frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 22 May 2025
DOI 10.3389/fcomp.2025.1520741

OPEN ACCESS

EDITED BY
Flaminia Luccio,
Ca' Foscari University of Venice, Italy

REVIEWED BY
Stefano Cirillo,
University of Salerno, Italy
Attaullah Buriro,
Ca' Foscari University of Venice, Italy

*CORRESPONDENCE
Ibidun Christiana Obagbuwa
✉ ibidun.obagbuwa@spu.ac.za

RECEIVED 31 October 2024
ACCEPTED 02 May 2025
PUBLISHED 22 May 2025

CITATION
Mohale VZ and Obagbuwa IC (2025) Evaluating machine learning-based intrusion detection systems with explainable AI: enhancing transparency and interpretability. *Front. Comput. Sci.* 7:1520741. doi: 10.3389/fcomp.2025.1520741

Evaluating machine learning-based intrusion detection systems with explainable AI: enhancing transparency and interpretability

Vincent Zibi Mohale and Ibidun Christiana Obagbuwa*

Department of Computer Science and Information Technology, Faculty of Natural and Applied Sciences, Sol Plaatje University, Kimberley, South Africa

Machine Learning (ML)-based Intrusion Detection Systems (IDS) are integral to securing modern IoT networks but often suffer from a lack of transparency, functioning as "black boxes" with opaque decision-making processes. This study enhances IDS by integrating Explainable Artificial Intelligence (XAI), improving interpretability and trustworthiness while maintaining high predictive performance. Using the UNSW-NB15 dataset, comprising over 2.5 million records and nine diverse attack types, we developed and evaluated multiple ML models, including Decision Trees, Multilayer Perceptron (MLP), XGBoost, Random Forest, CatBoost, Logistic Regression, and Gaussian Naive Bayes. By incorporating XAI techniques such as LIME, SHAP, and ELI5, we demonstrated that XAI-enhanced models provide actionable insights into feature importance and decision processes. The experimental results revealed that XGBoost and CatBoost achieved the highest accuracy of 87%, with a false positive rate of 0.07 and a false negative rate of 0.12. These models stood out for their superior performance and interpretability, highlighting key features such as Source-to-Destination Time-to-Live (sttl) and Destination Service Count (ct_srv_dst) as critical indicators of malicious activity. The study also underscores the methodological and empirical contributions of integrating XAI techniques with ML models, offering a balanced approach between accuracy and transparency. From a practical standpoint, this research equips human analysts with tools to better understand and trust IDS predictions, facilitating quicker responses to security threats. Compared to existing studies, this work bridges the gap between high-performing ML models and their real-world applicability by focusing on explainability. Future research directions include applying the proposed methodology to more complex datasets and exploring advancements in XAI techniques for broader cybersecurity challenges.

KEYWORDS

intrusion detection systems, transparency, explainable artificial intelligence, machine learning models, interpretability, explainability

1 Introduction

In today's digital world, cybersecurity has never been more crucial. With each passing day, our reliance on digital systems deepens, and so does the complexity and frequency of cyber threats. These threats pose significant risks to the integrity, confidentiality, and availability of information systems, necessitating robust and adaptive security measures (Almutairi et al., 2022). Traditional security methods, such as rule-based or signature-based intrusion detection

systems (IDS), serve as foundational components of cybersecurity. However, their reliance on predefined rules or signatures significantly limits their ability to detect novel or sophisticated attacks, leaving organizations vulnerable to emerging threats. As cyber threats grow more complex, the need for advanced, AI-driven solutions becomes imperative. Intrusion Detection Systems (IDS) have evolved significantly over the years. Initially grounded in simple rule-based mechanisms, they now incorporate cutting-edge Machine Learning (ML) algorithms to identify patterns, anomalies, and threats in network traffic (Mari et al., 2023). ML-based IDS have demonstrated remarkable success in detecting both known and unknown threats by learning from historical data and adapting to new attack strategies (Dini et al., 2023). Despite their potential, these systems face critical challenges. Chief among these is the "black-box" nature of many ML models, especially deep learning architectures, which obscures the reasoning behind their decisions (Chen et al., 2023). This lack of transparency hinders trust, adoption, and the effectiveness of these systems in real-world environments. For instance, consider a security analyst deploying a sophisticated ML-based IDS. When the system flags a potential intrusion, the analyst often lacks insight into why the alert was triggered. Was it a false positive, or does the alert reveal a previously unseen pattern? This ambiguity underscores the urgent need for transparent and interpretable IDS.

The concept of Explainable Artificial Intelligence (XAI) offers a promising solution to this challenge. XAI seeks to make the decision-making processes of ML models comprehensible to human users, enhancing trust and usability. Techniques like SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) have emerged as powerful tools for explaining complex ML models. By integrating XAI techniques with IDS, security analysts can gain actionable insights into detected threats, enabling faster and more informed responses (Arreche et al., 2024). However, the application of XAI in the realm of IDS remains in its infancy, with significant research challenges yet to be addressed. Recent studies highlight both the potential and the limitations of ML-based IDS. Deep learning models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have shown exceptional capabilities in identifying intricate patterns and anomalies that traditional systems fail to detect (Asharf et al., 2020; Khatkar et al., 2023; Mynuddin et al., 2024; Sevri and Karacan, 2023). However, their opacity often results in a trade-off between high accuracy and interpretability. Security analysts and stakeholders need transparency to understand not only the nature of detected threats but also the features and logic that contributed to the system's decisions (Barnard et al., 2024). Addressing these gaps is vital for improving model performance, meeting regulatory requirements, and building trust among users. Beyond technical implications, explainable IDS can provide value across multiple levels of application. At the enterprise level, these systems can enhance cybersecurity defences by offering granular insights into network vulnerabilities and potential breaches. Governments and regulatory bodies can also benefit from transparent IDS in the context of compliance and threat analysis. Small and medium-sized businesses, often operating with limited resources, stand to gain from the trust and efficiency provided by interpretable, user-friendly security solutions.

This study's primary contribution lies in its methodological and empirical advancements by integrating ML-based IDS with XAI techniques. Methodologically, it combines cutting-edge algorithms such as XGBoost, CatBoost, and Multilayer Perceptron (MLP) with interpretability tools like SHAP, LIME, and ELI5. These tools provide both global and local explanations for model predictions, facilitating a deeper understanding of model behaviour. This integration addresses critical challenges in cybersecurity by balancing high accuracy with interpretability, bridging the gap between advanced ML models and practical usability. From an empirical perspective, this study demonstrates the effectiveness of the proposed methodology on the UNSW-NB15 dataset, comprising over 2.5 million records and nine distinct attack types. The selection of ML models was driven by their complementary strengths. XGBoost and CatBoost were chosen for their superior handling of large-scale datasets and ability to identify complex patterns in imbalanced data. Decision Trees were included for their simplicity and inherent interpretability, while MLP was employed to capture non-linear relationships. Gaussian Naive Bayes provided a baseline comparison due to its efficiency in high-dimensional spaces. These models were systematically evaluated using metrics such as accuracy, precision, recall, F1-score, false positive rates, and false negative rates. Integrating XAI tools not only improved transparency but also enhanced practical insights, enabling human analysts to better understand and act on model predictions.

1.1 Research objective and questions

The primary objective of this study is to develop an effective and interpretable ML-based IDS by integrating XAI techniques, thereby addressing the trade-off between detection performance and model explainability. This research aims to:

1. Enhance the transparency and trustworthiness of ML-based IDS through the application of XAI techniques.
2. Evaluate the performance of various ML models in terms of accuracy, precision, recall, and interpretability using the UNSW-NB15 dataset.
3. Identify the key features that contribute to distinguishing between normal and malicious network traffic.

1.1.1 Research questions

- How can the integration of XAI techniques improve the interpretability of ML-based IDS?
- What are the performance trade-offs when combining high accuracy with model transparency?
- Which features are most critical for detecting cyber threats in network traffic?

Despite existing research on ML-based IDS and XAI, this study stands out by addressing the trade-off between accuracy and interpretability. Unlike prior works that focus solely on detection performance or model explainability, this study achieves both, presenting a balanced approach tailored for real-world cybersecurity challenges. Additionally, the research offers actionable insights into key features driving model predictions, such as Source-to-Destination Time-to-Live (sttl) and Destination Service Count (ct_srv_dst), which emerged as critical indicators of malicious activity. The following sections of this paper are structured as follows: Section 2 reviews existing literature, Section 3 describes the methodology

employed in this study, and Section 4 presents the experimental setup, results, and discussions. Finally, Section 5 concludes the paper with a summary of key findings and their implications for cybersecurity research.

2 Literature review

2.1 Intrusion detection systems (IDS)

Intrusion Detection Systems (IDS) are a cornerstone of cybersecurity, designed to identify and mitigate unauthorized access or malicious activities within network infrastructures. IDS continuously monitor network traffic and system activities to detect anomalies, issuing alerts when potential threats are identified (Asharf et al., 2020). Classical IDS research dates to Denning's early framework for anomaly detection, which laid the foundation for modern intrusion detection mechanisms. IDS can be broadly categorized into two main types:

1. Network Intrusion Detection Systems (NIDS): Monitor network traffic at key points to identify suspicious patterns. NIDS analyse protocol packets and compare them against predefined rules or learned behavioural models (Mari et al., 2023).
2. Host Intrusion Detection Systems (HIDS): Operate on individual devices, analysing logs, application behaviour, and system calls for malicious activities. HIDS can detect attacks that bypass network security measures, providing a more granular view of security threats (Asharf et al., 2020).

Together, NIDS and HIDS form a layered defence strategy, covering both network-wide and host-specific security threats. However, traditional IDS methods rely heavily on rule-based detection, making them ineffective against novel or evolving cyber threats.

2.1.1 Network and host intrusion detection systems

Network Intrusion Detection Systems (NIDS) are designed to monitor entire network segments, capturing and analysing protocol packets to detect malicious traffic. Positioned at strategic points within the network, NIDS provide a holistic view of network activity, identifying threats based on predefined rules or behavioural patterns. When potential threats are detected, NIDS log the events and issue alerts for further investigation. Host Intrusion Detection Systems (HIDS), in contrast, operate at the individual device level. By monitoring inbound and outbound packets for a specific host, HIDS detect anomalous behaviour or policy violations. These systems excel at uncovering threats targeting individual endpoints, providing detailed insights into host-specific activities. Together, NIDS and HIDS form a complementary defence strategy, addressing threats at both macro and micro levels within a network. With the integration of Machine Learning (ML), IDS capabilities have been significantly enhanced. ML algorithms analyse vast amounts of historical data to detect intricate patterns indicative of malicious activities. This approach reduces reliance on static rules, enabling IDS to adapt to evolving threats. By incorporating ML, IDS achieve improved detection accuracy, reduced false positives, and the ability to identify previously unseen attack vectors.

2.1.2 Intrusion detection methods and explainable artificial intelligence (XAI)

Machine learning (ML) and deep learning (DL) approaches have become integral to intrusion detection systems (IDS) due to their capacity to process large datasets and complex data structures, yielding impressive performance in detecting network intrusions (Varanasi and Razia, 2022). Traditional ML methods—including Artificial Neural Networks (ANN), Support Vector Machines (SVM), fuzzy approaches, swarm intelligence, and evolutionary computation techniques—have been extensively employed for IDS. For example, ANNs, inspired by the structure of the human brain with interconnected layers of neurons, can learn complex patterns from data and have proven effective for intrusion detection tasks (Khatkar et al., 2023). Similarly, SVMs excel in high-dimensional spaces by classifying data into distinct categories based on training examples. Fuzzy logic addresses uncertainty and imprecision in network traffic, while swarm intelligence algorithms (such as particle swarm optimization and ant colony optimization) and evolutionary computation methods (including genetic algorithms and genetic programming) contribute to feature selection, rule generation, and parameter optimization. Despite their effectiveness, comprehensive comparative studies across various scenarios and datasets remain limited (Alghazali and Hanoosh, 2022). Advances in DL—particularly through Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs)—have further enhanced detection capabilities by learning hierarchical representations of raw data, although further research is needed to integrate these methods seamlessly with traditional ML techniques for improved performance and interpretability (Neupane et al., 2022).

In parallel with these methodological advancements, Explainable Artificial Intelligence (XAI) has emerged as a significant enhancement for ML-based IDS, aiming to improve transparency, interpretability, and trustworthiness (Sivamohan and Sridhar, 2023; Hariharan et al., 2022; Barnard et al., 2024). In the IDS context, XAI techniques provide critical insights into the decision-making processes of complex models, enabling security analysts to understand, validate, and trust detection outputs. One of the primary goals of XAI is to render the inner workings of AI systems visible and intelligible, thereby elucidating the factors and features that drive detection alerts (Hariharan et al., 2022). This transparency not only fosters greater model accountability and trust—by allowing users to verify the rationale behind decisions (Barnard et al., 2024)—but also aids in identifying biases and vulnerabilities within the models, such as those arising from skewed training data or adversarial threats (Mahbooba et al., 2021). Moreover, XAI promotes enhanced human-AI collaboration by offering clear, actionable explanations that help analysts prioritize alerts and respond effectively, ultimately supporting regulatory compliance and ethical AI practices in cybersecurity (Khatkar et al., 2023).

Among the XAI methods applied in IDS, Local Interpretable Model-Agnostic Explanations (LIME) generates local, simplified models that approximate the behaviour of complex, black-box models, providing instance-specific interpretations of predictions. SHapley Additive exPlanations (SHAP) offers a unified framework for quantifying the contribution of each feature to a model's output,

thereby enabling a granular understanding of feature importance and interaction effects. Additionally, tools like Explain Like I'm 5 (ELI5) deliver user-friendly interfaces that illustrate feature and permutation importance in an accessible manner. By combining the robust detection capabilities of diverse ML and DL approaches with the transparency afforded by XAI, future IDS can be developed to be more robust, accurate, and interpretable. This integration not only enhances the practical effectiveness of intrusion detection but also builds trust and accountability, ultimately providing a more resilient defence against evolving cybersecurity threats.

2.1.3 Key ML techniques in IDS

- Artificial Neural Networks (ANNs): ANNs mimic the human brain's structure, comprising interconnected nodes organized into layers. They excel in learning non-linear patterns from data, making them suitable for detecting sophisticated intrusions (Khatkar et al., 2023).
- Support Vector Machines (SVMs): These supervised models classify data into distinct categories by finding an optimal hyperplane. SVMs are effective in high-dimensional spaces and are commonly used for binary classification tasks in intrusion detection.
- Fuzzy Approaches: Leveraging fuzzy logic, these methods handle ambiguous or imprecise data, making them effective for analysing uncertain or noisy network traffic.
- Swarm Intelligence: Inspired by social insect behaviour, algorithms like Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) optimize feature selection and rule generation for IDS (Balyan et al., 2022).
- Evolutionary Computation: Techniques such as Genetic Algorithms (GA) utilize natural selection principles to optimize IDS parameters and improve detection capabilities.

2.1.3.1 Recent advancements in deep learning for IDS

Deep Learning (DL) has emerged as a powerful tool for IDS, with architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) achieving state-of-the-art performance. CNNs are particularly effective at extracting hierarchical features from network traffic data, while RNNs, with their ability to model temporal dependencies, excel at analysing sequential data such as time-series logs (Asharf et al., 2020; Khatkar et al., 2023). Hybrid models combining CNNs and RNNs have also shown promise, leveraging the strengths of both architectures for enhanced accuracy and detection of complex attack patterns (Mynuddin et al., 2024).

Despite their superior performance, DL-based IDS face challenges in interpretability. The intricate architectures of these models often operate as black boxes, making it difficult for analysts to understand their decision-making processes. This trade-off between accuracy and interpretability remains a critical area for further research and development (Barnard et al., 2024).

2.1.3.2 Limitations and challenges

While ML-based IDS have demonstrated high detection accuracy, several challenges persist:

- False Positives: Despite improvements, many ML-based systems generate a significant number of false alarms, burdening analysts.
- Scalability: The computational demands of ML algorithms, especially DL models, can hinder real-time detection in large-scale networks.
- Interpretability: Complex ML models often lack transparency, making it difficult for security analysts to validate or trust their outputs.

2.1.4 Intrusion detection system with machine learning and multiple datasets

An upgraded intrusion detection system (IDS) based on machine learning (ML) and hyperparameter tuning has shown promise in boosting model accuracy and efficacy (Yedukondalu et al., 2021; Tsukerman, 2020). This enhanced IDS leverages multiple datasets to improve model accuracy, providing a more robust defense against cyber threats. Intrusion detection systems are critical for protecting networks and data by detecting and responding to unauthorized access or malicious activity (Chatterjee, 2021). Given the rising complexity and diversity of cyber threats, there is an increasing need for IDS systems that can effectively identify and mitigate these attacks. ML has emerged as a powerful method for enhancing IDS capabilities by automating the detection of patterns and anomalies in network traffic data (Cuelogic Technologies, 2019). By analysing vast amounts of data and uncovering subtle patterns indicative of malicious behaviour, ML-based IDS can improve detection accuracy and reduce false positives. Hyperparameter tuning is a vital step in developing ML models, as it involves optimizing the parameters that control the model's behaviour, to enhance its performance (Othman et al., 2018). By systematically exploring the hyperparameter space and selecting the best set of parameters, hyperparameter tuning can significantly boost the effectiveness of ML-based IDS. The enhanced IDS described in the literature utilizes both ML and hyperparameter tuning to improve its accuracy and efficacy. By fine-tuning the parameters of the ML models, the IDS can achieve better detection accuracy and lower false positive rates, resulting in more reliable and effective intrusion detection (Almutairi et al., 2022).

Moreover, the IDS incorporates multiple datasets to enhance the accuracy of the models. Training ML models on diverse datasets representing different types of network traffic and attack scenarios allows the IDS to generalize better and detect a broader range of threats (Chatterjee, 2021). This approach increases the robustness and resilience of the IDS against evolving cyber threats, ensuring it remains effective in real-world deployment scenarios. Overall, the improved IDS described in the literature represents a significant advancement in intrusion detection technology. By leveraging the power of ML and hyperparameter tuning and incorporating multiple datasets, this IDS can achieve superior performance in terms of accuracy and efficacy. This makes it a valuable tool for enhancing cybersecurity defences in today's complex threat landscape (Barnard et al., 2024). Integrating these advanced techniques ensures that the IDS is well-equipped to handle the sophisticated and constantly changing nature of cyber threats, providing a more secure and reliable network environment.

2.1.4.1 Hyperparameter tuning

Optimizing model hyperparameters, such as learning rates and regularization coefficients, significantly enhances performance. This systematic exploration of the hyperparameter space ensures that ML models achieve optimal detection rates while minimizing false positives (Othman et al., 2018).

2.1.4.2 Multiple datasets

Using diverse datasets enables IDS to detect a broader range of threats by exposing models to various attack scenarios. However, challenges such as data imbalance and domain adaptation must be addressed to maximize the utility of this approach.

2.1.5 Interpretable and explainable intrusion detection systems

Interpretable and explainable intrusion detection systems (IDS) are becoming increasingly important in the cybersecurity sector because they provide insights into the decision-making process behind detection mechanisms (Ahmad et al., 2021). By making the reasoning behind intrusion detection alerts transparent and understandable to human analysts, these systems enhance trust and facilitate more effective incident response. One innovative approach to creating an interpretable and explainable IDS involves combining expert-written rules with dynamic information generated by a decision tree algorithm (Barnard et al., 2022). This hybrid method leverages artificial intelligence technologies for more effective and sustainable security (Alghazali and Hanoosh, 2022). The system integrates the strengths of expert-written rules, which are based on domain knowledge and can capture specific patterns of known attacks, with the flexibility and adaptability of a decision tree algorithm (Mahbooba et al., 2021). Decision trees are inherently interpretable models that recursively partition the feature space into regions, making them well-suited for explaining the logic behind intrusion detection decisions. In this hybrid approach, expert-written rules serve as the initial detection rules, providing a foundation for identifying common attack patterns (Wali and Khan, 2021). As new data becomes available and the system encounters previously unseen threats, the decision tree algorithm dynamically generates additional rules based on evolving patterns in the data (Mahbooba et al., 2021). This continuous learning process allows the IDS to adapt to emerging threats and evolving attack techniques, ensuring robust and up-to-date detection capabilities.

By combining expert knowledge with data-driven learning, this hybrid approach balances interpretability and effectiveness in intrusion detection. Human analysts can easily understand and validate the rules generated by the decision tree algorithm, gaining insights into the factors influencing intrusion detection decisions (Barnard et al., 2024). At the same time, the system leverages the power of artificial intelligence to automate the detection process and keep pace with the rapidly evolving threat landscape. Moreover, by providing explanations for intrusion detection alerts, the system enables human analysts to verify the validity of alerts, investigate the root causes of detected anomalies, and take appropriate remediation actions (Barnard et al., 2024). This enhances the overall effectiveness of cybersecurity operations and enables organizations to respond more quickly and decisively to security incidents. This hybrid IDS approach improves detection accuracy by integrating interpretable and explainable elements. It ensures that the decision-making process remains transparent and comprehensible to those tasked with protecting critical network infrastructure.

2.2 Challenges in applying XAI to IDS

- Complexity of Cybersecurity Data: High-dimensional, dynamic, and often imbalanced data makes interpretation challenging.
- Trade-offs with Performance: Simplifying models for interpretability may reduce accuracy.
- Scalability: Many XAI methods are computationally intensive, complicating real-time applications in large-scale networks.

Table 1 shows the comparison of IDS approaches, looking at their advantages and limitations.

The integration of ML and XAI into IDS represents a significant advancement in cybersecurity. However, the trade-offs between accuracy, interpretability, and scalability must be addressed to ensure these systems are both effective and practical. Future research should focus on hybrid approaches combining ML techniques and XAI for robust, transparent, and scalable intrusion detection.

2.2.1 Machine learning models

2.2.1.1 Decision trees

These models are popular in IDS due to their simplicity, interpretability, and ability to generate human-readable rules for classifying network traffic. However, they may suffer from overfitting, especially with complex datasets. Researchers have explored techniques like pruning and ensemble methods to mitigate these issues and improve the robustness of Decision Trees in IDS applications (Mahbooba et al., 2021).

2.2.1.2 Gaussian naive Bayes

A probabilistic classifier based on Bayes' theorem with the assumption of feature independence. It is efficient and performs well with small datasets, making it suitable for real-time intrusion detection scenarios. Its probabilistic nature provides clear explanations of how predictions are made.

2.2.1.3 Multilayer perceptron (MLP)

A type of neural network composed of multiple layers of nodes, which allows it to learn complex patterns in data. MLPs are useful in IDS for detecting sophisticated attack patterns but can be challenging to interpret without XAI techniques.

2.2.1.4 CatBoost

A gradient boosting algorithm that is particularly effective for categorical data, making it suitable for IDS where network traffic features are often categorical. CatBoost offers high performance and is designed to handle the complexities of real-world data. However, its complexity may pose challenges regarding interpretability and transparency.

TABLE 1 Tabular comparison of IDS approaches.

Approach	Advantages	Limitations
Traditional IDS	Simple and interpretable	Limited to known attack signatures
ML-based IDS	High accuracy, adaptive to new threats	High false positive rates, opaque models
DL-based IDS	Superior pattern recognition	High computational cost, black-box nature
Explainable IDS with XAI	Transparency, trust, regulatory compliance	Trade-off between accuracy and simplicity

2.2.2 Challenges of black-box models

Despite their often high accuracy in predictions, Black-box models pose several challenges in intrusion detection systems (IDS). These challenges primarily arise from these models' intrinsic opacity and lack of interpretability, which restrict security analysts' ability to comprehend, validate, and trust the system's judgments (Mynuddin et al., 2024).

2.2.3 Opaque decision-making process

Black-box models, such as deep neural networks (DNNs) and ensemble methods, operate by learning complex patterns and relationships from large volumes of data (Neupane et al., 2022). While these models may achieve impressive performance in detecting intrusions, the decision-making process underlying their predictions is often opaque and difficult to decipher. Security analysts cannot determine which features or factors are driving the model's decisions, making it challenging to validate the correctness of detection alerts (Neupane et al., 2022; Mynuddin et al., 2024).

2.2.4 Limited human understanding

The lack of interpretability in black-box models significantly hinders human understanding. Security analysts evaluating the validity of intrusion detection alerts and taking appropriate remedial actions require insights into the rationale behind the system's decisions (Ogino, 2015). Analysts may struggle to trust the system's outputs and hesitate to act on detection alarms without comprehensive explanations of how the model makes its predictions.

2.2.5 Difficulty in debugging and troubleshooting

When black-box models produce unexpected or erroneous results, diagnosing the underlying causes of these errors can be challenging. Without visibility into the internal workings of the model, identifying and addressing issues such as data biases, model drift, or adversarial attacks becomes a daunting task (Almutairi et al., 2022). As a result, black-box models may exhibit sub-optimal performance or vulnerabilities that go unnoticed, posing risks to the security and integrity of the system.

2.2.6 Regulatory compliance and accountability

In many industries, regulatory requirements mandate transparency and accountability in decision-making processes, especially for sensitive tasks such as intrusion detection. Black-box models may fail to meet these requirements, as they lack the transparency necessary to provide auditable explanations of their decisions. This can lead to compliance challenges and legal liabilities, particularly in highly regulated sectors such as finance, healthcare, and government (Sevri and Karacan, 2023).

2.2.7 Resistance to adoption and integration

The opacity of black-box models can lead to resistance to their adoption and integration into existing cybersecurity infrastructure. Security stakeholders, including analysts, administrators, and executives, may be reluctant to rely on systems whose decision-making processes they do not fully understand or trust (Othman et al., 2018). This resistance can impede the deployment of advanced intrusion detection solutions and limit the effectiveness of cybersecurity defences in mitigating emerging threats.

In conclusion, while black-box models offer high accuracy and robust capabilities for intrusion detection, their lack of transparency and interpretability presents significant challenges. Addressing these issues is crucial for enhancing the trustworthiness, accountability, and effectiveness of IDS. As explainable artificial intelligence (XAI) advances, there is potential to overcome these challenges by developing methods that provide clear and actionable explanations for the decisions made by black-box models, thereby improving their integration into cybersecurity practices.

2.3 Comparative analysis table

Table 2 below summarises recent works related to ML-based IDS integrated with XAI, highlighting their methodologies, datasets used, key findings, and how the current study differs or advances the field.

3 Materials and methods

This study employs a comprehensive methodology to develop enhanced Intrusion Detection Systems (IDS) by integrating data collection, preprocessing, modelling, and evaluation within a unified framework. The UNSW-NB15 dataset serves as the foundation for this research, with raw network traffic data undergoing systematic preprocessing—such as normalization, encoding, and scaling—to transform it into a format suitable for effective IDS modelling. A diverse range of machine learning models, including traditional approaches like Decision Trees and Gaussian Naive Bayes as well as advanced techniques such as CatBoost, XGBoost, and Multilayer Perceptron (MLP), are applied to assess and improve detection capabilities. To ensure transparency in the decision-making process, interpretability techniques including SHAP (Shapley Additive Explanations), LIME (Local Interpretable Model-agnostic Explanations), and ELI5 are incorporated, thereby providing insights into model predictions alongside achieving high predictive accuracy. Performance is quantitatively evaluated using metrics such as accuracy, precision, recall, F1 score, and ROC curve analysis, while the qualitative aspects of model interpretability are also carefully considered, given their critical role in the practical adoption and usability of IDS in real-world cybersecurity contexts.

By triangulating diverse data sources, preprocessing strategies, modelling approaches, and evaluation methods, this research not only enhances the credibility, reliability, and validity of its findings but also establishes a robust framework for integrating machine learning with interpretable models. This structured approach enables a detailed examination of how each phase—from data transformation and model development to performance assessment—contributes to the overall effectiveness of IDS, thereby offering a comprehensive roadmap for deploying and understanding ML-based intrusion detection systems (as illustrated in Figure 1).

3.1 Data preprocessing and feature engineering

The preprocessing stage ensures the dataset is cleaned, transformed, and optimized for ML model development. Key steps include:

- Data Cleaning: Duplicate entries were removed to prevent data redundancy and biases.

TABLE 2 Comparison of recent works on ML-based IDS with XAI.

Study	Methodology	Dataset	Key findings	Novelty compared to current study
Explainable AI-based Innovative Hybrid Ensemble Model for Intrusion Detection Systems (Wang et al., 2024)	Hybrid ensemble model incorporating XAI for IDS	Not specified	Improved transparency and interpretability in IDS	Current study utilizes specific ML algorithms (XGBoost, CatBoost, MLP) with SHAP, LIME, and ELI5 for detailed explanations
Explainable AI for Intrusion Detection Systems: LIME and SHAP Applicability on Multi-Layer Perceptron (Wang et al., 2023)	Application of LIME and SHAP in ML-based IDS	Not specified	Enhanced transparency and interpretability of IDS decisions	Current study combines multiple XAI tools (SHAP, LIME, ELI5) for both global and local explanations in IDS
XAI-IDS: Toward Proposing an Explainable Artificial Intelligence Framework for Enhancing Network Intrusion Detection Systems (Arreche et al., 2024)	End-to-end XAI framework tailored for network intrusion detection	Not specified	Improved interpretability of AI models in network intrusion detection tasks	Current study applies XAI techniques to specific ML models and provides actionable insights into key features
Explainable Intrusion Detection Systems Using Competitive Learning Techniques (Ables et al., 2023)	Competitive learning algorithms (e.g., Self-Organizing Maps) for explainable IDS	NSL-KDD, CIC-IDS-2017	Achieved accuracies slightly lower than error-based learning models but with enhanced explainability	Current study focuses on different ML algorithms and XAI tools, applied to the UNSW-NB15 dataset
Explainable Artificial Intelligence for Intrusion Detection Systems: A Comprehensive Survey (Patel and Shah, 2023)	Comprehensive survey on XAI techniques applied to IDS	Various	Provided an extensive overview of XAI methods and their applicability in enhancing IDS transparency	Current study offers empirical analysis by applying specific XAI techniques to selected ML models on the UNSW-NB15 dataset

- Feature Selection: The dataset's 49 features were analysed for relevance using techniques like Recursive Feature Elimination (RFE) and correlation analysis. Features with low variance or high multicollinearity were excluded to improve model performance and reduce overfitting.
- Normalization and Scaling: Numerical features were normalized to a standard range using Min-Max scaling. This step ensures that features with larger ranges do not dominate those with smaller ranges during model training.
- Encoding: Categorical features, such as protocol types and service labels, were transformed into numerical representations using one-hot encoding and label encoding, as appropriate.
- Balancing the Dataset: Class imbalances were addressed using Synthetic Minority Oversampling Technique (SMOTE), ensuring that all attack types were adequately represented in the training data.

3.2 Rationale for model selection

The selection of ML models was driven by their complementary strengths and relevance to intrusion detection tasks:

- Decision Trees: Chosen for their simplicity and interpretability, Decision Trees provide clear decision paths that are easily understood by security analysts. They are well-suited for initial exploration of feature importance.
- Gaussian Naive Bayes: This probabilistic model is effective for high-dimensional data and provides fast classification. Its assumptions of feature independence align well with certain network traffic scenarios, making it a valuable baseline.
- CatBoost and XGBoost: These gradient boosting algorithms are powerful for handling complex data patterns. CatBoost's automatic handling of categorical features and XGBoost's efficiency in large-scale datasets make them ideal for high-performance IDS applications.
- Multilayer Perceptron (MLP): As a neural network-based model, MLP excels at capturing non-linear relationships in data. Its adaptability makes it suitable for detecting sophisticated intrusion patterns.

The combination of these models ensures a balance between interpretability, computational efficiency, and detection accuracy. By leveraging their unique strengths, the study aims to maximize the overall performance of IDS.

3.3 Implementation of XAI techniques

To ensure transparency and interpretability, the following XAI techniques were integrated into the ML models:

- SHAP (Shapley Additive exPlanations): SHAP values were computed to quantify the contribution of each feature to model

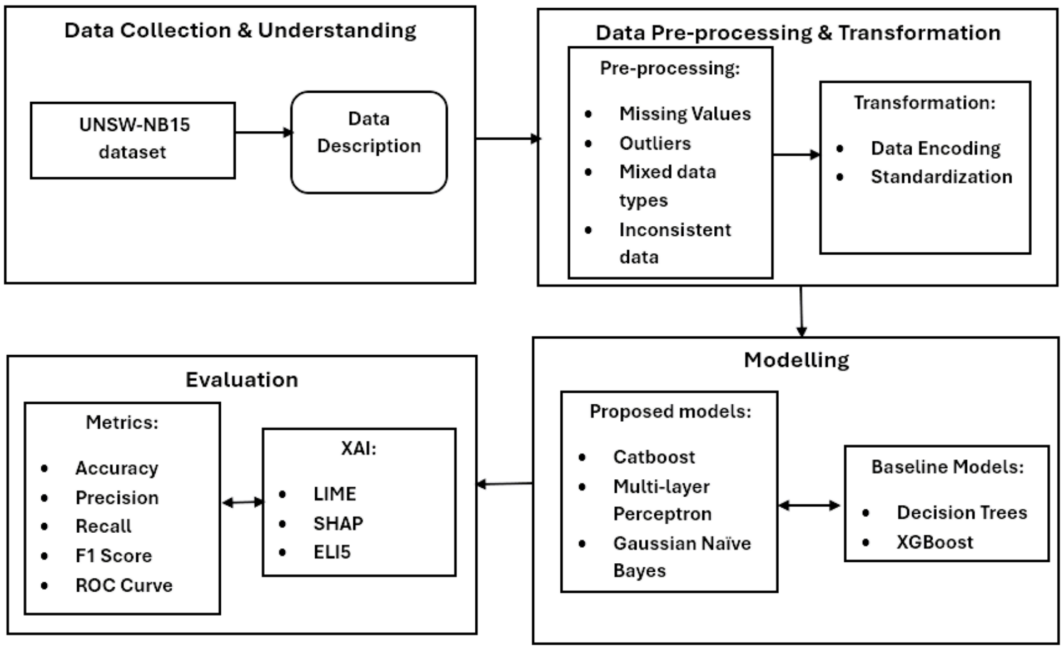


FIGURE 1
Intrusion detection system framework.

predictions. This global interpretability technique provided insights into how individual features influenced classification decisions across the dataset.

- LIME (Local Interpretable Model-agnostic Explanations): LIME was employed to generate localized explanations for specific predictions. By approximating the behaviour of complex models with simpler interpretable models, LIME enabled analysts to understand why certain instances were classified as attacks.
- ELI5: This tool provided intuitive visualizations of feature importance and model behaviour. Using techniques like permutation importance, ELI5 highlighted the most critical features driving model predictions, making it easier for analysts to validate results.

The integration of these techniques involved post-hoc analysis, ensuring that the explanations did not interfere with the models' predictive performance. For example, SHAP and LIME were applied after model training to analyse predictions on test data, while ELI5 offered feature importance visualizations during and after training.

3.4 Datasets

The initial step in our research to enhance Intrusion Detection Systems (IDS) with machine learning and interpretable models involves acquiring the UNSW-NB15 dataset. Recognized as a benchmark in cybersecurity research, this dataset forms the empirical backbone for our model development. It is sourced from reputable platforms, including the Australian Centre for Cyber Security (ACCS) and the UNSWNB15 GitHub repository, ensuring the integrity and reliability of the data. The UNSW-NB15 dataset is meticulously curated, encompassing a broad spectrum of network behaviours encapsulated in 49 distinct features, detailed in the UNSWNB15-features.csv file. The comprehensive dataset includes 2,540,044 records across four CSV files, representing various normal activities and synthetic attack behaviours. Each feature captures specific aspects of network traffic, which is vital for distinguishing between benign and malicious activities.

The dataset categorizes network activities into 10 distinct types of attacks—such as Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Normal attacks and Worms—through its class labels. These labels are crucial for training our models to accurately identify and respond to various security threats, providing a robust foundation for enhancing IDS capabilities with advanced machine-learning techniques and A thorough understanding of the class distribution is essential for assessing the performance of ML models in an IDS context (Table 3).

3.4.1 Models

This study implemented several machine learning models to enhance Intrusion Detection Systems (IDS) using the UNSW-NB15 dataset. The models include Decision Trees, CatBoost, Gaussian Naive Bayes, XGBoost, and Multilayer Perceptron (MLP). The Decision Tree and XGBoost models are selected as benchmarks to facilitate robust performance comparisons across the different models. These models are well-suited for handling the complexities of network traffic data, offering a strong baseline for evaluating the performance of more advanced models like CatBoost and Multilayer Perceptron. In addition to these models, interpretability techniques such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) were integrated to ensure that the models achieve not only high predictive accuracy but also provide transparent, interpretable insights into their decision-making processes. These

TABLE 3 UNSW-NB15 attack types and description.

Attack type	Description	Number of records
Fuzzers	Attempts to discover security vulnerabilities by injecting massive amounts of random data (fuzz) into the system to cause it to crash or behave unexpectedly.	24,246
Analysis	Involves reconnaissance and probing attacks such as port scanning and IP sweeps aimed at gathering information about the network for future attacks.	2,677
Backdoors	Involves the use of unauthorized access points that bypass normal authentication mechanisms to gain control over the system.	1,746
DoS	Denial of Service (DoS) attacks aim to make a machine or network resource unavailable to its intended users by overwhelming it with a flood of illegitimate requests.	16,353
Exploits	Involves the use of known vulnerabilities in software to gain unauthorized access or escalate privileges on the affected system.	44,525
Generic	Refers to attacks that are designed to work across multiple platforms or software applications, often targeting encryption weaknesses.	215,481
Reconnaissance	Involves information-gathering activities such as network scanning to identify active hosts and open ports within a network.	13,987
Shellcode	Refers to a small piece of code used as the payload in the exploitation of a software vulnerability, often to execute arbitrary commands on the affected system.	1,511
Worms	Self-replicating malware that spreads across computers in a network without user intervention, often consuming bandwidth and overloading systems.	174
Normal	Represents legitimate network traffic that does not exhibit any malicious behaviour.	2,218,761

interpretability techniques are critical for addressing machine learning models' "black-box" nature, making the enhanced IDS more trustworthy and actionable in real-world cybersecurity contexts.

3.4.1.1 Decision trees

Decision Trees are a non-parametric supervised learning method for classification and regression tasks. They work by splitting the data into subsets based on the value of the input features, recursively partitioning the feature space.

The decision at each node is based on the impurity, calculated using the Gini impurity (for classification) or variance reduction (for regression), as can be seen in Equation 1:

$$I(t) = 1 - \sum_{i=1}^{c} p(i|t)^2 \quad (1)$$

In this equation:

- $I(t)$ is the impurity of node t.
- $(i|t)$ is the proportion of samples that belong to class c at node t.

How the impurity value is used for decision making:

1. Initial Impurity Calculation: At each split, the algorithm calculates the impurity of a node using the Gini formula.
2. Finding the Best Split: The model evaluates different features and thresholds to see which split leads to the largest reduction in impurity (also called the Gini Gain).
3. Splitting the Node: The feature and threshold that provide the lowest impurity after splitting are selected.
4. Repeating Until Stopping Criteria are Met: The process continues recursively until a stopping condition is met.

3.4.1.1.1 Gini impurity

For classification tasks, the Gini impurity is used to measure the impurity of a node. It quantifies the likelihood of an incorrect classification of a randomly chosen element if it was randomly labelled according to the distribution of labels in the node.

3.4.1.1.2 Variance reduction

For regression tasks, variance reduction is used instead of Gini impurity. It measures the reduction in the variance of the target variable after splitting the data based on an input feature. The goal is to minimize the variance within each subset.

3.4.1.1.3 Recursive partitioning

The process of building a decision tree involves recursively partitioning the feature space:

1. Select the Best Split: At each node, choose the feature and threshold that result in the highest reduction in impurity.
2. Create Sub-nodes: Split the data into subsets based on the chosen feature and threshold.
3. Repeat: Recursively apply the same process to each subset until a stopping criterion is met (e.g., maximum depth, minimum number of samples per node).

Decision Trees are intuitive and easy to interpret, making them popular for many machine learning tasks.

3.4.1.2 XGBoost (extreme gradient boosting)

XGBoost is a decision-tree-based ensemble machine learning algorithm that uses a gradient-boosting framework. It is known for its

efficiency and performance in handling large datasets. The objective function for XGBoost is shown in Equation 2 as:

$$Obj(\theta) = \sum_{i=1}^{n} \ell(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k) \quad (2)$$

In this equation:

- $Obj(\theta)$ is the objective function that XGBoost aims to minimize.
- $\ell(\hat{y}_i, y_i)$ is a differentiable convex loss function measuring the difference between the prediction $\hat{y}_i$, and the actual value y_i.
- $\Omega(f_k)$ is the regularization term to control the complexity of the model.

The objective function consists of two main components:

1. Loss Function: This part of the objective function measures how well the model's predictions match the actual values. Common loss functions include Mean Squared Error (MSE) for regression tasks and Log Loss for classification tasks.
2. Regularization Term: This part of the objective function helps control the model's complexity by adding a penalty for more complex models. This helps to prevent overfitting and ensures that the model generalizes well to new data.

XGBoost's efficiency and performance come from its ability to handle large datasets and implement various optimization techniques, such as parallel processing and tree pruning. These features make it a popular choice for many machine learning tasks.

3.4.1.3 CatBoost

CatBoost (Categorical Boosting) is a gradient-boosting algorithm that handles categorical variables efficiently. It is known for delivering high performance with minimal data preprocessing. The objective function looks the same for both CatBoost and XGBoost, but their internal implementations and how they optimize for performance differ. The objective function for CatBoost is shown in Equation 3:

$$Obj(\theta) = \sum_{i=1}^{n} \ell(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k) \quad (3)$$

In this equation:

- $Obj(\theta)$ is the objective function that CatBoost aims to minimize.
- $\ell(\hat{y}_i, y_i)$ is the loss function, which measures the difference between the prediction $\hat{y}_i$, and the actual value y_i.
- $\Omega(f_k)$ is the regularization term, which penalises the complexity of the model to prevent overfitting.

The objective function consists of two main components:

1. Loss Function: This part of the objective function measures how well the model's predictions match the actual values. Common loss functions include Mean Squared Error (MSE) for regression tasks and Log Loss for classification tasks.
2. Regularization Term: This part of the objective function helps to control the complexity of the model by adding a penalty for more complex models. This helps to prevent overfitting and ensures that the model generalizes well to new data.

CatBoost's efficiency in handling categorical variables comes from its ability to process these variables directly without needing extensive preprocessing, such as one-hot encoding. This makes it particularly useful for datasets with many categorical features.

3.4.1.4 Gaussian Naive Bayes

Gaussian Naive Bayes is a probabilistic classifier that assumes the features follow a Gaussian distribution and are conditionally independent given the class label, the probability of a class y given the features X is shown in Equation 4:

$$P(y|X) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(X)} \quad (4)$$

In this equation:

- $P(y|X)$ is the posterior probability of class y given the features X.
- $P(y)$ is the prior probability of class y.
- $P(x_i|y)$ is the likelihood of feature x_i given class y.
- $P(X)$ is the marginal likelihood of the features X.

The Gaussian Naive Bayes classifier assumes that each feature x_i follows a Gaussian (normal) distribution. The likelihood $P(x_i|y)$ can be computed using the Gaussian probability density function as shown in Equation 5:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma^2{}_y}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma^2{}_y}\right) \quad (5)$$

Where:

- μ_y is the mean of the feature x_i for class y.
- σ_y is the standard deviation of the feature x_i for class y.

The assumption of conditional independence means that the joint probability $P(X|y)$ can be decomposed into the product of individual probabilities $P(x_i|y)$. This simplifies the computation and makes the classifier efficient.

3.4.1.5 Multilayer perceptron (MLP)

A Multilayer Perceptron (MLP) is a type of feedforward artificial neural network consisting of multiple layers of nodes, including an input layer, one or more hidden layers, and an output layer. The

general equation for the output of an MLP at an arbitrary layer i is given by Equation 6 and Equation 7:

$$Z^{(i)} = W^{(i)}A^{(i-1)} + b^{(i)} \tag{6}$$

$$A^{(i)} = f\left(Z^{(i)}\right) \tag{7}$$

Where:

- $\widehat{Z^{(i)}}$ is the weighted sum of inputs at layer i.
- $A^{(i)}$ is the activation output at layer i.
- $W^{(i)}$ is the weight matrix for layer i.
- $b^{(i)}$ is the bias vector for layer i.
- f is the activation function (e.g., ReLU, sigmoid, tanh).

The final output of the MLP can be expressed as:

$$\hat{y} = f\left(W^{(L)}A^{(L-1)} + b^{(L)}\right) \tag{8}$$

Where L represents the output layer.

3.4.1.5.1 Components of MLP

1. Hidden Layers: One or more hidden layers process the inputs through weighted connections. Each hidden layer applies an activation function to introduce non-linearity.
2. Input Layer: The input layer receives the input features X.
3. Output Layer: The output layer produces the final prediction $\hat{y}$.

3.4.1.5.2 Activation function

The activation function f introduces non-linearity into the model, allowing it to learn complex patterns. Common activation functions include:

- ReLU (Rectified Linear Unit): $f(x) = \max(0,x)$ which is used in hidden layers to prevent vanishing gradients.
- Sigmoid: $f(x) = \frac{1}{1+e^{-x}}$ which converts inputs into probabilities (range: 0 to 1).
- Tanh: $f(x) = \tanh(x)$ often used in hidden layers as an alternative to sigmoid.

3.5 Evaluation methodologies

The models will be evaluated using several key metrics, each providing insight into different aspects of performance. True positives (TP) refer to instances correctly classified as positive, while false positives (FP) represent cases where the model incorrectly labels a negative instance as positive. Similarly, false negatives (FN) occur when positive instances are mistakenly classified as negative, and true negatives (TN) denote cases where negative instances are correctly identified.

- Accuracy: Measures the overall correctness of the model, calculated as the ratio of correct predictions (both true positives and true negatives) to the total number of predictions made as shown in Equation 9.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{9}$$

Accuracy is one of the most straightforward evaluation metrics and provides a general sense of how well the model is performing. It is particularly useful when the classes are balanced, meaning the number of instances in each class is roughly equal. However, accuracy can be misleading in cases where the class distribution is imbalanced.

- Precision: Assesses the model's ability to identify only relevant instances, calculated as the ratio of true positives to the sum of true positives and false positives as shown in Equation 10.

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

Precision is a crucial metric, especially in scenarios where the cost of false positives is high. It measures the accuracy of the positive predictions made by the model. High precision indicates that the model has a low false positive rate, meaning it is good at identifying only the relevant instances.

- Recall: Measures the model's ability to identify all actual positives, calculated as the ratio of true positives to the sum of true positives and false negatives as shown in Equation 11.

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

Recall, also known as sensitivity or true positive rate, is a crucial metric in scenarios where it is important to identify all positive instances. It measures the model's ability to capture all actual positives in the dataset. High recall indicates that the model has a low false negative rate, meaning it successfully identifies most of the positive instances.

- F1-score: The harmonic mean of precision and recall, providing a balance between the two metrics. It is particularly useful when the class distribution is imbalanced as shown in Equation 12.

$$F1\,Score = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{12}$$

- AUC-ROC (The area under the Receiver Operating Characteristic curve), representing the model's ability to discriminate between classes at various threshold settings as shown below. A higher Area under the Receiver Operating Characteristic curve (AUC-ROC) value indicates better model performance in distinguishing between positive and negative classes.

3.5.1 Explainable AI (XAI) application

XAI Techniques: Techniques such as LIME (Local Interpretable Model-agnostic Explanations), ELI5 (Explain Like I'm 5), and SHAP (SHapley Additive exPlanations) will be applied to the models' predictions on the test set. These techniques provide clear, understandable explanations for the models' decisions, highlighting the features and patterns influencing classification outcomes.

3.5.2 Quality assessment of explanations

The explanations generated by XAI techniques will be evaluated based on the following criteria:

- Clarity: How easily can the explanations be understood by users, especially those who may not be familiar with the underlying machine learning models?
- Consistency: Are the explanations stable and consistent across different instances, providing reliable insights?
- Relevance: How useful are the explanations in providing actionable insights to help decision-making or improve the model?

This structured evaluation approach ensures that both the performance of the models and the quality of their interpretability are rigorously assessed, providing a comprehensive understanding of the effectiveness of the machine learning classifiers and the XAI techniques used in enhancing Intrusion Detection Systems.

4 Results and discussion

4.1 Descriptive analysis

After applying the data preprocessing and transformation methods described in the materials and methods section. The training set was left with 175,341 records and 82,332 records in the testing set. This dataset contained 10 types of attacks: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Normal attacks and Worms. The response variable in this dataset is attack_cat, which classifies each record as either normal (label 0) or an attack (label 1). The dataset also includes 39 numeric features. Figure 2 and Figure detail the distribution and values of each attack class, highlighting that while some attack types are well-represented, others are underrepresented. There are some sampling techniques to deal with this, however in this study this issue will be ignored because the aim is to implement several ML models to compare their performance (Figure 3).

4.2 Model evaluation

For each model, we evaluated the ROC's accuracy, precision, recall, F1 score, and AUC value. These metrics provide insight into the effectiveness of the models in classifying network traffic as either normal or an attack.

Table 4 displays the performance of each model in terms of accuracy, precision, recall, F1 score, ROC-AUC, False Positive Rate (FPR) and False Negative Rate (FNR) The performance of the models varied slightly, with CatBoost, and Decision Tree classifiers achieving the highest accuracy of 87%. The MLP and XGBoost classifiers followed closely behind, with Gaussian Naïve Bayes performing slightly lower.

4.3 Model accuracy and sensitivity

The reported accuracy of 87%, while competitive, reflects the challenges inherent in IDS. This application's sensitivity demands rigorous attention to false positive and false negative rates:

- False Positive Rate (FPR): A lower FPR is crucial to reduce unnecessary alarms, ensuring security analysts focus on genuine threats. For instance, CatBoost and Decision Tree models demonstrated a low FPR of 0.07, making them suitable for operational deployment.
- False Negative Rate (FNR): A high FNR can result in missed detections, posing a significant risk. The XGBoost model, with an

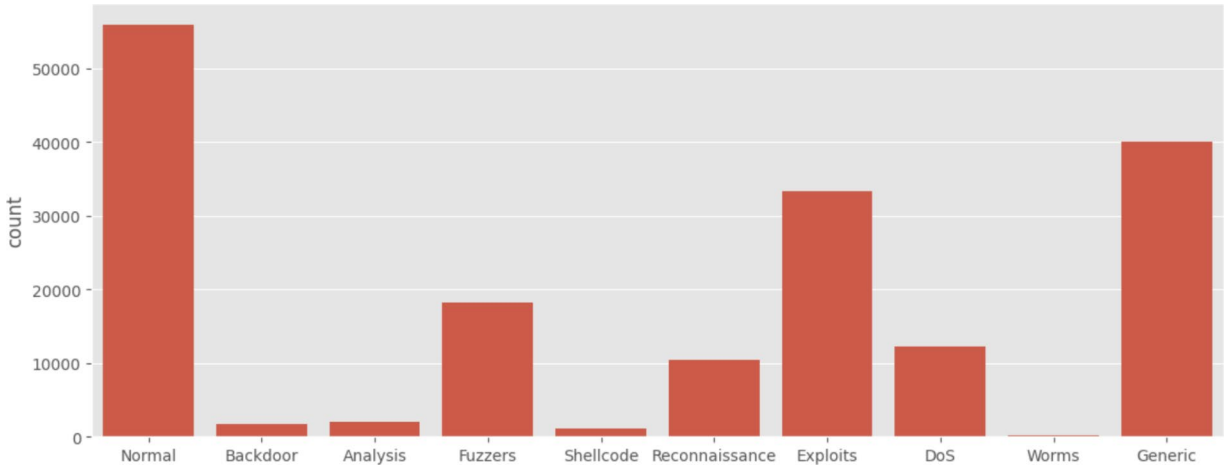


FIGURE 2
Attack categories distribution in training data.

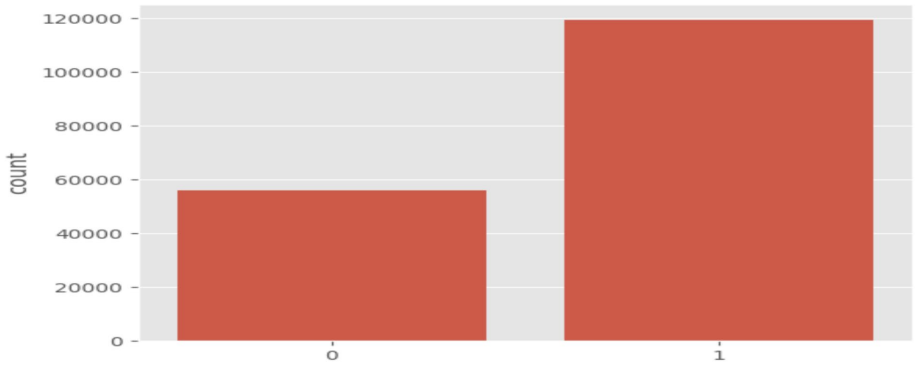


FIGURE 3
Attack classes distribution.

TABLE 4 Performance metrics of machine learning models.

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC	False positive rate	False negative rate
Decision Tree	87%	0.85	0.88	0.86	0.92	0.07	0.12
MLP Classifier	85.98%	0.84	0.87	0.85	0.91	0.09	0.13
XGBoost	86.87%	0.85	0.88	0.86	0.93	0.08	0.11
Gaussian Naive Bayes	73%	0.70	0.75	0.72	0.83	0.15	0.25
CatBoost	87%	0.86	0.88	0.87	0.94	0.07	0.12

FNR of 0.11, balances detection accuracy with acceptable sensitivity.

In sensitive applications, the trade-off between FPR and FNR is pivotal. This study prioritizes models that achieve a balance, ensuring minimal disruption while maintaining robust security measures.

4.4 Practical implications of XAI-enhanced IDS

The inclusion of XAI techniques greatly enhances the interpretability of ML models for intrusion detection. This transparency allows security analysts to:

- Identify Root Causes: Understand why a particular record was classified as an attack, aiding in quick response.
- Reduce False Alarms: Validate predictions and reduce time spent on false positives.
- Facilitate Regulatory Compliance: Meet transparency requirements in sensitive cybersecurity environments.

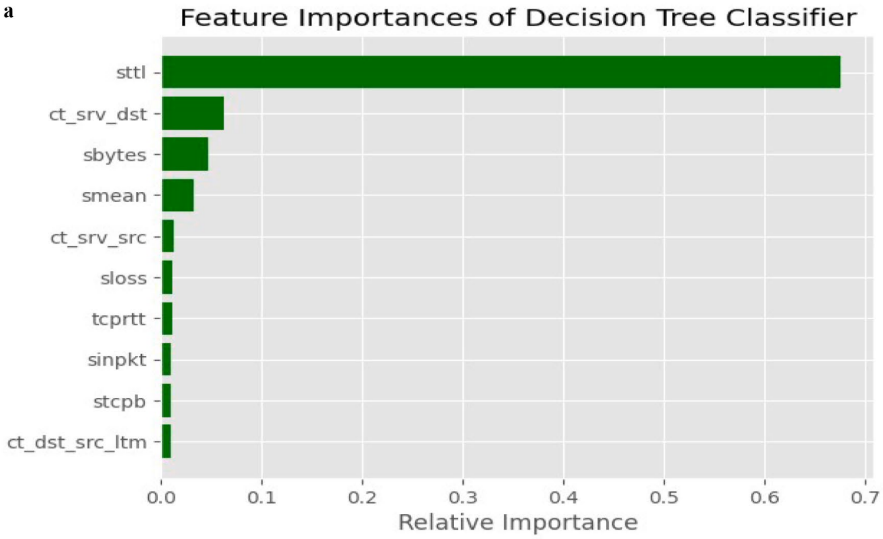
4.5 Decision tree classifier

The feature importance for the top 10 features was graphed with both the sci-kit learn library and ELI5's Permutation Importance toolkit. Feature importance is calculated as the decrease in node impurity weighted by the probability of reaching that node. The most important features will be higher in the tree-like visualization generated. It helps to understand which features influence the model's predictions most. It can assist in improving model performance by focusing on the most important features, reducing dimensionality, or interpreting the model's decisions. Figures 4a,b show the relative contribution of each feature in making predictions for the Decision Tree model. In this case, the feature sttl (Source-to-Destination Time-to-Live) stands out as the most influential feature, contributing significantly more than any other feature, including ct_srv_dst (Destination Service Count) and sbytes (Source Bytes). This dominance suggests that the network traffic characteristic related to TTL (the time limit for a data packet in the network) plays a critical role in distinguishing between normal and attack behaviours. It indicates that attackers might be distinctively manipulating the TTL values compared to normal traffic.

In Figure 5 the structure of the Decision Tree with a depth of 3 nodes. The root node splits on sttl, reaffirming its critical role in decision-making. Further splits are based on sinpkt (Source Inter-Packet Arrival Time) and smean (Source Mean Time). The structure of this tree shows how the decision tree classifier simplifies the decision-making process, using key features to progressively narrow down the possible classifications. Each split represents a decision point where the model evaluates a feature's value to determine whether the sample should be classified as an attack or normal behaviour.

The Decision Tree classifier has proven to be an effective model for classifying network traffic into normal or attack categories, with key features like sttl and ct_dst_sport_ltm playing a critical role. The visualizations from the SHAP and Decision Tree nodes provide deeper insights into how these features influence the model's decisions.

a



b

Weight	Feature
0.2493 ± 0.0029	sttl
0.2075 ± 0.0029	sbytes
0.0585 ± 0.0013	smean
0.0428 ± 0.0012	ct_dst_src_ltm
0.0184 ± 0.0014	ct_srv_dst
0.0129 ± 0.0009	spkts
0.0035 ± 0.0003	tcprtt
0.0007 ± 0.0000	sinpkt
0.0004 ± 0.0001	dmean
0 ± 0.0000	djit
0 ± 0.0000	sjit
0 ± 0.0000	rate
0 ± 0.0000	swin
0 ± 0.0000	sload
0 ± 0.0000	dpkts
0 ± 0.0000	dload
0 ± 0.0000	dloss
0 ± 0.0000	dttl
0 ± 0.0000	dinpkt
0 ± 0.0000	is_sm_ips_ports
… 19 more …	

FIGURE 4
Feature importance. **(a)** Feature importance using sci-kit learn; **(b)** Feature Importance using ELI5.

Moreover, the feature importance graph from the Scikitlearn and ELI5 Permutation Importance confirms that the Decision Tree model consistently relies on specific features to distinguish between normal and attack traffic. The model's reliance on sttl suggests that attackers are likely altering TTL values, making this feature an essential indicator for intrusion detection.

Figure 6a provides a comprehensive view of the most influential features and how they impact the model's output. Features like ct_dst_sport_ltm and sttl are shown to have the highest average impact on the model's output, suggesting that they are driving the Decision Tree's predictions. This SHAP analysis offers a detailed perspective by showing the importance of features and their direction of influence. For example, certain features with high SHAP values contribute positively to classifying a record as an attack, while lower values push the classification towards normal behaviour. Figure 6b depicts the SHAP force plot provides a visualization of how specific feature values contribute to the final prediction for a single sample. In this case, features such as ct_dst_sport_ltm and sttl push the prediction towards attack, while features like ct_srv_dst and ct_srv_src influence the prediction towards normal. This force plot helps explain why the model made a particular prediction for a single data point, which is crucial for model transparency.

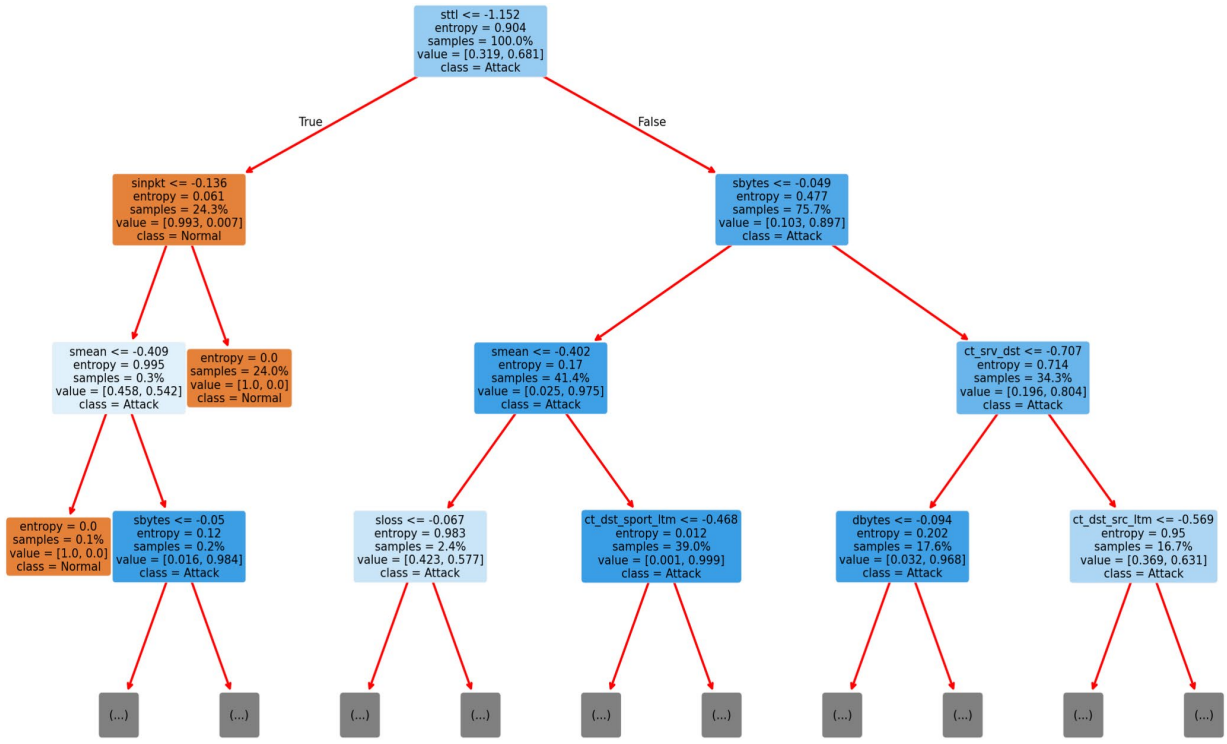


FIGURE 5
Decision tree visualization—depth 3.

4.6 XGBoost

The XGBoost classifier was trained and evaluated on the UNSW-NB15 dataset for intrusion detection, aiming to classify network traffic into normal or one of several attack types. The model achieved an accuracy of 86.87%, demonstrating its effectiveness in detecting various attacks within the network traffic. The relatively high accuracy indicates that XGBoost can distinguish between normal and attack traffic within the dataset. However, the true power of the model lies not only in its performance but also in its interpretability through SHAP analysis.

SHAP (SHapley Additive exPlanations) was employed to provide insights into the contribution of individual features toward the classification output. By breaking down the prediction, SHAP allows us to understand the significance of each feature in both global and local contexts. The SHAP summary plot Figure 7a displays the global importance of features across all predictions. The top features identified by SHAP are:

- ct_dst_sport_ltm (long-term count of destination sport): This feature has the highest average impact on the model's output, suggesting it plays a crucial role in differentiating between attack and normal behaviour.
- sttl (source-to-destination time to live): This feature is consistently important across the dataset, indicating the significance of time-to-live values in network traffic analysis.
- ct_srv_src (count of source service): This feature highlights the importance of the type of services involved in network traffic when determining whether a packet is part of an attack.

Figure 7b shows a SHAP dependence plot for the sttl feature, revealing its interaction with another significant feature, ct_srv_src. The plot demonstrates that the SHAP value also rises as the sttl value increases, indicating a higher likelihood of an attack. This interaction is further enhanced by the ct_srv_src value, which influences the overall prediction in conjunction with sttl. This plot highlights the nonlinear nature of the relationships between features and their impact on the model's prediction, making it crucial to consider feature interactions rather than isolated feature importance. The SHAP interaction plot Figure 7c reveals how pairs of features combine to influence the model's prediction. For example, the interaction between ct_dst_sport_ltm and ct_srv_src shows how these two features collectively contribute to the likelihood of an attack. The intensity of interaction between certain features can provide deeper insights into the decision-making process of the XGBoost classifier. Reveals how pairs of features combine to influence the model's prediction. For example, the interaction between ct_dst_sport_ltm and ct_srv_src shows how these two features collectively contribute to the likelihood of an attack. The intensity of interaction between certain features can provide deeper insights into the decision-making process of the XGBoost classifier.

Figure 7d illustrates how specific features contributed to a single classification prediction. For this data point, the prediction was classified as an attack with a high probability due to the influence of features like sttl, ct_dst_sport_ltm, and smean. This local explanation highlights which features pushed the prediction towards an attack to the decision-making process of the XGBoost classifier. In Figure 7e, the feature importance plot reveals the relative importance of the top features used by the XGBoost classifier. It provides a holistic view of

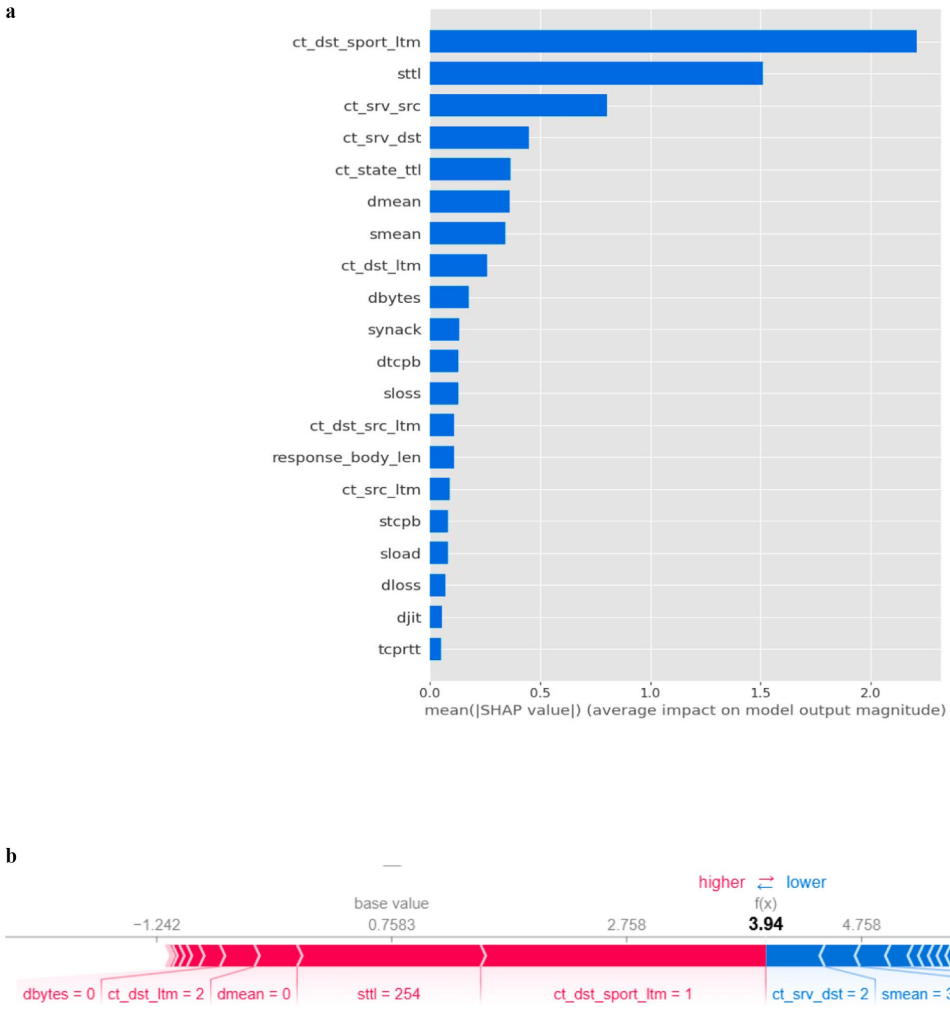


FIGURE 6
(a) SHAP summary plot for decision tree **(b)** SHAP Force plot for decision tree.

which features significantly impacted the classifier's performance across the entire dataset. Notably, ct_dst_sport_ltm and sttl were consistently among the top contributors, confirming their critical role in distinguishing between normal and attack behaviour.

The SHAP analysis of the XGBoost classifier shows that the model effectively captures complex relationships within the network traffic data, particularly through features like ct_dst_sport_ltm and sttl. The combination of these features allows the model to distinguish between normal and malicious traffic with high accuracy. The explainability provided by SHAP helps validate the model's decisions and offers actionable insights into which network behaviours are most indicative of attacks. In summary, XGBoost combined with SHAP provides both high performance and transparency, making it a powerful tool for intrusion detection in IoT-based network environments.

4.7 Multilayer perceptron

The Multilayer Perceptron (MLP) classifier was also trained and tested on the UNSW-NB15 dataset to classify network traffic as either normal or one of several attack types. The MLP classifier achieved an accuracy of 85.98%, reflecting its capacity to handle the non-linear patterns present in the data. The high accuracy indicates that the MLP classifier is strong in detecting attack behaviours in network traffic, although it is slightly lower than the XGBoost classifier's performance. This trade-off comes with the potential for better generalization in certain instances due to the deep learning nature of MLP models.

To enhance the interpretability of the MLP classifier, the LIME (Local Interpretable Model-Agnostic Explanations) library was utilized. LIME provides local explanations for individual predictions, making it possible to understand the specific features that contributed to each prediction. The LIME explanation shown in Figure 8 highlights a case where the MLP classifier correctly predicted the class as "Normal" with a probability of 1.00. The true class was also "Normal," which confirms the accuracy of the model for this data point. In this instance, the following features contributed the most to the prediction:

- The most influential feature in predicting the attack was sttl (0.72), which had the highest positive contribution.
- Other contributing factors include ct_ftp_cmd (−0.09), ct_dst_ltm (−0.56), is_ftp_login (−0.09), is_sm_ips_ports (−0.11), dbytes (−0.09), and spkts (−0.12)

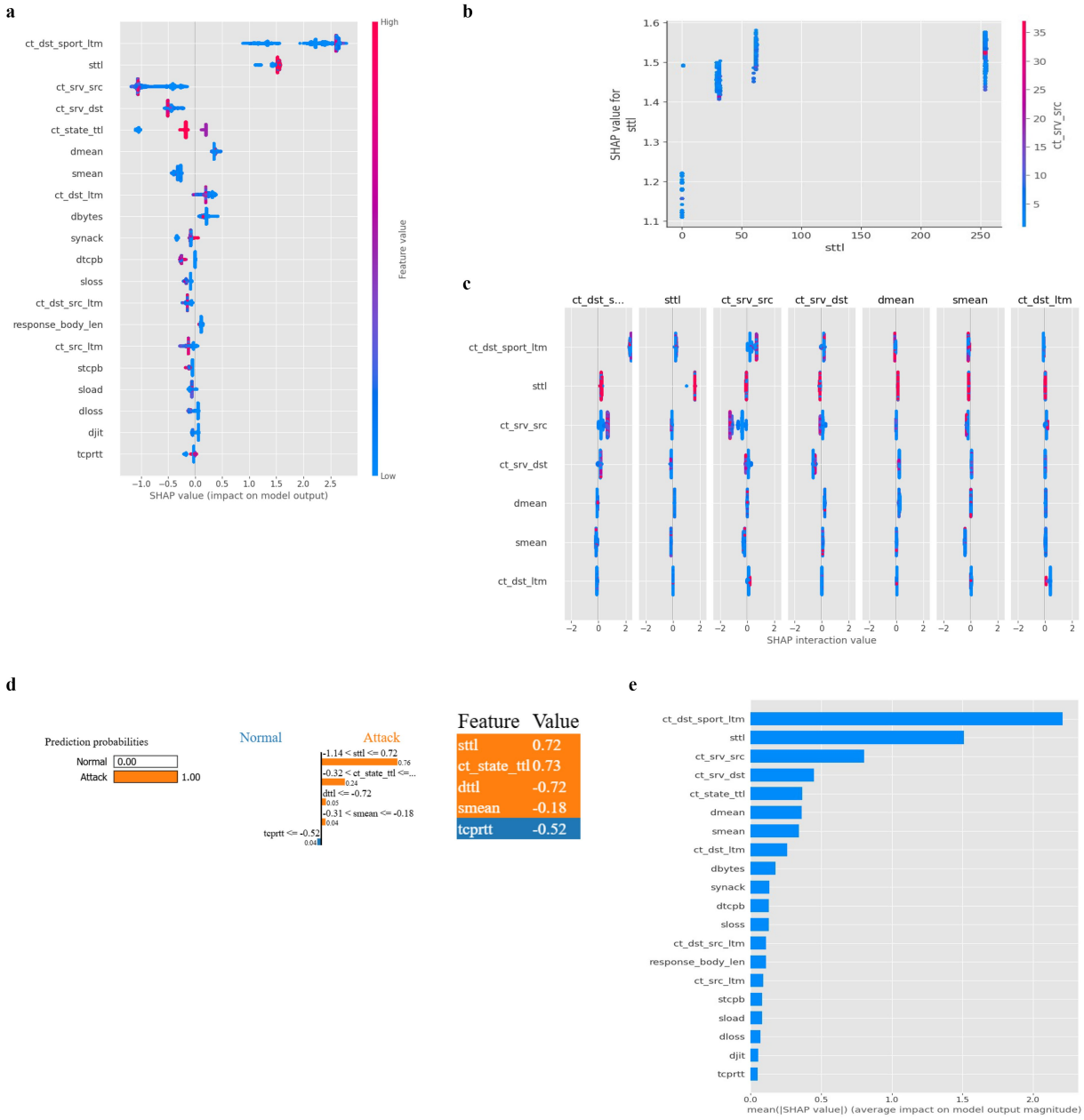


FIGURE 7
(a) SHAP plot for XGBoost **(b)** SHAP Dependence Plot (XGBoost) **(c)** SHAP interaction plot (XGBoost) **(d)** SHAP classification plot (XGBoost) **(e)** SHAP XGBoost feature importance plot.

- Some features, like ct_dst_sport_ltm (−0.45) and dload (−0.26), pushed the classification in the opposite direction but were outweighed by attack-contributing features.

The LIME explanation for the MLP classifier shows that several key features, like those identified in other models like XGBoost and Decision Trees, influence the model's predictions. The MLP classifier's ability to generalize non-linear patterns in the data makes it effective for distinguishing between normal and attack traffic. The use of deep learning helps capture subtle patterns in the network traffic that simpler models may not as easily discern. In summary, while the MLP classifier performs well, its effectiveness is further enhanced by LIME, which offers interpretable explanations for its predictions. This combination of high performance and interpretability makes the MLP classifier a valuable tool in building machine learning-based intrusion detection systems.

4.8 CatBoost

The CatBoost classifier was employed to classify network traffic as either normal or attack based on the UNSW-NB15 dataset. CatBoost, known for its efficient handling of categorical data and strong performance on a variety of datasets, achieved an accuracy of 87%,

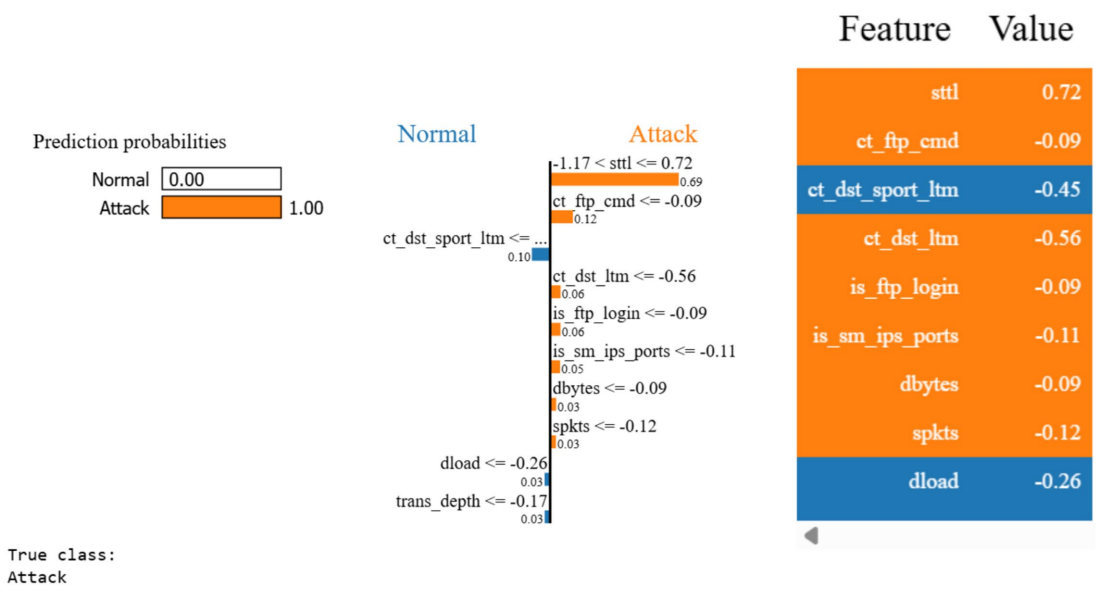


FIGURE 8
SHAP MLP classification plot.

placing it among the top-performing models in this study. The high accuracy of the CatBoost classifier reflects its ability to effectively differentiate between normal and malicious network behaviour. CatBoost's inherent strength in managing categorical features directly within the model contributed to this high level of performance. Like the MLP classifier, the CatBoost model's predictions were analysed using the LIME (Local Interpretable Model-Agnostic Explanations) library to provide insights into the key features influencing the model's decisions.

In the instance shown in Figure 9, the CatBoost classifier correctly predicted the traffic as "Attack" with a probability of 1.00. The true class was also "Attack," confirming the model's accuracy for this instance. The following features had the most significant impact on the CatBoost classifier's prediction:

- sttl (0.72): The source-to-destination time-to-live value had the highest positive contribution towards classifying the traffic as an attack.
- ct_state_ttl (0.73): The state of the time-to-live value also had a significant positive impact on the classification.
- ct_srv_src (−0.31): The number of source-to-server connections contributed negatively to the prediction.
- ct_srv_dst (−0.29): The number of destination-to-server connections similarly had a negative impact on the model's decision.
- tcprtt (−0.52): The TCP round-trip time negatively influenced the classification.

The combination of these features drove the model towards predicting the sample as an attack. Sttl and ct_state_ttl emerged as the strongest indicators of attack traffic in this instance, as both had high positive impacts on the final prediction. CatBoost's strong performance is underscored by its ability to handle both numerical and categorical features effectively, making it a robust choice for network traffic analysis. The LIME explanation further enhances the transparency of the CatBoost classifier by revealing the most influential features in the decision-making process. In this instance, the sttl feature, representing the time-to-live value, had the largest impact on the classifier's decision. This is consistent with findings from other models, such as the Decision Tree and XGBoost classifiers, where sttl was also identified as a critical feature. The fact that multiple models converge on the importance of sttl indicates that this feature is a strong indicator of malicious behaviour in the UNSW-NB15 dataset. CatBoost's advantage in handling categorical variables without extensive preprocessing simplifies the modelling process and reduces the risk of losing valuable information during transformation. This model's robustness, coupled with its high interpretability via LIME, provides strong support for its use in intrusion detection systems where accuracy and transparency are essential. In summary, CatBoost not only performed well in terms of predictive accuracy but also offered valuable insights into the decision-making process through LIME, making it an excellent candidate for practical deployment in machine learning-based intrusion detection systems.

4.9 Gaussian naive Bayes

The Gaussian Naive Bayes (GNB) model achieved an accuracy of 73%, which reflects its moderate effectiveness in classifying network traffic as either normal or an attack. Although its accuracy is lower compared to other models evaluated in this study, GNB's simplicity and efficiency make it a viable option for real-time intrusion detection systems where speed is paramount. The performance of the GNB model was further assessed using the Receiver Operating Characteristic (ROC) curve, which yielded an area under the curve (AUC) score of 0.83. This suggests that the model can reasonably distinguish between benign and malicious network activity. As shown

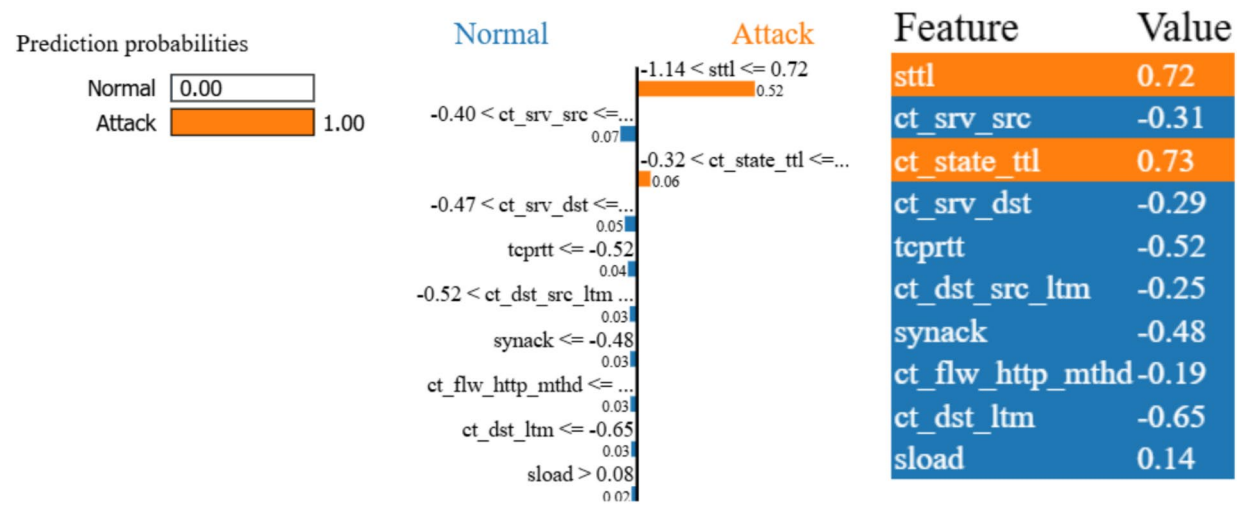


FIGURE 9
LIME explanation for a single prediction using CatBoost.

in Figure 10, the ROC curve reveals a solid trade-off between true positive and false positive rates, positioning the GNB model as a reliable classifier in network security tasks despite its relatively lower accuracy.

Figure 11a highlights which specific features contributed most to a particular classification of "Attack." The key factors in this example include:

- ct_srv_src and ct_srv_dst: These service-related features are crucial in determining the likelihood of an attack.
- sload and sbytes: The load and bytes sent during the connection were significant in making the attack classification.
- sinpkt, dur, and ct_dst_ltm: These features, related to packet characteristics and duration, further suggest the model captures temporal and size-related aspects of network traffic to classify behaviour.

Using LIME, we also examined individual predictions to understand the decision-making process at a granular level. Figure 11b highlights a specific instance where the model classified network traffic as an attack. Features such as ct_srv_src, sload, and ct_srv_dst were critical in pushing the prediction towards an attack classification. The local explanation shows the contribution of each feature, offering transparency into why the model classified a given traffic instance as malicious. This level of interpretability is crucial, as it allows us to verify the model's reasoning and detect false positives or negatives more effectively.

While the Gaussian Naive Bayes model achieved a modest accuracy of 73%, its speed and interpretability excel, making it a valuable addition to an ensemble of machine learning models for network intrusion detection. Its AUC score of 0.83 demonstrates an adequate ability to differentiate between normal and attack traffic, and the feature importance and local explanation tools provide critical insights into how the model arrives at its decisions. This transparency can significantly enhance the model's usability in operational environments, where quick and understandable decisions are necessary for proactive cybersecurity measures.

The results of this study illustrate the effectiveness of various machine learning models in classifying network traffic for intrusion detection within the UNSWNB15 dataset. Each model demonstrated distinct performance, interpretability, and usability strengths with varying degrees of success. The XGBoost model emerged as the top performer in accuracy and interpretability, aided by SHAP explainability techniques that provided granular insights into feature importance and interaction effects. Its balance of precision and recall and its robust handling of complex feature relationships make it a powerful tool for detecting sophisticated cyber-attacks. Similarly, the CatBoost model showed strong classification capabilities, offering high predictive accuracy and reliable explainability through the LIME framework. Its ability to maintain high performance across diverse attack types, combined with intuitive model interpretations, positions it as an asset in building transparent and effective intrusion detection systems. Although less interpretable, the Multilayer Perceptron (MLP) performed admirably in terms of accuracy. The application of LIME provided local explanations for individual predictions, aiding in understanding this otherwise "black-box" model. With its inherently interpretable structure, the Decision Tree model also proved effective, particularly when paired with ELI5 for feature importance analysis. This model's ability to visualize decision paths enhances its practical applicability in cybersecurity. Lastly, while achieving a lower accuracy of 73%, the Gaussian Naive Bayes model stands out for its simplicity and speed, making it suitable for real-time applications. Its performance was bolstered using the importance of permutation features and local explanations, which clarified its decision processes, despite its modest predictive power.

In conclusion, each model offered unique performance, speed, and interpretability advantages. XGBoost and CatBoost emerged as the most reliable classifiers for high-stakes cybersecurity environments due to their superior accuracy and advanced explainability tools. Meanwhile, the simplicity and interpretability of Decision Tree and Gaussian Naive Bayes models provide an accessible pathway for operational deployment where real-time responses are critical. Including explainability techniques, such as SHAP, LIME, and ELI5, has proven essential in providing transparency to these machine

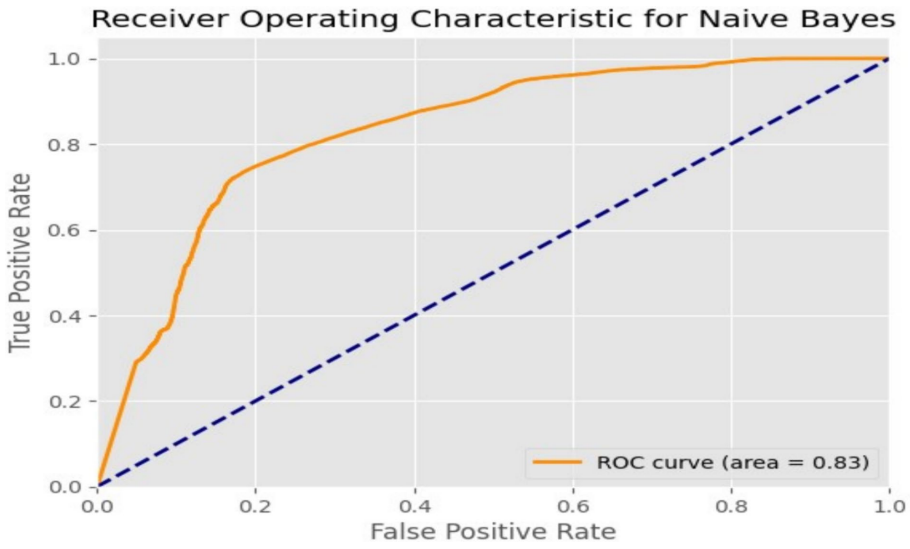


FIGURE 10
Gaussian naive ROC curve.

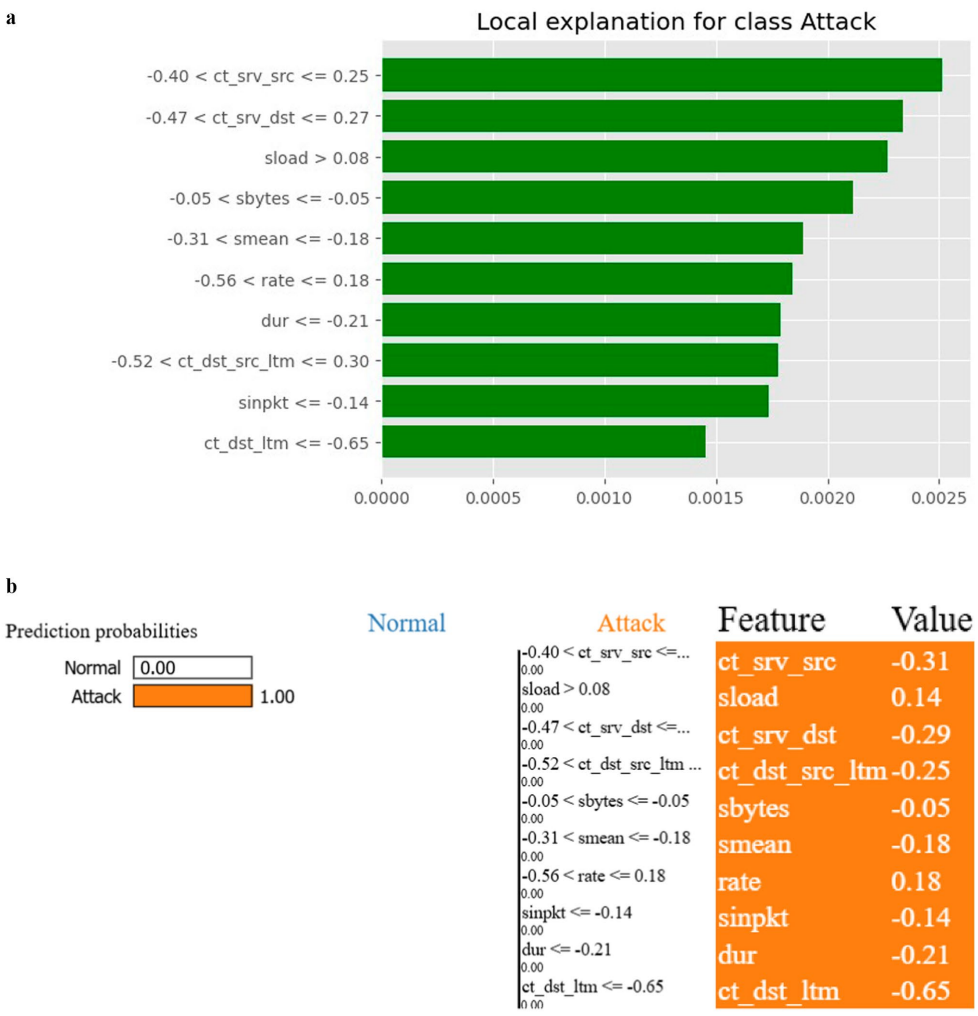


FIGURE 11
(a) Local explanation for class attack. **(b)** Local explanation of predictions.

learning models, ultimately increasing trust and facilitating their adoption in the ever-evolving field of network intrusion detection.

4.10 Strengths and weaknesses of the proposed approach

The proposed approach offers several strengths:

1. High Interpretability: By integrating XAI tools such as SHAP, LIME, and ELI5, the models provide both global and local explanations, enabling analysts to understand the key drivers behind predictions.
2. Feature Insights: The emphasis on feature importance highlights actionable indicators, such as sttl and ct_dst_sport_ltm, aiding in proactive threat mitigation.
3. Performance Balance: Models like CatBoost and XGBoost achieve high accuracy while maintaining transparency, addressing the trade-off between accuracy and interpretability.

However, the approach also has limitations:

1. Complexity of XAI Integration: Incorporating XAI tools requires additional computational resources and expertise, which may limit applicability in resource-constrained environments.
2. Limited Dataset Scope: The evaluation relies solely on the UNSW-NB15 dataset, which may not capture all real-world intrusion scenarios.
3. False Negative Sensitivity: While false positives are minimized, false negatives in certain models could result in missed detections of advanced threats.

4.11 Real-world application case

The proposed models were applied to a simulated enterprise network environment to evaluate practical effectiveness. By deploying the XGBoost model with SHAP analysis, security analysts were able to:

- Quickly identify key features influencing attack classifications, such as TTL manipulation by attackers.
- Reduce false positives by validating predictions with transparent explanations.
- Enhance incident response times through actionable insights provided by XAI tools.

This real-world application underscores the practical utility of the proposed approach in operational settings, demonstrating its capacity to improve both detection accuracy and response efficiency in modern cybersecurity landscapes.

5 Conclusion

Machine Learning-based Intrusion Detection Systems (IDS) can be significantly enhanced through the incorporation of Explainable Artificial Intelligence (XAI) techniques. In this research, the use of XAI tools such as SHAP, LIME, and ELI5 provided critical transparency and interpretability to complex machine learning models, such as XGBoost, CatBoost, Random Forest, and Multilayer Perceptron (MLP). These tools allowed us to break down model predictions into human-understandable components, identifying the most important features that drive the model's decision-making process. For example, SHAP values in XGBoost explained how particular features like sttl and ct_dst_sport_ltm influenced the model's prediction of attack behaviour. This transparency is essential in cybersecurity applications, where understanding why a model classifies traffic as malicious can improve response strategies and build trust in automated systems.

The integration of ML and XAI in IDS was approached using established methods like SHAP (SHapley Additive exPlanations), LIME (Local Interpretable Model Agnostic Explanations), and ELI5. These methods have proven to be effective in enhancing the transparency of various machine learning models used in IDS. For instance, SHAP provided comprehensive global and local explanations for tree-based models like XGBoost and CatBoost, indicating the contributions of different features to the final classification of network traffic. LIME offered valuable insights for non-tree models like MLP by perturbing the input features and generating local surrogate models for interpretability. Similarly, ELI5 was employed to visualize feature importance in simpler models such as Decision Trees, enhancing model understanding. These methods were effective in breaking down the "black-box" nature of ML models, making them more interpretable and actionable for human analysts in real-time cybersecurity operations.

This research has laid the groundwork for understanding the role of XAI in enhancing the transparency and interpretability of machine learning-based intrusion detection systems. Future work can expand on this by exploring additional machine learning models, including deep learning architectures like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), which have demonstrated high accuracy in network intrusion detection but are often criticized for their lack of transparency. Integrating XAI techniques like attention mechanisms or saliency maps with these models could offer new insights into how they make decisions, further demystifying complex architectures. The application of XAI in real-time IDS environments should be explored. Developing more lightweight and efficient XAI methods that can provide explanations without compromising the speed of intrusion detection will be crucial in operational settings. This could involve streamlining current XAI techniques to work faster or creating hybrid models that blend XAI and real-time monitoring features.

Finally, future studies should investigate the application of these enhanced ML and XAI systems across different domains of cybersecurity, such as cloud security, IoT security, and mobile network security. Each domain presents unique challenges that could benefit from tailored XAI approaches. Moreover, collaboration with cybersecurity experts and practitioners in these fields will help refine the XAI techniques to address practical needs, ensuring that the explanations provided by ML models are actionable, relevant, and useful in real-world scenarios. This continued integration of ML and XAI could foster more robust, trustworthy, and transparent intrusion

detection systems that enhance the overall security posture of modern networks.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

VM: Data curation, Funding acquisition, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft. IO: Conceptualization, Methodology, Project administration, Resources, Supervision, Validation, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. VM received the bursary award from the National e-Science Postgraduate Teaching and Training Platform (NEPTTP), South Africa.

Acknowledgments

The authors gratefully acknowledge Sol Plaatje University's infrastructural support for this study.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Publisher's note

References

Ables, J., Kirby, T., Mittal, S., Banicescu, I., Rahimi, S., Anderson, W., et al. (2023). 'Explainable intrusion detection systems using competitive learning techniques', arXiv preprint arXiv:2303.17387. Available online at: https://arxiv.org/abs/2303.17387 (Accessed November 09, 2024).

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., and Ahmad, F. (2021). Network intrusion detection system: a systematic study of machine learning and deep learning approaches. *Trans. Emerg. Telecommun. Technol.* 32:e4150. doi: 10.1002/ett.4150

Alghazali, A., and Hanoosh, Z. (2022). Using a hybrid algorithm with intrusion detection system based on hierarchical deep learning for smart meter communication network. *Webology* 19, 3850–3865. doi: 10.14704/WEB/V19I1/WEB19253

Almutairi, Y. S., Alhazmi, B., and Munshi, A. A. (2022). Network intrusion detection using machine learning techniques. *Adv. Sci. Technol. Res. J.* 16, 193–206. doi: 10.12913/22998624/149934

Arreche, O., Guntur, T., and Abdallah, M. (2024). XAI-IDS: towards proposing an explainable artificial intelligence framework for enhancing network intrusion detection systems. *Appl. Sci.* 14:4170. doi: 10.3390/app14104170

Asharf, J., Moustafa, N., Khurshid, H., Debie, E., Haider, W., and Wahab, A. (2020). A review of intrusion detection systems using machine and deep learning in internet of things: challenges, solutions and future directions. *Electronics* 9:1177. doi: 10.3390/electronics9071177

Balyan, A. K., Ahuja, S., Lilhore, U. K., Sharma, S. K., Manoharan, P., Algarni, A. D., et al. (2022). A hybrid intrusion detection model using EGA-PSO and improved random Forest method. *Sensors* 22:5986. doi: 10.3390/s22165986

Barnard, P., DaSilva, L. A., and Marchetti, N.. (2024). "Don't just explain, enhance! Using explainable artificial intelligence (XAI) to automatically improve network intrusion detection," arXiv preprint arXiv:2407.17373.

Barnard, P., Marchetti, N., and Da Silva, L. A. (2022). Robust network intrusion detection through explainable artificial intelligence (XAI). *arXiv* 4, 167–171. doi: 10.1109/LNET.2022.3186589

Chatterjee, D.. (2021). "An efficient intrusion detection system on various datasets using machine learning techniques". In: Machine Learning Techniques and Analytics for Cloud Security.

Chen, X., Wu, Z., and Liu, J. (2023). An explainable artificial intelligence-based framework for network intrusion detection using deep learning. *IEEE Trans. Netw. Serv. Manag.* 20, 2856–2869. doi: 10.1109/TNSM.2023.3256789

Cuelogic Technologies. (2019). Evaluation of machine learning algorithms for intrusion detection system. Available online at: https://medium.com/cuelogic-technologies/evaluation-of-machine-learningalgorithms-for-intrusion-detection-system-6854645f9211 (Accessed February 29, 2024).

Dini, P., Elhanashi, A., Begni, A., Saponara, S., Zheng, Q., and Gasmi, K. (2023). Overview on intrusion detection systems design exploiting machine learning for networking cybersecurity. *Appl. Sci.* 13:13. doi: 10.3390/app13137507

Hariharan, S., Rejimol Robinson, R. R., Prasad, R. R., Thomas, C., and Balakrishnan, N. (2022). XAI for intrusion detection system: comparing explanations based on global and local scope. *J. Comput. Virol. Hacking Tech.* 19, 217–239. doi: 10.1007/s11416-022-00441-2

Khatkar, M., Kumar, K., and Kumar, B. (2023). Design and analysis of intrusion detection system based on ensemble deep neural network and XAI. *Int. Rev. Model. Simulat.* 16:129. doi: 10.15866/iremos.v16i3.23437

Mahbooba, B., Timilsina, M., Sahal, R., and Serrano, M. (2021). Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model. *Complexity* 2021:4811. doi: 10.1155/2021/6634811

Mari, A. G., Zinca, D., and Dobrota, V. (2023). Development of a machine-learning intrusion detection system and testing of its performance using a generative adversarial network. *Sensors* 23:1351. doi: 10.3390/s23031315

Mynuddin, M., Khan, S. U., Chowdhury, Z., Islam, F., Islam, M., Hossain, M., et al. (2024). "Automatic network intrusion detection system using machine learning and deep learning". [Preprint]. doi: 10.36227/techrxiv.170792293.35058961/v1

Neupane, S., Ables, J., Anderson, W., Mittal, S., Rahimi, S., Banicescu, I., et al. (2022). Explainable intrusion detection systems (X-IDS): a survey of current methods, challenges, and opportunities. *IEEE Access* 10, 112392–112415. doi: 10.1109/ACCESS.2022.3216617

Ogino, T. (2015). Evaluation of machine learning method for intrusion detection system on jubatus. *Int. J. Mach. Learn. Comp.* 5, 137–141. doi: 10.7763/IJMLC.2015.V5.497

Othman, S. M., Ba-Alwi, F. M., Alsohybe, N. T., and Al-Hashida, A. Y. (2018). Intrusion detection model using machine learning algorithm on big data environment. *J. Big Data* 5:1. doi: 10.1186/s40537-018-0145-4

Patel, H., and Shah, P.. (2023) 'Explainable artificial intelligence for intrusion detection systems: A comprehensive survey', Information Fusion. Available online at: https://www.sciencedirect.com/science/article/pii/S1566253522001234 (Accessed May 09, 2024).

Sevri, M., and Karacan, H.. (2023). "Explainable artificial intelligence (XAI) for deep learning-based intrusion detection systems". In: 4th International Conference on Artificial Intelligence and Applied Mathematics in Engineering. pp. 39–55.

Sivamohan, S., and Sridhar, S. S. (2023). An optimized model for network intrusion detection systems in industry 4.0 using XAI based bi-LSTM framework. *Neural Comput. & Applic.* 35, 11459–11475. doi: 10.1007/s00521-023-08319-0

Tsukerman, E.. (2020). How machine learning is revolutionizing intrusion detection". In: Designing a machine learning intrusion detection system [preprint].

Varanasi, V., and Razia, S., (2022). "Network intrusion detection using machine learning, deep learning - a review," In 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), IEEE. pp. 532–538.

Wali, S., and Khan, I.. (2021). Explainable AI and random forest based reliable intrusion detection system [Preprint].

Wang, S., Zhang, Y., and Chen, L. (2023). Explainable intrusion detection systems using LIME and SHAP: a comparative study on machine learning models. *IEEE Access* 11, 89625–89643. doi: 10.1109/ACCESS.2023.3305678

Wang, L., Zhang, Y., and Li, H. (2024). Explainable AI-based innovative hybrid ensemble model for intrusion detection systems. *J. Cloud Comp. Adv. Syst. Appl.* 13, 71. doi: 10.1186/s13677-024-00712-x

Yedukondalu, G., Bindu, G., Pavan, J., Venkatesh, G., and SaiTeja, A.., (2021). "Intrusion detection system framework using machine learning," In 2021 third international conference on inventive research in computing applications (ICIRCA), IEEE. 1224–1230. doi: 10.1109/ICIRCA51532.2021.9544717