Explainable Artificial Intelligence Models for Intrusion Detection Systems

Yılmaz Atay1, Zeynep Bölükbaşı2, Zeynep Yiğit1, Şeref Sağıroğlu1

1Gazi University Engineering Faculty Computer Engineering Department

2TED University Engineering Faculty Computer Engineering Department

Ankara, Türkiye

**Abstract**

In the digital era, ensuring information security and maintaining communication integrity are critical for sustainable progress and building trust in digital systems. Intrusion Detection Systems (IDS) are essential for identifying unauthorized access attempts and strengthening cybersecurity defenses. This study investigates the integration of Machine Learning (ML) with Explainable Artificial Intelligence (XAI) within IDS frameworks, aiming to enhance transparency, interpretability, and overall effectiveness. An extensive literature review was conducted to examine current IDS methodologies, relevant datasets, and prevailing ML techniques. In parallel, experimental research was carried out to evaluate the impact of leading XAI methods—such as SHAP and LIME—on IDS performance. The results indicate that the combination of ML and XAI significantly improves both the reliability of IDS and the clarity of its decision-making processes. Overall, this study contributes to the development of more secure and transparent AI-driven cybersecurity solutions, paving the way for advancements in both research and practical applications within the field.

**Keywords:** Explainable Artificial Intelligence, Machine Learning, Intrusion Detection System, Experiment, Dataset, LIME, SHAP, Cybersecurity.

# Introduction

The interaction between information, communication, and digitalization forms the backbone of modern organizations. Embracing these elements is essential for fostering innovation, increasing efficiency, and maintaining a competitive advantage in an ever-evolving digital landscape. Focusing on the fundamental role of information and communication in digital transformation reveals that securing key elements within information systems is a critical priority. As advancements in digitalization accelerate, concerns surrounding the security, confidentiality, integrity, and sustainability of information and communication networks become increasingly significant.

Information security refers to the effective protection of sensitive data against unauthorized access, various breaches, and other threats. Similarly, the security of communication networks aims to maintain uninterrupted information exchange. Secure communication channels prevent interference, tampering, and unauthorized access to transmitted data. This is particularly crucial for digital businesses and e-commerce platforms, which heavily rely on the integrity of communication networks. The interactive environment facilitated by digitalization can also increase systems’ vulnerability to cyberattacks. Therefore, implementing effective cybersecurity measures such as encryption, authentication, and intrusion detection systems (IDS) is critical for protecting against potential threats. Additionally, security protocols must be continuously monitored and updated to address emerging risks and ensure the resilience of digital infrastructures. By prioritizing information security and securing communication networks, organizations can establish a safe and trustworthy digital environment. After addressing the critical aspects of information security and communication, it is essential to conduct an in-depth examination of the specific technologies that support these security measures. Among the most important systems supporting security measures are Intrusion Detection Systems (IDS) and Intrusion Prevention

Systems (IPS). These systems play a crucial role in identifying, mitigating, and preventing security threats while ensuring the integrity and resilience of digital infrastructures.

Intrusion and penetration generally refer to unauthorized access to a system or network. Such attempts can be carried out by exploiting security vulnerabilities or bypassing relevant protocols. Types of intrusions include brute force attacks, SQL injections, phishing attacks, and malware distribution, among others. Unauthorized intrusions can lead to severe consequences such as data theft, service disruptions, and financial losses. IDSs play a critical role in monitoring network traffic for suspicious activities and potential threats. IDS enables the detection of anomalies that indicate unauthorized access or malicious attempts. To achieve this, IDS analyzes network packets, system logs, and other sources to identify patterns that match known attack signatures or behaviors. According to Fuchsberger, IDS is highly significant in alerting administrators to potential security breaches, allowing timely intervention and providing an essential layer of defense [1].

Dorothy E. Denning’s paper, An Intrusion-Detection Model (1987), was one of the first studies to propose a systematic approach for detecting unauthorized access and anomalies in computer systems [2]. Denning’s approach was based on profiling by utilizing metrics such as user activity patterns and resource usage to establish a baseline of normal activities. According to this model, when activities deviate significantly from the baseline, the system flags them as potential intrusions. This study emphasized the importance of real-time monitoring and dynamic adaptation to new threats. Denning’s study laid the theoretical foundation for incorporating more sophisticated techniques, such as machine learning, to reduce false positives and enhance detection capabilities. In addition to IDS, another study focusing on proactive defense introduced Snort, an open-source network intrusion detection and prevention system, which provides a flexible and powerful tool for real-time traffic analysis and intrusion detection [3]. The Snort tool utilized a rule-based language to define traffic patterns and signatures indicative of malicious activity, enabling real-time traffic analysis. The integration of machine learning (ML) into IDS has significantly enhanced the capabilities of these systems by automating and improving traditional IDS approaches. ML techniques offer advanced methodologies for analyzing large volumes of network data, identifying patterns, and predicting potential threats with higher accuracy and efficiency. In this study, some of the main contributions in the literature regarding the relationship between machine learning and IDS are summarized before moving on to the implementation phase. By leveraging historical attack data, ML algorithms automate the learning process and enable the detection of previously unseen intrusion types.

In machine learning, techniques such as supervised, unsupervised, and reinforcement learning are widely applied. These methods assist in developing models that classify network activities as either normal or malicious based on learned patterns from data. To highlight the significance of the topic, it is essential to first discuss the literature emphasizing the importance of machine learning and deep learning in the IDS domain. Javaid et al. proposed a deep learning approach based on a sparse autoencoder to detect intrusions in networks [4]. Recent deep learning models have demonstrated the capability to effectively learn and

recognize complex patterns in network traffic, achieving high accuracy in distinguishing between normal and malicious activities. The study involved training an autoencoder on a large network traffic dataset, where input data were compressed into a lower-dimensional representation and then reconstructed. Since malicious traffic tends to cause higher reconstruction errors compared to normal traffic, the reconstruction error was used to identify anomalies. This approach significantly enhances the capabilities of intrusion detection systems by providing an effective solution for detecting previously unseen attack patterns [4].

Another study [5] presented a detailed analysis of deep learning techniques used in IDS. This study categorized different approaches such as Convolutional Neural Networks (CNNs), Autoencoders, and Generative Adversarial Networks (GANs), discussing their applications and outlining potential future research directions. The authors reviewed existing literature to highlight the strengths and limitations of these methods and emphasized the need for more robust and scalable solutions. Their evaluations and analyses demonstrated that contemporary approaches offer significant improvements in detection accuracy and adaptability. Moreover, the study indicated that challenges such as high computational costs and the need for large labeled datasets remain critical research topics for future studies. Karataş et al. conducted a comprehensive literature review on deep learning techniques used in IDSs. Their study compared the performance of deep learning models with traditional approaches, showcasing the advantages of deep learning in terms of accuracy and scalability. The research analyzed the performance of various architectures, including CNNs and Recurrent Neural Networks (RNNs), and discussed their strengths and weaknesses in the context of intrusion detection. By examining results from numerous experiments, the authors concluded that deep learning models significantly outperform traditional methods in detecting complex cyber threats [6].

The literature reviews presented above provide a general perspective on the topic. In the subsequent sections of this study, a comprehensive literature analysis on IDS will be conducted. The integration of machine learning with IDS continues to drive innovation in the field by enhancing the ability of systems to detect and respond to cyber threats in real time. Ongoing research and developments in this area continue to push boundaries, positioning IDS as a critical component of modern cybersecurity strategies. It is also essential to highlight the role of Explainable Artificial Intelligence (XAI) in Intrusion Detection Systems. XAI addresses the interpretability requirement, which is crucial for complex AI systems, by making AI-driven decision-making processes as transparent and understandable as possible. This study explores the significance of XAI, its necessity, benefits, and relationship with IDS. XAI encompasses a range of methods that enable users to understand, interpret, and trust the outputs generated by machine learning algorithms. Explainability technology focuses on developing AI systems whose actions can be easily interpreted by humans, thereby enhancing transparency and accountability in AI-driven decisions. Interpretability in machine learning is vital for verifying AI decisions and identifying potential biases. Transparency is particularly important because it provides insights into how and why a model makes specific decisions, which is necessary for building trust and ensuring proper validation. Accountability, on the other hand, is a crucial factor that ensures AI systems are held responsible for their decisions and assists in meeting regulatory compliance requirements. By making AI operations more

comprehensible, XAI significantly increases user confidence and allows model predictions to be explained in human terms. This level of transparency enables users to make more informed choices based on AI-generated outputs. The integration of XAI into IDS presents opportunities for advancements in cybersecurity by ensuring transparency, interpretability, and trust in AI-driven systems. As cyber threats continue to grow in complexity, the need for explainable and accountable AI models will become increasingly significant. Research in this field suggests that XAI can enhance the effectiveness and reliability of IDS, ultimately contributing to the development of more secure and trustworthy digital environments.

In this study, a comprehensive literature review on ML and XAI research for IDS has been conducted, followed by practical implementations exploring the use of XAI technologies in IDS-related problems.

First, the *Introduction* section provides an overview of the study's purpose, scope, and significance. The *Literature Review* section examines IDS methods and datasets in detail. Next, the *IDS Classification* section introduces IDS systems and discusses their classification based on different characteristics. The *Machine Learning in IDS* section presents a detailed analysis of machine learning techniques applied in IDS. The *XAI Methods* section introduces explainable artificial intelligence techniques and explores their applications in IDS systems. In the *Experimental Studies* section, the results for experimental studies on the application of XAI technologies in IDS problems are presented. Finally, the *Conclusion and Recommendations* section summarizes the study’s findings and provides suggestions for future research.

# Literature Review

In this section, the effectiveness of IDS problems and XAI technologies is discussed. The approaches, methods, techniques, and datasets used in IDS are examined under two main headings: intrusion detection systems and IDS datasets.

# Intrusion Detection Systems

Zhong et al. examined the applications of graph neural networks (GNNs) in network security and intrusion detection, highlighting their effectiveness in enhancing IDS performance [7]. Lampe and Meng discussed the advantages of deep learning methods over traditional approaches for in-vehicle network (IVN) intrusion detection and proposed hybrid approaches [7]. Neupane et al. emphasized the importance of explainable intrusion detection systems (X-IDS) and analyzed White-Box and Black-Box approaches [8]. Thakkar and Lohiya investigated the security challenges of IDSs in IoT networks and the use of machine learning techniques, stating that these methods are insufficient [9].

Additionally, studies evaluating the performance of ML and DM based IDSs using different datasets are available [7]. Ring et al. detailed network-based intrusion detection datasets to assist researchers in selecting the most appropriate datasets [10].

Meena and Choudhary compared the performance of J48graft decision tree and Naïve Bayes algorithms on network traffic datasets, concluding that J48graft performed better [11]. Kasongo et al. improved accuracy rates by utilizing XGBoost for feature selection to enhance IDS performance [12]. Türk et al. demonstrated the effectiveness of machine learning and deep learning algorithms on NSL-KDD and UNSW-NB15 datasets [13]. Vihbute et al. found that the KNN model achieved high accuracy in detecting network anomalies [14]. Zhang et al. concluded that ensemble learning methods provided the highest accuracy in network intrusion detection [15]. Fosić et al. highlighted the effectiveness of the RF algorithm in anomaly detection within NetFlow data streams [16]. Imanbayev et al. reported that the Gradient Boosting algorithm showed the best performance in 5G networks [17]. Azam et al. demonstrated the success of decision tree-based methods in intrusion detection [18]. Belavagi and Muniyal found that Random Forest achieved the highest accuracy on the NSL-KDD dataset [19]. Table 1 summarizes various studies in the field of intrusion detection systems and provides the datasets, methods, and contributions to the literature.

Table 1. Summary of Literature on Intrusion Detection Systems

|  |  |  |  |
| --- | --- | --- | --- |
| Refs | Dataset | Methods | Contribution to the Literature |
| [7] | Network traffic and system logs | GNN-based modeling | This study demonstrated that GNNs can enhance IDS performance by capturing  complex relationships in network data. |
| [7] | IVN datasets | DNN, CNN, LSTM, DTL, GAN | The study highlights that deep learning-based IVN attack detection strategies are more complex but more effective than traditional methods. |
| [8] | Various datasets for X-IDS | White-box and Black-box approaches (LIME, SHAP) | Explainable IDS (X-IDS) systems aim to address security concerns by increasing the explainability of black-box models. |
| [9] | KDD CUP 99,  NSL-KDD, MNIST, UNSW-NB15, DREBIN | ML and DL techniques (Decision Tree, KNN, SVM, PCA) | This study demonstrated that machine learning and deep learning techniques can be effectively used for securing IoT networks. |
| [7] | CIC-IDS-2017, CSE-CIC-IDS-2018 | ML and DM techniques | The study provides a comprehensive analysis of IDS datasets capable of  detecting modern attacks. |
| [10] | 34 different datasets | Analysis of various network datasets | This study assists researchers in selecting appropriate datasets by thoroughly analyzing dataset characteristics and attack scenarios. |
| [11] | KDD99, NSL-KDD | J48graft, Naive Bayes | This study contributed to the literature by analyzing the effectiveness of the J48graft algorithm in detecting network traffic attacks. |
| [12] | UNSW-NB15 | Feature selection with XGBoost, ML techniques (SVM, KNN, DT) | The study demonstrated that ML-based IDS performance can be improved through XGBoost-based feature selection. |

|  |  |  |  |
| --- | --- | --- | --- |
| [13] | NSL-KDD, UNSW-NB15 | RF, LSTM, LR, KNN | This study extensively analyzed the effectiveness of machine learning algorithms in binary and multi-class classification problems. |
| [14] | NSL-KDD | RF, SVM, KNN | The study demonstrated that the KNN algorithm is an effective solution for detecting network anomalies with high accuracy. |
| [15] | KDD CUP99, NSL-KDD | Decision Tree, Naïve Bayes, SVM, RF, XGBoost | The study systematically examined the advantages and disadvantages of different machine learning algorithms. |
| [16] | UNSW-NB15 | RF, SGD, SVM, KNN, GNB | The study demonstrated that the RF algorithm performs exceptionally well in NetFlow data and is effective in anomaly detection. |
| [17] | CICIDS2017,  CSE-CIC-IDS2018 | LR, RF, Gradient Boosting, Autoencoder | Gradient Boosting was found to deliver high accuracy and successful results for securing 5G networks. |
| [18] | NSL-KDD, KDD CUP99 | Decision Tree | The study showed that decision trees provide high accuracy and low false positive rates in IDS performance. |
| [19] | NSL-KDD | LR, GNB, SVM, RF | The study demonstrated that the RF algorithm achieved the best performance in detecting network attacks with high accuracy. |

# IDS Datasets

The IDS datasets used in this study include UNSW-NB15, NSL-KDD, CICIDS2017, CIRA-CIC-DoHBrw-2020, and CICIoT2023. Additionally, several datasets exist in the literature but were not utilized in this study, including KDD Cup 99, DEFCON, Kyoto 2006+, UMASS, ISCX2012, CSE-CIC-IDS2018, and 5G-NIDD. General descriptions of these datasets are provided below under specific headings.

# XAI Methods Used in IDS

Intrusion Detection System (IDS) utilizes various Explainable AI (XAI) methods, which can be categorized into feature-based explanations, perturbation-based explanations, decomposition-based explanations, and hybrid-based explanations. Feature-based explanations aim to determine the impact of each feature on the prediction and are implemented using methods such as SHapley Additive exPlanation (SHAP). Perturbation-based explanations analyze the model’s decision-making process by making small changes to the input data and observing variations in the output space, with techniques like Local Interpretable Model-Agnostic Explanation (LIME). Decomposition-based explanations assign an importance measure to input features by decomposing the model signal with Integrated Gradients being an example of this approach. Hybrid-based

explanations combine multiple XAI techniques, offering more flexible and comprehensive insights into the model’s decisions. In the study conducted by Hariharan et al. (2023), Permutation Importance (PI), SHapley Additive exPlanation (SHAP), Local Interpretable Model-Agnostic Explanation (LIME), and Contextual Importance and Utility (CIU) methods were used. These methods enabled both global and local-level explanations, enhancing the interpretability of IDS models' decision mechanisms [20]. On the other hand, Neupane et al. (2022) examined feature-based, perturbation-based, decomposition-based, and hybrid-based explanations in IDS. The study assessed the effectiveness of different XAI approaches using methods such as SHAP, LIME, and Integrated Gradients. By integrating multiple explanation techniques, hybrid-based explanations provided more flexible and comprehensive insights into IDS models [8]. Both studies contribute significantly to improving the transparency and reliability of IDS models by exploring different aspects of Explainable AI techniques. In this study, SHAP and LIME are implemented as XAI approaches in IDS.

# IDS Classification

IDSs play a crucial role in modern cybersecurity infrastructure, providing significant protection against unauthorized access and malicious activities. Various types of IDS exist, each targeting different aspects of network security, with distinct methods and applications [21]. Table 2 presents an overview of IDS types, detection methods, strengths and weaknesses, and the most suitable systems for their implementation.

Table 2. Classification of IDS Types

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IDS Type | Detection Method | Strengths | Weaknesses | Most Suitable System |
| Network-Based IDS (NIDS) | Network traffic analysis | Effective at monitoring network-level activities | Struggles with large volumes of data | Organizations with high network traffic |
| Host-Based IDS  (HIDS) | Monitoring host  activities | Provides detailed  analysis of specific host systems | Limited to  specific devices | Critical systems or servers |
| Signature-Based IDS | Predefined signature matching | Quickly detects known threats | Ineffective against new threats | Environments requiring rapid identification of known threats |
| Anomaly-Based  IDS | Behavioral analysis | Effective against  unknown threats | High false  positive rate | Organizations with unique  network patterns |
| Hybrid IDS | Combination of  detection methods | Provides  comprehensive security coverage | Complex  deployment and management | Organizations requiring  high-level security |

# Network-Based Intrusion Detection System (NIDS)

Network-Based Intrusion Detection Systems (NIDS) monitor network traffic, analyze data packets, and detect suspicious activities. NIDS are strategically deployed at key points within a network, such as its perimeter or critical subnetworks, to examine packet headers and payloads for known attack patterns and anomalies. According to the literature, NIDS is particularly effective in environments with extensive network infrastructures, as it can monitor large volumes of traffic. However, due to the high data throughput, the system may

become overwhelmed and potentially miss threats [22]. Tools such as Snort and Bro (Zeek) serve as examples of NIDS, offering robust network traffic monitoring capabilities [3].

# Host-Based Intrusion Detection Systems (HIDSs)

HIDSs operate on individual hosts or devices by monitoring activities such as file integrity, log files, and running processes. HIDS provides a detailed analysis of host-specific activities, making it particularly useful for detecting suspicious behavior at the device level. These systems are especially effective in environments that require deep security monitoring, such as critical systems or servers. According to Debar et al. [23], HIDS can detect intrusions that NIDS might overlook, such as unauthorized modifications to system files. However, HIDS is limited to specific devices and does not provide a comprehensive view of network-wide security.

# Signature-Based Intrusion Detection System (SIDS)

A Signature-Based Intrusion Detection System (SIDS) detects malicious activities based on predefined patterns or signatures of known threats. When incoming traffic matches a known signature, the IDS generates an alert. This method is highly effective in detecting known threats quickly and accurately, producing a low false positive rate. However, since the signature database must be frequently updated, it is ineffective against new or unknown threats. Tools such as Snort utilize signature-based systems to enable rapid vulnerability detection in environments where common, known threats are prevalent [24].

# Anomaly-Based Intrusion Detection System (AIDS)

An Anomaly-Based Intrusion Detection System (AIDS) establishes a baseline of normal network behavior and flags deviations from this baseline as potential threats. According to Javitz and Valdes, this method does not rely on predefined signatures but instead identifies unusual behavioral patterns, making it particularly effective against unknown threats and zero-day vulnerabilities. However, anomaly-based IDS tends to produce a higher rate of false positives and requires fine-tuning to improve accuracy. These systems are ideal for organizations with unique network patterns or those requiring adaptive security measures, such as research institutions [25].

# Hybrid Intrusion Detection System (HIDS)

A Hybrid Intrusion Detection System (HIDS) combines the features of both signature-based and anomaly-based detection methods, leveraging the strengths of each approach to provide comprehensive security protection. According to Bace and Mell, hybrid IDS offers a robust, multi-layered security approach that significantly reduces the likelihood of false positives. However, the complexity associated with deployment and management can be a disadvantage. Hybrid IDS is most suitable for organizations that require high-level security across diverse threat environments, ensuring effective detection and management of both known and unknown threats. by integrating different IDS types, organizations can develop a

security strategy that maximizes detection and response capabilities against cyber threats [21].

# Machine Learning in IDS

ML approaches are frequently used in IDSs to detect network attacks. These approaches analyze network traffic to differentiate between normal and malicious behaviors (anomalies). Machine learning builds models by analyzing data and uses these models to predict future events. In IDS, ML is used to identify features and classify attacks. ML methods learn from datasets and detect attacks based on this learned information. Some commonly used ML techniques in IDS include Artificial Neural Networks (ANN), Support Vector Machines (SVMs), Decision Trees (DTs), Naïve Bayes (NB), and Logistic Regression (LR) [26].

# Support Vector Machine (SVM)

SVM is a model used for classification, regression, and outlier detection. It separates data linearly based on a hyperplane. SVM is a binary classifier but can also perform multi-class classification. It is one of the most effective ML techniques for handling non-linear data, although its training time is longer than other techniques [26]. In IDS applications, SVM has been observed to produce successful results due to its ability to handle complex feature spaces [???]. SVM is frequently preferred in various IDS studies [???]. In the study of Mohammadi et al. (2021), a comprehensive analysis of SVM-based intrusion detection and feature selection systems was presented [27].

# Naive Bayes (NB)

Naïve Bayes is a classification algorithm based on Bayes’ Theorem. This classifier assumes that the probability of each feature belonging to a particular class is independent of other features. Predictions are made by calculating the probability of each class and selecting the class with the highest probability. Research findings indicate that Naïve Bayes performs best on real-time datasets, making it highly suitable for fast and efficient IDS applications [26].

# k-Nearest Neighbour (KNN)

KNN is used for both classification and regression problems. It is a lazy learner, meaning it stores all training data and classifies new data based on its similarity to existing records. The classification of test data is based on distance metrics such as Euclidean distance. However, KNN is computationally expensive, especially for large datasets, as it requires storing and comparing all previous data points [26].

# 4.4. Logistic Regression (LR)

LR predicts discrete values (0 or 1) based on independent variables. It estimates whether an event will occur or not using a logistic function. The threshold value is

typically, 0.5, where values above 0.5 are classified as 1, and values below 0.5 are classified as 0. LR is known for its high performance, low computational cost, and interpretability, making it a popular choice for IDS applications [26].

# Karar Ağaçları (DTs)

DTs are used for both classification and regression problems, but they are primarily preferred for classification tasks. While a regression tree deals with continuous values, a decision tree works with categorical labels. Decision trees classify each instance through a sequence of hierarchical decisions, represented in a tree structure, making them highly interpretable and effective for IDSs [26].

# Random Forest (RF)

RF is an ensemble learning method that builds multiple decision trees and aggregates their predictions. The final prediction is obtained by averaging the predictions of all trees. RF generally performs more accurately than individual decision trees and can be used for both classification and regression tasks. Additionally, RF provides an easily interpretable ranking of feature importance, making it useful for feature selection in IDS applications [26]. Table 3 presents studies using ML techniques, their datasets, and corresponding results.

Table 3. Machine Learning-Based IDS Studies (2016-2020) [26]

|  |  |  |  |
| --- | --- | --- | --- |
| References | Dataset | Method | Evaluation |
| Mehmood and Rais (2016) | KDD-99 | SVM, J48, NB, DT | J48 outperformed other algorithms in terms of accuracy and misclassification rate. |
| Belavagi and Muniyal (2016) | NSL-KDD | SVM, LR, NB, RF | RF achieved the highest TPR and the  lowest FPR. |
| Aburomman and Reaz (2016) | KDD-99 | PCA, LDA | Overall accuracy: 0.92162,  FP: 0.0196, FN: 0.10849 |
| Ashfaq et al. (2017) | NSL-KDD | Semi Supervised Learning (SSL) approach based on  fuzziness | – |
| Al-Yaseen et al. (2017) | KDD-99 | SVM, EVM | 95.75% accuracy with a short training time |
| Othman et al. (2018) | KDD-99 | Chi-square, SVM with  SGD | High performance, low FPR. |
| Gautam and Doegar (2018) | KDD-99 | NB, Ensemble Methods, Adaptive Boost and PART | Accuracy: %99,97,  Recall: %99,98,  Precision: %99,99 |
| Saranya et al. (2020) | KDD-99 | LDA, CART, RF | RF outperformed other methods with 99.65% accuracy. |

# Gradient Boosting (GB)

GB is an algorithm developed by Leo Breiman, the creator of the Random Forest algorithm, and is suitable for both regression and classification tasks [28]. Similar to the AdaBoost algorithm, it integrates weak classifiers to form a model primarily composed of decision trees. The goal of the Gradient Boosting approach is to minimize errors by improving predictions based on the learning rate.

# XGboost

XGBoost is an algorithm developed with a focus on speed and performance, utilizing gradient-boosted decision trees. Initially created by Tianqi Chen, it is now supported by the Distributed Machine Learning Community (DMLC). Known as Extreme Gradient Boosting, XGBoost optimizes memory and hardware resource usage for tree-based boosting methods. XGBoost can implement three main gradient boosting techniques: Gradient Boosting, Regularized Boosting, and Stochastic Boosting. Unlike other libraries, it allows the addition and optimization of regularization parameters. This algorithm significantly reduces computation time, efficiently utilizes memory resources, handles missing values, and supports parallel processing in tree structures. Additionally, XGBoost has the capability to continue training pre-trained models with additional data [29].

# Multilayer Perceptrons (MLPs)

MLPs are a widely used neural network model consisting of three layers: input, output, and hidden layers. Each layer is composed of units known as neurons. MLP initially starts with a single hidden layer structure, a predefined number of neurons, and a small training dataset. It is highly effective in solving complex problems due to its multi-layered architecture. Each neuron processes weighted inputs through an activation function before passing the output to the next layer. The model is optimized using the backpropagation algorithm, which updates the network’s weights to improve prediction accuracy [30].

# XAI Methods

This section focuses on analyzing machine learning approaches used to address IDS problems through XAI methods.

# 5.1. Model-Agnostic Methods

Model-agnostic methods are approaches that can work with any machine learning model without requiring knowledge of its internal structure or specific architecture. These methods treat the model as a black box and analyze it based only on input-output relationships. As a result, they offer the ability to explain different types of models. Due to their flexibility, model-agnostic methods can be applied across various domains and, unlike model-specific approaches, are adaptable to a wide range of machine learning models. They are particularly preferred because they provide interpretability independent of the model itself [31].

*LIME (Local Interpretable Model-agnostic Explanations):* is a method that enhances the interpretability of individual predictions and can be applied to any machine learning model without requiring insight into its internal workings. Developed by Ribeiro et al., this approach observes small perturbations in input data to determine which features contribute most to the model's predictions. To improve local accuracy, LIME employs a data-specific surrogate model that approximates the complex model’s behavior in a localized manner. This allows for

a better understanding of which features drive the model’s decision for a given instance. As a result, LIME provides locally interpretable explanations for each individual prediction.

Eqn (1) illustrates the formula underlying LIME, which describes the process of learning a simplified model that approximates the behavior of the original complex model. The formula aims to minimize the differences between the original model's predictions and those of the surrogate model, thereby creating an interpretable local model [32].

|  |  |
| --- | --- |
|  | Eqn(1) |

The parameters in the formula can be explained as follows: L(f ,g ,πx) represents the loss function, which quantifies the difference between the original complex model (f) and the interpretable surrogate model (g). The summation symbol denotes the aggregation over all instances in the local neighborhood. π x(z) is the proximity measure that assigns weights to samples based on their similarity to the instance x. f(z) represents the prediction of the complex model for a given instance z, while g(z′) corresponds to the prediction of the interpretable surrogate model. The term (f(z)−g(z′))² measures the squared difference between the predictions of the two models, ensuring that the surrogate model closely approximates the behavior of the original model. This formulation enables LIME to generate locally interpretable explanations by optimizing the surrogate model to approximate the predictions of the complex model while maintaining interpretability.

*SHAP (Shapley Additive Explanations):* Lundberg and Lee, SHAP utilizes coalition game theory to calculate each feature’s contribution to a prediction. Their method ensures that predictions are fair and consistent and explains the impact of each feature on the prediction [61]. The formula presented in Eqn (2) is associated with the Shapley Additive Explanations

(SHAP) method and is used to calculate the impact of a feature on a model.

|  |  |
| --- | --- |
|  | Eqn(2) |

ϕi represents the contribution of a feature, which is computed as a summation over S subsets. S denotes subsets of features, while m represents the total number of features. The terms f(S∪{i}) and f(S) indicate the change in the model’s output when a feature is included or excluded.

All subsets are considered with equal weight, reflecting the fair distribution property of Shapley values in game theory. In the study [33], Eqn (3) formula is introduced as a crucial tool for enhancing the explainability of machine learning models, enabling the decomposition of model predictions based on individual features.

*Anchors:* Ribeiro et al. developed the Anchors method, which provides local explanations for black-box models by generating if-then rules that explain specific predictions with high precision [34].

*LORE (Local Rule-based Explanations):* Guidotti et al. developed LORE, which provides local explanations by generating a decision tree based on synthetic data. This method offers both the main decision rule and alternative decision rules for a given prediction [35].

# 5.2 Post-hoc Interpretation

Post-hoc interpretation refers to methods aimed at explaining the decisions of machine learning models after the model has been trained without directly accessing the model’s internal structure or operational mechanisms; these methods attempt to understand the model based on its inputs and outputs. Such techniques are particularly useful for enhancing the interpretability of black-box models, making their predictions more transparent and understandable [31].

*GRAD-CAM (Gradient-weighted Class Activation Mapping):* Selvaraju et al. developed GRAD-CAM, a technique that generates class-specific heatmaps to enhance the interpretability of deep learning models in image classification tasks [36].

*CEM (Contrastive Explanation Method):* Dhurandhar et al. developed CEM, which identifies the minimum set of features necessary for a model to predict a specific class, as well as the features required to alter the prediction to a different class [37].

Overall, XAI methods are essential for enhancing the transparency, fairness, reliability, usability, and trustworthiness of AI systems. By enabling users to understand model decisions, XAI supports model validation, legal compliance, and user acceptance. Explanations play a crucial role in model verification and debugging processes [38]. Table 4 provides a general framework for the applied XAI methods, offering an overview of their applications [38].

Table 4. XAI Methods

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | Full Name | Type | Explainability | Scope | Description |
| LIME | Local Interpretable Model-Agnostic Explanations | Model-Agnostic | Post-hoc | Local | Provides interpretable models for individual predictions. It perturbs the input data and observes the resulting changes  in predictions. |
| SHAP | SHapley Additive Explanations | Model-Agnostic | Post-hoc | Local/ Global | Uses cooperative game theory-based SHapley values to calculate and explain the  contribution of each feature to predictions. |
| Anchors | Anchors | Model-Agnostic | Post-hoc | Local | Identifies key features that influence a specific prediction by highlighting  high-sensitivity decision rules. |
| LORE | Local Rule-Based Explanations | Model-Agnostic | Post-hoc | Local | Generates local explanations by constructing a decision tree based on synthetic data. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GRAD-CAM | Gradient-weighted Class Activation Mapping | Model-Specific | Post-hoc | Local | Creates class-specific heatmaps to visualize the most influential regions in an image for the model’ s  prediction. |
| CEM | Contrastive Explanation Method | Model-Agnostic | Post-hoc | Local | Identifies the most critical features for a given prediction and emphasizes the minimum changes required to alter the prediction to a different class. |

In this study, Logistic Regression, KNN, SVM, Decision Tree, Random Forest Classifier, Gradient Boosting Classifier, Gaussian Naïve Bayes Classifier, XGBoost, and MLP algorithms were tested on the datasets of UNSW-NB15, NSL-KDD, CICIDS2017, CIRA-CIC-DoHBrw2020, and CICIoT2023 for solving IDS problems. The outputs were evaluated using XAI approaches. The overall framework used in this study is presented in Figure 1.

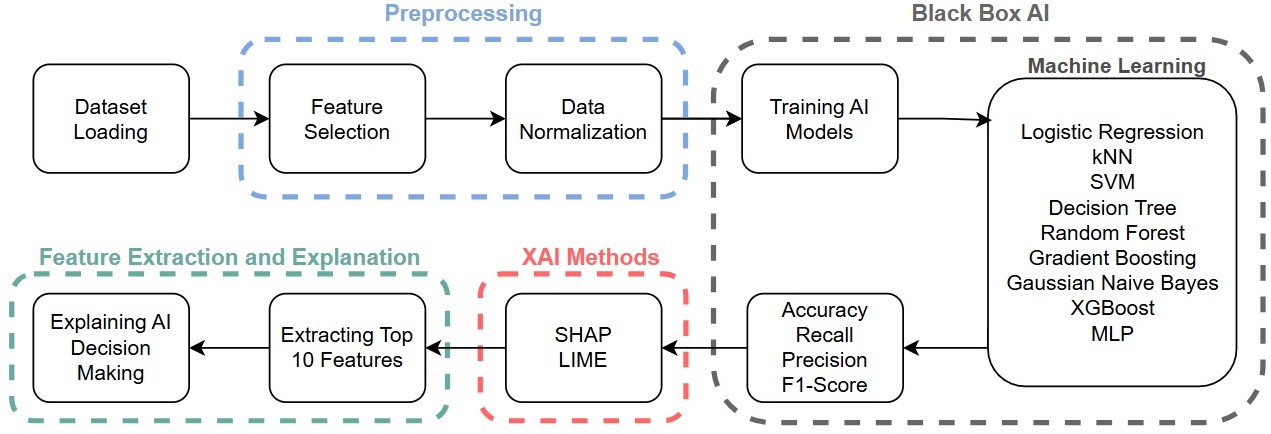


Figure 1. Integration of Machine Learning and XAI Methods in Intrusion Detection Systems

According to the general framework process presented in Figure 1, the workflow begins with loading the dataset, followed by preprocessing steps such as feature selection and normalization. Subsequently, various machine learning models are trained, and their performance is evaluated using accuracy, precision, recall, and F1-score metrics. To enhance interpretability, XAI methods such as SHAP and LIME are applied to explain model decisions and identify the most important features.

# Experimental Studies

This section presents the results for experimental studies conducted on Intrusion Detection System (IDS) problems using various datasets and machine learning methods. The experiments were performed on the UNSW-NB15, NSL-KDD, CICIDS2017, CIRA-CIC-DoHBrw2020, and CICIoT2023 datasets, utilizing different machine learning algorithms such as Logistic Regression, KNN, SVM, DT, RF Classifier, GB Classifier, Gaussian Naïve Bayes Classifier, XGBoost, and MLP. Additionally, the interpretability of these models was analyzed using XAI methods such as SHAP and LIME. First, experiments were conducted on the UNSW-NB15 dataset, where Logistic Regression, KNN, SVM, DT, RF, GB Classifier, Gaussian Naïve Bayes Classifier, XGBoost, and MLP were tested. The models were evaluated based on Accuracy, Recall, Precision, and F1-score metrics. The results obtained are presented in Table 5.

Table 5. Results for Experiments Conducted on UNSW-NB15 Dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithms** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| Logistic Regression | 0.92 | 0.92 | 0.92 | 0.92 |
| KNN | 0.95 | 0.95 | 0.95 | 0.95 |
| SVM | 0.94 | 0.94 | 0.94 | 0.94 |
| Decision Tree | 0.96 | 0.96 | 0.96 | 0.96 |
| Random Forest Classifier | 0.97 | 0.97 | 0.97 | 0.97 |
| Gradient Boosting Classifier | 0.95 | 0.95 | 0.95 | 0.95 |
| Gaussian Naive Bayes Classifier | 0.76 | 0.76 | 0.79 | 0.76 |
| XGBoost | 0.97 | 0.97 | 0.97 | 0.97 |
| MLP | 0.96 | 0.96 | 0.96 | 0.96 |

In the subsequent tests, the same approaches were applied to NSL-KDD dataset, and the results are presented in Table 6. The XAI analyses and overall evaluations of these results are provided in detail below.

Table 6. Results for Experiments Conducted on NSL-KDD Dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithms** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| Logistic Regression | 0.95 | 0.95 | 0.96 | 0.95 |
| KNN | 0.99 | 0.99 | 0.99 | 0.99 |
| SVM | 1.00 | 1.00 | 1.00 | 1.00 |
| Decision Tree | 0.99 | 0.99 | 0.99 | 0.99 |
| Random Forest Classifier | 0.99 | 0.99 | 0.99 | 0.99 |
| Gradient Boosting Classifier | 0.99 | 0.99 | 0.99 | 0.99 |
| Gaussian Naive Bayes Classifier | 0.89 | 0.92 | 0.87 | 0.89 |
| XGBoost | 0.99 | 0.99 | 0.99 | 0.99 |
| MLP | 0.99 | 0.99 | 0.99 | 0.99 |

The CICIDS2017 dataset was another test set used in this study. Similar to previous experiments, Logistic Regression, KNN, SVM, DT, RF, GB Classifier, Gaussian Naïve Bayes Classifier, XGBoost, and MLP were trained and tested on this dataset. The performance scores obtained from these experiments are presented in Table 7.

Table 7. Results for Experiments Conducted on CICIDS2017 Dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithms** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| Logistic Regression | 0.98 | 0.99 | 0.98 | 0.99 |
| KNN | 1.00 | 1.00 | 1.00 | 1.00 |
| SVM | 1.00 | 1.00 | 1.00 | 1.00 |
| Decision Tree | 1.00 | 1.00 | 1.00 | 1.00 |
| Random Forest Classifier | 1.00 | 1.00 | 1.00 | 1.00 |
| Gradient Boosting Classifier | 1.00 | 1.00 | 1.00 | 1.00 |
| Gaussian Naive Bayes Classifier | 0.99 | 0.80 | 0.88 | 0.80 |
| XGBoost | 1.00 | 1.00 | 1.00 | 1.00 |
| MLP | 1.00 | 1.00 | 1.00 | 1.00 |

To observe and analyze the performance of machine learning approaches on up-to-date datasets, this study utilized the CIRA-CIC-DoHBrw2020 dataset. This dataset was tested using commonly preferred approaches in the literature, including Logistic Regression, KNN, Support Vector Machine (SVM), Decision Tree, Random Forest Classifier, Gradient Boosting Classifier, Gaussian Naïve Bayes Classifier, and XGBoost Classifier. Fundamentally, this dataset consists of two layers. In the first layer, the dataset is categorized into “DoH” and “non-DoH” traffic; In the second layer, the input data is further classified into “benign” and “malicious” traffic. As these datasets were tested separately, two different tables were obtained. The test scores from these experiments are presented in Table 8.

Table 8. Results for Experiments Conducted on CIRA-CIC-DoHBrw2020 Dataset

1. DoH and non-DoH traffic

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithms** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| Logistic Regression | 0.92 | 0.92 | 0.92 | 0.92 |
| KNN | 0.99 | 0.99 | 0.99 | 0.99 |
| SVM | 0.97 | 0.97 | 0.97 | 0.97 |
| Decision Tree | 1.00 | 1.00 | 1.00 | 1.00 |
| Random Forest Classifier | 1.00 | 1.00 | 1.00 | 1.00 |
| Gradient Boosting Classifier | 0.99 | 0.99 | 0.99 | 0.99 |
| Gaussian Naive Bayes Classifier | 0.87 | 0.87 | 0.87 | 0.87 |
| XGBoost | 1.00 | 1.00 | 1.00 | 1.00 |
| MLP | 0.99 | 0.99 | 0.99 | 0.99 |

1. benign and malicious traffic

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithms** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| Logistic Regression | 0.98 | 0.97 | 0.98 | 0.97 |
| KNN | 1.00 | 1.00 | 1.00 | 1.00 |
| SVM | 1.00 | 1.00 | 1.00 | 1.00 |
| Decision Tree | 1.00 | 1.00 | 1.00 | 1.00 |
| Random Forest Classifier | 1.00 | 1.00 | 1.00 | 1.00 |
| Gradient Boosting Classifier | 1.00 | 1.00 | 1.00 | 1.00 |
| Gaussian Naive Bayes Classifier | 0.85 | 0.94 | 0.85 | 0.88 |
| XGBoost | 1.00 | 1.00 | 1.00 | 1.00 |
| MLP | 1.00 | 1.00 | 1.00 | 1.00 |

Finally, the CICIoT2023 dataset was used to analyze the methods applied to the IDS problem. This dataset is significant due to its recentness and the diverse features it contains. In this study, examples from the first eight files of the dataset were combined, and machine learning methods were tested. The results obtained from these experiments were presented in Table 9. The remaining tables 10-14 show a comparison table of the data sets used and studies in literature.

Table 9. Results for Experiments Conducted on CICIoT2023 Dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithms** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| Logistic Regression | 0.84 | 0.86 | 0.84 | 0.82 |
| KNN | 0.94 | 0.94 | 0.94 | 0.94 |
| SVM | 0.82 | 0.83 | 0.82 | 0.78 |
| Decision Tree | 0.99 | 0.99 | 0.99 | 0.99 |
| Random Forest Classifier | 0.99 | 0.99 | 0.99 | 0.99 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Gradient Boosting Classifier | 0.99 | 0.99 | 0.99 | 0.99 |
| Gaussian Naive Bayes Classifier | 0.76 | 0.80 | 0.76 | 0.73 |
| XGBoost | 0.99 | 0.99 | 0.99 | 0.99 |
| MLP | 0.98 | 0.98 | 0.98 | 0.98 |

Table 10. Results for Experiments Conducted on UNSW-NB15 Dataset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Studies** | **Methods** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| [13] | Random Forest | 0.97 | 0.98 | 0.98 | 0.98 |
| [39] | Random Forest | 1.00 | 1.00 | 1.00 | 1.00 |
| [40] | Random Forest | 0.97 | N/A | N/A | N/A |
| [41] | Random Forest | 0.97 | 0.81 | 0.99 | 0.89 |
| [42] | SVM | 0.98 | 0.97 | 0.98 | 0.98 |
| [43] | Random Forest | 0.98 | 0.98 | 0.98 | 0.98 |
| Our study | XGBoost | 0.99 | 0.99 | 0.99 | 0.99 |

Table 11. Results for Experiments Conducted on NSL-KDD Dataset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Studies** | **Methods** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| [14] | MLP | 0.98 | 0.98 | 0.97 | 0.98 |
| [14] | KNN | 0.98 | 0.98 | 0.98 | 0.98 |
| [15] | LSTM | 0.80 | 0.95 | 0.69 | 0.80 |
| [19] | RF | 0.99 | 0.99 | 0.99 | 0.99 |
| [44] | RF | 0.99 | N/A | N/A | N/A |
| [45] | XGBoost | 0.97 | 0.97 | 0.97 | 0.97 |
| [46] | NB | 0.97 | 0.97 | 0.98 | N/A |
| [47] | RF | 0.99 | N/A | N/A | N/A |
| [48] | RF | 0.99 | N/A | N/A | N/A |
| Our study | XGBoost | 0.99 | 0.99 | 0.99 | 0.99 |

Table 12. Results for Experiments Conducted on CIRA-CIC-DoHBrw2020 Dataset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Studies** | **Methods** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| [49] | RF, DT, SVM | 0.99 | N/A | N/A | N/A |
| [50] | GB | 0.90 | 0.90 | 0.90 | N/A |
| Our study | XGBoost | 1.00 | 1.00 | 1.00 | 1.00 |

Table 13. Comparative Results for CICIDS2017 Dataset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Studies** | **Methods** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| [51] | KNN | 0.99 | 0.99 | 0.99 | 0.99 |
| SVM | 0.75 | 0.99 | 0.75 | 0.76 |
| DT | 0.99 | 0.94 | 0.99 | 0.99 |
| [52] | KNN | N/A | 0.96 | 0.96 | 0.96 |
| RF | N/A | 0.98 | 0.97 | 0.97 |
| MLP | N/A | 0.77 | 0.83 | 0.76 |
| NB | N/A | 0.88 | 0.04 | 0.04 |
| [53] | RF | 0.99 | 0.99 | 0.99 | 0.99 |
| GNB | 0.81 | 0.03 | 0.73 | 0.07 |
| MLP | 0.99 | 0.98 | 0.85 | 0.91 |
| [54] | MLP | 0.99 | 0.99 | 0.99 | 0.62 |
| [55] | RF | 0.99 | N/A | 0.99 | N/A |
| GNB | 0.33 | 0.87 | 0.40 | N/A |
| [52] | LR | 0.99 | 1.00 | 1.00 | 1.00 |
| [56] | XGBoost | 0.98 | 0.98 | 0.98 | 0.98 |
| Our study | XGBoost | 1.00 | 1.00 | 1.00 | 1.00 |

Table 14. Comparative Results for CICIoT2023 Dataset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Studies** | **Methods** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| [57] | SVM | 0.98 | 0.88 | 0.68 | 0.75 |
| DT | 0.99 | 0.95 | 0.95 | 0.95 |
| RF | 0.99 | 0.96 | 0.96 | 0.96 |
| LR | 0.98 | 0.86 | 0.89 | 0.87 |
| [58] | DT | 0.99 | 0.87 | 0.91 | 0.89 |
| RF | 0.99 | 0.99 | 0.99 | 0.99 |
| k-NN | 0.99 | 0.98 | 0.99 | 0.99 |
| LR | 0.88 | 0.94 | 0.52 | N/A |
| Naive Bayes | 0.97 | 0.94 | 0.99 | 0.97 |
| [59] | MLP | 0.99 | 0.99 | 0.99 | 0.99 |
| [60] | XGBoost | 0.97 | 0.97 | 0.97 | 0.97 |
| Our study | XGBoost | 0.99 | 0.99 | 0.99 | 0.99 |

The results presented above are significant for analyzing and comparing the performance of machine learning algorithms across different datasets. Additionally, the machine learning approaches used in this study have been compared with other studies in the literature. A comparative analysis between the results for previous studies and the experimental results obtained in this study indicates that the XGBoost algorithm achieved the best performance across all datasets. This superior performance can be attributed to comprehensive data preprocessing and model optimization processes. For the UNSW-NB15, NSL-KDD, CICIDS2017, CIRA-CIC-DoHBrw2020, and CICIoT2023 datasets, key preprocessing steps included cleaning noisy and missing data, selecting high-information features, and applying data standardization techniques. Additionally, dataset-specific feature analysis and hyperparameter optimization enabled XGBoost to learn more effectively from the data. Furthermore, feature engineering played a crucial role in enhancing model performance. Key variables such as protocol types, traffic volume, and session duration were carefully extracted, leading to higher accuracy, recall, precision, and F1-score compared to previous studies. These findings highlight that meticulously applied data preprocessing and the robust structure of XGBoost offer an effective solution for IDS systems.

First, the SHAP approach was applied to the XGBoost model trained and tested on the UNSW-NB15 dataset. The output presented in Figure 2, using the SHAP waterfall plot, illustrates how different features contribute to the model reaching a reconstruction error of f(x) = 10.117. The baseline value is E[f(x)] = 1.342, and it is observed that some features significantly increase or decrease the model’s output. Among the most influential features, “sttl” contributed the most, increasing the model output by +2.26, while “ct\_dst\_sport\_ltm” also had a significant positive impact, adding +1.93 to the model output. On the other hand, “ct\_dst\_src\_ltm” had a negative effect of -1.33, reducing the model output, whereas “smean” and “service” contributed +1.61 and +1.25, respectively. The impact of other features was relatively lower, and their effects on the model output are clearly visualized in the waterfall plot, where blue represents decreasing effects and red represents increasing effects.

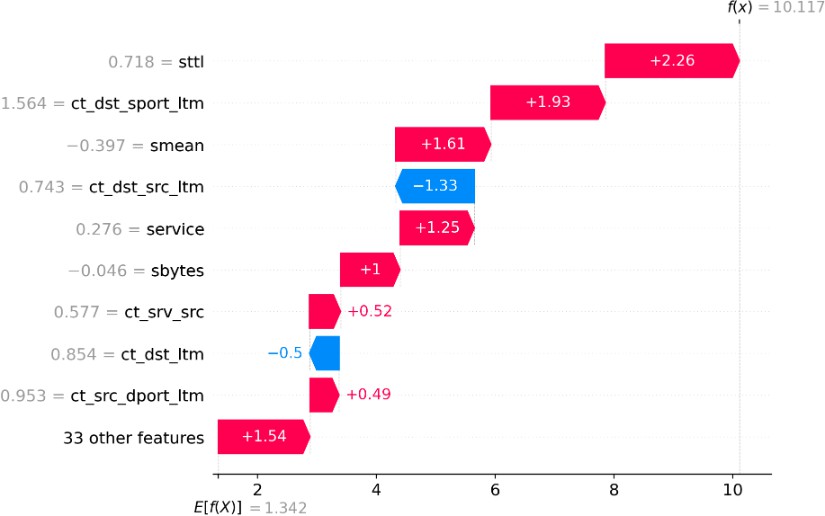


Figure 2. XGBoost-SHAP Waterfall Analysis Output on the UNSW-NB15 Dataset

Figure 3 illustrates which features play a more significant role in the decision-making process of the XGBoost model. In this model, the LIME method was applied, and the attack detection probability was found to be 100%. Among the key features supporting the attack classification, “dbytes,” “udp,” “smean,” and “is\_ftp\_login” were identified with respective values of 1.73, 0, 0.01, and -0.09. On the other hand, features that contributed to the model’s preference for the normal class, including “sttl,” “tcp,” “ct\_dst\_src\_ltm,” and “http”, had values of -1.17, 0, -0.59, and 0, respectively.

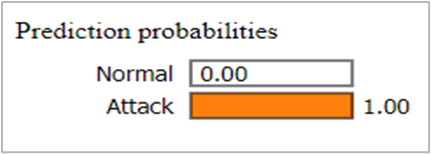
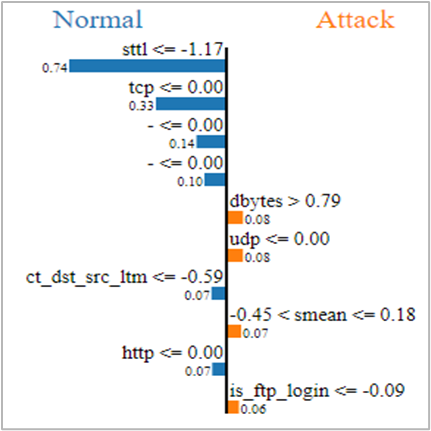


Figure 3. XGBoost-LIME Output on the UNSW-NB15 Dataset

In this study, another dataset used for XAI analysis was NSL-KDD. The XGBoost algorithm modeled on this dataset was evaluated using the SHAP approach. The output presented in Figure 4, through the SHAP waterfall plot, illustrates how different features contribute to the model reaching a reconstruction error of f(x) = -12.006. The baseline value was determined as E[f(x)] = 0.21, where the “service” feature significantly increased the model output, while the

“src\_bytes” feature caused a noticeable decrease. In the graph, blue represents decreasing effects, while red represents increasing effects, clearly visualizing each feature’s impact on the model output. This approach allows the critical factors influencing the model's decision-making process to be visualized from different perspectives, making them more interpretable.

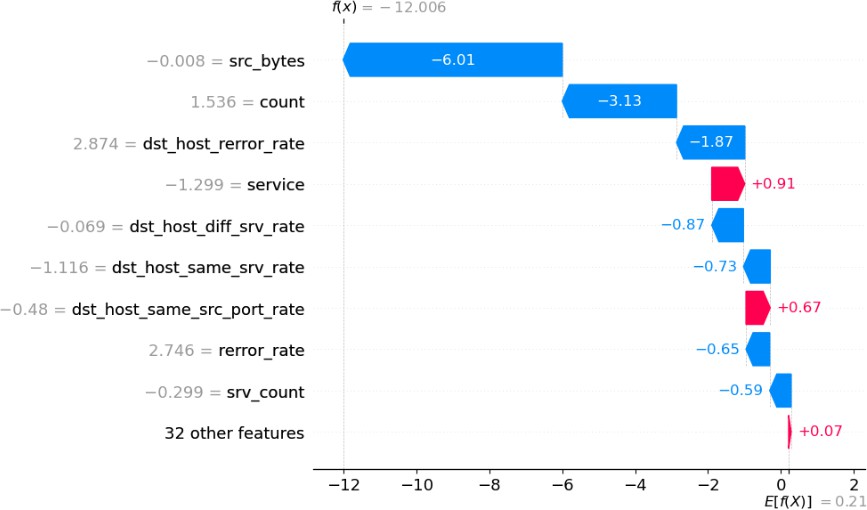


Figure 4. XGBoost-SHAP Waterfall Analysis Output on the NSL-KDD Dataset

Figure 5 illustrates which features contributed the most to the decision-making process of the XGBoost model. When the LIME method was applied to this model, the attack detection probability was found to be 100%. Among the features influencing this prediction, “src\_bytes,” “su\_attempted,” “dst\_host\_count,” “num\_compromised,” “dst\_bytes,” “srv\_serror\_rate,” “wrong\_fragment,” and “rerror\_rate” positively contributed to the attack classification, with respective values of -0.01, -0.02, -1.73, -0.01, 0, -0.63, -0.09, and -0.37. On the other hand, “protocol\_type” and “root\_shell” contributed to the normal classification, with values of -0.12 and -0.04, respectively.

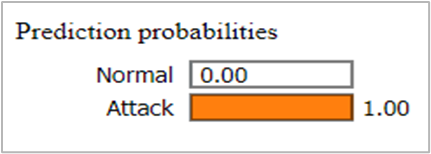
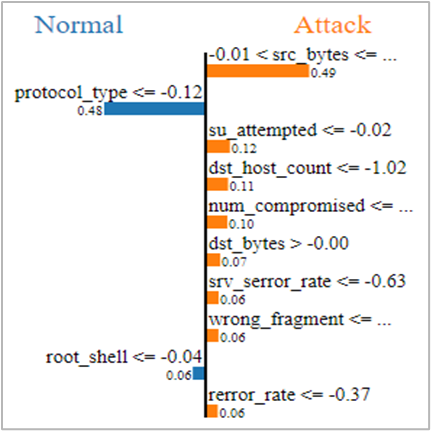


Figure 5. XGBoost-LIME Output on the NSL-KDD Dataset

Another dataset analyzed in the XAI evaluations was CIRA-CIC-DoHBrw2020. This dataset was used for training and testing machine learning models and was structured into two separate layers: the first layer distinguishes between "DoH" and "non-DoH" files, while the second layer categorizes files as "benign" or "malicious". For each layer, separate models were trained, and SHAP and LIME analyses were performed. The SHAP method was applied to the XGBoost model based on the first layer of the CIRA-CIC-DoHBrw2020 dataset. The SHAP waterfall plot, presented in Figure 6, illustrates how different features influenced the model's reconstruction error of f(x) = 4.108. The baseline value was E[f(x)] = 4.489, where the "PacketLengthStandardDeviation" feature significantly increased the model output, while the "PacketLengthVariance" feature substantially decreased the output. In the visualization, red represents increasing effects, while blue represents decreasing effects, making the key factors in the model's decision-making process more interpretable.

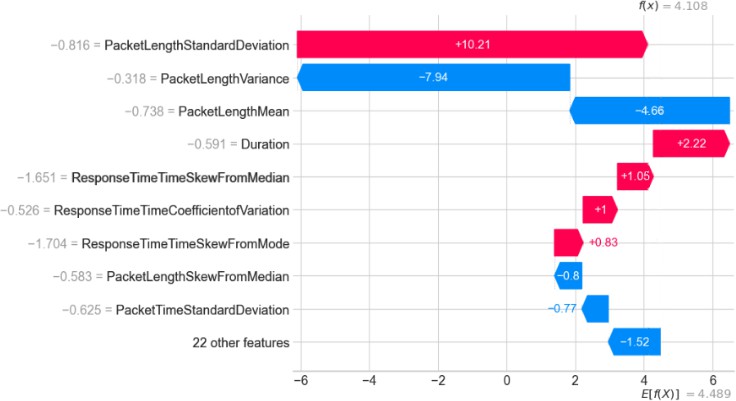
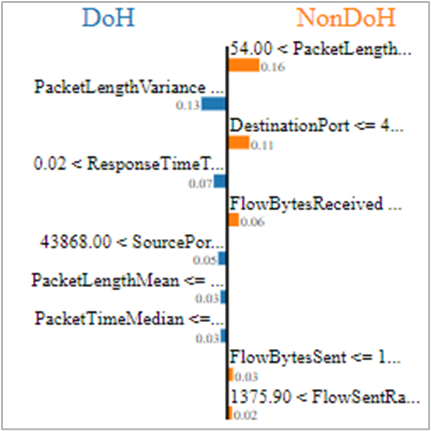


Figure 6. SHAP Waterfall Plot for XGBoost Model on the First Layer of the CIRA-CIC-DoHBrw2020 Dataset

Figure 7 illustrates the distribution of features influencing the classification decision of the XGBoost model. According to the LIME analysis, the probability of detecting DoH is 0%, while the probability of detecting Non-DoH is 100%. In this classification, the features that positively contributed to the DoH classification include “PacketLengthVariance” (30.25), “ResponseTimeTimeMedian” (0.02), “SourcePort” (50582), “PacketLengthMean” (60.50), and “PacketLengthMedian” (0.01). For the Non-DoH classification, the positively contributing features are “PacketLengthMode” (55), “DestinationPort” (443), “FlowBytesReceived” (66), “FlowBytesSent” (55), and “FlowSentRate” (3614.14).



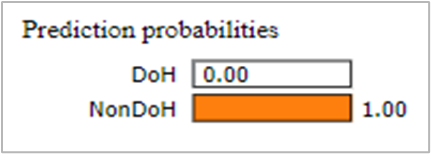


Figure 7. Results for XGBoost-LIME

Beyond the first layer of CIRA-CIC-DoHBrw2020 dataset, the XGBoost model was also applied to its second layer, with global evaluations conducted using the SHAP method. The SHAP waterfall plot, presented in Figure 8, illustrates the extent to which different features contributed to the model reaching a reconstruction error of f(x) = 14.402. The baseline value was determined as E[f(x)] = 11.896, where the “PacketLengthMode” feature significantly increased the model output, while the “PacketTimeMedian” feature notably decreased the model outcome. In the visualization, blue highlights features with a decreasing effect, whereas red emphasizes features with an increasing effect. This approach provides a clear evaluation of the key factors influencing the model’s decision-making process.

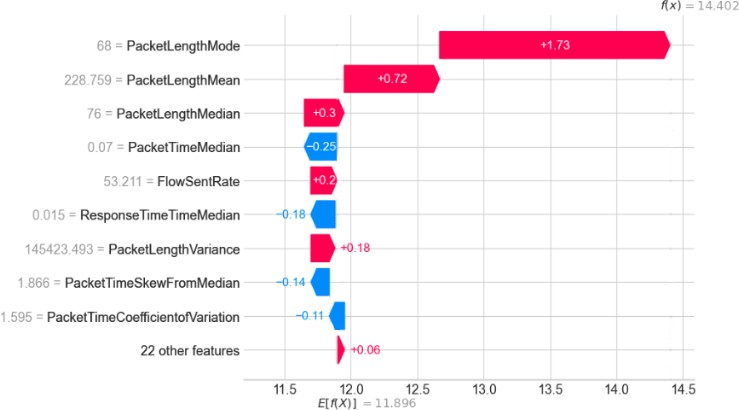


Figure 8. SHAP Waterfall Plot for XGBoost Model on the Second Layer of CIRA-CIC-DoHBrw2020 Dataset

Figure 9 analyzes the key features influencing the benign and malicious classifications in the XGBoost model. According to the LIME method, the probability of benign detection is 0%, while the probability of malicious detection is 100%. The features that positively contributed

to the benign classification include “PacketLengthMode” (68), “PacketTimeMedian” (0.07), “PacketTimeVariance” (189.81), “PacketTimeSkewFromMode” (0.63), and “SourcePort” (45252). On the other hand, the features that positively influenced the malicious classification are “PacketLengthMedian” (76), “PacketTimeMode” (0), “PacketLengthVariance”

(145423.49), “FlowReceivedRate” (142.14), and “PacketLengthMean” (228.76).

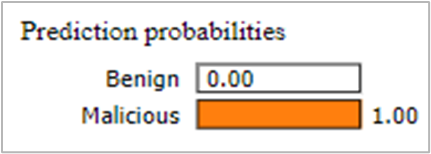
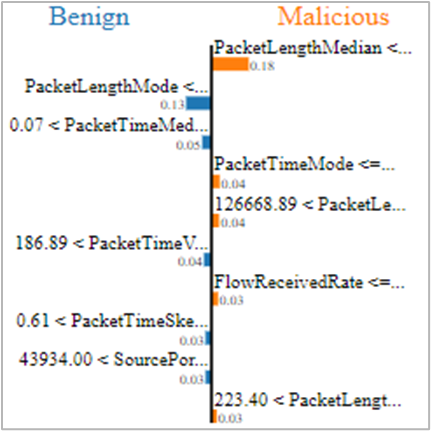


Figure 9. XGBoost-LIME Output (Second Layer)

In this study, another dataset used to conduct a more comprehensive and effective XAI analysis on the applied machine learning methods was CICIDS2017 dataset. Based on the test results from this dataset, the XGBoost model, which achieved the best performance, was analyzed using SHAP. The SHAP waterfall plot in Figure 10 details how different features influenced the model’s reconstruction error of f(x) = -13.084. The baseline value was determined as E[f(x)] = -14.191, where the “Init\_Win\_bytes\_backward” feature significantly increased the model output, whereas the “Destination Port” feature substantially reduced it. The effects of the features are visually represented with red indicating an increasing effect and blue denoting a decreasing effect. These results clearly highlight the key factors influencing the decision-making process, providing a transparent evaluation of the model’s outputs.

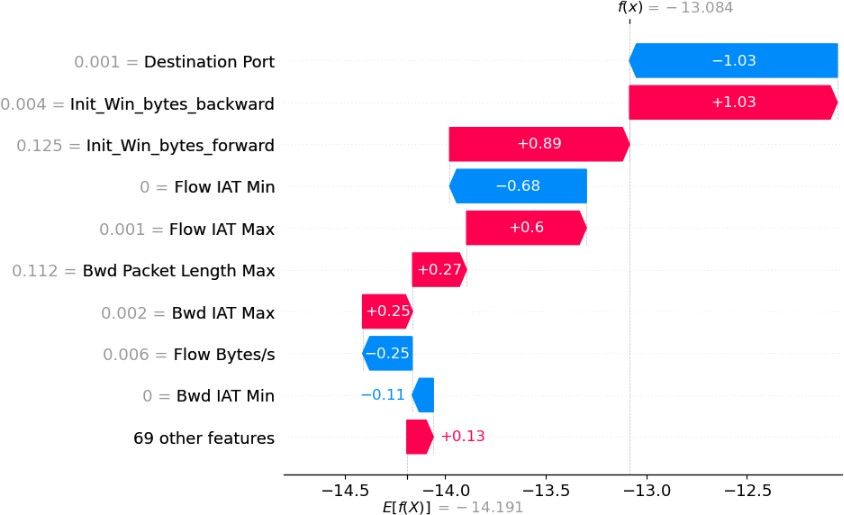


Figure 10. LR-SHAP Waterfall Output on the CICIDS2017 Dataset

Figure 11 illustrates the features that influenced the decision-making process of the XGBoost model. The LIME analysis revealed that the model determined the attack probability to be 0%. The features that positively contributed to the attack classification include “Bwd IAT Max” ,“Init\_Win\_bytes\_forward” , “Flow IAT Std”, “Flow IAT Max” and “PSH Flag Count” with respective values of 0, 0.13, 0, 0, and 1. On the other hand, the features that supported the normal classification were “Destination Port” , “Active Std” ,“Idle Mean” and “Init\_Win\_bytes\_backward” all contributing to the model’s decision with a value of 0.

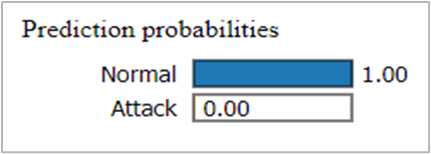
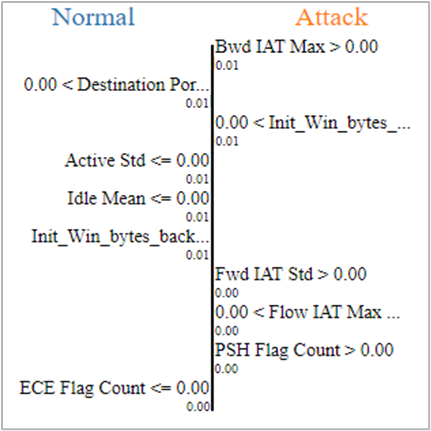


Figure 11. XGBoost-LIME Output on the CICIDS2017 Dataset

The final recent dataset analyzed in this study was CICIoT2023, one of the latest datasets introduced in the literature for this field. The performance of machine learning algorithms was comparatively evaluated on this dataset, and XAI analyses were conducted. First, the XGBoost model applied to this dataset was analyzed using SHAP. The SHAP waterfall plot in Figure 12 illustrates which features contributed to the model reaching a reconstruction error

of f(x) = -14.659. The baseline value was determined as E[f(x)] = -14.437, where the “flow\_duration” feature significantly increased the model output, while the “IAT” feature substantially decreased it. The figure presents red for increasing effects and blue for decreasing effects, clearly depicting the impact of each feature on the model’s output. This visualization provides a clear understanding of the critical factors influencing the model's decision-making process.

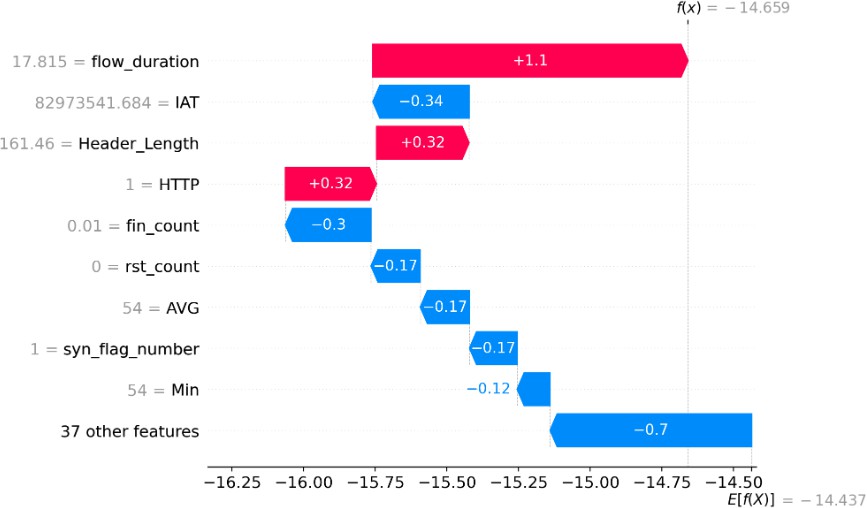


Figure 12. XGBoost-SHAP Waterfall Analysis Output on the CICIoT2023 Dataset

Figure 13 illustrates the features that influenced the decision-making process of the XGBoost model. When the LIME method was applied, the model determined the attack probability to be 0%. The features that positively contributed to the attack classification include “Rate”, “Min” , “HTTPS” , “HTTP” and “Weight” with respective values of -0.09, -0.35, -0.24,

-0.23, and 0. Additionally, the features that supported the normal classification were “rst\_count”, “IAT” , “Duration” and “Covariance” with values of -0.12, 0, -0.17, and -0.1, respectively.

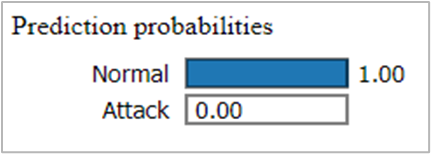
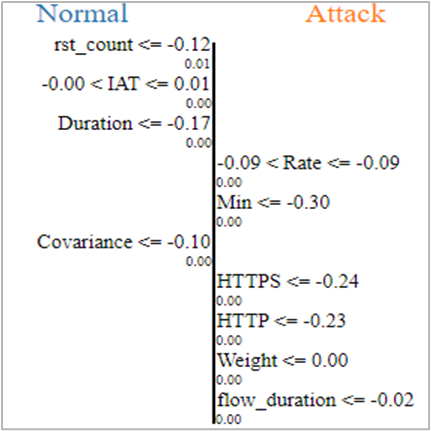


Figure 13. XGBoost-LIME Output on the CICIoT2023 Dataset

# Conclusion and Recommendations

This study comprehensively examines the use of ML and explainable artificial intelligence (XAI) techniques to enhance network security. It provides a detailed analysis of IDSmethods available in the literature and their applicability to network environments. Experiments were conducted on UNSW-NB15, NSL-KDD, CICIDS2017, CIRA-CIC-DoHBrw2020, and CICIoT2023 datasets using Logistic Regression, KNN, SVM, Decision Tree, Random Forest, Gradient Boosting, Gaussian Naïve Bayes, XGBoost, and MLP algorithms. The models were evaluated using accuracy, recall, precision, and F1-score as performance metrics. The results indicate that Decision Tree, Random Forest, Gradient Boosting, and KNN algorithms achieved high accuracy and reliability. The application of XAI techniques, specifically LIME and SHAP, provided insights into the decision-making mechanisms of the models, helping to identify the most influential features in attack detection. This study contributes to the literature by demonstrating the effectiveness of ML and XAI techniques in detecting network vulnerabilities. Moreover, the use of XAI ensures that IDS systems become more transparent and interpretable. Future research in this domain should focus on expanding datasets and exploring more advanced solutions to enhance cybersecurity. Ultimately, this study highlights that ML techniques, combined with XAI methods, can be effectively utilized for security vulnerability detection. This approach enhances the transparency and reliability of security systems, serving as a foundation for future research and applications. Future research directions include the integration of DL techniques and their application to larger-scale datasets. Advanced DL models such as Deep Neural Networks (DNN), RNNs, and CNN could be explored to evaluate their impact on IDS performance. These models have the potential to detect more complex and stealthy attack types, further advancing the field of cybersecurity.

This study systematically explores the application of ML and XAI techniques in IDS to enhance network security. A comprehensive literature review was conducted, followed by experimental evaluations using UNSW-NB15, NSL-KDD, CICIDS2017, CIRA-CIC-DoHBrw2020, and CICIoT2023 datasets. Multiple ML algorithms, including Logistic Regression, KNN, SVM, Decision Tree, Random Forest, Gradient Boosting, Gaussian Naïve Bayes, XGBoost, and MLP were tested and analyzed using accuracy, recall, precision, and F1-score. The findings indicate that DT, RF, GB, and KNN models achieved high accuracy and reliability. The integration of XAI methods, specifically LIME and SHAP, provided critical insights into the decision-making mechanisms of IDS models, improving transparency and interpretability. These techniques facilitated the identification of key features influencing attack detection, contributing to the development of more trustworthy and effective cybersecurity solutions.

This study underscores the effectiveness of ML and XAI in detecting network vulnerabilities and highlights their role in improving IDS transparency. Future research should focus on expanding datasets and exploring more advanced techniques to further enhance cybersecurity measures. In particular, DL models, including DNN, RNN, and CNN provide promising opportunities for detecting more advanced and hard-to-detect cyber threats. Investigating these models on large-scale datasets will provide deeper insights into improving IDS performance, reinforcing network security in increasingly complex digital environments. This research serves as a foundation for future advancements in AI-driven cybersecurity, emphasizing the importance of explainability, model reliability, and adaptability in modern IDS frameworks.

# Declaration

This is to confirm that this manuscript has not been published elsewhere and is not under consideration by any other journal. All authors have approved the manuscript, including the full text and the number of figures/tables, and have declared that there are no conflicts of interest.

# Acknowledgment

This study was supported by the Scientific Research Projects Coordination Unit (BAP) of Gazi University, under grant number FGA-2022-8043.

# References

1. Fuchsberger, A. (2005). Intrusion detection systems and intrusion prevention systems.

*Information Security Technical Report*, *10*(3), 134-139.

1. Denning, D. E. (1987). An intrusion-detection model. *IEEE Transactions on software engineering*, (2), 222-232.
2. Roesch, M. (1999, November). Snort: Lightweight intrusion detection for networks. In *Lisa*

(Vol. 99, No. 1, pp. 229-238).

1. Javaid, A., Niyaz, Q., Sun, W., & Alam, M. (2016, May). A deep learning approach for network intrusion detection system. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)* (pp. 21-26).
2. Aldweesh, A., Derhab, A., & Emam, A. Z. (2020). Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowledge-Based Systems*, *189*, 105124.
3. Karatas, G., Demir, O., & Sahingoz, O. K. (2018, December). Deep learning in intrusion detection systems. In *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)* (pp. 113-116). IEEE.
4. Thakkar, A., & Lohiya, R. (2020). A review of the advancement in intrusion detection datasets. *Procedia Computer Science*, *167*, 636-645.
5. Neupane, S., Ables, J., Anderson, W., Mittal, S., Rahimi, S., Banicescu, I., & Seale, M. (2022). Explainable intrusion detection systems (x-ids): A survey of current methods, challenges, and opportunities. IEEE Access, 10, 112392-112415.
6. Thakkar, A., & Lohiya, R. (2021). A review on machine learning and deep learning

perspectives of IDS for IoT: recent updates, security issues, and challenges. *Archives of Computational Methods in Engineering*, *28*(4), 3211-3243.

1. Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2019). A survey of network-based intrusion detection data sets. *Computers & Security*, *86*, 147-167.
2. Meena, G., & Choudhary, R. R. (2017, July). A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA. In *2017 International Conference on Computer, Communications and Electronics (Comptelix)* (pp. 553-558). IEEE.
3. Kasongo, S. M., & Sun, Y. (2020). Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. *Journal of Big Data*, *7*(1), 105.
4. Türk, F. (2023). Analysis of intrusion detection systems in UNSW-NB15 and NSL-KDD datasets with machine learning algorithms. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 12(2), 465-477.
5. Vibhute, A. D., Patil, C. H., Mane, A. V., & Kale, K. V. (2024). Towards detection of network anomalies using machine learning algorithms on the NSL-KDD benchmark datasets. *Procedia Computer Science*, *233*, 960-969.
6. Zhang, C., Jia, D., Wang, L., Wang, W., Liu, F., & Yang, A. (2022). Comparative research on network intrusion detection methods based on machine learning. *Computers & Security*, *121*, 102861.
7. Fosić, I., Žagar, D., Grgić, K., & Križanović, V. (2023). Anomaly detection in NetFlow network traffic using supervised machine learning algorithms. *Journal of industrial information integration*, *33*, 100466.
8. Imanbayev, A., Tynymbayev, S., Odarchenko, R., Gnatyuk, S., Berdibayev, R., Baikenov, A., & Kaniyeva, N. (2022). Research of machine learning algorithms for the development of intrusion detection systems in 5G mobile networks and beyond. *Sensors*, *22*(24), 9957.
9. Azam, Z., Islam, M. M., & Huda, M. N. (2023). Comparative analysis of intrusion detection systems and machine learning based model analysis through decision tree. *IEEE Access*.
10. Belavagi, M. C., & Muniyal, B. (2016). Performance evaluation of supervised machine learning algorithms for intrusion detection. *Procedia Computer Science*, *89*, 117-123.
11. Hariharan, S., Rejimol Robinson, R. R., Prasad, R. R., Thomas, C., & Balakrishnan, N. (2023). XAI for intrusion detection system: comparing explanations based on global and local scope. *Journal of Computer Virology and Hacking Techniques*, *19*(2), 217-239.
12. Bace, R. G., & Mell, P. (2001). Intrusion detection systems.
13. Zhang, Y., & Lee, W. (2000, August). Intrusion detection in wireless ad-hoc networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking* (pp. 275-283).
14. Debar, H., Dacier, M., & Wespi, A. (1999). Towards a taxonomy of intrusion-detection systems. *Computer networks*, *31*(8), 805-822.
15. Axelsson, S. (1999, November). The base-rate fallacy and its implications for the difficulty of intrusion detection. In *Proceedings of the 6th ACM Conference on Computer and Communications Security* (pp. 1-7).
16. Javitz, H. S., & Valdes, A. (1991, May). The SRI IDES Statistical Anomaly Detector. In *IEEE Symposium on Security and Privacy* (pp. 316-326).
17. Kocher, G., & Kumar, G. (2021). Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges. Soft Computing, 25(15), 9731-9763.
18. Mohammadi, M., Rashid, T. A., Karim, S. H. T., Aldalwie, A. H. M., Tho, Q. T., Bidaki, M.,

... & Hosseinzadeh, M. (2021). A comprehensive survey and taxonomy of the SVM-based intrusion detection systems. *Journal of Network and Computer Applications*, *178*, 102983.

1. Sahingoz, O. K., Çebi, C. B., Bulut, F. S., Fırat, H., & Karataş, G. (2019). Saldırı Tespit Sistemlerinde Makine Öğrenmesi Modellerinin Karşılaştırılması. *Erzincan University Journal of Science and Technology*, *12*(3), 1513-1525.
2. Dhaliwal, S. S., Nahid, A. A., & Abbas, R. (2018). Effective intrusion detection system using XGBoost. *Information*, *9*(7), 149.
3. Chan, K. Y., Abu-Salih, B., Qaddoura, R., Ala’M, A. Z., Palade, V., Pham, D. S., ... & Muhammad, K. (2023). Deep neural networks in the cloud: Review, applications, challenges and research directions. *Neurocomputing*, *545*, 126327.
4. Bonifazi, G., Cauteruccio, F., Corradini, E., Marchetti, M., Terracina, G., Ursino, D., & Virgili, L. (2024). A model-agnostic, network theory-based framework for supporting XAI on classifiers. *Expert Systems with Applications*, *241*, 122588.
5. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135-1144).
6. Karampinis, I., Morfidis, K., & Iliadis, L. (2024). Derivation of Analytical Equations for the Fundamental Period of Framed Structures Using Machine Learning and SHAP Values. *Applied Sciences*, *14*(19), 9072.
7. Ribeiro, M. T., Singh, S., & Guestrin, C. (2018, April). Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32, No. 1).
8. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, *51*(5), 1-42.
9. Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision* (pp. 618-626).
10. Dhurandhar, A., Chen, P. Y., Luss, R., Tu, C. C., Ting, P., Shanmugam, K., & Das, P. (2018). Explanations based on the missing: Towards contrastive explanations with pertinent negatives. *Advances in neural information processing systems*, *31*.
11. Capuano, N., Fenza, G., Loia, V., & Stanzione, C. (2022). Explainable artificial intelligence in cybersecurity: A survey. *IEEE Access*, *10*, 93575-93600.
12. Kocher, G., & Kumar, G. (2021). Analysis of machine learning algorithms with feature selection for intrusion detection using UNSW-NB15 dataset. *Available at SSRN 3784406*.
13. Ahmad, M., Riaz, Q., Zeeshan, M., Tahir, H., Haider, S. A., & Khan, M. S. (2021). Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set. *EURASIP Journal on Wireless Communications and Networking*, *2021*, 1-23.
14. Disha, R. A., & Waheed, S. (2021, September). A Comparative study of machine learning models for Network Intrusion Detection System using UNSW-NB 15 dataset. In *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)* (pp. 1-5). IEEE.
15. Fathima, A., Khan, A., Uddin, M. F., Waris, M. M., Ahmad, S., Sanin, C., & Szczerbicki, E. (2023). Performance evaluation and comparative analysis of machine learning models on the

UNSW-NB15 dataset: a contemporary approach to cyber threat detection. *Cybernetics and Systems*, 1-17.

1. Hammad, M., El-Medany, W., & Ismail, Y. (2020, December). Intrusion detection system using feature selection with clustering and classification machine learning algorithms on the unsw-nb15 dataset. In *2020 international conference on innovation and intelligence for informatics, computing and technologies (3ICT)* (pp. 1-6). IEEE.
2. Abrar, I., Ayub, Z., Masoodi, F., & Bamhdi, A. M. (2020, September). A machine learning approach for intrusion detection system on NSL-KDD dataset. In *2020 international conference on smart electronics and communication (ICOSEC)* (pp. 919-924). IEEE.
3. Liu, J., Kantarci, B., & Adams, C. (2020, July). Machine learning-driven intrusion detection for Contiki-NG-based IoT networks exposed to NSL-KDD dataset. In *Proceedings of the 2nd ACM workshop on wireless security and machine learning* (pp. 25-30).
4. S. Y. Kayode, A. I. Abiodun, S. Y. Misra, M. K. Holone and R. Colomo-Palacios, “A machine learningbased intrusion detection for detecting internet of things network attacks,”Alexandria Engineering Journal, vol. 61, no. 12, pp. 9395–9409, 2022.
5. Rastogi, S., Shrotriya, A., Singh, M. K., & Potukuchi, R. V. (2022). An analysis of intrusion detection classification using supervised machine learning algorithms on nsl-kdd dataset. *Journal of Computing Research and Innovation*, *7*(1), 124-137.
6. Hong, R. F., Horng, S. C., & Lin, S. S. (2021, November). Machine learning in cyber security analytics using NSL-KDD Dataset. In *2021 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)* (pp. 260-265). IEEE.
7. Jafar, M. T., Al-Fawa'reh, M., Al-Hrahsheh, Z., & Jafar, S. T. (2021). Analysis and investigation of malicious DNS queries using CIRA-CIC-DoHBrw-2020 dataset. *Manchester Journal of Artificial Intelligence and Applied Sciences (MJAIAS)*.
8. Alzighaibi, A. R. (2023). Detection of DoH Traffic Tunnels Using Deep Learning for Encrypted Traffic Classification. *Computers*, *12*(3), 47.
9. Maseer, Z. K., Yusof, R., Bahaman, N., Mostafa, S. A., & Foozy, C. F. M. (2021). Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. *IEEE access*, *9*, 22351-22370.
10. Jairu, P., & Mailewa, A. B. (2022, May). Network anomaly uncovering on CICIDS-2017 dataset: a supervised artificial intelligence approach. In 2022 IEEE International Conference on Electro Information Technology (eIT) (pp. 606-615). IEEE.
11. Jabbar, A. F., & Mohammed, I. J. (2020, November). Development of an optimized botnet detection framework based on filters of features and machine learning classifiers using CICIDS2017 dataset. In *IOP Conference Series: Materials Science and Engineering* (Vol. 928, No. 3, p. 032027). IOP Publishing.
12. Rosay, A., Carlier, F., & Leroux, P. (2020). MLP4NIDS: An efficient MLP-Based network intrusion detection for CICIDS2017 dataset. In *Machine Learning for Networking: Second IFIP TC 6 International Conference, MLN 2019, Paris, France, December 3–5, 2019,* Revised Selected Papers 2 (pp. 240-254). Springer International Publishing.
13. Stiawan, D., Idris, M. Y. B., Bamhdi, A. M., & Budiarto, R. (2020). CICIDS-2017 dataset feature analysis with information gain for anomaly detection. *IEEE Access*, *8*, 132911-132921.
14. Korium, M. S., Saber, M., Beattie, A., Narayanan, A., Sahoo, S., & Nardelli, P. H. (2024). Intrusion detection system for cyberattacks in the Internet of Vehicles environment. Ad Hoc Networks, 153, 103330.
15. Kumar, A. G., Rastogi, A., & Ranga, V. (2024, May). Evaluation of Different Machine Learning Classifiers on New IoT Dataset CICIoT2023. In *2024 International Conference on Intelligent Systems for Cybersecurity (ISCS)* (pp. 1-6). IEEE.
16. Chhikara, R., & Kashyap, N. (2024, May). A Comparative Analysis of Machine Learning Prediction Algorithms for Detecting IoT Botnet Activities. In *2024 International Conference on Intelligent Systems for Cybersecurity (ISCS)* (pp. 1-6). IEEE.
17. Gheni, H. Q., & Al-Yaseen, W. L. (2024). Two-step data clustering for improved intrusion detection system using CICIoT2023 dataset. *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, *9*, 100673.
18. Kaur, N., & Gupta, L. (2024, June). Enhancing IoT Security in 6G Environment With Transparent AI: Leveraging XGBoost, SHAP and LIME. In *2024 IEEE 10th International Conference on Network Softwarization (NetSoft)* (pp. 180-184).
19. Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions.

*Advances in neural information processing systems*, *30*.