

HGLV-DNN: An Efficient Cyber-Attack Detection Using Hybrid Global-Local-Visual Deep Neural Network

Asmaa Rashid Salman^{1,*}, Noor Muneam Abbas¹, Duaa H. Hameed¹, Nidaa Flaih Hassan¹

¹ Computer Science College, University of Technology, Baghdad, Iraq

*Corresponding author: Asmaa Rashid Salman

Abstract

This study proposes HGLV-DNN, a hybrid global-local-visual deep neural framework for accurate and robust multi-class network intrusion detection. The method first converts preprocessed CIC-IDS2017 network flows into time-frequency scalogram images via the Continuous Wavelet Transform to capture transient and multiscale patterns. These images are processed in parallel by a Convolutional Neural Network (for local spatial features) and a Transformer (for long-range global dependencies). The resulting feature sets are concatenated, reduced with Principal Component Analysis, and classified using a Random Forest ensemble to enhance stability and generalization. The pipeline includes targeted undersampling and oversampling augmentation to produce a balanced dataset (75% for training and 25% for testing) and applies 10-fold cross-validation during evaluation. Experimental results on the CIC-IDS2017 dataset demonstrate near-perfect detection performance with an Area Under the ROC Curve (AUC) of 0.9999 and overall accuracy of 99.92%, along with Precision = 99.91%, Recall = 99.90%, and F1-score = 99.90%. The proposed HGLV-DNN framework outperforms several baseline and hybrid approaches, with the closest competitor (Decision Tree) achieving 99.81% accuracy. These findings indicate that integrating CWT-based visual representations with complementary local and global deep feature extractors and ensemble classification yields a highly effective and noise-resilient intrusion detection solution.

1.Introduction

The massive scale of system networking and the spread of Internet-connected device networks have radically expanded the quantity and complexity of network traffic. As a result, companies have a constantly growing list of cybersecurity risks, including denial-of-service (DoS) and botnet attacks, as well as advanced data exfiltration and zero-day exploits [1,2]. Conventional Intrusion Detection Systems, (IDS) be they signature-based or anomaly-based, frequently find it hard to keep pace with these emerging threats because they rely on fixed rules, handcrafted features and have limited generalization capabilities. With the current trend of attack strategies becoming dynamic and stealthy, the need to have intelligent and adaptive detection mechanisms that are high-performance has never been higher [3,4].

Over the last few years, machine learning (ML) and deep learning (DL) have become strong paradigms of intrusion detection, which can automatically identify latent structures in high-dimensional network data [5]. Traditional ML models like Decision Trees, Support Vector Machines, and Random Forests are highly competitive in terms of classification and usually need a significant amount of manual feature engineering, and fail to perform as well on large or skewed data. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer models have demonstrated better capabilities to learn hierarchical and contextual representations with raw data. However, the majority of current deep learning-based IDS systems process network flows as tables, therefore, not utilizing the depth of temporal and frequency domain information present in network traffic patterns [6,7].

Temporal analysis and visual transformation techniques have been used in the recent past to overcome these limitations. The representation of network flow characteristics as two-dimensional images, e.g. scalograms obtained through the Continuous Wavelet Transform

(CWT), allows one to identify localized changes, transient signatures and multi-scale relationships in traffic data. Such visual encodings provide new possibilities to exploit the capabilities of advanced computer vision architectures that are good at the recognition of complex spatial patterns and inter-feature interactions [8,9].

Nevertheless, such improvements do not eliminate problems. Visual based methods place much emphasis on local spatial characteristics and ignore global contextual dependencies which are fundamental in modeling distributed attacks and coordinated threat patterns [10]. Furthermore, deep learning models are prone to overfitting, sensitivity to class imbalances, and low interpretability on real-world network data. Accordingly, an efficient intrusion detection paradigm should incorporate complementary learning paradigms, which will provide equilibrium between global awareness, local sensitivity, and decision strength [11]. This paper proposes HGLV-DNN a Hybrid Global-Local-Visual Deep Neural Network system that can be used to detect network intrusion based on multiple classes. The suggested approach converts the preprocessed CIC-IDS2017 network flow data in the form of CWT-based scalogram images, which are further processed by a Convolutional Neural Network to capture local features and a Transformer module to capture long-range dependencies. The obtained learned features are combined, dimensionality reduced with the help of Principal Component Analysis (PCA), and are subsequently divided into classes with the help of a Random Forest (RF) ensemble to increase the stability and generalization. Also, data balancing methods, such as directed undersampling and oversampling, are also used to address the issue of class imbalance and enhance the effectiveness of learning.

The key contributions of this study include:

1. Design of a Novel Hybrid Framework (HGLV-DNN)

A novel hybrid framework, termed Hybrid Global-Local-Visual Deep Neural Network (HGLV-DNN), is proposed. This architecture integrates Convolutional Neural Networks (CNNs) and Transformers in a unified structure, enabling the simultaneous extraction of local and global features. By modeling long-range dependencies and spatial correlations, the proposed framework provides a comprehensive and hierarchical representation of network intrusion patterns.

2. Utilization of Continuous Wavelet Transform (CWT) for Time–Frequency Feature Extraction

Network traffic data are transformed into scalogram images through the Continuous Wavelet Transform (CWT), allowing for a rich time–frequency representation of network behavior. These visual representations enhance the robustness and performance of the proposed HGLV-DNN model, particularly in the presence of noisy data.

3. Employing Ensemble Learning for Accurate and Stable Decision-Making

An ensemble learning approach is adopted by incorporating the Random Forest algorithm, which aggregates multiple decision trees through a voting mechanism. This strategy improves stability, accuracy, and resilience against noise and data imbalance, while enhancing the generalization capability of the model in dynamic and complex network environments.

The remainder of this paper is organized as follows: Section 2 reviews related work in intrusion detection and hybrid deep learning models. Section 3 describes the proposed HGLV-DNN architecture and the methodology in detail. The findings and comparative assessment are outlined in Section 4, and Section 5 provides concluding remarks along with potential avenues for future investigation.

2.Related Works

In [12], the authors discuss the problem of increasing security in the Internet of Things (IoT) environment, where the number of connected devices is rapidly increasing, thereby increasing the risk of vulnerabilities. They point out the drawbacks of current deep learning-based intrusion detection techniques, especially in poor feature representation and low correlation of the extracted traffic features. To address these problems, the authors introduce MBConv-ViT, a hybrid network that is based on both MobileNet Convolution and Vision Transformer networks to combine local and global features. This combination increases the correlation of features and detection accuracy.

Authors in [13] concentrate on creating a powerful and lightweight intrusion detection system (IDS) for resource-constrained IoT devices, which are confronted with more complex and dynamic cyber threats. They cite constraints of conventional IDSs which include high computational loads and poor feature extraction. They suggest a solution to these problems, FFT-RDNet. This hybrid framework applies a dual-path feature embedding module to combine both time-domain statistical characteristics and frequency-domain energy characteristics using Fast Fourier Transform (FFT). The model is estimated on both NSL-KDD and CIC-IDS2018 datasets and demonstrates a higher level of accuracy, precision, recall and F1 score at significantly lower computational complexity.

In [14], the authors point to the susceptibility of the IoT networks to cyberattacks because of their multifaceted architecture and unsecured communication framework. They observe that the conventional intrusion detection systems have problems of the data imbalance, intricate attack patterns, and ineffective feature extraction. In order to solve these problems, the authors introduce a hybrid CNN-BiLSTM-Transformer model which is effective in capturing both spatial and temporal dependencies in the IoT traffic. They use Borderline-SMOTE to deal with the imbalance in classes and a feature selection algorithm that combines XGBoost, Chi-square and Mutual Information to remove redundancy and increase the discriminative ability. CIC-IDS 2017 and BoT-IoT experiments reveal a high accuracy of 99.80% and 97.95%, respectively, which is higher than the conventional IDS work.

The authors in [15] discuss the emerging menace of zero-day attacks in more and more interconnected online spaces and the shortcomings of conventional, signature-based Intrusion Detection Systems. They present a self-supervised contrastive learning model based on transformer encoders to improve generalization in identifying previously unseen attacks on raw packet sequences. The method combines the use of packet-level data augmentation with transformer-based feature recognition to learn meaningful traffic representations automatically, thereby removing the need to use handcrafted NetFlow features. There are significant improvements in performance, and up to 3 percent improvement in AUC intra dataset and 20 percent improvement in AUC inter-dataset.

In [16], authors concentrate on the improvement of network intrusion detection to provide stable and secure network environments. Their proposed solution starts with the normalization and encoding of intrusion data with the help of solo thermal coding and deep feature extraction with the help of the ResNet18 architecture. They use the CEKL loss function to tackle the problem of data imbalance in training and use a Softmax layer to perform final classification. The detection model that results thereof constitutes the basis of a full intrusion detection system. Experimental analysis of BoT-IoT and ToN-IoT databases reveals high accuracy 99.02 and 99.06, respectively, and high recognition rates and low false alarm rates, proving the efficiency of the model and its practical applicability.

Authors, in [17], show an intrusion detection system (IDS) that is designed to be used in smart health applications and combines Particle Swarm Optimization (PSO) and AdaBoost algorithms to boost the detection of malware. PSO is used with NSL-KDD dataset (125,973 instances, 41 features) to select the features and 12 significant attributes are identified, which are related to intrusion detection. The system effectively classifies several types of attacks, like DoS, U2R, R2L, or Probe attacks. AdaBoost is the best classifier in terms of recall, which indicates good detection ability. The experimental findings reveal that the PSO-AdaBoost yields remarkable results and can be used to secure medical IoT systems and enhance healthcare outcomes by means of powerful and smart intrusion detection.

The authors in [18] discuss the increasing cybersecurity threats posed by the rapid implementation of smart city applications that use IoT, where devices and cloud communications are interconnected, forming multiple attack surfaces. They present an anomaly detection framework in order to improve intrusion detection, which unites multiple machine learning models that include SVM, ANN, KNN, LR, DT, and RF. The research also uses ensemble techniques like bagging and boosting to enhance the detection performance, compared with conventional one-classifier frameworks. The presented approach incorporates feature selection and cross-validation, which means that it is more robust. Experimental assessments of UNSW-NB15 and CICIDS2017 yield better outcomes in terms of accuracy, precision, recall, and F1-score, especially in the detection of infrequent attack types. In [19], the authors present a two-step Intrusion Detection System (IDS) to improve the security of the IoT network and enhance detection. The first stage includes splitting network data into four categories nominal, integer, binary, and float, and categorizing each of them with the assistance of various variations of Naive Bayes. The last classification is by majority voting. In the second stage, the data labeled as normal in phase one is once more examined using an unsupervised Elliptic Envelope strategy in order to identify hidden anomalies. Assessed on NSL-KDD, UNSW-NB15, and CIC-IDS2017 data, the presented IDS reached the accuracy of 97, 86.9, and 98.59, respectively, outperforming several existing approaches.

The authors in [20] propose a Hybrid Learning Model (HLM) to enhance intrusion detection performance as digital transformation challenges intensify. Conventional IDSs struggle to cope with the complicated patterns classification, which is why the integration of parametric and non-parametric classifiers is encouraged. The HLM operates in two steps: non-parametric base learner (np-BL) step with KNN, DT, RF, GBM and SVC-RBF models and a parametric meta-learning (PML) step with LR, NBC, LDA, QDA and Linear SVM. A stacking ensemble is used to combine the two phases to improve flexibility and robustness. Experimental evaluations demonstrated higher accuracy and lower false alarm rates, respectively.

In [21], the authors address the growing complexity of the contemporary cybersecurity threats and suggest a comprehensive ensemble learning approach to intrusion detection. Since it is understood that conventional AI-based systems cannot exhibit uniform performances under different settings, the study incorporates feature selection, model optimization as well as evaluation as a single unit. Fourteen machine learning models (including individual classifiers and ensemble techniques, such as bagging, boosting, stacking and blending) were evaluated on RoEduNet-SIMARGL2021 and CICIDS-2017 datasets. Findings indicate that ensemble techniques are always better than single models and Random Forest does better on RoEduNet-SIMARGL2021 and blending/bagging attains almost perfect F1 (>0.996) on CICIDS-2017. The feature selection based on information gain decreased the training time by 94 percent without any accuracy loss.

3. Proposed Method

In this study, a novel and efficient framework, termed Hybrid Global-Local-Visual Deep Neural Network (HGLV-DNN), is proposed for network intrusion detection. The overall structure of the proposed approach is designed to optimally extract the spatial and temporal information embedded within network data and to enhance detection performance by integrating local and global features. In the first stage, raw data are retrieved from the designated dataset and subjected to a preprocessing procedure, which includes handling missing values, data normalization, and preparing the data for subsequent stages. To address the issue of data imbalance among different classes, a data augmentation technique is applied. By generating synthetic samples for minority classes, this process balances the data distribution, mitigates overfitting, and ultimately improves the accuracy of the intrusion detection model. In the next stage, the preprocessed data are transformed into an image-based representation. For this purpose, the scalogram transformation derived from the Continuous Wavelet Transform (CWT) is utilized to visually represent the time–frequency characteristics of the network traffic signals. The output of this stage is a collection of scalogram images, each reflecting the dynamic behavioral patterns of network traffic. Subsequently, the generated images are simultaneously and independently fed into two deep learning networks: a Convolutional Neural Network (CNN) for extracting local features, and a Transformer network for capturing global dependencies. The CNN efficiently identifies spatial patterns within the images, while the Transformer, through its attention mechanism, models long-range dependencies and captures the global structure of the data. The features extracted from both networks are then concatenated to form a unified and richer feature representation. To further reduce dimensionality and eliminate redundant information, the Principal Component Analysis (PCA) algorithm is applied. Finally, the resulting feature vector is passed to a Random Forest classifier, which, through its ensemble of decision trees, performs accurate classification of network traffic and detects potential intrusions. In summary, the proposed HGLV-DNN approach integrates dual deep learning mechanisms (CNN and Transformer), optimized dimensionality reduction (PCA), and ensemble-based classification (Random Forest) to establish a robust and effective framework for network intrusion detection. This method not only enhances detection accuracy but also demonstrates reduced sensitivity to imbalanced data. Figure 1 illustrates the overall architecture of the proposed HGLV-DNN framework.

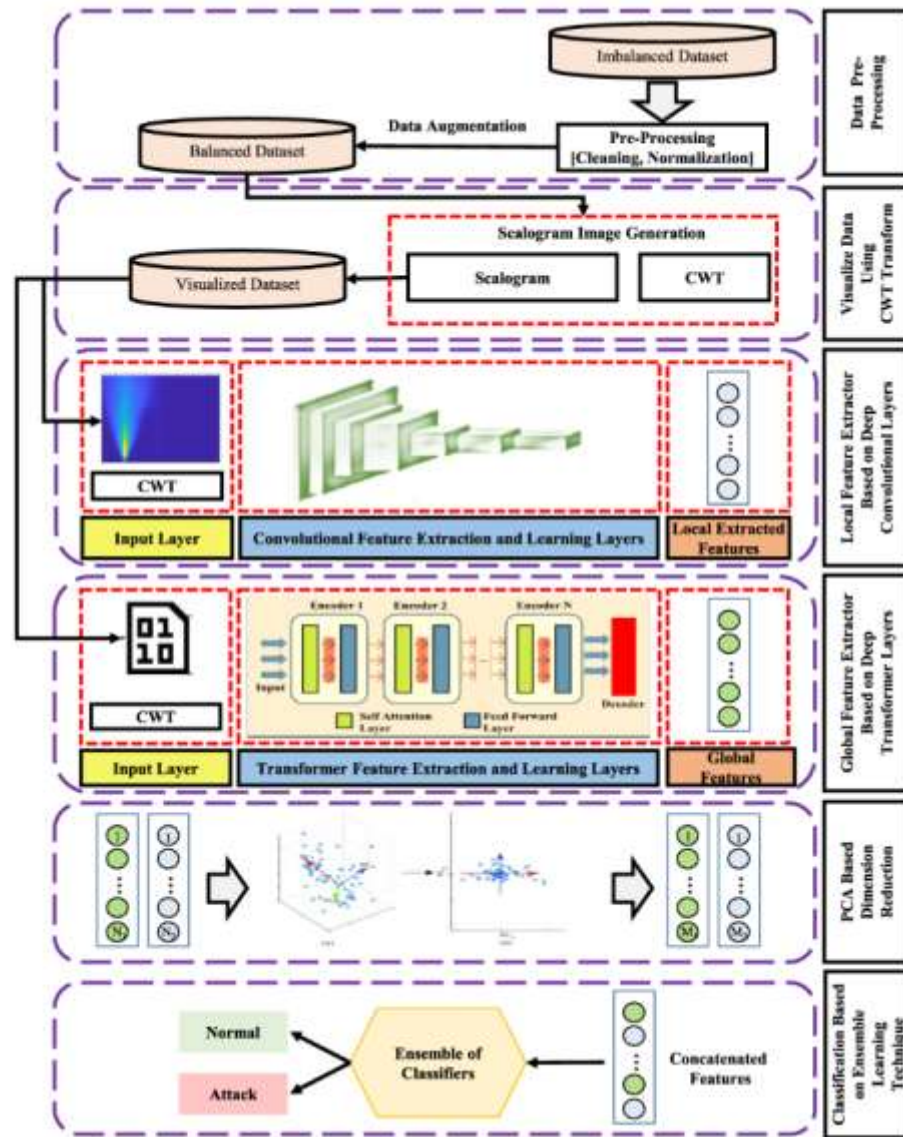


Figure 1. Diagram of the proposed method

3.1 Dataset and Data Preprocessing

The publicly available CIC-IDS2017 dataset that is offered by the Canadian Institute for Cybersecurity (CIC) is employed in this research to assess the efficiency of the suggested solution. This dataset contains traffic logs associated with the different types of cyberattacks like Port Scan, Brute Force, DoS, DDoS, Web Attack, Botnet, Infiltration, and Heartbleed. As there are few samples for attacks (like Botnet, Infiltration, and Heartbleed), such classes are combined into one class called Attack, and the model can be trained on a reasonable amount of data [22]. Table 1 presents the distribution of the data used in this research. As depicted, 118,015 samples are utilized for multi-class intrusion detection with seven classes used in the classes namely one Benign and six attack categories. The data set is divided such that 75

percent of the data are used for training section and 25 percent for testing to determine the effectiveness of the suggested system.

Table 1. Distribution of the CIC-IDS2017 dataset used in simulations

Category	No. of records	Training (75%)	Testing (25%)
Attack	2003	1503	500
Web attack	2180	1635	545
Brute force	13832	10374	3458
Port scan	25000	18750	6250
DDoS	25000	18750	6250
DoS	25000	18750	6250
Benign	25000	18750	6250
Total Recors	118015	88512	29503

3.1.1 Data Augmentation

As shown in Table 1, there is a significant class imbalance between the samples. To overcome this problem, a data balancing procedure has been included in the proposed methodology, which has two consecutive steps, i.e., undersampling and oversampling. The first step involved the undersampling of the majority classes (those with over 5,000 samples) to 5,000 samples to avoid overrepresentation. Then, the minority classes were oversampled to create new synthetic samples until each of the classes had 5,000 instances, so that the balance and the uniformity of the class distribution would be achieved. Two data augmentation methods were used concurrently during the oversampling phase to provide diversity and increase the training data. Scaling transformations that used different coefficients were applied to the data in the first technique, so that the model could fit the possible scale variations in actual network traffic. In the second method, the data was distorted by means of Gaussian random noise that has a zero mean and a small standard deviation. This is useful in assisting the model to become more robust and generalize in the presence of environmental noise and natural variations. Together, these strategies of augmentation have greatly increased the number and variety of the training data, which resulted in better model performance, less overfitting, better detection rates, and better generalization of the classifier in real-world situations. Table 2 shows the final distribution of the dataset after the balancing operations and Figure 2 visually compares the data distribution before and after augmentation.

Table 2. Distribution of the final dataset following augmentation

Category	No. of records	Training (75%)	Testing (25%)
Attack	5000	3750	1250
Web attack	5000	3750	1250
Brute force	5000	3750	1250
Port scan	5000	3750	1250
DDoS	5000	3750	1250
DoS	5000	3750	1250
Benign	5000	3750	1250
Total Records	35000	26250	8750

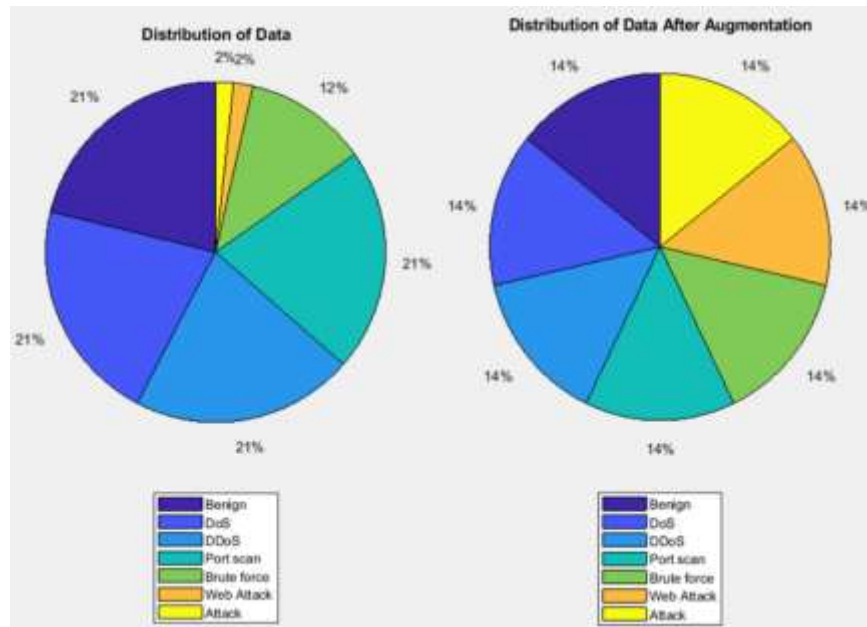


Figure 2. Comparison of dataset distribution prior to and following augmentation

2.3 Feature Visualization Using Scalogram Transformation

A scalogram presents a graphic display of energy distribution in the time-scale domain. The Continuous Wavelet Transform (CWT) can be an effective instrument in the framework of intrusion detection because it can transform numerical and statistical network data into visual representations. These visualizations reveal hidden and complicated patterns in network traffic, thus making it easier to understand the differences between normal and malicious activities. Scalograms are presented as color maps over time and scale axes, which highlight local and transient characteristics, which are key characteristics when it comes to determining patterns of cyberattacks. The raw data in the CIC-IDS2017 dataset consists of statistical and behavioral features linked to network packets. The conversion of these features into scalogram images makes it possible to use deep neural networks, a type of neural network that is effective at learning more intricate visual patterns. The initially numerical data have been turned into forms that do not only match computer vision models, but also allow discriminating between different types of attacks and normal behavior. In the Continuous Wavelet Transform, each feature vector is considered to be a time-varying function, and convolved with shifted and rescaled versions of a mother wavelet. Scalogram of a sound is mathematically defined as follows:

$$W(a, b) = \int_{-\infty}^{+\infty} x(t) \cdot \frac{1}{\sqrt{|a|}} \psi^* \left(\frac{t-b}{a} \right) dt \quad (1)$$

with $x(t)$ being the feature-vector, (a) being the scale parameter, and (b) being the time-shift parameter. The transform yields a matrix of the wavelet coefficients which characterize the energy of the signal at various scales and time intervals. The squared magnitude of the wavelet coefficients defines the scalogram:

$$S(a, b) = |W(a, b)|^2 \quad (2)$$

The scalogram transformation used improves the ability of the system to identify nonlinear, transient and multiscale patterns in the data-patterns that might otherwise remain hidden in traditional numerical representations. Figure 3 shows sample visualized characteristics of

one normal group and one attack group obtained by the use of the scalogram transformation. The first image (Figure 3a) relates to the normal class whereas the second image (Figure 3b) is associated with the attack class.

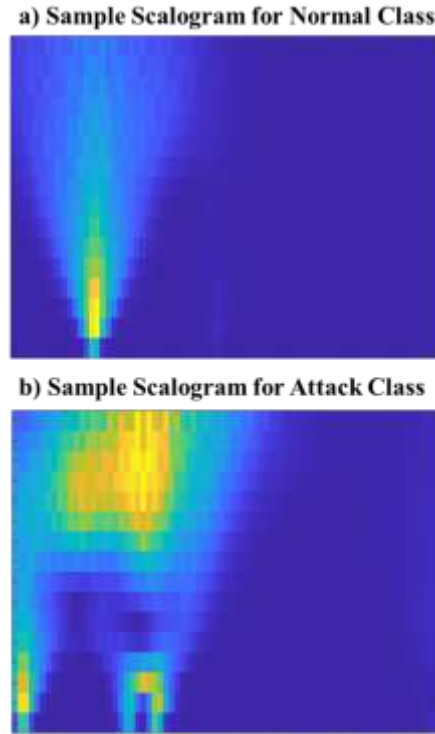


Figure 3. Representative feature visualizations for each class obtained through scalogram transformation

3.3 Feature Extraction

Transformer and Convolutional Neural Network (CNN) are used in the proposed approach as deep neural networks that extract features by being concentrated on global and local features respectively. The combination of global and local representations obtained with the help of these networks gives a more varied and discriminative feature space and enhances the capability of the model to describe complex network behaviors. The following subsections explain the process of feature extraction of each network in detail.

3.3.1 Global Feature Extraction Using the Transformer Neural Network

The scalogram images in the proposed method are processed to extract global features using a Transformer neural network. The Transformer, with its attention mechanism, is an effective model for capturing long-range dependencies and non-local relations between various parts of the image. The Transformer architecture used has n encoder and m decoder layers stacked together. Both the encoders and decoders have two basic sub-layers: a multi-head self-attention mechanism, and a feed-forward neural network (FFN), with each decoder having an extra cross-attention layer to allow communication with the encoder outputs.

First, the input scalogram image is decomposed into a collection of feature vectors $X = \{x_1, x_2, \dots, x_T\}$ where $x_i \in R^d$ is the feature vector for the corresponding patch of the image. Positional encoding (PE) is appended to each feature vector to maintain the spatial

arrangement and position context of the patches so that the model can capture the sequence of their arrangement. The last network contribution is expressed as:

$$Z_0 = X + PE \quad (3)$$

Where (PE) represents the positional encoding matrix, whose values are calculated in the following way:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{c^{2i/d}}\right), \quad PE_{(pos,2i+1)} = \cos\left(\frac{pos}{c^{2i/d}}\right) \quad (4)$$

In this case, pos denotes the patch position, i is the feature dimension index, d is the embedding dimension and c is an experimentally determined constant that determines the frequency range of the sinusoidal variations of the input sequence in relation to the length of the input sequence. The data are fed through the encoder layers having undergone positional embedding. Each encoder has three matrices Query (Q), Key (K), and Value (V) calculated by applying trainable weight matrices (W_Q, W_K, W_V) to the output of the previous layer (Z_{l-1}):

$$Q = Z_{l-1}W_Q, \quad K = Z_{l-1}W_K, \quad V = Z_{l-1}W_V \quad (5)$$

The weight to each head is calculated by the similarity between (Q) and (K), and then applied to corresponding (V) values:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (6)$$

In order to improve the representational ability of the model, multiple attention heads are computed in parallel and concatenated:

$$MultiHead(Q, K, V) = Concat(head_1, head_2, \dots, head_h)W_O \quad (7)$$

In which h represents the number of attention heads and (W_O) is the projection matrix of the output. The residual connection between the output of this layer and the input is added, and it is layer normalized to stabilize training:

$$\tilde{Z}_l = LayerNorm(Z_{l-1} + MultiHead(Q, K, V)) \quad (8)$$

The extracted features are passed through a feed-forward network to obtain nonlinear and compositional representations:

$$Z_l = LayerNorm(\tilde{Z}_l + FFN(\tilde{Z}_l)) \quad (9)$$

Where:

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (10)$$

In the decoder section, in addition to self-attention, a cross-attention mechanism is applied between the decoder inputs and the encoder outputs to learn the interactions between low-level and high-level representations. This is mathematically expressed as:

$$Attention_{cross}(Q_d, K_e, V_e) = softmax\left(\frac{Q_d K_e^T}{\sqrt{d_k}}\right)V_e \quad (11)$$

and where d and e represent the decoder and encoder respectively. The Transformer is used in the final step to generate a global feature vector F_g after going through n encoder and m decoder layers, which captures the abstract, long-range, and non-local information from the input scalogram image:

$$F_g = f_{Transformer}(X) \quad (12)$$

The vector (F_g) is the high level global features of the image extracted and later fused with the local features acquired by the CNN to reduce the dimensionality and provide classification. Figure 4 illustrates the architecture of the Transformer neural network employed in this study.

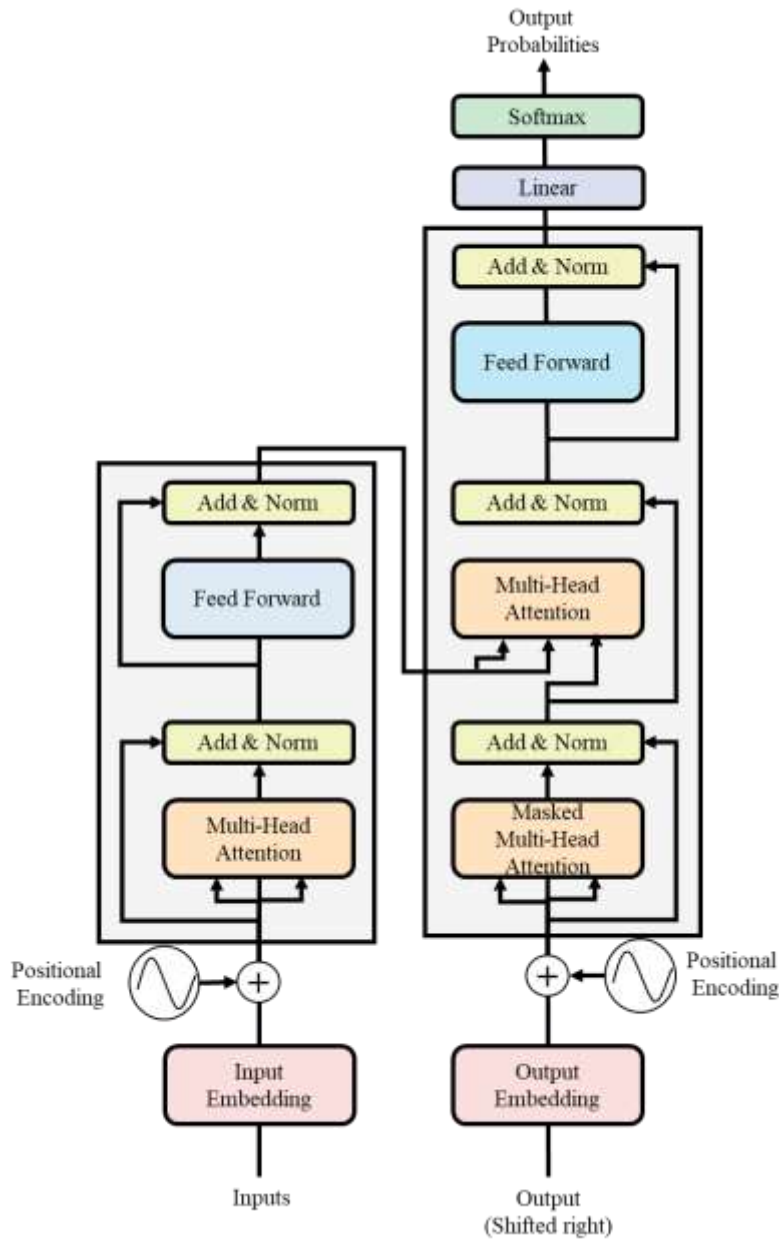


Figure 4. Structure of the Transformer Neural Network

3.3.2 Local Feature Extraction Using the CNN

The given section is dedicated to the local feature extraction of the visual patterns produced by the scalogram transformation with the help of Convolutional Neural Network (CNN). The visualized data are fed into a deep CNN during this stage to capture local-spatial features embedded in the scalogram images. CNNs use convolutional layers to extract image-based features that are complex and hierarchical. Besides, they are capable of recognizing multi-scale patterns, i.e. they can recognize features at various levels of spatial granularity in an image. The CNN architecture used in this work is shown in Figure 5. The CNN proposed to

extract features from scalogram images in the proposed framework is a combination of a convolutional layer, a ReLU layer, a pooling layer, and a fully connected layer. The convolutional layers are the first step in the network and they filter the input image with a series of filters to identify the basic local features like edges, corners and color changes. After this, activation functions of Rectified Linear Unit (ReLU) stages are applied. These layers introduce nonlinearity by transforming negative values to zero which avoids problems like vanishing gradients and enhances convergence in training. This is followed by downsampling of the feature maps in pooling layers that reduce the dimensionality of the data, the number of parameters and computational cost. The pooling makes the model more robust as it retains the most important spatial characteristics but becomes less vulnerable to the small variations and noise. Fully connected layers, in which each neuron has a connection to all neurons in the next layer are used in the final stages to integrate the extracted features and allow final decision-level representations. This cascade of layers works together to isolate informative and discriminative features of the scalogram images which offer the downstream parts of the model with meaningful inputs to be analyzed and make decisions. In general, the method allows the CNN to effectively locate and isolate salient local features within the scalogram representations, which leads to better accuracy and computational efficiency of the proposed framework.

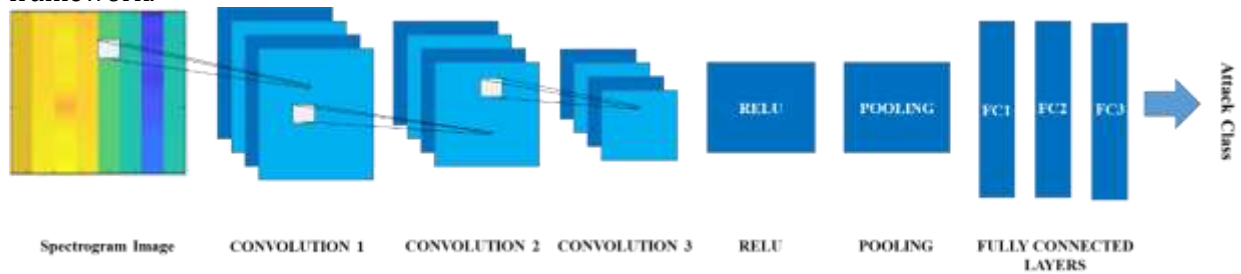


Figure 5. Architecture of the Convolutional Neural Network (CNN) used in the proposed model

3.4 Dimensionality Reduction Using the PCA Algorithm

As a linear dimensionality reduction approach, Principal Component Analysis (PCA) decreases the number of features in a dataset but retains the majority of its informational variance. PCA does this by converting the original data into a new coordinate system, whose axes (principal components) correspond to the directions of maximum variance. Here, the data are centered by first subtracting the mean of each of the features as shown in (13):

$$X_{\text{centered}} = X - \mu \quad (13)$$

where X is the data matrix and μ is the mean of features.

Then, the covariance of the centered data is calculated based on (14):

$$C = \frac{1}{n-1} \sum_{i=1}^n (X_i - \mu)(X_i - \mu)^T \quad (14)$$

where n is the quantity of data samples.

The covariance matrix is then decomposed to give the eigenvectors and eigenvalues of the covariance matrix:

$$C_{vi} = \lambda_i v_i \quad (15)$$

The eigenvectors are ranked by descending eigenvalue in order and the k largest eigenvectors (largest eigenvalues) are picked to create a new projection matrix (W).

Lastly, the original data are mapped onto the new k-dimensional feature space through Equation (16):

$$X_{\text{reduced}} = X_{\text{centered}}W \quad (16)$$

PCA speeds up the training of machine learning models by minimizing the features, and it also simplifies the structure of the models. Also, PCA transforms data into a lower-dimensional form, which makes the computations less complex and eliminates redundant or noisy dimensions. This results in the better model performance, noise reduction and generalization in the downstream classification tasks.

3.5 Data Classification Using the Random Forest Algorithm

The algorithm used in this study is the Random Forest (RF) algorithm which is used for data classification and intrusion detection. Random Forest is a powerful ensemble learning algorithm that employs the concept of Bootstrap Aggregating (Bagging), which helps to greatly improve the classification performance by combining the results of multiple decision trees. In this algorithm, multiple, different subsets of the training data are created through random sampling with replacement, and each subset is utilized to train a separate decision tree [23]. Moreover, the number of features that are taken into account at any split node is randomly chosen during the construction of each tree. This two-level randomization, both in data samples and features selection, decreases the correlation of the single trees, and as a result, minimizes the variance of the overall model. The final classification output of the Random Forest is obtained using a majority voting mechanism across all the trees which leads to a better accuracy, stability and generalization than any single decision tree. Figure 6 shows the overall design of the Random Forest algorithm, highlighting the ensemble-based decision-making procedure.

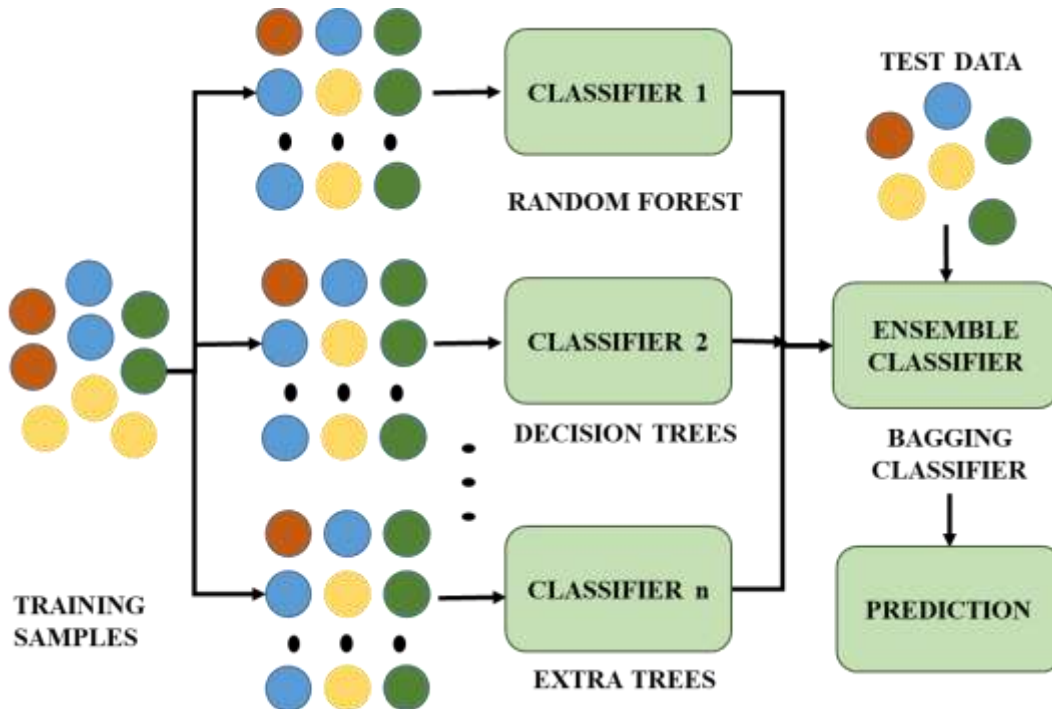


Figure 6. General structure of the Random Forest algorithm

4. Results and Discussion

This section explains the experimental findings of the suggested approach to network intrusion detection, as well as a comparative analysis with the previous research. The performance evaluation of the suggested method was conducted in MATLAB R2024b on a computing platform with an Intel Core i5 processor, 16 GB of RAM, and NVIDIA Quadro K2000 graphics card. As the main benchmark to be experimented with, the CICIDS2017 dataset was used. After data augmentation, the dataset was divided into a 75 percent training and 25 percent testing. A cross-validation strategy of (K = 10) was used to avoid overfitting. Besides, four popular metrics have been used to quantitatively measure the effectiveness of the proposed approach and compare it with the existing methods: Accuracy, Precision, Recall, and the F-Measure. Accuracy (also known as Detection Accuracy) is the proportion of samples correctly classified to the total number of samples. Precision is defined as the ratio of correctly recognized attack samples out of all of the instances that are predicted to be attack. Recall or Detection Rate is a measure of the rate of the real attacks that have been detected. Lastly F-Measure is the harmonic mean of Precision and Recall which gives a balanced analysis of the performance of the model using the two measures at the same time. The mathematical formulations associated with these metrics are as follows:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (17)$$

$$\text{Recall} = \text{Detection Rate} = \frac{TP}{TP+FN} \quad (18)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (19)$$

$$\text{F Measure} = 2 \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right) \quad (20)$$

where:

- (TP) is the count of attack cases that have been correctly detected as attacks.
- (TN) refers to the number of the normal traffic samples that were rightly identified as normal.
- (FP) is the count of normal traffic samples that are incorrectly identified as attacks.
- (FN) represents the samples of attack that are misclassified as normal.

Figure 7 presents the Receiver Operating Characteristic (ROC) curve of the presented approach. The ROC plot is used to measure the performance of the model in terms of classification by comparing the True Positive Rate (TPR) and False Positive Rate (FPR) of the model at different decision levels. The optimal mark of a ROC curve, which denotes the work of an ideal classifier, is the upper-left corner of the graph (point (0, 1)). This point represents a TPR of 1 (100 percent) and an FPR of 0 (0 percent), which means that all positive cases are accurately diagnosed and no negative cases are wrongly diagnosed as positive. As illustrated in Figure 7, the ROC curves of all the seven target classes are almost parallel to the vertical (TPR) axis and then run parallel to the horizontal (FPR) axis at a height close to 1.0. This good performance shows almost perfect classification of the suggested approach on all classes since the curves are placed very near the perfect top-left point of the graph.

The overall model achieves an Area Under the Curve (AUC) of 0.9999, demonstrating the proposed method's exceptional discriminative capability in distinguishing and accurately classifying different classes. Consequently, according to this analysis, it is possible to conclude that the proposed HGLV-DNN framework demonstrates excellent sensitivity and accuracy and has an almost optimal performance in detecting and classifying different types of intrusions.

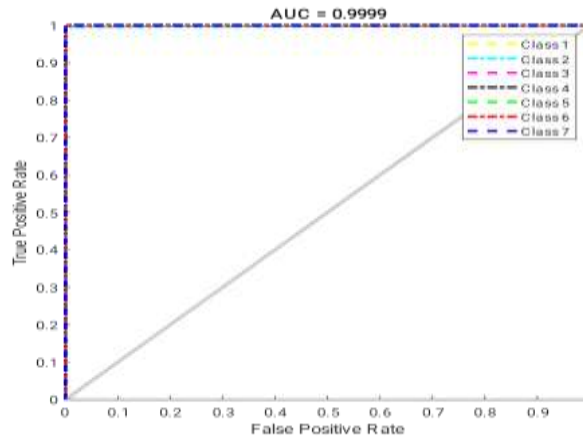


Figure 7. Receiver Operating Characteristic (ROC) Curve

Figure 8 shows the confusion matrix of the suggested approach in detail. This matrix shows the efficacy of this proposed framework in terms of classification with seven target classes, which are Benign, DoS, DDoS, Port Scan, Brute Force, Web Attack, and Attack. The main diagonal contains the counts of samples that were classified correctly (True Positives) whereas the off-diagonal values signify the misclassified samples. A review of the matrix demonstrates that the suggested approach has excellent performance since the total count of correctly classified samples (8,743) far exceeds the total number of misclassifications (7 samples only), which leads to an overall accuracy of almost 100%. Specifically, DoS (1,325 samples) and Brute Force (1,289 samples) classes are the ones where the maximum number of correct predictions is found. Moreover, the classifier performed error-free performance in the Brute Force, Web Attack, and Attack classes - i.e. all test samples that belong to this category were correctly classified. On the other hand, a total of seven misclassifications were registered throughout the whole dataset including two Benign samples and four Port Scan samples that were misclassified as DoS and one DDoS sample that was misclassified as Benign. This low error rate, mainly with the DoS class, proves the high level of discriminative ability and efficacy of the presented method in distinguishing between various types of network attacks.

		Confusion Matrix						
True Class	Benign	1250	2					
	DoS		1325					
	DDoS	1		1254				
	Port scan		4		1195			
	Brute force					1289		
	Web Attack						1232	
	Attack							1198
		Benign	DoS	DDoS	Port scan	Brute force	Web Attack	Attack
		Predicted Class						

Figure 8. Confusion Matrix of the Proposed Method

Figure 9 shows the performance of the proposed method across the seven intrusion detection classes in terms of Precision, Recall, and F1-Score. These findings suggest that the three metrics have extremely high scores, with the majority of the scores being close to or even equal to 100%, which proves the excellent accuracy and generalization capability of the hybrid framework suggested. In the case of the Brute Force, Attack and Web Attack classes, the three metrics (Precision, Recall and F1-Score) are all equal to 100 per cent indicating that the three types of attacks are detected without error. In the same way, Recall values of the DoS, DDoS, Brute Force, Web Attack, and Attack classes are all set at 100 and this means that almost all the real attack samples were correctly detected by the classifier (i.e. a very high detection rate). Even though the Benign, DoS and Port Scan classes have slightly lower Precision scores, the values remain above 99.5 with the difference attributable to a very small number of misclassifications. Overall, since F1-Score, being a comprehensive assessment measure, is nearly within the range of 100 percent in most classes, these results conclusively prove the high discriminative power, robustness and stability of the proposed approach in multi-class network intrusion detection.

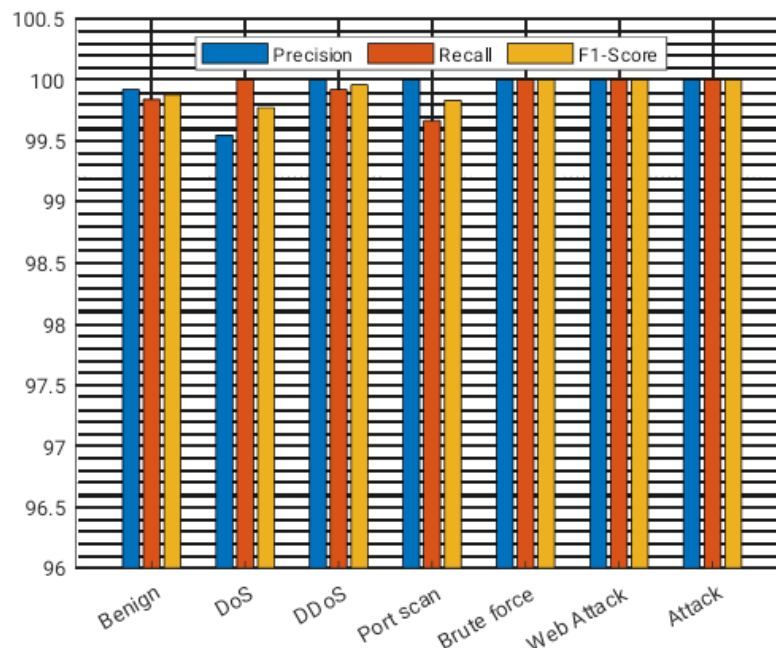
**Figure 9. Class-wise Evaluation Metrics for Test Data**

Table 3 shows a comparative analysis of the proposed HGLV-DNN (Hybrid Global-Local-Visual Deep Neural Network) with five popular classification algorithms in the field of network intrusion detection. The benchmark algorithms are Adaboost, K-Nearest Neighbors (KNN), Naïve Bayes Elliptic-Envelope (NB elliptic-envelope), Hybrid Learning Model (HLM), and Decision Tree whose reported values were obtained from references [17]-[21]. Based on the findings that are summarized in the table, the proposed HGLV-DNN was the most accurate with an overall accuracy of 99.92%, compared to all the competing methods. This accuracy is not only higher than the traditional models like Adaboost (98.5%), KNN (98.8%), and NB

elliptic-envelope (98.59%), but also higher than more sophisticated architecture models like HLM (99.63%) and Decision Tree (99.81%). The performance gap of 0.11% between the suggested model and the best competing model (Decision Tree, having an accuracy of 99.81) proves the higher accuracy and discriminative potential of HGLV-DNN to classify the attacks correctly. These findings clearly indicate that the proposed model is the most effective in terms of detection accuracy, owing to its innovative architecture and optimized feature integration.

Table 3. Comparison of the Proposed HGLV-DNN Model with Other Methods Based on Accuracy

Author	Method	Accuracy
Khan et al. [17]	Adaboost	98.5
Bukhari O et al. [18]	KNN	98.8
Vishwakarma M et al. [19]	NB elliptic-envelope	98.59
Rajathi, C. et al. [20]	Hybrid Learning Model (HLM)	99.63
Bibers et al. [21]	Decision Tree	99.81
-	HGLV-DNN	99.92

Table 4 compares the proposed HGLV-DNN model with the same five baseline classifiers Adaboost [17], KNN [18], NB elliptic-envelope [19], Hybrid Learning Model (HLM) [20], and Decision Tree [21] based on three important performance measures: Precision, Recall, and F-Score. It is analyzed that the proposed HGLV-DNN has the highest scores in all three measures, and the Precision = 99.91, Recall = 99.90, and F-Score = 99.90. These results are better than the best competing model, the Decision Tree, with an F-Score of 99.81 and indicate a significant improvement over classical models, including Adaboost (97.50%), KNN (98.50%), and NB elliptic-envelope (98.30%), with a performance gain of more than 1.4%. The HGLV-DNN model outperforms others in both Precision, representing its capability to limit false alarms, and Recall, reflecting its success in identifying real attacks. Since the F-Score is the harmonic average of Precision and Recall, therefore, a good balanced measure of accuracy and completeness, a F-Score of 99.90% results in the validation of the extraordinary balance, strength, and discriminatory ability of the suggested hybrid framework. These quantitative findings strongly prove the efficiency, reliability, and high generalization of HGLV-DNN in network intrusion detection context, achieving near-perfect detection with minimal false positives and false negatives.

Table 4. Comparison of the Proposed HGLV-DNN Model with Other Methods Based on Precision, Recall, and F-Score

Method	Precision	Recall	F-score
Adaboost [17]	98.30	96.66	97.50
KNN [18]	98.60	98.50	98.50
NB elliptic-envelope [19]	98.31	98.30	98.30
Hybrid Learning Model (HLM) [20]	99.26	98.99	98.72
Decision Tree [21]	99.81	99.81	99.81
Proposed (HGLV-DNN)	99.91	99.90	99.90

5. Conclusion

This paper introduced HGLV-DNN, a global-local-visual deep neural network architecture that was created to improve the quality and strength of network intrusion detection systems (NIDS). The proposed framework is effective in capturing the time-frequency scalogram images of the network behavior by converting CIC-IDS2017 dataset into a time-frequency images, which offers the opportunity of capturing both time-dependent and frequency-dependent properties of network behavior. The combination of Convolutional Neural Networks (CNNs) to extract local features and Transformers to model global dependencies allows for global representation learning at various spatial and contextual scales. Then Principal Component Analysis (PCA) and Random Forest (RF) classification is used to narrow down the learnt features to reduce overfitting and enhance the stability of the decisions. As experimental analysis proves, the HGLV-DNN architecture is state of the art, reaching overall accuracy of 99.92% and AUC of 0.9999, much higher than the traditional machine learning and deep learning baselines. The findings affirm that the framework has a high generalization capacity, high detection rate, and low rate of false alarm, which are essential for its practical application in the real-world cybersecurity setting. The results of this study suggest the possibility of using visual time frequency analysis, hybrid deep feature extraction and ensemble classification to build smart, robust and dynamic intrusion detection systems. Future research can aim at creating lightweight or real-time versions of the suggested architecture, add adaptive learning to monitor the appearance or further development of attack patterns, and investigate explainable artificial intelligence (XAI) techniques to increase transparency and confidence in automated network defense systems.

References

- [1] Abdulganiyu, O. H., Ait Tchakoucht, T., & Saheed, Y. K. (2023). A systematic literature review for network intrusion detection system (IDS). *International journal of information security*, 22(5), 1125-1162.
- [2] Al-Safaar, D. W., & Al-Yaseen, W. L. (2023). A survey of network intrusion detection systems based on deep learning approaches. *Научно-технический вестник информационных технологий, механики и оптики*, 23(2), 352-363.
- [3] Kimanzi, R., Kimanga, P., Cherori, D., & Gikunda, P. K. (2024). Deep Learning Algorithms Used in Intrusion Detection Systems--A Review. *arXiv preprint arXiv:2402.17020*.
- [4] AbdelHalim, A. P., Abo-Alian, A., & Badr, N. (2025). Deep Learning Techniques for Network Intrusion Detection: A Comparative Survey. *International Journal of Intelligent Computing and Information Sciences*, 25(2), 74-87.
- [5] Al Lail, M., Garcia, A., & Olivo, S. (2023). Machine learning for network intrusion detection—a comparative study. *Future Internet*, 15(7), 243.
- [6] Gueriani, A., Kheddar, H., & Mazari, A. C. (2023, November). Deep reinforcement learning for intrusion detection in IoT: A survey. In *2023 2nd International Conference on Electronics, Energy and Measurement (IC2EM)* (Vol. 1, pp. 1-7). IEEE.
- [7] Ganapaneni, S. B. A. K. S., & Najana, M. Attention-Based Deep Learning Frameworks for Network Intrusion Detection: An Empirical Study.
- [8] Enisoglu, R., & Rakocevic, V. (2025). A Novel Wavelet Transform and Deep Learning-Based Algorithm for Low-Latency Internet Traffic Classification. *Algorithms*, 18(8), 457.
- [9] Zeng, Y. (2025). CSAGC-IDS: A Dual-Module Deep Learning Network Intrusion Detection Model for Complex and Imbalanced Data. *arXiv preprint arXiv:2505.14027*.
- [10] Puri, V., Chandekar, P., & Chandan, S. (2025). Enhancing IoT Security Through a Hybrid Deep Learning Model: CNN Meets Transformer for Robust Intrusion Detection.

- [11] Rismawan, V. L., & Pramudya, E. R. (2025). Network Intrusion Detection System Using Convolutional Neural Network and Random Forest Classifiers. *INOVTEK Polbeng-Seri Informatika*, 10(2), 753-761.
- [12] Du, C., Guo, Y., & Zhang, Y. (2024). A Deep Learning-Based Intrusion Detection Model Integrating Convolutional Neural Network and Vision Transformer for Network Traffic Attack in the Internet of Things. *Electronics* (2079-9292), 13(14).
- [13] Xiang, B., Zheng, R., Zhang, K., Li, C., & Zheng, J. (2025). FFT-RDNet: A Time-Frequency-Domain-Based Intrusion Detection Model for IoT Security. *Sensors*, 25(15), 4584.
- [14] Zhang, C., Li, J., Wang, N., & Zhang, D. (2025). Research on Intrusion Detection Method Based on Transformer and CNN-BiLSTM in Internet of Things. *Sensors*, 25(9), 2725.
- [15] Koukoulis, I., Syrigos, I., & Korakis, T. (2025). Self-Supervised Transformer-based Contrastive Learning for Intrusion Detection Systems. *arXiv preprint arXiv:2505.08816*.
- [16] Li, J. (2025). Deep Learning Models in Network Intrusion Detection Systems. *Applied Mathematics & Nonlinear Sciences*, 10(1).
- [17] Sun, Z., An, G., Yang, Y., & Liu, Y. (2024). Optimized machine learning enabled intrusion detection 2 system for internet of medical things. *Franklin Open*, 6, 100056.
- [18] Bukhari, O., Agarwal, P., Koundal, D., & Zafar, S. (2023). Anomaly detection using ensemble techniques for boosting the security of intrusion detection system. *Procedia Computer Science*, 218, 1003-1013.
- [19] Vishwakarma, M., & Kesswani, N. (2023). A new two-phase intrusion detection system with Naïve Bayes machine learning for data classification and elliptic envelop method for anomaly detection. *Decision Analytics Journal*, 7, 100233.
- [20] Rajathi, C., & Rukmani, P. (2025). Hybrid Learning Model for intrusion detection system: A combination of parametric and non-parametric classifiers. *Alexandria Engineering Journal*, 112, 384-396.
- [21] Bibers, I., Arreche, O., Alayed, W., & Abdallah, M. (2025). Ensemble-IDS: An Ensemble Learning Framework for Enhancing AI-Based Network Intrusion Detection Tasks. *Applied Sciences*, 15(19), 10579.
- [22] Khan, Adnan Shahid, Zeeshan Ahmad, Johari Abdullah, and Farhan Ahmad. "A spectrogram image-based network anomaly detection system using deep convolutional neural network." *IEEE access* 9 (2021): 87079-87093.
- [23] Hazim Obaid, Zahraa, Behzad Mirzaei, and Ali Darroudi. "An efficient automatic modulation recognition using time-frequency information based on hybrid deep learning and bagging approach." *Knowledge and Information Systems* (2024): 1-18.