International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.6987
***Article Received**: 04 April 2023 **Revised**: 20 May 2023 **Accepted**: 31 May 2023*

An Effective Dual Level Flow Optimized AlexNet-BiGRU Model for Intrusion Detection in Cloud Computing

***[1]Rajesh Bingu, [2]S. Jothilakshmi**
[1]Research Scholar, Department of Information Technology,
Annamalai University, Chidambaram, Tamil Nadu 608002-India.
Email: govindajeeyardasan@gmail.com
[2]Associate Professor, Department of Information Technology,
Annamalai University, Chidambaram, Tamil Nadu 608002-India.
Email: jothi.sekar@gmail.com

Abstract: In recent years, several existing techniques have been developed to solve security issues in cloud systems. The proposed study intends to develop an effective deep-learning mechanism for detecting network intrusions. The proposed study involves three stages pre-processing, feature selection and classification. Initially, the available noises in the input data are eliminated by pre-processing via data cleaning, discretization and normalization. The large feature dimensionality of pre-processed data is reduced by selecting optimal features using the wild horse optimization-based feature selection (WHO-FS) model. The selected features are then input into a proposed dual-level flow optimized AlexNet-BiGRU detection model (DLFAB-IDS). Whereas the flow direction algorithm (FDA) approach optimally tunes the hyperparameters and helps to enhance the classification performance. In the proposed model, the intrusions are detected by AlexNet and the multiclass classification is performed through the BiGRU method. The proposed study used the NSL-KDD dataset, and the simulation was done by Python tool. The efficacy of a proposed model is measured by evaluating several performance metrics. The comparison over other existing techniques shows that the proposed model brings higher performance in terms of accuracy 96.81%, recall 95.84%, precision 96.24%, f1-score 96.75%, prediction time 0.43s and training time 152.84s.

Keywords*:* Intrusion detection system, pre-processing, feature selection, wild horse optimization, dual level flow optimized AlexNet-BiGRU detection model, classification, NSL-KDD.

I. INTRODUCTION

In recent years, the Internet has increased cyber-attacks because of frequent technological advancements [1]. Due to the higher amount of information and data, the Internet faces several problems making it a protective and stable system [2]. Various attacks are presented on commercial games and personal interests in the IoT and Internet environment [3]. Nowadays, the attack techniques and the number of attacks have rapidly enhanced. This leads to critical risks to stable and safe access to IoT and Internet [4]. Hence, developing an effective tool to detect several security threats and prevent such attacks is highly required. An intrusion detection system (IDS) is a security tool to avoid illegal external attacks and connections [5]. The IDS is also utilized for classification issues, where the system can detect whether the available network traffic scenario is normal or abnormal. The major intention of this IDS is to enhance classification accuracy by efficiently detecting intrusion behaviours. IDS can afford Internet users integrity, confidentiality and effortless usability [6].

In the past few decades, the IDS has mainly been used to detect network traffic and analyze threats or malicious behaviours in a realistic world [7]. Like the firewall-based security system, an IDS is also employed for security purposes. A system with higher accuracy, reduced overhead, and improved performance is considered a superior IDS [8]. The two types of attack detection techniques in IDS are misuse or signature-based detection and anomaly-based detection [9]. The network-based IDS can detect unauthorized, anomalous and illicit activities over the Internet. Establishing computational techniques to detect numerous cyber-attacks requires determining varied incident patterns and similarly detecting the threats through cyber security data [10]. It is termed a data-driven intelligent IDS. To generate a data-driven IDS, knowledge about learning methodologies is important. In machine learning algorithms, the detection of cyber-attacks is highly problematic because various detections of several classifiers result in varied contexts based on the aspects of data [11].

To detect several kinds of attacks, many of the existing studies employed machine learning algorithms [12]. The machine learning algorithms allow the network administrator to take preventive measures for intrusion activities [13]. Although, the machine learning algorithm cannot resolve the classification issue in numerous intrusion data. Several detection tasks attained reduced classification accuracy due to the high dimensionality of datasets. Recently, deep learning techniques gained higher attention owing to extracting optimal representations from the input data. Also, deep learning techniques provide better outcomes than machine learning algorithms. Some of the existing techniques widely utilized for detecting and classifying intrusions in the network are RNN (Recurrent Neural network) [14], CNN (Convolutional Neural Network) [15], ANN (Artificial Neural Network), SVM (Support Vector Machine) [16], RF (Random Forest) [17], KNN (K-Nearest Neighbour) [18], DBN (Deep Belief Network) [19], DBM (Deep Boltzmann Machines), Autoencoder [20] and so on.

Motivation

IDS is considered the major concept of cyber security measures in each organization. To ensure information security, IDS is more important because it can easily analyze several attacks available in the network. The cyber-attacks cause several security problems for society and lead to financial losses. The increased computational resources make detecting intrusions from the network difficult. Thus, an enhanced performance IDS can be more powerful in limiting cyber-attacks and malicious activities. Recently, the existing studies developed varied detection methodologies for enhancing data security. Among that, Artificial Intelligence (AI) based deep learning methods are more effective in detecting and classifying intrusions from the network. Deep learning ability is high and helps the system attain higher detection and classification results. Moreover, deep learning techniques automatically extract the deep features supported for detecting intrusions. Thus, it motivates the author to develop significant deep-learning techniques for intrusion detection in a cloud environment. The major contributions of the proposed study are outlined as,

- ❖ To propose an effective IDS through a hybrid deep learning mechanism named Dual Level Flow optimized AlexNet-BiGRU detection model (DLFAB-IDS).
- ❖ To minimize the unwanted data presented in the provided dataset, efficient pre-processing is performed.
- ❖ To reduce the system's computational complexity, the Wild Horse Optimization (WHO) algorithm enables the optimal selection of feature sets.
- ❖ Finally, with the proposed DLFAB-IDS, the intrusions are detected and classified.
- ❖ To prove the efficacy of proposed model, comparison analysis is performed over varied existing techniques.

The remaining part of this research paper is organized as follows: Section 2 provides the related works recently carried out by several researchers from the intrusion detection domain. Section 3 briefly explains the methodologies utilized in the proposed IDS, and Section 4 deals with the simulation results and discussion of the proposed system. Section 5 presents this paper's overall conclusion and highlights the future scope.

II. RELATED WORKS

The recent methods utilized for detecting intrusions in some of the existing studies are given as,

Ghosh et al. [21] developed an IDS in the cloud domain through hybrid optimization algorithms. This existing study combines Cuckoo Search (CS) algorithm and Particle Swarm Optimization (PSO) algorithm to classify the attacks. The NSL-KDD dataset demonstrates the efficiency of developed IDS. In this, pre-processing and normalization are performed initially, and then the redundant features in the data are performing the feature selection process. Finally, the provided input data is classified as normal or anomalous in the developed hybrid algorithms. The reduced convergence speed affects the classification performance, which is considered the major drawback of this existing study.

Yin et al. [22] developed a deep learning method for identifying intrusions from the network system. This existing study established an RNN-based IDS mechanism in which binary and multiclass classification is performed. Initially, the input data are pre-processed by performing numericalization and normalization. Here, the performance of a developed model is promoted by the number of neurons and varied learning rates in the neural network. The simulation results show that the developed RNN method is highly suitable for an effective IDS compared to other machine learning mechanisms such as RF, ANN and SVM.

Agarwal et al. [23] introduced a machine-learning model for classifying the varied intrusions available in the network system. This existing study uses three classification models: SVM, NB and KNN. This study adopts the UNSW-NB15 database for simulation, and three machine-learning approaches are utilized to detect the intrusions. The three approaches are executed, and comparing the outcomes of each model, the optimal approach is chosen to identify the attacks in the network system. Thus, this existing study intends to determine the efficient classifier highly suitable for categorizing intrusions in the network. The experimental analysis states that the developed model gains higher classification accuracy than others.

Su et al. [24] presented an optimized attention-based deep learning model for network traffic detection. The BAT algorithm is hybridized with the developed Bi-LSTM (Bi-directional Long Short-term memory) model. The developed system combines the BAT algorithm with an attention-based Bi-LSTM model to resolve the minimal accuracy issue. The attention function is employed to identify the network flow vector generated by the packet vectors in the developed Bi-LSTM approach. Several convolutional layers of the deep learning approach process the input data. Finally, the softmax layer is utilized for classifying the network traffic in the system. The analysis shows that the developed system is better than conventional methods. However, the classifier enhances the loss function because the BAT optimization easily falls into local optima.

Zhong et al. [25] developed a deep learning model for assisting IoT servers by detecting intrusions in a network system. This existing study established a sequential model to gather useful features from the network and application layers. Through tcpdump packets and system routines, the developed model gathers the needed features from both layers. This existing work used CNN and GRU approaches for intrusion detection systems and provides higher security to IoT users. The developed system detects the available intrusions in the network. However, the data integrity is not as effective. Table 1 mentions the comparison analysis of existing studies

Table 1: Existing studies and their limitations

Author	Methods	Purpose	Parameters	Dataset	Limitations
Ghosh et al. [21]	CS-PSO	➢ To detect intrusions and classify the type of attacks through hybrid optimization.	Accuracy	➢ NSL-KDD	➢ Reduced convergence rate and generated lower accuracy.
Yin et al. [22]	RNN	➢ Detecting intrusions using a deep learning model for attaining higher detection accuracy.	Accuracy, true positive rate (TPR), false positive rate (FPR)	➢ NSL-KDD	➢ Required more training time. ➢ Classification performance is affected.
Agarwal et al. [23]	NB, SVM and KNN	➢ To detect the classification accuracy of three varied methods while classifying attacks.	Accuracy rate	➢ UNSW-NB15	➢ Minimized prediction capability. ➢ Accuracy is not as much better.
Su et al. [24]	BLSTM, BAT	➢ To propose a traffic anomaly detection using the deep learning method.	Accuracy, FPR and detection rate	NSL-KDD	➢ Training time is increased due to more weight parameters
Zhong et al. [25]	GRU	➢ Proposing IDS for IoT servers through deep learning models	Precision, recall, F1-score	KDD99, ADFA-LD	➢ Data integrity is affected

Problem statement

During recent decades, the intrusions in a network system have enhanced rapidly due to the presence of sophisticated tools for initiating such anomalies. Intrusions in the cloud data centres heavily affect the reputation of cloud computing technology, thereby restricting the users from choosing the services offered by the cloud. One of the main reasons behind the complexity of resolving the intrusions in the cloud is its heterogeneous nature and the existence of virtualized environments. Several existing works attempt to develop a promising technology for avoiding network intrusions. However, they failed to detect the intrusions and had trouble classifying the attacks. This is due to higher system complexity, misclassification issues, higher data dimensionality, cost-effectiveness etc. In previous works, machine learning algorithms have been utilized to detect network intrusions in cloud environments. Affording an enhanced performance IDS is necessary for preventing several attacks. But, the existing machine learning techniques have reduced the ability to detect and classify the attacks, affecting data security. Many deep learning-based techniques are introduced in the literature to solve the intrusion detection and classification problem, but those techniques cannot provide a high accuracy rate. Also, the false positive alarms of those models are very high and unacceptable. The research community strives to find a new and optimal strategy to detect and efficiently classify intrusions in cloud data centres. A novel hybrid deep learning methodology is introduced in this work to fill the identified research gaps and to solve the problems of intrusions in the cloud reliably.

III. PROPOSED METHODOLOGY

A novel hybrid intrusion detection system is proposed in this work based on deep learning to attain higher accuracy in detection. The proposed model involves 3 main stages: pre-processing, feature selection and classification. In the initial pre-processing stage, data cleaning, data transformation and data normalization are performed to remove the unwanted data presented in the dataset, enhancing the data quality. In the second stage, the major features in the dataset are selected using the proposed wild horse optimization-based feature selection (WHO-FS) model. In the last stage, a hybrid neural network model is proposed named dual level flow optimized AlexNet-BiGRU detection model (DLFAB-IDS) to accurately detect and classify the type of intrusion detected in the host. Whereas the hyperparameters in the network model are tuned using the flow direction algorithm (FDA). This helps to reduce the overall training error, thereby increasing the classification accuracy. Figure 1 depicts the workflow of a proposed methodology.

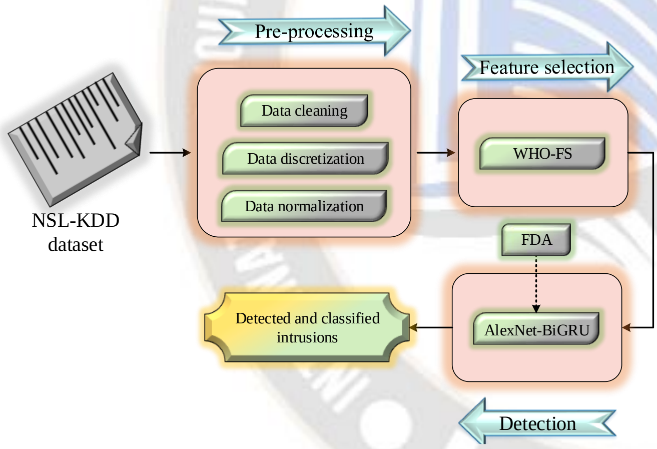


Figure 1: Block diagram of a proposed methodology

A. *Pre-processing*

Pre-processing is an initial stage performed in the proposed IDS framework. In this stage, input data quality is enhanced by removing unwanted data in the dataset. The proposed study used three varied data mining techniques for pre-processing the input data: data cleaning, data transformation and data normalization.

- *Data cleaning:* Generally, the raw input data is unsuitable to directly provide input of the deep learning model because the input data involves duplicate columns, several missing values, null values, erroneous labels and infinity values. Thus, data cleaning is necessary to enhance the performance of a proposed classification stage by removing unwanted data. By performing data cleaning, the noisy data is removed, inconsistencies are solved, and the outliers are detected and eliminated. The data cleaning process also determines errors and then updates, alternates or eliminates data to generate better performance.
- *Data discretization:* This process converts a large set of data into smaller hence it helps to handle the data as easier. In simple words, data discretization translates the characteristics of continuous data values into a finite series interval by reducing the information loss in the provided input data.
- *Data normalization:* The input data are normalized using the min-max normalization method. The normalization process can enhance the training speed because each input data utilized in the training process involves similar scales in the range of 0 to 1. It is represented as,

$$F_{norm} = \frac{F - F_{\min}}{F_{\max} - F_{\min}} \tag{1}$$

Where, F_{norm} mentions the normalization results, the initial value before the normalization process is mentioned as F, a maximum value of feature is specified as $F_{\max}$ and the minimum value of feature is indicated as $F_{\min}$ correspondingly. This normalization stage assists in limiting the noises in the provided input data.

B. *Optimized feature selection*

Feature selection reduces the number of input variables to enhance the proposed classification method's efficiency. In this process, the large feature dimensionality is minimized into smaller feature sets, which helps the classifier make accurate results. Optimally selecting the needed features from the pre-processed data reduces computational complexity and supports the classification stage to obtain higher accuracy. In the proposed study, the optimal feature selection is made using the WHO approach. The proposed feature selection approach is one of the meta-heuristic algorithms inspired by wild horses' social life strategies. The proposed WHO-FS model selects the optimal features based on the specific fitness function. The computation of a fitness function is given as,

$$Fitness\ value = \delta * e + (1-\mu) * \frac{|N_a|}{|N_b|} \tag{2}$$

Where, δ represents the parameter that influences the result of intrusion detection and classification stage, e specifies the presented error value in classification, N_a mentions the total features attained from pre-processed data and M_b represents

***Article Received**: 04 April 2023 **Revised**: 20 May 2023 **Accepted**: 31 May 2023*

the total features presented in the provided dataset. The proposed WHO-FS model contains five varied essential stages that are followed as,

- ✓ Generating an initial population. Then, create a feature group and choose the optimal search agent.
- ✓ Initiate the search process through search agents.
- ✓ Lead the feature group using the optimal search agent.
- ✓ Exchange and choose an appropriate search agent for selecting useful features.
- ✓ Save the optimal solution.

The search agent selects the essential features from the pre-processed input data based on the stages mentioned above.

1. Generating initial population

The population of input features are initialized in this stage. Let's consider, the initial random population as $\left(\vec{Z}\right)=\left\{\vec{Z}_1, \vec{Z}_2, \ldots., \vec{Z}_n\right\}$. The target function continuously computes the random population of features, and the target value is specified as $\left(\vec{T}\right)=\left\{T_1, T_2, \ldots., T_m\right\}$. At first, the initial feature population is partitioned into varied groups. If the amount of features in the population is M, then the amount of groups is given as, $G=\lceil M \times S_P \rceil$, where S_P mentions the percentage of search agents. In the initial stage of a WHO-FS approach, one of the optimal search agents is chosen as the leader. Based on the guidance of an optimal search agent, the remaining search agents explore the essential features.

2. Grazing strategy

To execute the grazing strategy of horses in WHO, consider the leader as a centre of an exploring region. In the proposed study, the optimal search agent is assumed as a leader, and the leader is placed at the centre of a feature exploring space. Thus, the other search agents are moved towards the leader search agent in the grazing area and are expressed as,

$$\vec{Z}^{\,j}_{i,G} = 2\alpha\cos(2\pi r\alpha)\times\left(leader\ searchagent^{j} - Z^{j}_{i,G}\right)+leader\ searchagent^{j} \tag{3}$$

Where, $Z^{j}_{i,G}$ mentions group member's present position, the position of a leader search agent is specified as $leader\ search\ agent^{j}$, α representing an adaptive mechanism. The uniform random distribution is signified as α ranges from -2 to 2. The new position of search agents while performing grazing strategy is represented as $\vec{Z}^{\,j}_{i,G}$.

3. Mating strategy

The mating strategy of horses in a WHO approach is executed here. In this, the specific characteristics of features are compared to other features in the group. By comparing the aspect of each feature, the search agent can easily identify the optimal one. The evaluation of a mating strategy is given as,

$$Z^{p}_{G,Q} = Crossover\left(Z^{k}_{G,i}, Z^{x}_{G,j}\right) \qquad i \neq j \neq q, p=k=end \tag{4}$$
$$Crossover = mean$$

Where, $Z^{p}_{G,Q}$ mentions the position of a feature p from the group q.

4. Strategy of group leadership

The leader in a search agent group is responsible for guiding the other group members to an appropriate region. The optimal search agent should lead their neighbours towards the important features. Here, the search agents moved their location that contains the most informative features. The leader search agent selects the features which contain more information that assists in detecting intrusions. Each search agent competes for a suitable region. This strategy is expressed as,

$$\overline{Best\ searchagent_{G_i}} = \begin{cases} 2\alpha\cos(2\pi r\alpha)\times\left(\mu - Best\ searchagent_{G_i}\right)+\mu & if\ r_3 > 0.5 \\ 2\alpha\cos(2\pi r\alpha)\times\left(\mu - Best\ searchagent_{G_i}\right)-\mu & if\ r_3 \leq 0.5 \end{cases} \tag{5}$$

$$Best\ searchagent_{G_i} = \begin{cases} Z_{G,i} & if\ \cos t\left(Z_{G,i}\right) < \cos t\left(Best\ searchagent_{G_i}\right) \\ Best\ searchagent_{G,i} & if\ \cos t\left(Z_{G,i}\right) > \cos t\left(Best\ searchagent_{G_i}\right) \end{cases} \tag{6}$$

Where, $\overline{Best\ search\ agent}_{G_i}$ mentions the upcoming position of a leader search agent of a group i, the position of a suitable region is specified as μ, the present position of leader search agent is signified as $Best\ search\ agent_{G_i}$, and an adaptive mechanism is represented as α.

5. Exchanging and Choosing the best search agent

Initially, the leader in search agent groups is randomly selected, and then it is selected based on the fitness function. According to the above equation (6), the leader search agent's position will be changed when one search agent is more optimal than the leader. The Pseudocode of a proposed WHO-FS approach is illustrated in Table 2.

Table 2: Pseudocode for proposed feature selection process

```
Start
Initialize the feature population randomly;
Initialize the entire search agents randomly;
do
      Evaluate fitness function using equation (2);
      Select a suitable search agent as a leader in a random manner;
   end
           Begins exploring criteria;
do
      Perform grazing strategy;
      Change the location of search agents into optimal search agent by equation (3);
      Perform mating strategy;
      Compares the aspects of features;
      Update the position of search agent by equation (4);
   end
          Execute group leadership strategy;
do
      Location of optimal search agent is changed into a suitable region in the exploring space;
      Update the position of optimal search agent by equations (5) and (6);
      Exchange and select best search agent
      Choose essential features by the optimal search agent
   If "Optimal solution is attained according to fitness"
Stop
     Or else
    Continue the process until attaining optimal solution
  end
Stop
```

Hence, the proposed WHO-FS approach selects the optimal search agent for important features. The feature selection process reduces the high dimensionality of data to small, thus enhancing the classifier's efficiency.

C. Proposed DLFAB-IDS model for intrusion detection and classification

The intrusions are detected and classified in the final classification stage with the proposed DLFAB-IDS model's assistance. The proposed classification model is developed by combining the two neural network models AlexNet and BiGRU. The AlexNet method can avoid vanishing gradient issues and generate reduced classification errors. Moreover, the Bi-GRU model makes the convergence rate faster and provides higher classification accuracy. By concerning the benefits of these two neural networks, the proposed study integrates models for detecting and classifying network intrusions in cloud environments. In the first level, the AlexNet model detects intrusions from the network. AlexNet is a type of CNN method which contains 5 convolutional layers, 3 max-pooling layers, 2 fully connected layers and 1 softmax layer correspondingly. The selected features are given as the input in the initial convolutional layer. Then, the max-pooling layer reduces the input size and helps to minimize the number of computations. Downsampling is performed in the pooling layer, extracting the most prominent features. In the AlexNet model, the Relu activation function is utilized in all layers to ease the training process. To avoid overfitting issues, the AlexNet model used dropout layers. Finally, the output layer is employed to detect network intrusions. The architecture of a proposed DLFAB-IDS model is shown in Figure 2. The hyperparameter settings are shown in Table 3.

Table 3: Hyperparameters

Batch size	32
Number of epochs	100
Activation	ReLU
Learning rate	0.01
Number of hidden layers	4
Output layer	2

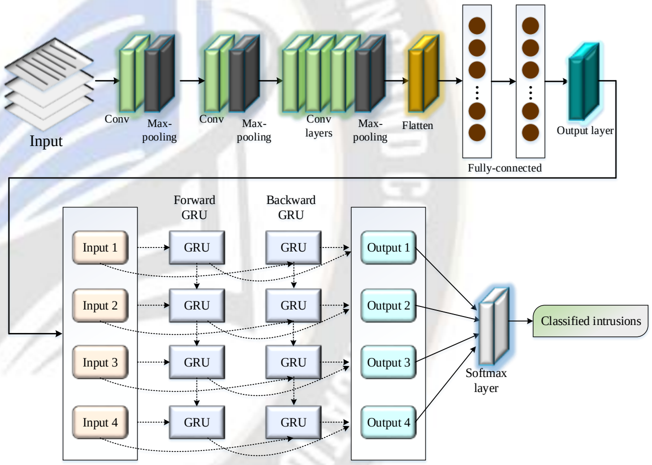


Figure 2: Structure of proposed DLFAB-IDS model

In the second level, the BiGRU model is hybridized with AlexNet to classify the varied types of attacks. The BiGRU model belongs to the RNN model type, mainly utilized in sequential modelling issues obtaining the sequences from both forward and backward directions. This BiGRU model extracts the contextual information from the input data and generates the system for accurate classification. The resultant of forward and backward direction in the BiGRU model is given as,

$$\overrightarrow{l_f} = \overrightarrow{GRU}\left(K_{fs}\right), \qquad m \in [1,\ M] \qquad (7)$$

$$\overleftarrow{l_b} = \overleftarrow{GRU}\left(K_{fs}\right), \qquad m \in [M,\ 1] \qquad (8)$$

Where, fs mentions the input feature sequence, $\overrightarrow{l_f}$ specifies a forward hidden state, $\overleftarrow{l_b}$ represents backwards hidden states. These two hidden states combine the features gathered around K_{fs} to fetch the contextual information. The combined contextual information in the hidden state l_t is represented as,

$$l_t = \left[\overrightarrow{l_f}, \overleftarrow{l_b} \right] \tag{9}$$

In the final softmax layer, the types of intrusions are classified. The formulation of a softmax function is given as,

$$S(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{i} e^{z_j}} \tag{10}$$

Where, x_i signifies the input of a softmax layer and is represented as,

$$x_i = \omega^i z + b_i \tag{11}$$

Where, ω represents the weight and a bias is denoted as b. The bias and weight are determined during the process of training. Thus, the proposed DLFAB-IDS model effectively detects and classifies intrusions from a cloud network system. To enhance the classification performance, the available loss function in the classifier should be minimized. Thus, the proposed DLFAB-IDS model's hyperparameters are optimally tuned through the FDA approach. This FDA approach is one of the physics-based approaches which is generally utilized for solving optimization issues. The FDA approach minimizes the loss function based on the fitness function by optimally tuning the hyperparameters. The computation of a fitness function is given as,

$$Fitness = Min[L_f] \tag{12}$$

$$\text{Here, } L_f = \frac{1}{P} \sum_{p=1}^{P} \left(y_a - y_p^{'} \right)^2 \tag{13}$$

Where, L_f denotes the loss function, P represents the total amount of iterations in the training set, y_a mentions the actual value and $y_p^{'}$ denotes the predicted value. The FDA approach is operated based on the assumptions listed below;

- Each feature contains varied characteristics and sizes.
- There have α positions around each feature; all input features have an aspect or objective value.
- The movement velocity of features directly corresponded to the slope.
- The data features have a velocity of V and features to a direction with minimal altitude.
- Appropriate hyperparameters with reduced loss function are considered the optimal solution.

Initially, the parameters of an FDA approach are randomly initialized, where the parameters are the total population of input parameters δ, the number of neighbouring features α and the neighbourhood radius Δ. The initial position of parameters is evaluated as,

$$parameters\_Z(i) = L_b + rand * (U_b - L_b) \tag{14}$$

Where, $Parameters\_Z(i)$ mentions the i^{th} feature position, the lower limit of a decision variable is represented as L_b and upper limits of the decision sequence is denoted as U_b and the random uniform distribution that ranges from 0 to 1 is mentioned as $rand$. Considering there has α neighbourhood around all parameters and its position is generated by,

$$Neighbor\_Z(j) = Parameters\_Z(i) + randm * \Delta \tag{15}$$

Where, $Neighbor\_Z$ mentions the j^{th} neighbour position and a random normal distribution ranges from 0 to 1 is specified as $randm$. When the value Δ is small, the search process is performed by the search agent in a smaller range, and when it is high, the search process is done in a larger range. Maintaining the global and local search of FDA, the value Δ is linearly decreased from higher to lower. Hence, the direction Δ is towards a random position for high exploring space.

$$\Delta = (rand * Zrand - rand * Parameters\_Z(i)) * \| Best\_Z - Features\_Z(i) \| * \omega \tag{16}$$

Where, $rand$ mentions the random uniform distribution, the random position is specified as $Zrand$, and a non-linear weight ranges from 0 to ∞ is mentioned as ω. In the above derivation, the term $Parameters\_Z(i)$ turn towards the random position $Zrand$. The following equation evaluates the non-linear weight.

$$\omega = \left(\left(1 - \frac{iter}{Max\_Iter} \right)^{2*randm} \right) * \left(\overline{rand} * \frac{iter}{Max\_Iter} \right) * \overline{rand} \tag{17}$$

Where, $\overline{rand}$ mentions the random uniform distribution. The random number $randm$ produces several solutions and enhances global exploration. The slope vector of i^{th} parameter corresponding to the j^{th} neighbour is formulated as,

$$Sv(i,j,l) = \frac{Parameter\_fitness(i) - Neighbor\_fitness(j)}{\| Parameter\_z(i,l) - Neighbor\_z(j,l) \|} \tag{18}$$

Where, $Parameter\_fitness(i)$ mentions the objective value for the parameter i and $Neighbor\_fitness(j)$ specifies the objective value for j^{th} neighbour and the problem dimension is specified as l. The new position of a search agent to choose the optimal hyperparameters is given as,

$$Seachagent\_newZ(i) = searchagent\_Z(i) + V * \frac{Searchagent\_Z(i) - Neighbor\_Z(j)}{\left\| Searchagent\_z(i) - Neighbor\_z(j) \right\|} \quad (19)$$

The new position of search agent i with optimal features is given as $Search\,agentZ(i)$. The FDA approach selects another search agent randomly, and when it reaches the objective function than that of the present search agent, it will remain in a similar direction. Or else the search agent will move in the superior slope direction. The direction of a search agent is represented as,

$$\begin{cases} \quad if \quad Parameters\_\ fitness(r) < Parameters\_\ fitness(i) \\ Searchagent\,Z(i) = Searchagent\_Z(i) + \overline{randm} * (Searchagent(r) - Searchagent\_Z(i)) \\ \quad else \\ Searchagent\,Z(i) = Searchagent\_Z(i) + 2randm * (Best\_Z - Searchagent\,Z(i)) \\ \quad end \end{cases} \quad (20)$$

Where, r represents a random integer. Thus, based on the theory of the FDA, the proposed study reduces the loss function by tuning the hyperparameters. Hence, due to fine hyperparameter tuning, the proposed classification model accurately classifies the network intrusions without affecting the performance. The Pseudocode of a proposed DLFAB-IDS-based intrusions classification is depicted in Table 4.

Table 4: Pseudocode of DLFAB-IDS model

```
Start
Perform DLFAB-IDS-based classification
do
        Initiate the parameters;
        Define the layers of AlexNet;
        Provide selected features as the input of input layer;
        Extracts features using convolutional layers;
        Perform feature reduction in max pooling layers;
        Detect intrusions in output layer;
            end
do
Combine BiGRU with AlexNet;
Determine the outcome of forward and backward direction in BiGRU by equations (7) and (8);
Compute the contextual information in hidden layer using $l_t = \left[ \overrightarrow{l_f}, \overleftarrow{l_b} \right]$;
Evaluate the softmax layer by equation (10);
end
Initiate the parameters in FDA;
Evaluate the fitness function by using equations (12) and (13);
Update the initial position of parameters using $parameters\_Z(i) = L_b + rand * (U_b - L_b)$;
Update the position of neighbouring parameters using equation (15);
Determine the term $\Delta$ using equation (16);
Evaluate non-linear weight by equation (17);
Compute the slope vector using equation (19);
Determine the optimal direction of search agent using equation (20);
If "Optimal solution attained"
end
      Or else
Repeat the process until attaining optimal solution;
Stop
```

IV. RESULTS AND DISCUSSIONS

This section describes the simulation result and analysis of the proposed DLFAB-IDS model. The proposed work used a Python tool for simulation, and the developed techniques' efficacy is proved by comparing the proposed results with other existing techniques. This section includes dataset description, configuration settings, performance metrics, performance metrics results, and comparative analysis. Table 5 shows the system configuration details.

Table 5: System configuration

Manufacturer	Acer©
Processor	Intel (R) Core (TM) i5-4670S CPU @ 3.10GHz 3.10 GHz
Installed memory (RAM)	16.0 GB (15.9 GB usable)
System type	64-bit Operating System
Pen and Touch	No pen or Touch Input is available for this display

A. *Dataset description*

The proposed study adopts the NSL-KDD dataset for experimental analysis. This NSL-KDD dataset is an advanced version of the traditional KDD cup 1999 dataset, which is highly employed for detecting intrusions from the network. Compared with the KDD Cup 1999 dataset, the NSL-KDD dataset generates more reasonable records in training and testing. The NSL-KDD dataset is generally built of the KDDTrain+ training dataset, KDDTest+ and KDDTest-21 testing dataset. The utilized NSL-KDD dataset involves varied normal records and four varied kinds of intrusion-affected records. The five classes presented in this dataset are normal, DoS, probe, R2L and U2R.

B. *Performance metrics*

The performance metrics are necessary to analyze the efficacy of a proposed techniques. The parameters utilized to measure the performance of a proposed model are accuracy, precision, recall and F1-score. The performance achieved by other recent existing methods is compared with the proposed model to know the proposed study's effectiveness.

***Article Received**: 04 April 2023 **Revised**: 20 May 2023 **Accepted**: 31 May 2023*

- *Accuracy:* It is utilized to analyze the efficiency of a classification model. In general, accuracy defines the ratio of correctly classified data to the total input data provided in the dataset. The model which produces higher classification accuracy is said as highly superior. The term accuracy is computed as,

$$Accuracy = \frac{Tp + Tn}{Tp + Tn + Fp + Fn} \quad (21)$$

- *Precision:* This metric computes the total number of correctly categorized data by the proposed DLFAB-IDS model. The precision is evaluated as the sum of accurately classified data to the total number of classified data and is expressed as,

$$\Pr ecision = \frac{Tp}{Tp + Fp} \quad (22)$$

- *Recall:* It defines the total number of correctly classified data by the proposed DLFAB-IDS-based classifier of all perfect predictions that could have been made. The term recall is also called sensitivity. The error rate of classification is also determined by measuring the recall metric. The evaluation of a recall measure is given as,

$$\mathrm{Re}call = \frac{Tp}{Tp + Fn} \quad (23)$$

- *F1-score:* The F1-score metric combines a proposed classifier's recall and precision into a single metric by acquiring their harmonic mean. In the F1 score, accurate recall and precision are termed as one, and if the precision or recall is zero, then the F1 score is assumed as zero. The formulation of an F1-score is represented as,

$$F_1 - score = 2 \times \frac{\Pr ecision \times recall}{\Pr ecision + recall} \quad (24)$$

Where, Tp mentions true positives, Tn specifies true negatives, Fp denotes false positives and Fn mentions false negatives.

C. Performance Evaluation and comparison analysis

This section provides the simulation evaluation of a proposed DLFAB-IDS using the NSL-KDD dataset. The overall performance comparison of proposed and existing techniques is illustrated in Table 6.

Table 6: Performance comparison through several metrics

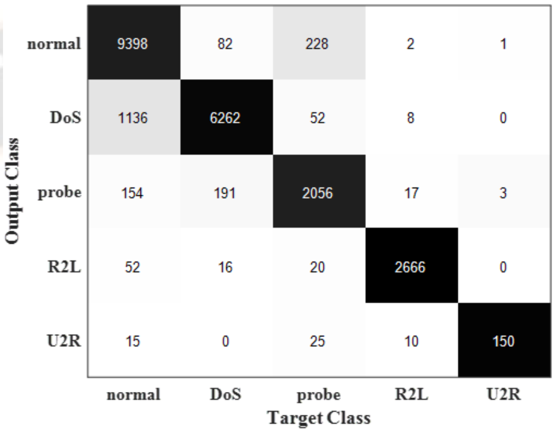
Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
CNN	79.05	81.45	79.05	76.03
RNN	81.29	83.07	81.29	79.25
Conditional VAE	80.10	81.59	80.10	79.08
DNN	78.50	81	78.50	76.50
RNN with MLESM	80.41	81.89	80.41	79.23
DAE-DNN	83.33	86.02	83.33	82.04
Proposed	96.81	96.24	95.84	96.75

The above table mentions the classification performance of a proposed with other existing techniques like CNN, RNN, Conditional variational autoencoder (VAE), RNN with a multilayered echo-state machine (ML-ESM) and Denoising autoencoder (DAE)-DNN [26]. The comparison analysis shows that the proposed model attains higher accuracy, precision, recall and F1-score classification performance. The performance attained in varied classes of attacks using the proposed DLFAB-IDS is shown in Table 7.

Table 7: Performance analysis of multiple classes

DLFAB-IDS(Proposed)					
Classes	Normal	DoS	probe	R2L	U2R
Precision	96.06	95.73	95.44	96.84	94.98
Recall	95.34	95.59	96.01	95.89	96.54
Overall Accuracy	96.51	96.25	95.61	95.73	96.19
F1-score	95.88	95.49	96.45	95.13	95.73
Specificity	96.67	96.33	95.84	95.64	96.48

The above table represents the attained performance of multiple classes in the NSL-KDD dataset. The performance is analyzed for attacks like normal, DoS, probe, R2L and U2R. The proposed model attains better performance results for each class, as in Table 7. The confusion matrix of a proposed classifier during the stage of testing is shown in Figure 3.



	normal	DoS	probe	R2L	U2R
normal	9398	82	228	2	1
DoS	1136	6262	52	8	0
probe	154	191	2056	17	3
R2L	52	16	20	2666	0
U2R	15	0	25	10	150

Figure 3: Confusion matrix analysis

DOI: https://doi.org/10.17762/ijritcc.v11i7s.6987
***Article Received**: 04 April 2023 **Revised**: 20 May 2023 **Accepted**: 31 May 2023*

The confusion matrix is essential to exhibit the classification ability of a proposed DLFAB-IDS model. The dataset utilized in the proposed study contains five different classes, where 9711 normal data is initially tested. From that, 9398 input data are correctly classified as normal, and only a few are incorrectly classified as other classes. Similarly, 7458 DoS data are tested in the proposed DLFAB-IDS classifier, in which 6262 data are correctly detected as DoS and the resting data are misclassified. Consequently, from the total of 2421 probe data, 2056 data are accurately classified by the classifier, and only 395 data are misclassified. Using 2754 R2L data, 2666 data are perfectly classified as R2L and others are wrongly classified. From the 200 data, 150 are correctly classified, and the remaining data are misclassified. Hence, this analysis proves that the proposed classifier is highly suitable for detecting and classifying intrusions in the cloud network system. Figure 4 shows the proposed classifier's accuracy and loss during training and testing.

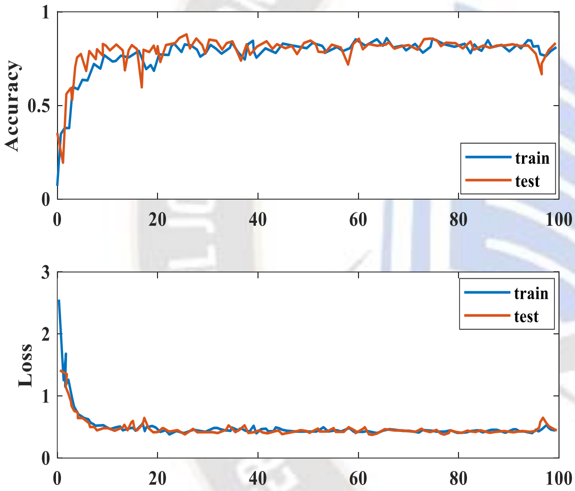


Figure 4: Accuracy and loss analysis

The accuracy and loss are analyzed during the training and testing process of NSL-KDD with the proposed DLFAB-IDS model. The attained accuracy and loss are determined by varying the epochs from 0 to 100. When the epoch size is 0 to 20, a testing stage's accuracy is higher than the training stage's. Then, it remains constant from the epoch of 20 to 100. On the other hand, the attained loss at the training and testing stage is approximately the same from the epoch of 0 to 100. The accuracy performance comparison of proposed and existing techniques is shown in Figure 5.

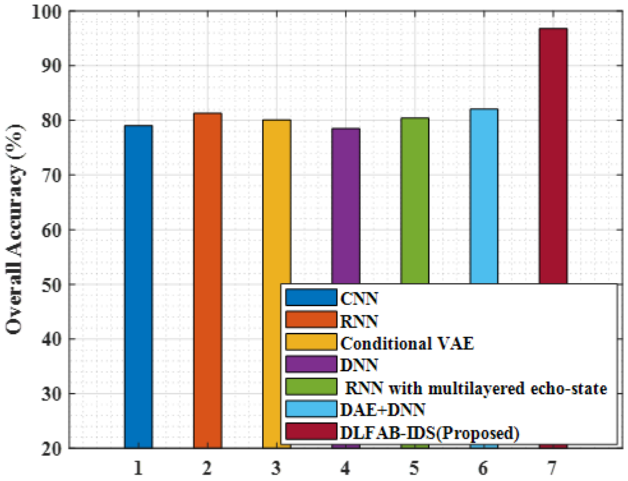


Figure 5: Accuracy performance comparison

The obtained accuracy of a proposed model is compared with other existing studies to show the efficacy of a proposed classifier. The comparison analysis shows that the proposed model obtains higher classification accuracy than other techniques. The existing CNN extracts the essential information from the provided input data but needs a lot of training data to make accurate decisions. Hence, the time complexity is enhanced using the CNN method, leading to higher computational. Thus, the existing CNN method produced reduced classification accuracy. Similarly, the existing RNN method easily falls into gradient vanishing issues, affecting a system's accuracy. Also, the existing DNN method fails to learn the useful features from the provided input data, reducing the classification accuracy. But, the proposed model used a hybrid mechanism to detect network intrusions. The higher learning ability in the proposed neural network model helps to attain improved accuracy than others. The attained accuracy of a proposed DLFAB-IDS mechanism is 96.81%, whereas the existing CNN is 79.05%, RNN is 81.29%, Conditional VAE is 80.10%, DNN is 78.50%, RNN+ML-ESM is 80.41%, and DAE-DNN is 83.33%. This analysis shows that the proposed model is superior to other existing techniques. Figure 6 mentions the precision performance comparison of both proposed and existing techniques.

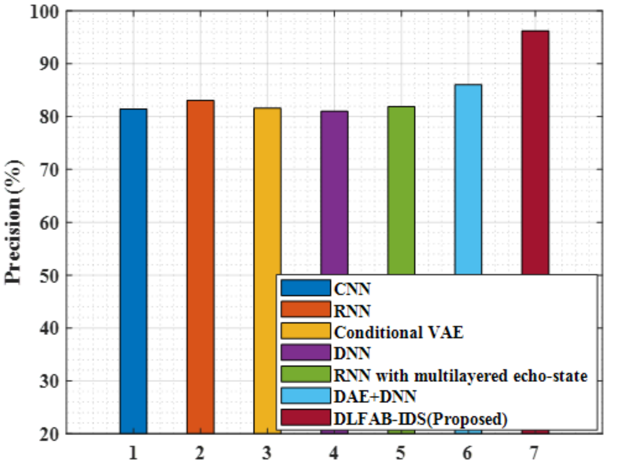


Figure 6: Precision performance comparison

The above graphical representation exhibits the comparison analysis of proposed and existing techniques regarding the precision measure. The simulation analysis shows that the proposed model achieves a higher precision value than others. Because of various challenging issues in existing techniques, precision performance is affected. The proposed model obtains the higher precision value of 96.24%, and the attained precision of existing CNN is 81.45%, RNN is 83.07%, Conditional VAE is 81.59%, DNN is 81%, RNN+ML-ESM is 81.89%, and DAE-DNN is 86.02%. Thus, the result analysis proves the efficacy of a proposed classification model. The recall performance comparison of both a proposed and existing models is shown in Figure 7.

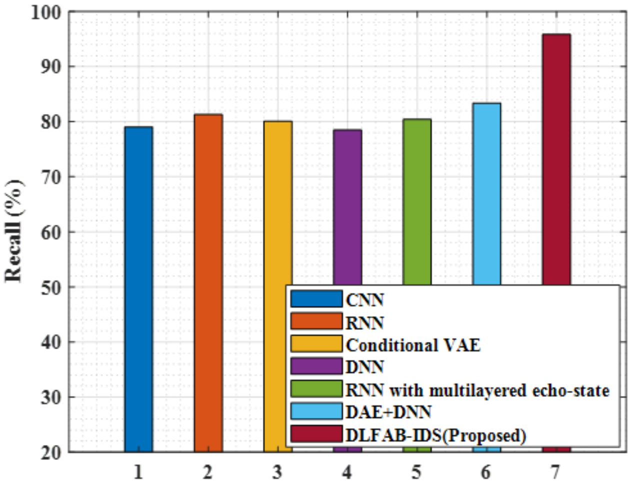


Figure 7: Recall performance comparison

The recall performance analysis is shown in the graph as mentioned above. Compared with other existing techniques, the recall performance in the proposed model is high. The other existing deep learning models attained reduced recall value due to overfitting issues, higher computational complexity, reduced learning ability etc. But the proposed model overcomes such issues by using effective techniques. The attained recall of a proposed model is 95.84%, the existing CNN is 79.05%, RNN is 81.29%, Conditional VAE is 80.10%, DNN is 78.50%, RNN+ML-ESM is 80.41%, and DAE-DNN is 83.33%. The F1-score comparison analysis is depicted in Figure 8.

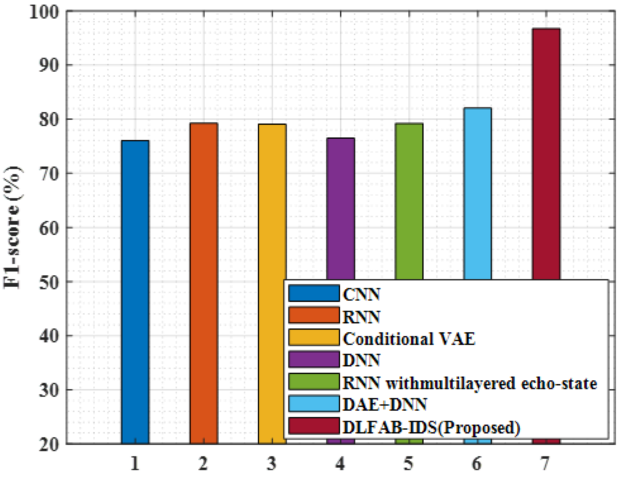


Figure 8: F1-score comparison analysis

The f1-score performance is also analyzed to exhibit the effectiveness of a proposed techniques. The comparison analysis shows that the proposed classifier obtains a higher F1-score value than others. The attained f1-score of a proposed classification model is 96.75%, and the existing CNN is 76.03%, RNN is 79.25%, Conditional VAE is 79.08%, DNN is 76.50%, RNN+ML-ESM is 79.23%, and DAE-DNN is 82.04%. Hence, it proves that the proposed model is more effective for detecting network intrusions than others. The training time comparison of the proposed and existing techniques is shown in Figure 9.

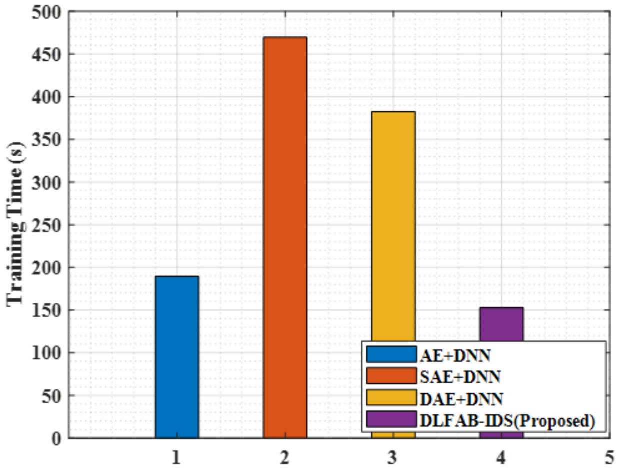


Figure 9: Training time comparison

Analyzing the training time is more important to represent the capability of a classification model. The classifier that required reduced training time is more important because it saves time and improves performance. The above graph reveals that the existing techniques increased training time and complexity. Existing techniques like AE+DNN (Autoencoder with DNN), SAE+DNN (Stacked Autoencoder with DNN) and DAE-DNN are used for training time comparisons. A proposed model's training time is 152.84s, the existing AE+DNN is 189.62s, SAE-DNN is 469.626s, and DAE-DNN is 382.48s. Figure 10 represents the prediction time comparison of both proposed and existing techniques.

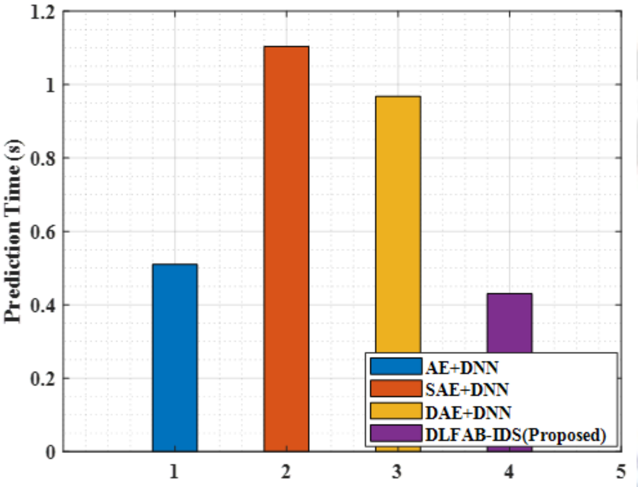


Figure 10: Prediction time comparison

The above graphical analysis shows the prediction time comparison of proposed and existing models. Compared with other existing techniques, the prediction time of a proposed model is reduced because of reduced feature dimensionality. The existing studies involve large feature dimensionality. Hence, it leads the system to attain a higher prediction time. The attained prediction time of a proposed model is 0.43s, the existing AE+DNN is 0.51s, SAE-DNN is 1.104s, and DAE-DNN is 0.968s. Therefore, the results attained in the experimental analysis show that the proposed deep learning model is highly efficient for identifying network intrusions and affording more security to a cloud environment.

V. CONCLUSION

This paper proposes a hybrid deep learning model to detect intrusions from the cloud network system. The three essential stages in the proposed study are pre-processing, selecting optimal features, and classification. Initially, data pre-processing is performed to enhance the quality of input data by eliminating the available noises. This stage sharpens the raw input data and makes it suitable to operate in the further stages. Then, the needed features are selected through WHO approach to reduce the feature dimensionality. The selection of optimal features helps to reduce the prediction time and avoids computational complexity. Next, such selected features are injected as an input of the deep hybrid learning-based classification stage. The AlexNet model is integrated with the Bi-GRU model to detect and classify network intrusions. To enhance the classifier's performance, hyperparameter tuning is done with the assistance of an FDA approach. Hence, it allows the system to perform higher classification and minimizes the training error. The result analysis shows that the proposed model obtains higher classification performance in terms of accuracy 96.81%, precision 96.24%, recall 95.84%, f1-score 96.75%, training time 152.84s and prediction time 0.43s. In future, different adversarial learning methods will be utilized to gain more detection performance. Furthermore, the result analysis will be improved by using more datasets.

ACKNOWLEDGEMENT

None

REFERENCES

[1] K.K. Nguyen, D.T. Hoang, D. Niyato, P. Wang, D. Nguyen, E. Dutkiewicz, "Cyberattack detection in mobile cloud computing: A deep learning approach". In 2018 IEEE wireless communications and networking conference (WCNC), 2018, pp. 1-6.

[2] N. Subramanian, A. Jeyaraj, "Recent security challenges in cloud computing". Computers & Electrical Engineering, vol. 71, 2018, pp. 28-42.

[3] G. Somani, M.S. Gaur, D. Sanghi, M. Conti, R. Buyya, "DDoS attacks in cloud computing: Issues, taxonomy, and future directions". Computer Communications, vol. 107, pp. 30-48, 2017.

[4] A. Bakr, A. El-Aziz, H.A. Hefny, "A Survey on mitigation techniques against DDoS attacks on cloud computing architecture". International Journal of Advanced Science and Technology, vol. 28, no. 12, pp. 187-200, 2019.

[5] P.S. Mishra, E.S. Pilli, V. Varadharajan, U. Tupakula, "Intrusion detection techniques in cloud environment: A survey". Journal of Network and Computer Applications, vol. 77, pp. 18-47, 2017.

[6] Y. Gao, Y. Liu, Y. Jin, J. Chen, H. Wu, "A novel semi-supervised learning approach for network intrusion detection on cloud-based robotic system". IEEE Access, vol. 6, pp. 50927-38, 2018.

[7] A.S. Saljoughi, M. Mehrvarz, H. Mirvaziri, "Attacks and intrusion detection in cloud computing using neural networks and particle swarm optimization algorithms". Emerging Science Journal, vol. 1, no. 4, pp. 179-91, 2017.

[8] S.I. Shyla, S.S. Sujatha, "Cloud security: LKM and optimal fuzzy system for intrusion detection in cloud environment". Journal of Intelligent Systems, vol. 29, no. 1, pp. 1626-42, 2019.

[9] P. Singh, V. Ranga, "Attack and intrusion detection in cloud computing using an ensemble learning approach". International Journal of Information Technology, vol. 13, pp. 565-71, 2021.

[10] V. Kharchenko, Y. Ponochovnyi, O. Ivanchenko, H. Fesenko, O. Illiashenko, "Combining Markov and Semi-Markov Modelling for Assessing Availability and Cybersecurity of Cloud and IoT Systems". Cryptography, vol. 6, no. 3, pp. 44, 2022.

[11] F.S.L. Filho, F.A. Silveira, A. de Medeiros Brito Junior, G. Vargas-Solar, L.F. Silveira, "Smart detection: an online approach for DoS/DDoS attack detection using machine learning". Security and Communication Networks, vol. 2019, pp. 1-5, 2019.

[12] Renato Costa, Deep Reinforcement Learning for Autonomous Robotics , Machine Learning Applications Conference Proceedings, Vol 2 2022.

[13] Ms. Mohini Dadhe, Ms. Sneha Miskin. (2015). Optimized Wireless Stethoscope Using Butterworth Filter. International Journal of New Practices in Management and Engineering, 4(03), 01 - 05. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/37

[14] M. Zekri, S. El Kafhali, N. Aboutabit, Y. Saadi, "DDoS attack detection using machine learning techniques in cloud computing environments". In 2017 3rd international conference of cloud computing technologies and applications (CloudTech), IEEE vol. 2017, pp. 1-7, 2017.

[15] A.R. Wani, Q.P. Rana, U. Saxena, N. Pandey, "Analysis and detection of DDoS attacks on cloud computing environment using machine learning techniques". In 2019 Amity International conference on artificial intelligence (AICAI), IEEE vol. 2019, pp. 870-875.

[16] M. Manickam, N. Ramaraj, C. Chellappan, "A combined PFCM and recurrent neural network-based intrusion detection system for cloud environment". International Journal of Business Intelligence and Data Mining, vol. 14, no. 4, pp. 504-27, 2019.

[17] S. Ho, S. Al Jufout, K. Dajani, M. Mozumdar, "A novel intrusion detection model for detecting known and innovative cyberattacks using convolutional neural network". IEEE Open Journal of the Computer Society, vol. 2, pp. 14-25, 2021.

[18] Rossi, G., Nowak, K., Nielsen, M., García, A., & Silva, J. Machine Learning-Based Risk Analysis in Engineering Project Management. Kuwait Journal of Machine Learning, 1(2). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/114

[19] A. Kajal, S.K. Nandal, "A hybrid approach for cyber security: improved intrusion detection system using Ann-Svm". Indian Journal of Computer Science and Engineering, vol. 11, no. 4, pp. 412-25, 2020.

[20] P.A. Resende, A.C. Drummond, "A survey of random forest based methods for intrusion detection systems". ACM Computing Surveys (CSUR), vol. 51, no. 3, pp. 1-36, 2018.

[21] J.K. Seth, S. Chandra, "MIDS: Metaheuristic based intrusion detection system for cloud using k-NN and MGWO. In Advances in Computing and Data Sciences: Second International Conference., ICACDS 2018., Dehradun., India., April 20-21., 2018., Revised Selected Papers., Part I 2, Springer Singapore, pp. 411-420, 2018.

[22] N. Balakrishnan, A. Rajendran, D. Pelusi, V. Ponnusamy, "Deep Belief Network enhanced intrusion detection system to prevent security breach in the Internet of Things". Internet of things, vol. 14, pp. 100112, 2021.

[23] W. Wang, X. Du, D. Shan, R. Qin, N. Wang, "Cloud intrusion detection method based on stacked contractive auto-encoder and support vector machine". IEEE transactions on cloud computing, vol. 10, no. 3, pp. 1634-46, 2020.

[24] P. Ghosh, A. Karmakar, J. Sharma, S. Phadikar, "CS-PSO based intrusion detection system in cloud environment". In Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS, Springer Singapore, vol. 2018, no. 1, pp. 261-269, 2019.

[25] C. Yin, Y. Zhu, J. Fei, X. He, "A deep learning approach for intrusion detection using recurrent neural networks". IEEE Access, vol. 5, pp. 21954-61, 2017.

[26] A. Agarwal, P. Sharma, M. Alshehri, A.A. Mohamed, O. Alfarraj, Classification model for accuracy and intrusion detection using machine learning approach. PeerJ Computer Science, vol. 7, pp. e437, 2021.

[27] T. Su, H. Sun, J. Zhu, S. Wang, Y. Li, "BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset". IEEE Access, vol. 8, pp. 29575-85, 2020.

[28] M. Zhong, Y. Zhou, G. Chen, "Sequential model based intrusion detection system for IoT servers using deep learning methods". Sensors, vol. 21, no. 4, pp. 1113, 2021.

[29] Makarand L, M. . (2021). Earlier Detection of Gastric Cancer Using Augmented Deep Learning Techniques in Big Data with Medical Iot (Miot). Research Journal of Computer Systems and Engineering, 2(2), 22:26. Retrieved from https://technicaljournals.org/RJCSE/index.php/journal/article/view/28

[30] Y.N. Kunang, S. Nurmaini, D. Stiawan, B.Y. Suprapto, "Attack classification of an intrusion detection system using deep learning and hyperparameter optimization". Journal of Information Security and Applications, vol. 58, pp. 102804, 2021.

[31] Tripathi, A. ., Pandey, R. ., & Singh, A. . (2023). Comparison of Performance of Boneh-Shaw Finger Printing Codes with Tardos Under Randomized Bits Collusion Attacks . International Journal of Intelligent Systems and Applications in Engineering, 11(2s), 01–10. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2501