Received: October 15, 2024. Revised: December 19, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

HSV-DFEN: A Hybrid Sequential-Visual Deep Feature Extractor Network for Intrusion Detection

Ahmed F. Mutar[1] Leyli M. Khanli[1] Hojjat Emami[2]*

[1]*Department of Computer Engineering, Faculty of Electrical and Computer Engineering, University of Tabriz, Iran.*
[2]*Department of Computer Engineering, Faculty of Engineering, University of Bonab, Bonab, Iran*
* Corresponding author's Email: emami@ubonab.ac.ir

Abstract: Detecting malicious activities in networks has become increasingly challenging as internet usage grows, making network security vital for ensuring secure communication between devices. While machine learning (ML) and deep learning (DL) have been integrated into intrusion detection systems (IDSs), there is limited research exploring their full potential. This paper introduces a hybrid sequential-visual deep feature extractor network (HSV-DFEN) for intrusion detection, using spectrogram transformations to combine visual and sequential data processing techniques. The model leverages convolutional neural networks (CNN) and bidirectional long short-term memory (BiLSTM) networks to extract time-frequency features from network traffic, and employs ensemble learning along with principal component analysis (PCA) for dimensionality reduction. Experimental results on the CICIDS2017 dataset demonstrate that HSV-DFEN achieves an average accuracy of 99.98%, significantly outperforming existing models in detecting various types of attacks, making it an effective solution for anomaly detection in network security.

Keywords: Spectrogram transformations, Deep Learning (DL), Machine Learning (ML), Bidirectional long short-term memory (Bilstm), Principal Component Analysis (PCA).

1. Introduction

The rapid expansion of Internet and communication technologies has led to a substantial increase in network size and the diversity of applications supported. Consequently, the volume of data transmitted across network nodes has escalated, necessitating robust security measures to protect against potential intrusions. One of the effective approaches to providing safe networks is implementing an intrusion detection .

System (IDS) [1]. The IDS is a pivotal tool designed to monitor network traffic and system activities for signs of malicious behavior and policy violations. The primary objective of an IDS is to detect and respond to unauthorized access attempts, thereby safeguarding the integrity of network infrastructure and data [2]. IDS can generally be categorized into two types: network-based IDS (NIDS), which scrutinizes network traffic for unusual activities, and host-based IDS (HIDS), which focuses on individual host systems for signs of intrusion. Initial IDS systems relied on signature-based detection, referred to as SIDS , which utilized predefined patterns of known attacks to identify intrusions[4]. While effective against known threats, SIDS systems struggled with detecting novel attacks, commonly known as zero-day exploits [1].

To address this limitation, anomaly-based IDS (AIDS)approaches emerged, focusing on deviations from normal behavior patterns. AIDS models can identify new emerging threats; however, they are susceptible to generating false alarms due to benign operations appearing unusual under specific conditions [5, 6]. Recent advancements in deep learning (DL) and machine learning (ML) have shown promising results in enhancing IDS capabilities. These technologies enable IDS to learn intricate patterns and identify anomalous behaviors more effectively than traditional methods [7].

This study aims to enhance IDS capabilities by developing a novel hybrid sequential-visual deep feature extractor network (HSV-DFEN) for intelligent intrusion detection. The proposed approach utilizes spectrogram transformation to convert raw network traffic data into spectral images, facilitating the extraction of sequential and visual features. The system employs a bidirectional long short-term memory (BiLSTM) network to capture sequential patterns and a convolutional neural network (CNN) to derive visual characteristics from spectrogram images. These features are fused and subjected to principal component analysis (PCA) to reduce dimensionality and eliminate redundancy. Finally, the extracted features are classified using ensemble learning methods to enhance detection accuracy and robustness.

The performance of the proposed HSV-DFEN model is evaluated using the publicCIC-IDS2017 dataset, which includes diverse cyber-attack types. Comprehensive experiments and comparisons with existing methods demonstrate a significant improvement in detection performance, particularly in addressing imbalanced data and mitigating overfitting. The results confirm the potential of the proposed model in advancing intrusion detection capabilities and providing robust defenses against emerging cyber threats.

To summarize, the contributions of this paper are as follows:

- **Spectrogram transformation preprocessed using short-time Fourier transform (STFT) to extract frequency features in the form of visual patterns**: This strategy leads to efficient identification of attack patterns.
- **Jointly extraction of time and frequency patterns:** Sequential time patterns are extracted using BiLSTM neural networks, while visual frequency patterns are extracted using convolutional neural networks (CNN). These features are stored as vectors and combined to provide comprehensive time-frequency features.
- **Dimensionality reduction using principal component analysis (PCA):** PCA is employed to reduce feature redundancy and dimensionality of feature vectors. This process simplifies model complexity and enhances computational efficiency.

  This combination of techniques presents a comprehensive and efficient approach for detecting attacks in computer networks, addressing complex security challenges effectively.

The structure of the paper is organized as follows: Section 2 provides a summary of related work. Section 3 details the proposed method and its components. Section 4 presents experimental results and discussions. Finally, Section 5 concludes the paper and outlines potential future research directions.

2. Related works

An Intrusion Detection System (IDS) based on deep learning leverages data from commonly used datasets like CIC-IDS 2017 and CIC-IDS 2018, which include various types of attacks. Through the application of algorithms, preprocessing steps are taken to normalize data and address missing values. Time stamps are typically selected and largely excluded from attack datasets using the Random Forest (RF) method. Subsequently, essential features are extracted from these datasets using a Deep Auto encoder (AE), refining the feature set.

In [10], a comprehensive comparison of the initial two datasets yielded promising results for each tested deep learning model. For the CIC-IDS 2017 dataset, the precision, F1-score, and recall were 99.5%, 98.7%, and 99.8%, respectively. Similarly, for the CIC-IDS 2018 dataset, the precision, F1-score, and recall were consistently 99.5% across all algorithms. They showed that the CNN model demonstrated the highest performance outcomes.

In [11], the authors employed data augmentation techniques on some emphasized databases, 5G-NIDD, FLNET2023, UNSW-NB15, and CIC-IDS-2017 to improve the operation of the selected deep learning-based models. The outcomes of the experiments indicated that just the plain model based on a CNN was able to provide an accurate identification of network attacks, further more complex models described only minor improvements to the performance. The findings suggest that the advancement of approaches based on deep learning for intrusion detection can be implemented without much disruption into cyber security paradigms to improve the detection and prevention of contemporary complex network attacks. The outcomes of the study showed that the ID models had high accuracy with the maximum being 91% in the augmented CIC-IDS-2017 database. In [12], the authors described a network intrusions detection system using the oversampling technique to correct the lack of balance in the data and the stacking feature embedding (SFE) approach and PCA for dimensionality reduction. The efficiency of the system is thoroughly evaluated utilizing three sophisticated benchmark datasets: CIC-IDS 2018, CIC-IDS 2017, and UNSW-NB15. They showed that

random forest (RF)and extra trees (ET) models achieved 99. 59% and 99. The results indicate that both the given models, DT and RF, are very accurate with 99 % accuracy. More specifically, it was possible to achieve an average detection rate of 94% on the CIC-IDS-2018 dataset.

In[13], researchers proposed multinomial mixture modeling (MMM) with median absolute deviation and random forest algorithm (MMM-RF) for classifying different types of attacks in a network. The MMM model includes the combination of the Expectation-Maximization (EM)algorithm and median absolute deviation (MAD). After that, the random forest model is applied to the CSE-CIC-IDS2018 database. The presented method achieved an accuracy of approximately 99%. In [14] The authors introduces two intrusion detection and classification models, named Trust-based Intrusion Detection and Classification System (TIDCS) and its accelerated version (TIDCS-A), which exploit a new feature selection algorithm to improve the detection accuracy with less computational overhead.

The authors in [15] introduced a dynamic ensemble algorithm for anomaly detection in streaming imbalanced data within the context of IoT using sample synthesis techniques such as Borderline-SMOTE and a chunk-based strategy for training classifiers based on five real-world datasets.

A new method for increasing the level of security of IDS in IoT-based smart cities proposed in [16] using ensemble techniques such as (SVM, ANN, KNN, LR, and DT). It showed that experimental results on datasets like UNSW-BC15 and CICIDS2017 show that these approaches can detect rare cyberattacks, increasing accuracy, precision, recall, and F1 score.

The authors in [17] proposes a two-stage Intrusion Detection System (IDS) for IoT networks using Naive Bayes classification and elliptic envelope methods to enhance anomaly detection. Stage one classifies the input data into four classes: nominal, integer, binary, and float, using different forms of the Naive Bayes classifier.The second stage filters the benign data extracted in the first stage through an elliptic envelope method.The method achieves a high accuracy rate on NSL-KDD, UNSW_NB15, and CIC-IDS2017 datasets and advocates for enhancing security in IoT networks.

Authors in [18] apply deep learning algorithms—specifically Deep Neural Networks (DNN), Long Short-Term Memory networks (LSTM), and Convolutional Neural Networks (CNN)—to enhance intrusion detection systems in the Internet of Things (IoT) using the CIC-IDS 2017 dataset. It underscores the inadequacies of traditional IDS methods and showcases how deep learning can improve detection

Table 1. Characteristics of anomaly detection methods

Ref.	method	Dataset	Advantage	Disadvantage (Limitation)	ACC
[14]	(TIDCS) Feature Selection algorithm	- NSL-KDD - UNSW	- Effectively Managing High-Dimensional data - Leveraging past node behaviors	- Biases - Challenges in real-time applications	91%
[15]	Dynamic Ensemble Algorithm, Borderline-SMOTE	- MBD - SMD - IoT Botnet - EMOS Cloud - CICIDS2017	- Dynamic Classifier Management - Real-World Applicability	- Computational complexity - Potential for Overfitting	79.5%. 72.7%. 78.4%. 91.5%. 81.0%
[16]	SVM, ANN, KNN, LR, DT	- UNSW-BC15 - CICIDS2017	- Robustness - More comprehensive analysis of data	- Complexity of managing multiple models - Data Dependency	98.8%
[17]	Naive Bayes classifier, elliptic envelope method.	- NSL-KDD - NSW_NB15 - CIC-IDS2017	- Two-Phase Approach - Multiple Data Type Classification	- Complexity of Implementation - Multiclass Classification Limitations	97% 86.9% 98.59%
[18]	DNN, LSTM, CNN	- CIC-IDS2017	- Ability to Handle Complex Data - Adapting Over Time	- Data Requirement - Computational Resources - Complexity of Implementation	98.61% 97.67% 94.61%

accuracy and efficiency. Table 1 summarizes the characteristics of recent anomaly detection methods.

We can structure the prior study by categorizing the approaches based on important methodologies such as data preprocessing , feature extraction, classification algorithms, and benchmark dataset performance.

In Preprocessing Techniques, the majority of the research employ oversampling or augmentation for imbalanced data [12, 15, 16], although such methods either raise computational costs or lack generalizability, as with Feature Extraction Traditional approaches, such as Random Forest and PCA [10, 12, 13], are useful, but they may not fully capture the temporal and visual patterns of network traffic data.

Many discussed approaches do not use sequential and visual intrusion detection features. Where CNN-based approaches [10, 11] focus on visual patterns but have inadequate capture of temporal dependency. RF and SFE approaches [12, 13] focus on feature selection but do not consider integrating the time and frequency domain information, which are essential to identify complicated attacks.

State-of-the-art approaches are typically overfitting [11, 12] or handle imbalanced datasets [15, 16], limiting their practical application.

It can be concluded that the existing proposed methods surpass preceding models in aspects of classification precision and computational performance.

3. Proposed method

The objective of this work is to detect various types of attacks in computer networks.

As shown in Fig. 1 the proposed method comprises six main steps, first input data is preprocessed. Due to the imbalance in the number of samples across different classes, data augmentation is applied. This improves model performance, reduces overfitting, and increases detection accuracy. Next, the augmented data are visualized using spectrogram transformation. In the visualization process, signals are converted into spectral images through a Short-Time Fourier Transform (STFT) to extract frequency features as visual patterns.

Subsequently, sequential data patterns are extracted using a Bidirectional Long Short-Term Memory (BiLSTM) neural network, while the visualized frequency patterns are extracted using a convolutional neural network (CNN). These extracted patterns are organized into feature vectors and combined to provide a wide range of time-frequency features. Afterward, redundancy among the extracted features is reduced using Principal Component Analysis (PCA), which maps the features to a low-dimensional space.

Finally, the resulting feature vectors are classified using Ensemble Learning. By integrating several methods, Ensemble Learning improves the precision of the final model compared to individual models, as different models can correct each other's errors, thereby reducing the likelihood of overfitting. The components of the proposed method are described below.

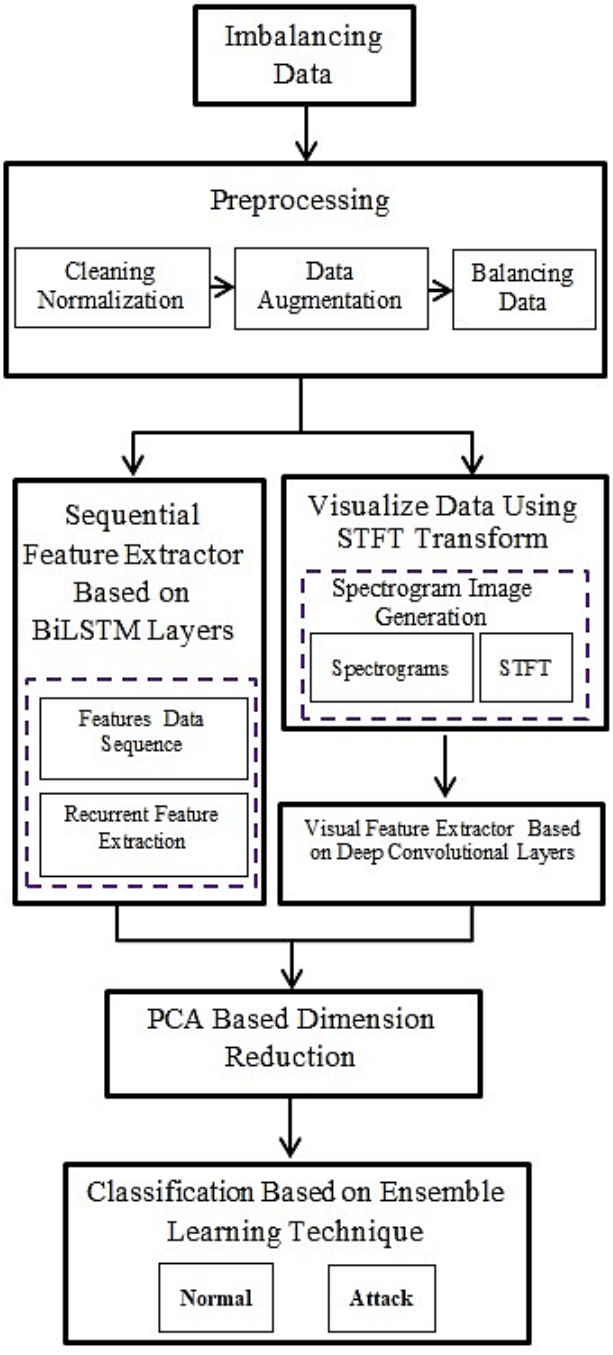

Figure. 1 Working principle of the proposed method

Table 2. Data distribution in CIC-IDS2017 dataset

Seq	Category	No. of records	Training (75%)	Testing (25%)
1.	Benign	25000	18750	6250
2.	DoS	25000	18750	6250
3.	DDoS	25000	18750	6250
4.	Port scan	25000	18750	6250
5.	Brute force	13832	10374	3458
6.	Attack	2003	1503	500
7.	Total Records	118015	88512	29503

3.1. Data preprocessing

In this study, the publicly available CIC-IDS2017 dataset from the Canadian Institute for Cybersecurity is utilized to evaluate the presented model. This database includes records of various types of attack classes, including Port Scan, Brute Force, DoS, DDoS, Web Attack, Botnet, Infiltration, and Heartbleed. Due to the very low number of records for certain attacks, following the approach presented in[19], Botnet, Infiltration, and Heartbleed attacks are grouped into a unique category named "Attack" to ensure the system is trained with sufficient data. Each record in the CIC-IDS2017 dataset comprises 78 numeric features and one target label.

Table 2 lists the data distribution in the CIC-IDS2017 dataset. As shown in this table, 118,015 data samples are used for multiclass intrusion detection, considering seven different classes, including one Benign class and six attack classes. 75% of the dataset is considered for training, whereas the residual 25% is utilized to assess the performance of the suggested method.

3.1.1. Data augmentation

Number of samples in the classes is highly imbalanced. Therefore, in the proposed method, data augmentation is employed to balance the data by increasing the number of samples in the minority classes. This involves generating new samples in the minority classes to balance the number of samples across different classes. In this study, two techniques—scaling and noise addition—are used to increase the diversity and volume of data. Firstly, the data are scaled by multiplying them by various scale factors, allowing the model to adapt to the scaling variations that may occur in real-world data. Secondly, random noise, extracted from a normal distribution with a mean of zero and a low standard deviation, is added to the data. Adding noise helps the model to be more robust against environmental noise and natural data fluctuations. These two methods increase the diversity and number of training samples, improving the model's performance under various and more realistic conditions Fig. 2 and Table 3 show the difference between the distribution of data before and after using the augmentation techniques.

3.2. Feature visualization using spectrogram

A spectrogram provides a visual representation of the frequency spectrum over time. Spectrogram images provide a high degree of frequency precision, but this comes with a trade-off in time precision. This image is shown as a color map in terms of time and frequency, helping to understand how the frequency spectrum changes. Essentially, spectrograms present spatial and temporal features in a visual format that can be easily learned by deep convolutional neural networks (CNNs) [20].

The segmentation of waveforms into short-term segments forms the basis for time-frequency techniques. In short-time Fourier transform (STFT), there are two steps: first, the data are divided into equal segments, and then the Fourier transform is applied to each segment. This operation generates time-frequency information, known as a spectrogram. The STFT approach heavily relies on the choice of an appropriate window. The frequency resolution of a rectangular window is inadequate. A triangular window is better than a rectangular one because it provides a decreasing frequency range. The functions employed as the primary window function in short-time Fourier transform (STFT) must possess a confined quantity of energy. As a result of this limitation, the frequency axis in STFT is also reduced to enhance the distinction of the time axis. The window should be widened to improve the frequency axis selectivity[21, 22].

The continuous-time spectrogram Equation is as follows:

$$X(t,f) = \int_{-\infty}^{+\infty} x(t).w(t-\tau).e^{-j\pi/\tau}\,d\tau \qquad (1)$$

Where:

X(t,f) is the STFT at time t and frequency f, x(t) is the input signal, w(t−τ) is the window function that moves along the signal, $e^{-j\pi/\tau}$ represents the Fourier transform's frequency domain component.

The window function w(t−τ) plays a crucial role in localizing the signal in time, and f is the frequency to which the signal is transformed.

Signals that need to be transformed are often divided into equal frames and then transformed using discrete-time methods. Artifacts at the borders resulting from this method are minimized. These

Table 3. Data distribution after augmentation

Category	No. of records	Training (75%)	Testing (25%)
Benign	25000	18750	6250
DoS	25000	18750	6250
DDoS	25000	18750	6250
Port scan	25000	18750	6250
Brute force	25000	18750	6250
Web attack	25000	18750	6250
Attack	25000	18750	6250
Total Records	175000	131250	43750

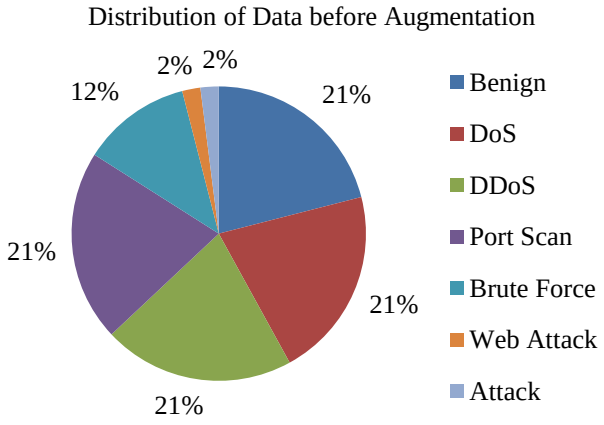

Figure. 2 Distribution of data: (a) before and (b)after augmentation

frames are then subjected to the Fourier Transform[23]. Consequently, the following matrix, which preserves the magnitude and phase for each time and frequency point, is constructed[14]:

$$X(\tau,k) = \sum_{n=1}^{N} x[n]w[n-\tau]e^{-jnk} \quad (2)$$

Where:

X(τ,k) is the STFT at discrete time step τ and frequency bin *k*,

x[n] is the discrete signal, w[n−τ] is the discrete window function,

N is the total number of points in the Fourier transform.

The linear spectrogram represents the squared amplitude of the short-time Fourier transform (STFT), as indicated in Equation .

$$S(\tau,k) = |X(\tau,k)|^2 \quad (3)$$

Where:

S(τ,k) is the spectrogram at time τ and frequency bin k, $|X(\tau,k)|^2|$ denotes the squared magnitude of the STFT.

Fig. 3 shows an example of the visualized features for each class of data using spectrogram transformation.

3.3. Feature extraction

In the proposed method, two neural networks, BILSTM and CNN, are utilized to extract sequential and spatial features. Combining the sequential and spatial features extracted by these networks provides a diverse range of features. The feature extraction process using each network is described separately below.

3.3.1. Sequential feature extraction.

BILSTM neural networks can identify complex patterns in sequential data due to their unique structure, which includes input, output, and forget gates that allow memory units to retain relevant information and discard irrelevant information. This makes BILSTM suitable for extracting sequential features.

The architecture of the BILSTM neural network used includes a combination of fully connected layers, dropout layers, and BILSTM layers.

The BILSTM layers in this network are employed to model and extract temporal and sequential features from data. These layers, with their long-term and short-term memory capabilities, can learn complex relationships and temporal dependencies in intrusion detection data, which is crucial for identifying suspicious patterns and abnormal behaviors in the network.

Following the BILSTM layers, Dropout layers are employed to prevent overfitting. These layers randomly deactivate some units of the network

during training, which helps the model generalize better and adapt to new and unseen data. In the final stages, fully connected layers are used, where all neurons are connected to all neurons in the subsequent layer. These layers are used to combine and aggregate the features extracted by the previous layers and make the final decision for data classification. The combination of these layers allows the neural network to extract significant features from the intrusion detection data, achieving high accuracy and efficiency in identifying attacks and intrusions.

Fig. 4 presents the BILSTM neural network structure used in the proposed model.

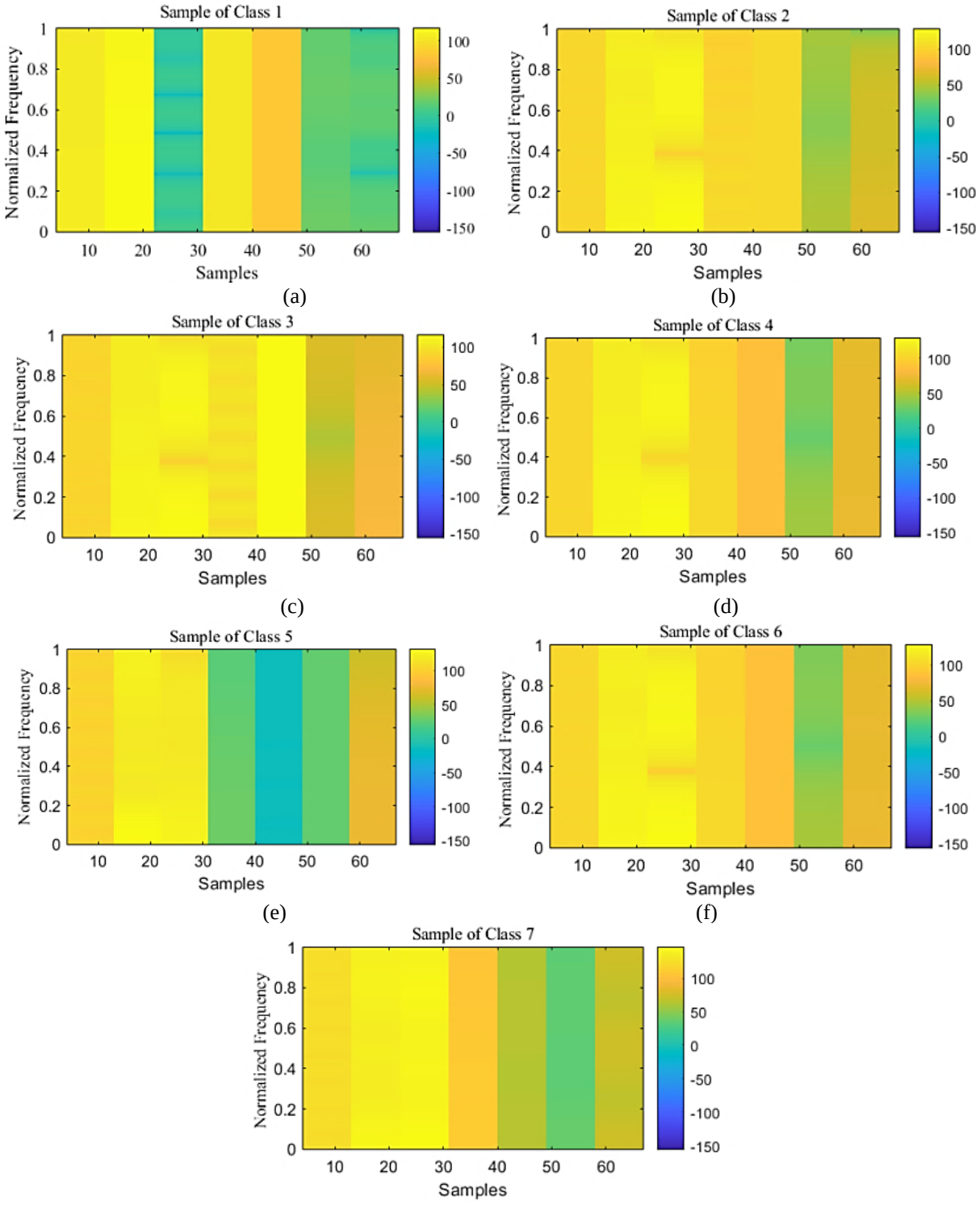

Figure. 3 Examples of visualized features for each class using spectrogram transform: (a) sample of class 1, (b) sample of class 2, (c) sample of class 3, (d) sample of class 4, (e) sample of class 5, (f) sample of class 6, and (g) sample of class 7

3.3.2. Spatial feature extraction

Feature extraction from the visualized frequency patterns generated by the spectrogram transformation is performed by leveraging a CNN network. The visualized information is applied to a deep CNN to extract spatial features presented in the visual data.

CNNs employ convolutional layers to derive intricate attributes from images. CNNs can identify multi-scale features, meaning they can recognize features at various scales within the image. Fig. 5 illustrates the structure of the CNN network used in this work .

The proposed architecture for feature extraction from spectrogram images employs a CNN comprising four types of layers with three training epochs, including convolutional layers , ReLU activation layers, pooling layers, and fully connected layers.

The convolutional layers are the first stage in this network, applying various filters to the image to identify local patterns and characteristics, including boundaries, colors, and vertices.

Following convolutional layers, ReLU layers are used as activation functions, converting negative values to zero, enabling the network to model non-linear relationships, and preventing issues like the vanishing gradient problem.

The pooling layers are used to reduce the dimensions of the feature maps and decrease the number of parameters and computations. These layers help the model preserve essential spatial features and provide greater robustness to small variations and noise by dimensionality reduction. In the final stages, fully connected layers are employed, each with 20 neurons, allowing the model to learn complex features where all neurons are connected to all neurons in the subsequent layer. These layers combine the extracted features and make the final decision.

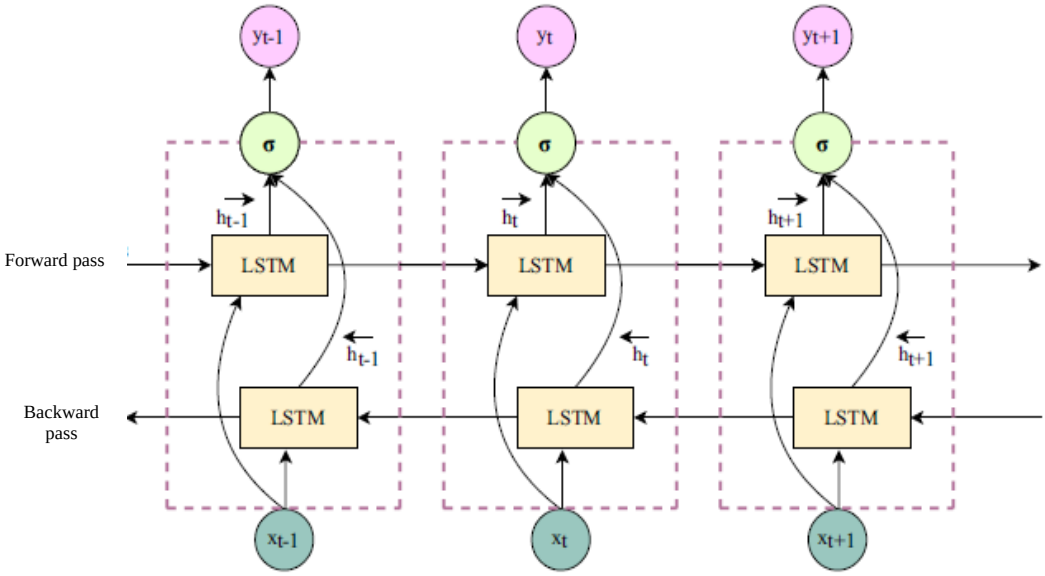

Figure. 4 BILSTM neural network structure[15]

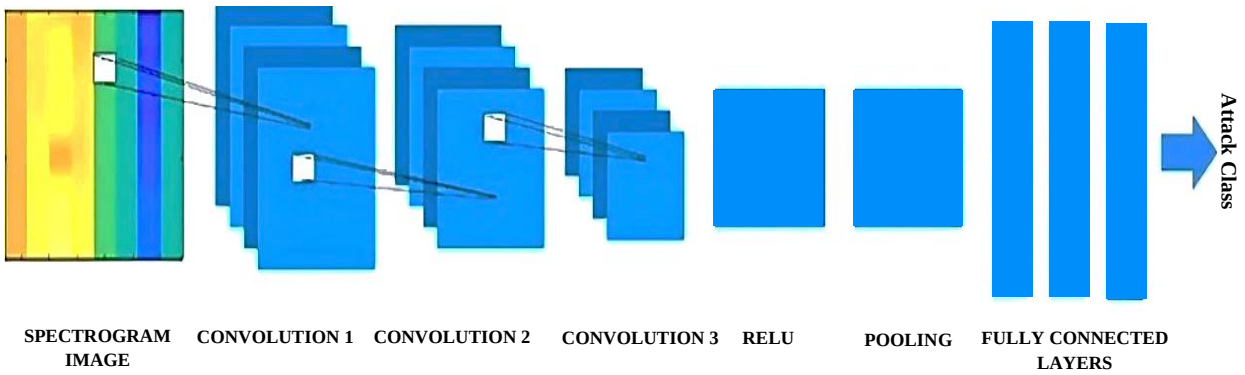

Figure. 5 Structure of the CNN used in this research

The number of training epochs was set to 3 for the CNN, balancing between sufficient training time to learn the features and avoiding overfitting.

This combination of layers works cohesively to extract useful and meaningful information from the spectrogram images, preparing it for analysis and final decision-making. This approach enables the neural network to efficiently and accurately identify and extract critical features in spectrogram images

3.4. Dimensionality reduction

In the proposed model, principal component analysis (PCA) is used to reduce the number of variables in the dataset while preserving the maximum variance (information) present in the data. This method transforms the original dataset into a set of principal components using a change of basis. In this technique, the data is first centralized using Eq. (4), where the mean of each feature is subtracted from its value:

$$X_{centered} = X - \mu \tag{4}$$

In Eq. (4), X represents the data matrix and μ is the vector of their means.

Then, the covariance matrix of the centered data is calculated using Eq. (5):

$$C = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \mu)(X_i - \mu)^T \tag{5}$$

Where n is the number of data points.

Xi represents the i-th data point (a row in the data matrix).

Next, eigenvectors and eigenvalues are computed. In other words, we decompose the covariance matrix to obtain the eigenvectors v_i and eigenvalues λ_i.

$$C_{vi} = \lambda_i v_i \tag{6}$$

Where:
vi are the eigenvectors (directions of maximum variance), λi are the eigenvalues, which represent the amount of variance explained by each eigenvector.

Then, the eigenvectors are sorted based on their corresponding eigenvalues, and *k* eigenvectors with the highest eigenvalues are selected. Finally, the original data is mapped to the new space of principal components using Eq. (7):

$$X_{reduced} = X_{centered}W \tag{7}$$

Where *W* is a matrix containing the selected k eigenvectors.

Reducing the number of features allows our proposed model to train faster and become simpler. Additionally, the PCA maps the data to a lower-dimensional space, which leads to computational complexity reduction. Moreover, removing dimensions with less information helps to reduce noise and improve the model's performance.

3.5. Data classification using ensemble learning

In this work, the bagging technique was employed to classify data. Bagging (bootstrap aggregating) is a powerful ensemble learning method used to reduce the variance of machine learning models and improve their accuracy. In the bagging phase, random and resampled subsets of the dataset are used to create different sets, and then learning models are trained on these subsets. In this technique, the outputs of all trained models are combined, and the majority vote is considered as the final prediction.

By training different models on different samples, bagging reduces the variance of the final model, thus preventing overfitting. Combining multiple models in this way improves the accuracy of the final model because different models compensate for each other's mistakes. Additionally, bagging creates more stable and robust models by reducing the sensitivity of the model to noise present in the training data. Fig. 6 illustrates the framework of the ensemble learning method.

4. Experimental results and discussion

In this section, we evaluate the performance of the proposed method and compare it with previous state-of-the-art models. The proposed technique was implemented in MATLAB R2023b on a system with an Intel Core i7 CPU, 8 GB RAM, and an NVIDIA Quadro K2000 GPU.

4.1. Benchmark dataset

We utilized the CICDS2017 dataset for experiments, with 75% of the dataset used for training after the data augmentation stage, and the remaining 25% for testing. To prevent overfitting, we employed K-cross validation with K=10 in this research.

4.2. Performance measures

Four performance measures, including accuracy, precision, recall, and F-score, are used to evaluate the performance of the proposed method compared to other works. These metrics provide a comprehensive assessment of the model's ability to classify different

types of attacks and intrusions of network traffic. The metrics are described as follows:

Precision: This is the proportion of instances that are accurately identified as attacks out of all instances that are predicted as attacks [8].

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

Where:

TP (True Positives) are instances correctly predicted as attacks.

FP (False Positives) are instances incorrectly predicted as attacks.

Recall: This is the proportion of all instances correctly identified as attacks out of all instances that are truly attacks [24][8].

$$Recall = Detection\ Rate = \frac{TP}{TP+FN} \quad (9)$$

Where:

FN (False Negatives) are actual attacks that were not identified by the model.

Accuracy: This refers to the proportion of instances that are correctly classified out of the total number of instances.

Also known as detection accuracy, it serves as a valuable performance metric, particularly when dealing with a balanced dataset [8].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

Where:

TN (True Negatives) are instances correctly identified as normal traffic.

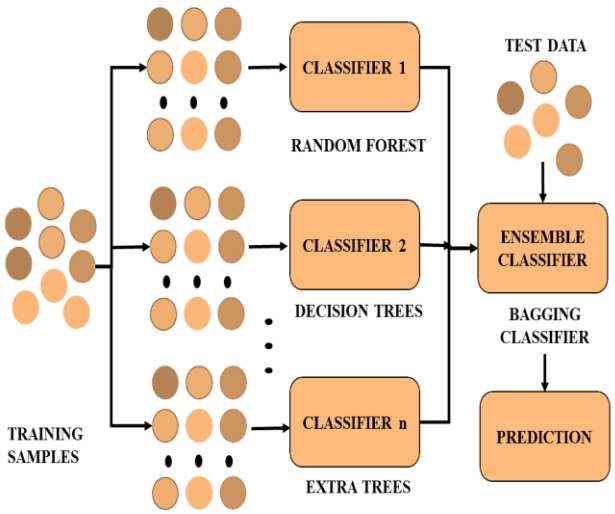

Figure. 6 Structure of the proposed ensemble learning model

F-Score: This is the harmonic mean of precision and recall, serving as a statistical method to evaluate a system's accuracy by taking into account both its precision and recall. In essence, it provides a balanced perspective on the system's performance[8].

$$F\ Measure = 2\left(\frac{Precision \times Recall}{Precision+Recall}\right) \quad (11)$$

4.3. Results

The ROC curve that is displayed in Fig. 7 graphically represents the true positive rate (TPR) versus the false positive rate (FPR) at different threshold levels, illustrating the balance between the classifier's specificity and sensitivity. An effective classifier gravitates towards the upper-left corner of the ROC curve, indicating a high true positive rate (TPR) and a low false positive rate (FPR). On the other hand, an ineffective classifier leans towards the lower-right corner, signifying a low TPR and a high FPR. A classifier that makes random predictions falls on the diagonal line of the ROC curve, where the TPR and FPR are identical.

As observed in Fig. 7, the ROC curve for the proposed method exhibits high TPR and low FPR, with the break-even point of the curve near the upper-left corner. Thus, the model's performance in detecting attacks with high accuracy can be deduced.

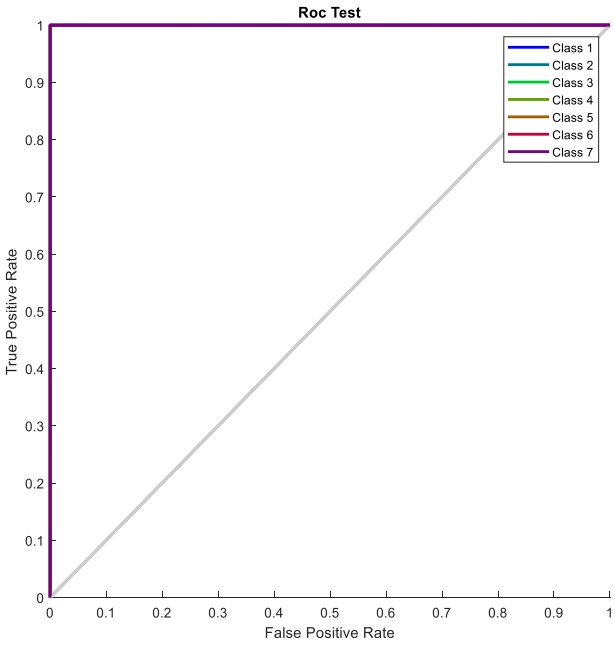

Figure. 7 The ROC curve on the training dataset generated by the proposed method

Table 4. Confusion matrix of classification outcomes generated by the proposed method.

		Actual								
		Attack	**Benign**	**Brute force**	**DDoS**	**DoS**	**Port scan**	**Web Attack**		
Predicted	**Attack**	6373	0	0	0	0	0	0	6373	100%
	Benign	0	6167	0	0	0	0	0	6167	100%
	Brute Force	0	0	6240	0	0	0	0	6240	100%
	DDoS	0	0	0	6323	0	0	0	6323	100%
	DoS	0	0	0	9	6240	0	0	6249	99.85%
	Port scan	0	0	0	3	0	6156	0	6159	99.95%
	Web Attack	0	0	0	0	0	0	6239	6239	100%
		6373	6167	6240	6335	6240	6156	6239		
		100%	100%	100%	99.81%	100%	100%	100%		

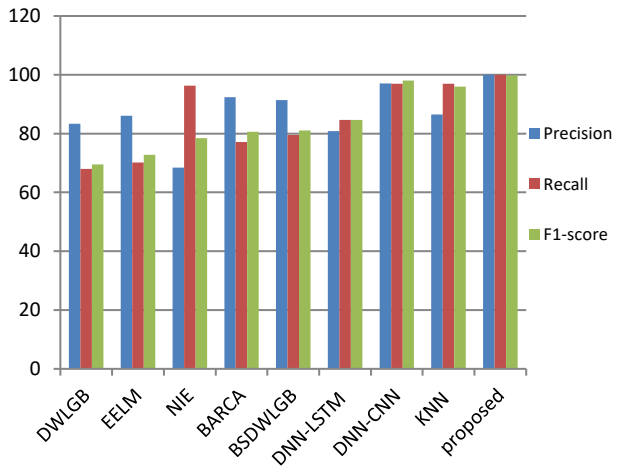

Figure. 8 The results generated by comparison algorithm in terms of performance metrics

Table 4 shows the confusion matrix of classification outcomes generated on the CICIDS2017 dataset. In this matrix, rows represent the actual classes, and columns represent the predicted classes by the model. Each cell in the matrix indicates the number of samples assigned from the actual class to the predicted class. Values on the main diameter of the matrix, such as 6373 for "attack", 6167 for "benign", and 6240 for "brute force", represent the number of samples correctly detected as positive (Tp).Values outside the main diameter, such as 9 samples "DDoS" which are wrongly classified as "DoS" or 3 "DDoS" samples which are classified as "port scan" which represents the errors of the model in recognizing the classes.

In Table 3, the last row for each class shows the amount of TPR ($\frac{\text{Tp}}{\text{Tp+Fp}}$) which is calculated by dividing the correct positive detections by the total positive detections. As seen in Table 3, the TPR rate for the DDoS class is 99.81% and for the other classes, it is 100%. In addition, the last column shows the level of precision for each class.

This rate for DoS and port scan attacks is equal to 99.85% and 99.95%, respectively. Also, the amount of precision criterion for other classes is equal to 100%. According to the results, it can be seen that the proposed method has shown a verygood performance in correctly detecting all types of attacks

The numerical values of F-measure, recall , and precision obtained by the comparison algorithm on the CICIDS2017 dataset shown in Fig. 8. The results confirm that the suggested model achieved high values for precision, recall, and F-score among all existing methods. After the proposed method, neural networks DNN-CNN and DNN-BILSTM show better performance compared to other methods.

The improvement rate of the proposed method compared with its counterparts is shows in Fig. 9.

To effectively verify the performance of the proposed method has been compared with other techniques like TIDCS [14], (BSDWLGB , DWLGB, EELM, NIE, and BARCA)[15] ,KNN [16] NB elliptic-envelope [17] ,(DNN BILSTM,DNN CNN) [18] and In terms of overall performance, our proposed is superior to the all anomaly detection algorithms of listed in Table 5 based on the accuracy measure. It is worth mentioning that the accuracy of the proposed technique was calculated by averaging the results of 30 simulation tests.

The reason for this is that results vary randomly in each run, so an average of 30 runs was taken to obtain the desired results.

The main reason for the high accuracy of the proposed method is the balanced data augmentation technique using the augmentation strategy and the extraction of a diverse spectrum of features with the

help of CNN and the BILSTM model. This enables the classifier to distinguish classes with higher accuracy. Moreover, reducing the dimensionality of features using the PCA algorithm decreases the complexity of the data and consequently enhances the accuracy of the group-based learning technique in classifying attacks

5. Conclusion

This study introduced a novel hybrid sequential-visual deep feature extractor network (HSV-DFEN) for developing an intelligent intrusion detection system using spectrogram transformation. Our method successfully captured time-frequency patterns in network traffic by merging CNNs for visual feature extraction with BiLSTM networks for sequential feature extraction.

It offered a robust and comprehensive solution for detecting various attack types, including DoS, DDoS, Port Scans, and Brute Force Attacks. The model's performance was rigorously evaluated using the CICIDS2017 dataset, with 10-fold cross-validation (K=10) applied to prevent overfitting.

The experimental findings consistently showed that the suggested model was superior to the current approaches in terms of accuracy, precision, recall, and F-score. When a 99.98% accuracy percentage was attained, the other classes' precision and recall criteria were 100%. Additionally, the F1-Score was 99.1%. Using several models, ensemble learning significantly improved the classification process and produced higher accuracy and robustness than standalone models.

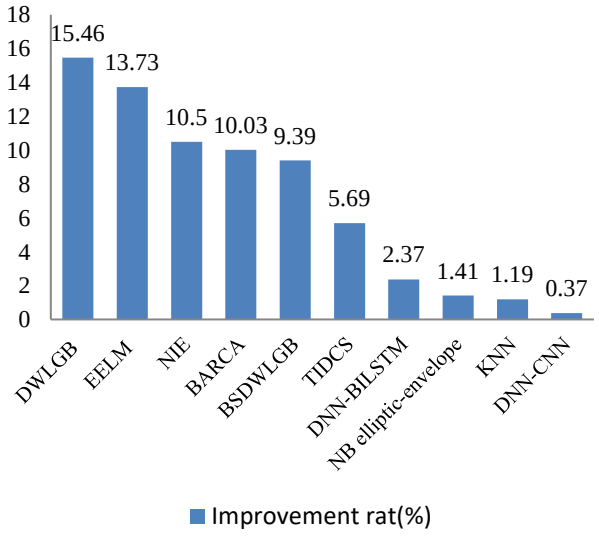

Figure. 9 The improvement rate of the proposed method HSV-DFEN vs. counterpart algorithms

Table 5. Comparison Accuracy of Different Methods on the CICIDS-2017 Dataset

Method	Accuracy
DWLGB	86.59
EELM	87.91
NIE	90.48
BARCA	90.87
BSDWLGB	91.40
TIDCS	94.6
DNN BILSTM	97.67
NB elliptic-envelope	98.59
KNN	98.8
DNN CNN	99.61
Proposed Method	**99.98**

Despite its strong performance on established attack classes, the model's ability to detect zero-day attacks remains limited due to its reliance on pre-existing datasets. Addressing this limitation could involve integrating additional anomaly detection mechanisms to handle novel, previously unseen attack patterns better. Future research should focus on improving the model's real-time functionality, making it more suitable for deployment in real-world network environments. Enhancing computational efficiency through techniques like model quantization or pruning could enable the model to operate in dynamic, resource-constrained settings, further advancing its practical applicability in modern network security.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, methodology, and formal analysis Ahmed F. M.; investigation, Ahmed F. M.; writing-original paper draft, Ahmed F. M.; writing review and editing: Ahmed F.M. and Dr. Hojjat E.; supervision: Dr. Leyli M. K. and Dr. Hojjat E.

Notation List

Symbol	Meaning
X(t,f)	Short-Time Fourier Transform (STFT) at time
x(t)	Input signal in the time domain.
w(t−τ)	Window function used for segmenting the signal.
$e^{-j\pi/\tau}$	Frequency domain component of the Fourier Transform.
X(τ,k)	Discrete STFT at time step

S(τ,k)	Spectrogram, the squared magnitude of the STFT.
X centered	Data matrix after centering.
μ	Mean vector of the data.
C	Covariance matrix.
λi	Eigenvalue, representing the variance explained by each principal component.
Vi	Eigenvector, direction of maximum variance.
W	Transformation matrix containing selected eigenvectors.
TN	True Negatives
FN	False Negatives
TP	True Positive
FP	False Positive

References

[1] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges", *Cybersecurity*, Vol. 2, No. 1, pp. 1-22, 2019.

[2] N. Kaja, A. Shaout, and D. Ma, "An intelligent intrusion detection system", *Applied Intelligence*, Vol. 49, pp. 3235-3247, 2019.

[3] A. Thakkar and R. Lohiya, "A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions", *Artificial Intelligence Review*, Vol. 55, No. 1, pp. 453-563, 2022.

[4] B. S. Bhati, C. S. Rai, B. Balamurugan, and F. Al-Turjman, "An intrusion detection scheme based on the ensemble of discriminant classifiers", *Computers & Electrical Engineering*, Vol. 86, p. 106742, 2020.

[5] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system", *IEEE Access*, Vol. 7, pp. 41525-41550, 2019.

[6] U. S. Musa, M. Chhabra, A. Ali, and M. Kaur, "Intrusion detection system using machine learning techniques: A review", In: *Proc. of 2020 international conference on smart electronics and communication (ICOSEC)*, pp. 149-155, 2020.

[7] M. Imran, N. Haider, M. Shoaib, and I. Razzak, "An intelligent and efficient network intrusion detection system using deep learning", *Computers and Electrical Engineering*, Vol. 99, p. 107764, 2022.

[8] L. Mohammadpour, T. C. Ling, C. S. Liew, and A. Aryanfar, "A survey of CNN-based network intrusion detection", *Applied Sciences*, Vol. 12, No. 16, p. 8162, 2022.

[9] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: hybrid deep neural network for network intrusion detection system", *IEEE Access*, Vol. 10, pp. 99837-99849, 2022.

[10] R. Selvam and S. Velliangiri, "An Improving Intrusion Detection Model Based on Novel CNN Technique Using Recent CIC-IDS Datasets", In: *Proc. of in 2024 International Conference on Distributed Computing and Optimization Techniques (ICDCOT)*, pp. 1-6, 2024.

[11] R. Mohammad, F. Saeed, A. A. Almazroi, F. S. Alsubaei, and A. A. Almazroi, "Enhancing Intrusion Detection Systems Using a Deep Learning and Data Augmentation Approach", *Systems*, Vol. 12, No. 3, p. 79, 2024.

[12] M. A. Talukder et al., "Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction", *Journal of Big Data*, Vol. 11, No. 1, p. 33, 2024.

[13] M. Hammad, N. Hewahi, and W. Elmedany, "MMM-RF: A novel high accuracy multinomial mixture model for network intrusion detection systems", *Computers & Security*, Vol. 120, p. 102777, 2022.

[14] Z. Chkirbene, A. Erbad, R. Hamila, A. Mohamed, M. Guizani, and M. Hamdi, "TIDCS: A dynamic intrusion detection and classification system based feature selection", *IEEE Access*, Vol. 8, pp. 95864-95877, 2020.

[15] J. Jiang et al., "A dynamic ensemble algorithm for anomaly detection in IoT imbalanced data streams", *Computer Communications*, Vol. 194, pp. 250-257, 2022.

[16] O. Bukhari, P. Agarwal, D. Koundal, and S. Zafar, "Anomaly detection using ensemble techniques for boosting the security of intrusion detection system", *Procedia Computer Science*, Vol. 218, pp. 1003-1013, 2023.

[17] M. Vishwakarma and N. Kesswani, "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*, Vol. 7, p. 100233, 2023.

[18] J. Jose and D. V. Jose, "Deep learning algorithms for intrusion detection systems in internet of things using CIC-IDS 2017 dataset", *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 13, No. 1, pp. 1134-1141, 2023.

[19] A. S. Khan, Z. Ahmad, J. Abdullah, and F. Ahmad, "A spectrogram image-based network anomaly detection system using deep convolutional neural network", *IEEE Access*, Vol. 9, pp. 87079-87093, 2021.

[20] Z. Neili and K. Sundaraj, "A comparative study of the spectrogram, scalogram, melspectrogram and gammatonegram time-frequency representations for the classification of lung sounds using the ICBHI database based on CNNs", *Biomedical Engineering/Biomedizinische Technik*, Vol. 67, No. 5, pp. 367-390, 2022.

[21] S. Sandoval and P. L. De Leon, "Recasting the (Synchrosqueezed) short-time Fourier transform as an instantaneous spectrum", *Entropy*, Vol. 24, No. 4, p. 518, 2022.

[22] M. Li, Y. Liu, S. Zhi, T. Wang, and F. Chu, "Short-time Fourier transform using odd symmetric window function", *Journal of Dynamics, Monitoring and Diagnostics*, Vol. 1, No. 1, pp. 37-45, 2022.

[23] Ö. F. Alçin, S. Siuly, V. Bajaj, Y. Guo, A. Şengu, and Y. Zhang, "Multi-category EEG signal classification developing time-frequency texture features based Fisher Vector encoding method", *Neurocomputing*, Vol. 218, pp. 251-258, 2016.

[24] A. Mutar and H. Dway, "Smoke detection based on image processing by using grey and transparency features", *J Theor Appl Inf Technol*, Vol. 96, No. 21, pp. 6995-7005, 2018.