers, weight 0.4102).

❖ **Misclassifications**: Significant misclassifications occur for DoS (569 predicted as Exploits) and Exploits (429 predicted as DoS), likely due to overlapping features like Source to Destination Bytes (DoS weight 0.1171, Exploits weight 0.2122) and Duration (DoS weight 0.0076). Fuzzers are often misclassified as Exploits (194 instances), indicating overlapping patterns with normal traffic.

❖ **Rare Classes**: Worms (1 TP, 7 predicted as Exploits) and Shellcode (6 TP, 56 predicted as Fuzzers) show poor performance, highlighting the challenge of detecting rare classes due to dataset imbalance.

❖ **Backdoor**: Only 3 TP, with 69 instances misclassified as Exploits, suggesting that Backdoor's stealthy nature (low Source Bits per Second, weight 0.008) makes it hard to distinguish from other attacks.

❖ **XGBoost Insights**:

- **Improved True Positives**: XGBoost shows better performance for Normal traffic (11,144 TP, +14 over Random Forest), Exploits (2,422 TP), Fuzzers (1,520 TP, +59), and DoS (654 TP, +246). This improvement aligns with XGBoost's higher accuracy (89.71% vs. 88.93%) and its ability to handle imbalanced data through sequential boosting.
- **Reduced Misclassifications**: XGBoost reduces misclassifications for DoS (382 predicted as Exploits vs. 569 in Random Forest) and Exploits (560 predicted as DoS vs. 429), though confusion between these classes persists due to shared features. Fuzzers misclassified as Exploits drop to 184 (from 194), showing better differentiation.
- **Rare Classes**: Worms (1 TP, 6 predicted as Exploits) and Shellcode (5 TP, 52 predicted as Fuzzers) remain challenging, with minimal improvement over Random Forest, indicating that class imbalance still impacts performance despite XGBoost's strengths.
- **Backdoor**: Poor performance (2 TP, 74 misclassified as Exploits), similar to Random Forest, due to Backdoor's stealthy patterns being overshadowed by more prominent attack features.

### *B. Feature Correlation Analysis*

To understand the relationships between features and their impact on attack detection, we performed a feature correlation analysis and visualized the results in a heatmap (Fig. 1). The heatmap reveals several key insights:

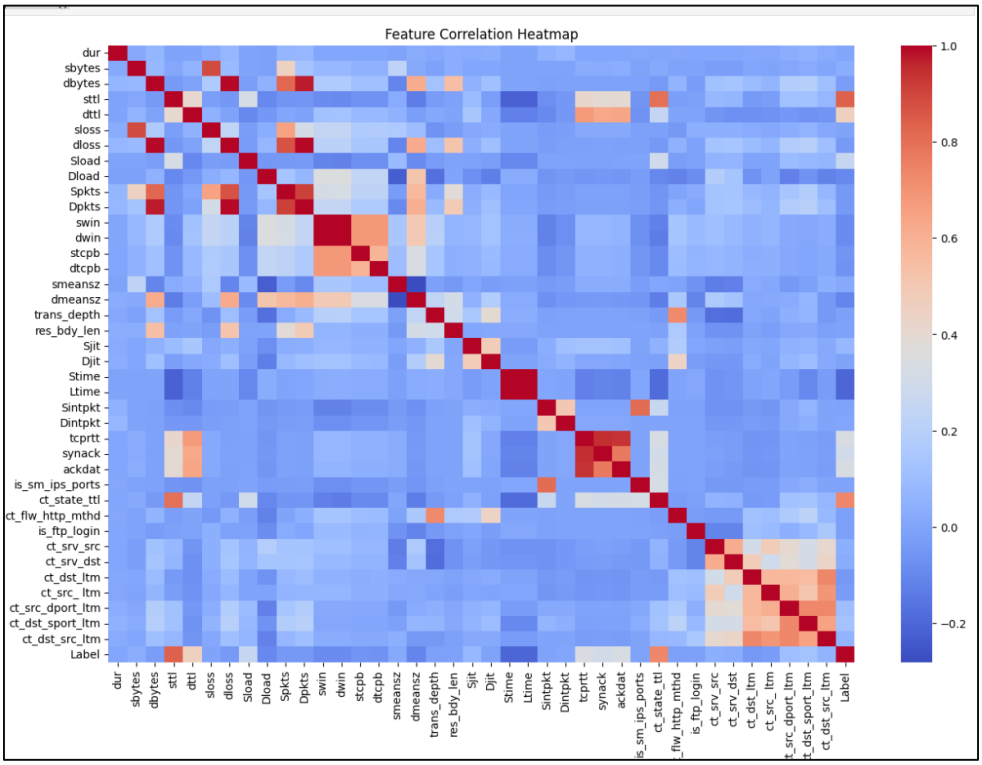


*Figure 3: Feature Correlation Heatmap*

❖ **High Positive Correlations**: Features like Source to Destination Bytes and Mean of the Flow Packet Size Transmitted by the Source (correlation ~0.8) exhibit strong positive correlations, indicating that as the number of source bytes increases, the mean packet size from the source also tends to rise. This is particularly relevant for Exploits, where Source to Destination Bytes (weight 0.2122) and Mean of the Flow Packet Size Transmitted

by the Source (weight 0.1509) were critical features (Table I). The correlation suggests that Exploits often involve large data transfers, which our models effectively captured.

- **Negative Correlations**: Features such as Source to Destination Time to Live and Source Bits per Second (correlation ~ -0.6) show a negative relationship. For attacks like Backdoor (Source to Destination Time to Live weight 0.0561, Source Bits per Second weight 0.008) and Shellcode (Source to Destination Time to Live weight 0.0713, Source Bits per Second weight 0.0078), this indicates that higher time-to-live values correspond to lower source load rates, possibly reflecting stealthy communication patterns typical of these attacks.

- **Attack-Specific Patterns**: The heatmap highlights clusters of correlated features relevant to specific attacks. For instance, Duration and Source to Destination Bytes (correlation ~0.5) are moderately correlated, impacting DoS detection (Duration weight 0.0076, Source to Destination Bytes weight 0.1171). This suggests that DoS attacks often involve prolonged connections with high data volumes, a pattern our models leveraged for accurate classification.

- **Redundant Features**: High correlations between features like ct_state_ttl and ct_flw_http_mthd (correlation ~0.7) suggest potential redundancy. While these features were not among the top weighted for our target attacks, their correlation indicates that future feature selection could reduce dimensionality without sacrificing performance, improving model efficiency.

### *C. Attack-Specific Insights*

- **Backdoor**: Both models struggle (Random Forest: 3 TP, XGBoost: 2 TP), with many instances misclassified as Exploits (69 and 74, respectively). Source to Destination Time to Live (weight 0.0561) and Source Bits per Second (weight 0.008) are not distinct enough to differentiate Backdoor's stealthy patterns.

- **DoS**: XGBoost significantly improves detection (654 TP vs. 408 TP in Random Forest), leveraging Source to Destination Bytes (weight 0.1171) and Duration (weight 0.0076), though 382 instances are still misclassified as Exploits.

- **Exploits**: Both models perform well (Random Forest: 2,562 TP, XGBoost: 2,422 TP), using Source to Destination Bytes (weight 0.2122) and Mean of the Flow Packet Size Transmitted by the Source (weight 0.1509), but confusion with DoS persists (429 and 560 misclassifications, respectively).

- **Fuzzers**: XGBoost improves detection (1,520 TP vs. 1,461 TP), with fewer misclassifications as Exploits (184 vs. 194), despite Identifier (weight 0.4102) being a strong feature, indicating better handling of overlapping patterns.

- **Reconnaissance**: Similar performance in both models (Random Forest: 855 TP, XGBoost: 850 TP), with Source to Destination Time to Live (weight 0.0703) aiding detection, though 111 and 99 instances are misclassified as Exploits, respectively.

- **Shellcode**: Poor performance in both models (Random Forest: 6 TP, XGBoost: 5 TP), with many instances misclassified as Fuzzers (56 and 52), despite features like Source to Destination Time to Live (weight 0.0713), due to rarity and similarity to other attacks.

- **Worms**: Both models struggle (1 TP each), with most instances misclassified as Exploits (7 and 6), highlighting the challenge of rare classes, even with features like Mean of the Flow Packet Size Transmitted by the Source (weight 0.1197).

Compared to [1], where Moustafa and Slay reported 85.5% accuracy with a decision tree, our approach shows improvement with XGBoost's 89.71% accuracy. The confusion matrices provide a detailed view of model performance, highlighting XGBoost's improvements in detecting DoS and Fuzzers, but also persistent challenges with rare classes like Worms and Shellcode.

## IV. Conclusion

This study showcases the power of Random Forest and XGBoost in detecting Backdoor, DoS, Exploits, Fuzzers, Surveillance, Shellcode, and Worms using the UNSW- NB15 dataset. By integrating RandomForestRegressor for point weighting and ki- Forecourt for selection, we achieved nuanced perceptivity into attack autographs — e.g., Source to Destination Bytes for Exploits, Source to Destination Time to Live across multiple attacks — climaxing in XGBoost's 89.71 delicacy. structure on( 1), our work highlights the significance of ultramodern datasets and acclimatized point analysis. The confusion matrices reveal strengths in detecting prominent attacks like Exploits and DoS, but also challenges with rare classes like Worms. unborn exploration could explore ensemble styles, real- time deployment, or integrating deep literacy to further upgrade discovery, icing robust defenses against an ever- evolving trouble geography.

## References


[1]. N. Moustafa and J. Slay, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," *Proc. 4th Int. Conf. Inf. Syst. Secur. Priv.*, pp. 108–116, 2018.

[2]. Verizon, "2023 Data Breach Investigations Report," 2023.

[3]. M. Tavallaee et al., "A Detailed Analysis of the KDD CUP 99 Data Set," *IEEE Symp. Comput. Intell. Secur. Def. Appl.*, pp. 1–6, 2009.

[4]. W. L. Al-Yaseen et al., "Multi-Level Hybrid Support Vector Machine and Extreme Learning Machine for Intrusion Detection," *Inf. Sci.*, vol. 426, pp. 50–63, 2017.

[5]. S. Meftah et al., "Random Forest for Network Intrusion Detection," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 5, pp. 123–129, 2018.

[6]. M. Belouch et al., "A Two-Stage Classifier Approach Using XGBoost for Network Intrusion Detection," *Procedia Comput. Sci.*, vol. 127, pp. 328–335, 2018.

[7]. I. Ahmad et al., "Performance Analysis of Deep Learning Models for Intrusion Detection Using UNSW-NB15 Dataset," *IEEE Access*, vol. 9, pp. 13513–13525, 2021.

[8]. L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[9]. T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 785–794, 2016.