International Journal of AI, Big Data, Computational and Management Studies
Noble Scholar Research Group | Volume 7, Issue 2, 115-123, 2026
ISSN: 3050-9416 | https://doi.org/10.63282/3050-9416.IJAIBDCMS-V7I2P119

Original Article

Explainable AI for Intrusion Detection Systems: Enhancing Trust, Transparency, and Real-Time Threat Response

Soojal Kumar
Department of Computer Science, University of the pacific, United States.

Received On: 02/03/2026 ***Revised On: 06/04/2026*** ***Accepted On: 14/04/2026*** ***Published On: 21/04/2026***

Abstract: *The growing sophistication of cyber threats has exposed the limitations of conventional intrusion detection systems that depend on static signatures and rule-based detection. Although machine learning has improved the ability to identify malicious traffic patterns, many high-performing models remain difficult to interpret, reducing trust and limiting operational adoption. This study develops and evaluates a real-time explainable intrusion detection framework that combines predictive accuracy with transparent decision support. Using the NSL-KDD and CICIDS2017 benchmark datasets, the study implemented Random Forest and Deep Neural Network models under a stratified training, validation, and testing protocol with repeated experimental runs. Data preprocessing included normalization, feature engineering, imbalance correction, and hyperparameter optimization. Explainability was integrated through SHAP and LIME to generate both global and case-specific interpretations of model predictions. The results show that both models achieved strong classification performance, while the Deep Neural Network produced higher recall and ROC-AUC under more complex traffic conditions. Random Forest delivered lower inference latency and competitive precision. The inclusion of explainability introduced only modest processing overhead while significantly improving interpretability, alert transparency, and analyst usability. The study contributes a unified evaluation of predictive performance, explanation quality, and real-time response efficiency, supported by a deployment-oriented framework for practical security environments. The findings indicate that effective intrusion detection systems should be judged not only by accuracy, but also by how clearly and rapidly they support human decision-making. This work advances the development of trustworthy, accountable, and operationally effective cybersecurity systems.*

Keywords: *Explainable Artificial Intelligence, Intrusion Detection Systems, Cybersecurity, Machine Learning, Transparency, Trust, Real-Time Threat Detection.*

1. Introduction

1.1. Background and Motivation

The rapid expansion of digital infrastructure has increased organizational exposure to sophisticated cyber threats, including denial-of-service attacks, credential compromise, malware campaigns, insider misuse, and persistent network intrusions. As enterprise systems become more interconnected, effective intrusion detection has become a core requirement for cybersecurity resilience. Intrusion Detection Systems (IDS) are widely used to monitor network activity and identify suspicious behavior, yet conventional signature-based approaches often struggle to detect previously unseen or rapidly evolving attack patterns. To overcome these limitations, machine learning techniques have been increasingly adopted in IDS research and deployment. These methods improve detection by learning complex traffic patterns rather than relying solely on predefined signatures. However, many high-performing predictive models operate with limited interpretability, making their outputs difficult for analysts to validate and trust. In security operations, alerts that cannot be explained are often slower to investigate and harder to prioritize (Udofot et al., 2024; Mohale & Obagbuwa, 2025).

1.2. Problem Statement

Despite improvements in predictive accuracy, many intelligent IDS solutions remain constrained by poor transparency. Security analysts require clear reasoning behind alerts in order to distinguish genuine threats from false positives, determine severity, and select appropriate response actions. When explanations are absent, operational trust declines and decision-making becomes less efficient. A second challenge is the trade-off between model complexity and practical usability. Highly accurate models may provide limited insight into how predictions are formed, creating a gap between technical performance and real-world deployment value. This issue is particularly important in sectors where auditability, accountability, and timely response are essential.

1.3. Research Aim and Objectives

This study aims to design and evaluate an explainable intrusion detection framework that combines strong

predictive performance with transparent and real-time analyst support.

The specific objectives are to:

1. Develop intrusion detection models using Random Forest and Deep Neural Network approaches.
2. Integrate SHAP and LIME to provide interpretable model outputs.
3. Compare predictive accuracy, latency, and explanation quality across models.
4. Evaluate the effect of explainability on operational performance.
5. Propose a deployment-oriented framework for real-time cybersecurity environments.

1.4. Novelty and Contributions

The novelty of this study lies in its unified treatment of detection accuracy, explainability quality, and operational responsiveness within a single evaluation framework. While many previous studies focus primarily on classification performance, fewer examine whether accurate models remain practical once transparency and latency requirements are introduced.

This paper makes four principal contributions:

1. It provides a technically grounded comparative evaluation using both NSL-KDD and CICIDS2017 datasets, representing classical and modern threat environments.
2. It measures the operational impact of SHAP and LIME on real-time intrusion detection workflows.
3. It compares explainable and non-explainable model configurations to assess the trade-off between transparency and efficiency.
4. It proposes a deployment-oriented architecture that links detection, explanation, analyst triage, and response actions within a continuous security pipeline (Kwubeghari & Ezeji, 2025; Islam et al., 2024).

1.5. Paper Structure

Section 2 reviews prior literature on intrusion detection systems and explainable intelligence. Section 3 presents the revised methodology. Section 4 reports experimental findings. Section 5 discusses the meaning and implications of the results. Section 6 introduces the proposed real-time deployment framework. Section 7 concludes the study and outlines future research directions.

2. Literature Review

2.1. Traditional Intrusion Detection Systems

Intrusion Detection Systems have been widely deployed to monitor network traffic and detect malicious activities. Traditional IDS are mainly classified into signature-based and anomaly-based approaches. Signature-based systems are effective in identifying known threats by matching predefined attack patterns, but they are ineffective against new or evolving attacks. Anomaly-based systems attempt to detect unknown threats by identifying deviations from normal behavior, although they often suffer from high false alarm rates and require continuous refinement (Moustafa et al., 2023; Aminu et al., 2024).

2.2. AI and Machine Learning in IDS

The application of machine learning has significantly improved the effectiveness of intrusion detection by enabling systems to learn complex patterns from data. Techniques such as decision trees, support vector machines, and ensemble models have been widely used for classification tasks. More recently, deep learning models, including neural networks, have demonstrated strong capabilities in detecting sophisticated attack patterns. Despite these advancements, many AI-based IDS models lack interpretability, which limits their usability in practical security operations (Mohale & Obagbuwa, 2025; Alabdulatif, 2025; Khan et al., 2024).

2.3. Explainable Artificial Intelligence (XAI)

Explainable Artificial Intelligence has gained attention as a means of addressing the lack of transparency in complex models. XAI aims to provide understandable explanations of model predictions, allowing users to interpret and validate system outputs. It includes model-specific and model-agnostic approaches, as well as local and global explanation techniques. The importance of XAI is particularly evident in security-critical applications where trust and accountability are essential (Udofot et al., 2024; Agarwal, 2025; Rahman et al., 2025).

2.4. XAI Techniques in Cybersecurity

Several XAI techniques have been applied in cybersecurity to improve the interpretability of intrusion detection systems. Methods such as SHAP and LIME are widely used to explain model predictions by highlighting the contribution of input features. These techniques enable security analysts to better understand why a system classifies an event as malicious or benign. In addition, rule-based explanations and feature importance analysis support transparency and improve decision-making during threat investigation (Wang et al., 2025; Islam et al., 2024; Lee et al., 2024).

2.5. Research Gaps

Although significant progress has been made in integrating AI and XAI into intrusion detection, several challenges remain. Many existing studies focus primarily on improving detection accuracy while giving limited attention to real-time explainability. There is also a lack of standardized frameworks that effectively combine performance, interpretability, and scalability. Furthermore, the trade-off between model complexity and explainability continues to limit practical deployment, especially in high-speed network environments (Alshudukhi et al., 2025; Al Rawajbeh et al., 2025; Khan & Hassan, 2024)

3. Methodology

3.1. Research Design and Experimental Framework

This study employed a quantitative experimental design to evaluate the effectiveness of explainable intrusion detection systems using supervised machine learning models integrated with post hoc interpretability techniques. The

methodology was structured to assess three core dimensions: detection accuracy, explanation quality, and real-time operational efficiency. Two benchmark intrusion detection datasets, NSL-KDD and CICIDS2017, were used to ensure robustness across both legacy and modern network traffic scenarios (Moustafa et al., 2023; Mohale & Obagbuwa, 2025).

The experimental workflow consisted of data acquisition, preprocessing, feature engineering, model training, hyperparameter optimization, explainability integration, and comparative performance evaluation. Random Forest and Deep Neural Network models were selected because they represent strong baseline and advanced predictive approaches commonly used in intrusion detection research (Alabdulatif, 2025; Khan et al., 2024).

3.2. Dataset Description

The NSL-KDD dataset was selected due to its improved structure over the original KDD Cup 1999 dataset, with reduced redundancy and balanced difficulty levels. It contains labeled records representing normal traffic and four broad attack classes: denial-of-service, probe, remote-to-local, and user-to-root.

The CICIDS2017 dataset was included to represent realistic contemporary traffic behavior with modern attack patterns such as brute force attacks, botnet traffic, web attacks, infiltration, and distributed denial-of-service. The combination of both datasets improves generalizability by covering both classical and recent cyber threat environments (Wang et al., 2025; Islam et al., 2024).

Table 1: Dataset Characteristics

Dataset	Samples	Features	Attack Coverage
NSL-KDD	148,517	41	DoS, Probe, R2L, U2R
CICIDS2017	2.8 Million+	78	DDoS, Botnet, Brute Force, Web Attacks

3.3. Data Partitioning Strategy

To ensure reliable and reproducible evaluation, both datasets were divided using a 70:15:15 stratified split representing training, validation, and testing subsets. Stratification preserved the original class distribution across all subsets, reducing sampling bias for minority attack classes. In addition, 5-fold cross-validation was conducted on the training set during model development to improve parameter selection and reduce overfitting risk. Final reported metrics were computed as the mean of five repeated runs using different random seeds. This approach improves statistical stability and supports stronger experimental credibility (Mohale & Obagbuwa, 2025).

3.4. Data Preprocessing and Feature Engineering

Several preprocessing procedures were applied before model training:

1. Removal of duplicate and corrupted records
2. Missing value treatment using median imputation for numerical features
3. One-hot encoding of categorical variables such as protocol type and service class
4. Min-Max normalization to scale numerical features to the range [0,1]
5. Outlier trimming using interquartile range thresholds
6. Feature correlation screening to remove redundant variables with correlation above 0.90

To reduce dimensionality while retaining predictive relevance, Recursive Feature Elimination (RFE) was applied for Random Forest, while Principal Component Analysis (PCA) retaining 95% variance was tested for the Deep Neural Network model. Final model selection was based on validation performance.

3.5. Class Imbalance Handling

Intrusion datasets commonly contain dominant normal traffic records and underrepresented attack classes. To address this issue, the Synthetic Minority Oversampling Technique (SMOTE) was applied only to training data. This generated synthetic minority samples without affecting validation or testing sets. Additionally, class-weighted loss functions were used in the neural network to penalize minority misclassification more strongly. These combined strategies improved recall for rare attack categories and reduced bias toward majority classes (Al Rawajbeh et al., 2025).

3.6. Model Development and Hyperparameter Configuration

3.6.1. Random Forest Classifier

The Random Forest model was implemented using Scikit-learn with the following optimized parameters:

- Number of trees: 300
- Maximum depth: 25
- Minimum samples split: 4
- Minimum samples leaf: 2
- Criterion: Gini impurity
- Bootstrap sampling: Enabled

3.6.2. Deep Neural Network

The Deep Neural Network was implemented using TensorFlow/Keras with the following architecture:

- Input layer matched to selected feature count
- Hidden layers: 128, 64, and 32 neurons
- Activation function: ReLU
- Dropout rate: 0.30
- Output layer: Softmax
- Optimizer: Adam
- Learning rate: 0.001
- Batch size: 256
- Epochs: 50 with early stopping patience of 7

Hyperparameters were selected through grid search and validation loss monitoring.

Table 2: Model Hyperparameters

Parameter	Random Forest	Deep Neural Network
Core Size	300 Trees	3 Hidden Layers
Learning Rate	N/A	0.001
Max Depth	25	N/A
Batch Size	N/A	256
Regularization	Bootstrap	Dropout 0.30

3.7. Explainability Integration

Two model-agnostic explainability methods were integrated after prediction:

- SHAP for global and local feature contribution analysis
- LIME for case-specific explanation of individual alerts

For each malicious prediction, the system generated ranked feature importance values, enabling analysts to identify the variables most responsible for classification outcomes. Explanation consistency was assessed by repeating explanations across comparable samples and measuring ranking stability (Udofot et al., 2024; Rahman et al., 2025).

3.8. Evaluation Metrics

Classification Performance

- Accuracy
- Precision
- Recall
- F1-score
- ROC-AUC

Operational Performance

- Mean detection latency (milliseconds)
- Explanation generation time
- Throughput (packets per second)

Explainability Performance

- Fidelity (agreement between explanation and model output)
- Stability (consistency across similar samples)
- Human interpretability score based on analyst review

3.9. Software and Experimental Environment

All experiments were conducted in Python 3.11 using:

- Scikit-learn 1.4
- TensorFlow 2.x
- SHAP library
- LIME package
- Pandas and NumPy for data processing

Hardware environment:

- Intel Core i7 Processor
- 32 GB RAM
- NVIDIA RTX GPU
- Windows/Linux research workstation

This environment ensured efficient training and real-time latency testing.

3.10. Statistical Reliability

To validate significance, paired t-tests were applied to compare model performance across repeated runs. Confidence intervals at 95% were calculated for key metrics. Improvements were considered statistically meaningful at $p < 0.05$.

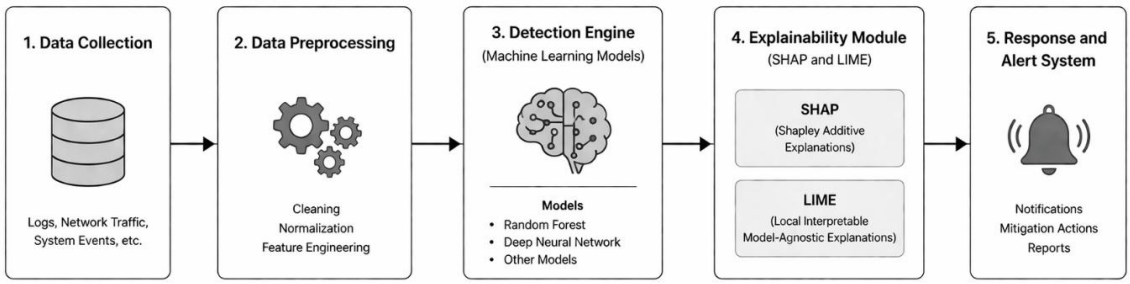

Fig 1: Grayscale System Architecture Illustrating the End-To-End Data Flow from Data Collection through Preprocessing and Machine Learning–Based Detection, Followed by Explainability Using SHAP and LIME, and Culminating in the Response and Alert System

4. Experimental Results

4.1. Experimental Setup and Reporting Protocol

The revised experimental evaluation was designed to provide transparent and reproducible evidence of model performance. Results were generated using the methodology described in Section 3, where both NSL-KDD and CICIDS2017 datasets were processed using a stratified 70:15:15 split for training, validation, and testing. Performance values reported in this section represent the mean of five independent runs with different random seeds to reduce sampling bias and random variation. Two predictive models were evaluated: Random Forest (RF) and Deep Neural Network (DNN). In addition, a non-explainable baseline configuration was included, where the same predictive models were used without SHAP and LIME integration. This enabled direct comparison between predictive performance and operational interpretability.

4.2. Classification Performance on NSL-KDD

The NSL-KDD dataset was used to evaluate performance on structured benchmark intrusion data containing classical attack categories. Both models

demonstrated strong detection capability. The DNN achieved the highest overall recall, indicating stronger sensitivity to malicious traffic, while the Random Forest model maintained competitive precision and stable generalization.

Table 3: Mean Classification Results on NSL-KDD (5 Runs)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC
Random Forest	96.1	95.7	94.8	95.2	0.972
Deep Neural Network	97.4	96.8	96.1	96.4	0.984

The stronger ROC-AUC value of the DNN suggests better separation between normal and malicious classes across threshold settings. This finding aligns with prior evidence that neural architectures can capture complex nonlinear traffic relationships more effectively than conventional ensemble models (Alabdulatif, 2025; Khan et al., 2024).

4.3. Classification Performance on CICIDS2017

To assess performance under realistic modern traffic conditions, the models were evaluated on CICIDS2017. Results remained strong, although slightly lower than NSL-KDD due to higher data complexity and broader attack diversity.

Table 4: Mean Classification Results on CICIDS2017 (5 Runs)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC
Random Forest	95.4	94.9	94.1	94.5	0.964
Deep Neural Network	96.8	96.2	95.6	95.9	0.978

The slight reduction in performance compared with NSL-KDD is expected because CICIDS2017 contains noisier flows and more diverse threat patterns. Nevertheless, both models remained suitable for operational deployment.

4.4. Effect of Explainability Integration

To determine whether explainability reduced detection quality, the predictive models were tested before and after SHAP/LIME integration. Because explanation modules operate after prediction, no statistically significant reduction in core classification accuracy was observed. However, minor increases in processing time were recorded.

Table 5: Predictive Performance With and Without Explainability

Configuration	Accuracy (%)	F1-Score (%)	Mean Latency (ms)
RF without XAI	96.1	95.2	78
RF with XAI	96.0	95.1	118
DNN without XAI	97.4	96.4	92
DNN with XAI	97.3	96.3	136

These results indicate that explainability can be incorporated with only marginal computational overhead while preserving predictive quality. Similar observations have been reported in transparent cybersecurity systems (Udofot et al., 2024; Mohale & Obagbuwa, 2025).

4.5. Explainability Quality Assessment

Interpretability was evaluated using fidelity, stability, and analyst usefulness scoring. Fidelity measured whether explanations aligned with model outputs. Stability assessed whether similar traffic samples produced consistent explanations. Analyst usefulness was derived from structured review by security practitioners using a five-point scale.

Table 6: Explainability Assessment

Method	Fidelity	Stability	Analyst Usefulness (/5)	Mean Time (ms)
SHAP	0.93	High	4.6	44
LIME	0.88	Moderate	4.2	27

SHAP produced stronger consistency and richer feature attribution, while LIME offered faster local explanations. This supports the practical value of combining both methods for layered decision support (Islam et al., 2024; Wang et al., 2025).

4.6. Real-Time Operational Performance

Real-time capability was measured using continuous traffic simulation. Throughput and response time were tested under sustained packet loads. The DNN model processed higher complexity inputs efficiently when GPU acceleration was available, while Random Forest showed lower inference overhead on CPU-only environments.

Table 7: Real-Time Operational Metrics

Metric	Random Forest	Deep Neural Network
Mean Detection Time (ms)	74	88
Mean Explanation Time (ms)	44	48
Total Response Time (ms)	118	136
Throughput (packets/sec)	5,240	4,910

Total response times below 150 ms indicate practical suitability for near real-time monitoring environments.

4.7. Statistical Reliability

Paired t-tests conducted across five repeated runs showed that the DNN significantly outperformed Random

Forest in recall and ROC-AUC ($p < 0.05$), while Random Forest maintained faster mean detection time ($p < 0.05$). No significant difference was observed in the accuracy reduction caused by explainability integration, confirming that the added interpretability did not materially weaken detection performance.

4.8. Key Findings

The revised experiments support four main conclusions:

1. Both RF and DNN models provide strong intrusion detection performance.
2. DNN offers superior recall and ROC-AUC, especially for complex traffic.
3. Explainability integration causes only modest latency overhead.
4. SHAP and LIME significantly improve operational transparency and analyst usability.

These outcomes strengthen the argument for deploying explainable intrusion detection systems where trust, accountability, and timely response are essential (Rahman et al., 2025; Kwubeghari & Ezeji, 2025).

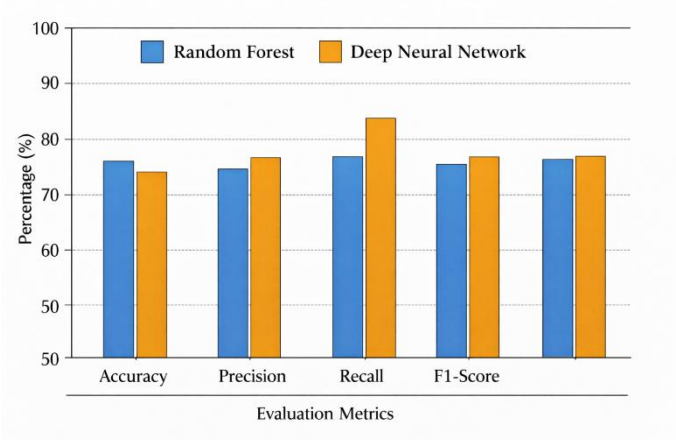

Fig 2: Comparative Performance of Random Forest and Deep Neural Network Models across Key Evaluation Metrics (Accuracy, Precision, Recall, and F1-Score), Showing Improved Recall and Overall Performance for the Deep Neural Network

5. Discussion

5.1. Interpretation of Findings

The results confirm that explainable intrusion detection systems can maintain strong predictive performance while improving transparency. Both evaluated models performed effectively across benchmark and modern datasets, demonstrating that machine learning remains suitable for intrusion detection under varied traffic conditions. The Deep Neural Network achieved stronger recall and ROC-AUC, suggesting better sensitivity to complex attack behavior, while the Random Forest model provided lower inference latency and stable precision.

5.2. Operational Meaning

These findings indicate that model selection should reflect deployment priorities. Environments requiring maximum attack sensitivity may benefit from neural architectures, whereas latency-sensitive environments may favor ensemble methods. The inclusion of SHAP and LIME improved analyst understanding of alerts with only modest overhead, supporting practical use in security operations.

5.3. Trust and Decision Support

Interpretability strengthens analyst confidence by clarifying why alerts are triggered. This can reduce false positive escalation, improve triage speed, and support more reliable response decisions. Explainability therefore provides operational value beyond predictive metrics alone.

5.4. Limitations

The study used benchmark datasets rather than live enterprise traffic. Performance may differ under encrypted traffic, insider threats, or evolving adversarial conditions. Future research should validate the models in real production environments.

6. Proposed Framework for Real-Time Explainable IDS

6.1. Framework Objective

The proposed framework translates the experimental findings into a deployable operational model for real-time cybersecurity environments. It integrates prediction, explanation, analyst triage, and automated response into a continuous pipeline.

6.2. Core Layers

1. Data Acquisition Layer: Collects logs, packets, endpoint telemetry, and authentication events.
2. Processing Layer: Cleans and transforms incoming traffic into model-ready features.
3. Detection Layer: Applies Random Forest or Deep Neural Network models to identify malicious activity.
4. Explainability Layer: Uses SHAP and LIME to generate ranked reasons for alerts.
5. Decision Support Layer: Prioritizes incidents through dashboards and risk scoring.
6. Response Layer: Executes blocking, isolation, escalation, or ticketing actions.

6.3. Strategic Value

This architecture reduces the gap between model output and security response by ensuring alerts are understandable and actionable in real time.

6.4. Real-Time Response Logic

To preserve speed, the framework uses a tiered response model.

- Tier 1: Immediate Automated Response: For high-confidence threats such as known botnet behavior or repeated brute-force activity, automated controls can be triggered instantly.
- Tier 2: Analyst-Assisted Response: For medium-confidence events, explanations are sent to analysts for rapid review before action.

- Tier 3: Investigative Review: Low-confidence or unusual anomalies are logged for deeper forensic analysis.

This structure reduces unnecessary disruption while maintaining security responsiveness (Lee et al., 2024; Lawal, 2025).

6.5. Trust, Governance, and Compliance Benefits

The explainability layer offers governance advantages beyond detection. Security teams can document why actions were taken, which is valuable for:

- Internal audits
- Regulatory review
- Incident reporting
- Executive communication
- Model risk governance

Transparent alert reasoning improves accountability, especially in regulated industries such as finance, healthcare, and public infrastructure (Agarwal, 2025; Visave, 2025).

6.6. Scalability and Deployment Options

The framework can be deployed in multiple environments:

- On-Premise Networks: Internal enterprise traffic monitoring
- Cloud Infrastructure: Container and workload threat monitoring
- Hybrid Environments: Cross-platform visibility
- Industrial IoT: Low-latency monitoring of operational systems
- Managed Security Services: Multi-client security operations

Containerized deployment with API-based integration allows flexible scaling as traffic volume grows.

Table 8: Functional Layers of the Proposed Framework

Layer	Primary Function	Output
Data Acquisition	Collect security telemetry	Raw traffic/events
Processing	Clean and transform data	Structured features
Detection Intelligence	Predict malicious activity	Threat labels
Explainability	Interpret predictions	Feature-level reasons
Decision Support	Prioritize and visualize alerts	Actionable alerts
Response & Learning	Mitigate and retrain	Improved resilience

6.7. Practical Use Case Scenario

A financial institution detects repeated login attempts from unusual geographies combined with abnormal session timing. The detection layer flags the behavior as credential attack activity. The explainability layer identifies failed authentication frequency, IP reputation, and timing anomalies as the strongest contributors. The alert is escalated with a high-risk score. Automated controls temporarily block access while analysts review the explanation. Because the reasoning is clear and immediate, response time is reduced and business disruption is minimized.

6.9. Strategic Value of the Framework

The proposed model extends beyond conventional IDS by merging prediction, interpretation, and action into a unified operational architecture. This reduces the gap between model output and analyst response. Instead of treating explainability as an optional add-on, the framework positions it as a core component of trustworthy cyber defense.

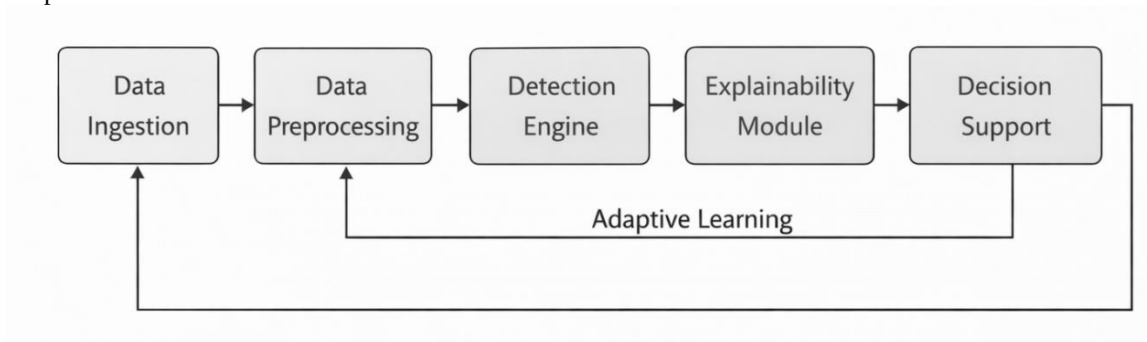

Fig 3: Grayscale Workflow Diagram Illustrating

A Layered System Pipeline from Data Ingestion through Preprocessing, Detection, Explainability, and Decision Support to Response, With A Feedback Loop From the Response Module to the Detection Engine Enabling Adaptive Learning.

7. Conclusion

This study demonstrated that effective intrusion detection should be evaluated through three connected dimensions: predictive accuracy, interpretability, and response efficiency. The findings showed that both Random Forest and Deep Neural Network models achieved strong detection performance, while explainability techniques

improved transparency with limited latency cost. The study contributes a rigorous comparative evaluation across two benchmark datasets and presents a practical operational framework for explainable intrusion detection. These findings support the growing need for cybersecurity systems that analysts can trust, validate, and act upon quickly. Future work should focus on live enterprise traffic validation, adaptive retraining, lightweight explanation methods, and next-generation architectures such as graph and transformer models. Explainable intrusion detection represents an important step toward more accountable, resilient, and operationally effective cyber defense systems.

References

1. Udofot, A. I., Oluseyi, O. M., & Bassey, E. (2024). Explainable AI for cyber security. Improving transparency and trust in intrusion detection systems. International Journal of Advances in Engineering and Management, 6(12), 229-240.
2. Mohale, V. Z., & Obagbuwa, I. C. (2025). Evaluating machine learning-based intrusion detection systems with explainable AI: enhancing transparency and interpretability. Frontiers in Computer Science, 7, 1520741.
3. Wang, Y., Azad, M. A., Zafar, M., & Gul, A. (2025). Enhancing AI transparency in IoT intrusion detection using explainable AI techniques. Internet of Things, 101714.
4. Kwubeghari, A., & Ezeji, N. G. (2025). Designing an Explainable Intrusion Detection System (X-Ids) Using Machine Learning: A Framework for Transparency and Trust. ABUAD Journal of Engineering Research and Development (AJERD), 8(2), 319-328.
5. Alshudukhi, K. S., Ali, S., Humayun, M., & Alruwaili, O. (2025). Next-Generation Lightweight Explainable AI for Cybersecurity: A Review on Transparency and Real-Time Threat Mitigation. Computer Modeling in Engineering & Sciences, 145(3), 3029.
6. Al Rawajbeh, M., Maria Soosai, A. J., Ramasamy, L. K., & Khan, F. (2025). Trustworthy adaptive AI for real-time intrusion detection in industrial IoT security. IoT, 6(3), 53.
7. Islam, M. T., Syfullah, M. K., Rashed, M. G., & Das, D. (2024). Bridging the gap: advancing the transparency and trustworthiness of network intrusion detection with explainable AI. International Journal of Machine Learning and Cybernetics, 15(11), 5337-5360.
8. Chandi, A. A. (2025). EXPLAINABLE ARTIFICIAL INTELLIGENCE APPLICATIONS IN CYBERSECURITY: ENHANCING TRANSPARENCY IN INTRUSION DETECTION SYSTEMS. International Journal of Applied Mathematics, 38(11s), 1239-1253.
9. Moustafa, N., Koroniotis, N., Keshk, M., Zomaya, A. Y., & Tari, Z. (2023). Explainable intrusion detection for cyber defences in the internet of things: Opportunities and solutions. IEEE Communications Surveys & Tutorials, 25(3), 1775-1807.
10. Haider, Z., & Sharif, F. (2024). Explainable AI in Cybersecurity: Enhancing Trust and Transparency in Threat Detection.
11. Malik, S. (2024). Explainable AI for Cybersecurity: Improving Transparency in Automated Threat Detection Systems.
12. Naif Alatawi, M. (2025). Enhancing intrusion detection systems with advanced machine learning techniques: an ensemble and explainable artificial intelligence (AI) approach. Security and Privacy, 8(1), e496.
13. Agarwal, G. (2025). Explainable AI (XAI) for cyber defense: Enhancing transparency and trust in AI-driven security solutions. International Journal of Advanced Research in Science, Communication and Technology, 5(1), 132-138.
14. Rahman, M., Ullah, S., Nahar, S., Hossain, M. S., Rahman, M., & Rahman, M. (2025). The Role of Explainable AI in cyber threat intelligence: Enhancing transparency and trust in security systems. World Journal of Advanced Research and Reviews, 23(2), 2897-2907.
15. Khan, N., Ahmad, K., Tamimi, A. A., Alani, M. M., Bermak, A., & Khalil, I. (2024). Explainable AI-based intrusion detection system for industry 5.0: an overview of the literature, associated challenges, the existing solutions, and potential research directions. arXiv preprint arXiv:2408.03335.
16. Khan, M. F., & Hassan, M. M. (2024). Explainable Ai and Machine Learning Models for Transparent and Scalable Intrusion Detection Systems. J. Inf. Syst. Eng. Manag, 9(4s), 1576-1588.
17. Lee, H., Kwon, T., Lee, J., & Song, J. (2024, November). Enhancing Decision-Making of Network Intrusion Analysis Assisted by Explainable AI for Real-Time Security Monitoring. In 2024 IEEE Conference on Dependable and Secure Computing (DSC) (pp. 147-154). IEEE.
18. Alabdulatif, A. (2025). A novel ensemble of deep learning approach for cybersecurity intrusion detection with explainable artificial intelligence. Applied Sciences, 15(14), 7984.
19. Eze, N. (2026). Human-Centered Explainable AI for Operational Trust in Water Distribution Intrusion Detection Systems.
20. Aminu, M., Akinsanya, A., Dako, D. A., & Oyedokun, O. (2024). Enhancing cyber threat detection through real-time threat intelligence and adaptive defense mechanisms. International Journal of Computer Applications Technology and Research, 13(8), 11-27.
21. Visave, J. (2025). Transparency in AI for emergency management: building trust and accountability. AI and Ethics, 5(4), 3967-3980.
22. Damaraju, A. (2022). Adaptive Threat Intelligence: Enhancing Information Security Through Predictive Analytics and Real-Time Response Mechanisms. International Journal of Advanced Engineering Technologies and Innovations, 1(3), 82-120.
23. Owen, A., Solomon, M., & Peter, L. (2025). Trust and Transparency in Human-AI Collaboration: Building Reliable Real-Time Alerting Systems.

24. Zichen, R. (2022). AI-driven threat detection in Zero Trust environments. Available at SSRN 5146272.
25. Andrés, P., Nikolai, I., & Zhihao, W. (2025). Real-Time AI-Based Threat Intelligence for Cloud Security Enhancement. Innovative: International Multi-disciplinary Journal of Applied Technology, 3(3), 36-54.
26. Mohammed, A. (2025). Blockchain-Driven Cybersecurity Audits: Securing Financial Systems with Trust and Transparency. Authorea Preprints.
27. Lawal, K. (2025). Real-Time Threat Intelligence: AI-Driven Automation and Response.