Received: July 4, 2023. Revised: September 13, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Parallel ABILSTM and CBIGRU Ensemble Network Intrusion Detection System

N. Girubagari[1]* **T. N. Ravi[1]**

[1]*Department of Computer Science, Thanthai Periyar Government Arts and Science college (Autonomous),*
Affiliated to Bharathidasan University, Tiruchirappalli, India
* Corresponding author's Email: ngiruba@gmail.com

Abstract: Perpetual improvements are happening in communication networks, both in hardware and software. This constant improvement provides better communication speeds and improves the user experience. The concept of smart cities is a boon that evolves towards the complete automation of the city and the responsible conservation of natural resources. Modern communication technologies support the gradual growth of smart cities simultaneously; the changes in the network architectures open a gateway to attackers, making the cyber network vulnerable. Manual construction and enhancement of network security schemes and protocols are tedious, time-consuming processes that may not be applicable in real time. This dynamic non-deterministic problem can be solved by combining the combination of Artificial Intelligence techniques that can do marvels in anomaly detection in a typical cyber network behaviour. This work, named parallel ABILSTM and CBIGRU ensemble network intrusion detection system (PACENIDS), is intended to use an ensemble of Altered Bi-directional Long Short-Term Memory (ABILSTM) and Customized Bi-directional Gated Recurrent Unit (CBIGRU) to improve the detection of real-time network intrusion attempts with more accuracy and to prevent the network from intimidating attacks on time. The parallel operational nature of the proposed algorithm ensures a swifter performance towards attack detection. This paper uses an impact based fuzzy feature selection algorithm to improve the performance of the proposed approach. The NSL-KDD dataset is used to evaluate the suggested approach. The proposed PACENIDS achieves 96.59% for binary classification, 94.47% and 97.67% for multiclass without and with feature selection, respectively. The experimental result shows that the suggested ensemble approach increases accuracy and precision and reduces the false alarm rate in the target intruder detection system.

Keywords: Artificial intelligence, Attacks detection, Anomaly detection, Network security, Long short-term memory, Gated recurrent unit.

1. Introduction

Communication networks play a vital role in the day-to-day operations of this newfangled internet of things (IoT) enabled smart city environments. It is indubitably understood that a small network glitch can trigger serious chaos in the regular operations of the world [1]. Due to the convenience gained by the development in the communication network industry, communication gadgets are treated today as an extension of one's physical body. This dependency creates a great probability of increasing attempts to unauthorized access and gaining access to delicate private data in the network [2]. Sometimes, the victim may be an individual, a company, or an entire organization/government. Recent statistics show that most attackers target large organizations to gain profit in less duration [3]. Therefore, cyber security and network security are inevitable entities of the world in this information age. Thus, an efficient attacks detection system in network community is the immediate need for network, cyber, and Information security.

Almost all organizations use some strategic approaches to vanquish intruder attacks on the devices and networks with the help of firewalls, intrusion preventions systems, customized security protocols, network access controls and security information and event management [4]. Several techniques have been evolved to discern intruder activities using anomaly-based, signature-based, statistics based, pattern-based and rule-based systems [5]. The learning-based models and deep learning algorithms also contribute significantly to the detection process [6].

Recent studies show that hybridization, stack and ensemble of multiple approaches can achieve higher detection accuracy and precision[7]. At the same time, the accumulated processing time of more than one method in the ensembles causes a notable increase in the processing time[8]. Higher processing times prevent a method's applicability in real-time environments since a server must validate massive network transactions in a fraction of a second. Therefore, it is important to balance improving the accurate detection of attacks and maintaining overall network performance [9]. This work introduces an ensemble to amplify the detection process's accuracy and maintain the running speed by incorporating a parallel processing methodology.

Support vector machines (SVM), decision trees (DT), k-nearest neighbours (KNN), artificial neural networks (ANN), and deep neural networks (DNN) are just a few of the machine learning techniques that researchers frequently employ to identify network intrusion. The performance of these methods depends on datasets. For most classification models, these datasets are computationally expensive because they frequently require many features for training. Additionally, using many features may lead to poor performance because some features could be redundant or unimportant to a model's performance. Therefore, feature selection must be done before training to remove duplicate and unnecessary dataset features. And the traditional machine learning algorithms decrease the performance due to the large dataset and increase the complexity. To overcome these issues, this article makes the following notable contributions:

- A new learning ensemble model for attacks detection
- A novel impact based fuzzy features selection (IFFS) model for features selection.
- An improvement on accuracy and false alarm rate over average processing time.

The remainder of this article is organized as follows: Section 2 reviews the existing anomaly detection methods for finding attacks or intrusions and list out its limitations. Section 3 describes the preliminary concepts of Bidirectional LSTM and GRU methods. Section 4 explains the proposed ensemble method with the features selection algorithm Impact based Fuzzy Feature Selection (IFFS). Section 5 analyzes the performance of the proposed method with the existing methods, with possible comparison between them and section 6 concludes the research paper.

2. Related works

The convolutional neural network (CNN), a kind of deep learning architecture, is used by researchers to detect attacks or intrusions. The CNN approaches use pixel data and image processing concepts to identify hidden patterns and abnormalities. A set of the most relevant intruder detection methods are chosen with care to study the methodologies involved and compare the performance parameters. The selected methods are meticulously learned about their working principles, advantages, and limitations.

An improved anomaly-based CNN model is proposed in [10] for network intrusion detection. It consisted of a two-step preprocessing method for feature selection and a CNN-based classifier for Anomaly-based detection. An innovative approach is used here for applying CNN in network intrusion detection. Dimensionality reduction and feature engineering are performed to integrate the deep feature synthesis process into the proposed model. The performance of the proposed method is measured through two benchmark datasets, network security laboratory-knowledge discovery in datasets (NSL-KDD) and the 2015 dataset based on the University of New South Wales (UNSW-NB15). Moderate accuracy and recall are identified as the advantages of this method, whereas the applied convolutional architecture takes more time to train the system, which is a possible limitation.

The author in [11] proposed an efficient method involving fast deep learning to detect network intrusion without manual feature extraction (F-CNN). The author created a preprocessing technique to receive multi-packets in an input unit that compresses the large raw traffic input data. Complications of this algorithm have been reduced significantly by this. A four-layer convolutional neural network model is used for network intrusion detection. Thus, the technique gains an advantage by improving detection accuracy and training efficiency. The F-CNN model handles the learning model's wavelength, direction, Phase offset, Aspect ratio, and standard deviation components. The results are compared with CNN-LSTM and IDS-DNN models, and the Benign, FTP Brute force, SSH brute force, DoS golden eye, DoS slowIoris and infiltration attacks are detected more accurately. The preprocessing stage adds some extra time for a higher number of steps to the overall process, which is noted as the limitation of F-CNN work.

The author [12] proposed a novel model named assessing deep neural network and shallow for network intrusion detection systems in cyber security (ADNN). After a detailed study of several network intrusion detection models such as DNN1, DNN2, DNN3, DNN4, DNN5, adaboost, decision tree, K-nearest neighbour, linear regression, naïve bayes, random forest, support vector machine-linear and radial basis function (RBF) kernel models with DARPA/KDDCup-'99 datasets, a 7-layer based deep learning neural network architecture is proposed in ADNN work for network intrusion detection. Keras tool with tensor-flow library is used to test the selected methodologies. A tensor streaming model with Nvidia GK1 10BGL-Tesla-k40 GPU executes the selected learning models in ADNN. Improved F1-score of 3-layer DNN is the stated advantage in ADNN work, whereas the higher processing time of ADNN proposed 7-layer DNN model is the limitation.

A bayesian hyperparameter optimization for deep neural network-based network intrusion detection (BHODNN) is proposed in [13] to improve the reliability in detecting modern, sophisticated and unpredictable security attacks. The operation of BHODNN work is based on hyperparameter tuning in deep learning. A Bayesian optimization-based auto-tuning method is introduced for hyperparameter tuning to improve the deep learning architecture. Several tuning factors are used to verify the activation functions, which are used to enhance the model. These functions include rectified linear unit (ReLU) and hyperbolic tangent (TanH). A bayesian optimization technique is used for global minimization. The Gaussian process is used to improve the analytical traceability, and the expected improvement model is used to maintain the expectation-exploitation ratio in the fine-tuning process. The NSL-KDD dataset is used as an evaluation dataset for the BHODNN method. The novel hyperparameter fine-tuning methodology can be the advantage of BHODNN work, whereas average performance in terms of attack detection and precision is the observed limitation of this work.

The author in [14] contributed a deep belief network integrating improved kernel-based extreme learning machine for network intrusion detection (DBN-EGWO-KELM). The method is produced to detect network intrusion attacks using a deep belief network. Generally, conventional back propagation networks use random values as seeds during training. The author eliminates some training issues caused by initial random seeds by providing determined initial seeds to improve. DBN-EGWO-KELM is designed to improve the local optima, thus reducing the training time by fast neural network convergence. kernel-based extreme learning machine (KELM) is used as the core, and the seed initialization is optimized by the enhanced grey wolf optimization (EGWO) technique in DBN-EGWO-KELMwork. A novel inner and outer hunting principle is introduced to the standard grey wolf optimization to bring up the EGWO algorithm. Restricted boltzmann machines, deep belief neural network pre-training, backpropagation based supervised fine-tuning and kernel parameter optimization modules are well established in DBN-EGWO-KELMwork. The experimental setup includes different benchmark datasets such as KDDCup'99, NSL-KDD, UNSW-NB15 and CICIDS2017 to measure the performance metrics of the existing and proposed methods. The overall processing time of every attack detection was also logged for every method individually during the experiments. Achievement of higher accuracy, precision, recall and F-score are the major advantage of this method, whereas higher processing time is realized as the limitation of DBN-EGWO-KELM work.

Network intrusion detection system on IoT networks is proposed in [15] using an anomaly detection scheme. This deep learning-based model uses CNN to create a binary and multiclass classification model and detect innovative cyber attacks launched by intruders at any time. This work also generates four new datasets, combines them as a single dataset and tries to identify more attacks. The result of the proposed work is validated using the benchmark datasets BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, IoT-23 and IoT-DS. The work's accuracy, precision, recall and F1-score measures are compared with the existing deep learning methods developed using CNN. The method outperformed all the other methods. The attack detection rate higher than 99% is the notable advantage of the work, whereas the large number of epochs required for the model convergence is the limitation of the proposed work.

By the above study on some existing CNN methods, the training time, the convergence time or the processing time is the major difficulty of the intrusion detection systems. Combined approaches with the most successful recurrent neural network (RNN) approach, LSTM, are now considered to reduce the training time. The notable feature of LSTM is the looping constraint on the hidden layer through which the output of the current state can be combined with the input for the next state by omitting some details. LSTM models are becoming the most efficient way to predict unexpected behaviours or patterns from given contexts.

The author in [16] proposed a model in 2020 for cyber attack detection to safeguard network nodes. While the traditional methods can identify only low-frequency attacks, the networks learned across multiple levels of temporal hierarchy over complex network traffic sequences and identify attacks through the proposed hierarchical LSTM (HLSTM) model. A multi-classification experiment on the NSL KDD dataset was performed to evaluate the HLSTM model. During the data cleaning stage, the useless features were identified and removed from the input. In the remaining 40 features, the seven symbolic features were converted into numerical features. Labels for all features were processed using the one-hot encoding method, and then the features were normalized. The detection performance for each attack category, DoS, probe, R2L, and U2R, was compared with all the mentioned existing methods. The notable strength of HLSTM is the easy detection of Dos and probe attacks, whereas the low-frequency attacks, U2R and R2L, become a complex task for the proposed model. Hence, the proposed model raised an open challenge for the researchers to build an efficient intrusion detection system, which should react to all kinds of data sets and for all kinds of attacks, even low-frequency attacks.

To reduce the high rates of false positive rate in predicting attack behaviour on Networks, a bidirectional LSTM (BiDLSTM) based system was proposed in [17]. This model concentrated on detecting all kinds of attacks, and additional care has been given to U2R and R2L attacks, which was a complex task in the other proposed models. The authors concentrated on Anomalies, the users' behaviours differed significantly from regular activity, using RNN techniques. Moreover, RNN is an exact method to brilliantly reuse the successive learning outcomes in the system for anomaly detection. The authors used the NSL KDD dataset to analyze the model's performance. Finally, the low-frequency attacks U2R and R2L were detected by BiDLSTM with high accuracy rate than the conventional LSTM methods. This model's notable feature is its better detection accuracy rate (7% more than conventional methods) for all the classes of attacks, such as prob, R2L, and U2R. Even though it has a higher prediction rate, the method requires more training time than the conventional methods. It is the limitation of the BiDLSTM model.

Stand-alone classifier models help to classify the attacks and normal behaviours in a network intrusion detection system. They all struggled with low false detection rates. When the irrelevant features from the dataset are removed, the model will work faster and achieve a better-improved accuracy rate. A two-phase combined approach is proposed in [18] on the NSL KDD dataset, which used a forward statistical search algorithm to build an optimal subset of features and a BiDirectional LSTM approach for the classifications of attacks. In the data preprocessing step, the Chi-square statistical model is used to fix the optimal subset of features based on Feature ranking and forward the best search approach.

A summary of existing methods, methodologies used, advantages, limitations and the datasets used are given in Table 1.

The selected feature set is evaluated using a bidirectional LSTM layer, one for forward and one for backward direction. The results are merged and fed as input to the following subsequent layers. The experiment results are analyzed with the existing methods and show a 4.26% performance improvement over other methods and reduced computational time. Although it has a notable feature in its performance, it has a high complexity on training time because of the new arriving attacks on the network system, which is a limitation of the two-phase approach, the combined statistical BiDirectional LSTM model. A more detailed study on network intrusion detection is needed to build an efficient model for all kinds of data sets and live networks. A summary of existing methods, methodologies used, advantages, limitations and the datasets used are given in Table 1.

3. Preliminary concept

Even though various models have been proposed recently using the LSTM approach, they are applicable with some preconditions like long term memory and large-size inputs with complex analysis. Thus, a model is required to predict the situations exactly from the training on past, present and future happenings and independent of whether the input size is large or small, the memory requirement is long or short and so on. Given the above fact, some standard models are needed to ensemble together and to be worked in parallel. The bidirectional long short-term memory (BiLSTM) and the bidirectional gated recurrent unit (BiGRU) are the base of this proposed work, and the fundamental concepts of these modules are described here to elucidate the proposed modules in a better way.

Table 1. Existing methods summary

Ref	Work	Methodology	Advantages	Limitations	Datasets
[10]	A Convolutional Neural Network for Improved Anomaly-Based Network Intrusion Detection	CNN	Moderate Accuracy	Training Time	NSL-KDD, UNSW-NB15
[11]	A Fast Deep Learning Method for Network Intrusion Detection Without Manual Feature Extraction	Multi-Packet input CNN	Accuracy	Processing Time	IDS 2018
[12]	Assessing Deep Neural Network and Shallow for Network Intrusion Detection Systems in Cyber Security	CNN	F1-Score	Processing Time	KDDCup
[13]	Bayesian Hyperparameter Optimization for Deep Neural Network-Based Network Intrusion Detection	Bayesian HyperParameter Optimization	Convergence time	Low Accuracy	NSL KDD
[14]	Deep Belief Network Integrating Improved Kernel-Based Extreme Learning Machine for Network Intrusion Detection	Kernel-Based Extreme Enhanced Grey Wolf Optimization – KELM-EGWO	Higher Accuracy Precision	Higher Processing Time	KDDCup 99, NSL KDD, UNSW-NB15 CICIDS 2017
[15]	Design and Development of a Deep Learning based model for Anomaly Detection in IoT Networks	CNN Based on binary and Multiclass classification	Higher Accuracy, Low false alarm rate	Convergence Time	BoT-IoT MQTT-IoT-IDS2022
[16]	Hierarchical long short-term memory network for cyber attack detection.	HLSTM	Easy detection of High Frequency Attacks	Detection of low frequency attacks	NSL-KDD
[17]	A Bidirectional LSTM Deep Learning Approach for Intrusion Detection	BiDLSTM with RNN	High accuracy rate	Training time	NSL KDD
[18]	χ2-bidLSTM: a Feature Driven Intrusion Detection System Based On χ2 Statistical Model And Bidirectional LSTM	Combined statistical BiDLSTM with Chi-square statistical model	Performance measures	Training time	NSL KDD

3.1 Bidirectional long short-term memory (BiLSTM)

Recurrent neural networks (RNN) are a kind of artificial neural network (ANN) which has a provision to maintain some essential knowledge persistently. Long short-term memory (LSTM) has several advantages for long-term dependency-based problems. LSTM is designed to provide abundant context-related information with more memory power than the other types of RNN [19]. The input gate, output gate and forget gate are the basic building blocks of an LSTM cell. A typical LSTM cell structure is illustrated in Fig. 1.

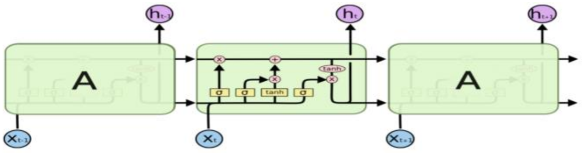

Figure. 1 LSTM cell structure

There are individual equations available for the operation of the Input gate, Output gate, and Forget gate, given below as Eqs. (1), (2) and (3), respectively.

$$i_t = \sigma(x_t * U_i + H_{t-1} * W_i) \quad (1)$$

$$o_t = \sigma(x_t * U_o + H_{t-1} * W_o) \quad (2)$$

$$f_t = \sigma\left(x_t * U_f + H_{t-1} * W_f\right) \tag{3}$$

Where, x_t is the current state input of timestamp t, U_i is the input weight matrix, H_{t-1} is the previous timestamp hidden state, W_i is the input weight matrix associated with the hidden state, U_o is the output weight matrix, W_o is the output weight matrix associated with the hidden state, U_f is the forget gate weight matrix, W_f is the forget gate weight matrix associated with the hidden state. The cell update equations for the input gate, output gate, and forget gates are given in Eqs. (4), (5) and (6) in order.

$$C_t = f_t * C_{t-1} + i_t * N_t \tag{4}$$

$$H_t = o_t * \tanh\,(C_t) \tag{5}$$

$$State\ update = \begin{cases} forget\ everything\ if\ f_t = 0 \\ forget\ nothing\ if\ f_t = 1 \\ update\ bias\ otherwise \end{cases} \tag{6}$$

Where N_t is the New information calculated using Eq. (7).

$$N_t = \tanh(x_t * U_c + H_{t-1} * W_c) \tag{7}$$

The output of the LSTM model is normalized using a softmax layer with a softmax function that operates with the H_t parameter as in the following Eq. (8).

$$\sigma(H_t) = \frac{e^{H_t}}{\sum_{j=1}^{k} e^{H_j}} \tag{8}$$

However, in many intrusion detection applications, the model requires the present and past experiences as two inputs to predict the exact happenings. BiLSTM is a derived model of LSTM in which two dedicated LSTM networks are integrated to operate with forward and backward directions. The sensitiveness property towards context makes the BiLSTM model provide better solutions for complex problems [20]. This hybrid deep learning model can produce more accurate results in detection and prediction. The BiLSTM architecture is illustrated in Fig. 2.

3.2 Bidirectional Gated Recurrent Unit (BiGRU)

A gated recurrent unit (GRU) is a recurrent neural network with a controlled accessing mechanism. Update gate (z), reset gate (r) and current memory gate $\left(\hat{h}_t\right)$ are the basic building blocks of a GRU. The equations for GRU vectors are given below.

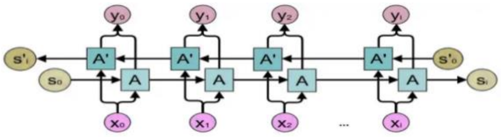

Figure. 2 BiLSTM architecture

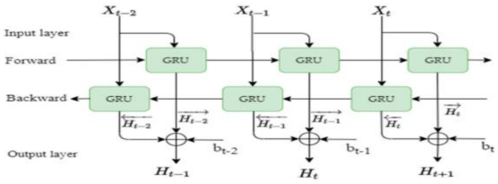

Figure. 3 BiGRU architecture

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \tag{9}$$

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \tag{10}$$

$$\hat{h}_t = \phi_h\left(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h\right) \tag{11}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \hat{h}_t \tag{12}$$

where x_t refers to the input vector, h_t is the output vector, z_t is the update gate vector, r_t is the reset gate vector, W, U are the parameter matrices, b is the parameter vector, and $\hat{h}_t$ refers to the candidate activation vector.

In general, GRU is configured with a sigmoid function $\left(\sigma_g\right)$ or a Hyperbolic tangent function (ϕ_h) for activation. The memory efficiency of GRU is somewhat better than the LSTM, and the performance speed is also comparably higher than LSTM models [21]. The BiGRU sequence processing model is an extension of GRU, in which two independent GRU units are appointed to take care of forward and backwards directions. The vanishing gradient problem is eliminated in the BiGRU model; thus, the convergence time is significantly reduced during the training phase [22]. The update and reset gates are used to decide the information priority for updating the output and to retain the data for further training. This way, BiGRU permits only relevant information to pass through, making better predictions. The BiGRU architecture is given in Fig 3.

4. Proposed parallel ABILSTM and CBIGRU ensemble IDS

The proposed model, "Parallel ABILSTM and

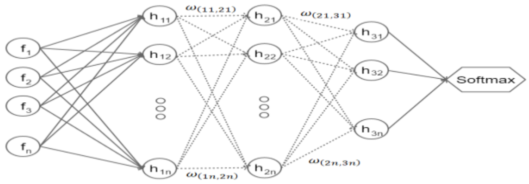

Figure. 4 IFFS-RNN architecture

CBIGRU Ensemble Network Intruder Detection System", is an ensemble model that stacks a novel impact based fuzzy feature selection (IFFS) method to operate with altered BiLSTM and customized BiGRU. The altered BiLSTM model is established from the standard BiLSTM architecture, and the regular BiGRU model is customized to serve the intruder detection purpose with betterment.

4.1 Impact-based Fuzzy Feature Selection (IFFS)

Usually, a typical network log record contains about 41 features such as duration, service, protocol_type, flag, src-bytes, dst-bytes, land, urgent, hot, count, srv_count, reerror_rate and so on. They identify the major attack classifications such as DoS, probe, U2R, R2L and unclassified attacks. Some features are more important than others based on the training mechanisms used in different intruder detection procedures. Since feature selection is the preliminary task of any machine learning method, a dedicated impact based fuzzy feature selection procedure is introduced in this work. The network transactions usually come under the timeseries based sequential streaming data. Due to the input type, a regular RNN is constructed to classify the attack types for the IFFS module. The RNN input layer can receive all available dataset features.Three hidden layers are used for the weight convergence. The output layer is a softmax that could classify the different types of attacks. The IFFS-RNN architecture is given in Fig. 4.

The number of nodes,f_1, f_2, f_3,…f_n in input layer depends on the dataset's number of features 'n'. To use different datasets, the IFFS-RNN model is designed to use variable nodes in input and hidden layers. The number of hidden layers is set to 3 as a constant. Each hidden layer have n nodes for each features, h_{1i}, h_{2i} and h_{3i} for i = 1 to n features. The weights $\omega_{(x,y)}$ of the hidden layer edges are maintained in a table to determine the influential features among other input features.

Let F be the set of input features$\{f_1, f_2 \cdots f_n\}$. The weights updates are calculated for a particular feature by training the network excluding the feature and then by training the network including that feature. The weight updates are logged into the weight update table (WUT), which calculates the impact index of a particular feature. The impact index is further fed to the fuzzy part for the feature elimination dispersal. The IFFS-RNN weight update calculation is performed using Algorithm 1.

Algorithm 1: IFFS-RNN weight update

```
Input: Input data Features, F
Output: Weight Update Difference Table
Let $n_t$ be the number of trial data
Let $\omega_p$ be the set of current weights $\omega_{(x,y)}$
Let $\omega_u$ be the set of updated weights
Let $\Delta = \{\delta_1, \delta_2 \ldots \delta_n\}$ be the set of difference in weights

Step 1: Initialize weights as
    $\forall i = 1 \rightarrow 3 :: \forall j = 1 \rightarrow i_n \coloneqq \omega_{(i,j)} = random(0,1)$
Step 2: Initialize $\Delta$ as $\forall i = 1 \rightarrow n \coloneqq \delta_i = 0$
Step 3: for k = 1 to $n_t$
    $\forall i = 1 \rightarrow n :: \forall j = 1 \rightarrow n$
    $\coloneqq Train\ Network\ with\ features\ \{f_i \rightarrow f_n \epsilon F, \sim f_i \in F\}$
    Compute $\Delta$ as
        $\forall i = 1 \rightarrow n \coloneqq \delta_i = \delta_i + |\omega_p - \omega_u|$
end for
Step 4: Compute $\Delta$ average as
        $\forall i = 1 \rightarrow n \coloneqq \delta_i = \frac{\delta_i}{nt}$
Step 5: return $\Delta$
```

The features are selected based on the weight update table Δ using Eq. (13).

$$\forall i = 1 \rightarrow n \coloneqq \begin{cases} Select\ if = \delta_i > \frac{1}{2} \\ Reject\ otherwise \end{cases} \tag{13}$$

The selected features are applied for furthermore proceedings of ABILSTM and CBIGRU.

4.2 Altered BiLSTM (ABILSTM)

The applications of BiLSTM are naturally gain the inherited advantages of a LSTM model. Two independent LSTM layers are used in the standard BiLSTM model, whereas, ABILSTM is created with an interlink between the forward and backward layer in the cell level. A bias-boost operation is introduced in ABILSTM cells to react based on the output of opposite direction layer mutually to enhance the true positive predictions in network intruder detection. An intruder attack is identified undeniably by enabling the cell level intercommunication between the LSTM layers. The ABILSTM cell architecture is given in Fig. 5.

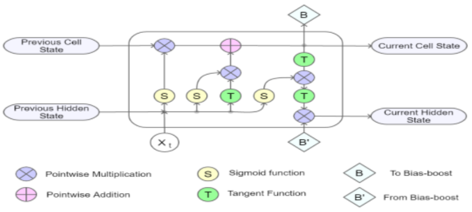

Figure. 5 ABILSTM cell architecture

The equations for input gate, forget gate, output gate, state update, and activation functions of ABILSTM are given below.

$$i_t = \sigma_g\left(W_i x_t + U_i h_{t-1} + B' b_i\right) \tag{14}$$

$$f_t = \sigma_g\left(W_f x_t + U_f h_{t-1} + B' b_f\right) \tag{15}$$

$$o_t = \sigma_g\left(W_o x_t + U_o h_{t-1} + B' b_o\right) \tag{16}$$

$$\tilde{c}_t = \sigma_g\left(W_c x_t + U_c h_{t-1} + B' b_c\right) \tag{17}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{18}$$

$$h_t = o_t \odot \sigma_h(c_t) \tag{19}$$

These equation variables are restricted to follow the basic rules such as, $x_t \in \mathbb{R}^{\eta}$, $i_t, f_t, o_t \in (0,1)^h$, $h_t, \tilde{c}_t \epsilon (-1,1)^h$, $c_t \in R^h$, where η refers the number of features selected by IFFS-RNN module. An array of η number of cascading ABILSTM units is formed to handle the input network transactional streaming data flow. By this way, ABILSTM can handle situations such as a new intruder trying to peek into the network and an existing compromised node changing its behavior to get super user access.

4.3 Customized BiGRU (CBIGRU)

Whenever the context of the input stream is changed beyond a threshold, the standard GRU cell will erase the historical data by triggering the reset gate. This reset option makes the GRU model to run in memory constrained applications. A novel restore reset option is introduced in the CBIGRU model to overcome the limitations. The restore reset option consumes more memory but can prevent the accuracy loss caused by adventitious resets. A Restore Reset Management Function (RRMF) is added to the typical GRU cell structure as in Fig. 6.

The RRMF is responsible for sustaining current convergence direction, update of new weight information and to restore the previous weight values based on the value of the decision factor Γ. The value of Γ is calculated by finding the backpropagation error using Eq. (20).

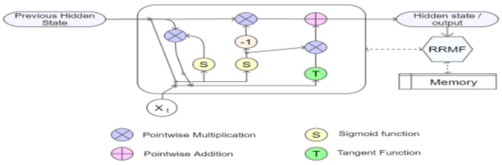

Figure. 6 CBIGRU cell structure

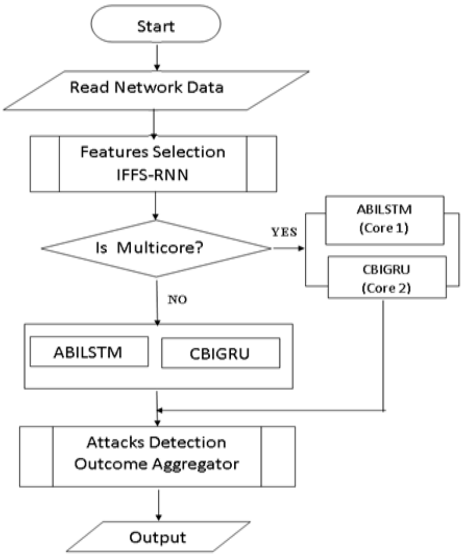

Figure. 7 PPOI flow diagram

$$\Gamma = \left[\varepsilon_p - \varepsilon_c\right] \tag{20}$$

Where ε_p is the previous propagation error value, ε_c is the current propagation error value. The restore reset decision is made by Eq. (21).

$$decision = \begin{cases} Reset\ if\ \Gamma = 1 \\ Restore\ if\ \Gamma = -1 \\ update\ otherwise \end{cases} \tag{21}$$

The CBIGRU cells are arranged as a cascading array as in regular BiGRU models. The selected features from the IFFS-RNN module are fed through the CBIGRU network for training and testing to engender an accurate intrusion detection method.

4.4 Parallel Processing Optimized Integration of ABILSTM and CBIGRU (PPOI)

It is natural to consume more processing time for the ensemble models due to added functionalities of more than one algorithm in sequence. PPOI module is designed to overcome this issue by enabling the concurrent execution of ABILSTM and CBIGRU whenever possible. Most of the computer architectures are utilizing multicore processors. PPOI takes advantage of the multicore architecture and can perform relatively faster processing than the other methods. The flowchart of PPOI method is given in Fig. 7.

Table 2. Dataset description

Category		Train Data	Test Data
Normal		67343	9711
Attack	Dos	11656	7458
	Probe	45927	2421
	R2L	995	2754
	U2R	52	200
Total		125973	22544

In this way, the integration of ABILSTM and CBIGRU is developed to diminish the disadvantages and facilitate the advantages of both models to operate swiftly with more accuracy.

5. Result and analysis

This section analyzes the performance of the proposed approach PACENIDS through experiments. The suggested approach is implemented using the Python programming language. The real-time data set NSL-KDD [23-25] is used to analyze the performance of the PACENIDS. An improved version of the KDD'99 dataset is the NSL-KDD dataset. It has 41 features and one class label. The label includes four categories of attack data (Dos, Probe, U2R, R2L) and normal data. The NSL-KDD dataset's training set (KDDTrain +) and test set (KDDTest +) are utilized as the model's training set and test set, respectively. A total of 125973 training records and 22544 testing records are there. Table 2 shows the description of the types and numbers of data sets.

5.1 Performance metrics

The performance in terms of accuracy, precision, recall, specificity, FAR and F-score is measured for the NSL KDD dataset. The measurements are logged for all DoS, probe, U2R R2L and unclassified attacks (UCA). The result averages are calculated for the different attack types and discussed the performance in this section.

Accuracy

Accuracy is the direct proportional parameter for any classification procedure. That is, the higher accuracy values represent the improved algorithm quality. It is calculated using the following Eq. (22)

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{22}$$

Where TP - true positive is the number of normal instances correctly classified as normal.

FP - false positive is the attack instances incorrectly classified as normal.

TN - true negative is the attack instances correctly classified as attack.

FN - false negative is the normal instances incorrectly classified as attacks.

Precision

Precision is an important parameter to evaluate a classification algorithm's performance. Precision is the measurement of the closeness between the results during an experiment. Precision is calculated using the following Eq. (23).

$$Precision = \frac{TP}{TP+FP} \tag{23}$$

Recall

Sensitivity or recall also called the true positive rate (TPR), is the probability of retrieving relevant information from the positive input. Sensitivity shows the reliability of an intrusion detection algorithm since higher sensitivity means higher detection of intrusions in a network environment. It will be calculated by the following Eq. (24).

$$Recall = \frac{TP}{TP+FN} \tag{24}$$

F-score

F-score or F-measure is the measurement index that refers to the harmony between sensitivity and precision. A higher F-score indicates a perfect balance between the sensitivity and the precision. The formula of F-Score is given below.

$$FScore = 2 \times \frac{Recall \times Precision}{Recall+Precision} \tag{25}$$

Specificity

Specificity refers the true negative rate (TNR) that is, the efficiency in finding the negative results among the given input data precisely. The higher specificity leads to less false alarm during a normal network operation. The specificity formula is

$$Specificity = \frac{TN}{TN+FP} \tag{26}$$

False alarm rate

False alarm rate refers the false positive rate (FPR) that is, the efficiency in finding the negative events wrongly categorized as positive among the given input data. The formula is

$$FAR = \frac{FP}{TN+FP} \tag{27}$$

Four kinds of experiments are conducted to assess the performance of the suggested approach. The first experiment involves binary classification,

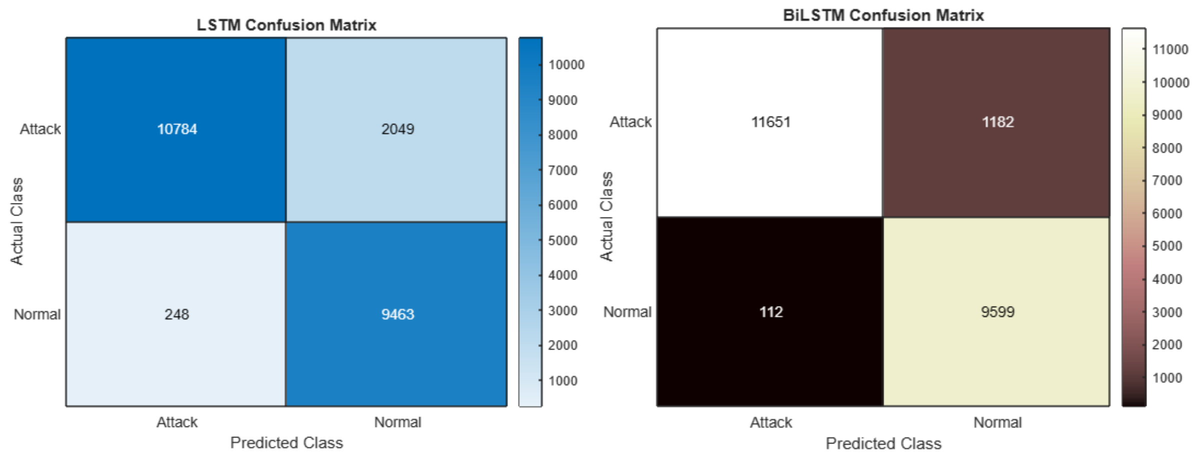

Figure. 8 Confusion matrices for LSTM and BiLSTM

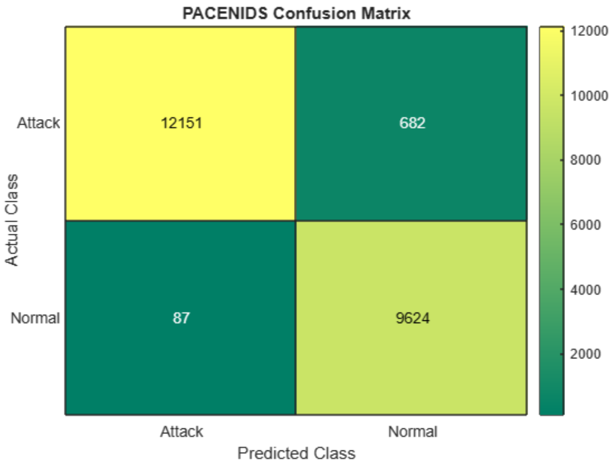

Figure. 9 Confusion matrix for PACENIDS

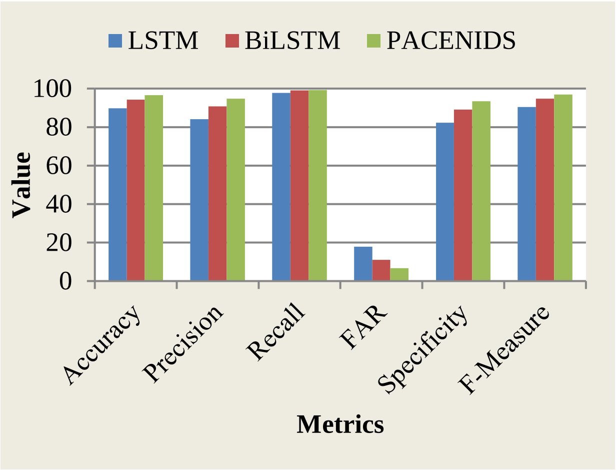

Figure. 10 Binary classification metrics comparison

Table 3. Performance metrics for binary classification

Metrics	Methods		
	LSTM	BiLSTM	PACENIDS
Accuracy	89.81	94.26	96.59
Precision	84.03	90.79	94.69
Recall	97.75	99.05	99.29
FAR	17.80	10.96	6.62
Specificity	82.20	89.04	93.38
F-Measure	90.38	94.74	96.93

the second involves multiple classes, and the third contains multiple classes with selected features. To compare the performance of the suggested approach, the standard LSTM and the bidirectional LSTM methods are used in each experiment initially. The final experiments with comparison of state-of-the-art methods.

5.2 Binary classification experiments

The LSTM, the bidirectional LSTM, and PACENIDS are used in this experiment to perform a binary classification (attack and normal) utilizing all 41 features from the NSL-KDD dataset. Fig. 8 shows the confusion matrices for traditional LSTM and the performance of the Bidirectional LSTM model, while Fig. 9 presents the proposed PACENIDS model. The performance outcomes for the traditional LSTM, BiLSTM, and suggested PACENIDS are summarized in Table 3. Based on the accuracy, precision, recall,FAR, specificity, and F-measure, the results demonstrate that the PACENIDS classifier efficiently identifies network attacks.

Fig. 10 depicts the binary classification performance metrics. The suggested PACENIDS method outperformed the existing methodology with the NSL-KDD dataset regarding binary classification accuracy. The suggested model PACENIDS achieved an accuracy of 96.59%, outperforming the performance of the other models. According to Table 3, the PACENIDS model increases the dataset's accuracy of the LSTM model by 7.27% and the BiLSTM model by 2.44%. In addition, it outperformed the other models regarding precision, recall, specificity, and f-measure. The projected PACENIDS also achieved a lower FAR of

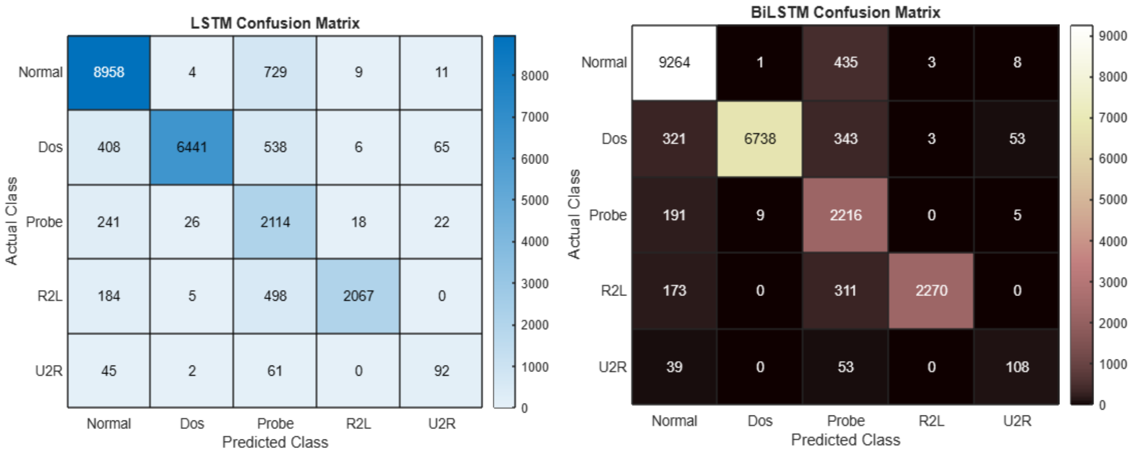

Figure. 11 LSTM and BiLSTM confusion matrices for multiclass classification

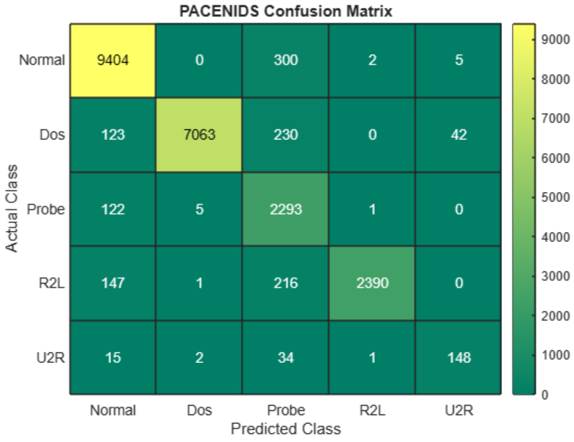

Figure. 12 PACENIDS confusion matrix for multiclass classification

Table 4. Performance metrics for multiclass classification

Method	Class	Performance Measure					
		Accuracy	Precision	Recall	FAR	Specificity	F-Measure
LSTM	**Normal**	92.77	91.07	92.25	6.84	93.16	91.66
	DoS	95.32	99.43	86.36	0.25	99.75	92.44
	Probe	90.54	53.65	87.32	9.07	90.93	66.47
	R2L	96.81	98.43	75.05	0.17	99.83	85.17
	U2R	99.09	48.42	46.0	0.44	99.56	47.18
BiLSTM	**Normal**	94.81	92.75	95.4	5.64	94.36	94.06
	DoS	96.76	99.85	90.35	0.07	99.93	94.86
	Probe	94.03	65.99	91.53	5.68	94.32	76.69
	R2L	97.83	99.74	82.43	0.03	99.97	90.26
	U2R	99.3	62.07	54.0	0.3	99.7	57.75
PACENIDS	**Normal**	96.83	95.85	96.84	3.17	96.83	96.34
	DoS	98.21	99.89	94.7	0.05	99.95	97.23
	Probe	95.97	74.62	94.71	3.88	96.12	83.47
	R2L	98.37	99.83	86.78	0.02	99.98	92.85
	U2R	99.56	75.9	74.0	0.21	99.79	74.94

6.62%. PACENIDS reduces the FAR for the LSTM and BiLSTM models by 62.80% and 39.59%, respectively.

5.3 Multiclass classification experiments

This section examines the effectiveness of the traditional LSTM, BiDLSTM model, and PACENIDS for mutli-class classification using NSL-KDD dataset and a 5-class (normal, DoS, probe, R2L, and U2R) classifier trained on all 41 features. The confusion matrices used to assess the traditional LSTM, BiLSTM, and PACENIDS are

Table 5. Selected features set

Method	Feature Id	Number of Features
LSTM [18]	2, 3, 4, 5, 6, 8, 10, 13, 14, 22, 24, 25, 27, 28, 29, 31, 33, 34, 38, 40, 41	21
BiLSTM [18]	2, 3, 4, 5, 6, 8, 10, 13, 14, 22, 24, 25, 27, 28, 29, 31, 33	17
PACENIDS	2, 3, 4, 5, 6, 8, 10, 13, 14, 22, 24, 25, 27, 28, 31	15

shown in Figs. 11 and 12. Table 4 shows the performance metrics for multiclass classification.

According to the results, the PACENIDS model increases the dataset's accuracy of the LSTM model by 8.26% and the BiLSTM model by 3.40%. PACENIDS reduces the FAR for the LSTM and BiLSTM models, respectively, by 89.57%, and 52.27%.

5.4 Multiclass classification with selected features experiments

This section examines the effectiveness of the traditional LSTM, BiDLSTM model, and PACENIDS for multiclass classification using the NSL-KDD dataset and a 5-class (Normal, DoS, Probe, R2L, and U2R) classifier trained on selected features. The suggested work is compared with chi-square based feature selection method [18]. The standard LSTM and BidLSTM with selected features are taken from the method [18]. Table 5 shows the selected features.

The confusion matrices used to assess the traditional LSTM, BiLSTM, and PACENIDS with selected features are shown in Figs. 13 and 14 and Table 6 shows the performance metrics for multiclass classification with selected features.

A comparative analysis of accuracy and FAR is made for LSTM, BiLSTM and the proposed approach PACENIDS with all the features in NSL KDD dataset and the selected features of the NSL KDD dataset using a different feature selection based approach. The suggested approach achieves 94.47% accuracy and lower 0.42% FAR for all the features and achieves a higher accuracy (97.67%) and lower FAR (0.027%) with selected features, when compared to other two approaches. Table 7, Figs. 15, and 16 show the comparison analysis of the accuracy and FAR.

The above figures shown that, the PACENIDS model increases the dataset's accuracy of the LSTM model by 7.14% and the BiLSTM model by 2.14%. PACENIDS also reduces the FAR for the LSTM and BiLSTM models. Therefore, the overall performance in attack detection and classification of PACENIDS method is recorded in the experiment. At the same time, it is realized that the proposed method consumes lesser average processing times (in millisecond (mS) unit) during the experiments carried out.

5.5 Comparison with state-of-the-art methods

This section analyzes the performance of the proposed PACENIDS with various state-of-the-art methods. The following methods are used for comparison: MCNN [10], MCNN-DFS [10], BO-GP [13], HLSTM-IDS [16]. Table 8 shows the performance metrics for different methods.

From Table 8, the proposed PACENIDS_IFFS method achieves high accuracy compared to other methods.

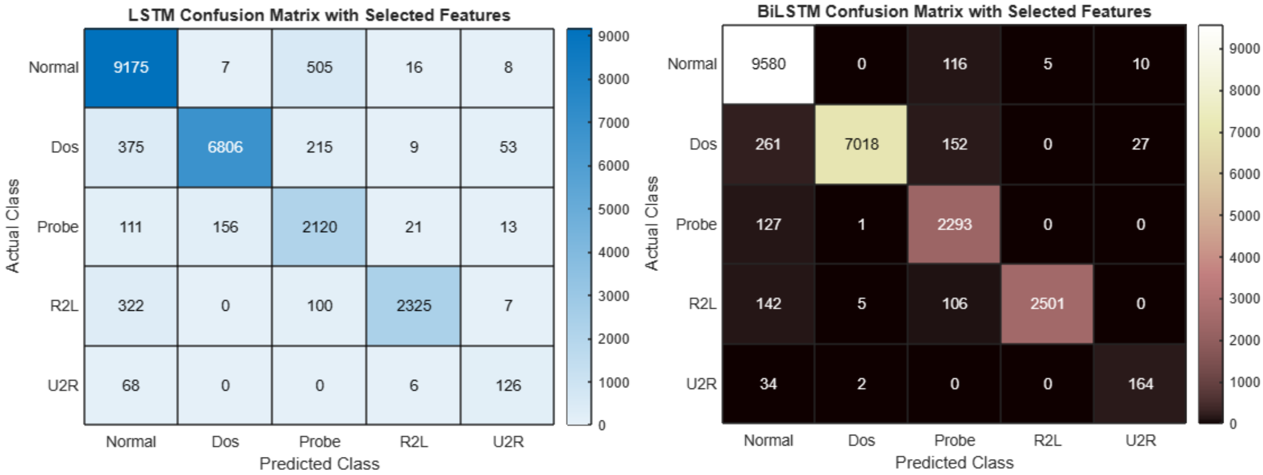

Figure. 13 Confusion matrices for LSTM and BiLSTM with selected features

Received: July 4, 2023. Revised: September 13, 2023.

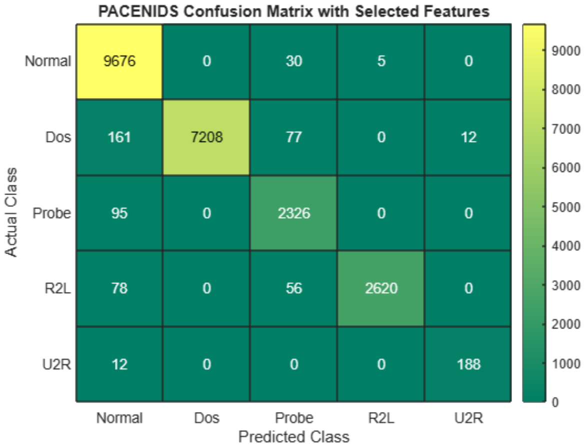

Figure. 14 Confusion matrix for PACENIDS with selected features

Table 6. Performance metrics using selected features

Method	Class	Performance Measure					
		Accuracy	Precision	Recall	FAR	Specificity	F-Measure
LSTM [18]	**Normal**	93.74	91.28	94.48	6.83	93.17	92.85
	DoS	96.38	97.66	91.26	1.08	98.92	94.35
	Probe	95.03	72.11	87.57	4.07	95.93	79.09
	R2L	97.87	97.81	84.42	0.26	99.74	90.63
	U2R	99.31	60.87	63.0	0.36	99.64	61.92
BiLSTM [18]	**Normal**	96.92	94.44	98.65	4.39	95.61	96.5
	DoS	98.01	99.89	94.1	0.05	99.95	96.91
	Probe	97.77	85.98	94.71	1.86	98.14	90.13
	R2L	98.86	99.8	90.81	0.03	99.97	95.1
	U2R	99.68	81.59	82.0	0.17	99.83	81.8
PACENIDS	**Normal**	98.31	96.55	99.64	2.7	97.3	98.07
	DoS	98.89	100.0	96.65	0.0	100.0	98.3
	Probe	98.86	93.45	96.08	0.81	99.19	94.75
	R2L	99.38	99.81	95.13	0.03	99.97	97.42
	U2R	99.89	94.0	94.0	0.05	99.95	94.0

Table 7. Accuracy and FAR comparison

Approach	Without Features Selection		With Features Selection		
	Accuracy	FAR	Feature Selection Method	Accuracy	FAR
LSTM	87.26	4.03	Chi-Square	91.16	3.77
BiLSTM	91.36	3.06	Chi-Square	95.62	2.11
PACENIDS	94.47	0.42	IFFS	97.67	0.027

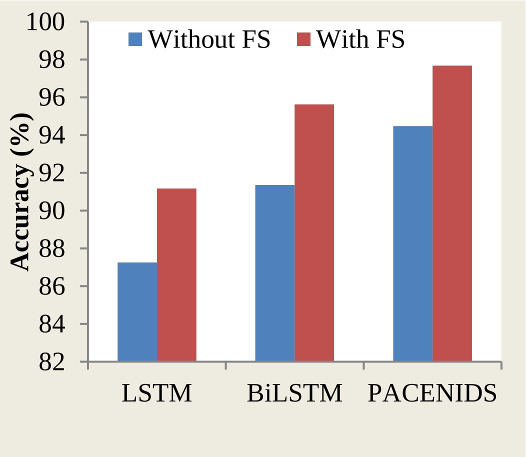

Figure. 15 Accuracy comparison

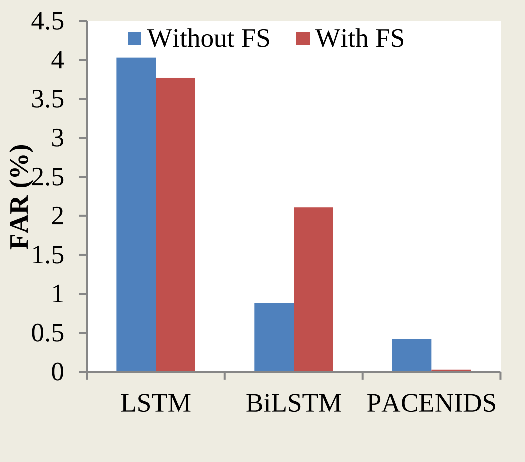

Figure. 16 FAR comparison

Table 8. Performance metrics for different methods

Method	Accuracy	Precision	Recall	F-measure
MCNN[10]	81.1%	83%	81%	80%
MCNN-DFS[10]	81.44%	81%	84%	80%
BO-GP[13]	82.95	79.73	81.35	80.4
HLSTM-IDS[16]	83.85	77.94	78.96	78.45
PACENIDS	94.47	89.41	89.22	89.31
PACENIDS_IFFS	97.67	96.30	96.76	96.53

6. Conclusion and future work

This work is implemented to determine the ability of an ensemble of two deep learning algorithms towards anomaly detection in smart city communication networks. Based on the experiments carried out in simulation and real-time implementation, the integration of proposed functional modules evidently produced better performance in network intrusion detection. The detection ability of the proposed PACENIDS work is surpassed for all major attack categories such as DoS Probe U2R and R2L. The parallel executable nature of the proposed method ensures the faster execution of the ensemble methods. The evaluation results show that there is only a very inconsiderable change in the network intrusion detection performance. Proposed PACENIDS work can be recommended for smart city communication network security against different types of intrusions following the produced evaluation results. Effectuation of bio-inspired optimization algorithms in the preprocessing stage and including hyperparameter fine-tuning to improve the processing speed and intrusion detection accuracy can be the possible future works originated from this work.

Conflicts of interest

The authors certify that they have NO affiliations with or involvement in any organization or entity with any financial or non-financial interest in the subject matter or materials discussed in this manuscript.

Author contributions

Conceptualization- N.G. , Methodology – N.G., T.N.R., Experiments – N.G., Validation –T.N.R., Analysis, Writing- Original draft preparation- N.G., Writing- Review and editing– T.N.R.

References

[1] M. Oberascher, W. Rauch, and R. Sitzenfrei, "Towards a smart water city: A comprehensive review of applications, data requirements, and communication technologies for integrated management", *Sustainable Cities and Society*, Vol. 76, 2022.

[2] D. Cohen, A. Elalouf, and R. Zeev, "Collaboration or separation maximizing the partnership between a "Gray hat" hacker and an organization in a two-stage cybersecurity game", *International Journal of Information Management Data Insights*, Vol. 2, No. 1, 2022.

[3] S. Krishnaveni, S. Sivamohan, S. S. Sridhar, and S. Prabakaran. "Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing", *Cluster Computing*, Vol. 24, No. 3, pp. 1761–1779, 2021.

[4] N. R. Sai, J. Bhargav, M. Aneesh, G. V. Sahit, and A. Nikhil, "Discovering Network Intrusion using Machine Learning and Data Analytics Approach", In: *Proc. of 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pp. 118-123, 2021.

[5] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges", *Cybersecurity*, Vol. 4, No. 18, pp. 1-27, 2021.

[6] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa, and C. F. M. Foozy, "Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset", *IEEE Access*, Vol. 9, pp. 22351-22370, 2021.

[7] M. P. Ramkumar, T. Daniya, P. M. Paul, and S. Rajakumar, "Intrusion detection using optimized ensemble classification in fog computing paradigm", *Knowledge-Based Systems*, Vol. 252, 2022.

[8] B. A. Tama and S. Lim, "Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation", *Computer Science Review*, Vol. 39, 2021.

[9] Y. N. Kunang, S. Nurmaini, D. Stiawan, and B. Y. Suprapto, "Attack classification of an intrusion detection system using deep learning and hyperparameter optimization", *Journal of Information Security and Applications*, Vol. 58, 2021.

[10] I. A. Turaiki and N. Altwaijry, "A convolutional neural network for improved anomaly-based network intrusion detection", *Big Data*, Vol. 9, No. 3, pp. 233-252, 2021

[11] W. Yue, J. Yiming, and L. Julong, "A fast deep learning method for network intrusion detection without manual feature extraction", *Journal of Physics: Conference Series*, Vol. 1738, No. 1, 2021.

[12] D. B. Mandru, M. A. Safali, N. R. Sai, and S. C. G. Kumar, “Assessing deep neural network and shallow for network intrusion detection systems in cyber security”, In: *Computer Networks and Inventive Communication Technologies: Proceedings of Fourth ICCNCT*, Springer Singapore, pp. 703-713, 2021.

[13] M. Masum, H. Shahriar, H. Haddad, M. J. H. Faruk, M. Valero, M. A. Khan, M. A. Rahman, M. I. Adnan, A. Cuzzocrea, and F. Wu, "Bayesian hyperparameter optimization for deep neural network-based network intrusion detection", In: *Proc. of 2021 IEEE International Conference on Big Data (Big Data)*, pp. 5413-5419, 2021.

[14] Z. Wang, Y. Zeng, Y. Liu, and D. Li, "Deep Belief Network Integrating Improved Kernel-Based Extreme Learning Machine for Network Intrusion Detection", *IEEE Access*, Vol. 9, pp. 16062-16091, 2021.

[15] I. Ullah and Q. H. Mahmoud, "Design and development of a deep learning-based model for anomaly detection in IoT networks", *IEEE Access*, Vol. 9, pp. 103906-103926, 2021.

[16] H. Hou, Y. Xu, M. Chen, Z. Liu, W. Guo, M. Gao, Y. Xin, and L. Cui, "Hierarchical long short-term memory network for cyberattack detection", *IEEE Access*, Vol. 8, pp. 90907-90913, 2020.

[17] Y. Imrana, Y. Xiang, L. Ali, and Z. A. Rauf, "A Bidirectional LSTM Deep Learning Approach for Intrusion Detection", *Expert Systems with Applications*, Vol. 185, p. 115524, 2021.

[18] Y. Imrana, Y. Xiang, L. Ali, Z. A. Rauf, Y. C. Hu, S. Kadry, and S. Lim. "χ2-bidLSTM: a Feature Driven Intrusion Detection System Based On χ2 Statistical Model And Bidirectional LSTM ", *Sensors*, Vol. 22, No. 5, 2022.

[19] Z. K. Abbas and A. A. A. Ani, "Detection of Anomalous Events Based on deep Learning-BiLSTM", *Iraqi Journal of Information and Communication Technology*, Vol. 5, No. 3, pp. 34–42, 2022.

[20] R. A. Saleh, M. Driss, and I. Almomani, "CBiLSTM: A Hybrid Deep Learning Model for Efficient Reputation Assessment of Cloud Services", *IEEE Access*, Vol. 10, pp. 35321-35335, 2022.

[21] Y. Duan, H. Li, M. He, and D. Zhao, "A BiGRU Autoencoder Remaining Useful Life Prediction Scheme With Attention Mechanism and Skip Connection", *IEEE Sensors Journal*, Vol. 21, No. 9, pp. 10905-10914, 2021.

[22] F. Teng, Y. Song, and X. Guo, "Attention-TCN-BiGRU: An Air Target Combat Intention Recognition Model", *Mathematics*, Vol. 9, 2021.

[23] M. A. Almaiah, O. Almomani, A. Alsaaidah, S. A. Otaibi, N. B. Hani, A. K. A. Hwaitat, A. A. Zahrani, A. Lutfi, A. B. Awad, and T. H. H. Aldhyani, "Performance Investigation of Principal Component Analysis for Intrusion Detection System Using Different Support Vector Machine Kernels", *Electronics*, Vol. 11, No. 21, 2022.

[24] https://www.unb.ca/cic/datasets/nsl.html

[25] https://www.kaggle.com/datasets/hassan06/nslkdd