Explainable and Trustworthy Machine Learning for Network Intrusion Detection

Mrs. Radhika. S[1], Selvaboopathi S[2], Jai Krishna V[3], Muhilan S[4], Kumaran V[5]
[1]*Assistant Professor, Department of CSE, Dhanalakshmi Srinivasan University, Trichy, India*
[2,3,4]*4th Year, Department of Information Technology, Dhanalakshmi Srinivasan University, Trichy, India*
[5]*Department of Artificial Intelligence and Data Science,*
Dhanalakshmi Srinivasan University, Trichy, India

***Abstract*—Modern network environments face increasingly sophisticated cyber threats, requiring intrusion detection systems that are both accurate and interpretable. This study proposes a leakage-aware and explainable machine learning framework for network intrusion detection using the UNSW-NB15 dataset. The framework integrates a Random Forest classifier with SHapley Additive exPlanations (SHAP) to achieve high predictive performance and transparent decision-making. A structured preprocessing pipeline involving duplicate removal, missing-value handling, leakage-prone feature elimination, categorical encod-ing, and feature alignment was employed to ensure experimental reliability and reproducibility.**

A comparative evaluation was conducted using Logistic Re-gression, Decision Tree, K-Nearest Neighbors, Random For-est, and XGBoost classifiers. Experimental results show that ensemble-based models outperform conventional approaches in intrusion detection tasks. The proposed Random Forest model achieved 97.60% accuracy, a 0.9781 F1-score, and a ROC-AUC score of 0.9970, while also demonstrating strong generalization on unseen testing data. SHAP-based analysis identified influ-ential network traffic features contributing to malicious traffic detection, improving model transparency and analyst trust. The findings demonstrate that combining ensemble learning with explainable AI provides reliable and interpretable intrusion detection for modern cybersecurity systems.

***Index Terms*—Network Intrusion Detection System (NIDS), Explainable Artificial Intelligence (XAI), Random Forest, SHAP, UNSW-NB15, Cybersecurity Analytics, Machine Learn-ing, Leakage-Aware Preprocessing**

I. INTRODUCTION

The rapid growth of digital networks, cloud platforms, and Internet-connected systems has increased the frequency and complexity of cyberattacks. Modern threats such as denial-of-service attacks, malware propagation, reconnaissance ac-tivities, and unauthorized access continue to challenge tradi-tional rule-based security mechanisms. As a result, Network Intrusion Detection Systems (NIDS) have become essential for identifying malicious network activities and protecting critical infrastructures.

Machine learning (ML)-based intrusion detection systems have gained significant attention because of their ability to learn complex traffic patterns and detect anomalies automatically. Unlike signature-based methods, ML models can generalize to previously unseen attacks. However, many existing ML-based IDS frameworks operate as black-box systems, limiting transparency and reducing analyst trust in operational cybersecurity environments.

Another major challenge in cybersecurity research is data leakage during preprocessing and model training. Leakage-prone attributes may unintentionally expose target information to the model, leading to unrealistic performance and poor generalization. Therefore, leakage-aware preprocessing is nec-Essary to ensure reliable and reproducible intrusion detection experiments.

To address these issues, this study proposes a leakage-aware and explainable intrusion detection framework using Random Forest (RF) and SHapley Additive explanations (SHAP) on the UNSW-NB15 dataset. The preprocessing pipeline includes duplicate removal, missing value handling, elimination of

leakage-related attributes (attack_cat and id), one-hot encoding, and feature alignment to maintain experimental consistency.

Random Forest was selected as the primary classifier because of its robustness, strong performance on high-dimensional tabular data, and ability to model non-linear rela-tionships in network traffic features. Comparative experiments were conducted using Logistic Regression, Decision Tree, K-Nearest Neighbors (KNN), Random Forest, and XGBoost classifiers. Experimental results showed that Random Forest achieved the best performance with 97.60% accuracy and a ROC-AUC score of 0.9970.

To improve interpretability, SHAP-based explainability analysis was integrated into the framework using SHAP TreeExplainer. SHAP identifies the contribution of individual traffic features toward malicious traffic predictions, improving transparency and supporting cybersecurity analysts in threat investigation and decision-making.

The major contributions of this work are summarized as follows:

1) A leakage-aware preprocessing framework for reliable intrusion detection experimentation on the UNSW-NB15
2) dataset.
3) Comparative evaluation of multiple machine learning classifiers for binary intrusion detection.
4) An explainable AI-based intrusion detection framework integrating Random Forest with SHAP interpretability.
5) Comprehensive evaluation using Accuracy, Precision, Recall, F1-score, ROC-AUC, confusion matrices, and testing-set validation.
6) An interpretable cybersecurity-oriented analysis to sup-port trustworthy AI-assisted intrusion detection systems.

II. RELATED WORK

Network Intrusion Detection Systems (NIDS) are essential for detecting malicious network activities and protecting mod-ern digital infrastructures. Traditional signature-based intru-sion detection methods are effective against known attacks but struggle to detect zero-day threats and evolving attack patterns. These limitations have encouraged the adoption of machine learning (ML)-based intrusion detection approaches.

Recent studies have explored various ML algorithms for intrusion detection, including Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees, Support Vector Machines (SVM), Random Forest, and deep learning models. Among these, ensemble learning methods such as Random Forest and XGBoost have shown strong performance due to their ability to model complex non-linear traffic behaviors and handle high-dimensional network features effectively curity because of its robustness, scalability, and reduced over-fitting sensitivity compared to traditional classifiers. Previous studies using datasets such as NSL-KDD, CICIDS2017, and UNSW-NB15 reported improved intrusion detection accuracy with ensemble-based methods. However, many existing works primarily focus on predictive performance while giving limited attention to explainability, reproducibility, and leakage-aware experimentation.

To improve transparency in AI-driven cybersecurity sys-tems, explainable artificial intelligence (XAI) techniques such as LIME and SHAP have recently been integrated into in-trusion detection research. Among these methods, SHAP has become popular because it provides both local and global fea-ture importance analysis based on game-theoretic principles. SHAP-based explainability helps cybersecurity analysts under-stand which network traffic features contribute to malicious traffic predictions.

Despite these advances, several intrusion detection studies still suffer from methodological limitations, including leakage-prone features, incomplete preprocessing transparency, and insufficient comparative evaluation across multiple classifiers. Such issues can produce unrealistic performance results and reduce the reliability of intrusion detection frameworks.

To address these limitations, this work proposes a leakage-aware explainable intrusion detection framework using Ran-dom Forest and SHAP on the UNSW-NB15 dataset. The framework removes leakage-related attributes, applies repro-ducible preprocessing procedures, and performs comparative evaluation using Logistic Regression, Decision Tree, KNN, Random Forest, and XGBoost under identical experimental conditions. In addition, SHAP-based explainability analysis is incorporated to improve transparency, interpretability, and trustworthiness in AI-assisted cybersecurity intrusion detection systems

III. DATASET AND PREPROCESSING

A. UNSW-NB15 Dataset

This study used the UNSW-NB15 benchmark dataset de-veloped by the Australian Centre for Cyber Security (ACCS) for modern intrusion detection research. The dataset contains both normal and malicious network traffic generated in a realistic hybrid testbed environment. It includes various attack categories such as DoS, exploits, reconnaissance, fuzzers, worms, and backdoors.

The intrusion detection problem was formulated as a binary classification task where benign traffic was labeled as class 0 and malicious traffic as class 1:

$$y = \begin{cases} 0, x = 0 \\ 1, x \neq 0 \end{cases}$$

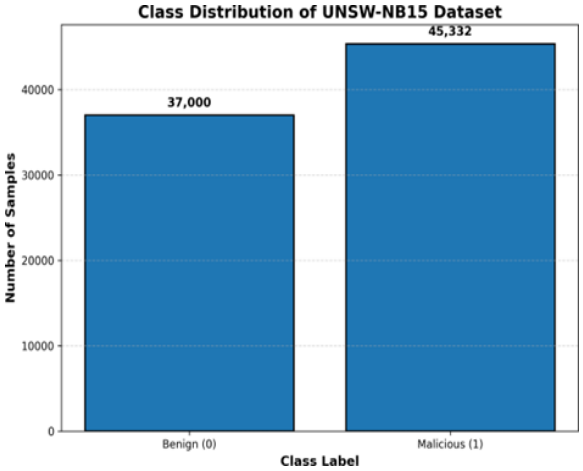

Fig. 1. Class distribution of the UNSW-NB15 dataset.

Table I Unsw-Nb15 Dataset Summary

Description	Value
Total Records	82,332
Normal Traffic	37,000
Attack Traffic	45,332
Original Features	45
Encoded Features	187
Classification Type	Binary

B. Data Preprocessing

A structured preprocessing pipeline was applied to ensure reliable and reproducible intrusion detection experiments

1) Duplicate Removal: Duplicate records were removed using drop_duplicates() to reduce redundancy and im-prove model generalization.
2) Missing Value Handling: Missing values were replaced with zero:
3) Duplicate Removal: Duplicate records were removed using drop_duplicates() to reduce redundancy and im-prove model generalization.
4) Missing Value Handling: Missing values were replaced with zero:

B. Training and Testing Strategy

After preprocessing, the dataset was divided into training and testing subsets using an 80:20 split with stratified sampling to preserve class distribution.

$$x_i = \begin{cases} 0, & \text{if } x_i \text{ is missing} \\ x_i, & \text{otherwise} \end{cases} \quad (2)$$

1) Leakage-Aware Feature Removal: To prevent data leak-age, the attributes attack_cat and id were removed before training:

$$X_{clean} = X - \{attack\_cat, id\} \quad (3)$$

This ensured that the model learned meaningful network traffic patterns instead of leakage-related information.

C. Categorical Encoding

Categorical attributes were converted into numerical form using one-hot encoding with pd.get_dummies(drop_first=True):

$$C = \{c1, c2, ..., ck\} \rightarrow \{b1, b2, ..., bk-1\} \quad (4)$$

After encoding, the dataset contained 187 features.

D. Feature Alignment

Feature alignment was performed to ensure consistency between training and testing datasets:

$$X_{test}^{aligned} = X_{test}[F_{train}] \quad (5)$$

where Ftrain represents the feature set used during training.

E. Reproducibility

To ensure reproducibility, all experiments used a fixed random seed (random_state = 42) during data splitting, model training, and SHAP analysis. The same preprocessing pipeline was applied consistently across training and testing datasets.

IV. METHODOLOGY

A. Experimental Framework

This study proposes a leakage-aware explainable intrusion detection framework using machine learning and SHAP-based explainability on the UNSW-NB15 dataset. The workflow consists of data preprocessing, feature encoding, baseline model comparison, Random Forest classification, and SHAP interpretation.

The intrusion detection problem was formulated as binary classification, where normal traffic was labeled as 0 and

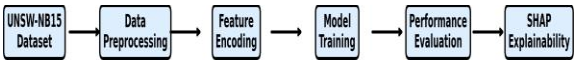

Fig. 2. Leakage-aware explainable intrusion detection framework.

B. Training and Testing Strategy

After preprocessing, the dataset was divided into training and testing subsets using an 80:20 split with stratified sampling to preserve class distribution

$$D = D_{train} \cup D_{test}$$

where Dtrain and Dtest denote the training and testing datasets respectively.

The experiments used:

- test_size = 0.2
- random_state = 42

The final encoded feature space contained 187 features.

C. Baseline Machine Learning Models

To evaluate the proposed framework, several machine learn-ing models were compared under identical preprocessing con-ditions:

- Logistic Regression (LR)
- Decision Tree (DT)
- K-Nearest Neighbors (KNN)
- Random Forest (RF)
- XGBoost (XGB)

These models represent linear, distance-based, tree-based, and ensemble learning approaches for intrusion detection.

D. Random Forest-Based Intrusion Detection

Random Forest was selected as the primary classifier be-because of its robustness, scalability, and ability to model com-plex non-linear traffic patterns.

The ensemble prediction is defined as:

$$\hat{y} = \text{mode}\{h_1(x), h_2(x), \ldots, h_n(x)\}$$

where hi(x) represents the prediction of the ith decision tree.

The model was configured using:

- n_estimators = 100
- random_state = 42
- n_jobs = -1

The prediction probability was computed as:

$$P(y = 1|x) = \frac{1}{n}\sum_{i=1}^{n} P_i(y = 1|x)$$

The final prediction threshold was

$$\hat{y} = \begin{cases} 1, & P(y=1|x) \geq 0.5 \\ 0, & P(y=1|x) < 0.5 \end{cases}$$

E. Performance Evaluation

The proposed framework was evaluated using standard cybersecurity classification metrics:

- Accuracy
- Precision
- Recall
- F1-score
- ROC-AUC
- Confusion Matrix
- ROC Curve

These metrics provide comprehensive evaluation of intru-sion detection effectiveness and generalization performance.

F. SHAP-Based Explainability

To improve transparency, SHAP (SHapley Additive ex-Planations) was integrated into the framework using TreeExplainer for Random Forest.

The SHAP explanation model is expressed as:

$$f(x) = \phi_0 + \sum_{i=1}^{M} \phi_i$$

where ϕi represents the contribution of feature i to the prediction.

The explainability framework included:

- SHAP TreeExplainer
- SHAP summary plots
- Feature importance analysis

A Subset Of 500 Testing Samples Was Used for SHAP Com-Putation to Reduce Computational Overhead While Preserving Representative Explanations.
The SHAP-Based Analysis Helps Cybersecurity Analysts Un-Derstand Important Traffic Features Influencing Malicious Traffic Detection and Improves Trust In AI-Driven Intrusion Detection Systems.

V. EXPERIMENTAL SETUP

A. Software Environment

The proposed framework was implemented in Python using widely adopted machine learning and scientific computing libraries, including:

- Pandas And Numpy for Preprocessing,
- Scikit-Learn for Machine Learning,
- Xgboost For Gradient Boosting,
- Matplotlib And Seaborn for Visualization,
- Shap For Explainability Analysis.

All experiments were executed within a Python virtual environment to ensure reproducibility.

B. Preprocessing Pipeline

The UNSW-NB15 dataset was processed using a leakage-aware preprocessing pipeline consisting of:

1) duplicate removal,
2) missing value handling using fillna(0),
3) binary label mapping,
4) removal of leakage attributes (attack_cat, id),
5) one-hot encoding using pd.get_dummies (). After encoding, the dataset contained 187 features.

C. Training and Testing Configuration

The dataset was divided into training and testing subsets using an 80:20 split with stratified sampling.
The experimental configuration included:

- test_size = 0.2
- random_state = 42

The trained Random Forest model was further evaluated on the independent UNSW-NB15 testing dataset to assess generalization capability.

D. Machine Learning Models

Five machine learning models were evaluated:

- Logistic Regression
- Decision Tree
- K-Nearest Neighbors (KNN)
- Random Forest
- XGBoost

The Random Forest classifier was selected as the proposed model due to its strong performance, robustness to noisy data, and ability to handle high-dimensional network traffic features.

Table II. Machine Learning Model Hyperparameters

Model	Hyperparameters
Logistic Regression	max_iter=1000
Decision Tree	random_state=42
KNN	n_neighbors=5
Random Forest	n_estimators=100
XGBoost	learning_rate=0.1

E. Evaluation Metrics

The models were evaluated using:

- Accuracy
- Precision
- Recall
- F1-score
- ROC-AUC
- Confusion Matrix
- ROC Curve

Accuracy was computed as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision and Recall were calculated using:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

The F1-score was computed as:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

F. ROC-AUC Analysis

ROC curves were generated to evaluate classifier discrimination capability across multiple thresholds. The proposed Random Forest model achieved an AUC score of 0.9970, indicating strong intrusion detection performance.

G. SHAP Explainability

SHAP explainability analysis was performed using shap. Tree Explainer to interpret feature contributions toward malicious traffic prediction.

The SHAP explanation model is defined as:

$$f(x) = \phi_0 + \sum_{i=1} \phi_i$$

Where Φi Represents the Contribution of Feature I.

A Subset Of 500 Testing Samples Was Used for Shap Analysis to Reduce Computational Overhead While Maintaining Representative Explanations.

Shap Summary Plots Were Generated to Visualize Important Network Traffic Features Influencing Intrusion Predictions

VI. RESULTS AND DISCUSSION

A. Comparative Performance Analysis

Multiple machine learning models, including Logistic Re-gression, Decision Tree, KNN, Random Forest, and XGBoost, were evaluated on the UNSW-NB15 dataset for binary intru-sion detection. The results show that ensemble learning meth-ods significantly outperform conventional linear and distance-based classifiers.

Among all models, the proposed Random Forest (RF) classifier achieved the best overall performance with:

- Accuracy: 97.60%
- Precision: 0.9828
- Recall: 0.9733
- F1-score: 0.9781
- ROC-AUC: 0.9970

XGBoost also achieved strong results with 97.33% accuracy and 0.9965 AUC, while Logistic Regression produced the lowest performance due to limited capability in modeling complex non-linear network traffic patterns.

B. Random Forest Performance

The Random Forest classifier achieved the highest detection performance due to its ensemble learning capability, robust-ness against noisy features, and ability to model complex non-linear traffic behavior. The model effectively handled the high-dimensional encoded feature space and demonstrated excellent discrimination capability with an AUC score of 0.9970.

Compared with single-tree and linear models, Random Forest reduced overfitting and improved generalization by aggregating predictions from multiple decision trees.

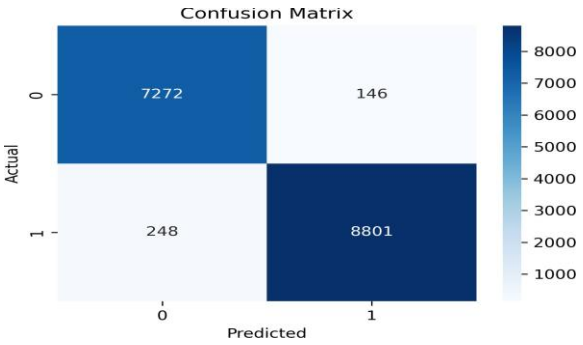

Fig. 3. Confusion matrix of the proposed Random Forest model.

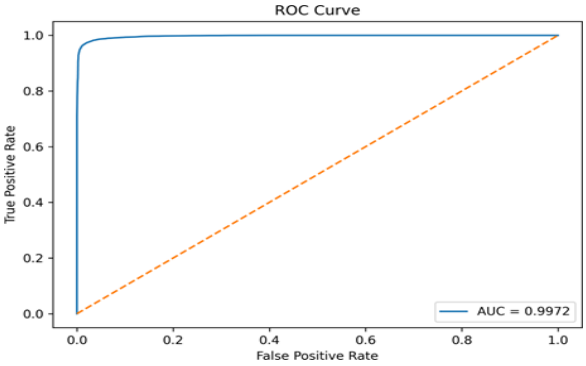

Fig. 4. ROC curve of the proposed Random Forest model.

C. Testing Dataset Evaluation

The trained Random Forest model was further evaluated on an independent testing dataset to assess generalization capability. The model achieved 90% testing accuracy and a weighted F1-score of 0.91, indicating strong robustness on unseen network traffic.

Table IV Independent Testing Dataset Evaluation

Metric	Value
Testing Accuracy	90%
Weighted Precision	0.92
Weighted Recall	0.90
Weighted F1-Score	0.91

D. ROC-AUC and Confusion Matrix Analysis

ROC analysis demonstrated strong separation between be-nign and malicious traffic classes. Random Forest achieved the highest ROC-AUC value, indicating reliable classification capability across different thresholds.

Table III. Comparative Performance of Machine Learning Models

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	75.28%	0.8445	0.6754	0.7505	0.8223
Decision Tree	96.42%	0.9666	0.9683	0.9675	0.9637
KNN	80.47%	0.8286	0.8136	0.8210	0.9083
Random Forest	97.60%	0.9828	0.9733	0.9781	0.9970
XGBoost	97.33%	0.9825	0.9687	0.9756	0.9965

Confusion matrix analysis further showed low false-positive and false-negative rates. Reducing false negatives is especially important in cybersecurity because undetected attacks may compromise network infrastructure, while lower false positives help reduce alert fatigue for analysts.

Table V. Random Forest Validation Confusion Matrix

Actual	Predicted	
	Benign	Malicious
Benign	7246	154
Malicious	242	8825

E. SHAP Explainability Analysis

To improve transparency, SHAP-based explainability analy-sis was integrated with the Random Forest model. SHAP iden-tified the most influential network traffic features contributing to malicious traffic predictions and provided both global and local interpretability.

The explainability framework supports:

- identification of important attack indicators,
- interpretation of model decisions,
- validation of prediction behavior,
- improved trust in AI-driven intrusion detection systems. The results demonstrate that combining leakage-aware pre-processing, ensemble learning, and explainable AI provides an accurate and interpretable intrusion detection framework suitable for modern cybersecurity applications.

VII. EXPLAINABILITY ANALYSIS USING SHAP

A. Importance of Explainability

Machine learning-based intrusion detection systems often operate as black-box models, limiting transparency in cyber-security decision-making. In practical environments, security analysts require interpretable explanations to understand why network traffic is classified as malicious.

To improve transparency and trustworthiness, the proposed framework integrates SHAP (SHapley Additive exPlanations) with the Random Forest intrusion detection model.

B. SHAP Methodology

SHAP is a game-theoretic explainability approach that mea-sures the contribution of each feature to the model prediction. The SHAP explanation model is expressed as:

$$f(x) = \phi_0 + \sum_{i=1}^{M} \phi_i$$

where:

- $f(x)$ represents the prediction,
- ϕ_0 is the base prediction,
- ϕ_i denotes the contribution of feature i,
- M represents the total number of features.

SHAP provides both global and local interpretability, making it suitable for cybersecurity analytics.

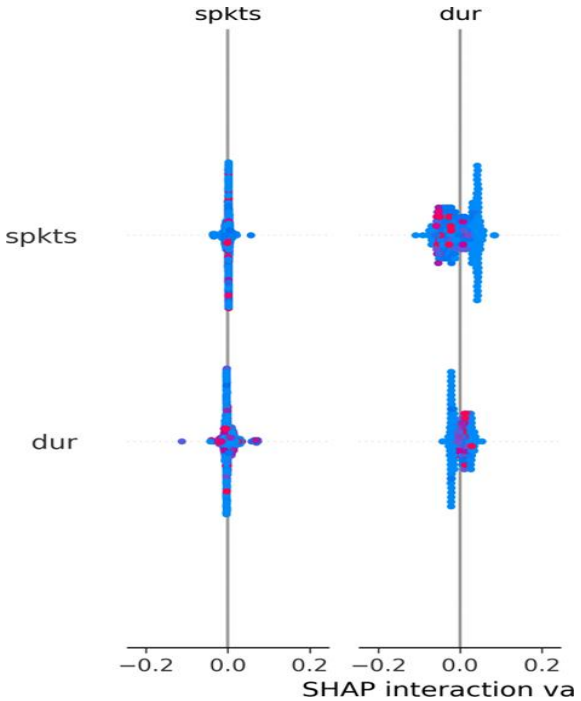

Fig. 5. SHAP summary plot of important network traffic features.

C. TreeExplainer Analysis

The explainability framework used
shap. TreeExplainer for interpreting the Random Forest model. To reduce computational complexity, SHAP analysis was performed on a subset of 500 testing samples.

The SHAP summary plot identified important traffic at-tributes influencing malicious traffic detection. Features with positive SHAP values increased the probability of attack clas-sification, while negative values supported benign predictions.

The analysis showed that the model learned meaningful cybersecurity-related behaviors, including:

- abnormal traffic patterns,
- protocol-level behaviors,
- service interaction anomalies,
- irregular connection characteristics.

D. Cybersecurity Interpretability

Explainability improves analyst trust and supports practical deployment of AI-driven intrusion detection systems. SHAP-based interpretation enables:

- understanding of model decisions,
- identification of important attack indicators,
- improved threat investigation,
- transparent intrusion analysis.

Unlike traditional black-box models, the proposed frame-work provides interpretable feature-level explanations that support operational cybersecurity analysis.

E. Leakage-Aware Explainability

Leakage-related attributes such as attack_cat and id were removed before training to prevent unrealistic learning behavior. SHAP analysis confirmed that the model relied on meaningful network traffic features rather than leakage-prone information.

This improves:

- experimental validity,
- model robustness,
- reproducibility,
- operational reliability.

F. Overall Findings

The integration of Random Forest and SHAP provides both high predictive performance and interpretable intrusion detection analysis. The explainability results demonstrate that the proposed framework captures meaningful attack behaviors while maintaining transparency and trustworthiness for cyber-security applications.

VIII. CONCLUSION

This study proposed a leakage-aware explainable intru-sion detection framework using Random Forest and SHAP on the UNSW-NB15 dataset. The framework incorporated preprocessing techniques such as duplicate removal, missing value handling, leakage feature elimination, one-hot encoding, and feature alignment to ensure reliable and reproducible experimentation.

Five machine learning models, including Logistic Regression, Decision Tree, KNN, Random Forest, and XGBoost,
were comparatively evaluated using Accuracy, Precision, Re-call, F1-score, ROC-AUC, and confusion matrix analysis. Among all models, Random Forest achieved the best overall performance with:

- Accuracy: 97.60%
- F1-score: 0.9781
- ROC-AUC: 0.9970

The model also demonstrated strong generalization capability on an independent testing dataset with 90% testing accuracy and a weighted F1-score of 0.91.

The results showed that ensemble learning methods out-perform conventional classifiers in handling high-dimensional network traffic data and complex attack behaviors. In addition, SHAP-based explainability analysis improved transparency by identifying important network traffic features influencing intrusion detection decisions.

The study further highlighted the importance of leakage-aware experimentation by removing leakage-prone attributes such as attack cat and id, thereby improving model reliability and preventing artificial performance inflation.

Overall, the proposed framework demonstrates that com-bining leakage-aware preprocessing, ensemble learning, and explainable artificial intelligence provides an accurate, in-treatable, and operationally reliable solution for modern network intrusion detection systems.

IX. FUTURE WORK

Future research can extend the proposed framework using advanced deep learning models such as LSTM, GRU, CNN, and Transformer-based architectures for capturing temporal and sequential attack behaviors in network traffic. Real-time deployment in Security Operations Center (SOC) environ-ments using streaming analysis and low-latency inference is another important direction.

Online learning and adaptive intrusion detection methods may help address concept drift and evolving cyber threats. Federated learning can also support privacy-preserving col-laborative intrusion detection across distributed organizations. From an explainability perspective, future work may com-bine SHAP with other XAI methods such as LIME and attention-based techniques to improve interpretability and an-alyst trust. Additionally, evaluating the framework against adversarial attacks and extending it toward multi-class intru-sion detection and cross-dataset validation can further improve robustness and real-world applicability.

REFERENCES

[1] E. Denning, "An Intrusion-Detection Model," IEEE Trans. Softw. Eng., vol. SE-13, no. 2, pp. 222–232, Feb. 1987.

[2] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in Proc. IEEE Symp. Security and Privacy, Oakland, CA, USA, 2010, pp. 305–316.

[3] J. Liao, C. H. R. Lin, Y. C. Lin, and K. Y. Tung, "Intrusion Detection System: A Comprehensive Review," J. Netw. Comput. Appl., vol. 36, no. 1, pp. 16–24, 2013.

[4] [4] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)," in Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS), Canberra, Australia, 2015, pp. 1–6.

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

[6] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, San Francisco, CA, USA, 2016, pp. 785–794.

[7] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 30, 2017, pp. 4765–4774.

[8] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You?: Explaining the Predictions of Any Classifier," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, pp. 1135–1144.

[9] J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," in Proc. Int. Conf. Learn. Representations (ICLR), 2015.

[10] S. M. Kasongo and Y. Sun, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset," J. Big Data, vol. 7, no. 105, pp. 1–20, 2020.

[11]] M. Ahmad, Q. Riaz, M. Zeeshan, H. Tahir, S. A. Haider, and M. S. Khan, "Intrusion Detection in Internet of Things Using Supervised Machine Learning Based on Application and Transport Layer Features Using UNSW-NB15 Dataset," EURASIP J. Wireless Commun. Netw., vol. 2021, no. 10, pp. 1–23, 2021.

[12] Y. Yin, J. Jang-Jaccard, W. Xu, A. Singh, J. Zhu, F. Sabrina, and J. Kwak, "IGRF-RFE: A Hybrid Feature Selection Method for MLP-Based Network Intrusion Detection on UNSW-NB15 Dataset," J. Big Data, vol. 10, no. 15, pp. 1–27, 2023.

[13] M. Zeeshan, Q. Riaz, M. A. Bilal, M. K. Shahzad, H. Jabeen, S. A. Haider, and A. Rahim, "Protocol-Based Deep Intrusion Detection for DoS and DDoS Attacks Using UNSW-NB15 and Bot-IoT Datasets," IEEE Access, vol. 10, pp. 2269–2283, 2022.

[14] Z. Zoghi and G. Serpen, "UNSW-NB15 Computer Security Dataset: Analysis Through Visualization," arXiv preprint arXiv:2101.05067, 2021.

[15] N. Thanh and T. V. Lang, "Use the Ensemble Methods When Detecting DoS Attacks in Network Intrusion Detection Systems," EAI Endorsed Trans. Context-Aware Syst. Appl., vol. 6, no. 19, pp. 1–10, 2019.

[16] L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," IEEE Commun. Surveys Tuts., vol. 18, no. 2, pp. 1153–1176, 2015.

[17] [17] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A Deep Learning Approach for Network Intrusion Detection System," in Proc. 9th EAI Int. Conf. Bio-inspired Inf. Commun. Technol., 2016, pp. 21–26.

[18] Kim, S. Lee, and S. Kim, "A Novel Hybrid Intrusion Detection Method Integrating Anomaly Detection With Misuse Detection," Expert Syst. Appl., vol. 41, no. 4, pp. 1690–1700, 2014.

[19] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman, "Robust Intelligent Malware Detection Using Deep Learning," IEEE Access, vol. 7, pp. 46717–46738, 2019.

[20] Zhang, Y. Wang, and M. P. Wellman, "Interpretable Machine Learning Models for Cybersecurity Applications," IEEE Trans. Dependable Secure Comput., vol. 18, no. 6, pp. 2892–2905, 2021.

[21] Molnar, Interpretable Machine Learning, 2nd ed. Morrisville, NC, USA: Lulu.com, 2020.

[22] Doshi-Velez and B. Kim, "Towards A Rigorous Science of Interpretable Machine Learning," arXiv preprint arXiv:1702.08608, 2017.

[23] C. M. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer, 2006.

[24] Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2011.

[25] Alpaydin, Introduction to Machine Learning, 4th ed. Cambridge, MA, USA: MIT Press, 2020.

[26] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, 2nd ed. New York, NY, USA: Springer, 2009.

[27] Pedregosa et al., "Scikit-learn: Machine Learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[28] S. M. Lundberg et al., "From Local Explanations to Global Understanding with Explainable AI for Trees," Nature Mach. Intell., vol. 2, no. 1, pp. 56–67, 2020.

[29] W. Wang, X. Guan, and X. Zhang, "A Novel Intrusion Detection Method Based on Principal Component Analysis in Computer Security," in Proc. Int. Conf. Neural Netw. Brain, 2010, pp. 657–662.