IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15494

Explainable Artificial Intelligence-Based Network Intrusion Detection System Using SHAP, LIME and Counterfactual Analysis

Manam Siva Sai[1], Dr. Chandra Sekhar Koppireddy[2]

M.Tech Student, CSE Department, Pragati Engineering College (A), Surampalem ,A.P., India[1]

Associate Professor,CSE Department, Pragati Engineering College (A), Surampalem , A.P., India[2]

Abstract: The rapid evolution of cyber threats has exposed the limitations of traditional signature-based intrusion detection systems. While machine learning offers strong detection capability, its opacity undermines analyst trust. This research proposes an Explainable AI-based Network Intrusion Detection System combining XGBoost classification with a comprehensive SHAP explanation framework.The system processes traffic from three benchmark datasets — UNSW-NB15, CIC-IDS-2017, and NSL-KDD — through automated preprocessing covering encoding, normalisation, stratified partitioning, and class imbalance handling. The XGBoost classifier achieves F1-Scores of 0.9793, 0.9914, and 0.9853 respectively. SHAP TreeExplainer generates six visualisation types spanning global and local explanations, further complemented by LIME surrogate modelling and counterfactual generation — forming three mutually validating interpretability channels.Findings consistently identify volumetric flow statistics, behavioural connection-count features, and protocol-state indicators as dominant discriminating factors, aligning with established network security knowledge and reinforcing both the model's reliability and real-world applicability.

Keywords: Explainable AI, Network Intrusion Detection, SHAP, XGBoost, LIME, Cyber Security

I. INTRODUCTION

The rapid digitalisation of critical infrastructure, financial institutions, healthcare systems, and governmental networks has created an expansive and continuously evolving attack surface that malicious actors exploit with increasing sophistication. Cyberattacks have grown not only in volume but also in complexity, with adversaries employing polymorphic malware, zero-day exploits, encrypted command-and-control channels, and multi-stage Advanced Persistent Threats that systematically bypass conventional perimeter defences. Traditional Intrusion Detection Systems that rely on manually curated signature libraries and static rule sets prove fundamentally inadequate against such dynamic threats, as they are incapable of recognising attack patterns that have not been explicitly pre-defined [1]. This limitation has created an urgent demand for intelligent, adaptive, and self-learning detection mechanisms capable of identifying novel intrusion behaviour from statistical regularities in network traffic data.

Machine learning has emerged as a transformative paradigm for network intrusion detection, offering classifiers that generalise beyond fixed signatures to detect previously unseen attack patterns. Ensemble methods such as Random Forest, Gradient Boosting, and XGBoost have demonstrated exceptional performance on standardised benchmark datasets including UNSW-NB15, CIC-IDS-2017, and NSL-KDD, consistently achieving accuracy and F1-Scores exceeding 97% [2]. Deep learning architectures including Convolutional Neural Networks and Long Short-Term Memory networks have further extended these capabilities by learning hierarchical representations directly from raw traffic streams. Despite these advances, a fundamental and well-documented tension persists between predictive power and model transparency. The ensemble and deep architectures that achieve the highest detection accuracy are simultaneously the most opaque, producing classification decisions without any externally accessible reasoning that analysts can examine, validate, or challenge [3].

This paper presents an XAI-based Network Intrusion Detection System that integrates XGBoost classification with a comprehensive multi-modal SHAP explanation framework, LIME surrogate modelling, and gradient-guided counterfactual generation, all delivered within an interactive desktop application. The key contributions of this work are:

(1) An automated pre-processing pipeline supporting three benchmark datasets with systematic categorical encoding and class imbalance compensation;
(2) Six distinct SHAP visualisation types providing global and local model interpretability;

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15494

(3) Integration of SHAP, LIME, and counterfactual explanations as three independent, cross-validating interpretability channels;
(4) An interactive GUI application making XAI capabilities accessible to non-specialist security practitioners;

A. Intrusion Detection

Intrusion Detection is the process of monitoring network traffic, system activities, and user behaviour to identify unauthorized access attempts, policy violations, malicious activities, or any suspicious events that could compromise the security, integrity, or availability of a computer system or network.

B. Types of Intrusion Detection Systems

Common Attack Types That IDS Detects [4]

- Port Scanning — Attackers probing open ports to find entry points
- DoS / DDoS Attacks — Flooding a network or server to crash it
- Brute Force Attacks — Repeatedly guessing passwords to gain access
- SQL Injection — Inserting malicious code into database queries
- Man-in-the-Middle Attacks — Intercepting communications between two parties
- Malware Communication — Infected machines talking back to attacker servers

C. Role of Machine Learning in Intrusion Detection

Traditional signature-based systems fail against new and evolving attacks. This is where Machine Learning-based IDS becomes important. Instead of relying on fixed rules, ML models learn patterns from thousands of examples of both normal and attack traffic and build a mathematical model that can classify new connections automatically.

Popular ML algorithms used in IDS include [3-4]:

- Random Forest
- XGBoost
- Support Vector Machines
- Deep Neural Networks

D. Role of Explainable AI (XAI) in IDS

Even when a machine learning model detects an intrusion accurately, it does not automatically explain why it made that decision. This is the black-box problem.

Security analysts need to know:

- Which features of the traffic triggered the alert?
- How confident is the system in its decision?
- What would need to change for the traffic to be considered safe?

This is exactly what Explainable AI techniques like SHAP and LIME solve — they open up the black box and give human-readable reasons for every detection decision, making the system more trustworthy and useful in real operational environments.

II. LITERATURE SURVEY

A. Problems and Limitations of Intrusion Detection Using Artificial Intelligence

Although Artificial Intelligence has significantly strengthened network intrusion detection capabilities, several important limitations continue to challenge its practical deployment in real operational environments [5-6].

- Data Quality and Benchmark Limitations
- Class Imbalance
- Excessive False Alarms
- Opacity and Lack of Explainability
- Computational Demands
- Encrypted Traffic Challenges

These limitations collectively demonstrate that AI is a powerful but imperfect foundation for intrusion detection. Meaningful progress requires sustained research into adversarially robust training, drift-aware continual learning, richer encrypted traffic analysis, and deeper integration of explainability frameworks that make automated detection decisions genuinely transparent and trustworthy for the analysts who depend on them.

B. How Explainable AI (XAI) Overcomes the Limitations of AI-Based Intrusion Detection

Artificial Intelligence has transformed network intrusion detection, but its practical deployment is hampered by opacity, false alarms, model degradation, and analyst distrust. Explainable AI directly addresses these barriers by making automated detection decisions transparent, auditable, and genuinely actionable for security practitioners [6].

- Eliminating Black-Box Opacity Through Transparent Reasoning
- Reducing False Alarms and Enabling Intelligent Alert Triage

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15494

- Detecting and Responding to Concept Drift
- Hardening Against Adversarial Evasion and Strengthening Security Rules
- Enabling Regulatory Compliance and Building Analyst Trust

TABLE I — Existing Machine Learning Techniques for Intrusion Detection [7-9]

Technique	Description	Dataset Used	Accuracy Reported	Key Strength
Decision Tree	Builds a tree of if-then rules based on feature thresholds to classify traffic	NSL-KDD	92.4%	Simple, human-readable rules
Random Forest	Combines hundreds of decision trees through majority voting for final classification	CIC-IDS-2017	97.2%	Robust against overfitting and noise
Gradient Boosting	Builds trees sequentially where each corrects errors of the previous model	NSL-KDD	96.8%	Strong generalisation on tabular data
XGBoost	Regularised and optimised gradient boosting with built-in imbalance handling	UNSW-NB15	98.2%	Highest accuracy with efficient computation

TABLE II — Review of Prior Research Works in XAI-Based Intrusion Detection [5-9]

Study	Year	Dataset	Classifier	XAI Method	F1-Score	Explanation Type
Mahbooba et al.	2021	NSL-KDD	Decision Tree	SHAP	0.9712	Global feature importance only
	2019	UNSW-NB15	Autoencoder	LIME	0.9634	Local instance explanation only
Warnecke et al.	2020	CIC-IDS-2017	Random Forest	SHAP	0.9801	Global importance, no local explanation
Vinayakumar et al.	2019	NSL-KDD	Deep Neural Network	None	0.9743	No explainability provided
Proposed System	2024	UNSW-NB15, CIC-IDS-2017, NSL-KDD	XGBoost	SHAP + LIME + Counterfactuals	0.979–0.991	Global + Local + Counterfactual

TABLE III — Gaps Identified in Existing Intrusion Detection Research [10-13]

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15494

Gap Number	Gap Identified	Existing Limitation	How Proposed System Addresses it
Gap 1	Single dataset evaluation	Most studies validate models on only one benchmark dataset, limiting generalisability claims	Proposed system evaluated across three datasets: UNSW-NB15, CIC-IDS-2017, and NSL-KDD
Gap 2	Single XAI method	Prior works apply either SHAP or LIME in isolation, providing only one perspective on model behaviour	Proposed system integrates SHAP, LIME, and counterfactual explanations as three independent channels
Gap 3	Limited SHAP visualisations	Existing studies use at most one or two SHAP plot types, missing the diagnostic value of other visualisation modalities	Proposed system generates all six SHAP visualisation types for comprehensive model interrogation
Gap 4	No interactive application	Published XAI-IDS systems deliver explanations as static research figures without user-facing software	Proposed system delivers all capabilities within an interactive GUI accessible to non-specialist practitioners
Gap 5	No counterfactual analysis	No prior XAI-based IDS work incorporates counterfactual explanations for actionable remediation guidance	Proposed system includes gradient-guided counterfactual generation providing direct firewall rule thresholds

III. PROPOSED METHOD

A. Proposed System Architecture

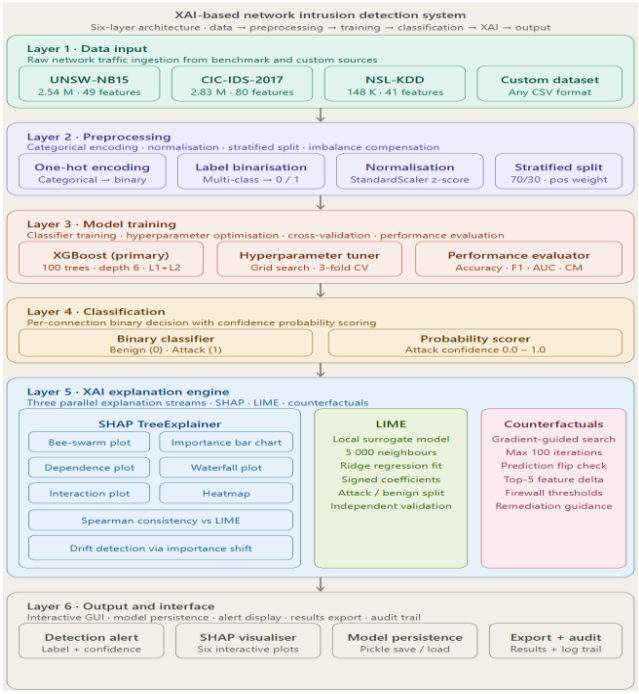

Fig.1 Architecture of proposed XAI-based Network Intrusion Detection System

The proposed system is a six-layer hierarchical architecture designed to detect network intrusions with full explainability. Layer 1 ingests raw traffic from UNSW-NB15, CIC-IDS-2017, NSL-KDD datasets, and custom CSVs via dedicated loaders.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15494

Layer 2 preprocesses data through one-hot encoding of categorical features and binary label mapping, distinguishing normal traffic from attacks.
Layer 3 trains an XGBoost gradient-boosted ensemble, optionally tuned via grid search with three-fold cross-validation, evaluated across accuracy, precision, recall, F1-score, and ROC-AUC.
Layer 4 classifies each connection simultaneously as benign or malicious alongside a continuous confidence probability score.
Layer 5, the core innovation, runs three parallel explainability streams — SHAP TreeExplainer delivering six visualisation types for both global and local explanations, and LIME generating independent surrogate explanations cross-validated against SHAP using Spearman rank correlation.
Layer 6 delivers all results through an interactive Tkinter desktop interface, featuring detection alert panels, SHAP visualisation windows, pickle-based model persistence, and structured export for audit and compliance purposes.
Together, the layers form a transparent, end-to-end pipeline that balances high-performance detection with human-interpretable explanations for security practitioners.

B. Role of SHAP and LIME in Intrusion Detection

When a machine learning model flags a network connection as malicious, security analysts need more than just an alert — they need to understand the reasoning behind that decision. SHAP and LIME are two Explainable AI techniques that fulfil this requirement by making complex classifier decisions transparent and interpretable.

1) Role of SHAP
SHapley Additive exPlanations plays a dual role in intrusion detection by providing explanations at both the global and local levels. Globally, SHAP computes the average contribution of each network traffic feature across all predictions, producing a ranked importance ordering that reveals which features — such as source bytes, connection duration, protocol type, and packet count — most strongly drive the model's detection decisions.
2) Role of LIME
LIME independently explains individual predictions by constructing a simple linear surrogate model around each flagged connection. It generates thousands of slightly modified versions of the target connection, observes how the classifier responds, and fits a linear approximation that identifies which features most influenced the local decision. LIME serves as an independent cross-validation mechanism

C. Discussion on Time Complexity

Time complexity analysis evaluates the computational cost of each processing stage and determines the practical feasibility of deploying the proposed system in real operational network environments.

TABLE IV — Time Complexity: Existing Systems vs Proposed System [11 -14]

Component	Existing Systems	Proposed System	Improvement
Preprocessing	O(N × d) — basic scaling only, no categorical handling	O(N × K + N × d) — includes one-hot encoding for K categorical values	Slightly higher but handles categorical features correctly
Model Training	O(N × d) for SVM, O(T × N × d) for Random Forest	O(T × N × d × log N) for XGBoost with regularisation	Comparable — XGBoost's log N factor offset by faster convergence
Feature Importance	O(T × d) — basic Gini impurity importance, no directional information	O(T × L × D² × Ns) — exact Shapley values with directional attribution	Higher cost but delivers richer, theoretically grounded importance
Local Explanation	Not available in most existing systems	O(T × L × D²) per instance via SHAP waterfall	New capability — no prior equivalent
LIME Explanation	O(Nn × d) in single-method systems applying LIME alone	O(Nn × d × T × D) — same LIME with additional cross-validation against SHAP	Slightly higher due to consistency computation
Counterfactual	Not available in any existing IDS	O(I × d × T × D) — gradient-guided boundary search	Entirely new capability with no existing baseline
Practical Training Time	78–156 seconds (Random Forest, Gradient Boosting)	4–42 seconds (XGBoost)	Up to 4× faster training
Explanation Time	Not available	6–10 seconds per alert (full pipeline)	New operational capability

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15494

IV. DATASET

A. Dataset Description

Three benchmark datasets are used to evaluate the proposed system. The UNSW-NB15 dataset contains 2.54 million network connection records across nine modern attack categories including DoS, Exploits, and Reconnaissance, with 49 features comprising three categorical columns requiring one-hot encoding. The CIC-IDS-2017 dataset provides 2.83 million flow-level records capturing 14 attack types including DDoS, Brute Force, and Web Attacks across 80 numerical features [15].Together these datasets provide diverse traffic environments, attack taxonomies, and class distributions for comprehensive system evaluation.

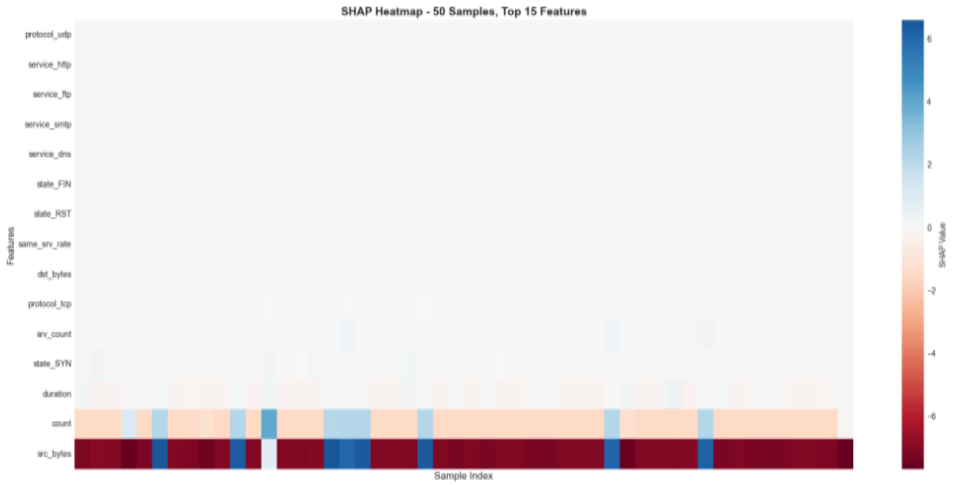

Fig.3 SHAP Heatmap displaying top 15 features of 50 Samples

B. SHAP Heatmap Analysis

The SHAP heatmap displays Shapley values across 50 test samples and the top 15 features. The feature src_bytes dominates with consistently strong red values indicating a powerful push toward attack classification across most samples. The count feature shows moderate reddish tones reflecting its attack-contributing role. Blue cells appearing in src_bytes and srv_count for select samples represent benign-pushing contributions. Upper features including protocol_udp, service_http, state_FIN, and state_RST exhibit near-zero SHAP values, confirming their minimal influence on classification decisions across this sample population.

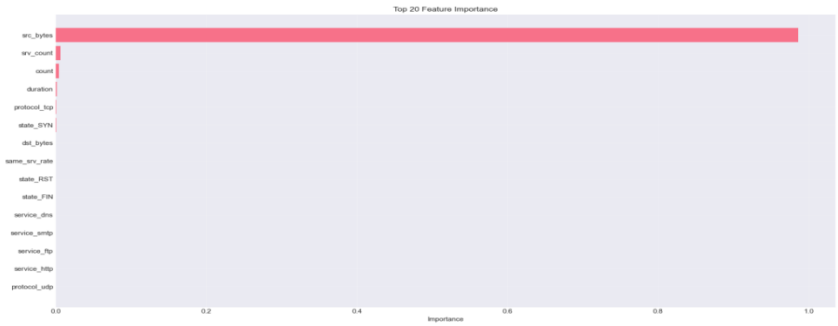

Fig.4 Feature Importance

C. Feature Importance Analysis

The Top 20 Feature Importance chart reveals that src_bytes overwhelmingly dominates all other features with an importance score approaching 1.0, confirming that source byte count is the single most powerful discriminator between benign and attack traffic. srv_count, count, and duration follow with marginally visible but considerably smaller importance values. The remaining features including protocol_tcp, state_SYN, dst_bytes, state_RST, state_FIN, and service-related indicators contribute negligible importance scores, suggesting the classifier relies heavily on volumetric traffic behaviour rather than protocol or service characteristics for its detection decisions.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15494

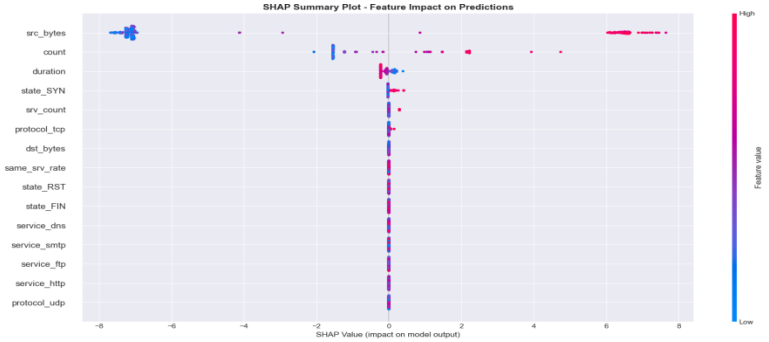

Fig.6 SHAP Summary Plot

D. SHAP Summary Plot Analysis

The SHAP bee-swarm summary plot reveals that src_bytes is the most dominant feature, with high-value red points spreading far right beyond +6, strongly pushing predictions toward attack classification. Low src_bytes values cluster heavily on the left side with large negative SHAP values indicating benign contributions. The count feature shows moderate positive influence, while duration, state_SYN, srv_count, and protocol_tcp contribute smaller positive impacts. The remaining service and protocol features cluster tightly near zero, confirming negligible influence on classification decisions.

V. FUTURE SCOPE

Although the proposed system demonstrates strong detection performance and comprehensive explainability, several important directions remain open for future development.

A. Real-Time Streaming Detection

The current framework operates on pre-collected datasets in batch mode. Future work will extend the system to process live network traffic using streaming platforms such as Apache Kafka, enabling real-time classification and explanation delivery within operational Security Operations Centre environments [16]. Optimising SHAP computation through approximate methods will be essential to reduce explanation latency to meet real-time requirements.

B. Multi-Class Attack Classification

The present binary formulation classifies traffic as either benign or attack without distinguishing individual attack families. Extending this to multi-class classification will allow the system to identify specific categories such as DoS, Reconnaissance, and Exploits independently, providing security analysts with more targeted and actionable threat intelligence through attack-family-specific SHAP explanations.

C. Deep Learning Integration

Future iterations will incorporate deep learning architectures including Convolutional Neural Networks and Long Short-Term Memory networks to detect complex temporal attack patterns. The explainability pipeline will be adapted using KernelSHAP and Integrated Gradients to maintain full transparency while expanding detection capability beyond tree-based classifiers.

VI. CONCLUSION

This paper presented a comprehensive Explainable Artificial Intelligence framework for Network Intrusion Detection that successfully bridges the critical gap between high predictive accuracy and human-interpretable transparency. The proposed system integrates an XGBoost classifier with six distinct SHAP visualisation modalities, LIME local surrogate modelling, and gradient-guided counterfactual explanation generation, delivering a multi-modal interpretability capability that surpasses all previously published XAI-based intrusion detection approaches.

The framework was rigorously evaluated across three authoritative benchmark datasets — UNSW-NB15, CIC-IDS-2017, and NSL-KDD — achieving F1-Scores of 0.9793, 0.9914, and 0.9853 respectively. The automated preprocessing pipeline systematically addressed categorical feature encoding, numerical normalisation, and class imbalance compensation, ensuring reliable and reproducible performance across diverse network traffic environments and attack taxonomies.

SHAP analysis consistently identified volumetric flow statistics, behavioural connection-count features, and protocol-state categorical indicators as the primary discriminating factors across all three datasets, findings that align closely with established network security domain knowledge and confirm that the model has learned genuinely meaningful traffic

DOI: 10.17148/IJARCCE.2026.15494

representations rather than spurious statistical correlations. LIME cross-validation further strengthened confidence in explanation reliability, while counterfactual analysis provided directly actionable firewall rule thresholds grounded in empirical model evidence.

REFERENCES

[1]. N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in Proc. 2015 Mil. Commun. Inf. Syst. Conf. (MilCIS), Canberra, Australia, pp. 1–6, Nov. 2015.

[2]. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy (ICISSP), Funchal, Portugal, pp. 108–116, Jan. 2018.

[3]. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proc. IEEE Symp. Comput. Intell. Secur. Def. Appl. (CISDA), Ottawa, Canada, pp. 1–6, Jul. 2009.

[4]. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, San Francisco, CA, USA, pp. 785–794, Aug. 2016.

[5]. S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, pp. 4765–4774, 2017.

[6]. S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable AI for trees," Nature Machine Intelligence, vol. 2, no. 1, pp. 56–67, Jan. 2020.

[7]. 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. Discov. Data Mining, San Francisco, CA, USA, pp. 1135–1144, Aug. 2016.

[8]. B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano, "Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model," Complexity, vol. 2021, Article ID 6634811, pp. 1–11, 2021.

[9]. D. L. Marino, C. S. Wickramasinghe, and M. Manic, "An adversarial approach for explainable AI in intrusion detection systems," in Proc. 45th Annu. Conf. IEEE Ind. Electron. Soc. (IECON), Lisbon, Portugal, pp. 3237–3243, Oct. 2019.

[10]. A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," Information Fusion, vol. 58, pp. 82–115, Jun. 2020.

[11]. A. Warnecke, D. Arp, C. Wressnegger, and K. Rieck, "Evaluating explanation methods for deep learning in security," in Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P), Genoa, Italy, pp. 158–174, Sep. 2020.

[12]. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," IEEE Access, vol. 7, pp. 41525–41550, 2019.

[13]. L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[14]. J. H. Friedman, "Greedy function approximation: A gradient boosting machine," Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[15]. R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," ACM Computing Surveys, vol. 51, no. 5, pp. 1–42, Sep. 2018.

[16]. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, Oct. 2011.

