future internet

MDPI

Article

XI2S-IDS: An Explainable Intelligent 2-Stage Intrusion Detection System

Maiada M. Mahmoud [1], Yasser Omar Youssef [2,*] and Ayman A. Abdel-Hamid [3]

[1] College of Computing and Information Technology, Arab Academy for Science, Technology, and Maritime Transport, Cairo P.O. Box 2033, Egypt; maiada@aast.edu
[2] School of Library and Information Studies, University of Oklahoma, Norman, OK 73019, USA
[3] College of Computing and Information Technology, Arab Academy for Science, Technology, and Maritime Transport, Alexandria P.O. Box 1029, Egypt; hamid@aast.edu
* Correspondence: yyoussef@ou.edu

Abstract: The rapid evolution of technologies such as the Internet of Things (IoT), 5G, and cloud computing has exponentially increased the complexity of cyber attacks. Modern Intrusion Detection Systems (IDSs) must be capable of identifying not only frequent, well-known attacks but also low-frequency, subtle intrusions that are often missed by traditional systems. The challenge is further compounded by the fact that most IDS rely on black-box machine learning (ML) and deep learning (DL) models, making it difficult for security teams to interpret their decisions. This lack of transparency is particularly problematic in environments where quick and informed responses are crucial. To address these challenges, we introduce the XI2S-IDS framework—an Explainable, Intelligent 2-Stage Intrusion Detection System. The XI2S-IDS framework uniquely combines a two-stage approach with SHAP-based explanations, offering improved detection and interpretability for low-frequency attacks. Binary classification is conducted in the first stage followed by multi-class classification in the second stage. By leveraging SHAP values, XI2S-IDS enhances transparency in decision-making, allowing security analysts to gain clear insights into feature importance and the model's rationale. Experiments conducted on the UNSW-NB15 and CICIDS2017 datasets demonstrate significant improvements in detection performance, with a notable reduction in false negative rates for low-frequency attacks, while maintaining high precision, recall, and F1-scores.

Keywords: IDS; XAI; SHAP; LSTM; UNSW-NB15; CICIDS2017; deep learning

Academic Editor: Massimo Cafaro

Received: 23 November 2024
Revised: 26 December 2024
Accepted: 4 January 2025
Published: 8 January 2025

Citation: Mahmoud, M.M.; Youssef, Y.O.; Abdel-Hamid, A.A. XI2S-IDS: An Explainable Intelligent 2-Stage Intrusion Detection System. *Future Internet* **2025**, *17*, 25. https://doi.org/10.3390/fi17010025

1. Introduction

In recent years, connected technologies such as the Internet of Things (IoT), 5G communication, and smart grids have rapidly developed. These advancements have fundamentally transformed how industries and individuals interact with the digital world. This expansion has resulted in an exponential increase in the number of connected devices and the complexity of communication networks, providing significant benefits but also amplifying the scope of security risks [1–3]. As cyber attacks continue to rise globally, impacting sectors from finance to healthcare, addressing these threats has become a critical concern for researchers and security professionals. Ensuring robust cybersecurity is essential not only for protecting sensitive data but also for maintaining trust in these advancing technologies [4].

As of 2024, a total of 1,187,441,064 malware samples have been identified, with more than 9 million new samples emerging each month, according to the AV-TEST daily report [5]. The institute registers over 300,000 new malware samples daily. Over the past five years,

malware activity has risen significantly, increasing from 647,920,244 samples in 2019 to over one billion by September 2024. Figure 1 illustrates the annual growth in malware from 2008 to 2024.

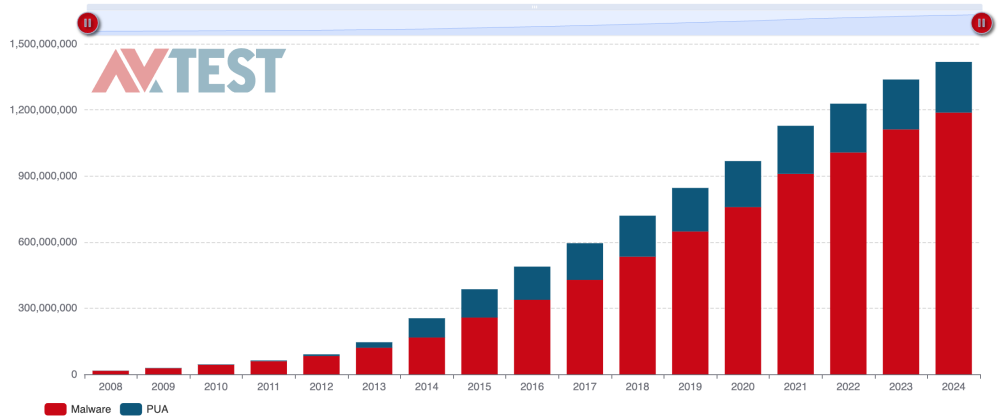

Figure 1. Amount of malware from 2008 to August 2024 [5].

Intrusion Detection Systems (IDSs) play a central role in safeguarding networks and computer systems from such threats [6]. Traditionally, IDS have been effective in monitoring network traffic and host activities to identify potential attacks [7]. However, the increasing sophistication of cyber attacks, particularly zero-day exploits and low-frequency attacks, presents new challenges that conventional IDS struggle to address. Recent research has focused on leveraging machine learning (ML) and deep learning (DL) techniques to enhance IDS capabilities [8–11]. These approaches offer the potential to detect threats with greater accuracy and scalability than traditional methods, revolutionizing the field of cybersecurity.

DL models have shown promising results in IDS for network traffic classification and their accuracy in detecting cyber attacks and potential threats has been thoroughly demonstrated in previous research [1,12–14]. Some of the most effective DL models are Long Short-Term Memory (LSTM): a special Recurrent Neural Network (RNN) efficient in the processing of time-series sequential data with long-term dependencies [15,16], Convolutional Neural Network (CNN): effective in capturing spatial relationships and particularly useful for extracting features from raw data [16,17], and AutoEncoder (AE): learns the underlying structure of input data and is used for feature extraction and anomaly detection in IDS applications [18].

Aminanto and Kim [19] used Artificial Neural Networks (ANNs) and stacked autoencoders (SAEs) to detect impersonation attacks, achieving an 85% detection rate with a 2.36% false alarm rate on the AWID dataset. Later, they improved performance to a 92% detection rate with a 4.4% false alarm rate by incorporating two cascading encoders with k-means clustering [8]. Shone et al. [10] proposed an unsupervised Non-symmetric Deep AutoEncoder (NDAE) combined with Random Forest (RF) for feature extraction and classification, achieving 97.9% accuracy on KDD Cup99 and 85.4% on NSL-KDD datasets, though with poor results for U2R and R2L attacks that did not surpass 9.5%. Yan and Han's Stacked Sparse AutoEncoder (SSAE) paired with SVM achieved a 99.3% accuracy and improved detection rates for low-frequency attacks on NSL-KDD [20]. Vinayakumar et al. [21] proposed a scalable deep neural network (DNN) framework that outperformed traditional machine learning models for intrusion detection across multiple datasets, with up to 91% accuracy on ADFA-Linux.

Liu et al. [22] developed a model combining neural networks and autoencoders for network and host-based intrusion detection, achieving 99% accuracy in multi-class classification on NSL-KDD and 92% for host-based intrusion detection (HIDS) on ADFA-

LD. Yang et al. [23] used variational autoencoders (VAEs) and a generative adversarial network (WGAN-GP) to augment unbalanced datasets, increasing detection rates for low-frequency attacks. Yu and Bian [24] applied few-shot learning using DNN and CNN, achieving strong results for unbalanced datasets with 92% accuracy on UNSW-NB15. SwiftIDS [25] emphasized real-time intrusion detection using LightGBM [26] for parallel data processing, while Kanna and Santhi [27] combined optimized CNN and hierarchical multi-scale LSTM [28] achieving up to 96% accuracy on various datasets.

Psychogyios et al. [29] shifted the perspective of IDS from reactive to proactive by redefining the UNSW-NB15 dataset into a time-series problem. The authors utilized CNN, LSTM, and attention models to forecast possible attacks and were able to achieve 85% F1-score, 90% precision, and 81% recall. Korium et al. [30] proposed an ML-based IDS tailored for cyber attacks in the Internet of Vehicles (IoV). Their approach used the datasets CIC-IDS-2017, CSE-CIC-IDS-2018, and CIC-DDoS-2019 to create a unified framework capable of detecting multiple attack types. The system integrates feature selection using Random Forest Regression, data balancing techniques like SMOTE and SMOTE-ENN, and ML models, including Random Forest, XGBoost, and CatBoost, achieving 99.8% accuracy with minimized overfitting and a detection time of 0.24 s.

Despite these advancements, several challenges remain. Key areas of concern include high false positive and false negative rates, scalability for large and complex networks, and the ability to adapt to zero-day attacks. The ability to detect abnormal patterns in real-world scenarios is essential for enabling prompt responses, regardless of the nature of the attack. Moreover, a key challenge in improving IDS performance is the issue of black box models. While DL models are highly effective, their lack of transparency raises concerns, especially when used in critical infrastructure. The need for explainability is increasingly important to ensure that security measures are not only accurate but also understandable, allowing for better trust and control in deployment. Balancing the need for accurate detection with explainability remains a significant challenge.

The complexity and lack of transparency in ML and DL models have given rise to a research field called eXplainable Artificial Intelligence (XAI). Its goal is to enhance trust in black-box model decisions by providing clear explanations of their logic and reasoning [31]. Explainability is particularly critical in IDS deployed in high-stakes, real-time environments such as healthcare and finance, where rapid, transparent decision-making is essential. Unlike traditional black-box models, explainable IDS frameworks allow security analysts to trust and understand the logic behind each classification, which is indispensable for taking informed, timely actions. This demand for interpretability aligns with a broader push in cybersecurity for XAI methodologies, positioning explainable IDS as pivotal for organizations that rely on prompt responses to potential threats.

XAI explanations can be global, offering a broad overview of how the model functions, or local, focusing on specific predictions. Additionally, explanation techniques are categorized as model-specific, designed for a particular learning model, or model-agnostic, which can be applied to any model regardless of its implementation. In [32], Dias et al. implemented a hybrid IDS approach by combining expert-written rules with machine learning algorithms to detect and classify network attacks. The model, based on a microservices architecture, includes an anomaly detector and a dynamic rule generator, which updates its knowledge base through Decision Trees. The model was validated in a DoS attack case study. In [33], Dong et al. introduced an interpretable, privacy-preserving NIDS called FEDFOREST, utilizing Federated Learning for data privacy and Gradient Boosting Decision Trees for classification. Tested on four datasets, FEDFOREST achieved accuracy scores between 67% and 89% and demonstrated the ability to detect unknown attacks by classifying them into similar categories.

Kumar and Ansari [34] introduced a lightweight, explainable IDS for Software-Defined Internet of Things (SD-IoT) environments, using the Sheep Flock Optimization Algorithm-Least Absolute Shrinkage and Selection Operator (SFOA-LASSO) for feature selection to enhance model efficiency. Their system, evaluated on SD-IoT and CIC-IoT-2023 datasets, employs machine learning algorithms like Decision Tree, Random Forest, XGBoost, and MLP, with SHAP used for interpretability. Similarly, Hooshmand et al. [35] developed a robust anomaly detection system addressing class imbalance using a hybrid sampling technique combining SMOTE and K-means undersampling with XGBoost (SKM-XGB). Evaluated on NSL-KDD and UNSW-NB15 datasets, the system uses SHAP for explanations and achieves high detection accuracy, particularly excelling in imbalanced network traffic data with over 99% accuracy in binary classification. Shtayat et al. [36] proposed a framework that utilizes an ensemble model of 3 CNN classifiers with both LIME and SHAP explanations. Using the ToN-IoT dataset for evaluation, the study demonstrates that ensemble learning effectively enhances both the accuracy and interpretability of DL-based IDSs in binary and multi-class classification tasks.

In this paper, we present an eXplainable Intelligent 2-Stage Intrusion Detection System (XI2S-IDS) that combines the power of ML and DL with an emphasis on explainability. XI2S-IDS addresses the aforementioned challenges by employing a two-stage detection process: an initial binary classification model to distinguish between normal and abnormal behavior, followed by a multi-class model to categorize different types of attacks. Integrating XAI into the initial stage enables security analysts to understand why certain traffic is classified as anomalous, while the second stage is tailored to detecting low-frequency attack types. The framework seeks to not only improve detection accuracy but also provide insights into the decision-making process of the system.

Detecting abnormal patterns in real-world scenarios is crucial not only for immediate response but also for ensuring that the system is robust across various attack types. This necessity reinforces the significance of the proposed two-stage approach: the binary classifier ensures immediate anomaly detection, while the second stage focuses on identifying the specific attack type, facilitating deeper analysis. Such a design supports the integration of XAI to provide meaningful, actionable insights during the first stage, aiding in the decision-making process of system experts.

The contributions of this paper can be summarized as follows: First, XI2S-IDS, a two-stage framework for intrusion detection, is introduced. It leverages both binary and multi-class classification to improve detection accuracy, particularly for low-frequency attacks. Second, by incorporating SHAP-based explanations, the proposed system enhances interpretability, enabling security professionals to better understand and trust the model's decisions. The framework is evaluated on two benchmark datasets—UNSW-NB15 and CICIDS2017—and demonstrates significant improvements in both detection rates and interpretability over traditional single-stage and black-box models.

The rest of this paper is organized as follows: Section 2 describes the materials and methods used, including details on the architecture, preprocessing steps, and model design of the XI2S-IDS framework. Section 3 presents the results of the experiments, providing a comparative analysis of the proposed system's performance across key evaluation metrics. Section 4 discusses the implications of our findings and situates them within the context of existing research. Finally, Section 5 concludes the paper with a summary of our contributions and potential directions for future research in enhancing IDS accuracy and explainability.

2. Materials and Methods

XI2S-IDS's architecture leverages the normal and attack class imbalance that is present in the dataset. It comprises two stages powered by XAI: the first stage classifies each record as either normal or attack, providing a global explanation of the trained model, while the second stage focuses solely on classifying the attack type. The first stage utilizes a binary ML classifier to distinguish between normal packets (labeled 0) and abnormal (attack) packets (labeled 1). The second part is a DL multi-class classifier that only trains on attack records and identifies their specific types, disregarding all normal records during training. Figure 2 illustrates the architecture of XI2S-IDS.

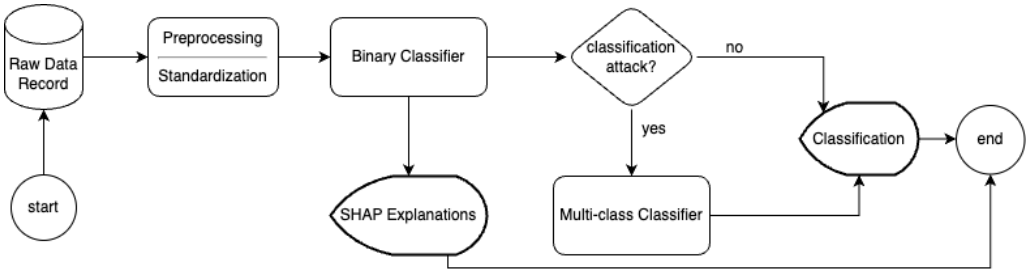

Figure 2. General architecture of XI2S-IDS.

Nearly all available IDS datasets are imbalanced [37], with the majority consisting of normal behavior records, while the remaining portion includes various attack types. Some attack types appear at a lower frequency, referred to as low-frequency attacks, as they are rarely captured in the datasets. These low-frequency attacks present a significant challenge for existing IDSs, as they are more difficult to detect.

In some of the related research, authors focus solely on the binary classification of dataset records, identifying normal and attack records [38–40]. Such studies achieve high accuracies; however, they conceal the challenge of identifying low-frequency attacks as it is out of scope for binary classification. In other research, authors attempted to tackle the issue of low-frequency records using multi-class classifiers; however, their results leave room for improvement.

The two stages of the XI2S-IDS framework combine the advantages of binary classification, which distinguishes between normal and abnormal behavior, with a multi-class classifier that specifies the type of attack. By separating these tasks into distinct stages, the architecture allows the binary classifier to handle the normal vs. abnormal classification, while the multi-class classifier focuses exclusively on distinguishing between different attack types.

Notably, in Stage II, the multi-class classifier is trained only on attack records, excluding normal records from the training dataset. This step is designed to help the multi-class model concentrate entirely on learning the distinct characteristics of various attack classes, which is particularly useful for detecting low-frequency or rare attacks that are often underrepresented in the dataset. By removing normal records, the model's ability to differentiate between attack classes is improved, as it no longer needs to balance the detection of normal traffic, which dominates the datasets due to their imbalanced nature. This targeted focus on attack instances refines the learning process, leading to a more specialized model for attack detection.

To enhance the interpretability of the model, global SHAP explanations are provided for the trained binary model in Stage I. SHAP offers a detailed, transparent view of the model's decision-making process, revealing how each feature contributes to predictions. This approach not only improves the model's transparency but also provides a deeper

understanding of the factors influencing the binary classification between normal and abnormal traffic.

This research hypothesizes that the exclusion of normal records from the multi-class classifier's training data enhances the overall performance of the IDS. By refining the focus on attack detection, the model becomes more adept at recognizing the nuances of various attack types, improving detection accuracy, especially for low-frequency attacks. The two-stage approach, therefore, optimizes the trade-off between specialized attack detection and overall system performance.

2.1. Evaluation Metrics

Evaluation of machine learning models often considers various metrics, with accuracy being one of the most commonly used. While accuracy provides a general measure of model performance, it can be misleading in the context of intrusion detection, especially when low-frequency attacks are present. For IDS evaluation, accuracy alone is insufficient, as it does not account for the critical balance between false positives (incorrectly flagged benign traffic) and false negatives (missed attacks). A model with high accuracy may still yield an unacceptable number of false negatives, which pose severe security risks. Conversely, a model with lower accuracy but fewer false negatives could be more desirable if it reliably detects critical threats. Moreover, in imbalanced datasets typical of IDS applications, accuracy can obscure true performance, as it often overlooks minority classes crucial for threat detection [41].

Therefore, in this study, we prioritize false negative rate (FNR) as it captures the percentage of missed attacks, along with precision, recall, F1-score, and accuracy, to offer a comprehensive performance assessment. These metrics are essential for understanding both detection capability and error balance in IDS, particularly under imbalanced data conditions. The formulas for these metrics are provided in Equations (1)–(5).

$$FNR = \frac{FN}{FN + TP} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1\text{-}score = 2\frac{Precision * Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \tag{4}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

2.2. Environment

The environment in which all experiments were conducted includes an Intel i7 processor as the Central Processing Unit (CPU). It has a Random Access Memory (RAM) capacity of 16 GB and features an "Nvidia RTX 2080 TI" Graphics Processing Unit (GPU). The system runs on Python 3.7.13 and uses SHAP for explanations, TensorFlow 1.13.2 for DL models as well as scikit-learn 1.0.2 for ML models.

2.3. Dataset

Two datasets were chosen as benchmarks for the evaluation of the framework: UNSW-NB15 and CICIDS2017. The CICIDS2017 dataset [42] comprises 2,273,097 normal records and 557,646 attack records distributed over 14 attack classes. It was collected over 5 days.

The UNSW-NB15 dataset [43–47] dataset consists of both real normal behavior and imitated attack behaviors. It has 49 feature and 9 attack classes. It contains 2,540,047 records, of which approximately 87% are normal records and approximately 13% are attack records. Creators of the dataset also provided 10% of the original dataset as a subset of the dataset, containing a total of 257,673 records, split between train and test sets. In this research, we have opted to use the original complete dataset that contains 2,540,047 records.

The percentage of records of each class in both datasets as well as a description of the attack classes are presented in Tables 1 and 2 for UNSW-NB15 and CICIDS2017 datasets, respectively, highlighting the high imbalance within the datasets.

Table 1. Types of attacks in the UNSW-NB15 dataset with descriptions and frequencies.

Attack Type	Description	Frequency (%)
Normal	Normal traffic records	87.3%
Fuzzers	Attacks that send a large volume of random and invalid traffic to the target system causing unexpected behavior	8.4%
Analysis	Attacks that use port scanning or malicious HTML penetration scripts to breach web applications	1.7%
Backdoor	Attacker gains access to a system and leaves a secret entry point for future access	0.9%
Denial of Service (DoS)	Attacker overwhelms the targeted network or server by flooding it with excessive data or requests to disrupt its normal functioning rendering legitimate users unable to access its services	0.6%
Exploits	Intrusions that take advantage of vulnerabilities or bugs within an operating system or software known by the attackers	0.5%
Generics	Attempts made to break the key or cryptographic system of a security system	0.1%
Reconnaissance (Reconn)	An attack that involves collecting information about the targeted computer network with the intention of circumventing its security controls	0.09%
Shellcode (SC)	Attack that launches a command shell under the attacker's control to evade security controls of a computer network	0.05%
Worms	Malicious programs that self-replicate and spread to other computer systems	0.006%

Table 2. Types of attacks in the CICIDS2017 dataset with descriptions and frequencies.

Attack Type	Description	Frequency (%)
Benign	Normal traffic	80.3
DDoS	Distributed Denial of Service attack, overwhelming a network by flooding it with traffic	4.52
PortScan	Scanning multiple ports on a host to find vulnerabilities	5.61
Bot	A compromised device controlled by an attacker, often part of a botnet	0.07
FTP-Patator	Repeated attempts to crack FTP credentials using different combinations of username and password	0.28
SSH-Patator	Repeated attempts to crack SSH credentials	0.21

Table 2. *Cont.*

Attack Type	Description	Frequency (%)
DoS—Slowloris	Denial of Service attack where partial HTTP requests are sent to exhaust server resources	0.20
DoS—slowhttptest	Denial of Service attack where partial HTTP requests are sent to exhaust server resources	0.19
Infiltration	Exploiting a vulnerable software to create a backdoor for later access	0.0013
DoS—Hulk	Denial of Service attack using the Hulk tool to generate large volumes of web traffic	8.16
DoS—GoldenEye	Denial of Service attack using the GoldenEye tool to overwhelm web servers	0.36
Heartbleed	Exploits a vulnerability in the Heartbeat feature of OpenSSL to extract sensitive information	0.0004
Web Attack—SQLi	SQL injection attacks, targeting databases by injecting malicious SQL queries	0.0007
Web Attack—XSS	Cross-Site Scripting, where malicious scripts are injected into web pages viewed by other users	0.02
Web Attack—Brute	Repeated attempts to guess web login credentials	0.05

2.4. Detailed Framework Workflow

The XI2S-IDS framework operates in three key phases, which are illustrated in Figure 3. First, the system preprocesses the data to ensure consistency and feature readiness. Second, a binary classifier is applied to differentiate between normal traffic and potential attacks. Finally, a deep learning-based multi-class classifier is used to further categorize detected attack traffic into specific types. Each phase is carefully constructed to maximize detection accuracy while balancing interpretability and computational efficiency.

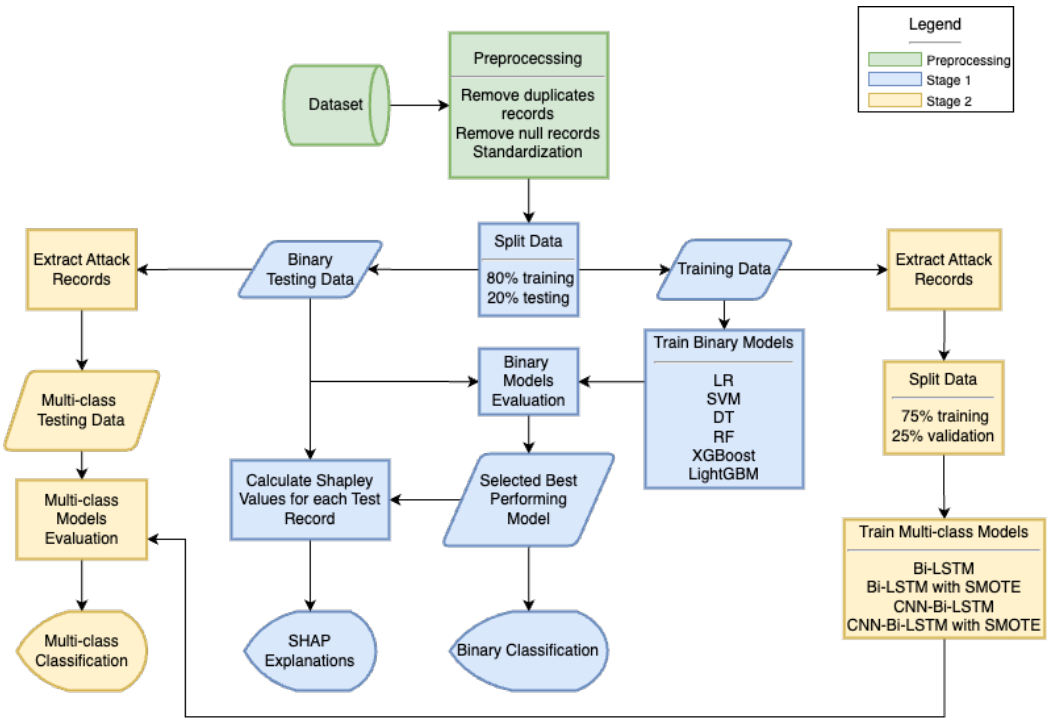

Figure 3. Detailed architecture of the 2 stages.

In the data preprocessing phase, raw network traffic data is cleaned to remove address features and null records. This step is crucial in IDS, as irrelevant attributes can dilute the

model's ability to detect attacks and unnecessarily increase computational overhead. A new feature column is introduced to represent the total number of bytes transferred in each record, as this information can be a strong indicator of anomalous behavior associated with certain attack types such as DoS attacks.

Next, data standardization is performed using a Standard Scaler to normalize numerical features, ensuring that features with large value ranges do not dominate the learning process. This is particularly important in IDS scenarios, where network traffic data often contains a mix of high-variance and low-variance attributes. Categorical features, such as protocol types and network services, are encoded using OneHotEncoding.

The dataset is split into training and testing sets with an 80/20 split for the binary classification stage. Additionally, a 75/25 split within the training data is used to create a validation set for the multi-class classifier, enabling hyperparameter tuning and early stopping to prevent overfitting. A third version of the dataset is also generated by removing all normal records, allowing the multi-class classifier to train exclusively on attack instances. This targeted approach ensures the model focuses on learning the distinct characteristics of various attack types, which is especially beneficial for improving detection rates of low-frequency attacks.

2.4.1. Stage I: Explainable Binary Classifier

The first stage of the framework is the binary classifier phase. A diverse selection of eight ML models, each with unique capabilities, was trained on the datasets to ensure a comprehensive exploration of potential solutions for intrusion detection. The models included variations of Stochastic Gradient Descent (SGD), Linear SVM [48], Decision Trees (DTs) [48], RF [49], XGBoost, and LightGBM. Hyperparameter tuning was performed using Optuna, a framework for automated optimization, to maximize detection effectiveness while minimizing false alarms. Table 3 details the range of parameters optimized for each model.

Table 3. Hyperparameters for ML binary classification.

Model	Parameters
LinearSVC	Loss: hinge; Regularization C: 5
SGD (log, hinge, mod. huber)	Penalty: l1, l2; Alpha: powers of 10 [(−5)–2]
Decision Tree	Max depth: [8–12], Min samples split: [4–6] Min samples leaf: [9–13]
Random Forest	Estimators: [200–400]; Max depth: [20–24]; Min samples split: [2–6]; Criterion: gini, entropy
XGBoost	Learning rate: [0.001–0.1]; Max depth: [4–12]; Colsample bylevel: [0.1–1]; Subsample: [0.1–0.7]; Estimators: [200–400]
LightGBM	Boosting type: gbdt, dart, rf; Num leaves: [2–256]; Feature fraction: [0.4–1]; Bagging fraction: [0.4–1]; Bagging freq: [1–7]; Min child samples: [5–100]; L1: [(1×10^{-8})–10];

SHAP [50] is an explanation technique built upon the concept of Shapley values in game theory that calculates the contribution of each feature to the final output of the model. A significant advantage of SHAP lies in its versatility, as it can be applied across various models and classifiers. The framework combines Shapley values with local interpretations of the model to set an importance score for the dataset's features. SHAP's explanation for a particular instance is derived through Equation (6) as follows:

$$g(s) = v_0 + \sum_{i=1}^{N} v_i s_i \tag{6}$$

where g represents the explanation model, s represents the simplified features, and v_i signifies the Shapley value corresponding to feature i. N denotes the maximum size of the feature vector. Here, s belongs to the set $\{0,1\}^N$, where the presence of '1' in s indicates that the features in the new dataset align with those in the original data, while '0' suggests a disparity between the features in the new and original datasets.

The most important features can then be selected based on Equation (7) as follows:

$$IF_j = \sum_{i=1}^{n} |v_j(x_i)| \tag{7}$$

where IF_j is the average absolute Shapley value for feature j and n represents the total count of records in the dataset.

2.4.2. Stage II: Attack Classifier

The second stage of the framework focuses on multi-class classification. To enhance the models' ability to distinguish between various attack behaviors, an attacks-only dataset was created by removing all normal records from the dataset. This step allowed the models to concentrate solely on attack patterns. Figure 4 illustrates the general workflow of the multi-class classification phase.

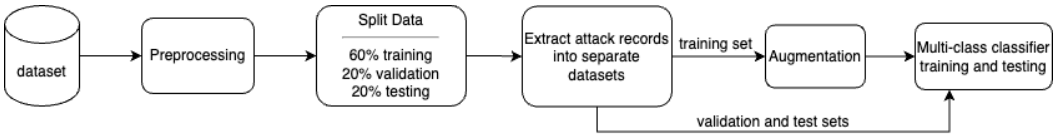

Figure 4. General workflow of the multi-class classifier phase.

The CICIDS2017 and UNSW-NB15 datasets exhibit significant class imbalances, with rare attack types like Heartbleed and Worms severely underrepresented. To address this, the Synthetic Minority Oversampling Technique (SMOTE) was applied to oversample these low-frequency classes. This augmentation ensures that the multi-class classifier effectively learns the patterns of rare attacks without being dominated by more frequent classes.

Two deep learning (DL) models were developed and tested exclusively on the attack data. The first model incorporated Bidirectional Long Short-Term Memory (BI-LSTM) layers. The second model combined BI-LSTM and Convolutional Neural Network (CNN) layers, drawing inspiration from a previous study [15].

The BI-LSTM model was chosen due to its demonstrated superior performance in intrusion detection systems (IDSs), as evidenced in recent studies [51,52]. It consists of two BI-LSTM layers, separated by a max-pooling layer. Critical feature extraction occurs in the multi-class classifier, where the max-pooling layer after BI-LSTM layers encodes the most important temporal features, enabling the model to focus on the characteristics that best distinguish between different attack types.

Additionally, both models were tested with and without the use of SMOTE upsampling on the attacks-only dataset. The final results were compared to identify the most effective model for multi-class classification of IDS, across both the UNSW-NB15 and CICIDS2017 datasets.

3. Results

3.1. Stage I: Binary Classifier Results

The models were evaluated on several performance metrics, including accuracy, precision, recall, F1-score, FNR, and False Positive Rate (FPR). These metrics served as indicators of each model's ability to correctly classify instances of both normal and intrusive network traffic.

The objective of the binary model is to detect the highest number of attack records while achieving a low FNR and maintaining a high F1-score and overall performance. Multiple classifiers were trained with a range of parameters, and the best parameters were selected for each classifier. The classifier with the best performance was chosen as the model for the binary classification stage of XI2S-IDS. Tables 4 and 5 show the results of the eight ML models on both CICIDS2017 and UNSW-NB15 datasets.

Table 4. Performance evaluation on the CICIDS2017 dataset.

Model	Accuracy	Precision	Recall	F1-Score	FNR	FPR
LinearSVC	0.9717	0.9754	0.8339	0.8991	0.0288	0.0246
SGD (log)	0.9766	0.9513	0.8908	0.9201	0.0192	0.0486
SGD (hinge)	0.9762	0.9480	0.8914	0.9188	0.0191	0.0520
SGD (mod. huber)	0.9685	0.9785	0.8089	0.8857	0.0329	0.0214
Decision Tree	0.9975	0.9957	0.9877	0.9917	0.0021	0.0042
Random Forest	0.9988	0.9983	0.9939	0.9961	0.00107	**0.0016**
XGBoost	0.9992	**0.9995**	**0.9995**	**0.9995**	0.0005	0.0023
LightGBM	**0.9993**	0.9979	0.9974	0.9977	**0.0004**	0.0025

Bold values indicate the best result across all models for each evaluation criterion.

Table 5. Performance evaluation on the UNSW-NB15 dataset.

Model	Accuracy	Precision	Recall	F1-Score	FNR	FPR
LinearSVC	0.9875	0.9129	**0.9967**	0.9530	**0.003**	0.013
SGD (log)	0.9857	0.9298	0.9601	0.9447	0.039	0.010
SGD (hinge)	0.9868	0.9183	0.9836	0.9499	0.016	0.012
SGD (mod. huber)	0.9848	0.9268	0.9554	0.9409	0.044	0.010
Decision Tree	0.9894	0.9910	0.9248	0.9568	0.075	**0.001**
Random Forest	0.9876	**0.9918**	0.9099	0.9491	0.090	**0.001**
XGBoost	0.9880	0.9888	0.9160	0.9510	0.083	**0.001**
LightGBM	**0.9952**	0.9821	0.9803	**0.9812**	0.019	0.002

Bold values indicate the best result across all models for each evaluation criterion.

According to the reported results, XGBoost slightly outperforms LightGBM on the CICIDS2017 dataset in terms of FPR (0.0023 vs. 0.0025), but LightGBM has a marginally better FNR (0.0004 vs. 0.0005). On the UNSW-NB15 dataset, LightGBM performs significantly better in terms of recall (0.9803 vs. 0.9160) and FNR (0.019 vs. 0.083), meaning it is better at detecting more attacks. XGBoost, however, has a slightly lower FPR (0.001 vs. 0.002). The remaining metrics are similar between the two ensemble models.

LightGBM is known for its speed and memory efficiency due to its use of histogram-based learning and leaf-wise growth strategy compared to XGBoost which can be slower, especially as datasets grow larger. Consequently, LightGBM is chosen for the remainder of the experiments. The confusion matrix of LightGBM on both datasets is shown in Figures 5 and 6.

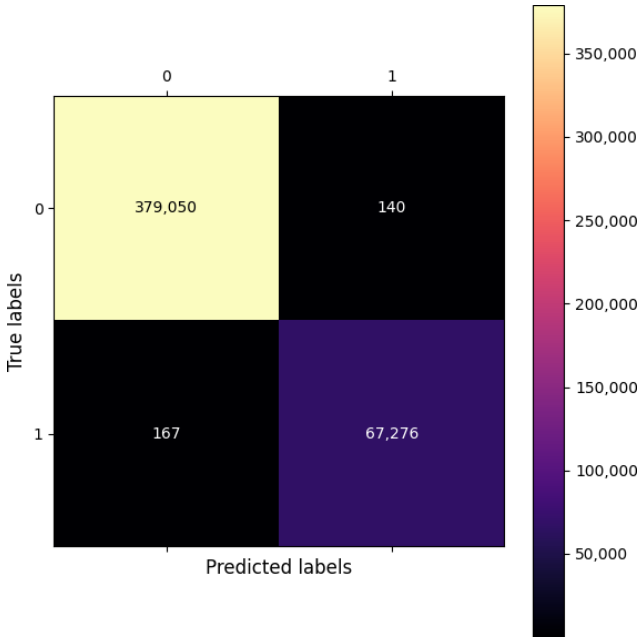

Figure 5. LightGBM confusion matrix on the CICIDS2017 dataset.

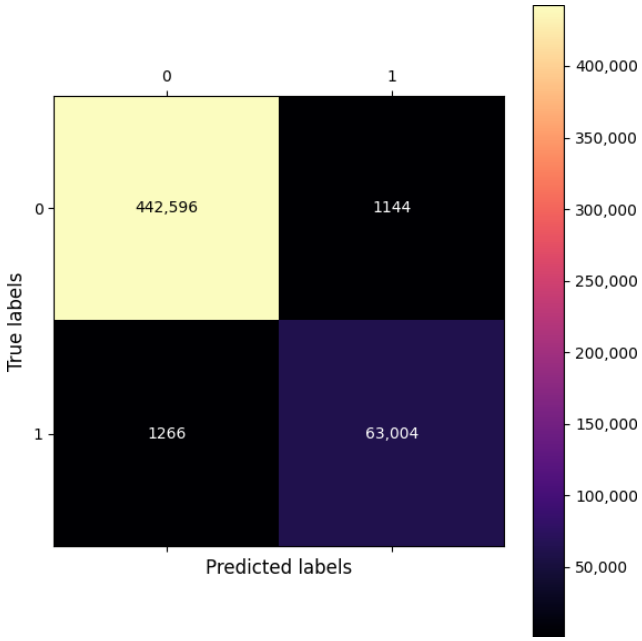

Figure 6. LightGBM confusion matrix on the UNSW-NB15 dataset.

For this experiment, SHAP explanations are generated for the chosen binary classifier of experiment I, LightGBM, to help in understanding the reasoning behind classifying the records as normal or attack. Figures 7 and 8 present the top 20 important features according to the model, while Figures 9 and 10 show a detailed description of the effect of each feature on the model for CICIDS2017 and UNSW-NB15 datasets, respectively.

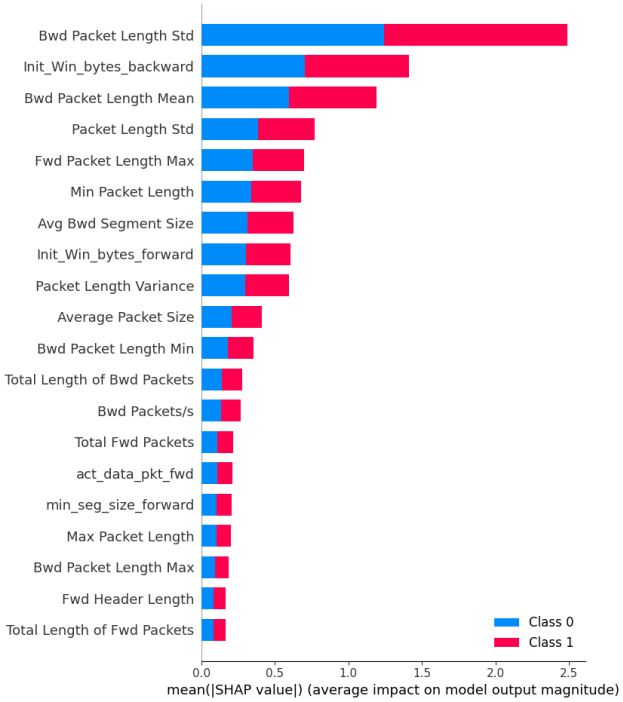

Figure 7. Top 20 features selected by LightGBM on the CICIDS2017 dataset.

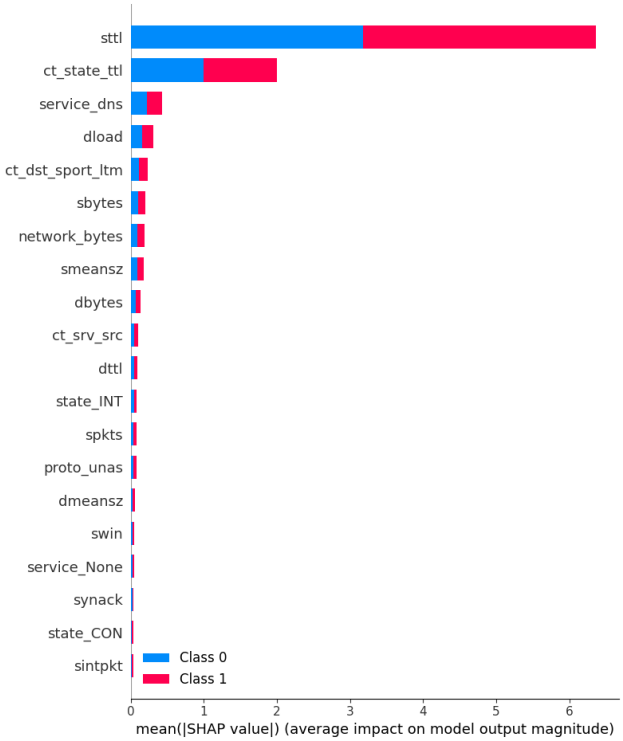

Figure 8. Top 20 features selected by LightGBM on the UNSW-NB15 dataset.

In Figures 9 and 10, the SHAP values along the x-axis show the contribution of each feature to the model's output. Positive SHAP values indicate that the feature increases the prediction, possibly predicting an intrusion, while negative SHAP values decrease the prediction, possibly predicting normal traffic. The color represents the value of the feature. Red corresponds to higher feature values, while blue represents lower feature values.

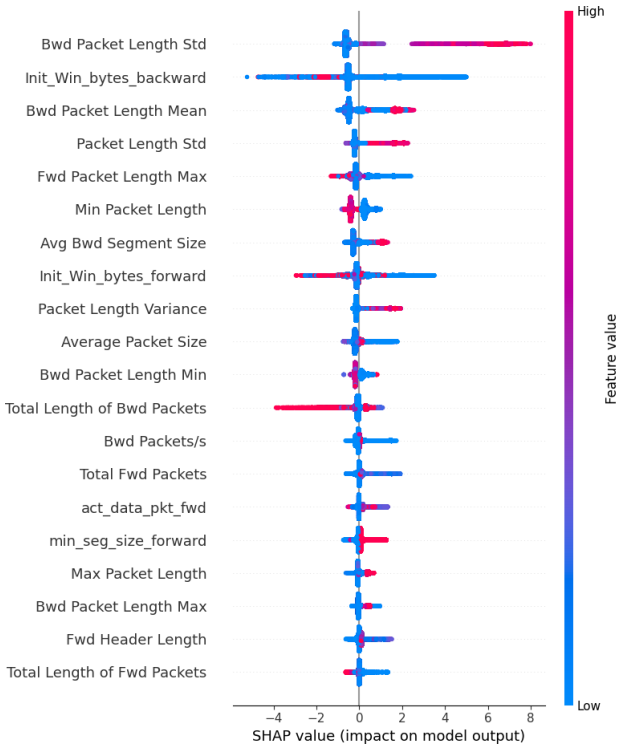

Figure 9. SHAP plot analysis on the CICIDS2017 dataset.

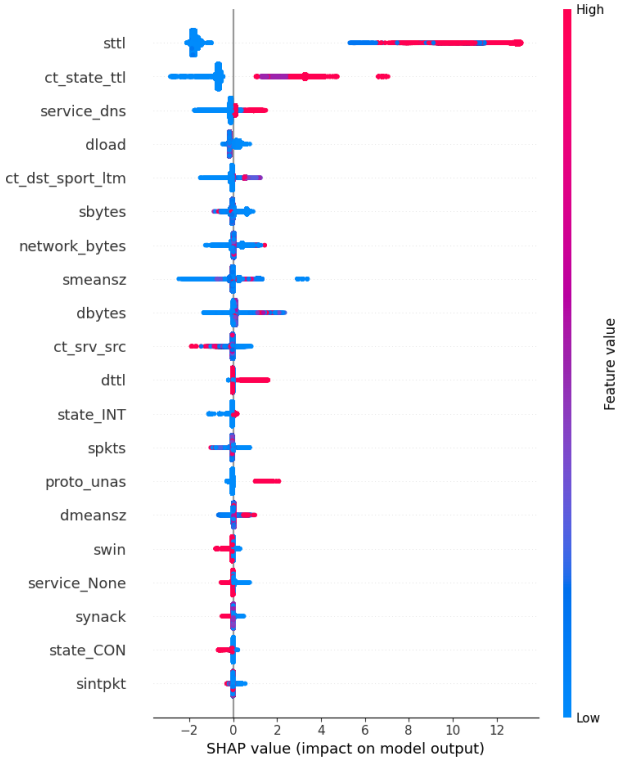

Figure 10. SHAP plot analysis on the UNSW-NB15 dataset.

The SHAP plot of CICIDS2017 (Figure 9) effectively shows how different network features contribute to the model's decision-making in intrusion detection. High variance in packet lengths and unusual flow characteristics are prominent indicators of potentially malicious activity in the CICIDS2017 dataset. These insights align with typical network in-

trusion patterns where attackers exploit packet size anomalies and irregular flow properties to bypass traditional defenses.

Packet lengths and variations, features Packet Length Std, Min Packet Length, Bwd Packet Length Std, and Bwd Packet Length Mean, suggest that anomalies in packet sizes are strong indicators of malicious traffic. Attackers often use packets of unusual sizes or highly variable packet lengths to evade detection or overwhelm the network.

Flow features, such as Init_Win_bytes_backward and Init_Win_bytes_forward, represent TCP connection properties that may hint at unusual or poorly structured traffic, often linked with scanning or DOS-type attacks.

Packet rates features, such as Bwd Packets/s and act_data_pkt_fwd, suggest that the rate of packet transmission and the data in forward packets are significant. High packet transmission rates may indicate attempts to flood a target, whereas low data volumes in forwarded packets might indicate reconnaissance activities.

The SHAP plot of UNSW-NB15 (Figure 10) highlights key indicators such as Time-to-Live (TTL) manipulation, high data transfers, anomalous packet sizes, and unusual service requests that appear to be strong signals of potential attacks. These patterns align well with known network attack behaviors such as scanning, DoS, data exfiltration, and spoofing.

TTL and state features are highly indicative of suspicious traffic in this model. Attackers often manipulate TTL values to avoid detection or maintain stealth in reconnaissance activities. Service and protocol features, such as service_dns and proto_unas, have notable impacts. DNS tunneling is a well-known technique for covert data exfiltration, and protocol anomalies often hint at unconventional or unauthorized access attempts.

Traffic volume and size features, such as dload, sbytes, smeansz, and network_bytes, capture unusual volumes or packet sizes, which are key indicators of both data exfiltration and DoS-type attacks. Large packet volumes or highly irregular packet sizes are red flags in network monitoring.

The SHAP results on both datasets illustrate the complexity of the model's decision-making process, encouraging greater trust from system experts by showing how the model's interpretation of features aligns with their understanding of network traffic patterns and malicious behavior. This transparency allows analysts to focus their efforts on the most critical indicators of attacks, streamlining the investigation process. Global explanations can also reveal systemic vulnerabilities or patterns in attack methods, such as frequent misuse of certain protocols or anomalies in data flow characteristics. This enables security teams to preemptively address recurring attacks or adjust monitoring tools to better detect emerging threats.

3.2. Stage II: Attack Classifier Results

In the second stage of XI2S-IDS, the focus is on classifying attack types using deep learning models trained exclusively on attack data. This stage builds upon the output of the binary classifier by categorizing detected attacks into their specific types, leveraging both BI-LSTM and a hybrid CNN-BI-LSTM architecture. To evaluate the performance of these models, we conducted experiments on the CICIDS2017 and UNSW-NB15 datasets, with and without SMOTE oversampling. Figures 11–14 show four confusion matrices for the four tested models on the CICIDS2917 dataset, while Figures 15–18 show four confusion matrices for the four tested models on the UNSW-NB15 dataset.

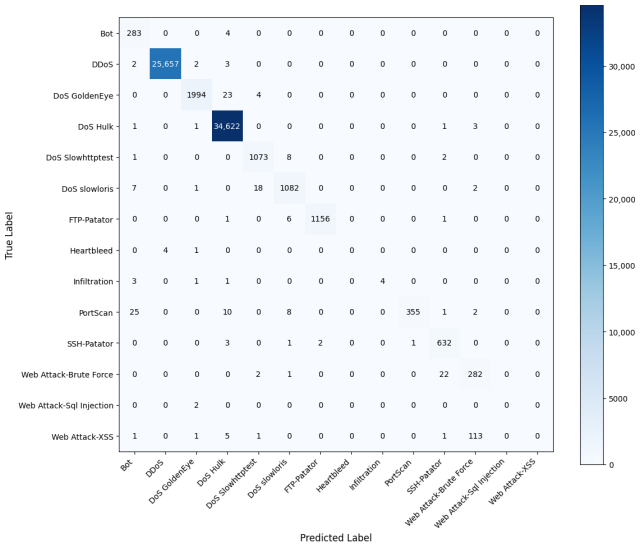

Figure 11. CICIDS2017 CM of the CNN-BI-LSTM model without SMOTE.

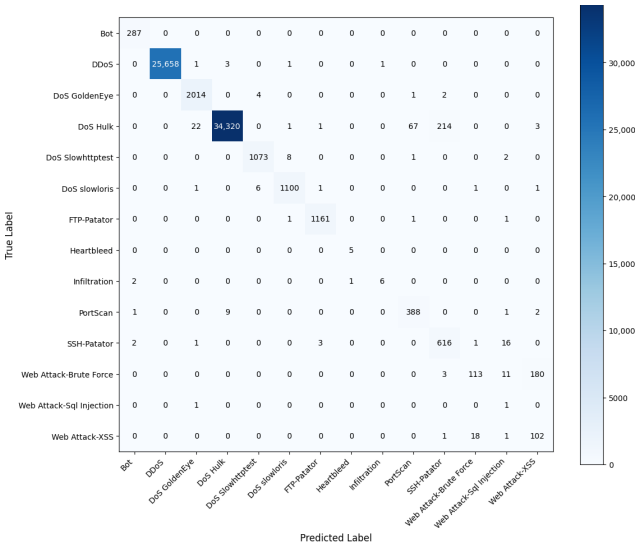

Figure 12. CICIDS2017 CM of the CNN-BI-LSTM model with SMOTE.

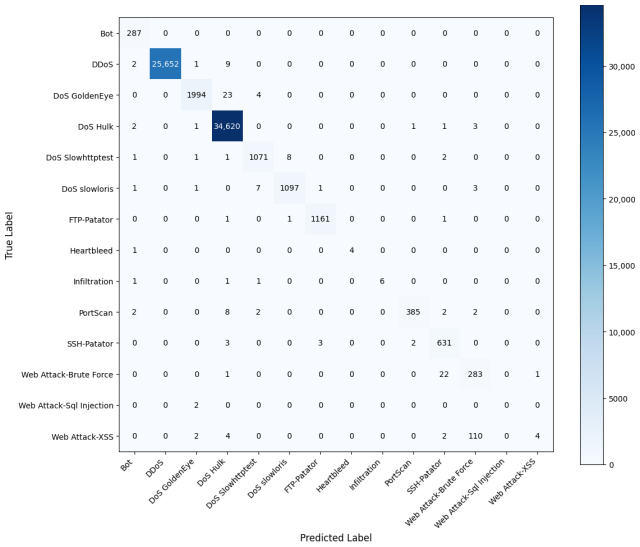

Figure 13. CICIDS2017 CM of the BI-LSTM model without SMOTE.

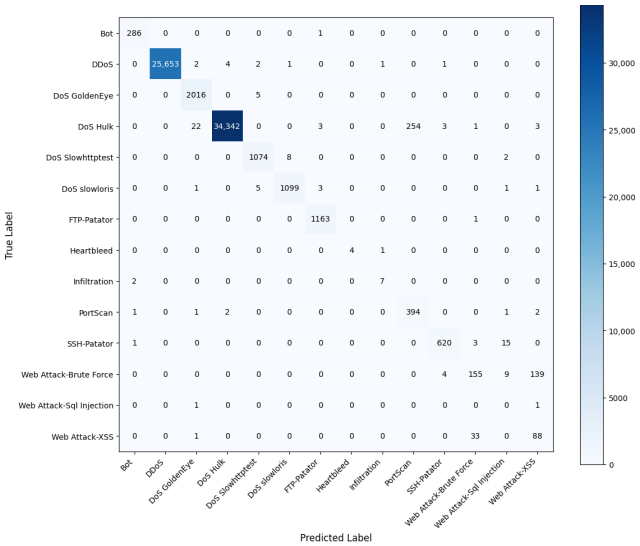

Figure 14. CICIDS2017 CM of the BI-LSTM model with SMOTE.

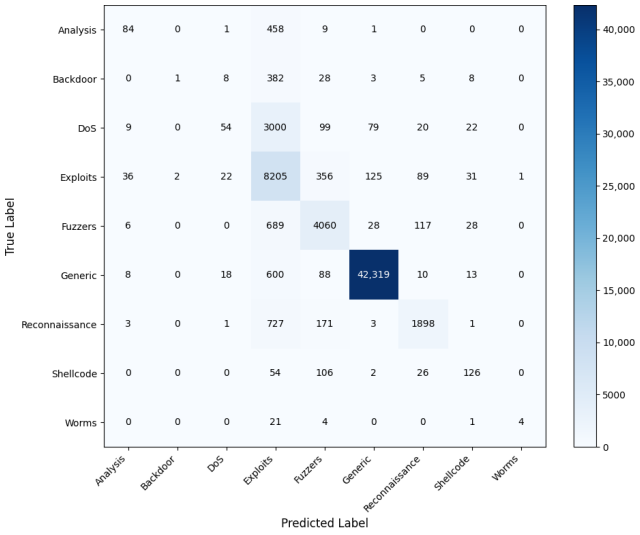

Figure 15. UNSW-NB15 CM of the CNN-BI-LSTM model without SMOTE.

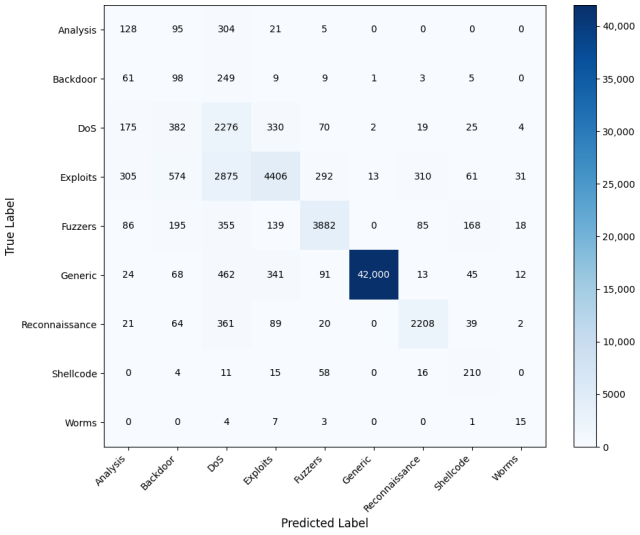

Figure 16. UNSW-NB15 CM of the CNN-BI-LSTM model with SMOTE.

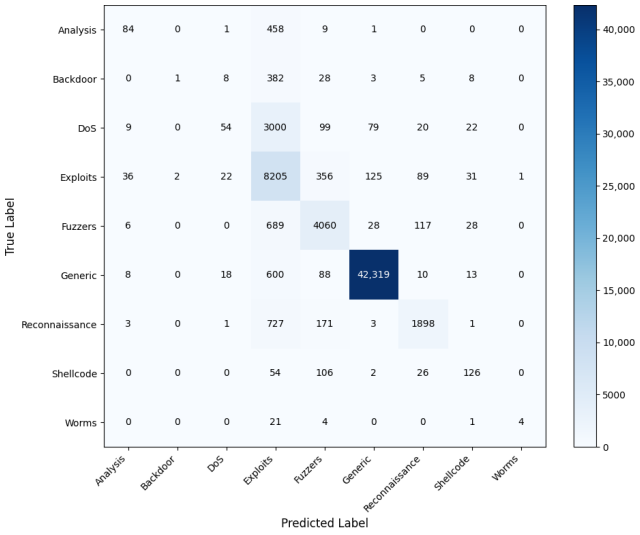

Figure 17. UNSW-NB15 CM of the BI-LSTM model without SMOTE.

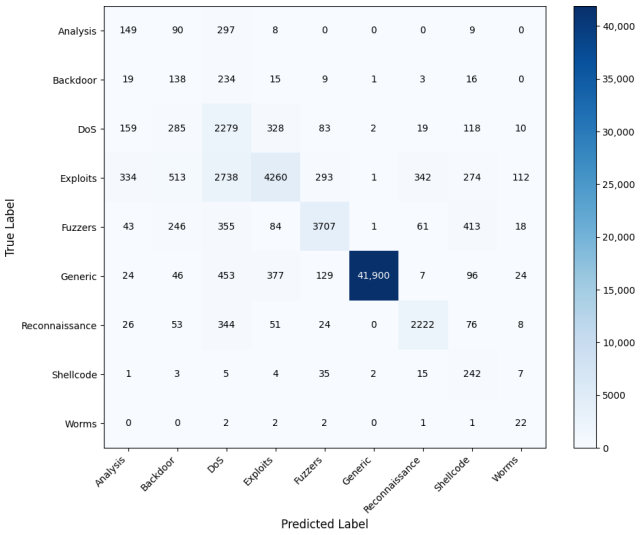

Figure 18. UNSW-NB15 CM of the BI-LSTM model with SMOTE.

The results clearly demonstrate that SMOTE oversampling improved the performance of both the BI-LSTM and CNN-BI-LSTM models. However, the models with SMOTE exhibited fairly similar performance, with BI-LSTM showing better detection of low-frequency attacks, while CNN-BI-LSTM was more effective at identifying common attacks. The models' performance without SMOTE was nearly identical, indicating that the addition of CNN layers did not contribute to any significant improvement in attack detection.

Given the primary objective of this research—enhancing the detection of low-frequency attacks—and the added complexity introduced by the CNN layers, the BI-LSTM model with SMOTE was selected as the multi-class classifier for XI2S-IDS.

3.3. XI2S-IDS Results

With the trained models, XI2S-IDS was ready for testing and classification. All testing records are passed to the binary classifier first. Whenever the classifier marks a record as an attack, it is sent to the multi-class classifier for further classification into the attack type. Meanwhile, global SHAP explanations of the binary classifier are available for system experts.

A prediction set is created in the testing phase. All records classified as normal are saved in the predictions set as a -1 value, while all abnormal classifications are passed to the multi-class classifier. The latter's predictions are saved in the prediction set with values

ranging from 0 to 8 for the 9 attack classes of the UNSW-NB15 dataset and from 0 to 13 for the 14 classes of attacks of the CICIDS2017 dataset.

When evaluating all test records, each prediction value is adjusted to correctly assign the class label for normal records, which were initially marked as -1. This adjustment aligns the predicted values with the true attack classes in the test set, ensuring the framework is properly assessed and accurate metric percentages can be generated. The pseudocode of the classification part for the UNSW-NB15 dataset is shown in Algorithm 1.

To evaluate the 2-stage IDS, a benchmark was created by training the same layers of the chosen BI-LSTM network on the full dataset similar to traditional 1-stage classification techniques. The results are compared together for both CICIDS2017 and UNSW-NB15 datasets as shown in Figures 19–22.

Algorithm 1 Classify Test Records.

Require: *records* $R \in$ *Test Data*
Require: *Binary classifier* BC, *Multiclass classifier* MC
Require: *empty list* $PRED$
for all $r \in R$ **do**
  $class \Leftarrow BC(r)$
  if *class* is 1 **then**
    $attack \Leftarrow MC(r)$
    $PRED \Leftarrow attack$
  else
    $PRED \Leftarrow -1$
  end if
end for
for all $p \in PRED$ **do**
  if $p \geq 6$ **then**
    $p = p + 1$
  end if
end for
for all $p \in PRED$ **do**
  if p is -1 **then**
    $p = 6$
  end if
end for

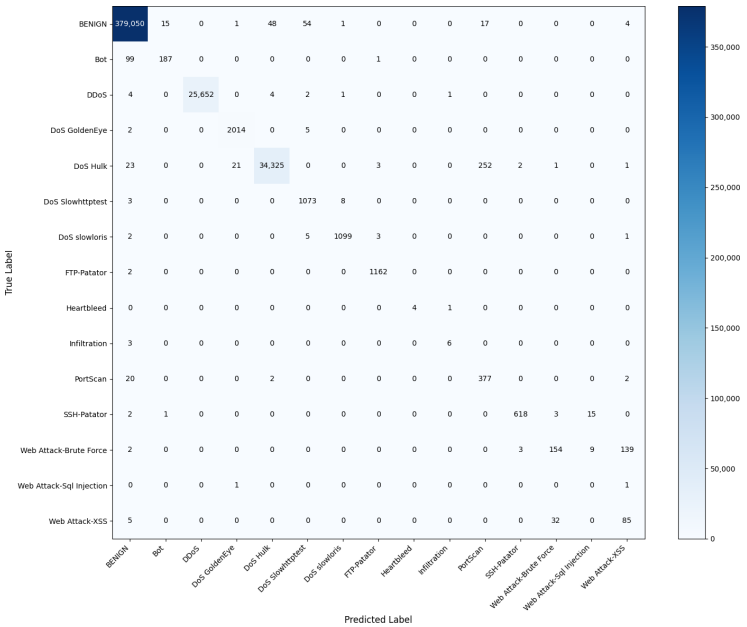

Figure 19. CM of 2-stage IDS on the CICIDS2017 dataset.

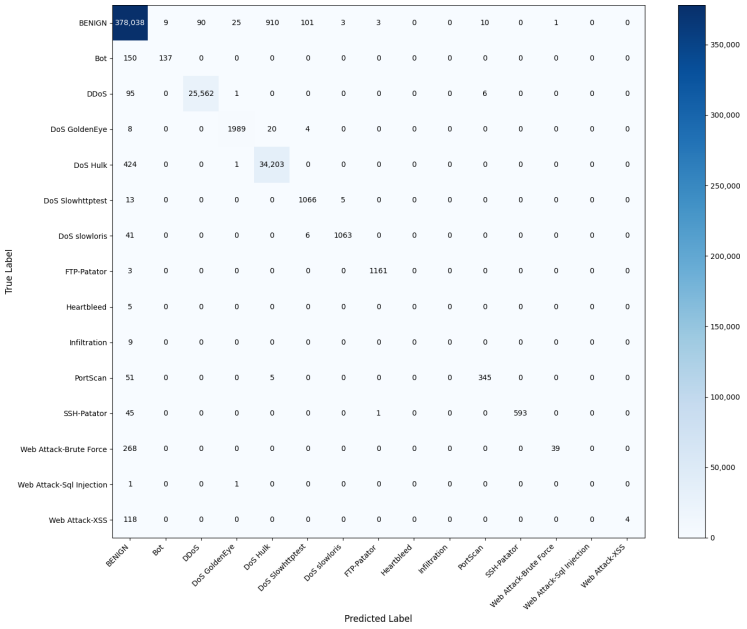

Figure 20. CM of 1-stage IDS on the CICIDS2017 dataset.

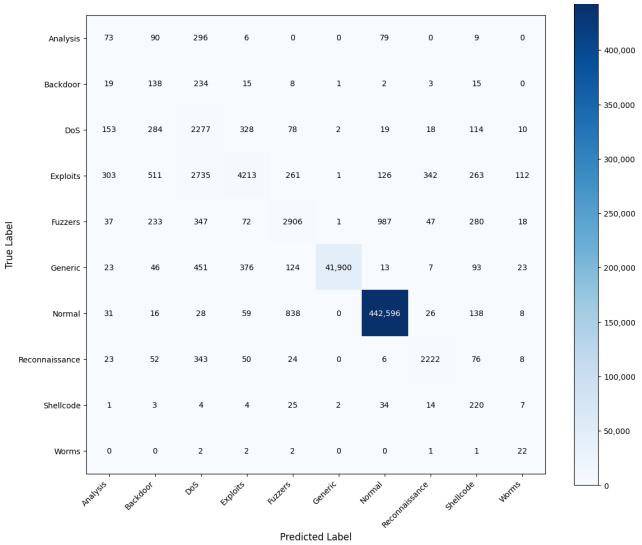

Figure 21. CM of 2-stage IDS on the UNSW-NB15 dataset.

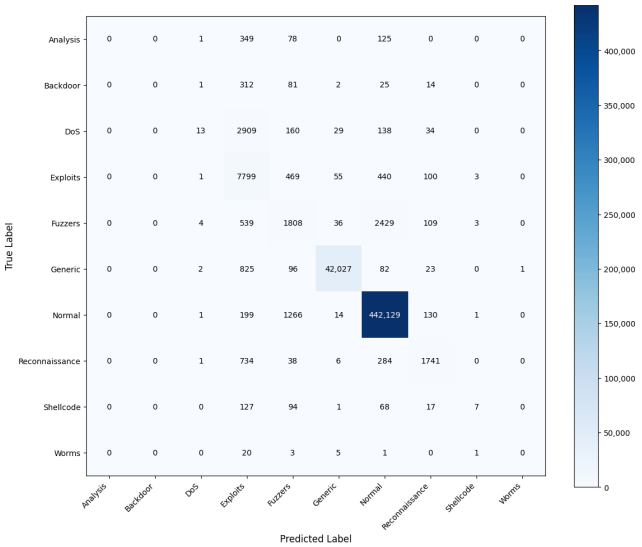

Figure 22. CM of 1-stage IDS on the UNSW-NB15 dataset.

Comparing the two confusion matrices of the CICIDS2017 dataset, it is clear that 2-stage IDS outperforms traditional 1-stage classification. The framework's 2-stage model is better at correctly identifying benign traffic. It also can handle low-frequency attack classes, such as Web Attack-XSS and Web Attack–Brute Force, better. The accuracy of the 2-stage framework is slightly higher than the traditional 1-stage IDS (0.9981 vs. 0.9945).

The same conclusion is reached for the UNSW-NB15 dataset. The framework's 2-stage model is better at correctly identifying benign traffic as well as handling low-frequency attack classes, such as Worms, Shellcode, Analysis, and Backdoor. The framework's accuracy is slightly higher (0.9774 vs. 0.9754).

Detailed results of the framework vs. the benchmark on both datasets are shown in Tables 6 and 7. The primary observation of the detailed results is that the 2-stage model generally outperforms the 1-stage model across multiple metrics, especially for low-frequency attack types where detection is more challenging.

Table 6. Performance metrics for the CICIDS2017 dataset.

	Precision		**Recall**		**F1-Score**	
	2-Stage	**1-Stage**	**2-Stage**	**1-Stage**	**2-Stage**	**1-Stage**
Benign	0.99	0.99	0.99	0.99	0.99	0.99
Bot	0.92	0.94	0.65	0.48	**0.76**	0.63
DDoS	0.99	0.99	0.99	0.99	0.99	0.99
DoS GoldenEye	0.99	0.99	0.99	0.98	0.99	0.99
DoS Hulk	0.99	0.97	0.99	0.00	**0.99**	0.98
DoS Slowhttptest	0.94	0.91	0.99	0.98	**0.97**	0.94
DoS slowloris	0.99	0.99	0.99	0.96	**0.99**	0.97
FTP-Patator	0.99	0.99	0.99	0.99	0.99	0.99
Heartbleed	0.99	0.00	0.80	0.00	**0.89**	0.00
Infiltration	0.75	0.00	0.67	0.00	**0.71**	0.00
PortScan	0.58	0.96	0.94	0.86	0.72	**0.91**
SSH-Patator	0.99	0.99	0.97	0.93	**0.98**	0.96
Web Attack–Brute	0.81	0.97	0.50	0.13	**0.62**	0.22
Web Attack-SQLi	0.00	0.00	0.00	0.00	0.00	0.00
Web Attack-XSS	0.36	0.99	0.70	0.03	**0.48**	0.06
Weighted avg	0.99	0.99	0.99	0.99	0.99	0.99

Bold values indicate the best F1-score across the 1-Stage and 2-Stage frameworks for each attack category and for the weighted average.

Table 7. Performance metrics for the UNSW-NB15 dataset.

	Precision		**Recall**		**F1-Score**	
	2-Stage	**1-Stage**	**2-Stage**	**1-Stage**	**2-Stage**	**1-Stage**
Analysis	0.11	0.00	0.13	0.00	**0.12**	0.00
Backdoor	0.10	0.00	0.32	0.00	**0.15**	0.00
DoS	0.34	0.54	0.69	0.00	**0.46**	0.01
Exploits	0.82	0.56	0.48	0.88	0.60	**0.69**
Fuzzers	0.68	0.44	0.59	0.37	**0.63**	0.40
Generic	0.99	0.99	0.97	0.98	0.99	0.99
Normal	0.99	0.99	0.99	0.99	0.99	0.99
Reconnaissance	0.83	0.80	0.79	0.62	**0.81**	0.70
Shellcode	0.18	0.47	0.70	0.02	**0.29**	0.04
Worms	0.11	0.00	0.73	0.00	**0.18**	0.00
Weighted avg	0.98	0.97	0.98	0.98	**0.98**	0.97

Bold values indicate the best F1-score across the 1-Stage and 2-Stage frameworks for each attack category and for the weighted average.

For the CICIDS2017 dataset, the 2-stage model performs significantly better in classes such as Bot and DoS Slowhttptest, where it achieves higher F1-scores compared to the 1-stage model. For instance, the 2-stage model achieves an F1-score of 0.76 for bot detection, compared to 0.63 for the 1-stage model, showing a marked improvement in handling precision–recall trade-offs. Similarly, for DoS Slowhttptest, the F1-score of 0.97 for the 2-stage model highlights superior classification accuracy. The 2-stage model also performs better on rare or difficult-to-detect classes like Web Attack–Brute Force and Infiltration, which are missed entirely by the 1-stage model.

Similarly, in the UNSW-NB15 dataset, the 2-stage model outperforms the 1-stage model on difficult-to-detect classes such as Analysis and Backdoor, where the 1-stage model fails to identify these attacks at all (with precision and recall both at 0). The 2-stage model shows some ability to detect these threats, achieving F1-scores of 0.12 and 0.15, respectively. For more common attack types like Exploits and Reconnaissance, the 2-stage model also demonstrates a better balance between precision and recall, leading to higher F1-scores, such as 0.60 for Exploits compared to 0.69 for the 1-stage model.

4. Discussion

In this paper, XI2S-IDS, an intelligent 2-stage approach to building a NIDS, has been presented. The approach makes use of the imbalance in the dataset by building a binary classifier that learns the normal behavior of users in the first stage, powered by SHAP explanations to increase trust in its decision, followed by a multi-class classifier for the second stage that only focuses on distinguishing the types of attacks as well as using SMOTE to oversample low-frequency attack records.

The results of the first stage show that LinearSVC and SGD variants (log, hinge, mod. huber) show decent performance but are clearly outperformed by tree-based ensemble models, especially in terms of recall and FNR. Ensemble models, XGBoost and LightGBM, dominate in both datasets, offering the best balance between high precision, recall, and low error rates, making them the most reliable models for both datasets.

Ensemble models perform better because they combine multiple Decision Trees, reducing both variance and bias. Their ability to handle complex, non-linear relationships leads to superior accuracy, recall, and F1-scores compared to simpler models. The boosting techniques used in these models allow them to iteratively learn from mistakes, making them more robust in detecting network traffic anomalies.

Linear models are precise but struggle with capturing complex patterns, while basic tree-based models can overfit and affect generalization. Ensemble models mitigate these issues by improving weak learners in each iteration, resulting in better performance across metrics. LightGBM was chosen as the binary classifier of the first stage. A comparison of LightGBM binary classification results with related work is shown in Table 8 for the CICIDS2017 dataset and in Table 9 for the UNSW-NB15 dataset. The comparison clearly demonstrates LightGBM's superior performance over recent related research.

Table 8. Comparison of proposed versus related work on the CICIDS2017 dataset for binary classification.

Classifier	Accuracy	Precision	Recall	F1-Score
J48 [53]	99.88	NaN	99	-
DNN 1 layer [21]	96.3	90.8	97.3	93.9
DBN [54]	97.7	96.08	97.53	97
DNN LSTM [55]	99.55	99.36	99.44	99.42
Proposed Model **LightGBM**	**99.9**	**99.7**	**99.7**	**99.7**

Bold values indicate the best results for each evaluation criterion.

Table 9. Comparison of proposed versus related work on the UNSW-NB15 dataset for binary classification.

Classifier	Accuracy	Precision	Recall	F1-Score
LSTM [56]	85.8	79.7	99.4	88.5
SKM-XGB [35]	99.08	**98.46**	97.46	97.84
DNN [57]	99	-	-	98.40
DNN LSTM [58]	81	83.55	81.38	80.51
Proposed Model **LightGBM**	**99.52**	98.21	**98.03**	**98.12**

Bold values indicate the best results for each evaluation criterion.

The second stage consisting of the attack-classifier highlights the correlated attacks that seem to be indistinguishable from classifiers. For example, in the CICIDS2017 dataset, the web attacks brute force and XSS seem to be tightly coupled. However, using SMOTE upsampling enhanced their classification rate. In UNSW-NB15, the Exploits attack was favored by the model when classifying different types of attacks, causing an issue in classification that was fixed using SMOTE upsampling. The detection of low-frequency attacks, a key challenge in the field, was significantly enhanced through the application of a two-stage classification process and the use of SMOTE for handling class imbalances. Comparing XI2S-IDS with the traditional benchmark of the 1-stage classifier highlighted the importance of learning normal behavior by leveraging the imbalance in the datasets. Results of the 1-stage benchmark classifier illustrate the classification of low-frequency attacks as normal records while dividing the classification into two stages effectively and correctly classifying low frequency attacks.

XI2S-IDS was able to decrease the FNR of the low-frequency attack classes in both datasets as shown in Tables 10 and 11. The performance on rare attack types such as Heartbleed and Infiltration was particularly notable. As seen in Table 10, the false negative rate (FNR) for Heartbleed dropped from 100% in the 1-stage model to 20% in XI2S-IDS, while Infiltration detection improved from 100% FNR to 33%. This dramatic improvement underscores the efficacy of the 2-stage approach in focusing on low-frequency attack detection, an area where most IDS traditionally struggle.

Table 10. FNR values of low-frequency attacks in the CICIDS2017 dataset.

CICIDS2017 Attacks FNR	Heartbleed	Infiltration	PortScan	SSH-Patator	Web Attack–Brute Force	Web Attack-Sql Injection	Web Attack-XSS
XI2S-IDS	20.00%	33.33%	3.99%	3.29%	50.16%	100.00%	32.79%
1-Stage IDS	100.00%	100.00%	13.97%	7.1987%	87.30%	100.00%	96.72%

Table 11. FNR values of low-frequency attacks in the UNSW-NB15 dataset.

UNSW-NB15 Attacks FNR	Analysis	Backdoor	DoS	Shellcode	Worms
XI2S-IDS	86.80%	68.28%	30.64%	29.94%	26.67%
1-Stage IDS	100.00%	100.00%	99.60%	97.77%	100.00%

Upon further examination of the confusion matrices for both XI2S-IDS and the benchmark model, it was observed that while the benchmark model exhibited a lower FNR for certain attack classes, such as Exploit, this came at the cost of increased misclassification in other attack categories. The benchmark model tended to classify multiple unknown attacks as Exploit, which is broadly defined as any attack exploiting system vulnerabilities. While this resulted in a lower FNR for Exploit attacks, it led to higher misclassification rates for other types of attacks, including Worms, Shellcode, DoS, Backdoor, and Analysis.

The performance of XI2S-IDS is compared against several related research on CICIDS2017 and UNSW-NB15 datasets as shown in Table 12 and Table 13, respectively. The superior performance of XI2S-IDS is attributed to its architecture and preprocessing steps. The two-stage design ensures that the binary classifier filters normal traffic effectively, enabling the second stage to focus solely on attack classification. This division of tasks optimizes the model's learning capacity for complex and subtle patterns. The use of SMOTE for oversampling addresses the inherent class imbalance in IDS datasets, allowing the multi-class classifier to better learn patterns associated with rare attacks.

Table 12. Multi-class results comparison of the XI2S-IDS and related research on the CICIDS2017 dataset.

Framework	Accuracy	Precision	Recall	F1
XI2S-IDS	**99.8%**	**99.9%**	**99.8%**	**99.8%**
GAN RF [59]	99.8%	98.6%	92.7%	95%
CNN LSTM ATT [29]		90%	81%	85%
XGB [30]	99.8%	99.8%	99.8%	99.8%

Bold values indicate the best results for each evaluation criterion.

Table 13. Multi-class results comparison of the XI2S-IDS and related research on the UNSW-NB15 dataset.

Framework	Accuracy	Precision	Recall	F1
XI2S-IDS	**97**%	**98.2**%	**97.3**%	**97.2**%
VAE [23]	93%	95.2%	91.9%	93%
DNN [21]	66%	62%	66%	59%
PSO-ACO-GA + Rotation forest and bagging [39]	91.2%	91.6%	91.3%	NA
SAE [60]	89.1%	89.1%	63.2%%	90.8%

Bold values indicate the best results for each evaluation criterion.

While XI2S-IDS demonstrates improvements in evaluation criteria over related work, it has certain limitations. The reliance on SMOTE for oversampling addresses class imbalance but introduces the risk of overfitting, as the synthetic samples may not fully capture the intricacies of real-world attack patterns, potentially impacting the model's ability to generalize to unseen data. Furthermore, while SHAP explanations provide valuable insights into the decision-making process, their computational complexity poses a challenge for real-time deployment, particularly in high-throughput environments where processing speed is critical.

5. Conclusions

In this paper, we proposed XI2S-IDS, an explainable two-stage intelligent intrusion detection system designed to enhance both the accuracy and interpretability of intrusion detection in network security. XI2S-IDS effectively addresses the challenges of low-frequency attack detection. This is achieved by combining a binary classification stage with a multi-class classifier trained exclusively on attack records, resulting in improved false negative rates (FNRs) for rare attack types.

XI2S-IDS integrates SHAP-based explanations, which provide valuable insights into feature importance and model behavior, enabling greater trust in automated IDS outputs. Experimental results on the UNSW-NB15 and CICIDS2017 datasets demonstrate the superior performance of XI2S-IDS compared to traditional single-stage classifiers, particularly in its handling of imbalanced attack types and low-frequency threats.

A significant challenge in developing IDS lies in the substantial class imbalance present in most available datasets [60,61], which leads to a high FNR for low-frequency attacks. Various researchers have proposed solutions to mitigate this issue, such as generating synthetic records to balance the dataset or using only a subset of the data [24,62,63]. XI2S-IDS addresses this challenge by training its multi-class classifier solely on attack records and employing SMOTE oversampling to enhance the detection of rare attacks, allowing the classifier to better learn the patterns of rare attacks and achieving low FNR as shown in Tables 10 and 11.

The binary classification stage plays a key role in determining the overall performance of XI2S-IDS, as it triggers the multi-class classification stage when an attack is detected. Therefore, it is essential for the binary classifier to learn normal patterns and subsequently detect any abnormal behavior regardless of the attack type. XI2S-IDS's binary classifier achieved an FNR of 0.0004 for the CICIDS2017 dataset and 0.019 for the UNSW-NB15 dataset, proving the effectiveness of the two stages in distinguishing and identifying attacks compared to related research.

Another ongoing challenge in IDS is the lack of real-time evaluation environments [60]. Many studies face limitations due to resource constraints or concerns around data privacy, making it difficult to assess the practical effectiveness of IDS in live systems. Real-time testing is crucial for guiding future IDS development and ensuring systems can adapt to dynamic network conditions. Detecting zero-day attacks also remains a critical unsolved problem in the field [64,65]. Previous studies have shown that new attacks can go undetected for an average of 312 days [66], presenting a major vulnerability for organizations.

This work highlights the significance of explainability in IDS, as well as the potential of a two-stage architecture in addressing complex detection challenges inherent in real-world network environments. Future work will focus on further reducing false negatives in the binary classification stage and refining the multi-class classification for highly correlated attack categories. Additionally, we plan to test XI2S-IDS in real-time scenarios to assess its performance under live network conditions. By advancing the interpretability and effectiveness of IDS frameworks, XI2S-IDS offers a promising direction for developing secure, transparent, and adaptable cybersecurity solutions in increasingly connected environments.

Author Contributions: All authors contributed to the conceptualization of this research and its methodology. A.A.A.-H. provided a comprehensive introduction to the Intrusion Detection field, while Y.O.Y. provided a detailed background on machine learning and deep learning techniques. M.M.M. was responsible for reviewing and summarizing the related research as well as conducting the technical implementation of the framework including data preprocessing, model training, and developing the framework's architecture under supervision from A.A.A.-H. and Y.O.Y. An in-depth analysis of the framework's results was collaboratively provided by A.A.A.-H. and Y.O.Y. The initial draft of the manuscript was written by M.M.M. All authors reviewed and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The UNSW-NB15 dataset used in this research is publicly available in the following URL (accessed on 6 January 2025), https://research.unsw.edu.au/projects/unsw-nb15-dataset [67] which includes CSV files, PCAP files, BRO files and Argus files. The CSV files of the complete dataset were chosen for this research. The CICIDS2017 dataset used in this research is publicly available in the following URL (accessed on 6 January 2025), https://www.unb.ca/cic/datasets/ids-2017.html [68].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Aldweesh, A.; Derhab, A.; Emam, A.Z. Deep Learning Approaches for Anomaly Based Intrusion Detection Systems: A Survey, Taxonomy, and Open Issues. *Know.-Based Syst.* **2020**, *189*, 105124. [CrossRef]
2. Karatas, G.; Demir, O.; Koray Sahingoz, O. Deep Learning in Intrusion Detection Systems. In Proceedings of the 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), Ankara, Turkey, 3–4 December 2018; pp. 113–116. [CrossRef]
3. Fatima, M.; Rehman, O.; Rahman, I.M.H.; Ajmal, A.; Park, S.J. Towards Ensemble Feature Selection for Lightweight Intrusion Detection in Resource-Constrained IoT Devices. *Future Internet* **2024**, *16*, 368. [CrossRef]
4. Yu, Y.; Long, J.; Cai, Z. Network Intrusion Detection through Stacking Dilated Convolutional Autoencoders. *Secur. Commun. Netw.* **2017**, *2017*, 1–10. [CrossRef]
5. AV-Test. Malware Statistics and Trends Report | AV-Test 2024. 2024. Available online: https://www.av-test.org/en/statistics/malware/ (accessed on 6 January 2025).
6. Mishra, P.; Varadharajan, V.; Tupakula, U.; Pilli, E.S. A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection. *IEEE Commun. Surv. Tutorials* **2019**, *21*, 686–728. [CrossRef]
7. Buczak, A.L.; Guven, E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *Commun. Surv. Tutorials* **2016**, *18*, 1153–1176. [CrossRef]
8. Aminanto, M.; Kim, K. *Improving Detection of Wi-Fi Impersonation by Fully Unsupervised Deep Learning*; Springer: Cham, Swizerland, 2018; pp. 212–223. [CrossRef]
9. Naseer, S.; Saleem, Y.; Khalid, S.; Bashir, M.K.; Han, J.; Iqbal, M.M.; Han, K. Enhanced Network Anomaly Detection Based on Deep Neural Networks. *IEEE Access* **2018**, *6*, 48231–48246. [CrossRef]
10. Shone, N.; Ngoc, T.N.; Phai, V.D.; Shi, Q. A Deep Learning Approach to Network Intrusion Detection. *IEEE Trans. Emerg. Top. Comput. Intell.* **2018**, *2*, 41–50. [CrossRef]
11. Mirsky, Y.; Doitshman, T.; Elovici, Y.; Shabtai, A. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. *arXiv* **2018**, arXiv:1802.09089.
12. Drewek-Ossowicka, A.; Pietrołaj, M.; Ruminski, J. A survey of neural networks usage for intrusion detection systems. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 497–514. [CrossRef]
13. Gupta, N.; Jindal, V.; Bedi, P. LIO-IDS: Handling class imbalance using LSTM and improved one-vs-one technique in intrusion detection system. *Comput. Netw.* **2021**, *192*, 108076. [CrossRef]
14. Anitha, T.; Aanjankumar, S.; Poonkuntran, S.; Nayyar, A. A novel methodology for malicious traffic detection in smart devices using BI-LSTM–CNN-dependent deep learning methodology. *Neural Comput. Appl.* **2023**, *35*, 20319–20338. [CrossRef]
15. Sinha, J.; Manollas, M. Efficient Deep CNN-BiLSTM Model for Network Intrusion Detection. In Proceedings of the Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Pattern Recognition, New York, NY, USA, 26–28 June 2020; AIPR 2020; pp. 223–231. [CrossRef]
16. Nguyen Dang, K.D.; Fazio, P.; Voznak, M. A Novel Deep Learning Framework for Intrusion Detection Systems in Wireless Network. *Future Internet* **2024**, *16*, 264. [CrossRef]
17. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **2021**, *8*, 53. [CrossRef] [PubMed]
18. Larsen, A.; Sønderby, S.; Winther, O. Autoencoding beyond pixels using a learned similarity metric. *arXiv* **2015**, arXiv:1512.09300. [CrossRef]
19. Aminanto, M.; Kim, K. *Detecting Impersonation Attack in WiFi Networks Using Deep Learning Approach*; Springer: Cham, Swizerland, 2017; pp. 136–147. [CrossRef]
20. Yan, B.; Han, G. Effective Feature Extraction via Stacked Sparse Autoencoder to Improve Intrusion Detection System. *IEEE Access* **2018**, *6*, 41238–41248. [CrossRef]
21. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access* **2019**, *7*, 41525–41550. [CrossRef]
22. Liu, J.; Xiao, K.; Luo, L.; Li, Y.; Chen, L. An intrusion detection system integrating network-level intrusion detection and host-level intrusion detection. In Proceedings of the 2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS), Macau, China, 11–14 December 2020; pp. 122–129. [CrossRef]
23. Yang, Y.; Zheng, K.; Wu, B.; Yang, Y.; Wang, X. Network Intrusion Detection Based on Supervised Adversarial Variational Auto-Encoder with Regularization. *IEEE Access* **2020**, *8*, 42169–42184. [CrossRef]
24. Yu, Y.; Bian, N. An Intrusion Detection Method Using Few-Shot Learning. *IEEE Access* **2020**, *8*, 49730–49740. [CrossRef]
25. Jin, D.; Lu, Y.; Qin, J.; Cheng, Z.; Mao, Z. SwiftIDS: Real-time intrusion detection system based on LightGBM and parallel intrusion detection mechanism. *Comput. Secur.* **2020**, *97*, 101984. [CrossRef]

26. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Proceedings of the Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2017; Volume 30.
27. Rajesh Kanna, P.; Santhi, P. Unified Deep Learning approach for Efficient Intrusion Detection System using Integrated Spatial–Temporal Features. *Knowl.-Based Syst.* **2021**, *226*, 107132. [CrossRef]
28. Zhao, Y.; Shen, Y.; Yao, J. Recurrent Neural Network for Text Classification with Hierarchical Multiscale Dense Connections. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; International Joint Conferences on Artificial Intelligence Organization; pp. 5450–5456. [CrossRef]
29. Psychogyios, K.; Papadakis, A.; Bourou, S.; Nikolaou, N.; Maniatis, A.; Zahariadis, T. Deep Learning for Intrusion Detection Systems (IDSs) in Time Series Data. *Future Internet* **2024**, *16*, 73. [CrossRef]
30. Korium, M.S.; Saber, M.; Beattie, A.; Narayanan, A.; Sahoo, S.; Nardelli, P.H. Intrusion detection system for cyber attacks in the Internet of Vehicles environment. *Ad Hoc Netw.* **2024**, *153*, 103330. [CrossRef]
31. Du, M.; Liu, N.; Hu, X. Techniques for interpretable machine learning. *Commun. ACM* **2019**, *63*, 68–77. [CrossRef]
32. Dias, T.; Oliveira, N.; Sousa, N.; PraÁa, I.; Sousa, O. A Hybrid Approach for an Interpretable and Explainable Intrusion Detection System. *arXiv* **2021**, arXiv:2111.10280.
33. Dong, T.; Li, S.; Qiu, H.; Lu, J. An Interpretable Federated Learning-based Network Intrusion Detection Framework. *arXiv* **2022**, arXiv:2201.03134. [CrossRef]
34. Kumar, C.; Ansari, M.S.A. An explainable nature-inspired cyber attack detection system in Software-Defined IoT applications. *Expert Syst. Appl.* **2024**, *250*, 123853. [CrossRef]
35. Hooshmand, M.K.; Huchaiah, M.D.; Alzighaibi, A.R.; Hashim, H.; Atlam, E.S.; Gad, I. Robust network anomaly detection using ensemble learning approach and explainable artificial intelligence (XAI). *Alex. Eng. J.* **2024**, *94*, 120–130. [CrossRef]
36. Shtayat, M.M.; Hasan, M.K.; Sulaiman, R.; Islam, S.; Khan, A.U.R. An Explainable Ensemble Deep Learning Approach for Intrusion Detection in Industrial Internet of Things. *IEEE Access* **2023**, *11*, 115047–115061. [CrossRef]
37. Ring, M.; Wunderlich, S.; Scheuring, D.; Landes, D.; Hotho, A. A Survey of Network-based Intrusion Detection Data Sets. *arXiv* **2019**, arXiv:1903.02460. [CrossRef]
38. Fernández Maimó, L.; Perales Gómez, Á.L.; García Clemente, F.J.; Gil Pérez, M.; Martínez Pérez, G. A Self-Adaptive Deep Learning-Based System for Anomaly Detection in 5G Networks. *IEEE Access* **2018**, *6*, 7700–7712. [CrossRef]
39. Adhi Tama, B.; Comuzzi, M.; Rhee, K.H. TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly based Intrusion Detection System. *IEEE Access* **2019**, *7*, 94497–94507. [CrossRef]
40. Alawad, N.A.; Abed-alguni, B.H.; Al-Betar, M.A.; Jaradat, A. Binary improved white shark algorithm for intrusion detection systems. *Neural Comput. Appl.* **2023**, *35*, 19427–19451. [CrossRef]
41. Kourid, A.; Chikhi, S.; Recupero, D.R. Fuzzy optimized V-detector algorithm on Apache Spark for class imbalance issue of intrusion detection in big data. *Neural Comput. Appl.* **2023**, *35*, 19821–19845. [CrossRef]
42. Sharafaldin, I.; Habibi Lashkari, A.; Ghorbani, A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy—Volume 1: ICISSP, INSTICC, SciTePress, Funchal, Portugal, 22–24 January 2018; pp. 108–116.
43. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia, 10–12 November 2015; pp. 1–6. [CrossRef]
44. Moustafa, N.; Slay, J. The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Inf. Secur. J. Glob. Perspect.* **2016**, *25*, 18–31. [CrossRef]
45. Moustafa, N.; Slay, J.; Creech, G. Novel Geometric Area Analysis Technique for Anomaly Detection Using Trapezoidal Area Estimation on Large-Scale Networks. *IEEE Trans. Big Data* **2019**, *5*, 481–494. [CrossRef]
46. Moustafa, N.; Creech, G.; Slay, J. Big Data Analytics for Intrusion Detection System: Statistical Decision-Making Using Finite Dirichlet Mixture Models. In *Data Analytics and Decision Support for Cybersecurity: Trends, Methodologies and Applications*; Palomares Carrascosa, I., Kalutarage, H.K., Huang, Y., Eds.; Springer International Publishing: Cham, Swizerland, 2017; pp. 127–156. [CrossRef]
47. Sarhan, M.; Layeghy, S.; Moustafa, N.; Portmann, M. NetFlow Datasets for Machine Learning-based Network Intrusion Detection Systems. *arXiv* **2020**, arXiv:2011.09144.
48. Mahmoud, M.M.; Belal, N.A.; Youssif, A. Prediction of Transcription Factor Binding Sites of SP1 on Human Chromosome1. *Appl. Sci.* **2021**, *11*, 5123. [CrossRef]
49. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
50. Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 4–9 December 2017; NIPS'17, pp. 4768–4777.

51. Imrana, Y.; Xiang, Y.; Ali, L.; Abdul-Rauf, Z. A bidirectional LSTM deep learning approach for intrusion detection. *Expert Syst. Appl.* **2021**, *185*, 115524. [CrossRef]
52. Imrana, Y.; Xiang, Y.; Ali, L.; Abdul-Rauf, Z.; Hu, Y.C.; Kadry, S.; Lim, S. x2-BidLSTM: A Feature Driven Intrusion Detection System Based on x2 Statistical Model and Bidirectional LSTM. *Sensors* **2022**, *22*, 2018. [CrossRef]
53. Kurniabudi, K.; Stiawan, D.; Dr, D.; Bin Idris, M.Y.; Bamhdi, A.M.; Budiarto, R. CICIDS-2017 Dataset Feature Analysis with Information Gain for Anomaly Detection. *IEEE Access* **2020**, *8*, 132911–132921. [CrossRef]
54. Manimurugan, S.; Al-Mutairi, S.; Aborokbah, M.M.; Chilamkurti, N.; Ganesan, S.; Patan, R. Effective Attack Detection in Internet of Medical Things Smart Environment Using a Deep Belief Neural Network. *IEEE Access* **2020**, *8*, 77396–77404. [CrossRef]
55. Razib, M.A.; Javeed, D.; Khan, M.T.; Alkanhel, R.; Muthanna, M.S.A. Cyber Threats Detection in Smart Environments Using SDN-Enabled DNN-LSTM Hybrid Framework. *IEEE Access* **2022**, *10*, 53015–53026. [CrossRef]
56. Keshk, M.; Koroniotis, N.; Pham, N.; Moustafa, N.; Turnbull, B.; Zomaya, A.Y. An explainable deep learning-enabled intrusion detection framework in IoT networks. *Inf. Sci.* **2023**, *639*, 119000. [CrossRef]
57. Houda, Z.A.E.; Brik, B.; Khoukhi, L. "Why Should I Trust Your IDS?": An Explainable Deep Learning Framework for Intrusion Detection Systems in Internet of Things Networks. *IEEE Open J. Commun. Soc.* **2022**, *3*, 1164–1176. [CrossRef]
58. Sharma, B.; Sharma, L.; Lal, C.; Roy, S. Explainable artificial intelligence for intrusion detection in IoT networks: A deep learning based approach. *Expert Syst. Appl.* **2024**, *238*, 121751. [CrossRef]
59. Lee, J.; Park, K.H. GAN-based imbalanced data intrusion detection system. *Pers. Ubiquitous Comput.* **2019**, *25*, 121–128. [CrossRef]
60. Khan, F.A.; Gumaei, A.; Derhab, A.; Hussain, A. A Novel Two-Stage Deep Learning Model for Efficient Network Intrusion Detection. *IEEE Access* **2019**, *7*, 30373–30385. [CrossRef]
61. Sivatha Sindhu, S.S.; Geetha, S.; Kannan, A. Decision tree based light weight intrusion detection using a wrapper approach. *Expert Syst. Appl.* **2012**, *39*, 129–141. [CrossRef]
62. Haider, W.; Hu, J.; Slay, J.; Turnbull, B.; Xie, Y. Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling. *J. Netw. Comput. Appl.* **2017**, *87*, 185–192. [CrossRef]
63. Shiomoto, K. Network Intrusion Detection System Based on an Adversarial Auto-Encoder with Few Labeled Training Samples. *J. Netw. Syst. Manag.* **2022**, *31*, 5. [CrossRef]
64. Huda, S.; Miah, S.; Mehedi Hassan, M.; Islam, R.; Yearwood, J.; Alrubaian, M.; Almogren, A. Defending unknown attacks on cyber-physical systems by semi-supervised approach and available unlabeled data. *Inf. Sci.* **2017**, *379*, 211–228. [CrossRef]
65. Hindy, H.; Atkinson, R.; Tachtatzis, C.; Colin, J.N.; Bayne, E.; Bellekens, X. Utilising Deep Learning Techniques for Effective Zero-Day Attack Detection. *Electronics* **2020**, *9*, 1684. [CrossRef]
66. Guo, Y. A review of Machine Learning-based zero-day attack detection: Challenges and future directions. *Comput. Commun.* **2023**, *198*, 175–185. [CrossRef] [PubMed]
67. Moustafa, N.; Slay, J. *The UNSW-NB15 Dataset*; UNSW: Sydney, Australia, 2015. [CrossRef]
68. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. The CICIDS2017 Dataset. 2017. Available online: https://www.unb.ca/cic/datasets/ids-2017.html (accessed on 6 January 2025).