ACIG
APPLIED CYBERSECURITY & INTERNET GOVERNANCE

NASK

SAE-AM: Enhancing Network Intrusion Detection Using Sparse Autoencoders with Attention Modules

Akanksha Pamena | Department of Computer Science, Central University of Kerala, Kerala, 671320, India | ORCID: 0009-0005-8798-5440

Manohar Naik S. | Department of Computer Science, Central University of Kerala, Kerala, 671320, India | ORCID: 0000-0002-1059-8945

Abstract

In response to the relentless evolution of cyber threats that continue to outpace traditional defence mechanisms, this study addresses key limitations in existing Intrusion Detection Systems (IDS), particularly those related to high dimensionality and computational inefficiency. We propose a novel framework, sparse autoencoders with attention modules (SAE-AM), which integrates SAE with both channel and positional attention mechanisms to enhance feature representation and optimise resource utilisation. While SAE effectively perform dimensionality reduction, the attention modules capture global dependencies across diverse input features. Leveraging a deep learning model, specifically a multi-layer perceptron classifier – our framework efficiently classifies normal and attack samples. Extensive evaluations on benchmark network intrusion datasets, CICIDS2017, NSL-KDD, and UNSW-NB15, demonstrate the robustness and superior performance of the proposed method, achieving 99% accuracy and minimal false alarm rates across all datasets. SAE-AM makes significant strides in overcoming the core challenges of NIDS by reducing dimensionality and improving computational efficiency. This novel approach contributes to enhanced network security, offering a

Received: 07.09.2025

Accepted: 03.11.2025

Published: 24.12.2025

Cite this article as:
A. Pamena, M.N. Sugali, "SAE-AM: Enhancing network intrusion detection using sparse autoencoders with attention modules," ACIG, vol. 4 no. 1, 2025, doi: 10.60097/ACIG/213872.

Corresponding author:
Akanksha Pamena, Department of Computer Science, Central University of Kerala, Kerala, 671320, India; E-mail: akanksha.pcs072206@cukerala.ac.in
0009-0005-8798-5440

lightweight and effective solution for real-world intrusion detection scenarios.

Keywords

intrusion detection, sparse autoencoders, dimensionality reduction, attention modules, multi-layer perceptron

1. Introduction

The escalating number of Internet users and their reliance on online services have led to a significant rise in cyberattacks targeting networks, causing disruptions to their usual operations. As a result, this appears to be contributing to the increasingly complex problem of cyberattacks on systems. Consequently, this issue needs quick attention in order to set up an intrusion detection system (IDS) that constantly monitors the inevitable and never-ending attacks on the Internet. An IDS is a surveillance system developed to observe and identify potentially malicious or unauthorised activities, triggering alerts upon their detection. IDS have consistently served as a safeguarding tool to protect network and information systems [1]. Current IDSs still struggle to provide effective detection services in the face of the ongoing rise in cyber threats to reduce false alarm rates (FAR) and unknown attacks [2]. By employing machine learning (ML) and deep learning (DL) techniques, researchers investigated the possibilities to enhance the existing conditions and services of IDS. In the last decade, these methods accomplished significant prominence in the field of network security [3].

Misuse detection and anomaly detection are the two categories in to which the IDS are divided according to the detection principles. Misuse detection is alternatively referred to as signature-based detection, and is limited in its ability to identify novel attack techniques, since it can only identify pre-existing patterns. These methods grew more inadequate and unfeasible as network traffic increased exponentially [4]. Conversely, anomaly-based IDS track changes in system behaviour to identify anomalies. Anomaly-based detection, as opposed to abuse detection, is more capable of identifying unknown attacks, thus becoming the focus of the IDS research [2].

However, network intrusion detection systems (NIDS) face many challenges to identify [5] malicious intrusions because of the massive increase in network business and security risks.

Since intrusion detection is considered as a classification problem, researchers have been using deep learning and machine learning methods to enhance the efficacy of IDS. Machine learning approaches have been widely used in IDS, and several researches have demonstrated positive results [6]. The two types of machine learning-based intrusion detection techniques are supervised learning and unsupervised learning. Machine learning techniques, such as in decision trees, logistic regression, AdaBoost, Gradient Boosting, Random forests, K-nearest neighbour classifiers, and Support Vector Machines, commonly employed in supervised learning to identify network behaviour by learning from the labelled data. Unsupervised intrusion detection techniques like hidden Markov models and K-means concentrate on the clustering challenge to classify network behaviours from the unlabelled data. To address the large-scale need, these detection systems also include deep learning techniques, including MLP, long-term short memory (LSTM), recurrent neural networks (RNN), and convolutional neural networks (CNN) are widely utilised for several approaches within the realm of artificial intelligence and machine learning. Further, researchers are interested in determining the best machine learning strategies for enhancing IDS performance [6].

Additionally, the effectiveness of anomaly-based intrusion detection (ID) algorithms in the detection process [7] is significantly affected by challenges such as high dimensionality, noisy data, and data complexity. To address these issues and enhance algorithm performance, a common strategy involves the implementation of data preprocessing, sampling methods and dimensionality reduction techniques. These techniques play a crucial role in mitigating dimensionality concerns, allowing researchers to navigate high-dimensional spaces more effectively [3]. However, there are various feature extraction techniques, like Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Random Forest Feature extractor, Generalised Discriminant Analysis (GDA), K-best feature selection, and Autoencoders (AE), offering diverse approaches to tackling these challenges.

However, despite the extensive research conducted in the field of IDS, the existing methods still grapple with a substantial false alert rate [8–12] and a minimal detection rate [11, 13–15]. Consequently, there has been a noticeable shift towards leveraging deep learning techniques as a means to alleviate some of these challenges. However, these deep learning approaches still face challenges such as high-dimensional data causing increased training time and resource usage, computational and resource limitations, a

high false-positive rate, and the prevalent issue of class imbalance leading to biased classifiers and difficulty in identifying the most informative features critical for distinguishing between normal and attack traffic [9, 11, 16–18]. Our main objective is to provide an effective, lightweight architecture that uses feature engineering to overcome these enduring difficulties, outperforming the state-of-the-art algorithms.

To address these limitations, in this paper, we use a variation of autoencoders for feature extraction. Autoencoders, a class of unsupervised deep learning models, have shown significant promise in network intrusion detection due to their ability to efficiently recognise and encode intricate patterns in data [19]. They consist of three main components: an encoder, a code (latent) layer, and a decoder. The encoder compresses the input into a lower-dimensional representation, which is then used by the decoder to reconstruct the original input. By minimising the reconstruction loss, autoencoders learn hidden and meaningful features from raw input data. Compared to traditional techniques like PCA and LDA, autoencoders offer greater flexibility and improved accuracy in feature extraction for IDS.

In this study, we propose a novel lightweight intrusion detection framework, termed SAE-AM, that combines sparse autoencoders (SAE) with attention mechanisms (channel and positional attention mechanisms) to address both computational complexity and the challenge of high-dimensional feature spaces. SAE enforce sparsity in latent representations, preserving only the most significant activations. The attention modules guide the model to focus on relevant features across the input dimensions, boosting the model's ability to capture global dependencies. To further enhance the detection of rare attacks, we adopt a hybrid resampling strategy using Random Under-Sampling (RUS) and SMOTE.

Our work makes the following significant contributions:

- A novel SAE-AM intrusion detection model is proposed, which integrates attention mechanisms within SAE to effectively detect anomalies in high-dimensional network traffic data.
- To address the issue of data imbalance, a hybrid resampling strategy combining RUS and SMOTE has been employed to enhance detection performance across minority classes.
- An efficient dimensionality reduction approach using SAE is applied to retain critical information while significantly reducing the input feature space.

- The integration of attention modules improves the model's capability to focus on salient features, resulting in superior classification performance.
- The proposed approach has been validated through comprehensive ablation studies and comparisons with state-of-the-art models using various evaluation metrics.

The paper is segmented into the subsequent sections. Section 2 describes overview of the prior research-related work while Section 3 presents the proposed methodology. The dataset descriptions as well as the implementation details are described in Section 4. In Section 5, the evaluation metrics and results of our proposed methodology and the comparison of performance with other models are presented. Section 6 provides the findings at the end.

2. Related Works

Over the years, various approaches leveraging machine learning, deep learning, and hybrid learning techniques have been proposed to enhance the efficiency and accuracy of IDS. This section reviews significant contributions in these areas, highlighting innovative methodologies and their experimental validations on benchmark datasets. By systematically evaluating and contrasting various methodologies, results, and conclusions, this review provides a clear understanding of the current state of research as described in Table 1. It highlights key trends and advancements, thereby identifying areas for future investigation.

2.1. Machine Learning-Based Approaches for IDS

Liu et al. [12] proposed a NIDS that leverages adaptive synthetic (ADASYN) oversampling technology and the Light GBM ensemble learning model. To tackle the problem of data imbalance, ADASYN oversampling is employed, and Light GBM is used as the classifier. Experimental validation on the NSL-KDD, UNSW-NB15, and CICIDS2017 datasets demonstrates the system's high detection accuracy of 89.79%, 83.98%, and 99.86%, respectively.

Hassan et al. [20] present a network intrusion detection model using an improved Binary Manta Ray Foraging (BMRF) Optimisation Algorithm and a Random Forest (RF) classifier. The BMRF algorithm, enhanced with adaptive S-shaped transfer functions, is used to choose the most pertinent features from intrusion detection datasets (NSL-KDD and CICIDS2017). However, it lags behind Naïve Bayes and XGBoost in execution time. Future work includes

exploring other classifiers to improve the BMRF algorithm for better feature selection and more efficient handling of data imbalance.

Das et al.'s [21] study introduces an ensemble-based machine learning approach to detect Distributed Denial of Service (DDoS) attacks. It combines supervised and unsupervised ensemble frameworks to achieve higher performance in detecting both known and unknown DDoS attacks. The supervised ensemble focuses on known attacks, while the unsupervised ensemble is effective in identifying previously unseen attacks through novelty and outlier detection. Experimental results using three benchmark datasets demonstrate the robustness and effectiveness of the proposed scheme in accurately detecting DDoS attacks with minimal false alarms.

Nguyen et al. [22] introduces a novel approach called Genetic Sacrificial Whale Optimisation (GSWO) to improve intrusion detection in Wireless Sensor Networks (WSNs). GSWO integrates a genetic algorithm (GA) and a modified whale optimisation algorithm (WOA) to overcome premature convergence and enhance global search capabilities. Furthermore, the CatBoost model is used for classification, adeptly managing categorical data with intricate patterns. A novel method for fine-tuning CatBoost's hyperparameters is introduced, utilising quantisation and the GSWO strategy, resulting in improved intrusion detection accuracy and real-time applicability.

2.2. Deep Learning-Based Approaches for IDS

Dahou et al. [23] designed a framework that integrates deep learning and metaheuristic optimisation algorithms for feature extraction and selection in Internet of Things (IoT) and cloud IDS. A CNN is used for feature extraction, while a novel feature selection method called Reptile Search Algorithm (RSA) optimises feature subset selection. Experimental results demonstrate RSA's superior performance over other optimisation methods as well as in testing scenarios. Future work includes improving RSA's convergence speed and exploring its application in training deep learning models for various IDS applications.

Yan et al. [24] introduces TL-CNN-IDS, an IDS based on transfer learning and ensemble learning. It employs feature engineering methods for enhanced model training, visualises network traffic data as images for CNN training, and utilises three CNN models (VGG16, Inception, and Xception) with hyperparameter optimisation and ensemble learning for improved intrusion detection. Evaluation

on CICIDS2017 and NSL-KDD datasets demonstrates efficient detection of various network attacks. Future work may focus on addressing dataset imbalances and classifying emerging threats using small-sample learning algorithms.

J. Zhang et al.'s study [25] proposes a network intrusion detection model that integrates a multi-head attention mechanism and Bi-directional LSTM (BiLSTM). The model uses embedding layers to convert high-dimensional feature vectors into low-dimensional ones, enhancing the information fusion. Multi-head attention assigns different weights to each vector, improving the detection accuracy by strengthening the relationships between vectors and attack types. BiLSTM captures long-distance dependencies in the data, further improving detection accuracy. A dropout layer is added to prevent overfitting. Experimental results demonstrate that the model surpasses others in accuracy and F1-score on the KDDCUP99, NSLKDD, and CICIDS2017 datasets.

Ullah et al. [26] proposed IDS for Imbalanced Network Traffic (IDS-INT), leverages transformer-based transfer learning to learn feature interactions in network traffic and uses the Synthetic Minority Oversampling Technique (SMOTE) to balance the dataset. Experiments conducted on UNSW-NB15, CIC-IDS2017, and NSL-KDD datasets demonstrate the effectiveness of IDS-INT on multi-class classification. Additionally, an explainable artificial intelligence approach is implemented to interpret the model, enhancing its trustworthiness and reliability.

Elnakib et al. [27] introduces an Enhanced Intrusion Detection Deep Learning Multi-class classification model (EIDM), designed to classify various types of attacks. The EIDM model is noted for its ability to classify all 15 classes individually without grouping similar classes, marking a significant advancement in IDS technology for IoT.

2.3. Hybrid Learning-Based Approaches for IDS

C. Zhang et al. [28] introduce a novel approach to abnormal traffic detection by addressing the uncertainty of samples in the dataset. This model integrates the concept of three-way decision into the random selection of feature attributes and evaluates attribute importance using decision boundary entropy. Additionally, a classifier evaluation function combining accuracy and diversity selects high evaluation value-based classifiers, and the gray wolf optimisation algorithm iteratively calculates optimal node weights to enhance prediction accuracy and robustness.

Wang et al. [29] introduces an outlier-detection algorithm for network traffic anomalies that integrates the BIRCH clustering algorithm with an autoencoder model. BIRCH pre-classifies datasets with complex data distributions, and the autoencoder detects outliers by setting a threshold. Validated on datasets, such as KDDCUP99, UNSW-NB15, CICIDS2017, and NSL-KDD, the BAE algorithm has demonstrated effective and accurate anomaly detection. However, the algorithm's computational complexity is high, and exploring alternative pre-classification algorithms and advanced autoencoders is necessary for better performance.

Altunay and Albayrak's [30] study proposed three deep learning models – CNN, LSTM, and a hybrid CNN+LSTM model – to detect intrusions in Industrial Internet of Things (IIoT) networks. Using the UNSW-NB15 and X-IIoTID datasets, both binary and multi-class classifications were conducted to identify normal and abnormal data. The models also demonstrated success in accurately detecting various attack types within the datasets. Addressing potential memorisation issues by incorporating synthetic data generation into the datasets and retraining the models is efficient.

Chintapalli et al. [31] proposed an efficient IDS, combining the Osprey Optimisation Algorithm (OOA) for feature selection and a modified Bi-LSTM network using the Exponential Linear Unit (ELU) activation function. The framework was tested on N-BaIoT, CICIDS2017, and ToN-IoT datasets, achieving high detection accuracies of 99.98%, 99.97%, and99.88%, respectively, along with reduced processing times. This IDS framework demonstrates a robust solution for protecting IoT systems from various cyber threats.

Peng et al. [32] introduces CBF-IDS, a new network intrusion detection system that combines CNNs and BiLSTMs with the focal loss function to address class imbalance. CBF-IDS efficiently extracts spatial and temporal features from network traffic and assigns higher weights to minority classes during training to improve performance. Despite its effectiveness, the hybrid model has increased complexity and computational demands.

Zhang and Wang [33] introduce a novel intrusion detection classification approach that integrates advanced feature engineering techniques and model optimisation. It utilises mutual information maximum relevance minimum redundancy (mRMR) feature selection and the synthetic minority oversampling technique (SMOTE) to enhance classifier accuracy by reducing feature redundancy and addressing class imbalance. Additionally, the Optuna method is

employed to optimise the hyperparameters of the CatBoost classifier, thereby enhancing the model's performance. It is evaluated on NSL-KDD, UNSW-NB15, and CICIDS2017 datasets.

The reviewed studies demonstrate significant advancements in the field of network intrusion detection, utilising a diverse array of machine learning, deep learning, and hybrid learning techniques. Overall, the continual evolution of IDS technologies through machine learning, deep learning, and hybrid learning approaches provides robust and effective solutions for countering the ever-growing threat of cyberattacks. Future research in this domain is expected to focus on refining these techniques, addressing computational complexities, and exploring new algorithms to maintain the efficacy and reliability of IDS in dynamic network environments.

2.4. Critical Analysis of Related Works

The literature reveals a growing trend in the application of machine learning, deep learning, and hybrid learning approaches for developing efficient IDS. These methods have demonstrated substantial promise in enhancing detection capabilities, especially in complex and evolving network environments. A notable strength observed in several studies is the effective handling of class imbalance through techniques such as ADASYN [12], SMOTE [33], and focal loss, which contribute to improved model generalisation. Additionally, the incorporation of metaheuristic optimisation algorithms – including GSWO [22], RSA [23], Opposition-based Owl Optimisation [31], and Binary Multiverse Relevance Feature Selection (BMRF) [20] – has led to significant advancements in feature selection and, consequently, better detection accuracy.

Hybrid models such as CNN+BiLSTM [32], CNN+LSTM [30], and CatBoost integrated with GWO [22], successfully combine spatial and temporal learning capabilities, offering robust intrusion detection performance in dynamic IoT and IIoT environments. Furthermore, ensemble learning and transfer learning techniques, including models like TL-CNN-IDS, have been adopted to enhance adaptability to emerging threats and improve detection rates across varying attack scenarios.

Despite these strengths, several limitations persist. The computational complexity and extended processing time associated with many hybrid and optimisation-based models pose a significant challenge, particularly in real-time or resource-constrained applications. Moreover, certain studies demonstrate a heavy reliance on

Table 1. Literature review on related studies.

Ref.	Year	Methodology	Datasets	Results	Findings
[34]	2021	ADASYN+ Light GBM	CICIDS2017 NSL-KDD UNSW-NB15	99.91 92.57 85.89	Oversampling complexity can lead to higher computational costs
[23]	2022	CNN-RSA	CICIDS2017 NSL-KDD KDDCUP99 BoT-IoT	99.99 99.23 99.9 99.99	Complexity and computational cost
[35]	2022	BMRF+RF	CICIDS2017 NSL-KDD	99.3 98.8	Slow execution time More dependency on classifier
[10]	2023	GWO+RF	CICIDS2017 NSL-KDD KDDCUP99	96.25 94.64 94.43	Complexity and overhead uncertain feature selection impact
[32]	2023	IG-FCBF+CNNmodels	CICIDS2017 NSL-KDD	99.85 99.53	More dependency on CNN models
[22]	2023	BIRCH-AE	CICIDS2017 NSL-KDD KDDCUP99 UNSW-NB15	92.58 87.88 87.61 95.95	Low performance results
[36]	2023	Multi-head Attention+BiLSTM	CICIDS2017 NSL-KDD KDDCUP99	99.08 95.19 98.28	Imbalanced training data Complex model leads to high computational cost
[37]	2023	CBF+CNN-BiLSTM	CICIDS2017 NSL-KDD UNSW-NB15	99.53 99.40 82.30	Increased model complexity High computational demands
[38]	2023	SMOTE+CatBoost	CICIDS2017 NSL-KDD UNSW-NB15	99.66 99.26 98.70	Complex implementation
[9]	2023	CNN+LSTM	CICIDS2017 NSL-KDD UNSW-NB15	99.15 98.17 98.81	Hybrid model leads to complex implementation
[39]	2023	EIDM	CICIDS2017	95	Deployment of multiple DL models require significant computational resources
[30]	2023	CNN+LSTM	UNSW-NB15 X-IIoTID	93.21 99.84	Deployment of multiple DL models require significant computational resources
[31]	2024	OOA+Bi-LSTM	CICIDS2017 ToN-IoT N-BaIoT	99.97 99.88 99.98	More computationally resource-intensive
[21]	2024	ML models	CICIDS2017 NSL-KDD UNSW-NB15	99.90 97.50 98.60	Low performance on UNSW-NB15 dataset
[12]	2024	GSWO-CatBoost	CICIDS2017 NSL-KDD WSN-DS WSNBFSF	99.74 99.76 99.62 99.99	More complex model

specific classifiers such as LightGBM [12] and CatBoost [22], which may restrict the generalisability of the proposed frameworks. A lack of standardisation in evaluation protocols and performance metrics across different datasets further complicates direct comparisons and objective assessments. Additionally, overdependence on synthetic sampling techniques like SMOTE and ADASYN can introduce the risk of overfitting or create unrealistic training conditions.

Several key research gaps have also been identified. The most current studies are confined to benchmark datasets such as NSL-KDD, CICIDS2017, and UNSW-NB15, with limited validation on real-world, encrypted, or highly imbalanced datasets. Furthermore, only a few works have considered adversarial threats such as data poisoning or evasion attacks, despite their increasing relevance in cybersecurity.

3. Methodology

In this section, we delineate the proposed methodology illustrated in Figure 1 and as shown in algorithm in Figure 2. SAE were employed for dimensionality reduction, aiming to decrease the number of features. In addition to that, attention modules are incorporated for the encoder part of SAE. A classifier, multi-layer perceptron (MLP) is used for classifying the binary and multi-class classification.

3.1. Sparse Autoencoders

Sparse autoencoders are a specialised type of autoencoders, which introduce L1 regularisation at the code layer, encouraging sparsity in hidden layer activations as depicted in Figure 3. This regularisation, applied through a penalty term in the loss function, promotes selective activation of neurons during training. The mean computed across training samples needs to be close to zero to meet the sparsity requirement [37, 39]. Unlike standard autoencoders, where all neurons can be activated simultaneously, SAE impose constraints to encourage more selective activation.

Training a sparse autoencoder involves optimisation of weights and biases to simultaneously minimise reconstruction error and satisfy sparsity constraints. This process facilitates dimension reduction, ultimately enhancing prediction accuracy with less training time [40]. SAE, characterised by their sparsity constraints on weights, are particularly valuable when a sparse and meaningful representation is desired and excels at compressed feature extraction, making them particularly suitable for applications like network intrusion detection. This suitability becomes especially

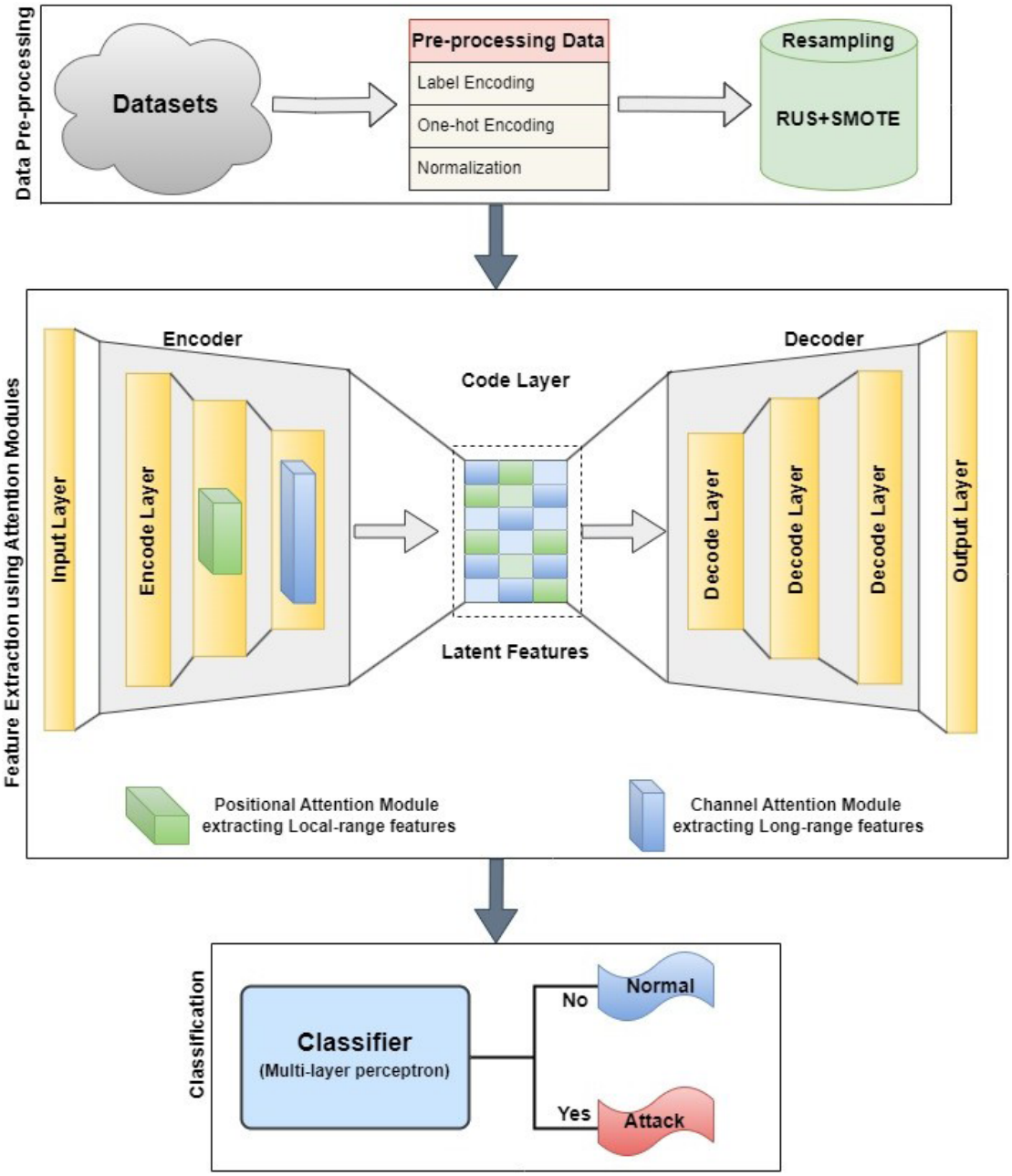

Figure 1. The framework of our proposed methodology.

beneficial in scenarios where interpretability or efficiency is of paramount importance.

Sparse autoencoders are particularly well-suited for network intrusion detection because they go beyond mere dimensionality

Algorithm 1 Enhanced Sparse Autoencoder with Channel and Positional Attention

Input: X: Training data, T: Total number of iterations, η: Learning rate, λ: Weight decay parameter, β: Sparsity penalty parameter, ρ: Desired average activation of hidden units
Output: h_c: Encoded features (30-dimensional)
Randomly initialize weights and biases W, b
for $t \leftarrow 0$ to $T - 1$ **do**
  for each batch X_{batch} do
    Forward Pass:
    Compute $h_1 = \sigma W_1 X_{batch}\, b_1$
    Apply Channel Attention: $h_2 = \text{ChannelAttention}\sigma W_2 h_1\, b_2$
    Apply Positional Attention: $h_3 = \text{PositionalAttentiona}\sigma W_3 h_2\, b_3$
    Compute code layer: $h_c = \sigma W_c h_3\, b_e$
    Decode: $\hat{h}_3 = \sigma W_4 h_c\, b_4$, $\hat{h}_2 = \sigma W_5 \hat{h}_3\, b_5$, $\hat{h}_1 = \sigma W_6 h_2\, b_6$
    Reconstruct: $\hat{X} = \sigma W_1 \hat{h}_1\, b_7$
    Compute Loss:
    Calculate average activation: $\hat{\rho} = \frac{1}{m} \sum_{i=1}^{m} h_i$
    Sparsity penalty: $KL\rho||\hat{\rho}$
    Total loss: $L = \frac{1}{2m} \sum_{i=1}^{m} || \hat{X}_i - X_{batch,i} ||^2 \frac{\lambda}{2} || W ||^2 \beta KL\rho || \hat{\rho}$
    Backpropagation:
    Compute gradients ∇_w, ∇_b
    Update Parameters:
    $W \leftarrow W - \eta \nabla_w L$
    $b \leftarrow b - \eta \nabla_b L$
  end for
end for
Return Encoded Features: $h_c = 0$

Figure 2. The algorithm of our proposed methodology.

reduction by learning compact and highly informative representations of input data. The sparsity constraint forces the model to activate only a limited number of neurons for any given input, leading to the extraction of critical, non-redundant patterns in network traffic [41, 42] This selective activation helps in identifying subtle and rare anomalies often present in malicious activities. In intrusion detection, where attack behaviours can be infrequent and masked by normal traffic, such focused representations improve the ability to detect anomalies and enhance classifier performance [43]. Moreover, SAE demonstrate robustness to noise and help mitigate the effects of data imbalance, making them a powerful feature extraction technique in the cybersecurity domain [44, 45]

3.2. Self-Attention Modules

The attention mechanism, inspired by human focus on key regions or words, is crucial for addressing intra-class differences and extracting context-rich information [46]. Following the

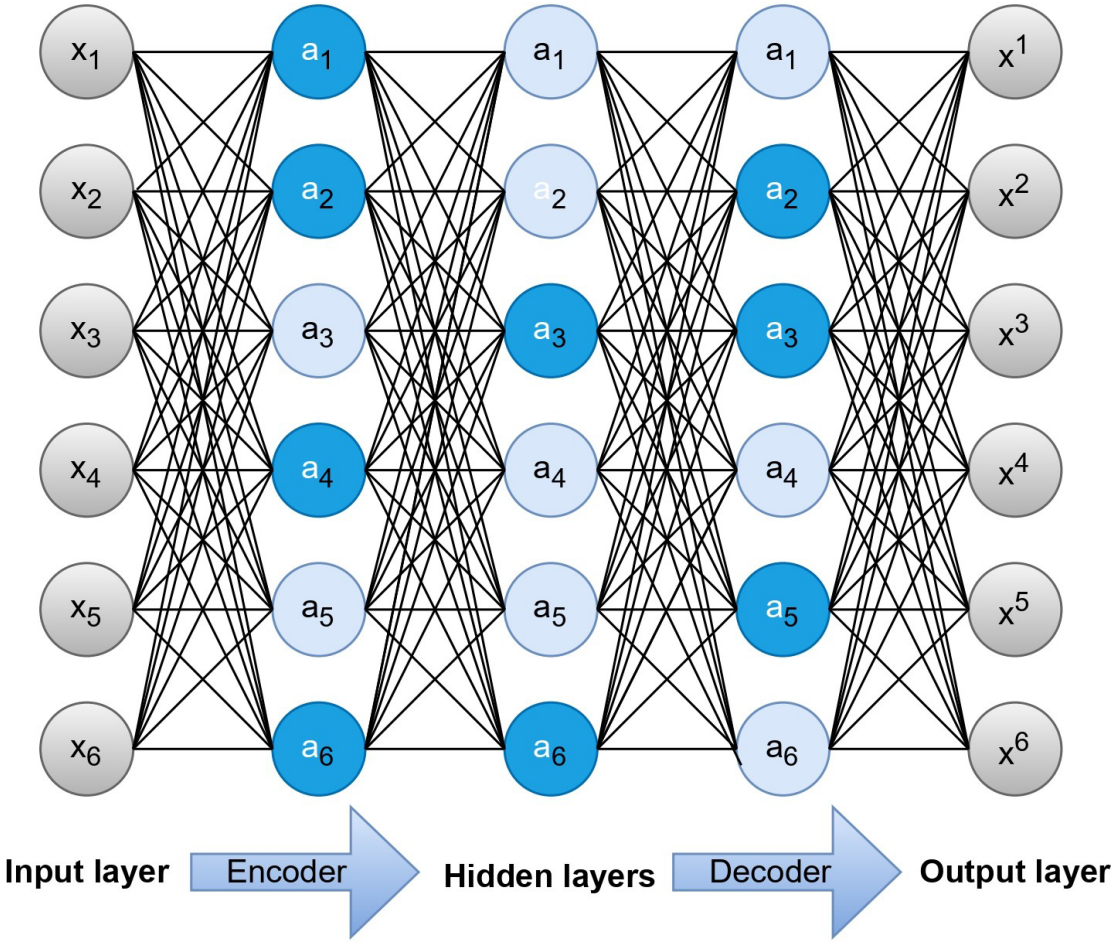

Figure 3. Architecture of sparse autoencoders.

encoder–decoder framework, attention selectively emphasises specific data segments by mapping queries to key-value pairs [36, 47].

Our approach incorporates attention modules to improve predictions by highlighting relevant feature sequences, assisting in detecting suspicious activities. Visualising attention probabilities reveals feature importance across traffic classes, enhancing model interpretability and adaptability through both positional and channel-wise attention.

3.2.1. Positional Self-Attention Module

Positional attention, on the other hand, is based on the transformer architecture and captures dependencies between features that may follow a temporal or sequential pattern. Features

like connection duration, protocol sequences, or timing information often contain contextual clues that are crucial for identifying complex or stealthy attacks. Positional attention enables the model to capture such long-range interactions by modeling the relationships between features regardless of their positions [36]. This is further supported by the idea of non-local operations, where the output at any position depends on the weighted sum of all positions in the input, giving the model a global perspective [34].

The positional self-attention module as shown in Figure 4 captures positional relationships within sequences, enhancing the model's contextual understanding [48]. Starting from a local feature $A \in R^{C\times H\times W}$, convolutional layers with batch normalisation and standard rectified linear unit (ReLU) activation produce feature maps B and C, reshaped to $R^{C\times N}$. A spatial attention map $S \in R^{N\times N}$ is then calculated as:

$$S_{ji} = \frac{\exp(B_i \cdot C_j)}{\sum_{i=1}^{N} \exp(B_i \cdot C_j)} \tag{1}$$

Meanwhile, A is processed to generate a new feature map D, reshaped to $R^{C\times N}$. A matrix multiplication between D and S^T is reshaped to $R^{C\times H\times W}$, scaled by a parameter α, and added to A to yield the final output E:

$$E_j = \alpha \sum_{i=1}^{N} S_{ji} D_i + A_j \tag{2}$$

Starting from zero, α gradually increases, adjusting feature weights. This operation combines information from all positions, enriching the model's global contextual awareness.

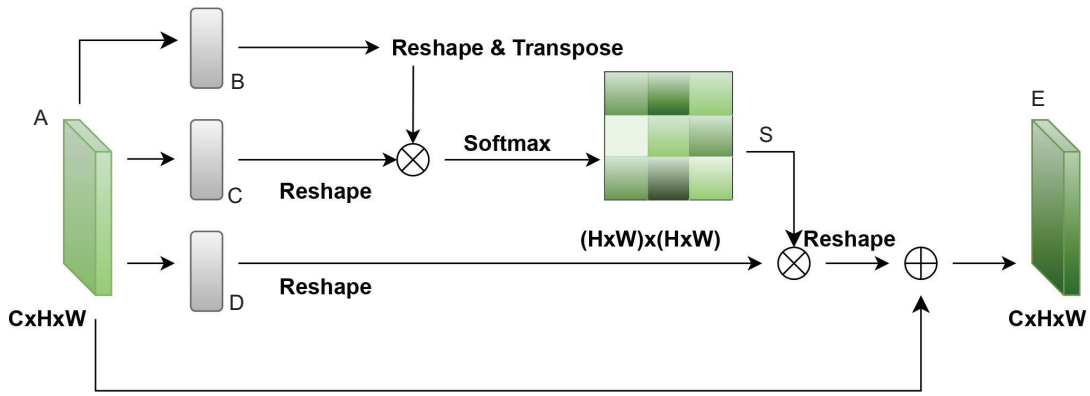

Figure 4. The structure of positional self-attention modules.

3.2.2. Channel Self-Attention Module

Channel attention helps the model identify and emphasise the most relevant feature channels by learning to assign different importance weights to each input feature. In the context of intrusion detection, where certain features like packet rates or byte counts are more informative, channel attention allows the model to focus on these critical features and reduce the influence of less relevant or noisy inputs. This concept is inspired by the squeeze-and-excitation networks, which have shown improved performance in various deep learning models by adaptively recalibrating feature responses [34].

The channel self-attention module, depicted in Figure 5, enhances model focus on key features by emphasising relevant channels and suppressing less useful ones [47, 48]. Unlike positional attention, which captures spatial dependencies, the channel attention map $X \in R^{C\times C}$ is computed by reshaping features $A \in R^{C\times H\times W}$, performing matrix multiplication with its transpose, and applying softmax:

$$X_{ji} = \frac{\exp(A_i \cdot A_j)}{\sum_{i=1}^{C} \exp(A_i \cdot A_j)} \tag{3}$$

The final output E is achieved by scaling and summing weighted channels with the original features:

$$E_j = \beta \sum_{i=1}^{C} X_{ji} A_i + A_j \tag{4}$$

This approach highlights class-specific channels, enhancing feature representation and model discriminability.

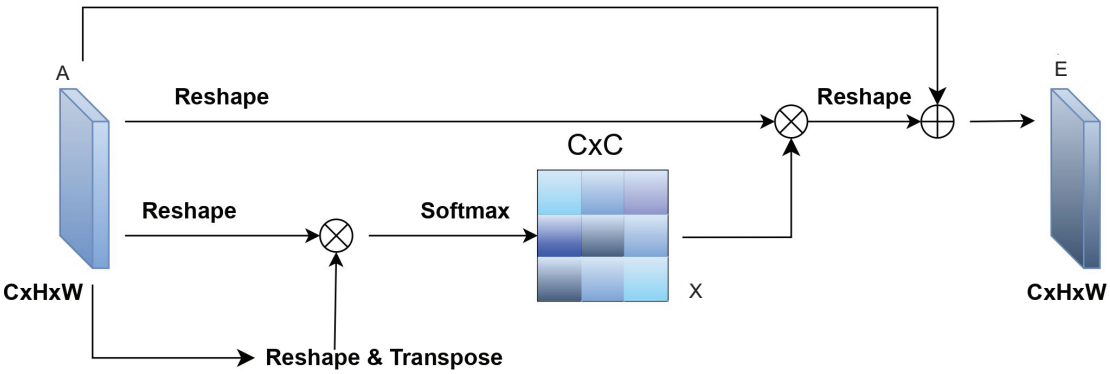

Figure 5. The structure of channel self-attention modules.

By combining both attention mechanisms, the model benefits from a dual focus: it learns which features are important (via channel attention) and how these features interact or evolve (via positional attention). This significantly enhances the model's ability to capture subtle patterns in high-dimensional network data, ultimately improving its effectiveness in detecting intrusions.

3.3. Multi-Layer Perceptron

A multilayer perceptron is a type of neural network known for its ability to model complex nonlinear relationships, making it well suited for high-dimensional data processing. Although MLPs were once considered state-of-the-art, they remain effective due to their adaptive learning capabilities, allowing them to adjust dynamically based on training data [49]. MLPs are among the most commonly used types of Artificial Neural Networks (ANNs) due to their straightforward architecture, minimal training requirements, and versatility. This layered architecture enables MLPs to learn intricate relationships, which is particularly useful in intrusion detection scenarios where data is often high-dimensional and nonlinear.

One of the key reasons MLPs are frequently employed in IDS is their capacity to automatically learn complex patterns without the need for extensive manual feature engineering [50]. MLPs function as feedforward networks, where data is passed from the input through hidden layers to the output. Learning is facilitated through back propagation, which iteratively adjusts the model's weights and biases to minimise prediction errors and improve performance [51].

Although other classifiers like Support Vector Machines (SVMs) are widely used, MLPs offer distinct advantages. While SVMs perform well for binary classification tasks, they often struggle with scalability in large, high-dimensional datasets [52, 53] In contrast, MLPs are more effective at handling such data by capturing complex, nonlinear decision boundaries [35]. Additionally, MLPs integrate seamlessly with deep learning frameworks, providing better scalability and adaptability for tasks like intrusion detection [54]. In this study, the MLP is chosen for its ability to model complex, nonlinear relationships in high-dimensional IDS datasets [55, 56]. MLPs are particularly well suited to process the latent representations generated by SAE, enabling a seamless and efficient deep learning pipeline [55, 57]. Furthermore, MLPs support gradient-based optimisation, making them easier to train and more compatible with modern deep neural

network architectures compared to traditional classifiers such as SVMs [36, 56].

Moreover, MLPs are versatile and can be effectively combined with advanced techniques such as SAE and attention mechanisms. This integration enhances both feature extraction and classification accuracy by enabling the model to focus on the most relevant features in the data [50, 58]. Therefore, among the various machine learning methods, the multilayer perceptron stands out, making it a focal point in this study's classification stage.

4. Datasets

4.1. CICIDS2017 Dataset

The CICIDS2017 dataset, created by the Canadian Institute for Cybersecurity, is a comprehensive collection of labeled network traffic data designed for studying various cyberattacks within a simulated network environment. This dataset is widely used in the intrusion detection research community due to its diverse range of attack types and realistic representation of network traffic. Collected over five consecutive days (Monday to Friday) in July 2017, the dataset includes both benign and malicious activities, making it well suited for training and evaluating IDS and other cybersecurity tools.

The dataset comprises 692,703 samples with 80 attributes, covering various aspects of network behaviour – ranging from basic packet-level details to advanced time- and host-based features. To further clarify our preprocessing steps, Figures 6 and 7 depict the distribution of attack categories before and after applying resampling techniques for both binary and multi-class classification tasks.

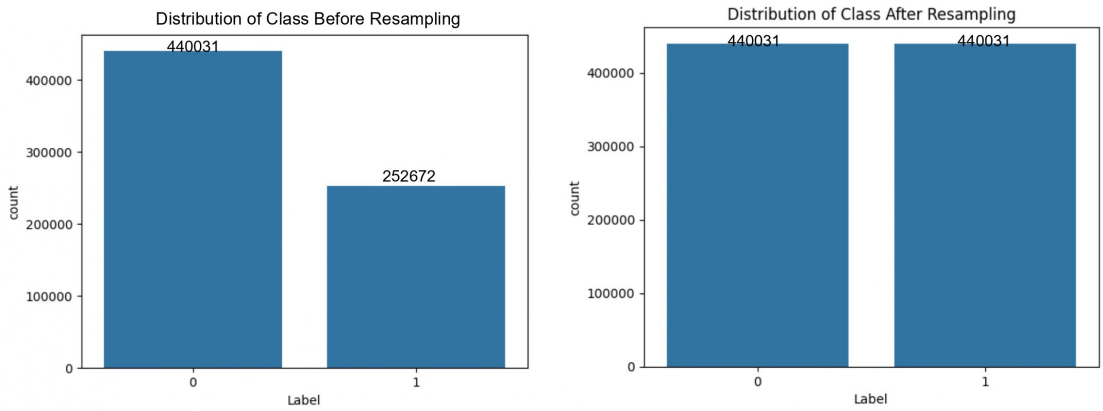

Figure 6. CICIDS2017 dataset binary class samples before and after resampling.

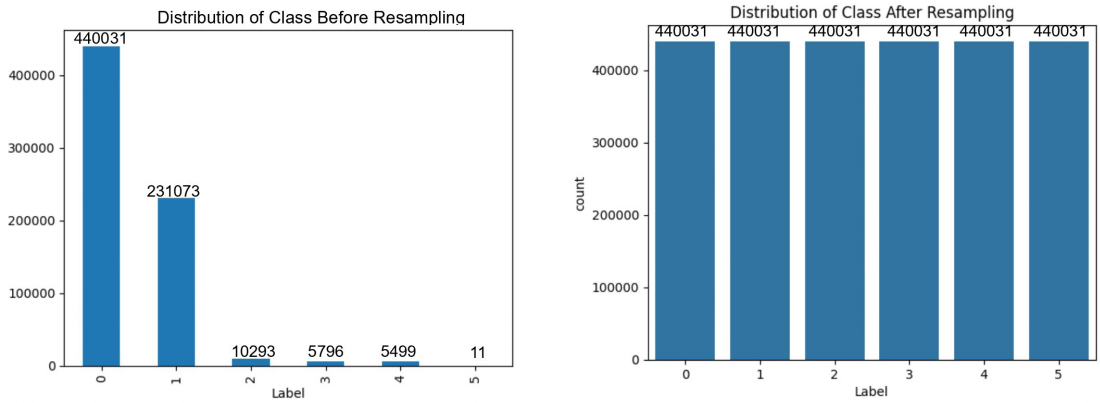

Figure 7. CICIDS2017 dataset multi-class samples before and after resampling.

4.2. NSL-KDD Dataset

The NSL-KDD dataset, commonly employed for network intrusion detection, serves as an effective benchmark for comparing various intrusion detection methods and is acknowledged for its popularity. However, a notable drawback is the presence of numerous redundant records, potentially impacting the efficacy of evaluated systems. To overcome this issue, researchers built a new refined dataset named NSL-KDD dataset. There are 125,973 records in the KDDTrain+ and 22,544 records in the KDDTest+ dataset with 42 features each. Among the 42 features, we found 6 binary, 3 category, 32 numerical are input attributes, and 1 class label. The datasets contain 22 and 38 attack types in the KDDTrain+ and KDDTest+ datasets, respectively. These attack types are categorised into four primary kinds: DoS, Probe, U2R, and R2L. Figures 8 and 9 illustrate the class-wise distribution of attacks prior to and following the application of resampling techniques, for both binary and multi-class classification scenarios.

4.3. UNSW-NB15 Dataset

The UNSW-NB15 dataset, derived from real network traffic data generated by IXIA PerfectStorm, encompasses a wide variety of attack types alongside normal traffic. This dataset is highly regarded for its realism, offering scenarios that closely mimic actual network conditions, unlike some synthetic datasets. The dataset includes separate training and testing sets, with 175,341 and 82,332 records, respectively, each containing 43 original features. After converting categorical features into binary representations, the dataset expands to 194 features, enhancing its complexity and usability for machine learning-based IDS development. Figures 10

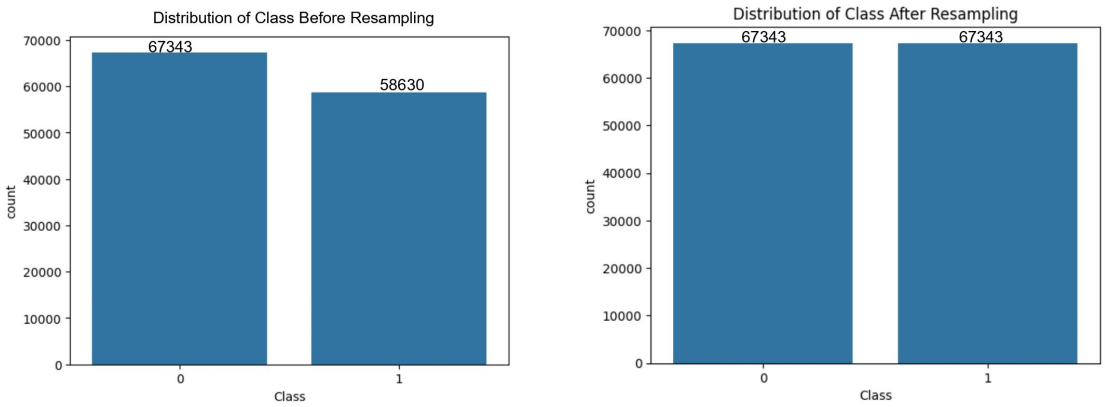

Figure 8. NSL-KDD dataset binary class samples before and after resampling.

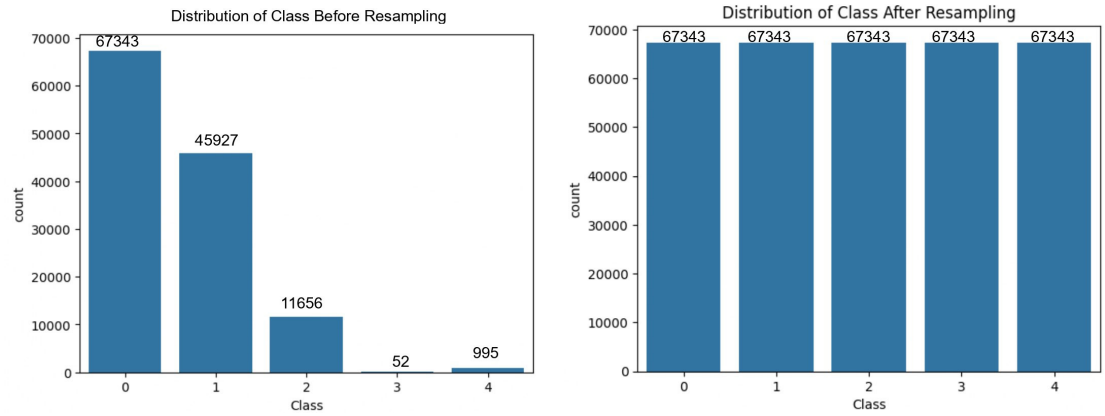

Figure 9. NSL-KDD dataset multi-class samples before and after resampling.

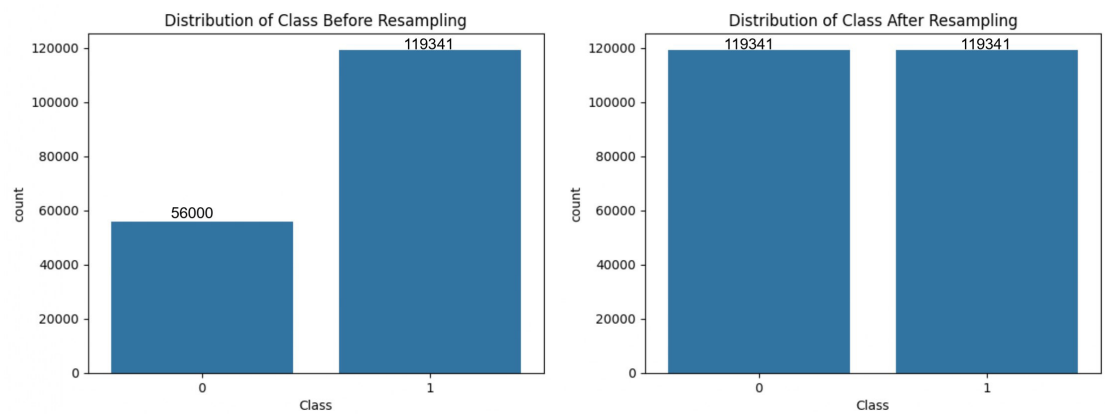

Figure 10. UNSW-NB15 dataset binary class samples before and after resampling.

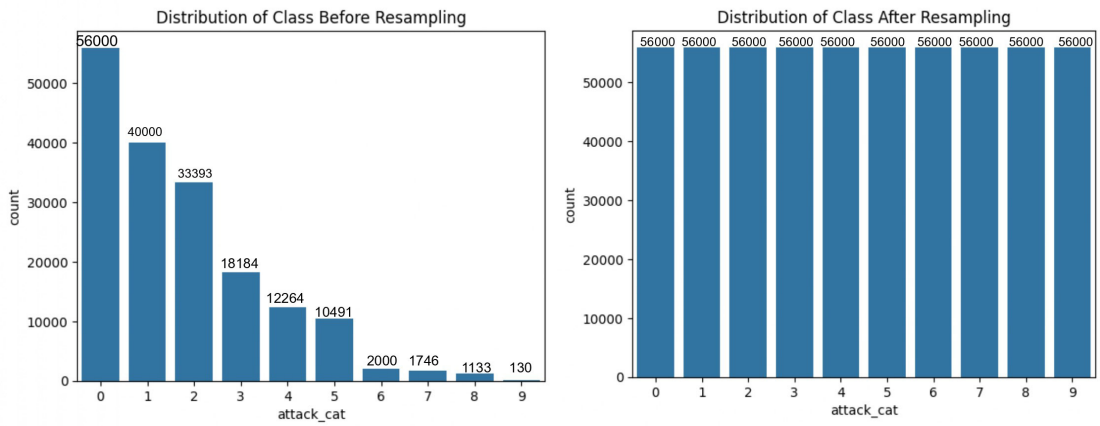

Figure 11. UNSW-NB15 dataset multi-class samples before and after resampling.

and 11 present the comparison of attack class distributions before and after resampling, highlighting the impact of the applied techniques on both binary and multi-class classification tasks.

For all the three datasets, we used a standard 80-20 split for training and testing the models, where 80% of the samples were allocated for training and 20% for testing. These datasets are frequently used in network intrusion detection for evaluating and comparing different intrusion detection methods. They are a comprehensive collection of labeled network traffic data designed for studying various cyber-attacks within a simulated network environment. These datasets are highly regarded for their realism, offering scenarios that closely mimic actual network conditions, unlike some synthetic datasets. The datasets include both benign and malicious activities, making it suitable for training and evaluating IDS and other cyber-security tools.

5. Implementation

Our proposed methodology has been implemented on Windows 11 operating system environments with AMD Ryzen 7 7730U with Radeon Graphics processor, 16GB RAM, and Python 3.10.12 version using Keras library with TensorFlow as back end. For the proposed methodology, the experiments, training, and testing were conducted using the popular benchmark CICIDS2017, NSL-KDD, and UNSW-NB15 datasets.

5.1. Data Preprocessing

In the data preprocessing stage, out of the 42 and 43 features in the NSL-KDD and UNSW-NB15 datasets, respectively, we

identified 37 numeric features and three non-numeric features. One-hot encoding is used in the dataset to encode non-numeric characteristics. For instance, the three-attribute protocol type in the datasets is substituted with three-dimensional (3D) vectors (1, 0, 0), (0, 1, 0), and (0, 0, 1). Finally, the NSL-KDD dataset contains 121-dimensional and the UNSW-NB15 dataset contains 194 characteristics after conversion. Additionally, we use Min-Max normalisation to normalise the data. Using this method, all the scaled data is obtained in the interval (0, 1).

5.2. Resampling Method

The NSL-KDD and UNSW-NB15 datasets exhibit notable class imbalance, where certain attack categories have significantly fewer instances, compared to others. This imbalance can lead to biased model learning, where the classifier favors majority classes and performs poorly on minority classes. Additionally, the presence of high-dimensional data with redundant and noisy features can further degrade model performance.

To address these challenges, we employed a two-fold strategy: (1) balancing the class distribution using resampling techniques, and (2) reducing dimensionality through feature extraction. For resampling, we utilised both Random Under Sampling (RUS) and Synthetic Minority Over-sampling Technique (SMOTE). RUS reduces the sample size of the majority class, while SMOTE generates synthetic samples for the minority class to balance the dataset.

The datasets were split into training, validation, and testing sets, and resampling was applied only to the training set to avoid data leakage. The specific number of samples before and after resampling for both binary and multi-class classification tasks are shown in Figures 6–11. This balanced and compact representation enhances the model's ability to generalise, reduces computational complexity, and improves detection accuracy across both frequent and rare attack types.

5.3. Feature Extraction Through Sparse Autoencoders with Attention Modules

For the effective feature extraction method, we integrated SAE with attention modules. SAE consist of three primary layers: the encoder layer, the code layer, and the decode layer.

To enforce sparsity in the learned representations, L1 regularisation is applied, encouraging many weights in the code layer to be exactly

zero. The model further integrates channel and positional attention modules, enhancing its ability to capture long-range dependencies and relationships within the input data. This architecture combines the feature-learning capabilities of SAE with channel attention in convolutional layers to adaptively weight channels and positional attention in transformer layers to capture sequence information. The decoder reconstructs the encoded data, utilising a sigmoid activation function in the output layer, while all other layers employ the ReLU activation function. The SAE-AM model is trained for up to 20 epochs, minimising reconstruction loss, with the mean square error (MSE) used to quantify the difference between actual and predicted values across the dataset.

In this experiment, we utilised the CICIDS2017, NSL-KDD, and UNSW-NB15 benchmark pre-processed datasets, which contain 79, 121, and 194 features, respectively. The dimensionality of these datasets was reduced to 30 features using the sparse autoencoder before being fed into a classifier. Our methodology combines the sparse autoencoder with L1 regularisation and attention modules to refine and optimise feature sequence predictions. The encoded features undergo both channel and positional attention mechanisms, demonstrating a synergistic relationship between SAE and attention modules.

5.4. Classifier

As shown in Table 2, the classifier, MLP comprises three dense layers with 160, 80, and 40 units, respectively, and an output layer with 2 units. Dropout layers (with a rate of 0.5) are inserted between hidden layers for data regularisation and to mitigate overfitting. The activation function for hidden layers is the ReLU [59], while the final layer uses the sigmoid activation function for binary classification. For multi-class classification, the output layer has six, five, or 10 units, utilising the softmax activation function to generate probability distributions across different classes. The binary cross-entropy loss function quantifies differences between predicted and real labels for binary tasks, while categorical cross-entropy is used for multi-class scenarios. The Adam optimiser, with a learning rate of 0.001, is employed for optimising model parameters.

5.5. Hyperparameter Tuning

To select the best model parameters for feature extraction, we performed hyperparameter tuning as shown in Table 3, using

Table 2. Summary of the MLP model.

Layer (type)	Output shape	Param#
dense_5 (Dense)	None, 160	5280
dropout (Dropout)	None, 160	0
dense_6 (Dense)	None, 80	12880
dropout_1 (Dropout)	None, 80	0
dense_7 (Dense)	None, 40	3240
dropout_2 (Dropout)	None, 40	0
dense_8 (Dense)	None, 1	41
Total params:	21441 (83.75 KB)	
Trainable params:	21441 (83.75 KB)	
Non-trainable params:	0 (0.00 Byte)	

Table 3. Hyperparameter tuning results.

Hyperparameter	Tested values	Optimal value
Learning rate (LR)	0.001, 0.01, 0.0001	0.0001
Regularisation (λ)	0.001, 0.01, 0.1	0.001
Dropout rate	0.3, 0.4, 0.5	0.5
No. of hidden units	160, 80, 40	160
Batch size	32, 64, 128	64
Optimizer	Adam, RMS prop	Adam

a reduced feature set of 30 and a lambda (λ) for regularisation strength. The optimal configuration of SAE integrated with attention modules was selected based on minimising validation loss, ensuring fine-tuned model performance.

In designing the SAE, we adopted a three-layer architecture comprising an encoder, a code layer, and a decoder. This struc-ture was chosen to balance complexity and performance, providing sufficient capacity for compressing high-dimensional input features into a reduced and informative representation. Specifically, the input features from datasets such as CICIDS2017 (79 features), NSL-KDD (121 features), and UNSW-NB15 (194 features) were compressed into 30 latent features. This dimensionality reduction was selected based on empirical trials to preserve essential information while reducing noise and computational cost. To determine the optimal number of features for downstream classification, we

conducted experiments with varying dimensionalities of the latent representation, specifically exploring feature sizes of 10, 20, 30, 40, and 50. We selected 30 as the optimal feature size based on empirical evaluation, where it consistently provided the best balance between classification accuracy and model complexity across all datasets. Reducing the features to 30 allowed the model to retain the most informative characteristics of the data while minimising redundancy and computational overhead. This configuration was particularly effective in improving generalisation performance without sacrificing important patterns necessary for accurate intrusion detection.

To determine the optimal sparsity level in the latent representation, we performed hyperparameter tuning by varying the L1 regularisation parameter (λ) within the range {0.0001, 0.001, 0.01, 0.1}. This parameter enforces sparsity in the code layer of the SAE, encouraging the network to learn compact and discriminative feature representations. Based on extensive validation experiments, λ = 0.001 was identified as the optimal value, striking a balance between sparsity and reconstruction accuracy. We employed the ReLU activation function in the encoder and decoder layers due to its efficiency and ability to model nonlinear patterns, while the final decoder layer used a sigmoid activation to reconstruct normalised inputs within the (0, 1) range.

Furthermore, we integrated channel and positional attention modules into the SAE to enhance its capability to capture long-range dependencies and feature relevance. Channel attention enables the network to focus on the most informative feature channels, while positional attention – drawn from transformer principles – helps model sequential or positional dependencies in the feature set. These modules were included to improve the contextual understanding of feature relationships. This is critical in intrusion detection scenarios, where subtle feature interactions can signal malicious activity.

Regarding the classifier, the MLP was constructed with three hidden layers containing 160, 80, and 40 neurons, respectively, followed by a task-specific output layer. This pyramidal structure enables progressive abstraction and feature transformation, allowing the model to capture complex patterns while mitigating overfitting risks. The sizes of the layers were chosen based on iterative experimentation, where this configuration provided the best performance across validation datasets. Dropout layers with a dropout rate of 0.5 were added after each hidden layer to prevent overfitting

and improve generalisation. The Adam optimiser was employed for training both SAE and MLP models due to its robustness and adaptability to different learning conditions. A learning rate of 0.0001 was used for training the SAE to ensure stable convergence during feature extraction, while a learning rate of 0.001 was optimal for training the MLP. These values were selected from a range of candidates during the hyperparameter tuning process, as detailed in Table 3. The choice of activation functions in the MLP followed common best practices: ReLU for hidden layers due to its non-saturating nature and computational efficiency, and sigmoid or softmax in the output layer depending on whether the classification task was binary or multi-class, respectively.

Overall, the architecture and hyperparameters were selected through rigorous experimentation and tuning to achieve an optimal balance of model accuracy, complexity, and interpretability, particularly in the context of high-dimensional, real-world intrusion detection datasets.

6. Results and Discussion

The model's performance is evaluated using a confusion matrix and several key metrics, which provide insights into the model's effectiveness. These metrics – accuracy, precision, recall, F1-score, FAR, and time (seconds/epoch) – facilitate comparisons with other binary and multi-class classification models.

Our proposed technique, SAE-AM, demonstrated superior performance compared to state-of-the-art algorithms on benchmark datasets, such as CI-CIDS2017, NSL-KDD, and UNSW-NB15, as shown in Table 4. These results underscore SAE-AM's reliability and effectiveness as a solution for NIDS, particularly in addressing the complex challenges posed by evolving cyber threats.

To further validate the robustness of the SAE-AM model and address concerns about potential overfitting due to high accuracy in binary classification, we present learning curves for both training and validation loss and accuracy in Figure 12. These curves demonstrate that the model exhibits consistent learning behaviour, with training and validation losses steadily decreasing and no signs of divergence across epochs. Similarly, the accuracy curves remain stable, indicating that the model generalises well and does not suffer from overfitting. This further reinforces the credibility of the reported result of 100% of the CICIDS2017 data set using binary classification.

Table 4. Performance results (in %) of proposed SAE-AM methodology.

Dataset	Classification	Accuracy	Precision	Recall	F1-Score	FAR	Time (sec/epoch)
CICIDS2017	Binary	100.00	100.00	100.00	100.00	0.000	15.3s
	Multi-class	99.84	99.84	99.92	99.84	0.0007	70.4s
NSL-KDD	Binary	99.81	99.33	99.22	98.73	0.010	16.6s
	Multi-class	99.38	98.68	99.72	99.81	0.018	47.8s
UNSW-NB15	Binary	99.83	98.56	97.84	98.25	0.019	13.2s
	Multi-class	89.84	91.98	94.72	95.85	0.094	75.8s

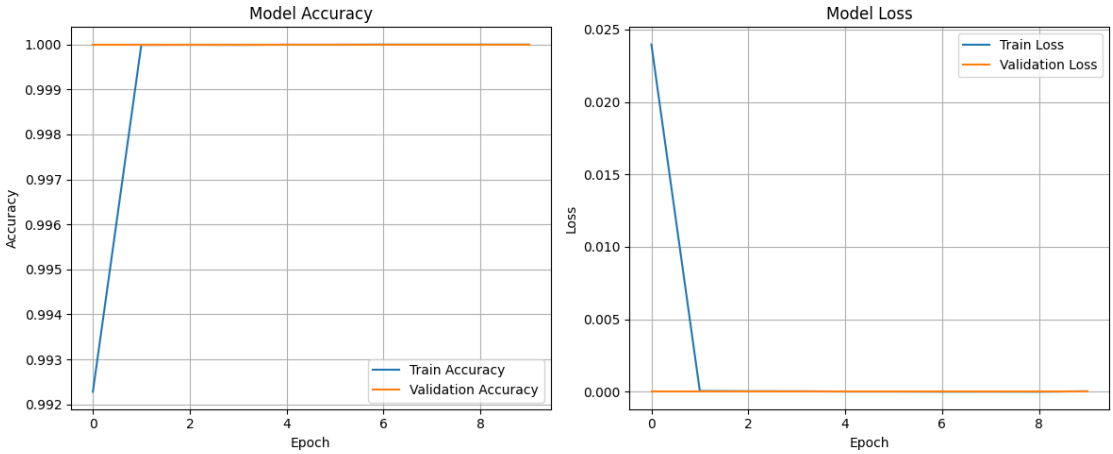

Figure 12. Learning curves of binary classification using CICIDS2017 dataset.

In addition to traditional evaluation metrics, the performance of SAE-AM can be further assessed through Receiver Operating Characteristic (ROC) curve analysis. ROC curves provide a comprehensive visualisation of the trade-off between the true positive rate (sensitivity) and the false positive rate (1-specificity) across different classification thresholds. A higher Area Under the Curve (AUC)–ROC value indicates superior performance in distinguishing between normal and abnormal classes, particularly in binary classification scenarios.

Figure 13 illustrates the ROC curves for binary and multi-class classifications of the CICIDS2017 dataset, showcasing the model's discrimination capability across multiple attack types. Additionally, Figure 14 depicts the ROC curves for binary and multi-class classifications of the NSL-KDD dataset. These figures provide visual insights into how well the SAE-AM model performs in terms of sensitivity and specificity thresholds for both datasets. Furthermore,

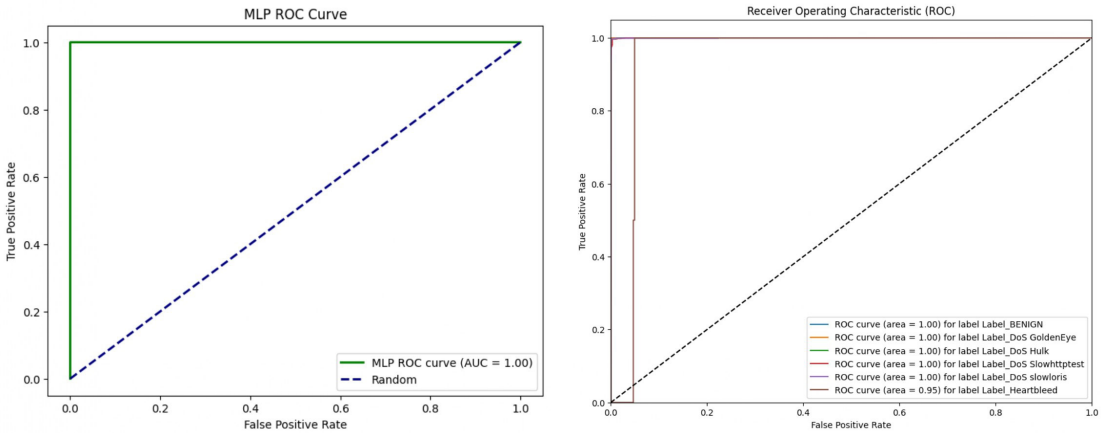

Figure 13. ROC curves of CICIDS2017 binary and multi-class classification.

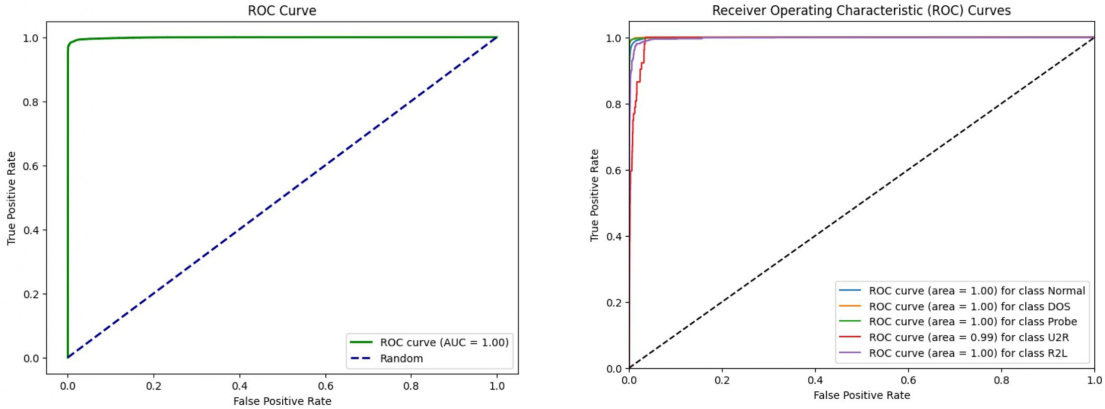

Figure 14. ROC curves of NSL-KDD binary and multi-class classification.

Figure 15 illustrates the ROC curves of binary and multi-class classification of the UNSW-NB15 dataset.

6.1. Ablation Studies

To evaluate the effectiveness of our proposed SAE-AM methodology, we conducted ablation studies across three datasets: CICIDS2017, NSL-KDD, and UNSW-NB15. The core structure of theSAE-AM framework integrates SAE with attention modules for enhanced feature extraction, followed by a MLP classifier to detect and classify various types of attacks. Table 5 presents the results of the ablation experiments, including configurations such as SAE+MLP (without attention modules), AM+MLP (without SAE), and SAE-AM+MLP (our proposed methodology), demonstrating the superiority of the proposed methodology.

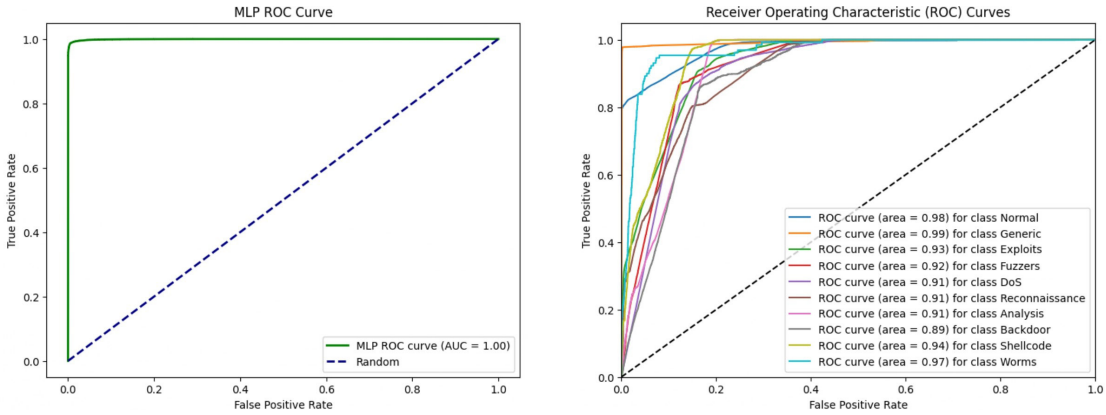

Figure 15. ROC curves of UNSW-NB15 binary and multi-class classification.

Table 5. Ablation study results (in %) across three datasets.

Model	Dataset	Accuracy	Precision	Recall	F1-score	FAR
	CICIDS2017	98.92	99.0	97.57	98.76	0.063
SAE+MLP	NSL-KDD	97.56	98.26	97.88	97.89	0.089
	UNSW-NB15	95.90	97.53	98.72	97.19	0.93
	CICIDS2017	97.45	93.03	96.35	97.73	0.058
AM+MLP	NSL-KDD	93.81	96.28	97.24	94.66	0.032
	UNSW-NB15	91.06	93.65	94.09	96.35	0.234
	CICIDS2017	100	100	100	100	0.000
SAE-AM+MLP	NSL-KDD	99.81	99.33	99.22	98.73	0.010
	UNSW-NB15	99.83	98.56	97.84	98.25	0.019

In our ablation studies, we systematically removed specific components of the SAE-AM framework to assess their impact on model performance. We evaluated the importance of SAE, which facilitate dimensionality reduction, and the dual attention modules that capture global dependencies in the data. Additionally, we tested the performance of the MLP classifier without these enhancements.

The results indicate that removing SAE leads to a significant drop in classification accuracy, confirming their essential role in efficient feature extraction and dimensionality reduction. Similarly, excluding the channel and positional attention modules resulted in a marked decline in the model's ability to capture global relationships, leading to higher FAR. These findings underscore the necessity of both SAE and attention modules in the SAE-AM framework,

as they optimise computational efficiency while maintaining high accuracy in detecting both normal and attack samples.

6.2. Comparison Analysis

To comprehensively evaluate the efficacy of our proposed SAE-AM methodology in NIDS, we conducted a comparative analysis using the CICIDS2017, NSL-KDD, and UNSW-NB15 datasets for binary classification and multi-class classification, as shown in Tables 6 and 7, respectively. We benchmarked against various machine learning, deep learning, and hybrid approaches. The performance of various IDS models was assessed based on several key criteria, including the overall accuracy, precision, recall, F1-score, FAR, time lapse (seconds/epoch), handling of imbalanced data, feature selection and extraction, model complexity and computational efficiency, robustness and generalisation, and novelty detection.

6.2.1. Binary Classification

The proposed SAE-AM model demonstrates outstanding performance in binary classification, achieving 100% accuracy on the CICIDS2017 dataset, indicating perfect classification of network traffic into benign and malicious categories. The model also achieved 100% precision, recall, and F1-score, highlighting its exceptional capability to correctly identify true positives while minimising false alarms, crucial for real-world applications.

On the NSL-KDD dataset, the model achieved an accuracy of 99.81%, suggesting excellent performance but indicating slight room for improvement in reducing false positives. For the UNSW-NB15 dataset, the model recorded an accuracy of 99.83%, reflecting its robust ability to distinguish between normal and malicious traffic.

In summary, the SAE-AM model showcases exceptional capabilities in binary classification across multiple datasets, positioning it as a reliable solution for network intrusion detection.

6.2.2. Multi-Class Classification

In the realm of multi-class classification, the proposed SAE-AM+MLP method was evaluated across three benchmark datasets, showcasing strong performance with some limitations in comprehensive metric reporting.

On the CICIDS2017 dataset, the model achieved an accuracy of 99.84%, and also outstanding experimental results indicating strong

Table 6. Comparison of proposed method (SAE-AM) with other methods of binary class classification.

Ref.	Method+classifier	Datasets	Accuracy	Precision	Recall	F1-score	FAR	Time
[23]	CNN-RSA	CICIDS2017	99.99	99.99	99.99	99.99	N/A	N/A
		NSL-KDD	99.23	99.23	99.23	99.23	N/A	N/A
[35]	BMRF+RF	CICIDS2017	99.3	99.6	94.3	96.9	N/A	15.17 seconds
		NSL-KDD	98.8	96.8	96.2	96.5	N/A	56.86 seconds
[10]	GWO+RF	CICIDS2017	96.25	96.14	93.75	N/A	N/A	50.63 seconds
		NSL-KDD	94.64	95.31	93.14	N/A	N/A	40.29 seconds
[32]	IG-FCBF+CNNmodels	CICIDS2017	99.85	99.85	99.85	99.85	N/A	N/A
		NSL-KDD	99.53	96.77	97.63	97.13	N/A	N/A
[37]	CBF+CNN-BiLSTM	CICIDS2017	99.53	99.54	99.53	99.53	N/A	N/A
		NSL-KDD	99.40	99.40	**99.40**	99.40	N/A	N/A
[22]	BIRCH-AE	CICIDS2017	92.58	97.27	86.29	91.42	N/A	N/A
		NSL-KDD	87.88	89.81	88.05	88.46	N/A	N/A
		UNSW-NB15	87.61	97.13	74.20	81.13	N/A	N/A
[38]	SMOTE+CatBoost	CICIDS2017	99.66	99.89	99.09	99.49	N/A	N/A
		NSL-KDD	99.26	**99.63**	99.22	**99.43**	N/A	N/A
		UNSW-NB15	82.30	82.76	82.30	79.61	N/A	N/A
[12]	GSWO-CatBoost	CICIDS2017	99.74	97.39	93.68	95.32	N/A	73 seconds
		NSL-KDD	99.76	96.17	95.14	95.63	N/A	37 seconds
[21]	ML models	CICIDS2017	99.90	99.90	99.90	N/A	0.10	N/A
		NSL-KDD	97.50	99.10	95.20	N/A	0.60	N/A
		UNSW-NB15	98.60	98.20	97.60	N/A	0.09	N/A
Proposed	**SAE-AM+MLP**	CICIDS2017	**100**	**100**	**100**	**100**	**0.000**	15.3 seconds
		NSL-KDD	**99.81**	99.33	**99.22**	**98.73**	**0.010**	16.6 seconds
		UNSW-NB15	**99.83**	**98.56**	**97.84**	**98.25**	**0.019**	13.2 seconds

effectiveness in identifying different types of attacks. Additionally, the model demonstrated a very low FAR of 0.0007 and a computational time of 70.4 seconds per epoch, making it suitable for real-time applications.

For the NSL-KDD dataset, the model recorded an accuracy of 98.38%, with good performance metric results. The FAR was 0.018, with a computational time of 47.8 seconds per epoch, reflecting a balanced and efficient choice for intrusion detection. On the UNSW-NB15 dataset, the model achieved an accuracy of 89.94%, along with outstanding experimental results. The FAR was 0.094, with a computational time of 75.8 seconds per epoch, indicating a robust performance with manageable computational demands.

Table 7. Comparison of proposed method (SAE-AM) with other methods of multi-class classification.

Ref.	Method+classifier	Datasets	Accuracy	Precision	Recall	F1-score	FAR	Time
[23]	CNN-RSA	CICIDS2017	99.91	99.91	99.88	99.91	N/A	N/A
		NSL-KDD	99.20	99.15	99.14	99.20	N/A	N/A
[36]	MHA+BiLSTM	CICIDS2017	99.08	100	99	99	N/A	N/A
		NSL-KDD	95.19	95	98	97	N/A	N/A
[34]	ADASYN+LightGBM	CICIDS2017	99.91	N/A	N/A	N/A	0.01	N/A
		NSL-KDD	92.57	N/A	N/A	N/A	6.41	N/A
		UNSW-NB15	85.89	N/A	N/A	N/A	14.79	N/A
[37]	CBF+CNN-BiLSTM	CICIDS2017	99.53	99.54	99.53	99.53	N/A	118.8 seconds
		NSL-KDD	99.40	99.40	99.40	99.40	N/A	71.14 seconds
		UNSW-NB15	82.30	82.76	82.30	79.61	N/A	127.65 seconds
[38]	SMOTE+CatBoost	CICIDS2017	99.72	N/A	N/A	N/A	N/A	N/A
		NSL-KDD	99.84	N/A	N/A	N/A	N/A	N/A
		UNSW-NB15	98.85	N/A	N/A	N/A	N/A	N/A
Proposed	**SAE-AM+MLP**	CICIDS2017	99.84	99.84	99.92	99.84	0.0007	70.4 seconds
		NSL-KDD	98.38	98.98	99.72	99.85	0.018	47.8 seconds
		UNSW-NB15	89.94	98.98	99.72	99.85	0.094	75.8 seconds

By employing these comprehensive evaluation criteria, the selected IDS models prove to be not only accurate and reliable but also efficient and adaptable to evolving threats.

6.3. Discussion

To justify the enhanced performance of the proposed model, we present a set of crucial design and methodological improvements.

First, the use of SAE for feature extraction enables the model to learn compact, non-redundant, and highly informative representations of network traffic. The imposed sparsity constraint ensures that only a limited number of neurons activate for any given input, helping the model to focus on the most critical patterns – especially those indicative of rare or subtle anomalies. Second, the integration of attention mechanisms – both channel and positional – enhances the model's ability to focus on the most relevant spatial and temporal features. This targeted emphasis improves the model's ability to distinguish between normal and anomalous traffic patterns, even in complex and high-dimensional datasets. Third, the end-to-end design of the proposed framework allows for the joint optimisation of feature extraction and classification, reducing error propagation and improving the overall learning efficiency. Unlike traditional multi-stage pipelines, this unified approach fosters

synergy between components. Additionally, by adopting class imbalance handling strategies, such as data-level balancing and sparsity-aware learning, the model significantly improves its detection capability for minority attack classes – a well-known challenge in IDS.

In terms of experimental results, SAE-AM demonstrates superior performance across multiple benchmark datasets. Specifically, in the binary classification task, the model achieves 100% accuracy, precision, recall, and F1-score on the CICIDS2017 dataset. For the NSL-KDD and UNSW-NB15 datasets, SAE-AM attains 99.81% and 99.83% accuracy, respectively, surpassing several recent state-of-the-art models – particularly in terms of FAR and training efficiency per epoch.

We have also provided detailed comparative performance tables – Table 6 for binary classification and Table 7 for multi-class classification – that benchmark SAE-AM against the existing methods. These comparisons clearly demonstrate the robustness, scalability, and generalisation capability of the proposed model, especially under challenging conditions involving class imbalance and complex traffic patterns. These results, combined with the carefully chosen design elements, justify the superior performance and practical applicability of SAE-AM in real-world intrusion detection scenarios.

Conclusions

The experimental results underscore the robustness and versatility of the SAE-AM methodology across various datasets and classification tasks. Achieving perfect 100% accuracy on the binary classification task of the CICIDS2017 dataset, along with high scores of 99.81% and 99.84% on the NSL-KDD and UNSW-NB15 datasets, respectively, highlights the model's potential for applications requiring high accuracy and minimal false alarms, crucial for operational efficiency in real-world scenarios.

Key advancements, such as the selective activation of SAE with L1 regularisation, minimizes resource consumption while effectively capturing pertinent patterns essential for intrusion detection. By enforcing sparsity, the model prioritises significant features, enhancing its capability to distinguish between normal and malicious network traffic. Furthermore, the integration of a dual attention mechanism – comprising positional and channel self-attention modules – greatly improves performance. The positional self-attention module aids in understanding spatial relationships between

features, while the channel self-attention module captures interactions across different feature channels, collectively enhancing detection accuracy and robustness against various intrusion scenarios.

Ablation experiments validate the model's efficacy as a reliable and efficient solution for network intrusion detection, as evidenced by consistently low FAR. Looking ahead, future research may explore further enhancements to the SAE-AM framework, potentially integrating more advanced attention mechanisms or evaluating its applicability in real-time intrusion detection scenarios. The demonstrated capabilities of the SAE-AM methodology position are as a promising tool for bolstering network security infrastructure and effectively addressing evolving cyber threats.

References

[1] Y. Yu, N. Bian, "An intrusion detection method using few-shot learning," *IEEE Access*, vol. 8, pp. 49730–49740, 2020, doi: 10.1109/ACCESS.2020.2980136.

[2] G. Andresini, A. Appice, N. Di Mauro, C. Loglisci, D. Malerba, "Multi-channel deep feature learning for intrusion detection," *IEEE Access*, vol. 8, pp. 53346–53359, 2020, doi: 10.1109/ACCESS.2020.2980937.

[3] Z. Yang, X. Liu, T. Li, D. Wu, J. Wang, et al., "A systematic literature review of methods and datasets for anomaly-based network intrusion detection," *Computers & Security*, vol. 116, Art. no. 102675, 2022, doi: 10.1016/j.cose.2022.102675.

[4] H. Mohammadian, A.A. Ghorbani, A. Habibi Lashkari, "A gradient-based approach for adversarial attack on deep learning-based network intrusion detection systems," *Applied Soft Computing*, vol. 137, Art. no. 110173, 2023, doi: 10.1016/j.asoc.2023.110173.

[5] Z. Ahmad, A.S. Khan, C. Shiang, F. Ahmad, "Network intrusion detection system: A systematic study of ma-chine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 01, pp. 1–29, 2021, doi: 10.1002/ett.4150.

[6] H. Zhang, L. Ge, G. Zhang, J. Fan, D. Li, C. Xu, "A two-stage intrusion detection method based on light gradient boosting machine and autoencoder," *Mathematical Biosciences and Engineering*, vol. 20, no. 4, pp. 6966–6992, 2023, doi: 10.3934/mbe.2023301.

[7] N.Y. Almusallam, Z. Tari, P. Bertok, A.Y. Zomaya, *Dimensionality reduction for intrusion detection systems in multi-data streams – A review and proposal of unsupervised feature selection scheme*. Cham: Springer, 2017, pp 467–487, doi: 10.1007/978-3-319-46376-6_22.

[8] J. Gu, L. Wang, H. Wang, S. Wang, "A novel approach to intrusion detection using SVM ensemble with feature augmentation," *Computers & Security*, vol. 86, pp. 53–62, 2019, doi: 10.1016/j.cose.2019.05.022.

[9] A. Nazir, R.A. Khan, "A novel combinatorial optimization based feature selection method for network intrusion detection," *Computers Security*, vol. 102, Art. no. 102164, 2021, doi: 10.1016/j.cose.2020.102164.

[10] S.M. Sohi, J.-P. Seifert, F. Ganji, "RNNIDS: Enhancing network intrusion detection systems through deep learning," *Computers Security*, vol. 102, Art. no. 102151, 2021, doi: 10.1016/j.cose.2020.102151.

[11] J. Zhang, Y. Ling, X. Fu, X. Yang, G. Xiong, R. Zhang, "Model of the intrusion detection system based on the integration of spatial-temporal features," *Computers & Security*, vol. 89, Art. no. 101681, 2020, doi: 10.1016/j.cose.2019.101681.

[12] J. Liu, Y. Gao, F. Hu, "A fast network intrusion detection system using adaptive synthetic oversampling and light GBM," *Computers & Security*, vol. 106, Art. no. 102289, 2021, doi: 10.1016/j.cose.2021.102289.

[13] B. Selvakumar, K. Muneeswaran, "Firefly algorithm based feature selection for network intrusion detection," *Computers & Security,* vol. 81, pp. 148–155, 2019, doi: 10.1016/j.cose.2018.11.005.

[14] J.O. Mebawondu, O.D. Alowolodu, J.O. Mebawondu, A.O. Adetunmbi, "Network intrusion detection system using supervised learning paradigm," *Scientific African*, vol. 9, Art. no. e00497, 2020, doi: 10.1016/j.sciaf.2020.e00497.

[15] Z. Wang, Y. Liu, D. He, S. Chan, "Intrusion detection methods based on integrated deep learning model," *Computers & Security*, vol. 103, Art. no. 102177, 2021, doi: 10.1016/j.cose.2021.102177.

[16] L. Ashiku, C. Dagli, "Network intrusion detection system using deep learning," *Procedia Computer Science*, vol. 185, pp. 239–247, 2021, doi: 10.1016/j.procs.2021.05.025.

[17] B.S. Bhati, C.S. Rai, B. Balamurugan, F. Al-Turjman, "An intrusion detection scheme based on the ensemble of discriminant classifiers," *Computers & Electrical Engineering*, vol. 86, Art. no. 106742, 2020, doi: 10.1016/j.compeleceng.2020.106742.

[18] J. Gu, S. Lu, "An effective intrusion detection approach using SVM with naïve Bayes feature embedding," *Computers & Security*, vol. 103, Art. no. 102158, 2021, doi: 10.1016/j.cose.2020.102158.

[19] M. Al-Qatf, Y. Lasheng, M. Al-Habib, K. Al-Sabahi, "Deep learning approach combining sparse auto encoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, pp. 52843–52856, 2018, doi: 10.1109/ACCESS.2018.2869577.

[20] I.H. Hassan, M. Abdullahi, M.M. Aliyu, S. Ali Yusuf, A. Abdulrahim, "An improved binary manta ray foraging optimization algorithm based feature selection and random forest classifier for network intrusion detection," *Intelligent Systems with Applications*, vol. 16, Art. no. 200114, 2022, doi: 10.1016/j.iswa.2022.200114.

[21] S. Das, M. Ashrafuzzaman, Frederick T. Sheldon, S. Shiva, "Ensembling supervised and unsupervised machine learning algorithms for detecting distributed denial of service attacks," *Algorithms*, vol. 17, no. 3, Art. no. 99, 2024, doi: 10.3390/a17030099.

[22] T.M. Nguyen, H. Hong-Phuc Vo, M. Yoo, "Enhancing intrusion detection in wireless sensor networks using agswo-catboost approach," *Sensors*, vol. 24, no. 11, Art. no. 3339, 2024, doi: 10.3390/s24113339.

[23] A. Dahou, M.A. Elaziz, S.A. Chelloug, M.A. Awadallah, M.A. Al-Betar, M.A.A Al-Qaness, A. Forestiero, "Intrusion detection system for IoT based on deep learning and modified reptile search algorithm," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, Art. no. 6473507, 2022, doi: 10.1155/2022/6473507.

[24] F. Yan, G. Zhang, D. Zhang, X. Sun, B. Hou, et al., "Tl-CNN-IDS: Transfer learning-based intrusion detection system using convolutional neural network," *The Journal of Supercomputing*, vol. 79, no. 15, pp. 17562–17584, 2023, doi: 10.1007/s11227-023-05347-4.

[25] J. Zhang, X. Zhang, Z. Liu, F. Fu, Y. Jiao, F. Xu, "A network intrusion detection model based on BILSTM with multi-head attention mechanism," *Electronics*, vol. 12, no. 19, Art. no. 4170, , 2023, doi: 10.3390/electronics12194170.

[26] F. Ullah, S. Ullah, G. Srivastava, J. Chun-Wei Lin, "IDSINT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic," *Digital Communications and Networks*, vol. 10, no. 1, pp. 190–204, 2024, doi: 10.1016/j.dcan.2023.03.008.

[27] O. Elnakib, E. Shaaban, M. Mahmoud, K. Emara, "EIDM: Deep learning model for Iot intrusion detection systems," *The Journal of Supercomputing*, vol. 79, no. 12, pp. 13241–13261, 2023, doi: 10.1007/s11227-023-05197-0.

[28] C. Zhang, M. Zhang, G. Yang, T. Xue, Z. Zhang, et al., "Three-way selection random forest optimization model for anomaly traffic detection," *Electronics*, vol. 12, no. 8, Art. no. 1788,, 2023, doi: 10.3390/electronics12081788.

[29] D. Wang, M. Nie, D. Chen, "BAE: Anomaly detection algorithm based on clustering and autoencoder," *Mathematics*, vol. 11, no. 15, Art. no. 3398, 2023, doi: 10.3390/math11153398.

[30] H.C. Altunay, Z. Albayrak, "A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks," *Engineering Science and Technology, an International Journal*, vol. 38, Art. no. 101322, 2023, doi: 10.1016/j.jestch.2022.101322.

[31] S.S.N. Chintapalli, S.P. Singh, J. Frnda, P.B. Divakarachari, V.L. Sarraju, P. Falkowski-Gilski, "OOA-modified bi-LSTM network: An effective intrusion detection framework for IoT systems," *Heliyon*, vol. 10, no. 8, Art. no. e29410, 2024, doi: 10.1016/j.heliyon.2024.e29410.

[32] H. Peng, C. Wu, Y. Xiao, "CBF-IDS: Addressing class imbalance using CNN-BILSTM with focal loss in network intrusion detection system," *Applied Sciences*, vol. 13, no. 21, Art. no. 11629, 2023, doi: 10.3390/app132111629.

[33] Y. Zhang, Z. Wang, "Feature engineering and model optimization-based classification method for network intrusion detection," *Applied Sciences*, vol. 13, no. 16, Art. no. 9363, 2023, doi: 10.3390/app13169363.

[34] J. Hu, L. Shen, G. Sun, "Squeeze-and-excitation networks,"in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR),* 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.

[35] I. Goodfellow, Y. Bengio, A. Courville, *Deep learning*. Cambridge, MA: MIT Press, 2016.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, et al., "Attention is all you need," in *Advances in neural information processing systems* (*NeurIPS*), 2017, pp. 5998–6008, doi: 10.48550/arXiv.1706.03762.

[37] A. Makhzani, *Unsupervised representation learning with autoencoders*. Toronto: University of Toronto, 2018.

[38] L. Dhanabal, S.P. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *International Journal of*

Advanced Research in Computer and Communication Engineering, vol. 4, no. 6, pp. 446–452, 2015.

[39] M.A. Ranzato, Y.-L. Boureau, Y. LeCun, "Sparse Feature Learning for Deep Belief Networks," *Advances in Neural Information Processing Systems* vol. 20, pp. 1185–1192, 2007. doi: 10.5555/2981562.2981711.

[40] S.-W. Lee, H. Mohammedsidqi, M. Mohammadi, S. Rashidi, A.M. Rahmani, et al., "Towards secure intrusion detection systems using deep learn-ing techniques: Comprehensive analysis and review," *Journal of Network and Computer Applications*, vol. 187, Art. no. 103111, 2021, doi: 10.1016/j.jnca.2021.103111.

[41] Q.V. Le, M. Aurelio Ranzato, R. Monga, M. Devin, G.S. Corrado, et al., "Building high-level features using large scale unsupervised learning," in *Proceedings of the 29th international conference on machine learning (ICML-12).* 2012, pp. 81–88.

[42] A. Ng, "Sparse autoencoder," CS294A Lecture Notes, Stanford University, 2011. [Online]. Available: https://web.stanford.edu/class/cs294a/sparseAutoencoder_2011new.pdf [Accessed: August 14, 2025].

[43] S. Dey, D. Mukhopadhyay, N. Das, "A deep learning approach to intrusion detection system using deep neural networks," *Journal of Supercomputing*, vol. 77, no. 3, pp. 2194–2215, 2021.

[44] M. Al-Qatf, Y. Lasheng, M. Al-Habib, K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, pp. 52843–52856, 2018. doi: 10.1109/ACCESS.2018.2869577.

[45] C. Yin, Y. Zhu, J. Fei, X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017, doi: 10.1109/ACCESS.2017.2762418.

[46] C. Liu, Y. Liu, Y. Yan, J. Wang, "An intrusion detection model with hierarchical attention mechanism," *IEEE Access*, vol. 8, pp. 67542–67554, 2020, doi: 10.1109/ACCESS.2020.2983568.

[47] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, H. Lu, "Dual attention network for scene segmentation," in G. Hoiem, H. Tu, Eds., *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, Piscataway, NJ, 2019, pp. 3141–3149.

[48] Y. Dong, Q. Liu, B. Du, L. Zhang, "Weighted feature fusion of convolutional neural network and graph attention network for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 31, pp. 1559–1572, 2022, doi: 10.1109/TIP.2022.3144017.

[49] A.S. Olufikayo, W. Sakpere, "Intrusion detection in cloud network environment: A multilayer perceptron model," *International Research Journal of Modernization in Engineering Technology and Science (IRJMETS)*, vol. 5, no. 12, pp. 181–187.

[50] M. Labonne, *Anomaly-based network intrusion detection using machine learning*. PhD thesis, Institut Polytechnique de Paris, Paris, France, 2020.

[51] Y. Yin, J. Jang-Jaccard, W. Xu, A. Singh, J. Zhu, et al., "IGRF-RFE: A hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset," *Journal of Big Data*, vol. 10, no. 1, Art. no. 15, Feb. 5, 2023, doi: 10.1186/s40537-023-00694-8.

[52] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruz-zaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, 2019, doi: 10.1186/s42400-019-0038-7.

[53] Z. Zhang, R. Jin, Z.-H. Zhou, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proceedings of the 23rd AAAI conference on artificial intelligence*, 2008.

[54] M. Mohammadi, A. Al-Fuqaha, M. Guizani, J. Oh, "A survey on deep learning techniques for network intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2475–2496, 2018, doi: 10.1109/COMST.2018.2844341.

[55] N. Shone, T.N. Ngoc, V.D. Phai, Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018, doi: 10.1109/TETCI.2017.2772792.

[56] R. Vinayakumar, K.P. Soman, P. Poornachandran, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[57] Y. Xia, X. Wang, H. Li, "A novel intrusion detection system based on a deep neural network," *The Scientific World Journal*, 2015.

[58] H. Taud, J. Mas, "Multilayer perceptron (MLP)," Chap. 27 in *Geomatic approaches for modeling land change scenarios.* Cham: Springer, 2018, pp. 451–455, doi: 10.1007/978-3-319-60801-3_27.

[59] P.P. Madani, N.N. Vlajic, "Robustness of deep autoencoder in intrusion detection under adversarial contamination," in *Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security (HotSoS '18)*, Raleigh, NC, USA, April 10–11, 2018, pp. 1–8, doi: 10.1145/3190619.3190637.