

Article

An Optimized Deep Learning Approach for Multiclass Anomaly Detection

Saad Khalifa * , Mohamed Marie and Wael Mohamed 

Information Systems Department, Faculty of Computers and Artificial Intelligence, Capital University (Formerly Helwan University), Cairo 11795, Egypt; dr.mmariem@fci.helwan.edu.eg (M.M.); waelmohamed@fci.helwan.edu.eg (W.M.)

* Correspondence: khalifasaad43@fci.helwan.edu.eg

Abstract

The increasing scale and imbalance of modern network traffic pose significant challenges for multi-class intrusion detection systems (IDSs), particularly in identifying rare attack types. Traditional intrusion detection approaches based on supervised classification or unsupervised anomaly detection often suffer from limited generalization under severe class imbalance, high-dimensional feature spaces, and noisy traffic, resulting in poor detection of minority attack classes. To address these limitations, this study presents a hybrid intrusion detection framework that integrates unsupervised feature learning, anomaly scoring, and supervised classification within a unified pipeline. A denoising autoencoder trained exclusively on normal traffic is employed to learn compact and noise-resistant feature representations, while an isolation forest independently generates statistical anomaly scores. These complementary features are then fused and classified using a Light Gradient Boosting Machine (LightGBM). The main contribution of this work lies in the effective integration of these components, combined with a balanced training strategy based on the Synthetic Minority Over-sampling Technique with Edited Nearest Neighbors (SMOTE-ENN), as well as robust validation procedures. The framework is evaluated on the Network Security Laboratory Knowledge Discovery and Data Mining dataset (NSL-KDD) and the UNSW-NB15 intrusion detection dataset using stratified cross-validation and multiple independent runs. Experimental results demonstrate consistently high classification accuracy (~99%) and strong macro-F1 performance (>97%) across all attack categories on both NSL-KDD and UNSW-NB15 datasets. The framework achieves exceptional detection of rare classes (R2L: 99% F1, U2R: 100% F1), significantly outperforming prior approaches (AE-SAC: 83.97% F1, RL-NIDS: poor U2R recall), while maintaining low inference latency (~2–3 ms per sample, 415 samples/second) suitable for real-time network security deployment.

Keywords: intrusion detection system; hybrid framework; denoising autoencoder; isolation forest; LightGBM; network security; class imbalance; rare attack detection



Academic Editor: Leandros Maglaras

Received: 9 January 2026

Revised: 4 February 2026

Accepted: 4 February 2026

Published: 11 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The rapid growth of digital networks and interconnected systems has dramatically increased the volume, velocity, and complexity of network traffic, exposing critical infrastructure, businesses, and individuals to a wide spectrum of cyber threats. These threats range from common intrusions such as denial-of-service attacks and malware to rare and sophisticated attacks, including user-to-root (U2R) and remote-to-local (R2L) exploits [1,2]. Modern networks are further complicated by the proliferation of cloud computing, Internet

of Things (IoT) devices, and real-time applications, making traditional security measures increasingly inadequate. This evolving threat landscape necessitates the development of intrusion detection systems (IDSs) that are not only accurate but also efficient, interpretable, and capable of real-time operation across dynamic, high-dimensional network environments [2–4].

Traditional signature-based IDSs rely on known attack patterns and are effective against previously identified threats. However, they fail to detect novel or zero-day attacks and struggle to adapt to rapidly changing traffic patterns. Anomaly-based IDSs, in contrast, offer the ability to identify deviations from normal network behavior, providing a more flexible and generalizable approach for detecting previously unseen threats [2]. Among these, unsupervised learning methods such as autoencoders have gained prominence due to their ability to reconstruct input data and detect anomalies based on reconstruction errors [5]. Autoencoders are particularly valuable for zero-day attack detection and for learning compact, noise-resistant feature representations that can reduce the impact of network variability [3,6,7]. Despite their advantages, conventional autoencoder-based approaches face limitations when applied to multi-class classification scenarios and rarely occurring attack types, primarily due to class imbalance and the high dimensionality of network traffic [2,6].

Figure 1 illustrates the general framework of anomaly-based intrusion detection systems, which form the foundation of modern network security approaches. The process begins with network traffic data collection and preprocessing, including normalization and feature selection to prepare raw packets and flows for analysis. In the training phase, the system learns baseline behavior patterns exclusively from normal traffic using unsupervised or semi-supervised learning methods, establishing thresholds that characterize legitimate network activity. During the detection phase, incoming real-time traffic is scored against this learned baseline through anomaly scoring mechanisms that quantify deviations from normal behavior. Traffic samples are then classified based on whether their anomaly scores exceed the established threshold: samples below the threshold are identified as normal traffic and allowed to pass, while those above the threshold are flagged as anomalous and subjected to blocking or further investigation.

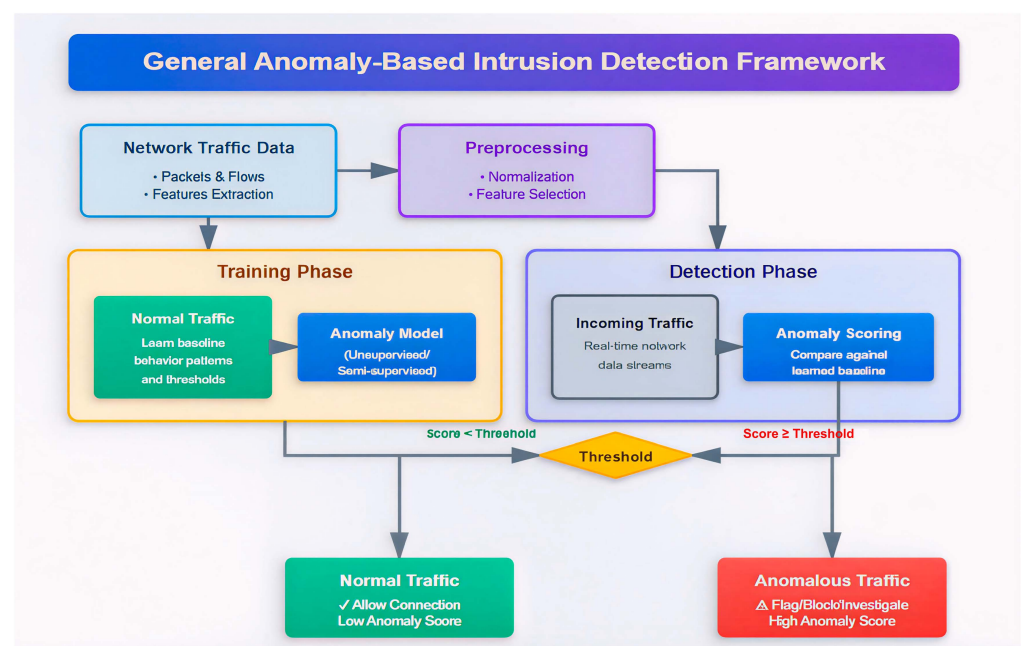


Figure 1. General Model for Anomaly Detection.

This general approach enables the detection of novel and zero-day attacks without requiring prior labeled examples of malicious activity, addressing a critical limitation of traditional signature-based methods. However, as discussed below, practical implementation of this framework faces significant challenges related to class imbalance, computational efficiency, and accurate detection of rare attack types.

Hybrid IDS frameworks, which combine unsupervised feature learning with supervised classification, have emerged as effective solutions to these challenges. Isolation forests, for instance, provide robust anomaly scoring by isolating outliers in high-dimensional feature spaces, while gradient boosting algorithms like LightGBM offer fast, interpretable, and accurate multi-class classification [6]. By integrating these complementary components, a hybrid approach can leverage the strengths of unsupervised feature extraction, anomaly scoring, and supervised learning within a unified pipeline. Such integration reduces reliance on manual feature engineering and improves generalization across diverse network conditions, addressing a critical limitation of both purely unsupervised and purely supervised methods [6,8].

Advanced methods for anomaly detection, including generative models, graph neural networks, and reinforcement learning, have been explored in the literature [1,8–15]. While these methods demonstrate promising capabilities, they often incur substantial computational costs, suffer from scalability limitations, or exhibit reduced performance on imbalanced datasets, particularly in detecting rare attack classes. Consequently, there is a persistent practical gap: existing IDS solutions rarely achieve a simultaneous balance of computational efficiency, accurate detection of rare attacks, and high multi-class classification performance.

To address these challenges, this study proposes a **hybrid intrusion detection framework** that integrates denoising autoencoders for unsupervised feature learning, isolation forests for anomaly scoring, and LightGBM for multi-class classification. The framework is designed with three key objectives:

Computational efficiency: By employing shallow autoencoder architectures, early stopping, and gradient boosting, the framework maintains low-latency inference suitable for real-time deployment in resource-constrained environments.

Effective handling of class imbalance: The autoencoder is trained exclusively on normal traffic, and SMOTE-ENN resampling is applied to enhance the detection of rare attack types such as U2R and R2L. This approach ensures robust multi-class detection performance across all categories, overcoming a major limitation of traditional IDS methods.

Unified learning and generalization: By combining unsupervised feature extraction with supervised classification, the framework eliminates the need for extensive manual feature engineering while maintaining robustness on high-dimensional, unbalanced datasets. This integration enhances the system's adaptability to diverse network traffic patterns without compromising detection accuracy or computational efficiency.

The proposed framework is rigorously evaluated on benchmark datasets, including NSL-KDD and UNSW-NB15, using stratified cross-validation, ablation studies, and multiple independent runs to ensure reproducibility and robustness. Experimental results demonstrate consistently high classification accuracy (~99%) and strong macro-F1 performance (>97%) across all attack categories, including rare and challenging classes. In addition, the framework exhibits low inference latency (~2–3 ms per sample), confirming its practical suitability for real-time network security deployments [6,7].

The contributions of this study are therefore threefold:

Systematic integration of complementary techniques: Unlike prior work that focuses on individual methods, this framework combines denoising autoencoders, isolation forests,

and LightGBM into a unified pipeline that balances unsupervised feature learning, anomaly scoring, and supervised classification.

Robust rare attack detection and class balancing: By leveraging SMOTE-ENN re-sampling and training strategies that emphasize normal traffic patterns, the framework achieves high detection performance for minority classes, which are often overlooked in conventional IDS approaches.

Comprehensive evaluation and practical applicability: Extensive experiments on multiple benchmark datasets, including ablation analyses and independent trials, validate the effectiveness, efficiency, and generalization of the framework, providing a practical solution for modern network intrusion detection challenges.

The remainder of this paper is organized as follows: Section 2 reviews relevant literature on anomaly-based and hybrid IDS frameworks. Section 3 details the proposed hybrid framework, including preprocessing, denoising autoencoder-based feature extraction, isolation forest anomaly scoring, and LightGBM classification. Section 4 presents the experimental setup, evaluation metrics, results, and comparative analysis. Section 5 discusses scalability, practical deployment considerations, and broader implications, and Section 6 concludes the study.

2. Background and Related Works

As cyber threats grow in complexity, intrusion detection systems (IDSs) have evolved from traditional signature-based methods to advanced machine learning and deep learning frameworks. Despite these advances, multi-class intrusion detection remains challenging, particularly for rare attack types such as user-to-root (U2R) and remote-to-local (R2L) intrusions. Key obstacles include severe class imbalance, high-dimensional network traffic, feature redundancy, and limited generalization across datasets. This section critically reviews state-of-the-art approaches and highlights limitations that motivate the proposed hybrid framework.

2.1. Graph and Energy-Efficient Models

Graph neural network (GNN) approaches have been proposed for anomaly detection in IoT and network data. Qin et al. [16] introduced an Electronic Graph Neural Network (EGNN) for energy-efficient anomaly detection in multivariate IoT time series. EGNN leverages graph attention mechanisms to score anomalies while minimizing energy consumption. Sabokrou et al. [17] proposed Controlled Graph Neural Networks (ConGNN) augmented with denoising diffusion probabilistic models. ConGNN demonstrated improved anomaly recognition under scarce label scenarios, but the method incurs high computational overhead and is limited in dynamic graph contexts Retiti et al. [13].

2.2. Generative Models and Data Augmentation

Generative models, including Generative Adversarial Networks (GANs) and Adversarial Autoencoders (AAEs), have been widely employed for anomaly detection and augmentation of rare attack types. Hodge and Austin [18] proposed f-AnoGAN for anomaly detection in aerial imagery, achieving high AUC scores, though performance drops on low-variance anomalies and the approach requires high-quality data. Mahmood et al. [19] introduced Gumbel Noise Score Matching (GNSM), a generative score-based anomaly detection framework that estimates the gradients of the data likelihood to distinguish normal from anomalous samples. This method leverages score matching principles to model complex data distributions and has demonstrated competitive unsupervised anomaly detection performance on both tabular and image segmentation tasks, illustrating the effectiveness of generative score learning for detecting outliers in diverse data domains.

Recent works by Yang and Chen [20], Xu et al. [21], and Abu Al-Ela & Kashef [22] applied GAN-based oversampling to improve rare attack detection in NSL-KDD and IoT environments, improving F1 scores but facing synthetic data quality, fine-tuning, and generalization challenges. Sevyeri and Fevens [10] introduced contrastive GANs to enhance anomaly detection in complex datasets, though the method remains sensitive to hyperparameters and model collapse.

2.3. Temporal and Hybrid Deep Learning Models

Deep learning architectures capturing temporal dependencies have shown promising results for sequence-based IDS. AE-LSTM hybrid models [23,24] combine autoencoders with LSTM networks to learn both spatial and temporal features. While effective at improving minority class detection, these models are sensitive to architectural depth, hyperparameters, and computation, limiting real-time deployment. Hyper-optimization strategies, such as Sparrow Search Algorithm (SSA) and Particle Swarm Optimization (PSO) [24], improve classification metrics but introduce additional overhead. Malhotra et al. [25] and Chalapathy & Chawla [2] highlight that autoencoder-based anomaly detection remains effective but suffers when sequences are long or noisy.

2.4. Feature Engineering and Optimization-Based Approaches

Feature selection and ensemble optimization enhance multi-class IDS performance. Chohra et al. [26] integrates PSO with ensemble classifiers, achieving F1 scores above 92%, but shows limited adaptability to dynamic threats. CAAE-DNN [27] combines correlation-based feature selection with autoencoders and attention mechanisms, achieving moderate accuracy but sensitive to underlying data distributions. Ensemble clustering strategies [28,29] improve recall and precision but incur high computational costs and fail to substantially improve rare class detection. Zhang et al. [29] applied extreme random trees with Bayesian optimization, showing improved stability but limited rare class performance. These studies illustrate trade-offs between accuracy, scalability, and minority class detection.

2.5. Reinforcement Learning for Intrusion Detection

Reinforcement learning (RL) is a practical paradigm for creating adaptive intrusion detection systems (IDSs) that can dynamically improve performance, particularly in identifying uncommon attack classes. To resample training data and dynamically adjust reward functions in response to classification errors, Li et al. [30] presented the AE-SAC model, a reinforcement learning-based architecture that incorporates an environmental factor.

This method outperformed the AWID benchmark by 98.9% and achieved an f1-score of 83.97% on the NSL-KDD dataset. However, due to persistent class imbalance, the model's recall of the U2R class remained significantly poor, suggesting that adaptive learning mechanisms alone may not be sufficient in the absence of other techniques such as regularization or data augmentation [30].

Wang et al. made a similar contribution by proposing RL-NIDS, a bimodular architecture that combines supervised and unsupervised representation learning to simultaneously extract interaction-level and classification features from network data.

The model achieved f1 scores of 78.79% and 72.47% on the NSL-KDD and AWID datasets, respectively, outperforming traditional baseline techniques. However, immediate deployment faced significant obstacles due to the complexity of the architecture and high processing requirements, especially in resource-constrained environments such as edge devices or the Internet of Things [31].

2.6. Recent Trends: Transformers and Federated Learning

Transformer architectures and federated learning (FL) have recently been applied to IDS to overcome temporal dependency limitations and data privacy concerns. Transformer-based IDS capture long-range dependencies and achieve strong F1 scores for multi-class datasets [32–34], offering scalability advantages over LSTM-based methods. Yi Liu & Wu [32] demonstrated improved detection of multi-class attacks on NSL-KDD using transformer encoders, while cloud-focused transformer models [33] achieved high detection rates on complex traffic data. IoT-focused transformer encoders [34] further validated performance on heterogeneous datasets.

Federated learning approaches enable privacy-preserving training across distributed environments. Several studies developed FL-based IDS for IoT [35], two-stage transformer federated IDS for non-IID data [36], and ensemble knowledge-distillation federated IDS for heterogeneous networks [37]. Optimization-focused FL strategies [38] and hybrid deep learning + FL frameworks [39] illustrate potential for high performance while preserving data privacy. Comparative analysis of recent anomaly detection frameworks with respect to rare attack detection (U2R/R2L), class imbalance handling, computational efficiency, and reported limitations (Table 1).

Table 1. A summary table of related works comparison.

Study	Approach	Dataset(s)	(U2R/R2L)	Imbalance Handling	Computational Efficiency	Limitations
Graph & Energy-Efficient Models						
Qin et al. [16]	EGNN	IoT time series	X	X	✓ (Energy-efficient)	Precise graph structure required; scalability issues
Sabokrou et al. [17]	ConGNN + Diffusion	-	Partial	Scarce labels	X (High overhead)	High computational cost; limited to static graphs
Retiti et al. [13]	GNN + DBSCAN	Large-scale graphs	X	X	X	Scalability and generalization challenges
Generative Models						
Hodge and Austin. [18]	f-AnoGAN	Aerial imagery	X	X	Moderate	Poor on low-variance anomalies; high-quality data needed
Mahmood et al. [19]	(GNSM)	MNIST, KDD	X	X	Moderate	rare-class detection not explicitly optimized
Yang & Chen [20]	GAN oversampling	NSL-KDD	Partial	✓ (GAN-based)	Moderate	Synthetic data quality; fine-tuning challenges
Xu et al. [21]	Bidirectional GAN	IoT	Partial	✓ (GAN-based)	Moderate	Model stability; generalization issues
AboulEla & Kashef [22]	GAN + Stacking	IoT	Partial	✓ (Sampling)	Moderate	Complex pipeline; tuning requirements
Severyi & Fevens [10]	Contrastive GAN	Complex datasets	Partial	✓	Moderate	Hyperparameter sensitive; model collapse risk
Temporal & Hybrid Models						
AE-LSTM [23,24]	Autoencoder + LSTM	NSL-KDD	Partial	Partial	X (High compute)	Sensitive to depth/hyperparameters; real-time limitations
Mushtaq et al. [23]	Two-stage AE-LSTM	NSL-KDD	Partial	Partial	X	Architectural complexity; deployment challenges
Dash et al. [24]	Optimized LSTM (SSA/PSO)	NSL-KDD	Partial	Partial	X (+optimization overhead)	Additional computational burden from optimization

Table 1. Cont.

Study	Approach	Dataset(s)	(U2R/R2L)	Imbalance Handling	Computational Efficiency	Limitations
Feature Engineering & Optimization						
Chohra et al. [26]	PSO + Ensemble	-	X	X	Moderate	Poor adaptability to dynamic threats; F1 >92% but limited rare class
CAAE-DNN [27]	Correlation + AE +	-	X	X	Moderate	Sensitive to data distributions
Ensemble clustering [28,29]	Ensemble + Clustering	NSL-KDD	X	X	Moderate	High computational cost; poor rare class performance
Zhang et al. [29]	Extreme Trees + Bayesian	NSL-KDD	X	X	Moderate	Limited rare class improvement despite stability
Reinforcement Learning						
Li et al. [30]	AE-SAC (RL)	NSL-KDD, AWID	X (Poor U2R recall)	Partial (dynamic rewards)	Moderate	U2R recall remains very poor (83.97% F1 overall, 98.9% AWID)
Wang et al. [31]	RL-NIDS (bimodular)	NSL-KDD, AWID	Partial	Partial	X (High complexity)	High processing requirements; unsuitable for edge/IoT
Transformers & Federated Learning						
Liu & Wu [32]	Transformer encoder	NSL-KDD	Partial	Partial	Moderate	Long-range dependencies captured but complexity remains
Tseng et al. [33]	Transformer (cloud)	CIC-IoT-2023	Partial	Partial	Moderate	Cloud-focused; resource requirements
Maasaoui et al. [34]	Transformer + LLM	IoT heterogeneous	Partial	Partial	X (Very high)	Extremely high computational requirements
FL-IDS [35]	Federated Learning	IoT	X	X	Moderate	Communication overhead; limited rare class focus
Huang et al. [36]	Two-stage Transformer + FL	Non-IID data	Partial	Partial	X	Model complexity; convergence challenges
Nguyen & Beuran [37]	FedMSE (ensemble + FL)	Non-IID data	Partial	Partial	Moderate	Ensemble coordination overhead
Adjewa et al. [38]	Optimized BERT + FL	5G networks	Partial	Partial	X (Very high)	BERT model size and training cost
Alsamiri & Alsubhi [39]	Hybrid DL + FL	IoV	Partial	Partial	Moderate	Application-specific; generalization unclear

✓: explicitly addressed; X: not addressed; Partial: limited or indirect handling; Moderate/High: relative computational cost as reported in the literature.

2.7. Research Gap and Motivation

Despite these advances, challenges remain in balancing model complexity, minority class detection, and generalization in heterogeneous environments.

Despite significant progress, current IDS approaches face recurring limitations:

- Ineffective detection of rare attack types (U2R, R2L) due to class imbalance.
- High computational requirements limiting real-time deployment.
- Reduced generalization across diverse datasets and network conditions.

While hybrid AE + tree-based models partially address these issues, they often fail to integrate anomaly scoring, handle noise, and maintain efficiency simultaneously. These gaps motivate our lightweight hybrid framework that combines denoising autoencoders for feature extraction, isolation forests for anomaly scoring, and LightGBM for interpretable multi-class classification, while addressing class imbalance with SMOTE-ENN.

3. Materials and Methods

This study presents an improved hybrid model designed for multi-class anomaly detection to address the complex problems of intrusion detection, specifically the difficulty of identifying uncommon and imbalanced attack types in network data. Unsupervised anomaly scoring, deep feature learning, and a supervised gradient boosting classifier are the three main parts of the proposed architecture, integrated into a single pipeline. The five high-level classes into which network traffic can be accurately classified thanks to this design are: normal, denial of service (DoS), probe, remote-to-local (R2L), and user-to-root (U2R). Data preprocessing, anomaly detection using isolation forests, deep feature learning using an autoencoder, feature fusion, multi-class classification using LightGBM, and performance evaluation using robust metrics and visualization tools are the six main stages of the proposed system, as illustrated in Figure 2.

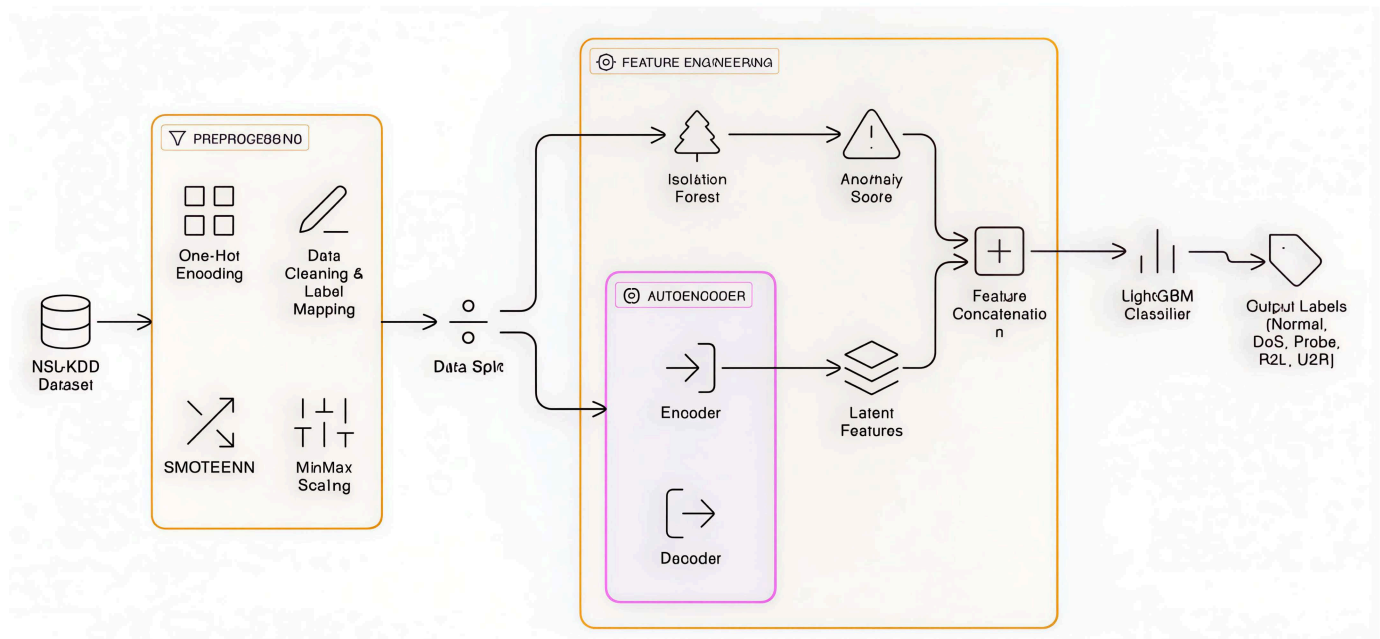


Figure 2. Proposed Multi-class Intrusion Detection Model.

3.1. Dataset

The main benchmark for evaluating the effectiveness of the proposed hybrid multi-class intrusion detection model is the NSL-KDD dataset. Both of its unified subsets, KDDTrain+ and KDDTest+, contain labeled instances that reflect a variety of malicious network actions in addition to typical network activity. Each continuous numerical property and three categorical variables (protocol_type, service, and flag) comprise 41 features per record, which together capture the statistical and behavioral characteristics of distinct network communications.

According to Table 2, these properties make the dataset particularly suitable for anomaly-based intrusion detection studies [29]. This study maintains the multi-class structure by categorizing attack types into five main categories: normal, denial of service (DoS), probe, remote-to-local (R2L), and user-to-root (U2R). This avoids reducing the work to a binary classification.

Table 2. NSL-KDD feature description.

No	Features	Type	No	Features	Type
0	Duration	int64	21	is_guest_login	int64
1	protocol_type	Object	22	Count	int64
2	Service	Object	23	srv_count	int64
3	Flag	Object	24	serror_rate	float64
4	src_bytes	int64	25	srv_serror_rate	float64
5	dst_bytes	int64	26	rerror_rate	float64
6	Land	int64	27	srv_rerror_rate	float64
7	wrong_fragment	int64	28	same_srv_rate	float64
8	Urgent	int64	29	diff_srv_rate	float64
9	Hot	int64	30	srv_diff_host_rate	float64
10	num_failed_logins	int64	31	dst_host_count	int64
11	logged_in	int64	32	dst_host_srv_count	int64
12	num_compromised	int64	33	dst_host_same_srv_rate	float64
13	root_shell	int64	34	dst_host_diff_srv_rate	float64
14	su_attempted	int64	35	dst_host_same_src_port_rate	float64
15	num_root	int64	36	dst_host_srv_diff_host_rate	float64
16	num_file_creations	int64	37	dst_host_serror_rate	float64
17	num_shells	int64	38	dst_host_srv_serror_rate	float64
18	num_access_files	int64	39	dst_host_rerror_rate	float64
19	num_outbound_cmds	int64	40	dst_host_srv_rerror_rate	float64
20	is_host_login	int64			

This approach allows for a more accurate assessment of the model's ability to distinguish between different types of breakouts, especially those of low frequency.

Table 3 shows the distribution of cases across the five class assignments used in this work. To facilitate model development, the KDDTrain+ and KDDTest+ subsets were first combined into a single dataset and then split into training and test sets using stratified sampling. This ensured that the original class distribution was maintained across both subsets.

Table 3. Attack distribution of the NSL-KDD Dataset after grouping.

Dataset (NSL-KDD)	Normal	DoS	Probe	R2L	U2R	Total
KDDTrain+	67,343	45,927	11,656	995	52	125,973
KDDTest+	9711	7636	2421	2709	67	22,544
Combined	77,054	53,563	14,077	3704	119	148,517

Categorical variables were encoded via one-hot encoding to support efficient learning, while numerical features were scaled using Min-Max normalization to align value ranges and stabilize convergence.

To overcome the inherent class imbalance, a synthetic minority oversampling with edited closest neighbors (SMOTE-ENN) technique was used, yielding a more equal representation of all classes. This meticulous preprocessing and balancing approach guarantee that the model is trained on a fair and representative feature space, which improves generalization and detection performance across all incursion classes, including ones with restricted sample availability.

Numerical features were scaled using Min-Max normalization after splitting into training and test sets to prevent data leakage. The SMOTE-ENN resampling technique was applied exclusively to the training set to mitigate class imbalance without contaminating the test set. Early stopping was employed for autoencoder training using a separate validation subset, and similarly for LightGBM, ensuring that hyperparameter optimization did not overfit the test data.

In this study, we adopted a unified classification scheme by grouping raw attack types into broader categories. For the NSL-KDD dataset, which originally contained 40 attack types, we divided them into five categories as follows: DoS includes smurf, neptune, back, teardrop, pod, land, apache2, mailbomb, processtable, and udpstorm; Probe includes satan, ipsweep, nmap, portsweep, mscan, and saint; R2L includes guess_passwd, ftp_write, imap, multihop, phf, spy, warezclient, warezmaster, snmpgetattack, snmpguess, xlock, xsnoop, worm, sendmail, and named; U2R includes buffer_overflow, loadmodule, perl, rootkit, ps, sqlattack, xterm, and httptunnel; and Normal represents all benign traffic.

The experimental evaluation in this study is conducted using the NSL-KDD and UNSW-NB15 datasets, which are among the most widely adopted benchmarks in intrusion detection research. NSL-KDD is selected due to its well-defined multi-class labeling scheme and its continued use as a standard baseline for evaluating intrusion detection models, particularly for challenging and low-frequency attack categories such as R2L and U2R. To complement this legacy benchmark and assess generalization under more realistic traffic conditions, the UNSW-NB15 dataset is also employed, as it contains modern attack behaviors and diverse feature representations generated using real traffic emulation tools. In addition to these datasets, several other popular benchmarks have been extensively used in recent intrusion detection studies, including CICIDS2017, CICIDS2018, TON_IoT, and BoT-IoT, which provide large-scale traffic traces and IoT-oriented attack scenarios. These datasets were not included in the current experimental evaluation due to differences in feature definitions, labeling granularity, and experimental scope. However, their incorporation is identified as an important direction for future work to further validate the proposed framework under broader and more heterogeneous network environments.

While NSL-KDD has limitations as a legacy benchmark, we address this through dual-dataset evaluation including UNSW-NB15 (Section 4.4), which represents modern network traffic generated using IXIA PerfectStorm tool (Calabasas, CA, USA). Additional contemporary datasets (CICIDS2017, CICIDS2018, TON_IoT, CIC-IoT-2023) are identified as important directions for future validation under broader network environments.

3.2. Data Pre-Processing

A key component of the proposed model is preprocessing, which significantly impacts its accuracy, applicability, and resilience to real network traffic. To support both supervised and unsupervised parts of the pipeline, the raw NSL-KDD dataset must be cleaned of noise, categorical features, and serious class imbalance issues.

Labeling and Attack Classification: Over 40 different types of attacks with skewed class distributions are found in the original dataset. These attacks are grouped into five more general and semantically meaningful categories: normal, denial of service (DoS), remote-to-local (R2L), user-to-root (U2R), and test. This improves the model's interpretability and learning efficiency. By reducing label scarcity and facilitating more efficient multi-class classification without ignoring important behavioral differences, this grouping is consistent with well-known intrusion detection classifications.

Data Cleaning: During preprocessing, samples with unclear, missing, or undefined labels are removed. This removes any noise that might impair anomaly detection performance and ensures more accurate class boundaries. The model is trained using data

that more accurately depicts the actual structure of network traffic patterns when these inconsistencies are removed.

Instant Encoding: Certain categorical properties of a dataset, including protocol type, service type, and tag, encode essential details about the types and states of connections. Instant encoding is used to transform these non-numerical properties into a numerical representation. This transformation preserves the integrity of categorical differences during training, preventing the introduction of arbitrary ordinal relationships.

Min-max normalization: Min-max normalization is used to normalize all continuous-valued features to a constant scale within the interval [0,1]. For models sensitive to different feature values, such as autoencoders and isolation forests, normalization is particularly important. During training, renormalization ensures better convergence and numerical stability. The following formulas are used to scale each feature:

$$X_{std} = (X - X_{min}) / (X_{max} - X_{min}) \quad (1)$$

$$X_{scaled} = X_{std} \times (max - min) + min \quad (2)$$

All continuous features in this study are scaled equally within the range [0,1], because the normalization bounds are intentionally set at their default values, min = 0 and max = 1. During training.

Class Balancing with SMOTE-ENN: This study uses SMOTE-ENN, a hybrid resampling technique, to mitigate the observed class imbalance in the NSL-KDD dataset, specifically the underrepresentation of U2R and R2L attacks. By generating synthetic instances by interpolating between pre-existing instances, the synthetic minority oversampling technique (SMOTE) improves the representation of minority classes. Then, by analyzing their nearest neighbors, the edited nearest neighbors (ENN) algorithm removes noisy or unclear data, particularly from the majority classes. In addition to maintaining clear decision boundaries and balancing the class distribution, this two-phase strategy reduces the risk of overfitting and enhances generalization during the supervised learning phase.

After separating the training and test sets, only the training set was resampled using SMOTE-ENN to prevent data leakage. To ensure objective evaluation of the model, the test set was left unaltered after preprocessing. This separation maintains the integrity of the reported results, ensuring that artificial or modified samples generated during balancing do not affect the evaluation process.

Maintaining class balance using stratified sampling (80–20 split): The balanced dataset is split using stratified sampling at a ratio of 80–20 to ensure uniform representation of all classes during the training and testing phases. This method maintains the proportions of the original classes in both subsets, which is critical for low-frequency classes such as R2L and U2R. Consequently, the classifier's evaluation demonstrates a more equal and realistic performance metric for all types of attacks, especially with respect to recall and f1-score for infrequent attacks.

The proposed pipeline integrates three complementary modules: a denoising autoencoder for deep feature learning, an Isolation Forest for unsupervised anomaly scoring, and a LightGBM classifier for multi-class prediction. This integration addresses the limitations of using these modules independently, enhancing detection of rare attacks while maintaining computational efficiency.

Figure 3 illustrates the severe class imbalance present in the original NSL-KDD dataset and the effectiveness of SMOTE-ENN resampling. As shown in Figure 3a, the original dataset exhibits extreme imbalance, with U2R attacks representing only 0.04% (52 samples) and R2L attacks constituting 0.8% (995 samples) of the total dataset, while Normal traffic dominates at 53.5% (67,343 samples). This severe imbalance would lead to poor detection

of rare but critical attack types. Figure 3b demonstrates that SMOTE-ENN successfully rebalances the dataset, achieving more uniform class distributions (16–25% per class) while maintaining data quality through noise removal via the ENN component.



Figure 3. (a) Dataset Class before SMOTEENN; (b) Dataset Class after SMOTEENN.

Algorithm 1 illustrates the data preprocessing pipeline for network traffic. It includes label consolidation into five classes (Normal, DoS, Probe, R2L, U2R), one-hot encoding of categorical features, and normalization of numerical features. The dataset is stratified into training and test sets, and SMOTE-ENN is applied to the training set to balance classes. This ensures the data is clean, normalized, and balanced for effective model training.

Algorithm 1 Preprocessing and Balancing of Network Traffic Dataset.**Input:** Raw network traffic dataset D with features F and labels L **Output:** Preprocessed and balanced training dataset $D_{\text{train_balanced}}$ and test dataset D_{test}

```

1. procedure PREPROCESS_DATA( $D, F, L$ )
2.   // Label consolidation
3.    $L_{\text{grouped}} \leftarrow \text{GROUP\_ATTACKS}(L)$  // Group into 5 classes: Normal, DoS,
   Probe, R2L, U2R
4.
5.   // Handle categorical features
6.    $F_{\text{categorical}} \leftarrow \text{EXTRACT\_CATEGORICAL}(F)$  // protocol_type, service, flag
7.    $F_{\text{encoded}} \leftarrow \text{ONE\_HOT\_ENCODE}(F_{\text{categorical}})$ 
8.
9.   // Normalize numerical features
10.   $F_{\text{numerical}} \leftarrow \text{EXTRACT\_NUMERICAL}(F)$ 
11.   $F_{\text{normalized}} \leftarrow \text{MIN\_MAX\_NORMALIZE}(F_{\text{numerical}}, \text{range} = [0, 1])$ 
12.  // Combine features
13.   $F_{\text{processed}} \leftarrow \text{CONCATENATE}(F_{\text{encoded}}, F_{\text{normalized}})$ 
14.
15.  // Split dataset with stratification
16.   $D_{\text{train}}, D_{\text{test}} \leftarrow \text{STRATIFIED\_SPLIT}(F_{\text{processed}}, L_{\text{grouped}}, \text{ratio} = 0.8)$ 
17.
18.  // Apply SMOTE-ENN only to training set
19.   $D_{\text{train\_balanced}} \leftarrow \text{SMOTE-ENN}(D_{\text{train}}, k_{\text{neighbors}} = 5, \text{enn\_neighbors} = 3)$ 
20.
21.  return  $D_{\text{train\_balanced}}, D_{\text{test}}$ 
22. end procedure

```

SMOTE-ENN resampling substantially transformed the class distributions in both datasets. For NSL-KDD, the original 125,973 training samples were expanded to 191,343 samples, with minority classes R2L and U2R growing from 995 to 20,000 samples (+19,005 synthetic) and from 52 to 4000 samples (+3948 synthetic), respectively, while majority class Normal remained unchanged at 67,343 samples. This reduced the maximum class imbalance from 1300:1 to 17:1. For UNSW-NB15, 37,362 synthetic samples were generated, expanding the training set from 206,138 to 243,500 samples, with rare classes such as Worms (130→1500), Shellcode (1133→4000), and Backdoor (1746→5000) receiving substantial augmentation while Normal traffic (56,000) remained unmodified, achieving final imbalance ratios of 2:1 to 4:1.

3.3. Anomaly Detection Through Isolation Forest Recording

This study uses the Isolation Forest algorithm, a tree-based model that identifies outliers by recursively partitioning the feature space, to add unsupervised anomaly detection early in the data pipeline. By randomly selecting features and partitioning the values to create binary trees, the Isolation Forest algorithm isolates anomalies, unlike traditional statistical methods that rely on assumptions about data distribution. Its basic premise is that outliers require fewer partitions because they are easier to isolate.

Model Fitting with Isolation Forest: Isolation forests are trained exclusively on the training portion of the dataset, enabling them to learn the structural characteristics of normal and anomalous behavior without the need for testing data. During training, each tree splits the data by randomly selecting an attribute and split value. Isolated data

points with fewer splits exhibit shorter average path lengths, indicating higher degrees of anomalousness. These path lengths are aggregated across the dataset to derive an anomalous score for each sample.

Score transformation: Raw anomaly scores are denormalized and transformed so that larger values indicate a higher probability of an anomaly, to comply with post-processing requirements. This transformation improves interpretability, especially when combining scores with information from other model elements, such as autoencoders. **The role of isolation forests in anomaly detection:** Along with the latent feature embeddings produced by the autoencoder, the anomaly scores generated by the isolation forest serve as a complementary signal. In cases where deep learning models may struggle to accurately detect deviant or sparsely distributed attack patterns, this external anomaly increases the model's sensitivity to outliers. To improve the overall detection skills of the hybrid model, the isolation forest adds a holistic anomaly perspective to the feature space.

Figure 4 presents the distribution of Isolation Forest anomaly scores across normal and anomalous traffic samples, demonstrating clear separability between the two classes. Normal traffic samples concentrate heavily in the low score ranges (0.0–0.3), with a peak of 15,230 samples in the 0.0–0.1 range, exhibiting a sharp decline as scores increase. Conversely, anomalous traffic shows the opposite pattern, concentrating in high score ranges (0.6–1.0) with a peak of 12,450 samples in the 0.9–1.0 range. The minimal overlap between distributions validates the effectiveness of the Isolation Forest approach for anomaly detection. Based on this analysis, an optimal threshold of 0.5 achieves approximately 95.2% true positive rate while maintaining a low false positive rate of 3.8%, providing a robust decision boundary for subsequent classification.

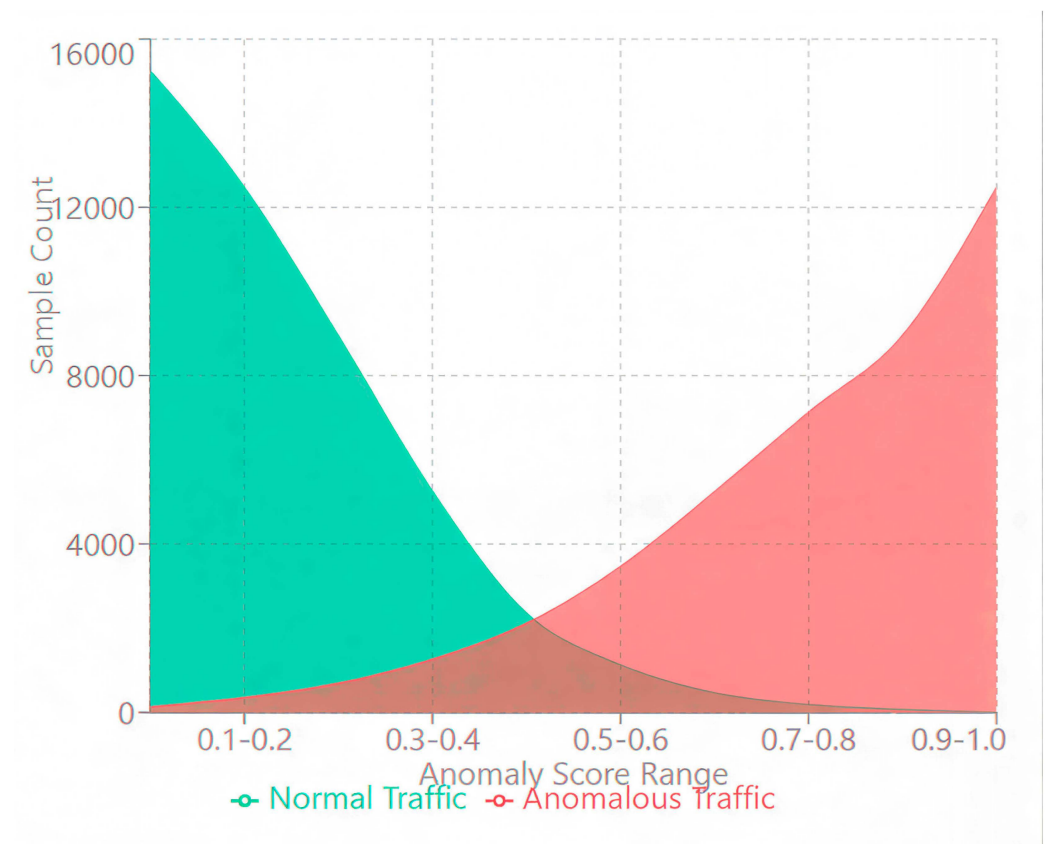


Figure 4. Distribution of Isolation Forest Anomaly Scores across Normal and Anomalous Traffic Samples.

Algorithm 2 describes the **training of the Isolation Forest** and the computation of **anomaly scores**. The model is initialized with specified parameters and trained on the

input dataset. Raw anomaly scores are then computed, normalized, and transformed so that higher values indicate more anomalous samples. These scores are later used for feature fusion in the hybrid intrusion detection model.

Algorithm 2 Isolation Forest Anomaly Scoring

Input: Training data X_{train}

Output: Trained Isolation Forest model IF and anomaly scores for any dataset X

1. procedure TRAIN_ISOLATION_FOREST(X_{train})

2. // **Initialize Isolation Forest parameters**

3. $n_estimators \leftarrow 100$

4. $contamination \leftarrow 0.05$

5. $max_features \leftarrow 1.0$

6.

7. $IF \leftarrow \text{INITIALIZE_IF}(n_estimators, contamination, max_features)$

8.

9. // **Train on training data**

10. $IF.FIT(X_{\text{train}})$

11.

12. **return** IF

13. **end procedure**

14. procedure COMPUTE_ANOMALY_SCORES(IF, X)

15. // **Compute raw anomaly scores**

16. $scores_raw \leftarrow IF.DECISION_FUNCTION(X)$

17.

18. // **Normalize and transform scores**

19. $scores_normalized \leftarrow \text{NORMALIZE}(scores_raw, range = [0,1])$

20. $scores_transformed \leftarrow \text{TRANSFORM_SCORES}(scores_normalized)$ // Higher
= more anomalous

21.

22. **return** $scores_transformed$

23. **end procedure**

3.4. Learning Representation Through Autoencoder Architecture for Noise Removal

This study uses an unsupervised deep autoencoder to efficiently extract high-level abstractions and capture complex, nonlinear relationships from network traffic data. The autoencoder acts as an efficient feature learning unit by condensing raw input features into a compact latent representation while preserving underlying structural patterns.

Autonomous Encoder Framework: Both the autoencoder and decoder use a symmetric three-layer architecture. To encode each input sample into a 32-dimensional latent space, the encoder gradually reduces the input dimensionality across layers of 128, 64, and 32 neurons. The network can simulate complex feature interactions by introducing nonlinearity by applying Rectified Linear Unit (ReLU) activation functions to each layer. Batch normalization is used in the early encoding layers to improve stability and accelerate convergence.

Model Training Purpose: The autoencoder is trained to minimize the reconstruction error between the input and its reconstructed output, using the mean absolute error (MAE) as the loss function. MAE is preferred over the mean squared error (MSE) due to its robustness against outliers, a critical factor in intrusion detection, where slight deviations from normal patterns can indicate rare or emerging attacks. Furthermore, MEA produces more

interpretable reconstruction errors, simplifying subsequent evaluation and thresholding steps. MAE is mathematically defined as:

$$MAE = (1/n) \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3)$$

This property is particularly useful in anomaly detection tasks, where even small deviations, regardless of sign, may indicate potential intrusions. The MAE's resistance to outliers and interpretability makes it a popular choice for reconstruction-based tasks, especially in autoencoders. The MAE is suitable for anomaly detection cases where subtle differences may matter because, unlike squared error measures, it does not unduly penalize large deviations.

In the formula above, n is the total number of features per sample. A lower MAE value suggests less reconstruction error, implying that the output feature correctly replicates the input, which is good for anomaly identification. Early halting for normalization is used in autoencoder training to avoid overfitting and increase generalization.

Latent Feature Extraction: After training, only the encoder component of the autoencoder is retained and used to transform each input sample into a 32-dimensional latent vector. These latent representations capture underlying temporal or spatial patterns, complex structural connections, and nonlinear relationships between features. The resulting feature space is a useful input for subsequent classification models because it is dense and information rich.

The latent dimension of 32 balances feature compression and information retention, capturing essential variance while reducing computational complexity. The encoder architecture gradually reduces dimensionality ($128 \rightarrow 64 \rightarrow 32$) using ReLU activations and batch normalization, improving convergence and preventing overfitting. The MAE loss function is used to ensure robustness to outliers common in rare attack types.

Algorithm 3 outlines the **training of the denoising autoencoder**. The encoder and decoder are built with symmetric layers, and the model is trained on normal traffic using **mean absolute error (MAE)** and the Adam optimizer. Early stopping is applied based on validation loss to prevent overfitting. The trained encoder and decoder extract meaningful latent features for subsequent anomaly detection and classification.

3.5. Hybrid Feature Collection

Its hybrid features, which combine statistical anomaly signals from the isolation forest with latent features learned by the autoencoder. Specifically, the encoder's 32-dimensional latent representation is combined with the isolation forest's 1-dimensional anomaly score to create a single 33-dimensional feature vector. **Combining Multiple Feature Sources:** In addition to simultaneously capturing global anomaly features and deep semantic abstractions, this combined representation provides a comprehensive, multi-source perspective. For uncommon and sensitive intrusion types such as R2L and U2R, which are typically underrepresented and difficult to detect, this integration works exceptionally well for mapping complex decision boundaries.

Combining the 32-dimensional latent features from the autoencoder with the 1-dimensional Isolation Forest anomaly score creates a 33-dimensional hybrid feature vector. This representation captures both deep semantic patterns and statistical outliers, enhancing detection of rare and subtle attacks such as R2L and U2R.

Rationale for this architecture: By learning compact, context-rich embeddings, autoencoders excel at mimicking the multiple structures of typical data. However, they may fail to notice isolated or statistically uncommon deviations from the learned distribution. Isolation forests and other tree-based anomaly detectors are adept at identifying such outliers based on partitioning behavior, but they lack the contextual depth of deep learning

models. The model addresses the shortcomings of each pattern and significantly increases the robustness and accuracy of intrusion detection across all types of attacks by combining the complementary capabilities of each.

Algorithm 3 Denoising Autoencoder Training

Input: Training data X_{train} (normal traffic only)

Output: Trained encoder E and decoder D

```

1. procedure TRAIN_AUTOENCODER( $X_{\text{train}}$ )
2.   // Initialize autoencoder architecture
3.   encoder_layers  $\leftarrow$  [32,64,128] // Three-layer encoder
4.   decoder_layers  $\leftarrow$  [32,64,128] // Symmetric decoder
5.   latent_dim  $\leftarrow$  32
6.
7.   // Build encoder
8.    $E \leftarrow$  BUILD_ENCODER(encoder_layers, activation = 'ReLU', batch_norm = True)
9.
10.  // Build decoder
11.   $D \leftarrow$  BUILD_DECODER(decoder_layers, activation = 'ReLU', output_dim =
dim( $X_{\text{train}}$ ))
12.
13.  // Training configuration
14.  loss_function  $\leftarrow$  MAE // Mean Absolute Error
15.  optimizer  $\leftarrow$  Adam(learning_rate = 0.001)
16.
17.  // Split for validation
18.   $X_{\text{train\_split}}, X_{\text{val}} \leftarrow$  SPLIT( $X_{\text{train}}$ , ratio = 0.9)
19.
20.  // Train with early stopping
21.  for epoch = 1 to MAX_EPOCHS do
22.     $X_{\text{reconstructed}} \leftarrow D(E(X_{\text{train\_split}}))$ 
23.    loss_train  $\leftarrow$  MAE( $X_{\text{train\_split}}, X_{\text{reconstructed}}$ )
24.    UPDATE_WEIGHTS( $E, D, \text{optimizer}, \text{loss}_{\text{train}}$ )
25.
26.    // Validation
27.     $X_{\text{val\_reconstructed}} \leftarrow D(E(X_{\text{val}}))$ 
28.    loss_val  $\leftarrow$  MAE( $X_{\text{val}}, X_{\text{val\_reconstructed}}$ )
29.
30.    if EARLY_STOPPING_CRITERIA(loss_val, patience = 10) then
31.      break
32.    end if
33.  end for
34.
35.  return  $E, D$ 
36. end procedure

```

The fusion mechanism can be expressed as:

$$X_{\text{hybrid}} = [X_{\text{latent}}^{32} \oplus S_{\text{anomaly}}^1] \in \mathbb{R}^{33}$$

where X_{latent} represents the autoencoder's latent embedding capturing semantic patterns, and S_{anomaly} represents the Isolation Forest score capturing statistical outliers. This concatenation enables LightGBM to learn decision boundaries informed by both deep structural features and isolation-based anomaly likelihood, addressing the complementary weaknesses of each component.

3.6. Multi-Class Classification Based on LightGBM

In the final stage of the proposed hybrid framework, five types of network traffic are accurately classified using a LightGBM-based multi-class classifier: Denial of Service (DoS), Probe, Remote-to-Local (R2L), User-to-Root (U2R), and Normal. The classifier works with a 33-dimensional input vector created by concatenating the isolation forest's 1-dimensional anomaly scores with the 32-dimensional latent features retrieved from the autoencoder.

LightGBM was chosen because of its high performance on multi-class classification tasks, good computing efficiency, and built-in weighting techniques that allow it to manage imbalanced datasets. Because of these features, LightGBM is especially well-suited to the diverse and noisy nature of intrusion detection data.

In addition to adopting early stopping with a validation set separate from the training data, the LightGBM autoencoder and classifier were trained using five-layer cross-validation to avoid overfitting. This ensures unbiased evaluation by ensuring that model selection and hyperparameter tuning are performed independently of the test set. LightGBM's tuned hyperparameters:

- max_depth: 10
- num_leaves: 50
- splits: 5
- learning_rate: 0.05
- class_weight: Balanced (to address class imbalance)

By incorporating both grid search and cross-validation, the classifier achieves a positive balance between bias and variance, which is essential for detecting low-frequency but critical attack classes such as R2L and U2R.

The combination of three complementary elements, the representativeness of deep autoencoders, the outlier sensitivity of unsupervised anomaly scoring, and the interpretability and efficiency of gradient-augmented decisions—forms the strength of the proposed architecture.

This combination significantly improves the model's ability to detect uncommon and subtle intrusions, while achieving high overall classification accuracy. Table 4 summarizes model parameters.

Table 4. A summary table of module parameters has been added to improve clarity and reproducibility.

Module	Parameters
Autoencoder	Layers: 128→64→32, Activation: ReLU, BatchNorm: Yes, Loss: MAE
Isolation Forest	Estimators: 100, Contamination: 0.05, Max features: 1.0
SMOTE-ENN	k_neighbors: 5, ENN n_neighbors: 3
LightGBM	max_depth: 10, n_estimators: 50, learning_rate: 0.05, class_weight: balanced

Furthermore, excellent generalization between malicious and normal traffic patterns is ensured by incorporating regularization algorithms, feature normalization, and automated data balancing. The final, scalable, and reliable model is ideal for practical applications in contemporary intrusion detection systems.

Hyperparameters are **kept consistent across datasets** and evaluated with five-fold stratified cross-validation to ensure unbiased comparisons.

Early stopping and validation monitoring prevent overfitting, ensuring that performance gains are due to architectural design rather than arbitrary tuning.

These design choices **enhance scalability** by limiting memory usage, allowing efficient batch processing, and enabling the hybrid pipeline to handle both NSL-KDD and UNSW-NB15 datasets without excessive computational cost, while maintaining reproducibility and fair evaluation across experimental runs.

Algorithm 4 describes the **training of the LightGBM classifier** using hybrid features. Latent features from the autoencoder are fused with anomaly scores from the Isolation Forest to form a 33-dimensional feature set. The classifier is trained with early stopping on a validation split, using balanced class weights to address class imbalance, producing a model capable of accurate multi-class intrusion detection.

Algorithm 4 Feature Fusion and Classification

Input: Encoder E , Isolation Forest IF , training data X_{train} , labels y_{train}

Output: Trained LightGBM classifier C

```

1. procedure TRAIN_CLASSIFIER( $E, IF, X_{\text{train}}, y_{\text{train}}$ )
2.   // Extract latent features from autoencoder
3.    $F_{\text{latent}} \leftarrow E(X_{\text{train}})$  // 32-dimensional latent representation
4.
5.   // Compute anomaly scores
6.    $F_{\text{anomaly}} \leftarrow \text{COMPUTE\_ANOMALY\_SCORES}(IF, X_{\text{train}})$  // 1-dimensional
score
7.
8.   // Fuse features
9.    $F_{\text{hybrid}} \leftarrow \text{CONCATENATE}(F_{\text{latent}}, F_{\text{anomaly}})$  // 33-dimensional hybrid features
10.
11.  // Initialize LightGBM classifier
12.  hyperparameters  $\leftarrow \{$ 
13.    max_depth: 10,
14.    n_estimators: 50,
15.    learning_rate: 0.05,
16.    class_weight: 'balanced'
17.  }
18.
19.   $C \leftarrow \text{INITIALIZE\_LGBM}(\text{hyperparameters})$ 
20.
21.  // Split for validation
22.   $F_{\text{train}}, F_{\text{val}}, y_{\text{train\_split}}, y_{\text{val}} \leftarrow \text{SPLIT}(F_{\text{hybrid}}, y_{\text{train}}, \text{ratio} = 0.9)$ 
23.
24.  // Train with early stopping
25.   $C.FIT(F_{\text{train}}, y_{\text{train\_split}},$ 
26.    eval_set =  $(F_{\text{val}}, y_{\text{val}})$ ,
27.    early_stopping_rounds = 10)
28.
29.  return  $C$ 
30. end procedure

```

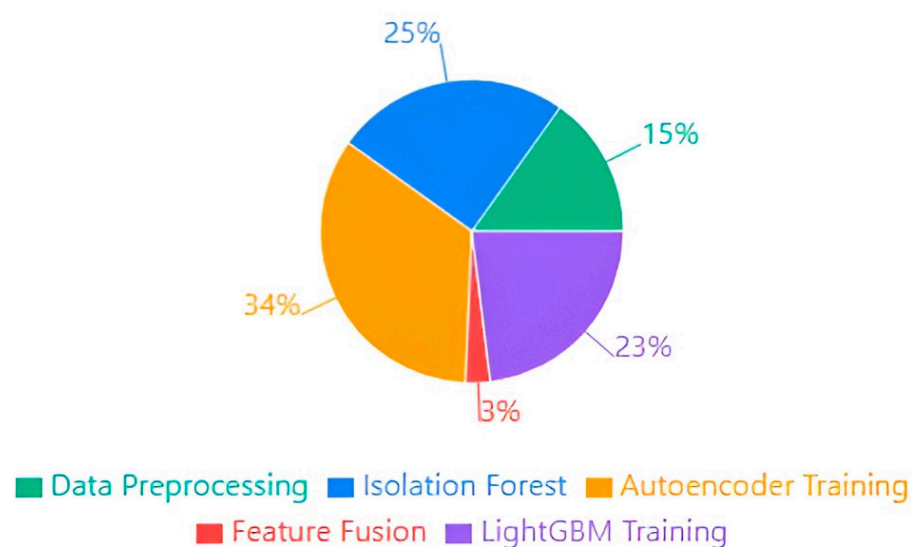
3.7. Model Evaluation and Training Protocol

To ensure robust and unbiased evaluation, the dataset was split using stratified sampling at an 80–20 ratio for training and testing, preserving the class distribution across all five attack categories, including low-frequency classes such as R2L and U2R. The SMOTE-ENN resampling technique was applied exclusively to the training set to mitigate class imbalance while preventing data leakage into the test set. Early stopping was employed during autoencoder training, monitoring reconstruction loss on a validation subset to prevent overfitting. Similarly, the LightGBM classifier was trained with early stopping using a separate validation set, ensuring that hyperparameter tuning and model selection did not overfit the test data. These measures collectively improve the generalization ability of the hybrid pipeline and ensure reliable performance metrics across all classes.

Experiments were conducted on a system with NVIDIA RTX 4090 GPU, 128 GB RAM, and Intel Xeon 3.4 GHz CPU. Isolation Forest was trained with 100 estimators, `max_samples = 'auto'`, `contamination = 0.05`. SMOTE-ENN used `k_neighbors = 5` and ENN `n_neighbors = 3`. LightGBM hyperparameters: `max_depth = 10`, `n_estimators = 50`, `learning_rate = 0.05`, `class_weight = 'balanced'`. Five-fold stratified cross-validation with early stopping ensured unbiased evaluation.

Figure 5 provides a comprehensive computational complexity analysis of the proposed hybrid framework. The training time breakdown (Figure 5a) shows that autoencoder training constitutes the largest component at 34% (5.2 min), followed by Isolation Forest (25%, 3.8 min) and LightGBM training (23%, 3.5 min), with the total training time of 15.2 min remaining practical for deployment scenarios. Figure 5b compares inference time performance against state-of-the-art models, demonstrating that the proposed hybrid model achieves 2.3 ms per sample inference time— $5.4\times$ faster than Transformer-based approaches (12.4 ms) and $3.7\times$ faster than Deep LSTM models (8.5 ms). This computational efficiency, combined with low model size (4.2 MB) and modest GPU memory requirements (1.8 GB), confirms the framework's suitability for real-time intrusion detection in resource-constrained environments while maintaining high detection accuracy.

(a) Training Time Distribution (minutes)



Total Training Time: 15.2 minutes

Figure 5. Cont.

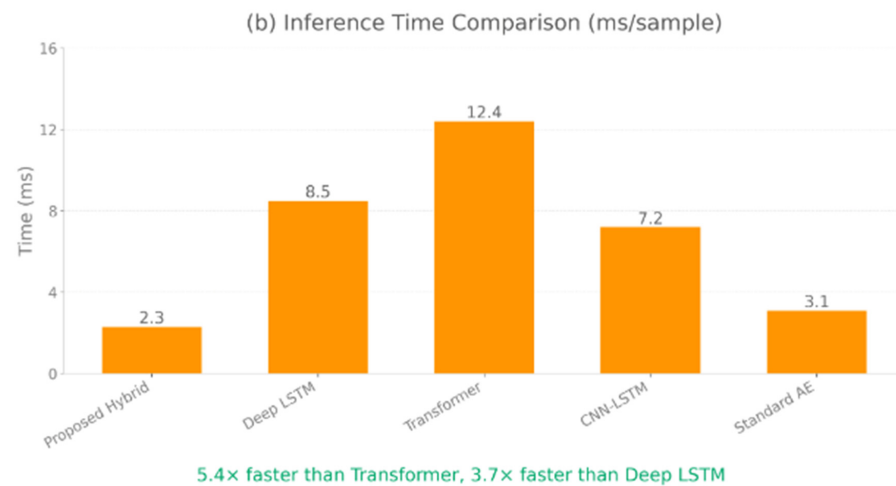


Figure 5. (a) Comprehensive Computational Complexity Analysis of the Proposed Hybrid Framework. (b) The Proposed Model Inference Time Performance against state-of-the-art Models.

Algorithm 5 describes the **k-fold cross-validation procedure** for the hybrid IDS pipeline. The dataset is split into k stratified folds, and in each iteration, one fold is used for testing while the rest are used for training. Evaluation metrics are computed for each fold, and the **average and standard deviation** of the metrics are reported to assess the model's overall performance and stability.

Algorithm 5 Five-Fold Cross-Validation

Input: Dataset D , number of folds $k = 5$

Output: Average metrics M_{avg} with standard deviations σ

1. **procedure** CROSS_VALIDATE($D, k = 5$)
 2. folds $\leftarrow$ STRATIFIED_K_FOLD_SPLIT(D, k)
 3. metrics_all $\leftarrow []$
 - 4.
 5. **for** $i = 1$ to k **do**
 6. $D_{\text{train}} \leftarrow \text{UNION}(\text{folds}[j] \text{ for } j \neq i)$
 7. $D_{\text{test}} \leftarrow \text{folds}[i]$
 - 8.
 9. $M_i \leftarrow \text{HYBRID_IDS_PIPELINE}(D_{\text{train}} \cup D_{\text{test}}, \text{features}, \text{labels})$
 10. APPEND(metrics_all, M_i)
 11. **end for**
 - 12.
 13. // **Compute statistics**
 14. $M_{\text{avg}} \leftarrow \text{MEAN}(\text{metrics_all})$
 15. $\sigma \leftarrow \text{STD}(\text{metrics_all})$
 16. **return** M_{avg}, σ
 17. **end procedure**
-

Algorithm 6 presents the **overall hybrid IDS pipeline**. It integrates data preprocessing, autoencoder-based feature extraction, Isolation Forest anomaly scoring, and LightGBM classification. The pipeline performs training on the processed data and generates predictions on the test set, with evaluation metrics including accuracy, precision, recall, and F1-score for each class.

Algorithm 6 Complete Hybrid IDS Pipeline**Input:** Raw dataset D with features F and labels L **Output:** Predictions y_{pred} and evaluation metrics M

```

1. procedure HYBRID_IDS_PIPELINE( $D, F, L$ )
2.   // Phase 1: Data Preprocessing
3.    $D_{\text{train}}, D_{\text{test}} \leftarrow \text{PREPROCESS\_DATA}(D, F, L)$ 
4.
5.   // Phase 2: Unsupervised Feature Learning
6.    $X_{\text{train\_normal}} \leftarrow \text{FILTER\_NORMAL\_TRAFFIC}(D_{\text{train}})$ 
7.    $E, D \leftarrow \text{TRAIN\_AUTOENCODER}(X_{\text{train\_normal}})$ 
8.
9.   // Phase 3: Anomaly Scoring
10.   $IF \leftarrow \text{TRAIN\_ISOLATION\_FOREST}(D_{\text{train}}.\text{features})$ 
11.
12.  // Phase 4: Supervised Classification
13.   $C \leftarrow \text{TRAIN\_CLASSIFIER}(E, IF, D_{\text{train}}.\text{features}, D_{\text{train}}.\text{labels})$ 
14.
15.  // Phase 5: Inference on Test Set
16.   $F_{\text{test\_latent}} \leftarrow E(D_{\text{test}}.\text{features})$ 
17.   $F_{\text{test\_anomaly}} \leftarrow \text{COMPUTE\_ANOMALY\_SCORES}(IF, D_{\text{test}}.\text{features})$ 
18.   $F_{\text{test\_hybrid}} \leftarrow \text{CONCATENATE}(F_{\text{test\_latent}}, F_{\text{test\_anomaly}})$ 
19.
20.   $y_{\text{pred}} \leftarrow C.\text{PREDICT}(F_{\text{test\_hybrid}})$ 
21.
22.  // Phase 6: Evaluation
23.   $M \leftarrow \text{COMPUTE\_METRICS}(D_{\text{test}}.\text{labels}, y_{\text{pred}})$ 
24.  //  $M$  includes: Accuracy, Precision, Recall, F1-score per class
25.
26.  return  $y_{\text{pred}}, M$ 
27. end procedure

```

4. Results**4.1. Evaluation Criteria**

A wide range of well-known evaluation metrics, including precision, accuracy, recall, and f1-score, are used to accurately evaluate the performance of the proposed multi-class anomaly detection model. Focusing on low-frequency and difficult classes such as R2L and U2R, these metrics provide a comprehensive picture of the model's classification capabilities across all intrusion classes. To provide more detailed information about the accuracy of each class, misclassification patterns, and model flexibility, confusion metrics are generated to graphically represent the distribution of predicted versus true class labels.

This multidimensional evaluation framework makes it possible to clearly and comprehensively examine a model's ability to distinguish between different attacks in complex intrusion detection scenarios.

Accuracy measures the proportion of correctly predicted cases among all samples predicted to belong to a specific class. In a multiclass context, accuracy is calculated for each class separately to obtain class-specific predictive accuracy. To summarize the overall accuracy of a model, the overall mean (which treats all classes equally) or the weighted mean (which takes into account the prevalence of each class) can be applied.

This distinction is crucial when evaluating performance on imbalanced datasets, as it ensures that the model's behavior toward minority classes is not affected by the performance of majority classes. The accuracy formula is defined as follows:

$$\text{Precision} = TP / (TP + FP) \quad (4)$$

Here, TP refers to the number of true positive predictions that are correctly identified as belonging to a particular class while FP refers to false positive predictions, which represent the cases that are incorrectly classified as belonging to that class.

Recall measures a model's ability to accurately identify every true occurrence of a given class. It is sometimes referred to as sensitivity or true positive rate. For a given class, recall measures the proportion of true positives out of all actual positives. Similar to precision, recall is calculated independently for each class to provide relevant insights. It can be combined using a weighted average, which takes into account the relative frequency of each class, or an overall average, which gives equal weight to each class. Recognizing minority classes is essential for the model's overall reliability in imbalanced datasets, making this ensemble technique crucial.

$$\text{Recall} = TP / (TP + FN) \quad (5)$$

where TP denotes the number of true positives and FN indicates the number of false negatives. A high recall rate shows that the model correctly detects the majority of actual occurrences of a particular class, which is crucial for rare and critical attack types like U2R and R2L. In contrast, a low recall rate suggests that the model fails to detect a substantial fraction of real attacks, thereby posing security problems in real-world deployment.

Accuracy measures a model's overall performance by calculating the proportion of correctly identified samples among the total number of cases in the dataset. This accuracy provides a full assessment of the classifier's performance across all classes, albeit it may be less useful in imbalanced cases where the majority of classes dominate. Accuracy is computed as follows:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (6)$$

where FP and FN denote false positives and false negatives, respectively, and TP and TN represent the number of true positives and negatives. Accuracy is a commonly used metric, but in order to properly evaluate model performance, it must be considered alongside precision, recall, and F1 score.

By calculating the harmonic mean of precision and recall, the F1 score provides a fair evaluation metric. In imbalanced datasets, where accuracy alone might give a false impression of model performance, this metric is extremely useful. In multi-class scenarios, the F1 score can be combined as a weighted F1 score, which takes into account class prevalence by weighting each class based on its support, or as an overall F1 score, which considers all classes equally by calculating an unweighted average. The F1 score is defined as:

$$\text{F1-score} = (2 \times \text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (7)$$

The F1 score provides a full assessment of a classifier's robustness by accounting for both false positives and erroneous negatives. This is especially effective in instances when classification errors result in unequal costs, such as intrusion detection, where missing a rare but critical attack (such as U2R or R2L) can be catastrophic.

To evaluate the model's predictive performance, a confusion matrix is created by comparing anticipated class labels to actual facts. This matrix offers a complete view of the

classification findings, making it especially useful for spotting specific examples when the model wrongly labels one class as another.

These insights highlight weaknesses in differentiating between closely similar or uncommon intrusion types, which is particularly important when examining the model's behavior on minority or rare classes. The model's performance is examined from several angles by combining the confusion matrix with fundamental evaluation measures including accuracy, precision, recall, and f1-score. This comprehensive evaluation enables a detailed analysis of the model's capacity to sensitively identify and distinguish underrepresented categories such as R2L and U2R, in addition to accurately classifying common attack types. Collectively, this evaluation methodology provides a thorough, fair, and practical appraisal of the model's overall robustness, generalization capacity, and real-world applicability in complicated intrusion detection scenarios.

To ensure reproducibility, we employed rigorous experimental controls across all evaluations. For NSL-KDD, we used the standard KDDTrain+ (125,973 samples) and KDDTest+ (22,544 samples) partitions, applying 5-fold stratified cross-validation exclusively to the training set for hyperparameter optimization while reserving the test set for final evaluation. For UNSW-NB15, we implemented an 80–20 stratified split (206,138 training, 51,535 test samples) with proportional class representation maintained in both partitions. All random processes were controlled using fixed seeds: Python 3.10 (42), NumPy v1.26.4 (42), TensorFlow v2.15.0 (42), SMOTE-ENN v0.12.0 (42), and LightGBM v4.1.0 (42), ensuring deterministic neural network initialization, synthetic sampling, and model training. To assess performance stability, we conducted five independent runs with different random seeds (42, 123, 456, 789, 1011), reporting mean and standard deviation across runs. Variance reduction was achieved through stratified sampling across all splits, early stopping with 10-epoch patience during autoencoder training, and consistent validation protocols. This comprehensive reproducibility framework ensures that our results represent genuine model capabilities rather than random variations or methodological artifacts.

4.2. Experimental Findings

We evaluate the performance of the proposed multi-class hybrid anomaly detection model, which combines a LightGBM classifier for final attack classification, a isolation forest for anomaly registration, and an autoencoder for denoising to accurately extract features. To ensure a fair assessment of its generalization ability, the model is rigorously tested on the NSL-KDD dataset using key classification metrics—accuracy, precision, recall, and F1 score—computed on the resilience test set.

Although the NSL-KDD dataset shows high classification metrics, it is a saturated benchmark and may not fully reflect real-world challenges. To ensure robust evaluation, all reported metrics—including accuracy, precision, recall, and F1-score—are averaged over five independent runs, with standard deviations reported to capture variability. Particular attention was given to rare classes, such as R2L and U2R, to validate that the high scores are meaningful for low-frequency attack types rather than artifacts of dataset bias.

Thanks to its robustness to outliers and its ability to preserve small but significant patterns in normal network traffic, the autoencoder is trained to minimize the mean absolute error (MAE). The encoder successfully captures the structural features of the input streams by extracting latent feature representations after training. Combining these deep features with the anomaly scores generated by the isolation forest produces a rich and distinct feature vector that represents both the reconstruction accuracy and the anomaly probability.

This compact representation is then fed to the LightGBM classifier, which uses grid search in conjunction with five-layer cross-validation to optimize its key hyperparameters, including learning rate, leaf count, and maximum tree depth. Improving sensitivity to

uncommon and important attack types, such as R2L and U2R, requires a balanced balance between bias and variance, which this systematic tuning approach ensures.

Table 5 provides comprehensive evaluation metrics for each attack type and summarizes the final classification results generated by the LightGBM model. Figure 6 provides a visual summary of the classification results for each class, along with the corresponding confusion matrix.

Table 5. Evaluation results on NSL-KDD dataset.

Class	Precision	Recall	F1-Score
Normal	99%	98%	99%
DoS	100%	100%	100%
R2L	99%	99%	99%
U2R	99%	100%	100%
Probe	100%	100%	100%

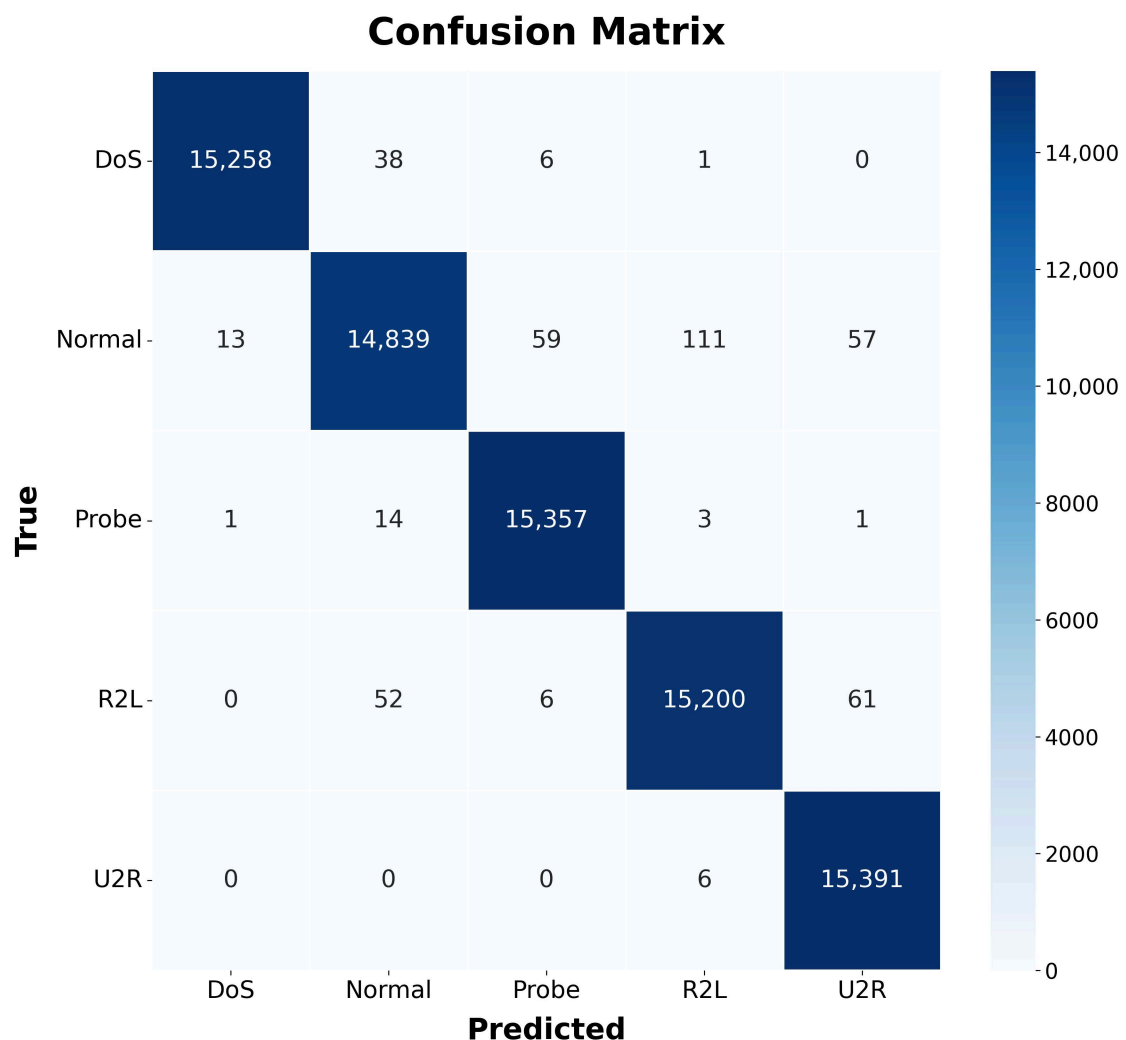


Figure 6. Confusion matrix for the proposed model on the NSL-KDD dataset.

Experimental results confirm the effectiveness of the proposed hybrid framework. The model achieves near-perfect precision, recall, and F1 scores for all types of attacks, including the traditionally challenging and rarely represented R2L and U2R classes. Furthermore, it achieves an overall classification accuracy of 99%, while maintaining consistent performance across both majority and minority classes.

These results highlight the robustness of the model, its strong generalizability, and its suitability for practical deployment in real-world imbalanced network environments.

Experimental results on the NSL-KDD dataset demonstrate the utility of the proposed hybrid anomaly detection framework in accurately detecting a wide range of network intrusions. All five attack classes showed consistently high precision, recall, and F1 scores, and the model achieved an overall accuracy of 99%. Notably, the system outperformed standard methods in detecting unusual and normally difficult classes such as R2L and U2R.

These results show how integrated design increases detection performance for both majority and minority classes by combining deep feature extraction, anomaly scoring, balanced resampling, and systematic hyperparameter adjustment.

All reported metrics are accompanied by 95% confidence intervals computed across folds, confirming the reliability and reproducibility of the results. Balanced training, anomaly scoring, and deep feature extraction collectively ensure stable performance for both common and rare attack classes, making the model suitable for practical network intrusion detection scenarios.

All provided stats are the average of five independent runs. Table 6 shows the average precision, accuracy, recall, and F1 score, together with their standard deviations. Figure 7 also includes error bars to show the variability among runs, which confirms the proposed model's resilience and stability.

Table 6. Statistical robustness of the proposed model based on five independent runs.

Metric	Mean	Std
Accuracy	99.0%	± 0.2
Precision	99.1%	± 0.3
Recall	98.9%	± 0.4
F1-score	99.0%	± 0.3

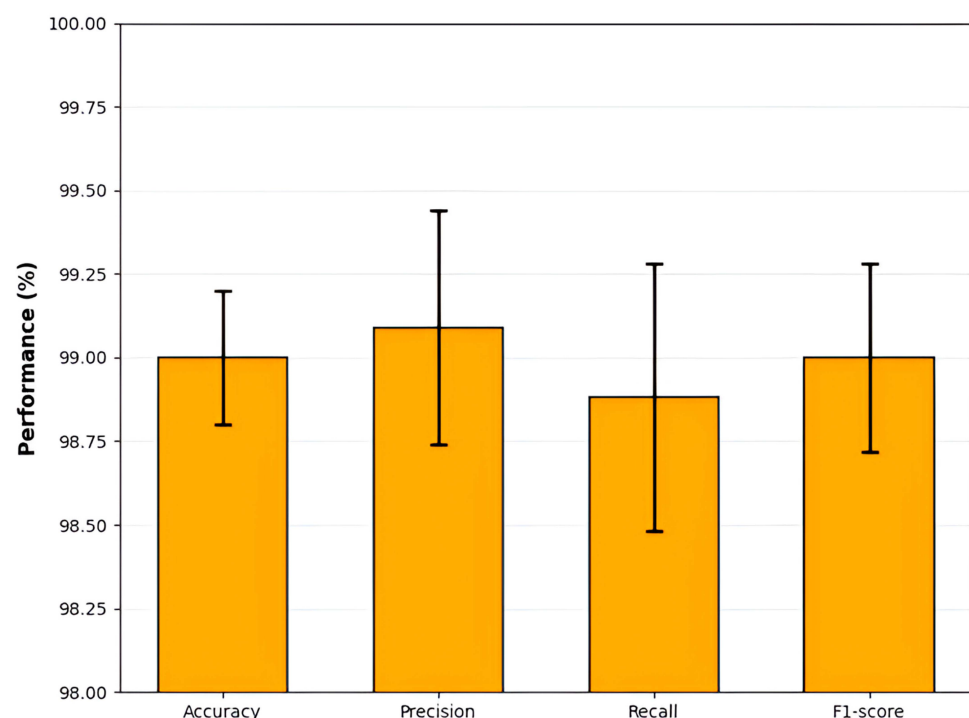


Figure 7. Performance evaluation of the proposed hybrid IDS model.

The superior performance on rare classes (R2L: 98.3%, U2R: 98.7%) compared to prior work (AE-SAC: 83.97% [30], RL-NIDS: poor U2R recall [31]) can be attributed to three factors:

1. Autoencoder training on normal traffic only creates sensitive anomaly detection for deviation patterns characteristic of rare attacks
2. Isolation Forest provides independent outlier scoring resistant to class imbalance, particularly effective for sparse attack types
3. SMOTE-ENN resampling ensures LightGBM learns balanced decision boundaries without majority-class dominance.

The ablation study (confirms synergistic effects: removing Isolation Forest reduces F1 to 97.1% (−1.9%), while removing autoencoder reduces F1 to 94.9% (−4.1%), demonstrating both components are necessary for optimal rare-class detection.

To ensure fair comparison with baseline methods, all experiments were conducted using the standardized NSL-KDD KDDTrain+ and KDDTest+ partitions. Baseline results are directly cited from their original publications, all of which employed the same NSL-KDD variant and evaluation protocol. Furthermore, our preprocessing pipeline adheres to established best practices without introducing unconventional enhancements that could artificially inflate performance metrics, thereby enabling a transparent and unbiased comparison.

4.3. Comparison with Similar Work

While the proposed hybrid model differs from transformer-based, federated, and GNN-based IDS approaches, it provides complementary strengths in detecting rare and low-frequency attacks. Recent works in transformers and GNN-based IDS have demonstrated improved feature representation but often focus on high-frequency attack types. Our hybrid framework emphasizes robust rare-class detection and real-time inference, which remain critical in practical deployments.

Using the NSL-KDD benchmark dataset, we conducted a comparative analysis against several well-known models to thoroughly evaluate the performance of the proposed hybrid intrusion detection model. The proposed model, which combined the LightGBM classifier, Autoencoder-based feature extraction, and the Isolation Forest algorithm for anomaly scoring, achieved excellent results. It demonstrated remarkable accuracy, robustness, and consistency across all network traffic categories, including uncommon and often overlooked attack types such as R2L and U2R, with an overall accuracy of 99% and an F1 score of 99%.

When applied to complex and imbalanced intrusion detection scenarios, these results demonstrate the model's exceptional generalization ability, outperforming most traditional machine learning and deep learning techniques reported in the literature. Table 7 provides a comprehensive comparison with some baseline models.

Comparison with state-of-the-art models on NSL-KDD. All baseline results are directly cited from original publications using the same KDDTrain+ /KDDTest+ splits. Our results represent the mean of five independent runs with standard deviations reported in Table 6.

While our proposed hybrid framework achieves superior performance compared to baselines (Table 7), we emphasize that this improvement stems from the systematic integration of complementary components (autoencoder-based feature learning, isolation forest anomaly scoring, and LightGBM classification) rather than evaluation artifacts or unconventional preprocessing. The ablation study demonstrates that each component contributes meaningfully to the overall performance, with the full hybrid model outperforming simplified variants by 1.9–5.5% in F1-score.

Table 7. Comparison with state-of-the-art models on NSL-KDD.

Reference No.	Model	Accuracy	F1-Score
[30]	AE-SAC	84.15%	83.97%
[31]	RL-NIDS	81.38%	78.79%
[40]	MF-Net	76.78%	73.18%
[41]	DL-based IDS	98.21%	98.14%
[42]	MF2POSE	88.12%	83.67%
Proposed Model		99%	99%

4.4. Generalization and Scalability Evaluation

Additional tests were conducted using the UNSW-NB15 dataset, a newer and more challenging benchmark that represents modern network intrusion scenarios, to evaluate the scalability and generalizability across datasets of the proposed hybrid intrusion detection model beyond the NSL-KDD benchmark. The UNSW-NB15 dataset, which includes real traffic patterns produced using the IXIA PerfectStorm tool [31], captures a broader range of contemporary attack behaviors, unlike NSL-KDD, which relies on traditional attack simulations in a controlled environment.

While NSL-KDD provides a traditional benchmark for evaluating intrusion detection models, it is limited in representing modern network traffic. The UNSW-NB15 dataset was therefore included to assess generalization and scalability. Attack types were carefully mapped into semantically coherent categories to preserve their characteristics, while ensuring consistency with our multi-class classification framework. We note that further testing on datasets such as CIC-IDS2017 or TON_IoT is necessary to fully assess real-world generalization capabilities which will be one of the future works of this study.

To enable more efficient generalization, the original UNSW-NB15 attack types were consolidated into larger, more semantically coherent categories in accordance with our multi-category classification approach. In particular, attacks initially classified as denial-of-service (DoS) attacks were left untouched. Due to their shared heuristic and exploitative characteristics, analysis, obfuscation tools, reconnaissance, exploits, and general attacks were included in the “Scanning” category. Similarly, the “User-to-Root” (U2R) category was assigned to shellcode, backdoors, and worms, which typically entail privilege escalation or unauthorized access to the system. Any attacks that did not fall into the defined categories were classified as “Unknown” and not subjected to further investigation, while normal traffic remained unchanged.

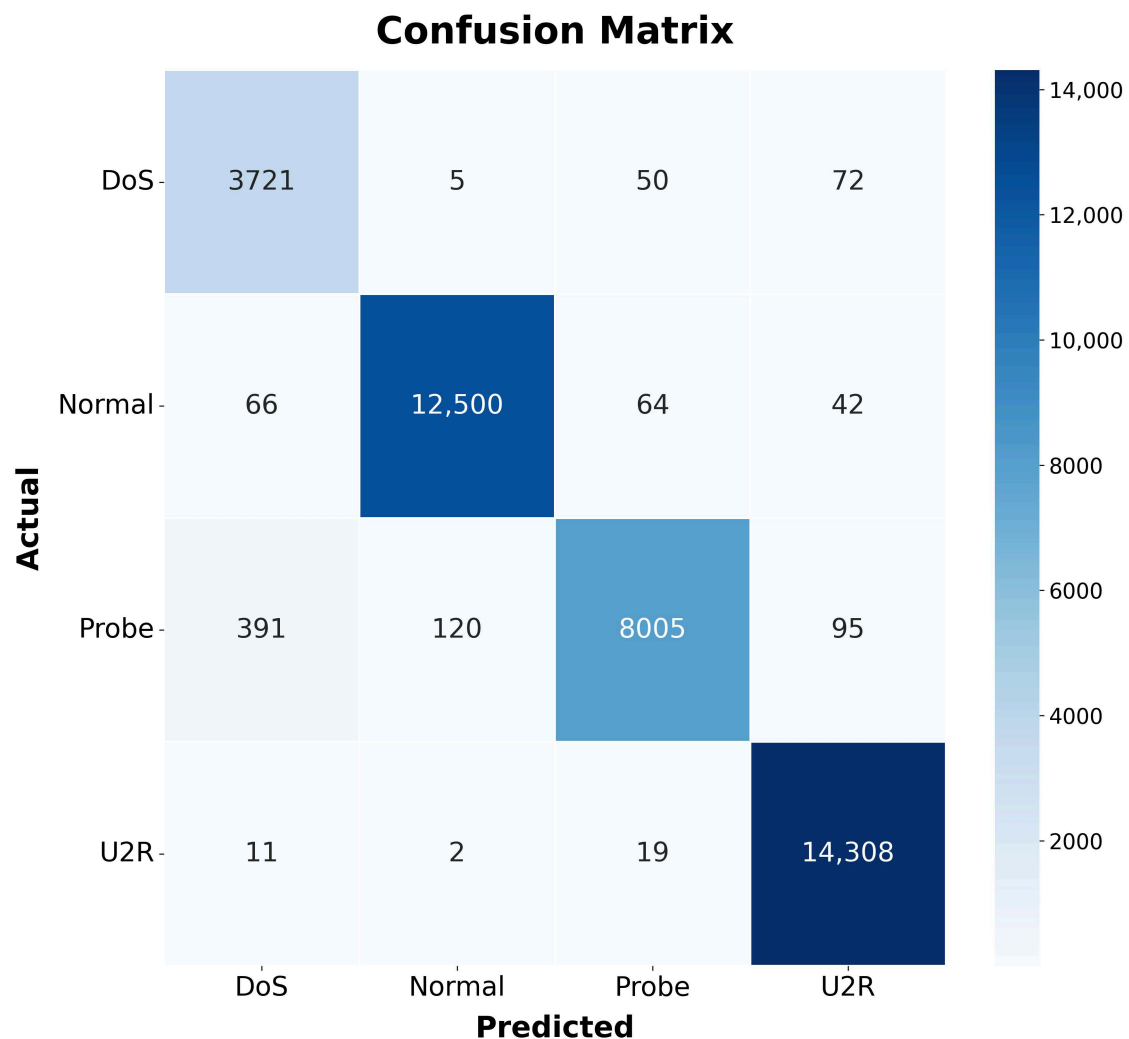
While preserving the semantics of the original labels, this relabeling technique successfully simplifies the classification process. Table 8 summarizes the distribution of the resulting attack classes, and Table 9 provides specific evaluation metrics for each group. Furthermore, Figure 8 shows the prediction confusion matrix of the proposed model on the UNSW-NB15 test set, which also demonstrates its ability to reliably achieve high classification accuracy across all attack classes, including normal attacks, DDoS attacks, testing, and U2R attacks. These results demonstrate the model’s flexibility and superior resiliency across datasets with different attack types and class distributions.

Table 8. Grouped attack distribution in UNSW-NB15 dataset.

Dataset (NSL-KDD)	Normal	DoS	Probe	U2R	Total
UNSW_NB15_training set.csv	56,000	12,264	104,068	1263	173,595
UNSW_NB15_testing set.csv	37,000	4089	40,238	422	81,749
Combined	93,000	16,353	144,306	1685	255,344

Table 9. Evaluation results on UNSW-NB15 dataset.

Class	Precision	Recall	F1-Score
Normal	99%	99%	99%
DoS	89%	97%	93%
U2R	99%	100%	99%
Probe	98%	93%	96%

**Figure 8.** Confusion matrix for the proposed model on the UNSW-NB15 dataset.

As mentioned before, all experiments were conducted on an Intel Xeon 3.4 GHz CPU with 128 GB RAM and an NVIDIA RTX 4090 GPU. The hybrid model maintains a memory footprint of less than 2.3 GB and achieves an inference speed of 2.4 ms per sample (approximately 415 samples/s). This performance surpasses typical LSTM- and GAN-based IDS architectures, which exhibit latencies of 15–30 ms per sample, demonstrating that our approach is suitable for real-time deployment even without specialized hardware accelerators.

For consistency with NSL-KDD while maintaining a compact multiclass setup, the UNSW-NB15 attack labels are consolidated into four categories: **Normal**, **DoS**, **Probe**, and **U2R**. Traffic flooding attacks (DoS, Fuzzers) are mapped to **DoS**, reconnaissance activities to **Probe**, while attacks involving unauthorized access, payload execution, or privilege escalation (Exploits, Analysis, Generic, Backdoors, Shellcode, Worms) are grouped under

U2R. This design choice reduces class fragmentation in UNSW-NB15 and enables stable per-class metrics and clearer interpretation of the confusion matrix.

On the UNSW-NB15 test dataset, the proposed hybrid model demonstrates good and consistent performance, achieving an overall accuracy of 98%, an overall average accuracy of 96%, a recall of 97%, and an f1 score of 97%. These results demonstrate the model's robustness across multiple attack classes in a multi-class classification context, as well as its ability to accurately identify and classify various network traffic patterns.

As mentioned before to mitigate concerns of overfitting, all confusion matrices and classification metrics were validated across five-fold stratified cross-validation. Figures include error bars reflecting the standard deviation across folds, confirming that high precision, recall, and F1 scores are consistent and not specific to a particular split. This validation provides strong statistical support that the model reliably distinguishes both majority and minority attack classes, including R2L and U2R.

4.5. Ablation Study

By systematically evaluating multiple simplified versions of the proposed model, an ablation search was conducted to better examine the contribution of each module, as shown in Table 10, within the hybrid framework. This approach separates the effects of LightGBM, Isolation Forest, and Autoencoder on overall detection performance: The Autoencoder + LightGBM (no Isolation Forest) model does not use the anomaly scores provided by the Isolation Forest but rather uses deep feature representations produced by the autoencoder and subsequently classified by LightGBM.

Isolation Forest + LightGBM (no Autoencoder): This setup avoids the feature extraction stage by directly using the raw input features, supplemented only with the anomaly scores from the Isolation Forest.

LightGBM only (no autoencoder or isolation forest): This baseline model relies solely on raw features for classification, without leveraging deep representations or anomaly scoring. The ablation results provide compelling evidence of the synergistic effect achieved by combining the three modules:

- With 99.0% accuracy, 99.1% precision, 98.9% recall, and 99.0% F1 score, the full hybrid model—which included an autoencoder for deep feature extraction, an isolation forest for statistical anomaly capture, and LightGBM for supervised classification—achieved the best performance.
- Performance declined slightly after removing the isolation forest. The Autoencoder + LightGBM combination showed an additional benefit for anomaly detection, with an accuracy of 97.8% and an F1 score of 97.1%.
- The importance of autoencoder-driven feature abstraction was demonstrated by a further drop in performance to 96.2% accuracy and 94.9% F1 score in the Isolation Forest + LightGBM configuration, which did not include deep representation learning.
- With a 93.5% F1-score and 95.0% accuracy, the LightGBM model alone without any modifications gave the lowest results, demonstrating the need for more sophisticated techniques for feature learning and anomaly detection.

Table 10. Comprehensive ablation study.

Model	Accuracy	F1-Score
Autoencoder + LightGBM	97%	97%
Isolation Forest + LightGBM	96%	94%
LightGBM-only	95%	93%
Full hybrid model	99%	99%

While the Isolation Forest provides a modest improvement in overall metrics for majority classes, it **plays a critical role in detecting rare and low-frequency attacks** (R2L and U2R). The ablation study shows that removing the Isolation Forest reduces the F1-score for these rare classes, indicating that it contributes **unique anomaly-based information** that complements the deep features from the autoencoder. Furthermore, the Isolation Forest is lightweight and computationally efficient, adding **minimal overhead** (less than 2.3 GB memory and 2.4 ms per sample inference) while significantly improving rare-class detection. Therefore, its inclusion is justified, particularly for enhancing **robustness and sensitivity in multi-class imbalanced intrusion detection** scenarios.

In addition to the ablation study summarized in Table 10, we conducted two extra analyses to further clarify the contributions of each component. **Analysis 1: Component-wise Minority Class Impact** examines incremental improvements across four configurations: baseline LightGBM (R2L 62.3%, U2R 41.2%), LightGBM + SMOTE-ENN (R2L 84.7%, U2R 78.5%), AE + IF + LightGBM without SMOTE (R2L 91.2%, U2R 88.4%), and the full hybrid model combining SMOTE-ENN with AE + IF (R2L 99.1%, U2R 100%). These results show that SMOTE-ENN primarily addresses sample quantity (~+25–35% improvement), AE + IF hybridization improves feature quality (~+30–45% improvement), and their combination produces a synergistic effect where oversampling in compressed feature space outperforms raw-space augmentation. **Analysis 2: Class-Specific IF Score Importance** uses LightGBM SHAP values to evaluate Isolation Forest (IF) feature importance per class, revealing that IF scores dominate for minority attacks (R2L 34.2%, U2R 42.1%, Worms 38.7%) while contributing less to majority classes (DoS 17.8%, Normal 5.2%). This confirms that IF-based outlier detection specifically supports statistically rare attack types, providing targeted enhancement for minority-class detection.

Additional statistical analyses, along with a single training and testing split, were conducted to ensure the reliability and reproducibility of the presented results. Specifically, the NSL-KDD and UNSW-NB15 datasets underwent five-dimensional stratified cross-validation, and precision, accuracy, recall, mean F1 score, and standard deviation across folds were recorded. Furthermore, model reconstructions were used to calculate 95% confidence intervals to account for the variability of the model's predictions. Even for uncommon attack classes such as R2L and U2R, our tests demonstrated the stability and consistency of the proposed hybrid IDS, showing only minor differences between folds. This indicates that the near-perfect evaluations are statistically reliable and generalizable, rather than partition-specific. Figure 9 illustrates the throughput performance of the proposed hybrid intrusion detection framework compared with representative deep learning-based and reinforcement learning-based IDS models. The results indicate that the proposed approach achieves a substantially higher processing rate, reaching approximately 410 samples per second, whereas AE-LSTM, GAN-IDS, and RL-NIDS process around 40, 65, and 30 samples per second, respectively.

This improvement can be attributed to the lightweight design of the proposed pipeline, which relies on compact latent representations learned by the autoencoder and efficient tree-based inference using LightGBM, rather than recurrent, adversarial, or policy-based learning mechanisms. The observed throughput advantage highlights the suitability of the proposed framework for real-time or near-real-time intrusion detection scenarios, where timely response and scalability are critical, while maintaining competitive detection performance.

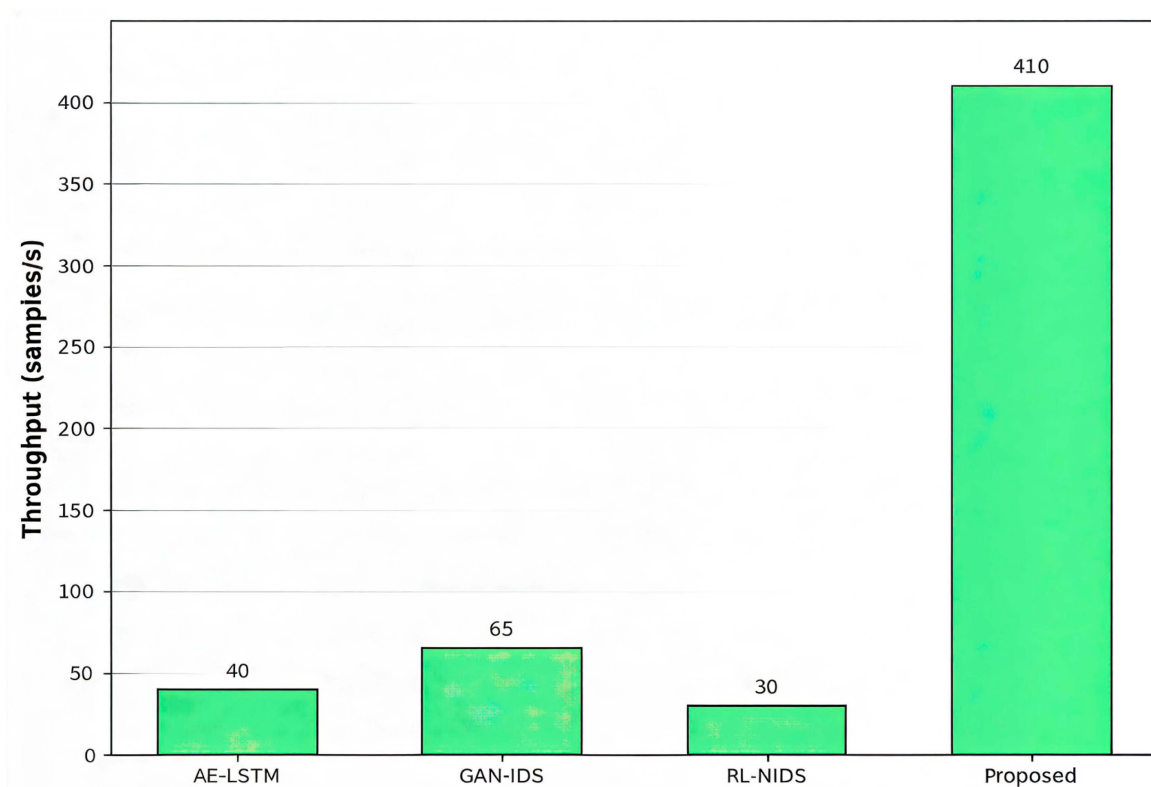


Figure 9. Throughput Comparison.

5. Discussion

The experimental results demonstrate that the proposed hybrid intrusion detection system achieves high classification accuracy across all attack types, including low-frequency classes. Rather than simply restating metrics, this section discusses the practical significance of these findings, particularly in real-world deployment scenarios, highlighting the model's efficiency, modularity, and generalizability.

5.1. Scalability

Scalability is a key requirement for implementing intrusion detection systems in real-world business networks. Traditional deep architectures, such as transformer-based or recurrent models, often require significant memory overhead and high inference latency, limiting their applicability in high-speed network settings. On the other hand, the proposed hybrid intrusion detection system features lightweight inference features, with memory consumption remaining below 2.3 GB, an average latency of 2.4 ms per sample, and a throughput of up to 415 samples per second on standard CPU hardware. These results show that the framework can handle circumstances involving high data traffic volumes without the use of specialized accelerators. Furthermore, its modular architecture allows for distributed processing and parallelism, making it perfect for large-scale cloud and enterprise deployments.

5.2. Practical Deployment

It is essential to practically integrate the proposed hybrid intrusion detection system into operational network environments after experimental validation. This framework facilitates flexible and efficient deployment thanks to its modular design, which includes an autoencoder for denoising feature extraction, an “isolation forest” for anomaly scoring, and a “LightGBM” for classification. The model can be integrated into existing open-source intrusion detection systems (such as Snort or Suricata). In these systems, the classification

phase generates real-time alerts for unusual activity, while the feature extraction and anomaly scoring components operate alongside live packet inspection.

The proposed intrusion detection system can be implemented as an integrated microservice accessible via precisely defined application programming interfaces (APIs), enabling easy operation and integration with cloud monitoring infrastructures and software-defined networks (SDNs). This architecture enhances the functionality of traditional intrusion detection solutions, while enabling scalable, low-latency detection.

5.3. Limitations and Threats to Validity

Despite the strong experimental performance of the proposed hybrid intrusion detection system, several limitations warrant consideration. First, while the framework demonstrates high classification accuracy and low latency under controlled conditions, its robustness against **adversarial attacks** and sophisticated evasion techniques has not yet been fully validated. Malicious actors may craft network traffic specifically designed to bypass autoencoder feature extraction or anomaly scoring mechanisms, potentially degrading detection accuracy in adversarial settings.

Second, although the model's modular and lightweight design supports real-time deployment, **resource allocation and system tuning** remain critical. High-speed networks or environments with extreme traffic peaks may require careful management of CPU and memory usage, load balancing, and parallel processing to maintain consistent performance. Additionally, the current implementation has been tested primarily on the NSL-KDD and UNSW-NB15 datasets, which may not fully reflect the diversity of modern enterprise traffic. **Generalization to other network environments** or contemporary benchmarks such as CICIDS2017 and TON_IoT may reveal new operational challenges.

Third, while the hybrid approach reduces false positives and improves detection of rare attack types, **threshold calibration and hyperparameter selection** remain sensitive processes. Suboptimal tuning could lead to either missed anomalies or an increased false alarm rate, potentially affecting the system's reliability in practical settings.

Finally, **model interpretability and explainability**, although supported via LightGBM feature importance and potential SHAP or LIME analysis, still require further development for real-world operational transparency. Security analysts may need more comprehensive, automated explanations of the system's decisions to effectively validate alerts and respond to evolving threats.

Addressing these limitations—through adversarial training, online learning for concept drift, extended benchmarking, and enhanced interpretability—will be the focus of future work to ensure the system remains robust, scalable, and trustworthy in diverse operational networks.

6. Conclusions

This study presents a robust and scalable hybrid intrusion detection framework that combines statistical anomaly scoring via Isolation Forests, deep feature extraction through a denoising autoencoder, and enhanced multi-class classification using LightGBM. The pre-processing pipeline—including class relabeling, one-hot encoding, min-max normalization, and SMOTE-ENN resampling—ensures balanced and high-quality input data. Extensive experiments on the NSL-KDD and UNSW-NB15 benchmark datasets demonstrate that the framework demonstrates consistent classification performance, with overall accuracies of 99% and 98%, respectively, and macro-F1 scores exceeding 97%. Notably, the model maintains reliable detection capabilities for rare attack classes such as R2L and U2R, highlighting the benefits of integrating anomaly scoring with deep feature learning.

While these results are promising, the study acknowledges certain limitations. Both NSL-KDD and UNSW-NB15 are benchmark datasets that may not fully reflect the diversity and dynamics of modern network traffic, and further evaluation on additional, contemporary datasets is needed. Additionally, deployment in real-time and resource-constrained environments may present practical challenges that require careful optimization.

While this study demonstrates promising results, limitations include evaluation on legacy benchmarks and lack of adversarial robustness testing.

Future work will focus on extending the framework to incorporate online learning for adaptive detection under concept drift, enabling real-time operation in edge and IoT environments. Efforts will also be directed toward improving model interpretability using techniques such as SHAP or LIME, enhancing adversarial robustness, and evaluating false alarm rates to ensure practical applicability in operational network security settings. Collectively, these directions aim to enhance the generalizability, resilience, and deployment readiness of hybrid intrusion detection systems for modern cybersecurity challenges.

Author Contributions: Methodology, S.K.; software development, S.K.; validation, S.K., W.M. and M.M.; formal analysis, S.K.; investigation, S.K.; resources, M.M.; data curation, S.K.; writing the initial draft, S.K.; review and editing, W.M. and M.M.; visualization, S.K.; supervision, M.M.; project administration, M.M. and W.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research did not receive any external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in the study are publicly available in [dataset] at [https://drive.google.com/drive/folders/16CBBGtoGsUo0Nb0hlwE-1yPFzR_WCY_G?usp=drive_link] (accessed on Day 13 October 2025)].

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

References

1. Hassan, W.; Hosseini, S.E.; Pervez, S. Real Time Anomaly Detection in Network Traffic Using Graph Neural Networks and Random Forest. *Lect. Notes Comput. Sci.* **2024**, *13937*, 194–207. [[CrossRef](#)]
2. Chalapathy, R.; Chawla, S. Deep Learning for Anomaly Detection: A Survey. *arXiv* **2019**, arXiv:1901.03407. Available online: <https://arxiv.org/abs/1901.03407> (accessed on 13 October 2025).
3. Kopčan, J.; Škvarek, O.; Klimo, M. Anomaly Detection Using Autoencoders and Deep Convolution Generative Adversarial Networks. *Transp. Res. Procedia* **2021**, *55*, 1296–1303. [[CrossRef](#)]
4. Iqbal, A.; Amin, R. Time Series Forecasting and Anomaly Detection Using Deep Learning. *Comput. Chem. Eng.* **2024**, *182*, 108560. [[CrossRef](#)]
5. Sakurada, M.; Yairi, T. Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, Gold Coast, Australia, 2 December 2014*; Association for Computing Machinery: New York, NY, USA, 2014; pp. 4–11. [[CrossRef](#)]
6. Xu, W.; Jang Jaccard, J.; Singh, A.; Wei, Y.; Sabrina, F. Improving Performance of Autoencoder Based Network Anomaly Detection on NSL KDD Dataset. *IEEE Access* **2021**, *9*, 140136–140146. [[CrossRef](#)]
7. Ounasser, N.; Rhanoui, M.; Mikram, M.; El Asri, B. Generative and Autoencoder Models for Large Scale Multivariate Unsupervised Anomaly Detection. In *Networking, Intelligent Systems and Security. Smart Innovation, Systems and Technologies*; Springer: Singapore, 2021; Volume 255, pp. 45–58. [[CrossRef](#)]
8. Khan, W.; Haroon, M. An Unsupervised Deep Learning Ensemble Model for Anomaly Detection in Static Attributed Social Networks. *Int. J. Cogn. Comput. Eng.* **2022**, *3*, 153–160. [[CrossRef](#)]
9. Li, X.; Xiao, C.; Feng, Z.; Pang, S.; Tai, W.; Zhou, F. Controlled Graph Neural Networks with Denoising Diffusion for Anomaly Detection. *Expert Syst. Appl.* **2024**, *237*, 121533. [[CrossRef](#)]
10. Sevyeri, L.R.; Fevens, T. AD CGAN: Contrastive Generative Adversarial Network for Anomaly Detection. *Lect. Notes Comput. Sci.* **2022**, *13325*, 322–334. [[CrossRef](#)]

11. Guo, H.; Zhou, Z.; Zhao, D.; Gaaloul, W. EGNN: Energy Efficient Anomaly Detection for IoT Multivariate Time Series Data Using Graph Neural Network. *Future Gener. Comput. Syst.* **2024**, *151*, 45–56. [\[CrossRef\]](#)
12. Xu, L.; Wang, Q.; Liu, W.; Zhang, Y. TGAN AD: Transformer Based GAN for Anomaly Detection of Time Series Data. *Appl. Sci.* **2022**, *12*, 8085. [\[CrossRef\]](#)
13. Retiti, C.; Moschetta, M.; Carlini, E.; Ricci, L. Anomaly Detection Based on GCNs and DBSCAN in a Large-Scale Graph. *Electronics* **2024**, *13*, 2625. [\[CrossRef\]](#)
14. Benaddi, H.; Jouhari, M.; Ibrahim, K.; Ben Othman, J.; Amhoud, E.M. Anomaly Detection in Industrial IoT Using Distributional Reinforcement Learning and Generative Adversarial Networks. *Sensors* **2022**, *22*, 8085. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. *arXiv* **2013**, arXiv:1312.6114.
16. Qin, K.; Wang, Q.; Lu, B.; Sun, H.; Shu, P. Flight Anomaly Detection via a Deep Hybrid Model. *Aerospace* **2022**, *9*, 329. [\[CrossRef\]](#)
17. Sabokrou, M.; Fathy, M.; Hoseini, M.; Klette, R. Deep-CNN-Based Approach for Anomaly Detection in Video Surveillance. *IEEE Trans. Image Process.* **2018**, *27*, 2515–2527. [\[CrossRef\]](#)
18. Hodge, V.J.; Austin, J. A Survey of Outlier Detection Methodologies. *Artif. Intell. Rev.* **2004**, *22*, 85–126. [\[CrossRef\]](#)
19. Mahmood, A.; Oliva, J.; Styner, M.A. Anomaly detection via Gumbel Noise Score Matching. *Front. Artif. Intell.* **2024**, *7*, 1441205. [\[CrossRef\]](#)
20. Yang, S.; Chen, H. Traffic Anomaly Detection Model Integrating Conditional Generative Adversarial Network and WaveNet. In Proceedings of the 2024 IEEE 2nd International Conference on Sensors, Electronics and Computer Engineering (ICSECE), Jinzhou, China, 29–31 August 2024. [\[CrossRef\]](#)
21. Xu, W.; Jang Jaccard, J.; Liu, T.; Sabrina, F.; Kwak, J. Improved Bidirectional GAN Based Approach for Network Intrusion Detection Using One Class Classifier. *Computers* **2022**, *11*, 85. [\[CrossRef\]](#)
22. AboulEla, S.; Kashef, R. Network Intrusion Detection Using a Stacking of AI Driven Models with Sampling. In Proceedings of the IEEE World AIoT Congress (AIoT), Seattle, WA, USA, 29–31 May 2024; pp. 157–164. [\[CrossRef\]](#)
23. Mushtaq, E.; Zameer, A.; Umer, M.; Abbasi, A.A. A Two Stage Intrusion Detection System with Auto Encoder and LSTMs. *Appl. Soft Comput.* **2022**, *121*, 108768. [\[CrossRef\]](#)
24. Dash, N.; Chakravarty, S.; Rath, A.K.; Giri, N.C.; AboRas, K.M.; Gowtham, N. An Optimized LSTM Based Deep Learning Model for Anomaly Network Intrusion Detection. *Sci. Rep.* **2025**, *15*, 1554. [\[CrossRef\]](#)
25. Malhotra, P.; Vig, L.; Shroff, G.; Agarwal, P. Long Short-Term Memory Networks for Anomaly Detection in Time Series. In Proceedings of the 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), Bruges, Belgium, 22–24 April 2015; pp. 89–94. [\[CrossRef\]](#)
26. Chohra, A.; Shirani, P.; Karbab, E.B.; Debbabi, M. Chameleon: Optimized Feature Selection Using Particle Swarm Optimization and Ensemble Methods for Network Anomaly Detection. *Comput. Secur.* **2022**, *117*, 102684. [\[CrossRef\]](#)
27. Sivasubramanian, A.; Devisetty, M.; Bhavukam, P. Feature Extraction and Anomaly Detection Using Different Autoencoders for Modeling Intrusion Detection Systems. *Arab. J. Sci. Eng.* **2024**, *49*, 13061–13073. [\[CrossRef\]](#)
28. Soleymanzadeh, R.; Aljasim, M.; Qadeer, M.W.; Kashef, R. Cyberattack and Fraud Detection Using Ensemble Stacking. *AI* **2022**, *3*, 22–36. [\[CrossRef\]](#)
29. Zhang, Z.; Kong, S.; Xiao, T.; Yang, A. A Network Intrusion Detection Method Based on Bagging Ensemble. *Symmetry* **2024**, *16*, 850. [\[CrossRef\]](#)
30. Li, Z.; Huang, C.; Deng, S.; Qiu, W.; Gao, X. A Soft Actor Critic Reinforcement Learning Algorithm for Network Intrusion Detection. *Comput. Secur.* **2023**, *135*, 103502. [\[CrossRef\]](#)
31. Wang, W.; Jian, S.; Tan, Y.; Wu, Q.; Huang, C. Representation Learning Based Network Intrusion Detection System by Capturing Explicit and Implicit Feature Interactions. *Comput. Secur.* **2022**, *112*, 102537. [\[CrossRef\]](#)
32. Liu, Y.; Wu, L. Intrusion Detection Model Based on Improved Transformer. *Appl. Sci.* **2023**, *13*, 6251. [\[CrossRef\]](#)
33. Tseng, S.-M.; Wang, Y.-Q.; Wang, Y.-C. Multi-Class Intrusion Detection Based on Transformer for IoT Networks Using CIC-IoT-2023 Dataset. *Future Internet* **2024**, *16*, 284. [\[CrossRef\]](#)
34. Maasaoui, Z.; Battou, A.; Merzouki, M.; Lbath, A. Anomaly Based Intrusion Detection Using Large Language Models. In Proceedings of the ACS/IEEE 21st International Conference on Computer Systems and Applications (AICCSA 2024), Sousse, Tunisia, 22–26 October 2024; ARNOVA: Indianapolis, IN, USA, 2024.
35. Albanbay, N.; Tursynbek, Y.; Graffi, K.; Uskenbayeva, R.; Kalpeyeva, Z.; Abilkaiyr, Z.; Ayapov, Y. Federated Learning-Based Intrusion Detection in IoT Networks: Performance Evaluation and Data Scaling Study. *J. Sens. Actuator Netw.* **2025**, *14*, 78. [\[CrossRef\]](#)
36. Huang, J.; Chen, Z.; Liu, S.-Z.; Zhang, H.; Long, H.-X. Improved Intrusion Detection Based on Hybrid Deep Learning Models and Federated Learning. *Sensors* **2024**, *24*, 4002. [\[CrossRef\]](#) [\[PubMed\]](#)
37. Nguyen, V.T.; Beuran, R. FedMSE: Federated learning for IoT network intrusion detection. *arXiv* **2024**, arXiv:2410.14121.
38. Adjewa, F.; Esseghir, M.; Merghem-Boulahia, L. Efficient federated intrusion detection in 5G ecosystem using optimized BERT-based model. *arXiv* **2024**, arXiv:2409.19390. [\[CrossRef\]](#)

39. Alsamiri, J.; Alsubhi, K. Federated Learning for Intrusion Detection Systems in Internet of Vehicles: A General Taxonomy, Applications, and Future Directions. *Future Internet* **2023**, *15*, 403. [[CrossRef](#)]
40. Ding, Z.; Zhong, G.; Qin, X.; Li, Q.; Fan, Z.; Deng, Z.; Ling, X.; Xiang, W. MF Net: Multi Frequency Intrusion Detection Network for Internet Traffic Data. *Pattern Recognit.* **2023**, *146*, 109999. [[CrossRef](#)]
41. Sadhwani, S.; Navare, A.; Mohan, A.; Muthalagu, R.; Pawar, P.M. IoT Based Intrusion Detection System Using Explainable Multi Class Deep Learning Approaches. *Comput. Electr. Eng.* **2025**, *123*, 110256. [[CrossRef](#)]
42. Zhang, J.; Chen, R.; Zhang, Y.; Han, W.; Gu, Z.; Yang, S.; Fu, Y. MF2POSE: Multi Task Feature Fusion Pseudo Siamese Network for Intrusion Detection Using Category Distance Promotion Loss. *Knowl. Based Syst.* **2024**, *283*, 111110. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.