future internet

MDPI

Article

A Scalable Hybrid Autoencoder–Extreme Learning Machine Framework for Adaptive Intrusion Detection in High-Dimensional Networks

Anubhav Kumar [1], Rajamani Radhakrishnan [1], Mani Sumithra [2], Prabu Kaliyaperumal [1], Balamurugan Balusamy [3] and Francesco Benedetto [4,*]

[1] School of Computer Science and Engineering, Galgotias University, Greater Noida 203201, India; dr.anubhavkumar@gmail.com (A.K.); prof.rrk8@gmail.com (R.R.); mega.prabu@gmail.com (P.K.)
[2] Department of Information Technology, Panimalar Engineering College, Chennai 600123, India; msumithra@panimalar.ac.in
[3] Associate Dean-Students, Shiv Nadar University, Delhi-NCR Campus, Noida 201305, India; kadavulai@gmail.com
[4] Signal Processing for TLC and Economics, University of Roma Tre, 00154 Rome, Italy
* Correspondence: francesco.benedetto@uniroma3.it

Abstract: The rapid expansion of network environments has introduced significant cybersecurity challenges, particularly in handling high-dimensional traffic and detecting sophisticated threats. This study presents a novel, scalable Hybrid Autoencoder–Extreme Learning Machine (AE–ELM) framework for Intrusion Detection Systems (IDS), specifically designed to operate effectively in dynamic, cloud-supported IoT environments. The scientific novelty lies in the integration of an Autoencoder for deep feature compression with an Extreme Learning Machine for rapid and accurate classification, enhanced through adaptive thresholding techniques. Evaluated on the CSE-CIC-IDS2018 dataset, the proposed method demonstrates a high detection accuracy of 98.52%, outperforming conventional models in terms of precision, recall, and scalability. Additionally, the framework exhibits strong adaptability to emerging threats and reduced computational overhead, making it a practical solution for real-time, scalable IDS in next-generation network infrastructures.

Keywords: intrusion detection system; autoencoder; extreme learning machine; cloud continuum; edge–fog–cloud orchestration; scalable iot security; real-time threat detection

Academic Editors: Dimitrios Dechouniotis and Ioannis Dimolitsas

Received: 18 April 2025
Revised: 9 May 2025
Accepted: 13 May 2025
Published: 15 May 2025

Citation: Kumar, A.; Radhakrishnan, R.; Sumithra, M.; Kaliyaperumal, P.; Balusamy, B.; Benedetto, F. A Scalable Hybrid Autoencoder–Extreme Learning Machine Framework for Adaptive Intrusion Detection in High-Dimensional Networks. *Future Internet* **2025**, *17*, 221. https://doi.org/10.3390/fi17050221

1. Introduction

The exponential growth of digital communication networks has significantly transformed the way information flows within interconnected systems. These networks, ranging from personal devices to enterprise infrastructures, manage vast volumes of data in real time, playing a crucial role in facilitating daily operations across various sectors [1,2]. Cloud platforms, corporate networks, and other digital frameworks together constitute the core of this interconnected digital environment [3]. However, as the complexity and scale of these systems increase, their susceptibility to various cyber threats also rises. Modern cyberattacks, characterized by their sophistication and persistence, pose significant challenges to network security [4]. Traditional security measures are increasingly strained by threats like Advanced Persistent Threats (APTs), zero-day exploits, and constantly evolving attack strategies. For example, APTs are characterized by prolonged, targeted intrusion efforts that are challenging to detect due to their subtle and covert tactics. Similarly, zero-day vulnerabilities exploit undiscovered software flaws, enabling attackers to

evade established security measures [5,6]. Additionally, dynamic attack strategies, in which attackers adjust their methods during an operation, make detection and mitigation more challenging [7]. Compounding these issues is the high-dimensional nature of traffic data in modern networks, which further complicates the process of identifying threats. Every data packet can contain numerous attributes, such as source and destination IP addresses, along with behavioral indicators, resulting in an overwhelming volume of information that conventional approaches find difficult to handle [8]. Although real-time anomaly detection is essential, traditional IDS often face challenges due to their high computational demands and constraints, limiting their overall effectiveness. The increasing complexity of network traffic necessitates the creation of scalable and efficient IDS capable of real-time operation, adapting to emerging attack patterns, and managing the growing volume and diversity of data. Today, cybersecurity threats are not only varied but also more focused and destructive [9,10]. For instance, Distributed Denial of Service (DDoS) attacks flood systems with fraudulent traffic, making them unreachable for authorized users. Brute force attacks methodically target authentication systems, and malware infections infiltrate networks to steal, damage, or erase data [11,12]. To highlight the scope of these threats, this research uses the CSE-CIC-IDS2018 dataset [13], a detailed collection containing more than 16 million examples of both benign and malicious network traffic. This dataset contains a range of attack families, including Botnet, Web Attacks, Infiltration, and DDoS, offering an accurate portrayal of the challenges encountered by contemporary networks. This diversity within the dataset highlights the need for resilient and flexible detection systems capable of effectively identifying both known and emerging threats. IDS act as the primary defense against such threats by constantly analyzing network traffic to detect any abnormal or suspicious behavior [14], as illustrated in Figure 1. However, traditional IDS methods are quickly losing their effectiveness. Signature-based systems depend on predefined attack patterns, which limits their ability to detect new or evolving threats. Although rule-based approaches offer flexibility, they typically necessitate significant manual configuration and face challenges in scaling to meet the increasing complexity of network environments. Both approaches struggle with high-dimensional traffic data, resulting in processing challenges and an increased likelihood of false positives or negatives.

Figure 1. Elements of Intrusion Detection System.

To address these limitations, IDS based on machine learning have surfaced as a viable alternative. These systems utilize sophisticated algorithms to analyze and categorize network traffic, allowing them to adapt to changing attack patterns without depending entirely on predefined rules or signatures. Machine learning models are particularly effective at detecting anomalies in high-dimensional datasets, allowing for the identification of new threats. Although promising, current machine learning methods encounter challenges related to scalability, computational efficiency, and real-time applicability, particularly when handling the large volumes of traffic common in modern networks. This study tackles these challenges by introducing a Scalable Hybrid AE-ELM IDS Framework, a new intrusion detection model that combines the advantages of Autoencoders (AE) and Extreme Learning Machines (ELM). Autoencoders are used for dimensionality reduction, condensing high-dimensional traffic data into a more compact form while preserving essential features and eliminating noise. This process greatly alleviates the computational load, enhancing the framework's suitability for real-time applications. After dimensionality

reduction, the simplified feature set is processed by ELM, a machine learning model recognized for its fast training time and excellent generalization ability [15]. ELM effectively classifies traffic into benign or malicious categories with high accuracy, providing a reliable and efficient detection system. An overview of the proposed framework is illustrated in Figure 2. The proposed framework's performance is validated using the CSE-CIC-IDS2018 dataset under realistic conditions. This diverse dataset, with its high-dimensional features, guarantees that the framework is capable of managing the complexity and scale associated with modern network traffic. Utilizing cutting-edge machine learning methods, the framework bridges key gaps in current IDS solutions, providing a scalable, efficient, and precise intrusion detection approach.

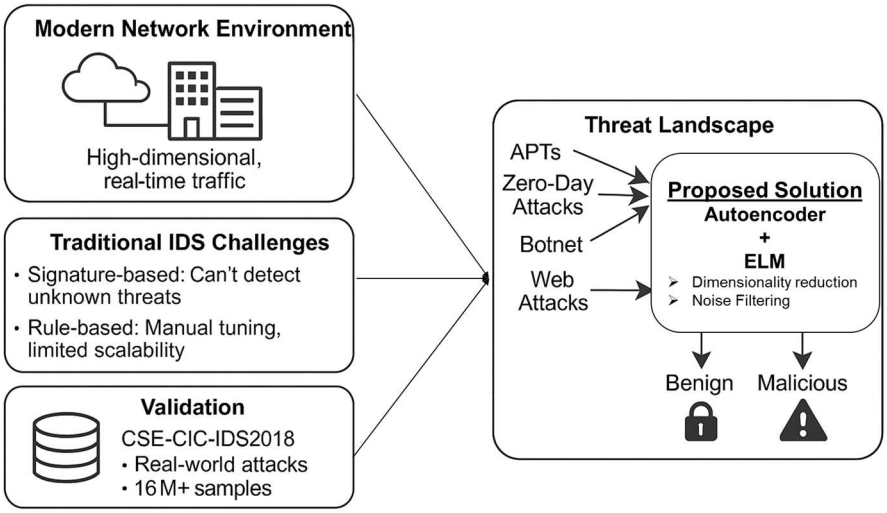

Figure 2. Motivation and overview of the proposed scalable hybrid AE+ELM IDS framework.

The key contributions of this study include:

1. Proposing a novel integration of Autoencoders and Extreme Learning Machines (AE-ELM) for intrusion detection, optimizing both scalability and efficiency.
2. Improving high-dimensional data handling via tuned Autoencoders that reduce computational cost without sacrificing feature integrity.
3. Employing ELM for fast, accurate classification, enabling real-time anomaly detection with minimal computational overhead.
4. Validating the framework on the CSE-CIC-IDS2018 dataset, using multiple evaluation metrics to demonstrate robustness and adaptability.
5. Demonstrating superior performance over traditional IDS approaches, showcasing effectiveness in addressing modern network security threats.

2. Related Works

Recent studies have explored various intelligent techniques for enhancing intrusion detection performance across multiple domains and datasets. A scalable machine learning-based Network IDS (ML-NIDS) was proposed in [16], employing SMOTE to address class imbalance, the Extra Trees Classifier for feature selection, and Extreme Learning Machine (ELM) for classification. The model was evaluated using the UNSW-NB15 dataset, which includes nine types of attacks, and achieved an accuracy of 98.43%, along with high ROC AUC and balanced precision-recall performance. However, the approach encounters limitations such as a high false negative rate for minority classes and a reliance on labeled data, which constrains its effectiveness in detecting previously unseen attack types.

A hybrid approach to feature selection for intrusion detection was developed in [17], integrating Genetic Algorithm (GA) with ELM. The methodology involves preprocessing

data with SMOTE-Tomek Links, optimizing input weights for ELM using GA, and performing classification with Support Vector Machine (SVM). Tested on the IoT_ToN and UNSW-NB15 datasets, the model demonstrated an accuracy of 99% for IoT_ToN and 86% for UNSW-NB15, surpassing Decision Tree, Random Forest, and Gradient Boosting in terms of precision and recall metrics. The approach is constrained by its reliance on supervised learning and significant computational demands.

An IDS tailored for IoT networks was proposed in [18], incorporating preprocessing with StandardScaler (scikit-learn v1.2.2) and SMOTE (imbalanced-learn v0.10.1), one-hot encoding for categorical data, and a single hidden-layer neural network based on ELM. The model was tested on the NSL-KDD and Distilled Kitsune datasets, achieving F1-scores of 0.9541 and 0.9555, respectively. Although the system exhibited strong performance and efficiency, its limitations include diminished effectiveness in nonlinear environments, restricted adaptability to novel attacks without retraining, and dependence on structured data.

A real-time intrusion detection system for IoT environments was created in [19], leveraging the PySpark framework and employing five machine learning techniques: Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors, and Extreme Gradient Boosting (XGB). The system attained an accuracy of 98.89% with XGB and a prediction time of 0.0311 s using Random Forest (RF). The IoT-23 dataset, comprising 1,444,674 records across five categories of attacks, was utilized for evaluation. Challenges include dependency on specific datasets, difficulties in scaling to larger datasets, and the possibility of neglecting significant features during the feature selection process.

The MMLCS-UAV framework, designed for cybersecurity in UAV networks to detect and classify intrusions, was introduced in [20]. The approach incorporates Quantum Invasive Weed Optimization (QIWO) for feature selection, Weighted Regularized Extreme Learning Machine (WRELM) for detecting intrusions, and Swallow Swarm Optimization (SSO) for optimizing parameters. The model demonstrated superior performance with an accuracy of 99.59%, a precision of 98.76%, and a recall of 98.77%, surpassing the GA-DBN and BOA-DBN models. Limitations of the approach include dependence on particular datasets, absence of real-time testing in UAV networks, and unaddressed scalability in dynamic environments.

A bagging ensemble method for network intrusion detection was introduced in [21], employing Extremely Randomized Trees (ERT) for feature selection and using k-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Classification and Regression Trees (CART) as base models. The KNN model was fine-tuned using Bayesian Optimization. When tested on the NSL-KDD dataset, the BO-KNN-Bagging model achieved an accuracy of 82.48%, a precision of 72.22%, a recall of 96.40%, and an F1-score of 82.58%, surpassing other ensemble methods. Limitations involve challenges related to generalization, scalability, and class imbalance.

A network IDS that utilizes Stochastic Gradient Descent (SGD) and Gaussian Naive Bayes (GNB), incorporating PCA and SVD for dimensionality reduction, was introduced in [22]. The WSN-DS dataset, which focuses on DoS attacks, reached an accuracy of 96% and an F1 score of 97%, while the WUSTL EHMS 2020 IoMT dataset recorded 87% accuracy and 93% F1 score. SGD outperformed GNB, enhancing real-time efficiency and minimizing false positives. However, the approach has limitations, including high computational costs and reliance on data quality and feature selection.

A hierarchical IDS, utilizing ELM as the main classifier, with Grey Wolf Optimization (GWO), Archimedes Optimization Algorithm (AOA), and Honey Badger Algorithm (HBA) employed for hyperparameter tuning and feature selection, was introduced in [23]. The system achieved an accuracy of 98.93% with a false alarm rate of 0.51% on the UNSW-NB15 dataset, and 99.63% accuracy with a false alarm rate of 0.01% on the CICIDS2017

dataset. Although ELM outperformed traditional classifiers, the use of meta-heuristic optimization introduces additional computational overhead and complexity, particularly in real-time applications.

A Smart Attacks Learning Machine Advisor (SALMA) system to protect smart cities from cyber threats was proposed in [24]. The system uses a five-level classification model based on Extreme Learning Machine (ELM) algorithms, classifying anomalies in Software-Defined Networking (SDN) environments. The conceptual framework involves a multilayer architecture for SDN-based smart cities, where flow statistics from SDN controllers feed into ELM classifiers. The NSL-KDD and KDDCUP99 benchmark datasets are employed for training and testing. Experimental results achieved high detection accuracies of 95% and 99.2%, respectively, outperforming several existing IDS models. Limitations include reliance on static datasets, which may not reflect evolving real-time smart city traffic, and the focus on known attack types, which might limit zero-day attack detection. Challenges include scaling the model for large, dynamic smart city infrastructures and integrating it with diverse IoT ecosystems.

Recent advancements in ELM-based models have addressed challenges in fault diagnosis under imbalanced data conditions. For example, the research in [25] proposed a novel chiller fault diagnosis framework that integrates an improved GAN with an enhanced Deep ELM, effectively mitigating the effects of data imbalance and improving diagnostic accuracy. Similarly, in another study, the authors introduced a new resampling technique combined with an improved ensemble ELM to enhance fault detection in imbalanced chiller data scenarios [26]. These studies demonstrate the adaptability and robustness of ELM in handling real-world, skewed datasets. Although these studies effectively handle imbalanced data in fault diagnosis scenarios using advanced ELM-based architecture, the present work leverages a Hybrid Autoencoder-ELM framework for intrusion detection, emphasizing efficient feature reduction and classification without requiring extensive data resampling or synthetic generation, thus maintaining the integrity of the original traffic data.

An efficient intrusion detection mechanism tailored for IoT environments using paired autoencoders combined with ELM was introduced in [27]. The proposed conceptual framework involves a two-layer detection system: first, data are partitioned into regions (quasi-normal, quasi-attack, divergence, and undetermined) based on predictions from multiple autoencoders trained separately on normal and various attack data. In the second layer, optimal autoencoders are selected for precise detection within each region. The NSL-KDD dataset was employed for experimentation. Results demonstrated improved overall accuracy (94.89%) and F1-Score (95.23%), outperforming baseline models. Notably, accuracy in the complex divergence region improved by 20.68%. The model maintained a lightweight nature, with faster training times and smaller model sizes than traditional techniques like SVM or Random Forest. Limitations include reliance on a single dataset and the increased model count, which may slightly affect resource overhead in constrained IoT environments.

GAOR, a Genetic Algorithm-based optimization technique to enhance the robustness of machine learning models against adversarial attacks in communication networks, was proposed in [28]. The conceptual framework integrates adversarial sample generation using genetic algorithms with tree-based models like Random Forest (RF) and XGBoost (v1.7.5, Python). The CIC-IDS2019 and 5G-NIDD datasets were employed, covering both legacy and 5G-specific traffic scenarios. GA was used for hyperparameter optimization and adversarial sample generation, aiming to improve model resilience. Results showed that GAOR models maintained high accuracy (up to 99.94%) and improved robustness, though performance varied against advanced adversarial attacks like ZooAttack with LightGBM (v3.3.5, Python). Limitations include computational overhead from GA opera-

tions and sensitivity to noisy or imbalanced data. Challenges involve balancing accuracy and robustness, and adapting to increasingly sophisticated attack vectors in evolving communication environments.

A Binary Chimp Optimization Algorithm with Machine Learning-based Intrusion Detection (BCOA-MLID) for securing IoT-assisted Wireless Sensor Networks (WSNs) was proposed in [29]. The conceptual framework combines four stages: data normalization, feature selection using BCOA, intrusion detection using a Class-specific Cost Regulation Extreme Learning Machine (CCR-ELM), and parameter optimization via a Sine Cosine Algorithm (SCA). The WSN-DS dataset was utilized, comprising 374,661 samples across five attack categories. Results showed that BCOA-MLID achieved superior accuracy (99.63%), sensitivity (97.91%), specificity (99.67%), and faster computation times compared to models like XGBoost and KNN-AOA. Limitations include dependency on supervised learning and limited attack categories in the dataset. Challenges involve handling dynamic traffic patterns, addressing data imbalance, and extending detection coverage to diverse IoT-based cyberattacks beyond the tested scenarios.

The reviewed studies collectively demonstrate significant progress in the field of intrusion detection using a variety of machine learning and optimization techniques. These approaches have shown promising results across diverse datasets and network environments, particularly in addressing challenges like class imbalance, real-time detection, and high-dimensional data. However, limitations such as dependency on labeled datasets, lack of adaptability to novel threats, and scalability constraints in dynamic environments persist. Table 1 summarizes recent ELM-based intrusion detection techniques from existing studies, highlighting these advances and their respective limitations. These gaps underscore the need for more robust, flexible, and intelligent IDS frameworks. Motivated by these findings, the present study proposes a two-stage anomaly detection mechanism combining dimensionality reduction and classification techniques, aiming to enhance detection performance while addressing the aforementioned limitations.

Table 1. Summary of recent ELM-based intrusion detection techniques in existing studies.

Ref.	Technique(s) Used	Dataset(s) Used	Metrics	Key Limitations	Planned Improvements/Advancements
[16]	SMOTE + Extra Trees + ELM	UNSW-NB15	98.43%, strong ROC AUC	High FNR for minority classes, needs labeled data	Introduce unsupervised and hybrid approaches to reduce labeling needs
[17]	GA + SMOTE-Tomek + ELM + SVM	IoT_ToN, UNSW-NB15	99% (IoT_ToN), 86% (UNSW-NB15)	Supervised only, high computational load	Adopt scalable and semi-supervised methods with fewer dependencies
[18]	ELM + SMOTE + one-hot encoding	NSL-KDD, Distilled Kitsune	F1: 0.9541, 0.9555	Weak in nonlinear environments, lacks adaptability	Use deep models with nonlinear handling and less preprocessing overhead
[19]	PySpark + ML (XGB, RF, etc.)	IoT-23	98.89% (XGB), RF prediction time: 0.0311 s	Dataset-dependent, scalability issues	Incorporate dataset-agnostic feature selection and real-time streaming
[20]	QIWO + WRELM + SSO	Custom UAV dataset	Accuracy: 99.59%, Precision: 98.76%, Recall: 98.77%	No real-time testing, dataset specific	Extend model testing across standard, diverse datasets for generalization

Table 1. *Cont.*

Ref.	Technique(s) Used	Dataset(s) Used	Metrics	Key Limitations	Planned Improvements/Advancements
[21]	ERT + BO-tuned Bagging (KNN, SVM, CART)	NSL-KDD	Accuracy: 82.48%, Recall: 96.40%	Generalization and imbalance issues	Integrate imbalance-aware training with adaptive learning layers
[22]	PCA/SVD + SGD/GNB	WSN-DS, WUSTL EHMS	Accuracy: 96% (WSN-DS), 87% (IoMT)	Data quality sensitive, high computational cost	Apply dimensionality reduction using Autoencoders to preserve variance
[23]	ELM + GWO, AOA, HBA for tuning and feature selection	UNSW-NB15, CICIDS2017	Accuracy: 98.93–99.63%, FAR: 0.01–0.51%	High computational complexity due to meta-heuristics	Replace meta-heuristics with light-weight deep encoders for efficiency
[24]	SALMA: SDN + ELM-based 5-level architecture	NSL-KDD, KDDCUP99	95–99.2%	Static datasets, zero-day limitations, IoT integration	Incorporate dynamic threat models with zero-day detection capabilities
[27]	Paired Autoencoders + ELM	NSL-KDD	Accuracy: 94.89%, F1: 95.23%	More models → higher overhead, single dataset dependency	Use a unified deep Autoencoder with ELM based classifier
[28]	GAOR: Genetic Algorithm + RF/XGBoost	CIC-IDS2019, 5G-NIDD	Accuracy: up to 99.94%	GA overhead, adversarial performance variation	Replace GA with adaptive deep learning for evolving threat adaptation

3. Materials and Methods

This section describes the approach used to develop a scalable Hybrid AE-ELM IDS framework aimed at enhancing the security of network environments. The framework integrates the capabilities of Autoencoders (AE) and Extreme Learning Machines (ELM) to effectively handle large-scale and intricate network traffic data. Dimensionality reduction is achieved using AE, which employs the Adam optimizer to reduce reconstruction error while preserving essential features, thereby simplifying the data's complexity. This is essential for managing the high dimensionality and diverse characteristics of network traffic. ELM, renowned for its fast training speed and strong generalization capabilities with low computational cost, is used to classify network traffic and identify anomalies. Its high efficiency makes it ideal for real-time intrusion detection applications where quick processing is crucial. The system is built for scalability, allowing it to manage high volumes of network traffic and swiftly adapt to emerging attack patterns, ensuring sustained effectiveness as the network environment evolves. The framework utilizes the CSE-CIC-IDS2018 dataset, which offers a detailed and accurate representation of current network activities and attack scenarios, ensuring the system's reliability in real-world use cases. The effectiveness of the proposed framework is assessed using standard evaluation metrics, including precision, recall, specificity, F-measure, and accuracy. These metrics are employed to benchmark the proposed system against established methods, confirming its effectiveness and showcasing its ability to manage diverse network traffic data more efficiently. Figure 3 provides a detailed illustration of the proposed framework, including data collection, preprocessing, dimensionality reduction, and anomaly detection.

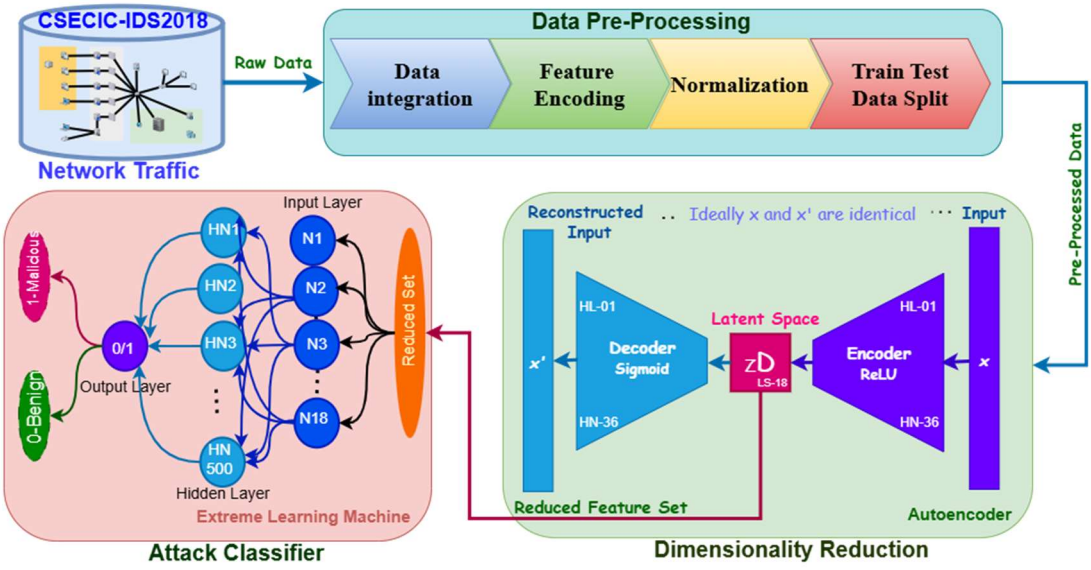

Figure 3. Operational architecture of proposed scalable hybrid AE-ELM IDS framework.

3.1. Data Preprocessing

This research utilizes the CSE-CIC-IDS2018 dataset, a comprehensive collection comprising 16,232,943 instances and 80 features, including seven attack families, fifteen attack types, and diverse network traffic patterns. This dataset is ideal for evaluating IDS models as it accurately mirrors real-world network activities and potential attack situations, offering a solid basis for model development and evaluation. The dataset undergoes four carefully executed preprocessing steps to prepare it for analysis. Data integration is achieved by combining ten separate CSV files with the pd.concat() function to form a single dataset, enabling smooth analysis. Feature encoding is used to standardize attributes such as IP addresses and timestamps, ensuring consistency and uniformity throughout the dataset. Next, normalization is applied using Min-Max scaling to bring all features within the [0, 1] range, ensuring a consistent scale and improving the performance of machine learning algorithms by reducing the effects of varying feature magnitudes. The dataset is split into training and testing sets using the split_data() function, with a 70:30 ratio, to assess the model's performance on unseen data effectively. To address the issue of class imbalance, a balanced subset of the original dataset was constructed by applying stratified random sampling, ensuring an equal number of normal and attack traffic instances. This approach facilitates fairer model training and evaluation, avoiding bias towards the majority class. These preprocessing steps improve the dataset's quality, ensuring its effectiveness for intrusion detection tasks, optimizing the accuracy and efficiency of the IDS framework in detecting malicious activities, and enhancing its scalability to handle larger and more diverse datasets.

3.2. Dimensionality Reduction Using Autoencoder

In this research, dimensionality reduction is performed using Autoencoders (AE), an unsupervised neural network model that effectively reduces high-dimensional data into a more compact, lower-dimensional form. The AE model consists of three key elements: the encoder, the latent space, and the decoder [30]. Figure 4 illustrates the process flow of preprocessing and dimensionality reduction using the Autoencoder. The encoder compresses the input data, transforming the original 80 features into a more compact, lower-dimensional representation. The optimized 18 features in this reduced representation are selected to maintain essential information while ensuring computational efficiency. The latent space acts as a condensed and informative representation of the original data,

capturing key patterns and potential anomalies. The decoder reconstructs the data from the latent space, with the reconstruction error being a crucial measure for detecting anomalies. A large reconstruction error indicates that there may be anomalies in the network traffic. The AE model is trained with the Adam optimizer, which speeds up convergence, using a learning rate of 0.001 over 37 epochs. The model uses ReLU activation functions in the hidden layers to apply non-linear transformations, while a sigmoid activation function is used in the output layer to maintain accuracy in the reconstruction.

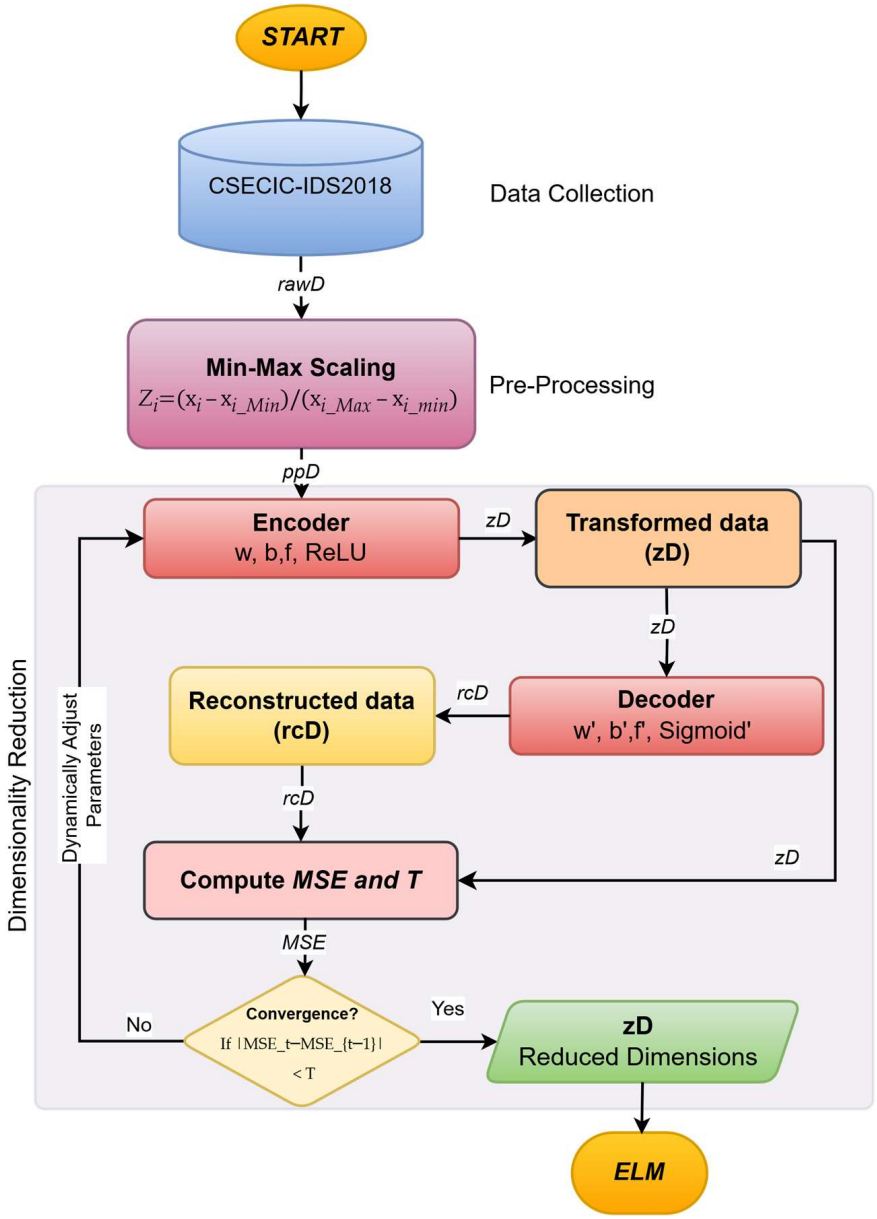

Figure 4. Process flow of preprocessing and dimensionality reduction using autoencoder.

Dynamic Thresholding for Convergence

To improve training efficiency and ensure reliable convergence, dynamic thresholds are applied based on the reconstruction error (Mean Squared Error-MSE) of the Autoencoder. The threshold value T is computed dynamically using the following statistical Formula (1):

$$T = \mu MSE + k * \sigma MSE \quad (1)$$

where, μMSE and σMSE denote the running mean and standard deviation of the reconstruction error over training iterations, and k is a sensitivity constant, which adjusts the threshold's responsiveness.

The training is considered to have converged when the change in reconstruction loss between two successive epochs satisfies the following Condition (2):

$$|MSE_t - MSE_{(t-1)}| < T \quad (2)$$

This dynamic thresholding mechanism helps stabilize the training process by ensuring that further adjustments to the Autoencoder's parameters are only made when significant changes in reconstruction error are detected, thereby avoiding unnecessary updates when the model is already converging. The overall process of dynamic thresholding and training is summarized in Algorithm 1 below.

Algorithm 1: Autoencoder-Based Feature Reduction with Dynamic MSE Thresholding

Begin
1. Load the dataset (rawD):
rawD ← load_data('CSE-CIC-IDS2018')
2. Apply preprocessing (ppD) on the data:
ppD ← min_max_scale(rawD)
3. Initialize Autoencoder parameters:
weights, biases ← initialize_parameters()
4. Set initial threshold (T):
T ← μ_MSE + k · σ_MSE
(where μ_MSE is the mean and σ_MSE is the standard deviation of MSE over initial epochs, k is a sensitivity constant)
5. Repeat until convergence:
a. Encode the input to obtain compressed representation (zD):
zD ← encoder(ppD, weights, biases, activation = ReLU)
b. Decode the latent representation to reconstruct input (rcD):
rcD ← decoder(zD, weights', biases', activation = Sigmoid)
c. Compute reconstruction error (Mean Squared Error—MSE):
MSE ← mean_squared_error(ppD, rcD)
d. Update parameters using the Adam optimizer:
θ ← θ − η · Adam(∇_θ MSE)
(where θ represents weights and biases, η is the learning rate)
e. Recompute threshold dynamically:
Maintain list MSE_history ← [MSE_1, MSE_2, . . ., MSE_t]
T ← mean(MSE_history) + k · std(MSE_history)
f. Check for convergence:
If |MSE_t − MSE_{t − 1}| < T:
→ break
6. Output:
zD (reduced feature representation)
End

3.3. Intrusion Detection Using Extreme Learning Machine

Following the dimensionality reduction via the Autoencoder, the Extreme Learning Machine (ELM) classifier is employed to identify network intrusions by classifying traffic as either benign or malicious. ELM is a fast and effective neural network model, making it ideal for real-time applications that require high speed and efficiency. The dimensionality-

reduced feature set, obtained from the Autoencoder, is fed into the input layer of the ELM, which contains 18 nodes, ensuring efficient processing of the data. The hidden layer, consisting of 500 nodes, uses ReLU activation functions to capture complex patterns within the data. The output layer uses a sigmoid activation function, enabling binary classification to differentiate between benign and malicious traffic. A significant advantage of ELM is its distinctive training method, which initializes the hidden layer weights randomly and calculates the output weights using a pseudo-inverse closed-form solution. This significantly shortens the training time in comparison to conventional neural networks, while still ensuring strong generalization capabilities. Hyperparameter tuning is performed to optimize performance, with a focus on adjusting the number of hidden nodes and the threshold value. A grid search technique is used to optimize the number of hidden nodes, beginning with 100, while an initial threshold of 0.5 is set to improve classification accuracy. Traffic is categorized based on the output of ELM, where values that meet or exceed the threshold are classified as malicious, and those below the threshold are identified as benign. This approach merges the dimensionality reduction capabilities of Autoencoders with the fast classification power of ELM, creating an efficient, scalable, and precise framework for protecting complex network environments from evolving cyber threats.

4. Results

This section presents the experimental results of the Hybrid AE-ELM IDS framework, assessed using a subset of the CSECIC-IDS2018 dataset, which includes 5,496,470 samples with a balanced distribution of normal and attack traffic instances, divided into a 70:30 train-test ratio. The framework was evaluated using Extreme Learning Machine (ELM) with four different sets of hidden nodes (100, 300, 500, and 600). The experiments were performed in a Python 3.0 environment with T4 GPU (manufactured by NVIDIA, Santa Clara, CA, USA) support on Google Colab to ensure efficient computation. Table 2 presents the hyperparameter configuration used in the proposed model, detailing the setup parameters.

Table 2. Hyperparameter configuration of the proposed model.

Hyperparameters	Configuration
Hidden layers (Fully Connected)	1
Hidden Nodes	36
Latent Nodes	18
Learning rate	0.001
Epochs	37
Batch_Size	32
Autoencoder—Optimizer	Adam
Hidden Layer—Activation function	ReLU
Output Layer—Activation function	Sigmoid
ELM-Optimizer	grid search
Train Test Split	70:30

The performance of the proposed system is assessed using metrics derived from the confusion matrix. Precision (3) indicates the proportion of true positives among all predicted positives, while Recall (4), also known as Sensitivity, measures the proportion of actual positives that are correctly identified. Specificity (5) refers to the proportion of true negatives correctly identified. F-measure (6) is the harmonic mean of Precision and Recall, providing a balance between the two. Accuracy (7) represents the overall correctness of the model.

$$Precision = \frac{True\ Positive}{Predicted\ Positives} \quad (3)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{4}$$

$$Specificity = \frac{True\ Negative}{True\ Negative + False\ Positive} \tag{5}$$

$$F1 - score = \frac{2 * (Precision * Recall)}{Precision + Recall} \tag{6}$$

$$Accuracy = \frac{True\ Positive + True\ Negative}{Predicted\ Positive + Predicted\ Negative} \tag{7}$$

The key findings in Figure 5 highlight the effectiveness of the Autoencoder (AE) in compressing the high-dimensional dataset from 80 features to 18, while preserving essential information. In Figure 5, the outer layers represent the number of epochs, while the inner vertical layer displays the corresponding mean squared error (MSE) values. Through the reduction of reconstruction error during training, the AE successfully retains the key features of the data while eliminating noise and unnecessary redundancy. The dimensionality reduction process ensures that the compressed feature set is optimized for subsequent anomaly classification, allowing the framework to efficiently process complex network traffic data with enhanced computational effectiveness.

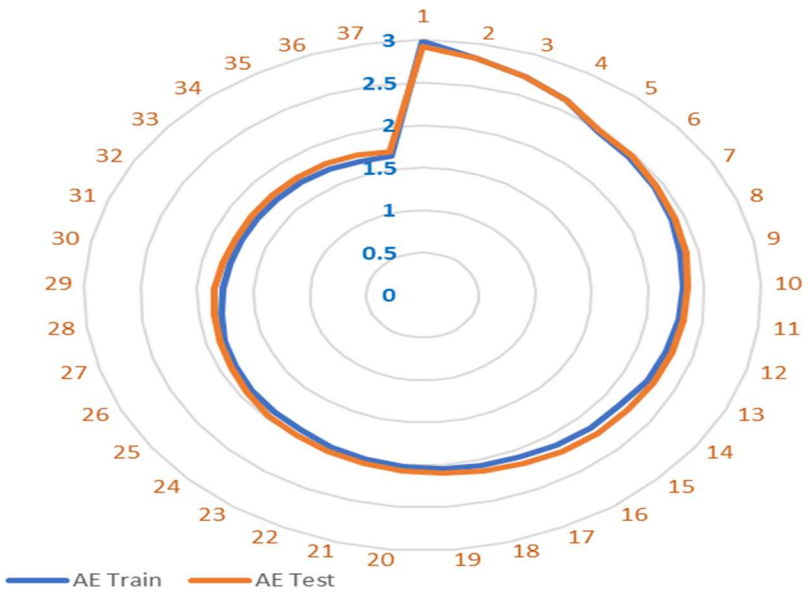

Figure 5. Loss function of the autoencoder dimensionality reduction model.

The classification phase, utilizing Extreme Learning Machine (ELM), demonstrated excellent accuracy and quick training times, confirming its effectiveness for real-time intrusion detection. The framework effectively managed the extensive CSE-CIC-IDS2018 dataset, with Figures 6 and 7 from the confusion matrix highlighting its scalability and robustness in handling a wide range of network traffic patterns and attack scenarios. In Figure 6, with 100 hidden nodes, the model correctly classified 753,830 benign instances (TN) and 765,659 malicious instances (TP). However, it misclassified 58,812 malicious instances as benign (FN) and 70,641 benign instances as malicious (FP), indicating moderate performance with some misclassifications. In contrast, Figure 7 shows that the model with 500 hidden nodes (HN 500) achieved significantly improved results, correctly classifying 814,562 benign instances (TN) and 810,040 malicious instances (TP). It misclassified only 9909 malicious instances as benign (FN) and 14,430 benign instances as malicious (FP),

demonstrating better performance in accurately predicting benign and malicious traffic with fewer misclassifications.

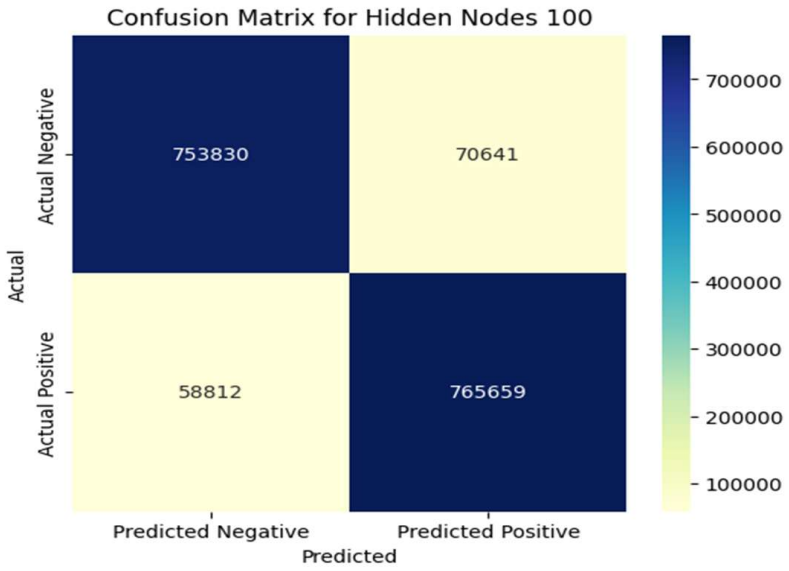

Figure 6. Confusion matrix for HN 100.

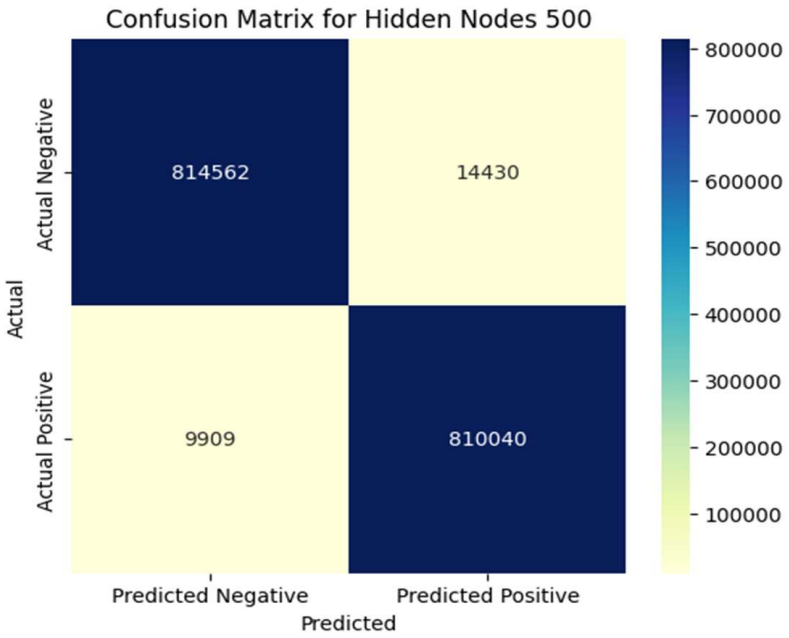

Figure 7. Confusion matrix for HN 500.

Table 3 presents the performance of the proposed system across different configurations of hidden nodes (HN) in the ELM classifier. The table clearly shows that as the number of hidden nodes increases, there is a consistent improvement in performance metrics, including *precision*, *recall*, *specificity*, *F1-Score*, and *accuracy*. Specifically, HN = 500 provides the highest values across all evaluation metrics, with an accuracy of 98.52% and an F1-Score of 98.51%. Notably, this configuration also maintains computational efficiency, as evidenced by the minimal increase in training and testing times, which is crucial for real-time applications. The HN = 500 configuration strikes an optimal balance between high performance and manageable computational cost, making it the most effective choice. We have further clarified that at this setting, precision, recall, and specificity all peaked, reinforcing the model's robustness in distinguishing between benign and malicious traf-

fic with minimal false positives and false negatives. The training and testing times, as illustrated in Figure 8, are calculated per data point using a T4 GPU to demonstrate the system's performance during the testing phase. The table indicates minimal increases in training and testing times as the number of hidden nodes grows. However, HN = 500 strikes a favorable balance between accuracy and efficiency, delivering superior classification results without a notable increase in processing time, making it the optimal choice for enhancing model performance. Figure 9 displays the ROC curve, with an AUC of 0.92, highlighting the model's strong discriminative performance. This AUC value indicates that the model effectively differentiates between benign and malicious traffic, with a low probability of misclassification. Table 4 presents a comparative analysis of the proposed AE-ELM framework with several benchmark models: SVM (Linear), CNN (Inception), DBN, and standalone AE. The results show that the proposed system outperforms all these methods in every metric. For example, the proposed system achieves a precision of 98.25%, a recall of 98.79%, a specificity of 98.26%, an F1-Score of 98.51%, and an accuracy of 98.52%. In comparison, the best-performing benchmark model, SVM (Linear), achieves an accuracy of 95.81%, which is significantly lower than the proposed system's accuracy. The improvements in precision (from 96.94% to 98.25%), recall (from 95.19% to 98.79%), and accuracy (from 95.81% to 98.52%) highlight the superior detection capabilities of our approach. Additionally, the proposed framework offers enhanced computational efficiency, which is crucial for real-time systems, as indicated by the higher accuracy and lower computational cost compared to models like CNN (Inception) and DBN. These findings position the proposed AE-ELM framework as a robust and efficient solution for detecting emerging cyber threats in complex network environments.

Table 3. Performance of the proposed system.

ELM	HN = 100	HN = 300	HN = 600	HN = 500
Precision	0.9155	0.9526	0.9609	0.9825
Recall	0.9287	0.9541	0.9517	0.9879
Specificity	0.9143	0.9383	0.9517	0.9826
F1-Score	0.9221	0.9533	0.9563	0.9851
Accuracy	0.9215	0.9568	0.9778	0.9852

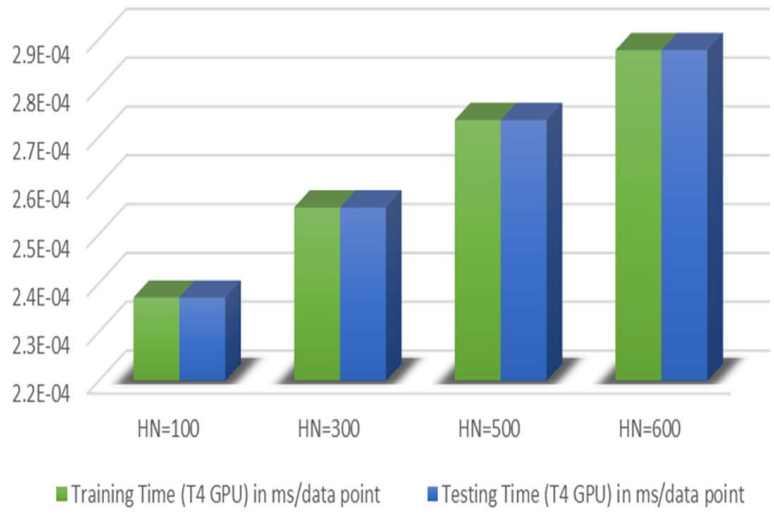

Figure 8. Processing time for training and testing per data point.

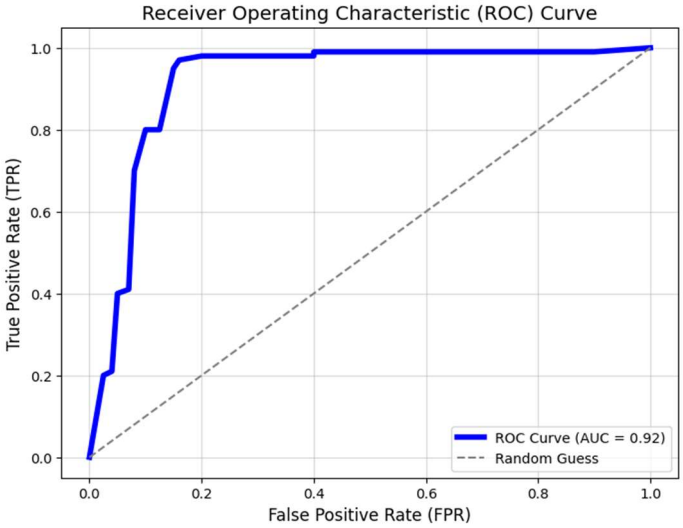

Figure 9. Receiver Operating Characteristic (ROC) curve of the proposed system.

Table 4. Performance comparison of proposed system.

Methods	Precision	Recall	Specificity	F1-Score	Accuracy
SVM (Linear) [31]	0.9694	0.9390	0.9744	0.9599	0.9581
CNN (Inception) [32]	0.9480	0.9428	0.8670	0.9421	0.9428
DBN [33]	0.9627	0.9519	0.9302	0.9366	0.9515
AE [34]	0.9493	0.9514	0.9298	0.9299	0.9299
Proposed	0.9825	0.9879	0.9826	0.9851	0.9852

5. Discussions

The proposed Hybrid AE–ELM IDS framework exhibits promising performance in efficiently detecting malicious network traffic with high accuracy and low computational cost. The integration of Autoencoders for dimensionality reduction compresses 80 original features into 18, significantly reducing input complexity while retaining essential patterns for effective classification. This transformation not only minimizes redundancy and noise but also enhances computational efficiency, which is critical for real-time environments.

The ELM, employed as the classifier, demonstrates excellent generalization and fast training performance across various hidden node configurations. The evaluation shows that an ELM configuration with 500 hidden nodes achieves the highest detection accuracy (98.52%) and F1-score (98.51%), with only marginal increases in training and testing times. This highlights the framework's suitability for time-sensitive applications where rapid threat detection is essential.

The proposed system outperforms several benchmark models, including SVM (Linear), CNN (Inception), and DBN, in terms of accuracy, precision, recall, specificity, and F1-score. Notably, it also achieves a higher Area Under the ROC Curve (AUC = 0.92), confirming its superior discriminative capability. These findings underscore the hybrid model's strength in classifying both benign and malicious traffic even under diverse and large-scale network conditions.

The framework's adaptability is a key strength. By applying dynamic thresholds and flexible classification strategies, it can adjust to variations in network behavior and evolving attack patterns. This ensures operational relevance, as real-world networks often face novel or stealthy threats. The AE–ELM architecture also supports scalability, allowing the model to handle increasing volumes of traffic without degrading detection performance. These features make the framework well-suited for deployment in live environments to identify both known and previously unseen intrusion attempts.

However, a notable limitation is that the current system does not explicitly analyze which features contribute most to classifying a sample as malicious. Addressing this in future work—by integrating explainable AI (XAI) techniques or feature attribution methods—could greatly enhance the framework's transparency and actionable insight.

Overall, the hybrid AE–ELM IDS offers a robust, scalable, and computationally efficient solution for detecting network intrusions. With further work focused on explainability and root-cause analysis, the framework holds strong potential for real-world cybersecurity applications.

6. Conclusions

The proposed Scalable Hybrid AE-ELM IDS framework efficiently addresses the critical challenges in network security, including managing high-dimensional data and adapting to emerging threats. Through the integration of Autoencoders for dimensionality reduction and Extreme Learning Machines for classification, the system effectively processes high-dimensional data while enabling real-time operations. The framework, evaluated on the CSE-CIC-IDS2018 dataset, demonstrates notable enhancements in precision, recall, and F1-score, outperforming conventional models in identifying various attack patterns. The use of dynamic thresholds and adaptive techniques enhances the system's resilience to emerging threats and diverse network traffic conditions. Although the AE-ELM framework demonstrates excellent scalability and computational efficiency, its performance is influenced by the configuration of thresholds and the nature of the dataset. Future improvements, such as evaluating the framework on a wider range of datasets and refining dynamic parameter tuning, could further strengthen its applicability in real-world security environments. The proposed framework presents a promising solution for contemporary cybersecurity, providing a reliable and scalable method for safeguarding interconnected digital ecosystems.

Author Contributions: Conceptualization, A.K., R.R., M.S., P.K., B.B. and F.B.; methodology, P.K., A.K. and R.R.; software, A.K., R.R. and M.S.; validation, A.K., R.R., M.S., P.K., B.B. and F.B.; formal analysis, A.K., M.S. and P.K.; investigation, A.K., R.R., M.S., P.K., B.B. and F.B.; resources, A.K., R.R. and P.K.; data curation, A.K., M.S. and P.K.; writing—original draft preparation, A.K., R.R. and P.K.; writing—review and editing, A.K., R.R., M.S., P.K., B.B. and F.B.; visualization, B.B. and F.B.; supervision, B.B. and F.B.; project administration, B.B. and F.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset used for this article is available online at https://www.unb.ca/cic/datasets/ids-2018.html (accessed on 17 April 2025).

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

References

1. Isong, B.; Kgote, O.; Abu-Mahfouz, A. Insights into Modern Intrusion Detection Strategies for Internet of Things Ecosystems. *Electronics* **2024**, *13*, 2370. [CrossRef]
2. El Hajla, S.; Ennaji, E.M.; Maleh, Y.; Mounir, S. Enhancing IoT network defense: Advanced intrusion detection via ensemble learning techniques. *Indones. J. Electr. Eng. Comput. Sci.* **2024**, *35*, 2010–2020. [CrossRef]
3. Attou, H.; Guezzaz, A.; Benkirane, S.; Azrour, M.; Farhaoui, Y. Cloud-Based Intrusion Detection Approach Using Machine Learning Techniques. *Big Data Min. Anal.* **2023**, *6*, 311–320. [CrossRef]
4. Paricherla, M.; Ritonga, M.; Shinde, S.R.; Chaudhari, S.M.; Linur, R.; Raghuvanshi, A. Machine learning techniques for accurate classification and detection of intrusions in computer network. *Bull. Electr. Eng. Inform.* **2023**, *12*, 2340–2347. [CrossRef]
5. Mundt, M.; Baier, H. Threat-Based Simulation of Data Exfiltration Toward Mitigating Multiple Ransomware Extortions. *Digit. Threat. Res. Pract.* **2023**, *4*, 54. [CrossRef]

6. Sarhan, M.; Layeghy, S.; Gallagher, M.; Portmann, M. From zero-shot machine learning to zero-day attack detection. *Int. J. Inf. Secur.* **2023**, *22*, 947–959. [CrossRef]
7. Zoppi, T.; Ceccarelli, A.; Bondavalli, A. Unsupervised Algorithms to Detect Zero-Day Attacks: Strategy and Application. *IEEE Access* **2021**, *9*, 90603–90615. [CrossRef]
8. Figueiredo, J.; Serrão, C.; de Almeida, A.M. Deep Learning Model Transposition for Network Intrusion Detection Systems. *Electronics* **2023**, *12*, 293. [CrossRef]
9. Kumar, S.V.; Periyasamy, M.; Radhakrishnan, R.; Karuppiah, T.; Elumalai, T. Network routing and scheduling architecture in a fully distributed cloud computing environment. *Indones. J. Electr. Eng. Comput. Sci.* **2024**, *36*, 1242–1252. [CrossRef]
10. Boukhalfa, A.; El Attaoui, A.; Rhouas, S.; El Hami, N. Unified and evolved approach based on neural network and deep learning methods for intrusion detection. *IAES Int. J. Artif. Intell.* **2024**, *13*, 4071–4079. [CrossRef]
11. Javed, A.; Ehtsham, A.; Jawad, M.; Awais, M.N.; Qureshi, A.-U.; Larijani, H. Implementation of Lightweight Machine Learning-Based Intrusion Detection System on IoT Devices of Smart Homes. *Future Internet* **2024**, *16*, 200. [CrossRef]
12. Ganesamoorthy, N.; Sakthivel, B.; Subbramania, D.; Balasubadra, K. Hen maternal care inspired optimization framework for attack detection in wireless smart grid network. *Int. J. Inform. Commun. Technol.* **2024**, *13*, 123–130. [CrossRef]
13. Zhou, Q.; Pezaros, D. Evaluation of Machine Learning Classifiers for Zero-Day Intrusion Detection—An Analysis on CIC-AWS-2018 Dataset. May 2019. Available online: http://arxiv.org/abs/1905.03685 (accessed on 10 April 2025.).
14. Dash, N.; Chakravarty, S.; Rath, A.K. Deep learning model for elevating internet of things intrusion detection. *Int. J. Electr. Comput. Eng* **2024**, *14*, 5874–5883. [CrossRef]
15. Wang, K.; Li, J.; Wu, W. A novel transfer extreme learning machine from multiple sources for intrusion detection. *Peer Peer Netw. Appl.* **2024**, *17*, 33–47. [CrossRef]
16. Moualla, S.; Khorzom, K.; Jafar, A. Improving the Performance of Machine Learning-Based Network Intrusion Detection Systems on the UNSW-NB15 Dataset. *Comput. Intell. Neurosci.* **2021**, *2021*, 5557577. [CrossRef]
17. Maseno, E.M.; Wang, Z. Hybrid wrapper feature selection method based on genetic algorithm and extreme learning machine for intrusion detection. *J. Big Data* **2024**, *11*, 24. [CrossRef]
18. Altamimi, S.; Abu Al-Haija, Q. Maximizing intrusion detection efficiency for IoT networks using extreme learning machine. *Discov. Internet Things* **2024**, *4*, 5. [CrossRef]
19. Alrefaei, A.; Ilyas, M. Using Machine Learning Multiclass Classification Technique to Detect IoT Attacks in Real Time. *Sensors* **2024**, *24*, 4516. [CrossRef]
20. Rizwanullah, M.; Mengash, H.A.; Alamgeer, M.; Tarmissi, K.; Aziz, A.S.A.; Abdelmageed, A.A.; Alsaid, M.I.; Eldesouki, M.I. Modelling of Metaheuristics with Machine Learning-Enabled Cybersecurity in Unmanned Aerial Vehicles. *Sustainability* **2022**, *14*, 16741. [CrossRef]
21. Zhang, Z.; Kong, S.; Xiao, T.; Yang, A. A Network Intrusion Detection Method Based on Bagging Ensemble. *Symmetry* **2024**, *16*, 850. [CrossRef]
22. Saleh, H.M.; Marouane, H.; Fakhfakh, A. Stochastic Gradient Descent Intrusions Detection for Wireless Sensor Network Attack Detection System Using Machine Learning. *IEEE Access* **2024**, *12*, 3825–3836. [CrossRef]
23. ElDahshan, K.A.; AlHabshy, A.A.; Hameed, B.I. Meta-Heuristic Optimization Algorithm-Based Hierarchical Intrusion Detection System. *Computers* **2022**, *11*, 170. [CrossRef]
24. Ali, H.; Elzeki, O.M.; Elmougy, S. Smart Attacks Learning Machine Advisor System for Protecting Smart Cities from Smart Threats. *Appl. Sci.* **2022**, *12*, 6473. [CrossRef]
25. Yang, W.; Zhang, H.; Lim, J.B.; Zhang, Y.; Meng, H. A new chiller fault diagnosis method under the imbalanced data environment via combining an improved generative adversarial network with an enhanced deep extreme learning machine. *Eng. Appl. Artif. Intell.* **2024**, *137*, 109218. [CrossRef]
26. Zhang, H.; Yang, W.; Yi, W.; Lim, J.B.; An, Z.; Li, C. Imbalanced data based fault diagnosis of the chiller via integrating a new resampling technique with an improved ensemble extreme learning machine. *J. Build. Eng.* **2023**, *70*, 106338. [CrossRef]
27. Xiao, Y.; Feng, Y.; Sakurai, K. An Efficient Detection Mechanism of Network Intrusions in IoT Environments Using Autoencoder and Data Partitioning. *Computers* **2024**, *13*, 269. [CrossRef]
28. Thompson, A.; Suomalainen, J. GAOR: Genetic Algorithm-Based Optimization for Machine Learning Robustness in Communication Networks. *Network* **2025**, *5*, 6. [CrossRef]
29. Aljebreen, M.; Alohali, M.A.; Saeed, M.K.; Mohsen, H.; Al Duhayyim, M.; Abdelmageed, A.A.; Drar, S.; Abdelbagi, S. Binary Chimp Optimization Algorithm with ML Based Intrusion Detection for Secure IoT-Assisted Wireless Sensor Networks. *Sensors* **2023**, *23*, 4073. [CrossRef]
30. Ortega-Fernandez, I.; Sestelo, M.; Burguillo, J.C.; Piñón-Blanco, C. Network intrusion detection system for DDoS attacks in ICS using deep autoencoders. *Wirel. Netw.* **2024**, *30*, 5059–5075. [CrossRef]

31. Almaiah, M.A.; Almomani, O.; Alsaaidah, A.; Al-Otaibi, S.; Bani-Hani, N.; Al Hwaitat, A.K.; Al-Zahrani, A.; Lutfi, A.; Awad, A.B.; Aldhyani, T.H.H. Performance Investigation of Principal Component Analysis for Intrusion Detection System Using Different Support Vector Machine Kernels. *Electronics* **2022**, *11*, 3571. [CrossRef]
32. El-Ghamry, A.; Darwish, A.; Hassanien, A.E. An optimized CNN-based intrusion detection system for reducing risks in smart farming. *Internet Things* **2023**, *22*, 100709. [CrossRef]
33. Ramalingappa, L.; Ekanthaiah, P.; Ali, I.; Manjunatha, A. Reliability analysis in distribution system by deep belief neural network. *Bull. Electr. Eng. Inform.* **2024**, *13*, 753–761. [CrossRef]
34. Khan, S.S.; Mailewa, A.B. Predicting anomalies in computer networks using autoencoder-based representation learning. *Int. J. Inform. Commun. Technol.* **2024**, *13*, 9–26. [CrossRef]