electronics

MDPI

Article

Towards Intelligent Threat Detection in 6G Networks Using Deep Autoencoder

Doaa N. Mhawi [1,*], Haider W. Oleiwi [2,*] and Hamed Al-Raweshidy [2]

[1] Computer Systems Department, Middle Technical University, Baghdad 8998+QHJ, Iraq
[2] Department of Electronic and Electrical Engineering, Brunel University London, London UB8 3PH, UK; hamed.al-raweshidy@brunel.ac.uk
[*] Correspondence: dododuaaenteesha@mtu.edu.iq (D.N.M.); haider.al-lami@brunel.ac.uk (H.W.O.)

Abstract

The evolution of sixth-generation (6G) wireless networks introduces a complex landscape of cybersecurity challenges due to advanced infrastructure, massive device connectivity, and the integration of emerging technologies. Traditional intrusion detection systems (IDSs) struggle to keep pace with such dynamic environments, often yielding high false alarm rates and poor generalization. This study proposes a novel and adaptive IDS that integrates statistical feature engineering with a deep autoencoder (DAE) to effectively detect a wide range of modern threats in 6G environments. Unlike prior approaches, the proposed system leverages the DAE's unsupervised capability to extract meaningful latent representations from high-dimensional traffic data, followed by supervised classification for precise threat detection. Evaluated using the CSE-CIC-IDS2018 dataset, the system achieved an accuracy of 86%, surpassing conventional ML and DL baselines. The results demonstrate the model's potential as a scalable and upgradable solution for securing next-generation wireless networks.

Keywords: 6G wireless communications; cybersecurity; deep learning; deep autoencoder; intrusion detection systems; machine learning

check for updates

Academic Editors: Tomotaka Kimura, Yu-Kuen Lai and Kouji Hirata

Received: 5 July 2025
Revised: 19 July 2025
Accepted: 23 July 2025
Published: 26 July 2025

Citation: Mhawi, D.N.; Oleiwi, H.W.; Al-Raweshidy, H. Towards Intelligent Threat Detection in 6G Networks Using Deep Autoencoder. *Electronics* **2025**, *14*, 2983. https://doi.org/10.3390/electronics14152983

1. Introduction

Although sixth-generation (6G) wireless communication systems remain in the conceptual and exploratory stages, they are projected to introduce transformative capabilities such as ultra-high data rates, extremely low latency, and intelligent automation across hyper-connected and virtualized environments [1,2]. These theoretical advancements are anticipated to support futuristic applications, including holographic communications, immersive extended reality (XR), autonomous systems, and large-scale Internet of Everything (IoE) integration [3]. However, as research into 6G progresses, it becomes clear that such innovations will significantly expand the network attack surface and introduce new cybersecurity challenges stemming from the increased architectural complexity and system heterogeneity [4–8].

Within this hypothetical framework, artificial intelligence (AI) is expected to play a foundational role in managing, optimizing, and securing 6G infrastructures. Nevertheless, the integration of AI also introduces risks, particularly when faced with unknown threats and adversarial behaviors in dynamic and decentralized virtual networks. Conventional intrusion detection systems (IDSs), typically designed for static or semi-dynamic architectures, are unlikely to remain effective in these high-speed, adaptive environments, especially when confronting zero-day or previously unseen attacks [9].

Sixth-generation networks are expected to support large-scale, heterogeneous devices and high-volume traffic flows managed through terahertz (THz) base stations, creating new operational dynamics in terms of computation, sensing, and communication [10]. Within this landscape, detecting sophisticated and previously unknown threats, particularly zero-day attacks, remains a significant challenge for conventional intrusion detection systems (IDSs) [11].

To explore a viable approach suitable for future 6G architectures, this study proposes a deep autoencoder-based intrusion detection model. Unlike traditional machine learning (ML) systems that depend on labeled data and are often vulnerable to class imbalance and high-dimensional noise, the proposed model leverages unsupervised learning to autonomously extract latent patterns from simulated network traffic. Anomalies are detected through reconstruction error, allowing the system to identify both known and novel threats in an efficient manner [12].

The contributions of this work are summarized as follows:

- Designing a deep autoencoder framework that learns latent patterns from raw traffic data, enabling unsupervised anomaly detection suited for 6G.
- Enhancing the detection of minority and rare attack types by mitigating class imbalance through feature generalization.
- Improving model accuracy and reducing false alarms using reconstruction-based scoring mechanisms.

This study evaluates the system using the SCE-CIC-IDS2018 dataset, which includes a diverse range of modern cyberattacks, to demonstrate its effectiveness in realistic conditions.

This model offers a lightweight and scalable solution for intelligent threat detection, positioning it as a viable security component for next-generation wireless networks. This work structure includes the related works presented in Section 2, while a background is provided in Section 3. Furthermore, Section 4 contains a detailed proposal and methodology, and Section 5 describes the dataset implementation. Section 6 discusses experimental findings. Section 7 summarizes the conclusions along with future directions.

2. Related Work

IDSs have gained the interest of academia and cybersecurity research communities, resulting in a plethora of articles published in recent years. Important research work in this field is discussed and summarized, focusing on imbalanced datasets and various types of ML and DL classifiers.

In [13], the authors explored the utilization of the random forest (RF) algorithm for feature selection (FS) integrated to various ML techniques, i.e., linear regression, k-nearest Neighbor (k-NN), CART, Bayesian methods, multi-layer perceptron (MLP), and XGBoost, developing an ID Support System. The experimental findings indicated that the MLP algorithm attained 96% accuracy.

In [14], researchers used the FS stage method called hybrid-correlation FS and forest-panelized attributes. In the classifier stage, they used four different classifiers (SVM, k-NN, RF, naïve Bayes). The prediction results reached 84%. This research faces complexity in the FS and classification steps. Furthermore, it prolonged training times in the training part.

The work In [15] utilized ensemble learning algorithms for detecting anomalies in communication networks. The system involved several stages and utilized diverse datasets. Additionally, the researchers utilized Adaboosting and bagging algorithms by using RF and support vector machine classification. It achieved high accuracy rates in three datasets; however, its complexity in execution and data training should be noted.

The research in [16] presented a novel meta-ML anomaly-detecting model for network traffic inspection. The model achieved high performance, although further improvements were still required to reduce false negative/positive rates and to improve the training process's computational efficiency.

In the study of [17], researchers employed the hybrid convolutional recurrent neural network (CRNN) as an IDS. They leveraged the strengths of CNN to extract the features and RNN to capture tentative features. The objective was to evaluate the effectiveness of the HCRNNIDS, and to achieve this, experiments were conducted using free ID data, particularly the contemporary/realistic CSE CIC-DS2018 dataset. Their simulations revealed the outperformance of the HCRNNIDS system over existing ID methodologies, achieving a remarkable malicious attack detection accuracy of 97.75% for the dataset when subjected to 10-fold cross-validation.

Table 1 presents a systematic comparison of related work, displaying datasets utilized, the achieved performance, and the employment of oversampling (OS) and undersampling (US) methodologies. However, the studies included have mainly employed outdated datasets, e.g., KDD-Cup99 and NSL-KDD, rendering the detection of the latest attacks difficult. Using previous datasets like KDD-Cup99 or NSL-KDD is insufficient for identifying evolving threats. Hence, a current dataset is required to develop a more effective IDS. In addition, the majority of earlier IDS deployments primarily tested the system's performance/accuracy to detect normal behavior and examine its efficiency.

Table 1. Related work.

Datasets and Refs.	Classifiers	Average Results	Over/Under Sampling
KDD-Cup99, [18].	K-Means and Sequential Minimal Optimization.	98.2	OS
NSL_KDD, [14].	SVM, KNN, NB, and RF.	99	No
UNSW_nb15, [15].	Ensemble learning methods.	97.3	US
CIC_IDS17, [16].	Metamodel.	98	OS/US
NSL-KDD, [19].	AE	90	OS

As far as authors are aware, this work attains an exceptional performance as compared with the existing systems (enhanced system to 86%). The proposed system surpasses state-of-the-art performance when tested using various datasets and significantly obtains remarkable rates for detection and false alarm/negative rates while requiring the least amount of time/complexity.

3. Background

3.1. Security Vulnerabilities in 6G and the Urgent Need for Intelligent Defense Mechanisms

The emergence of 6G wireless networks promises significant advancements in communication capabilities, including the use of THz frequency bands, intelligent reflecting surfaces (IRS), and ultra-dense network infrastructures. However, these innovations also expand the attack surface, exposing 6G systems to an increased variety of cyber threats such as jamming, spoofing, eavesdropping, and adversarial machine learning attacks [20]. Additionally, the inherently decentralized and heterogeneous architecture of 6G networks presents challenges in ensuring secure authentication, establishing trust, and protecting cross-domain data.

Traditional static security solutions lack the flexibility needed to address the dynamic and evolving nature of these threats. Consequently, there is an urgent demand for AI-driven, adaptive IDS capable of analyzing complex, high-volume network data in real time to identify novel and sophisticated attacks [20,21]. Recent studies emphasize the role of formal verification, behavioral modeling, and privacy-preserving AI techniques as critical components in securing 6G networks against both known vulnerabilities and zero-day attacks [22–24]. Such intelligent defense mechanisms are vital to safeguarding critical applications in 6G, including autonomous transportation, smart healthcare, and industrial automation, where reliability and trustworthiness are essential.

3.2. Datasets

Effective training and evaluation of IDS models require datasets that represent both normal and malicious network activities. Several publicly available datasets are commonly used in intrusion detection research:

KDD_Cup99 Dataset: Developed in 1998 by DARPA, this dataset contains around 5 million records, consisting of a mixture of normal and various attack traffic categorized into five groups. Despite its widespread use, it has limitations such as redundancy and outdated attack types.

NSL-KDD Dataset: An improved version of KDD_Cup99, this dataset removes redundant entries and provides a more balanced distribution of attack types, facilitating more accurate evaluation of IDS models [25].

CIC_IDS2017 Dataset: Created in 2017, this dataset features realistic network traffic with updated attack types and 86 features, including IP addresses, protocols, and timestamps. It addresses some limitations of earlier datasets by incorporating more current cyber threats [25].

CSE_CIC_IDS2018 Dataset: Developed by the Canadian Institute for Cybersecurity during 2018–2019, this dataset builds upon the standards set by CIC_IDS2017. It contains approximately 4.5 million records covering various attack types and benign activities, with detailed statistical features capturing traffic in both forward and backward directions [25]. The dataset's high quality, low duplication, and comprehensive attack coverage make it well-suited for evaluating modern IDS frameworks, particularly for environments like 6G Tables 2 and 3.

Table 2. SCE_CICIDS18 description.

Class_Name	Attacks_Number
Benign	2,856,035
Brute_Force	513
SQL injection	53
Infiltration	93,063
DoS	1,289,544
Bot	286,191
Total	4,525,399

Table 3. SCE_CICIDS18 information.

Datasets	Class1	Class2	Class3	Class4	Class5	Class6
KDD_Cup99	4,111,035	553,301	45,268	18,599	112	__
NSL_KDD	77,035	14,077	14,077	4,833	119	__
CIC_IDS17	2,311,035	453,438	15,967	1,966	35	21
CSE_CIC_IDS18	2,211,035	1,289,544	286,191	93,063	513	53

Table 3 shows that the datasets exhibit class imbalance, which can affect the accuracy of the system's evaluation. Thus, a balanced formulation is required to accurately measure the system's performance. The degree of imbalance can be quantified using (1) and can be utilized as a metric:

$$p = max(ci)/min(ci) \tag{1}$$

where *p* is the imbalanced ratio and *ci* is the dataset class.

The imbalance present in these datasets, quantified by the ratio between the most and least represented classes, poses challenges in evaluating IDS performance accurately. Addressing this imbalance is critical for robust detection, especially of rare attack types.

3.3. *Deep Learning Algorithms*

The AE architecture contains encoding/decoding operations: firstly, it converts the input data vector into a generally smaller form by the encoder. Secondly, it tries to decode the compressed vector to recover the original input by the decoder. The AE can extract important characteristics from unlabeled data after being trained in an unsupervised manner [26–29]. A traditional AE model with a single hidden layer is displayed in Figure 1.

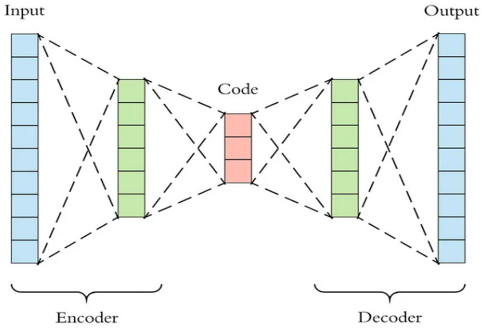

Figure 1. Traditional AE architecture.

A lower representation, e, is created from the input data vector z:

$$e = \varsigma \times (z \times W + b) \tag{2}$$

Here, W stands for the weighting matrix, b for the biasing vector, and ς for the encoder's activation function. The input (z) is then recreated from the encoded form (e) via decoding:

$$z = \zeta\ (eWT\ +\ b) \tag{3}$$

The reconstructed vector is represented by $\tilde{z}$, while the decoder's activation function is shown by ζ. An RNN LSTM unit is referred to as a memory block [29]. A typical LSTM architecture is shown in Figure 2. A cell (g), input (i), output (o), and forget gate(s) are all included (3). The layer of LSTM units can figure out the long-term relationships between different time-series data time steps. The hidden state (i.e., output state), which holds an output at the time step t, and the cell state, which retains knowledge gained from earlier time steps, are the two states of such an LSTM layer.

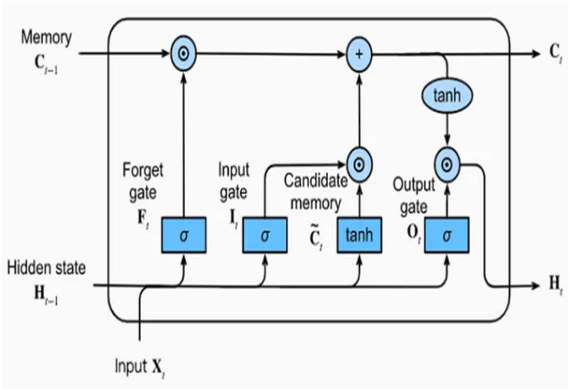

Figure 2. Traditional LSTM architecture.

The aforementioned gates are used to update the hidden and cell states at each time step *t*:

$$ct = lt \times \varTheta \times ct - 1 + it \times \varTheta \times gt \tag{4}$$

$$ht = ot \times \varTheta \times tanh(ct) \tag{5}$$

$$it = \sigma g(Wiz + Riht - 1 + bi) \tag{6}$$

$$ft = \sigma g\ (Wf z + Rfht - 1 + bf) \tag{7}$$

$$gt = tanh\ (Wgz + Rght - 1 + bg) \tag{8}$$

$$ot = \sigma g\ (Woz + Roht - 1 + bo) \tag{9}$$

where: *W* stands for the weighting matrix, *R* for the recurrent weighting matrix, and *b* for bias, whereas *σ* represents the sigmoid activation function.

4. Methodology

For developers, developing an IDS with accurate detection capabilities is considered the most critical factor in several recent relevant works. System accuracy is significantly influenced by class imbalance in the dataset, where a particular group constitutes the majority of the dataset and dominates. In similar situations, relying solely on statistical accuracy is not appropriate, as it becomes misleading. Consequently, to maximize attack effectiveness, skilled attackers may focus on a limited set of attack types when a severe imbalance exists between the majority and minority groups. Given that many modern IDSs rely on anomaly detection and employ six ML and DL (i.e., AE, and MLP) algorithms to identify anticipated data, several practical tools were developed previously. The system methodology is described as follows.

4.1. General Flowchart Structure of the Proposed System

Figure 3 demonstrates the general system steps to monitor untrustworthy traffic activities.

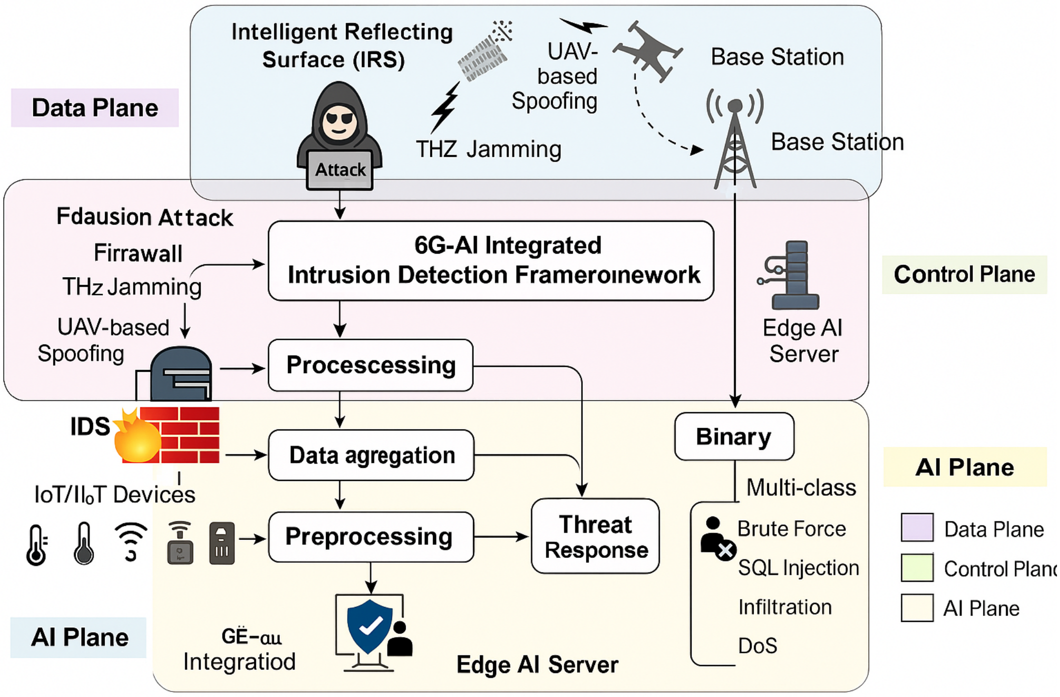

Figure 3. The general structure of the proposed system.

Figure 3 illustrates the system's stages for detecting suspicious and malicious traffic activities transmitted via 6G networks (anomalous behavior).

4.2. Preprocessing Stage

The dataset undergoes comprehensive preprocessing, consisting of three main steps: filtration, transformation, and normalization. It is then split into a 70% training set and a 30% testing set. The distribution processes are carried out by utilizing ML classifiers for robust and effective model training and evaluation.

1. Filtration removes outliers from the dataset using a median absolute deviation estimator (MADE):

$$MADE = P \times med\ (fj - \mid med(fj) \mid) \tag{10}$$

Here, med denotes the operator, *fj* denotes the attribute, and *P* (multiplicative) is a constant, equal to 1.4.

2. Transformation converts categorized features (e.g., protocol type, service, and flag) into enumerated values via the one-hot encoding function. For instance, the protocol type attribute (i.e., TCP, UDP, and ICMP) is encoded as binary-type vectors ([1,0,0], [0,1,0], and [0,0,1]), respectively.

$$fj > MADE \times P \tag{11}$$

3. Normalization applies Minimax scaling as:

$$xiValues = XVal - in/max - min \tag{12}$$

4.3. Feature Extraction Stage

It extracts the most relevant characteristic features. The proportion of zeros is calculated for the continuous dataset's feature. Features with more than 80% zeros are excluded from further analysis. A total of 100-dimensional feature vectors are constructed by combining the other 18 persisting features with 84 transformation vectors, while 20 variables construct the vectors of more specific features with related attributes. The resulting vector with the most correlated features is used as input to the classifier models in the next stage. Figure 4 presents a histogram of the dataset's null values.

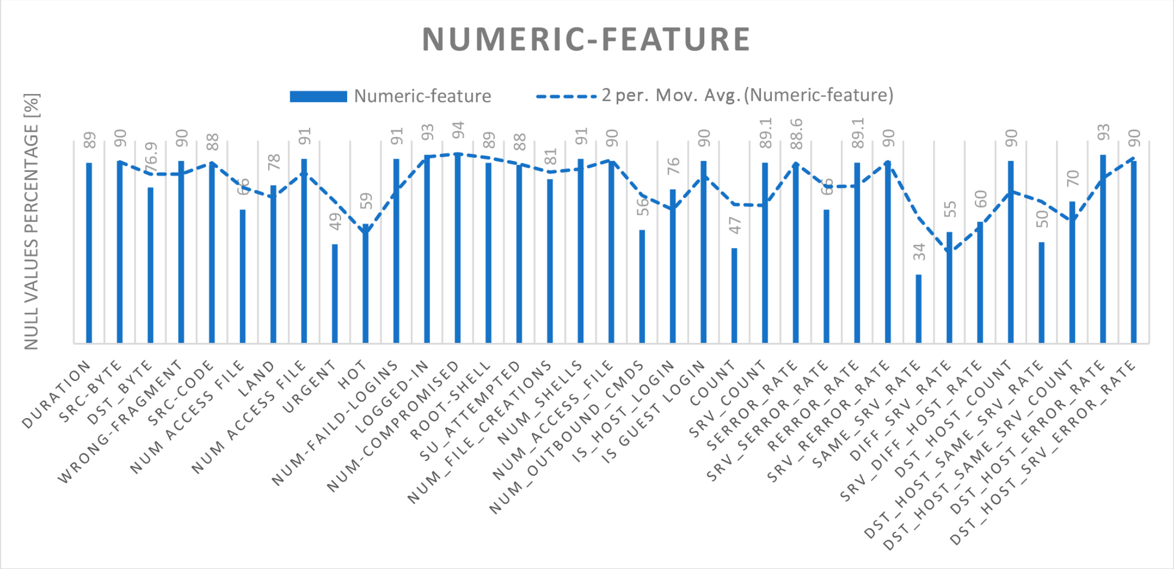

Figure 4. Histogram of the dataset's null values.

4.4. Classifiers Techniques Stage

To effectively distinguish between benign and malicious traffic activities in the SCE_CIC_IDS18 dataset, comprising various attack classes such as DoS, Bot, Infiltration, SQL Injection, and Brute Force, this study employs a deep autoencoder (DAE) model for feature extraction and classification. Additionally, the performance of the DAE is benchmarked against several machine learning and deep learning classifiers, including LSTM, SVM with linear and quadratic kernels, and discriminant analysis (DA) in both linear (LDA) and quadratic (QDA) forms. Algorithm 1 describes the outlines of DAE.

Algorithm 1 Outlines a deep autoencoder (DAE) designed for unsupervised feature extraction followed by supervised classification. Below is a clarified and enhanced step-by-step description of the architecture and training process.

```
Input:
z: Input feature vector (dimensionality = 100)
AE [100:50:100]: Autoencoder structure (encoder → bottleneck → decoder)
SCG: Scaled Conjugate Gradient optimization.
$\varsigma(s)$: Saturating linear activation function
MSE: Mean Squared Error loss function.
Output:
$\widetilde{z}$: Reconstructed input vector
$\hat{y}$: Class prediction using SoftMax (binary or multiclass)
Algorithm Steps:
Initialization:
Encoder: Compresses input vector $z \in \mathbb{R}$^100 into a 50-dimensional latent vector e.
Decoder: Reconstructs $\widetilde{z}$ from e using $\varsigma(s)$ activation.
Activation: Saturating linear units in decoder; SoftMax for final classification layer.
Unsupervised Pretraining:
Train the autoencoder using 100 epochs and SCG optimizer.
Loss function: MSE(z, $\widetilde{z}$;)
Stop training once least error $\leq$ 0.0083 is achieved.
Latent vector e retains essential features from input z.
Supervised Fine-Tuning:
The 50-dimensional vector e is fed into a fully connected SoftMax classifier.
Train the output classifier using labeled data for binary or multiclass decisions.
Use cross-entropy loss and early stopping to avoid overfitting.
Evaluation:
Model performance is assessed using accuracy, precision, recall, and F1-score.
Both binary and multiclass scenarios are supported.
```

Table 4 demonstrates the algorithm parameters and values.

Table 4. DAE Parameters with values.

Parameter	Value
Hidden Layers	3 (Input–Encoder–Decoder)
Neurons	100 → 50 → 100
Epochs	100
Optimizer	Scaled Conjugate Gradient
Loss Function	Mean Squared Error (MSE)
Activation Functions	Saturating Linear, SoftMax

Table 4. *Cont.*

Parameter	Value
Regularization	Early stopping + Dropout (if needed)
Architecture Objective	Minimize reconstruction error; extract latent features
Training Mode	Layer-wise pretraining → supervised fine-tuning

The DAE architecture was designed based on empirical analysis to ensure low reconstruction error and effective handling of high-dimensional network traffic data. A bottleneck layer with 50 neurons was selected to retain key discriminative features while removing irrelevant information, striking a balance between model complexity and generalization. During training, the reconstruction error, measured by MSE, reached an optimal value of 0.0083, validating the effectiveness of this configuration. The learned 50-dimensional feature representation was then passed to a fully connected SoftMax layer for binary or multiclass classification. A supervised fine-tuning phase was subsequently applied to further enhance classification performance, as illustrated in the corresponding architectural Figure 5.

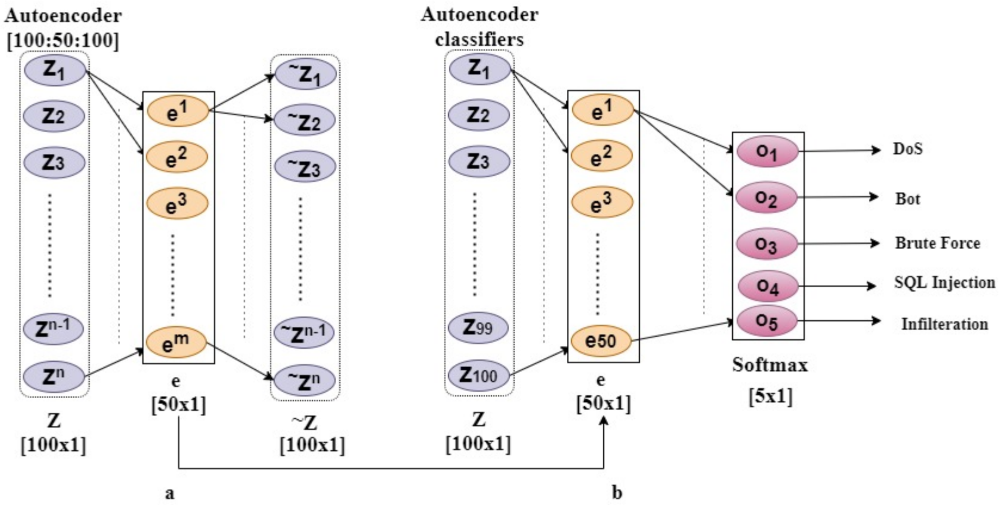

Figure 5. AE-based on classifiers: (**a**) after training AE [100:50:100] and (**b**) retrained using a supervised learning approach.

Designing the DAE architecture required thoughtful consideration of several factors to ensure efficient feature extraction and accurate data reconstruction. Key elements included the complexity and dimensionality of the input data, which influence the depth and width of the network. An optimal balance between compression and reconstruction layers was maintained to preserve critical features while reducing redundancy. The selection of activation functions played a vital role in enabling the model to learn nonlinear patterns effectively.

To avoid overfitting, the model's capacity was controlled through appropriate layer sizing and regularization techniques. The architecture was finalized through iterative experimentation, guided by performance metrics such as reconstruction accuracy and generalization ability. Moreover, task-specific needs, such as anomaly detection, and practical limitations, like computational resources and training time, shaped the final design. For comparative evaluation, other classification techniques were also implemented, as outlined in Algorithm 2.

Algorithm 2: Developing techniques (i.e., MLP, LSTM, SVM, and DA). Input: data-vector (z), i.e., AE [100:50:100], MSE, Wi, SCG.

```
•  Output: Reconstruct vector using (saturating linear activation function $\varsigma$ (s)) and
   measurements for binary and multi-classification forms.
•  Begin:
•  Initialization:
•  Fine-tuning strategy, least error = 0.0083, hidden neurons = 50, hyperparameter,
   Wi = 0, bias = 1, K = 1, learning rate = 0. 001.s, classifiers (Ci) (i.e., MLP, LSTM,
   LSVM, QSVM, LDA, and QDA).
•  For each Ci, apply the following steps:
•  Apply the following steps for C1/* where C1 is MLP*/.
•  For construction, apply the following steps:
•  Convert zi into a dimensional vector (e) (i.e., 50) using SCG (1)/* convert it into a
   smaller representation*/.
•  Extract important characteristics from unlabeled zi.
•  For reconstruction, apply the following steps:
•  Saturating linear activation function for ˜z using (2).
•  End for
•  Compute SoftMax function
•  Stopping criteria: when the error rate is less than 0.0083.
•  /* Note: MLP and AE designs have the same structure. */.
•  Apply the following steps for C2/* where C2 is LSTM*/.
•  Employed 50 cells to encode the input data.
•  Determine: Learning rate = 0.001, mini-batch size = 128.
•  The output was then input into a dense, fully connected layer with 2 or 4 neurons
   (a SoftMax activation function).
•  Compute ADAM (adaptive moment estimation).
•  Apply the following steps for C3/* where C3 is SVM*/.
•  The hyperplane with the most significant degree of class separation is found
   via SVM.
•  Compute the vectors Vi for both LSVM and QSVM.
•  Update the weight.
•  Apply the following steps for C4/* where C4 is DA*/.
•  Reducing dimensionality while maintaining strong class separability is the aim of
   DA./* It projects the data samples into a lower-dimensional space to maximize
   class separability and minimize sample dispersion within a class*/.
•  End for
•  Compute the measurements for binary and multi-classification forms.
•  End.
```

Algorithm 2 outlines the steps of alternative techniques, beginning with the MLP classifier. It is a supervised learning technique used to train this feed-forward neural network. Figure 6 shows the suggested MLP classifier mechanism. The MLP employs a SoftMax output layer for classifications and a hidden layer with 50 neurons. Moreover, the LSTM classifier (shown in Figure 7) is structured with an input layer and an output (dense) layer.

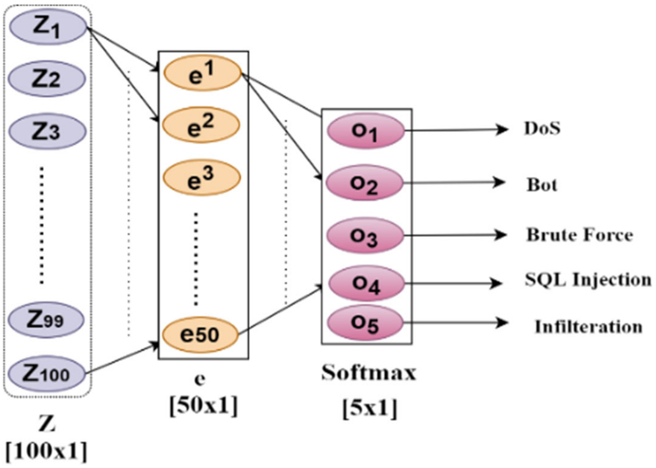

Figure 6. MLP structure.

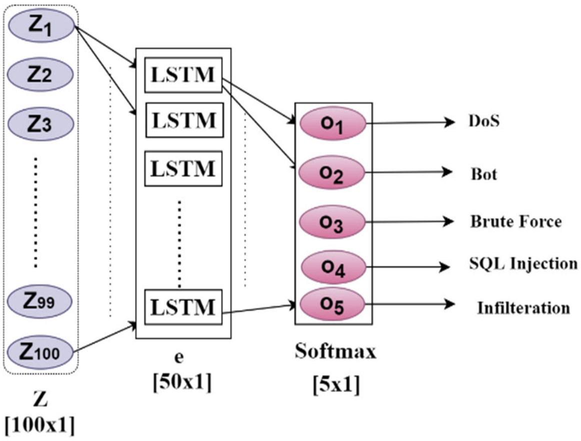

Figure 7. LSTM structure.

5. Implementation

The proposed system is developed and evaluated using the SCE-CIC-IDS2018 dataset, which is partitioned into 70% for training and 30% for testing. Performance assessment is conducted based on a set of extracted features. Subsequently, a pioneering ID approach is introduced using the DAE algorithm, with a comprehensive performance comparison against other contemporary techniques across binary and multiclass classification contexts.

The implementation is carried out in Python 3.8 within the Google Colab environment, utilizing the Scikit-learn and TensorFlow libraries. Experiments are run on a system equipped with an Intel Core i7 processor and a 64-bit Windows 11 operating system.

To evaluate the effectiveness of the proposed method, several key performance metrics are employed, including precision, detection rate (DR), F1-score, and overall accuracy.

$$Precision\ (P) = Tp/(Tp + Fp) \tag{13}$$

$$Recall = Tp/(Tp + Fn) \tag{14}$$

$$F1\text{-}Score = 2(Precision \times Recall)/(Precision + Recall) \tag{15}$$

$$Accuracy = (Tp + Tn)/(Tp + Fp + Tn + Fn) \tag{16}$$

True negatives (Tn refer to the number of instances correctly identified as normal, while true positives (Tp) represent the number of abnormal instances accurately detected.

False positives (*Fp*) are normal traffic patterns incorrectly classified as anomalous, whereas false negatives (*Fn*) are abnormal traffic patterns mistakenly identified as normal.

6. Results, Discussion, and Evaluation

The DAE-architected system performance is expressed in two classification forms: binary class and multi-class. Table 5 demonstrates the accuracy of the proposed classifiers in binary and multi-classification forms.

Table 5. Comparison accuracy of the different proposed classifiers in binary-accuracy and multi-accuracy forms.

Classifiers	Binary-Accuracy	Multi-Accuracy
DAE	84.2%	81.3%
MLP	84.2%	81.3%
LSTM	84.2%	81.3%
LSVM	84.2%	81.3%
QSVM	84.2%	81.3%
LDA	84.2%	81.3%
QDA	84.2%	81.3%

6.1. Binary Class Form Result

Table 6 illustrates the results of DAE with different developing classifiers in the binary classification forms (i.e., normal and abnormal) with various measurements.

Table 6. Measurements of different classifiers in binary classification forms.

Classes	DAE	MLP	LSTM	LSVM	QSVM	QDA	CNN-LSTM
Normal	84.7%	80.1%	73.1	78.1	80%	77%	79.6%
Abnormal	85.1%	79.2%	76%	81%	76.3%	78.9%	77.9%
			Recall				
Normal	84.2%	81.3%	76.5%	84%	81.1%	75%	75%
Abnormal	82.1%	79.1%	78.1%	71%	77.2%	82%	82%
			Accuracy				
Normal	84.7%	82%	77.8%	80%	87.9%	77.9%	80.2%
Abnormal	82.1%	81%	79.2%	75%	73.8%	84.1%	83.1%
			F1-Score				
Normal	84%	82.3%	78.9%	86%	88.2%	78.7%	87.6%
Abnormal	82%	71.4%	79.4%	78%	79.3%	84.5%	85.3%
			FP Rate				
Normal	87.6%	83.5%	78.5%	83%	87.2%	76.2%	78.7%
Abnormal	86.1%	70.9%	80.1%	77.1%	80.1%	85.5%	80.5%
			FN Rate				
Normal	85.6%	79.1%	73.5%	80%	80%	74.2%	81%
Abnormal	83.1%	73%	77%	73%	81%	79%	82%

The evaluation results in Table 6 reveal that the denoising autoencoder (DAE) classifier outperforms traditional and deep learning models in anomaly detection tasks. Among the classical models, the quantum support vector machine (Q-SVM) demonstrated superior performance over linear SVM (L-SVM), achieving higher F1-scores for both normal and abnormal classifications. Similarly, quadratic discriminant analysis (QDA) showed better overall performance than linear discriminant analysis (LDA), though LDA was more effective in identifying normal samples. The multi-layer perceptron (MLP) and long short-term memory (LSTM) models also yielded competitive results, with LSTM achieving a 78% average F1-score. However, the DAE classifier surpassed all others, reaching an average F1-score of 82% and an accuracy of 86%. The study also utilized ROC curves and AUC metrics to visualize classifier performance, highlighting DAE's superior capability in distinguishing between normal and anomalous instances.

Figure 8 offers a dual-view assessment of classifier performance. On the left, the bar chart compares F1-score and accuracy across models, with the DAE outperforming all others in both metrics. Q-SVM surpasses L-SVM in F1-score, while QDA shows a slight edge over LDA in anomaly detection. LSTM maintains moderate, balanced performance. On the right, the ROC curve highlights the DAE's strong classification capability, achieving an AUC of 0.91, indicating excellent distinction between normal and anomalous data. Collectively, these findings confirm the DAE's robustness, achieving the best accuracy (86%), highest F1-score (80%), and outstanding AUC, making it especially suitable for intrusion detection in next-generation 6G cybersecurity systems.

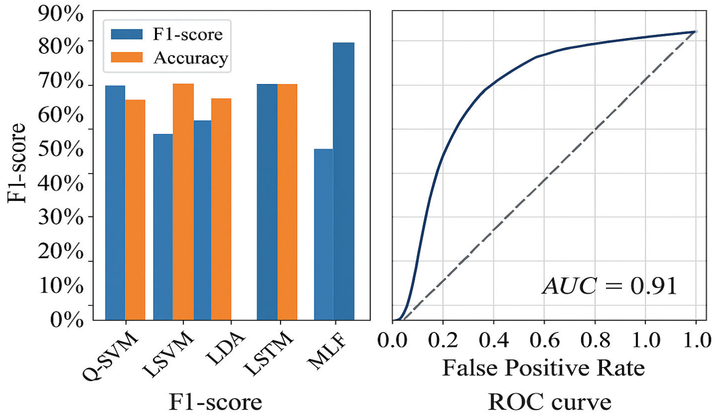

Figure 8. F1-score and ROC curve for all the classifiers.

6.2. Multi-Class Form Result

The findings of multi-classification forms are detailed in Table 7.

Table 7. Measurements of different classifiers in (multi-class) classification forms.

Precision Measurement						
	DAE	MLP	LSTM	LSVM	QSVM	QDA
DoS	84.7%	80.1%	73.1	78.1	80%	77%
SQL-Injection	84.7%	80.1%	73.1	78.1	80%	77%
Infiltration	84.7%	80.1%	73.1	78.1	80%	77%
Brute force	84.7%	80.1%	73.1	78.1	80%	77%
Bot	85.1%	79.2%	76%	81%	76.3%	78.9%
DoS	84.7%	80.1%	73.1	78.1	80%	77%
			Recall measurement			
DoS	84.2%	81.3%	76.5%	81%	81.1%	75%
SQL-Injection	84.2%	81.3%	76.5%	80%	81.1%	76%
Infiltration	84.2%	81.3%	76.5%	82%	81.1%	75%
Brute force	84.2%	80.6%	76.5%	80%	81.1%	75%
Bot	82.1%	79.1%	76.2%	71%	77.2%	82%
			Accuracy measurement			
DoS	84.7%	80%	77.8%	80%	87.9%	77.9%
SQL-Injection	84.7%	79%	78%	80%	87.9%	77.9%
Infiltration	84.7%	81%	76.3%	81%	79%	77.9%
Brute force	84.7%	80%	75.9%	80%	75.9%	71.6%
Bot	82.1%	83%	79.2%	75%	73.8%	80.1%

Table 7. *Cont.*

Precision Measurement						
			F1-Score measurement			
DoS	84%	82.3%	78.9%	76%	80.2%	78.7%
SQL-Injection	86%	71.4%	79.4%	78%	79.3%	84.5%
Infiltration	85%	70.4%	77.3%	78%	79.1%	82.5%
Brute force	86.1%	74.1%	78.4%	79%	73%	78%
Bot	87.8%	75%	74.9%	76%	78%	81%

Simulation experiments confirm that the DAE delivers the strongest overall performance. However, the impact of false-positive and false-negative alarms is determined by the security objectives, the protected network's characteristics, and the operational context. Balancing these error types is therefore a persistent challenge in ID design. Continuous refinement—through algorithm fine-tuning, threshold recalibration, and alignment with the organization's risk tolerance and operational requirements—is essential for maintaining an effective IDS.

6.3. *Reduction of Complexity and Procedural Time*

This section outlines the efficiency benefits of using DAE for intrusion detection. By compressing input data, DAEs effectively reduce dimensionality, which lowers computational demands during both training and inference. The model also enhances performance by extracting only the most relevant features, eliminating noise and redundancy. These optimizations contribute to faster classification and anomaly detection processes.

Additionally, reduced data dimensions lead to lower memory usage and quicker data processing. The system's architecture supports parallel computation, enabling deployment on GPUs and distributed platforms for improved speed.

To assess the system's computational efficiency, several metrics are proposed: training and inference times, resource usage (CPU, GPU, and memory), and scalability across datasets of different sizes and complexities. These measures provide a comprehensive evaluation of the system's performance in real-world and resource-constrained environments.

6.4. *Comparison with Relevant Works*

Table 8 illustrates the proposed study compared with other related studies in terms of datasets, FS techniques, used classifiers, and accuracy results.

Table 8 shows the best accuracy result of 86% when applying DAE to the SCE_CIC_IDS18 dataset with the feature extraction method.

Table 8. Comparison with other related studies.

Refs.	Datasets	Feature-Selection	Classifiers	Accuracy	Precision	Recall	F1-Score
[30]	NSL-KDD	CFS_FPA	KNN	84%	83%	82.7%	84%
		RF	Hybrid classifiers	82%	81.7%	82.3%	82%
[14]	UNSW_NB15	CFS_FPA		84.5%	84.2%	83.5%	84.3%
	CIC_IDS17			84.9%	84%	84.1%	85%
[16]	NSL-KDD	Ensemble learning methods		81%	81.3%	82.5%	81%
	UNSW_NB15	RF	Meta-Model	82%	81.7%	82.3%	82%
	CIC_IDS18			84%	83%	82.7%	84%
	SCE_CIC_IDS17			85%	84%	84.6%	85%
[15]	NSL_KDD	Correlation with random forest	Anomaly detection using ensemble learning	84.1%	84.2%	83.5%	84.3%
	SCE_CIC_IDS17			84.7%	84.6%	84%	84.1%

Table 8. *Cont.*

Refs.	Datasets	Feature-Selection	Classifiers	Accuracy	Precision	Recall	F1-Score
[31]	UNSW-NB15 and X-IIoTID	-	Standard CNN layers followed by LSTM temporal layers	82%	83.7%	81.3%	84%
[19]	NSL_KDD	Correlation with FS	AE	85%	85.1%	84.9%	85%
[32]	NSL_KDD	RFE, trained in Keras/TensorFlow	CNN-LSTM	82%	83%	84%	84.8%
Proposed system	SCE_CIC_IDS18	Feature extraction, not selection	DAE	86%	86.3%	85.7%	86%

7. Conclusions and Future Directions

This study presented a DAE-based ID framework that demonstrated superior performance on the CSE-CIC-IDS2018 dataset when benchmarked against several advanced classifiers, including LSVM, QSVM, LSTM, and MLP. The DAE architecture effectively captured intricate and nonlinear patterns in network traffic, enabling accurate differentiation between normal and malicious behavior, which traditional models often miss due to their limited representation capacity.

As wireless networks evolve toward 6G—with its requirements for ultra-reliable low-latency communication, massive device connectivity, and intelligent automation—the need for adaptive and efficient threat detection becomes critical. The proposed DAE model shows strong potential in addressing these needs due to its ability to autonomously learn abstract features from raw traffic data, supporting both scalability and real-time deployment.

Moving forward, future research will focus on expanding this work in several key directions. This includes integrating the proposed IDS into realistic 6G network environments, e.g., base station-level implementation or edge-cloud collaboration models. Additionally, further experimentation will be conducted using diverse and evolving datasets, e.g., CIC-FlowMeter and other IoT-oriented traffic repositories, to validate cross-domain generalizability. Furthermore, attention will be given to optimizing the system for resource-constrained environments, enhancing its suitability for deployment in embedded or hardware-accelerated platforms. Overall, this DAE-based detection framework offers a promising and intelligent solution to the growing challenges of ID in next-generation wireless systems. It lays the foundation for building more secure, adaptive, and autonomous cybersecurity defenses in the emerging 6G era.

Author Contributions: Conceptualization, D.N.M.; Methodology, H.W.O.; Software, D.N.M.; Validation, D.N.M. and H.W.O.; Formal analysis, D.N.M.; Investigation, D.N.M. and H.W.O.; Resources, H.W.O.; Data curation, D.N.M.; Writing—original draft, D.N.M.; Writing—review & editing, D.N.M. and H.W.O.; Supervision, H.W.O. and H.A.-R.; Funding acquisition, H.W.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

References

1. Zawish, M.; Dharejo, F.A.; Khowaja, S.A.; Raza, S.; Davy, S.; Dev, K.; Bellavista, P. AI and 6G Into the Metaverse: Fundamentals, Challenges and Future Research Trends. *IEEE Open J. Commun. Soc.* **2024**, *5*, 730–778. [CrossRef]
2. Ozpoyraz, B.; Dogukan, A.T.; Gevez, Y.; Altun, U.; Basar, E. Deep Learning-Aided 6G Wireless Networks: A Comprehensive Survey of Revolutionary PHY Architectures. *IEEE Open J. Commun. Soc.* **2022**, *3*, 1749–1809. [CrossRef]
3. Oleiwi, H.W.; Al-Raweshidy, H. Cooperative SWIPT THz-NOMA/6G Performance Analysis. *Electronics* **2022**, *11*, 873. [CrossRef]
4. Porambage, P.; Gur, G.; Osorio, D.P.M.; Liyanage, M.; Gurtov, A.; Ylianttila, M. The Roadmap to 6G Security and Privacy. *IEEE Open J. Commun. Soc.* **2021**, *2*, 1094–1122. [CrossRef]
5. Khan, N.A.; Schmid, S. AI-RAN in 6G Networks: State-of-the-Art and Challenges. *IEEE Open J. Commun. Soc.* **2024**, *5*, 294–311. [CrossRef]
6. Ahmad, I.; Rodriguez, F.; Kumar, T.; Suomalainen, J.; Jagatheesaperumal, S.K.; Walter, S.; Asghar, M.Z.; Li, G.; Papakonstantinou, N.; Ylianttila, M.; et al. Communications Security in Industry X: A Survey. *IEEE Open J. Commun. Soc.* **2024**, *5*, 982–1025. [CrossRef]
7. Naeem, F.; Ali, M.; Kaddoum, G.; Huang, C.; Yuen, C. Security and Privacy for Reconfigurable Intelligent Surface in 6G: A Review of Prospective Applications and Challenges. *IEEE Open J. Commun. Soc.* **2023**, *4*, 1196–1217. [CrossRef]
8. Wang, M.; Zhu, T.; Zhang, T.; Zhang, J.; Yu, S.; Zhou, W. Security and privacy in 6G networks: New areas and new challenges. *Digit. Commun. Netw.* **2020**, *6*, 281–291. [CrossRef]
9. Mhawi, D.N.; Hashim, S.H. Proposed Hybrid EnsembleLearninig algorithms for an Efficient Intrusion Detection System. *Int. J. Comput. Commun. Eng.* **2022**, *22*, 73–84.
10. Oleiwi, H.W.; Al-Raweshidy, H. SWIPT-Pairing Mechanism for Channel-Aware Cooperative H-NOMA in 6G Terahertz Communications. *Sensors* **2022**, *22*, 6200. [CrossRef]
11. Hasan, M.K.; Ghazal, T.M.; Saeed, R.A.; Pandey, B.; Gohel, H.; Eshmawi, A.A.; Abdel-Khalek, S.; Alkhassawneh, H.M. A review on security threats, vulnerabilities, and counter measures of 5G enabled Internet-of-Medical-Things. *IET Commun.* **2022**, *16*, 421–432. [CrossRef]
12. Nancy, P.; Muthurajkumar, S.; Ganapathy, S.; Kumar, S.V.N.S.; Selvi, M.; Arputharaj, K. Intrusion detection using dynamic feature selection and fuzzy temporal decision tree classification for wireless sensor networks. *IET Commun.* **2020**, *14*, 888–895. [CrossRef]
13. Chimphlee, S.; Chimphlee, W. Machine learning to improve the performance of anomaly-based network intrusion detection in big data. *Indones. J. Electr. Eng. Comput. Sci.* **2023**, *30*, 1106–1119. [CrossRef]
14. Mhawi, D.N.; Aldallal, A.; Hassan, S. Advanced Feature-Selection-Based Hybrid Ensemble Learning Algorithms for Network Intrusion Detection Systems. *Symmetry* **2022**, *14*, 1461. [CrossRef]
15. Oleiwi, H.W.; Mhawi, D.N.; Al-Raweshidy, H. MLTs-ADCNs: Machine Learning Techniques for Anomaly Detection in Communication Networks. *IEEE Access* **2022**, *10*, 91006–91017. [CrossRef]
16. Oleiwi, H.W.; Mhawi, D.N.; Al-Raweshidy, H. A Meta-Model to Predict and Detect Malicious Activities in 6G-Structured Wireless Communication Networks. *Electronics* **2023**, *12*, 643. [CrossRef]
17. Assi, J.H.; Sadiq, A.T. NSL-KDD dataset Classification Using Five Classification Methods and Three Feature Selection Strategies. *J. Adv. Comput. Sci. Technol. Res.* **2017**, *7*, 15–28.
18. Chandra, A.; Khatri, S.K.; Simon, R. Filter-based Attribute Selection Approach for Intrusion Detection using k-Means Clustering and Sequential Minimal Optimization Techniq. In Proceedings of the 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, 4–6 February 2019; pp. 740–745. [CrossRef]
19. Ieracitano, C.; Adeel, A.; Morabito, F.C.; Hussain, A. A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. *Neurocomputing* **2020**, *387*, 51–62. [CrossRef]
20. You, I.; Kim, J.; Pawana, I.W.A.J.; Ko, Y. Mitigating Security Vulnerabilities in 6G Networks: A Comprehensive Analysis of the DMRN Protocol Using SVO Logic and ProVerif. *Appl. Sci.* **2024**, *14*, 9726. [CrossRef]
21. Scalise, P.; Boeding, M.; Hempel, M.; Sharif, H.; Delloiacovo, J.; Reed, J. A Systematic Survey on 5G and 6G Security Considerations, Challenges, Trends, and Research Areas. *Futur. Internet* **2024**, *16*, 67. [CrossRef]
22. Li, Y. Adversarial Attack Model Based on Deep Neural Network Interpretability and Artificial Fish Swarm Algorithm. *Int. J. Electron. Secur. Digit. Forensics* **2024**, *16*, 614–632. [CrossRef]
23. Li, G.; Ota, K.; Dong, M.; Wu, J.; Li, J. DeSVig: Decentralized Swift Vigilance against Adversarial Attacks in Industrial Artificial Intelligence Systems. *IEEE Trans. Ind. Inform.* **2020**, *16*, 3267–3277. [CrossRef]
24. Sun, H.; Liu, Y.; Al-Tahmeesschi, A.; Nag, A.; Soleimanpour, M.; Canberk, B.; Arslan, H.; Ahmadi, H. Advancing 6G: Survey for Explainable AI on Communications and Network Slicing. *IEEE Open J. Commun. Soc.* **2025**, *6*, 1372–1412. [CrossRef]
25. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009, Ottawa, ON, Canada, 8–10 July 2009.

26. Karatas, G.; Demir, O.; Sahingoz, O.K. Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset. *IEEE Access* **2020**, *8*, 32150–32162. [CrossRef]
27. Wang, X.; Lyu, B.; Guo, C.; Xu, J.; Zukerman, M. A Base Station Sleeping Strategy in Heterogeneous Cellular Networks Based on User Traffic Prediction. *IEEE Trans. Green Commun. Netw.* **2024**, *8*, 134–149. [CrossRef]
28. Wei, Z.; Qu, H.; Jiang, W.; Han, K.; Wu, H.; Feng, Z. Iterative Signal Processing for Integrated Sensing and Communication Systems. *IEEE Trans. Green Commun. Netw.* **2023**, *7*, 401–412. [CrossRef]
29. Ikram, S.T.; Cherukuri, A.K.; Poorva, B.; Ushasree, P.S.; Zhang, Y.; Liu, X.; Li, G. Anomaly Detection Using XGBoost Ensemble of Deep Neural Network Models. *Cybern. Inf. Technol.* **2021**, *21*, 175–188. [CrossRef]
30. Mhawi, D.N.; Hashem, P.; Soukaena, H. Proposed Hybrid Correlation Feature Selection Forest Panalized Attribute Approach to advance IDSs. *Karbala Int. J. Mod. Sci.* **2021**, *7*, 15. [CrossRef]
31. Altunay, H.C.; Albayrak, Z. A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks. *Eng. Sci. Technol. Int. J.* **2023**, *38*, 101322. [CrossRef]
32. Bamber, S.S.; Katkuri, A.V.R.; Sharma, S.; Angurala, M. A hybrid CNN-LSTM approach for intelligent cyber intrusion detection system. *Comput. Secur.* **2025**, *148*, 104146. [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.