International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Performance Comparison Analysis of Classification Methodologies for Effective Detection of Intrusions

Rajesh Bingu[1*] S. Jothilakshmi[2] N. Srinivasu[3]

[*1]Research Scholar, Department of Information Technology,
Annamalai University, Chidambaram, Tamil Nadu 608002-India.
[2]Associate Professor, Department of Information Technology,
Annamalai University, Chidambaram, Tamil Nadu 608002-India.
[3]Professor, Department of Computer Science and Engineering,
Koneru Lakshmaiah Education Foundation, Vijayawada, India.
*Corresponding Email: govindajeeyardasan@gmail.com

Abstract: Intrusion detection systems (IDS) are critical in many applications, including cloud environments. The intrusion poses a security threat and extracts privacy data and information from the cloud. The user has an Internet function that allows him to store personal information in the cloud environment. The cloud can be affected by various issues such as data loss, data breaches, lower security and lack of privacy due to some intruders. A single intrusion incident can result in data within computer and network systems being quickly stolen or deleted. Additionally, intrusions can cause damage to system hardware, resulting in significant financial losses and exposing critical IT infrastructure to risk. To overcome these issues, the study employs the performance comparison analysis of Autoencoder Convolutional neural network (AE+CNN), Random K-means clustering assisted deep neural network (RF+K-means+DNN), Autoencoder K-means clustering assisted long short term memory (AE+K-means+LSTM), Alexnet+Bi-GRU, AE+Alexnet+Bi-GRU and Wild horse AlexNet assisted Bi-directional Gated Recurrent Unit (WABi-GRU) models to choose the best methodology for effective detection of intrusions. The data needed for the analysis is collected from CICIDS2018, UNSW-NB15 and NSL-KDD datasets. The collected data are pre-processed using data normalization and data cleaning. Finally, through this research, the best model suitable for effective intrusion detection can be identified and used for further processes. The proposed models, such as RF+K-means+DNN, AE+K-Means+LSTM, AlexNet Bi-GRU, AE+Alexnet+Bi-GRU and WABi-GRU can obtain an accuracy of 99.278%, 99.33%, 99.45%, 99.50%, 99.65% for the CICIDS dataset 2018 for binary classification. In multi-class classification, the AlexNet Bi-GRU, AE+Alexnet+Bi-GRU and WABi-GRU can attain accuracy of 99.819%, 99.852% and 99.890%. In NSL-KDD, the AlexNet Bi-GRU, AE+Alexnet+Bi-GRU and WABi-GRU achieve accuracy of 99.34%, 99.546% and 99.7%. In UNSW-NB 15 dataset, AlexNet Bi-GRU, AE+Alexnet+Bi-GRU and WABi-GRU achieve accuracy of 99.313%, 99.399% and 99.53%. AlexNet Bi-GRU-based models can obtain better performances than other existing models.

Keywords: Autoencoder (AE), long short-term memory (LSTM), AlexNet, BiGRU, Random forest (RF) and Convolutional Neural Networks (CNN).

1. INTRODUCTION

Cloud computing is essential in the IT business because it allows for the storage of enormous amounts of data at a low cost while maintaining excellent data security. Cloud is a multi-source environment. Depends on user's needs, end user should use multiple services[1]. Many business organizations can store their data in the cloud environment. This information is subject to security issues such as integrity, confidentiality and availability [2]. The cloud environment provides uninteresting services so that intruders can access the data, resulting in increased security problems and misuse of the information of cloud service providers and resources. Cloud computing is easily affected by various attacks such as data breaches, data loss, lower security, insecure interfaces, malicious insiders, etc. [3]. The data stored in the cloud environment is readily accessible to users across the globe through internet-capable devices. The cloud environment acts as a pay-per-use model for storing computing resources and data because the cloud is a highly flexible and scalable environment [4, 5].

Multiple attacks occur in the cloud environment; these are averted through the implementation of an intrusion detection system designed to mitigate cloud-based attacks. An intrusion detection mechanism is initially devised to mitigate assaults and uncertainty within the cloud environment [6]. A multitude of attack vectors manifest within the cloud environment, including Distributed Denial of Service (DDOS) and Confidentiality, Integrity, and Availability (CIA) attacks [7]. Various intrusion techniques, such as anomaly-based and network intrusion detection, are generated, but these models have limitations. Constraints such as adequate security are only provided for the cloud environment [8]. Therefore, the intrusion detection system is executed by various Deep Learning (DL) algorithms such as Genetic Algorithm (GA),

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Fuzzy Logic (FL), Recurrent Neural Network (RNN), Artificial Neural Network (ANN), and CNN [9, 10].

The primary purpose of these classifiers is to identify intrusions in cloud environments; feature dimensionality is added to the classifiers to improve their accuracy [11, 12]. Intrusion detection can also be performed by traditional classifiers in the DL algorithm, such as the softmax classifier in the output layer. This traditional model may reduce the overall accuracy and efficiency of the detection system. These limitations are alleviated by adding advanced DL classifiers instead of traditional methods [13]. The intrusion detection system is used in many applications, such as cloud environments, video anomaly detection and IoT environments. In video anomaly detection, the intruder object can be detected by various algorithm approaches such as Gaussian Mixture Model (GMM), SORT (Simple Online and Real Time Tracking), and You Only Look Once (YOLO) algorithm [14, 15].

Business organizations can store information about companies in a cloud environment. Still, it is less secure to store data in the cloud and prevent attacks such as firewalls, viruses and intruders [16]. These attacks are prevented from the cloud by an ID system using the Back Propagation Neural Network (BPNN) based on the DL algorithm [17, 18]. These are the intrusions or attacks by intruders in the cloud environment, which are efficiently provided by the DL algorithms. This improves efficiency and increases the security of the system in the cloud model and other applications [19, 20].

A. Motivation

The intrusion detection system is mainly used to detect attacks or abnormal activities in video surveillance, cloud environments and IoT applications. Several existing models related to intrusion detection are analyzed. Several limitations are identified in this analysis, such as the application is less secure, complex problems are difficult to handle, and the development of the classifier model incurs high costs. In view of the aforementioned constraints, a comparison analysis of effective classification methods has been developed to improve the intrusion detection system's performance and establish which model is most effective at identifying intrusions in a cloud environment. The primary contributions of the proposed study are outlined below.

- ✓ To introduce an effective performance analysis of classification methodologies for an effective intrusion detection mechanism.
- ✓ To introduce data cleaning and normalization in pre-processing to remove undesirable data and improve the quality.
- ✓ To develop an autoencoder-based convolutional neural networks model for classify intrusion detection.
- ✓ To develop a Random K-means clustering-assisted deep neural network to identify the types of attack in a cloud environment.
- ✓ To construct an Autoencoder, K-means clustering assisted long short-term memory for improving the classification accuracy.
- ✓ To generate a Wild horse, AlexNet assisted Bi-directional Gated Recurrent Unit models to identify attacks types and to enhance the accuracy of the models.
- ✓ To introduce an AlexNet Bi-GRU model for detecting intrusion in a cloud environment in an efficient manner.
- ✓ To construct an Autoencoder AlexNet Bi-GRU model to identify the several types of attacks in a cloud environment.
- ✓ To compare these four models with several performance metrics to show which model can obtain better performances and is suitable for intrusion detection.

The research works are structured according to the proposed model as follows: The survey pertaining to intrusion detection in a cloud environment is detailed in Section 2. The proposed methodology of the model is explicated in Section 3, the results and discussion are elaborated upon in Section 4, and the conclusion of the paper is presented in Section 5.

II. RELATED WORKS

The following section analyses several existing models related to intrusion detection in cloud environments.

Sethi et al. [21] proposed deep reinforcement based adaptive cloud IDS to improve security in cloud environments. Here, UNSW-NB15 dataset was used to improve the accuracy. Here, IDS architecture includes three main components present in the agent network, namely a protocol generation virtual machine, a structured protocol generator VM and an agent. Different types of virtual machines are hosted on the host networks. With this model, both precision and the false positive rate (FPR) can be reduced. IDS has been implemented in cloud environments to monitor and detect malicious activities in order to prevent data loss, fraud, and misuse. This model may take more time to detect intruders in cloud environment, one of the major limitations.

Abusitta et al. [22] proposed a proactive cooperative multi-cloud IDS based on an auto-encoder model to detect intruders in cloud environments. The feedback is reconstructed by denoising the Autoencoder (DA) from partial feedback. This DA was used to block the generation of a deep neural network and make results about intruders even there is a no feedback from IDS. This model can achieve 95% accuracy. The proposed method was implemented using a GPU-based tensor flow and subsequently assessed on a real-time dataset. One of the limitations was that developing the classifier model required higher costs.

Balamurugan et al. [23] developed a NK-RNN normalized K-means clustering with RNN to detect attacks or intrusions in cloud environment. User can access cloud environment using a one-time signature technique. Here, two main algorithms, Packet Inspection (PS) and NK-RNN, should detect cloud intrusion. PS was used to analyze convolution and port scanning attacks, and NK-RNN was used to distinguish intruder and normal packets. This helps prevent attacks from intruders on the cloud network. This model cannot solve the complex problems of intrusion detection.

Umair et al. [24] suggested a hybrid DL with the combination of CNN and LSTM to detect intruders in cloud

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

environments. To extract features from the data, a multi-layer convolutional neural network and a softmax layer were used. Here, two publicly available datasets were used namely KDDCUP 99 and NSL-KDD for intrusion detection. The features were extracted by a multi-layer convolutional neural network, and the intrusion detection can be classified by softmax classifiers. This model may require a larger number of parameters to detect intrusions into the cloud environment, which is one of the major limitations.

Snehi et al. [25] introduced a Deep Belief Network (DBN) as a potential solution for edge-of-things (EOT) network intrusion detection. The features were extracted from the input using the min-max scaling method. During feature selection, the nominal features were converted into numerical values to improve the classifier model. The data was collected from the public source dataset UNSW-NB15 and used for intrusion detection. This model can achieve 85% classification accuracy. The main limitation of the model was the lower accuracy because a single classifier model can be used in the classification phase. Various services related to intrusion detection systems are analyzed, described in Table 1.

TABLE I: PERFORMANCE ANALYSIS FOR EXISTING MODELS

Author and references	**Methods**	**Dataset**	**Demerits**	**Performances**
Sethi et al. [21]	Adaptive cloud IDS	UNSW-NB15	Consume more time	Accuracy FRP AUC
Abusitta et al. [22]	Proactive multi-cloud cooperative IDS	Real-time dataset	High cost	Accuracy of 95%
Balamurugan et al. [23]	NK-RNN	Real-time dataset	Doesn't handle complex problems	Precision Recall F1-Score
Umair et al. [24]	CNN and LSTM	NSL-KDD and KDDCUP 99	Need more number of parameters	Accuracy Precision Recall F1-Score
Snehi et al. [25]	DBN	UNSW-NB15	Less accuracy in classification	Accuracy of 85%

This table analyses the performances associated with intrusion detection systems using deep learning. Several limitations are identified in this analysis, such as existing models are unable to handle complex problems, classification takes longer, and more parameters are required. To overcome these limitations, an effective performance comparison for different classification models is developed to prove which classification model is the best and achieve better accuracy.

III. PROPOSED METHODOLOGY

Intrusion detection is an important system widely used in many applications to detect attacks and intrusions. This IDS is followed by several stages, namely pre-processing of feature extraction and classification. This paper will compare to prove which proposed model can achieve better results. This section briefly explains the proposed model, like AE+CNN, RF+K-means+DNN, AE+K-Means+LSTM Alexnet+Bi-GRU, AE+Alexnet+Bi-GRU and WABi-GRU models. The proposed model is shown in Figure 1.

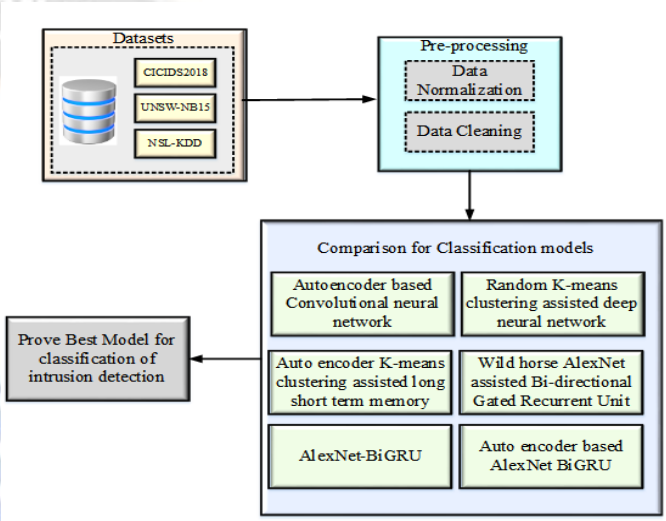

Figure 1: Schematic architecture of the recommended models

In this Figure, the comparison is performed for various classification models like AE+CNN, RF+K-means+DNN, AE+K-Means+LSTM, AE+Alexnet+Bi-GRU and WABi-GRU and WABi-GRU for intrusion detection. To prove which classifier models should attain better efficiency and accuracy than other models. This can be an effective model for classifying intrusion detection. Here, three datasets, namely CICIDS 2018, UNSW-NB15 and NSL-KDD are collected from public sources. The undesirable data is removed from the dataset in pre-processing stage and the comparison is performed.

A. Pre-processing

Pre-processing occurs during this phase to enhance the quality of the input images by eliminating extraneous and superfluous data present in the dataset. This pre-processing should be carried out in two stages: data normalization and data cleaning.

1) *Data Normalization:* This normalization mechanism is used to increase the speed in the training phase because all data is used in the training process, which includes similar scales ranging from 0 to 1. It is performed by min-max normalization.

$$G_{normal} = \frac{G - G_{\min}}{G_{\max} - G_{\min}} \quad (1)$$

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Here, the normalization results are denoted by G_{normal}, before normalization, the initial value is represented by G, the maximum and minimum values of the features are represented as $G_{\max}$ and $G_{\min}$. In the given input data, the noise is removed by this normalization mechanism.

2) *Data Cleaning:* In the deep learning model, the raw input data is not suitable because it contains noise, missing values, duplicate columns, num values, erroneous labels, infinity values, and redundant data. In this data-cleaning process, the unwanted data is removed to enhance the performance of the proposed classification stage. Due to this process, outliers are detected and eliminated, inconsistencies are solved.

B. Classification models

In the following sections, various classification models, such as AE+CNN, RF+K-Means+DNN, AE+K-Means+LSTM, and WABi-GRU models are described in detail.

1) Autoencoder-based Convolutional neural network: Due to the prevalence of autoencoder in compression mechanisms, they are also employed for anomaly or intrusion detection in various applications. Unsupervised learning of the AE variety comprises a hidden layer, an input layer, and an output layer. This Autoencoder contains two phases, namely decoder and encoder stages, and the encoder layer is an important layer between the encoder and decoder. The data reduction performed by the encoder is contingent upon the total number of neurons in the network layer and corresponds to the type of neural network component. The encoder compresses this data prior to passing it to the decoder via the code component. Finally, the decoder contributes to the restoration of raw data as a participant in the neural network. The encoding process from input to hidden layer is mathematically expressed in the following equation.

$$G = b\theta_1(Y) = \delta(Z_{pq}Y) + \alpha_1 \quad (2)$$

The decoding process is carried out from the hidden layer to the output layer and can be stated as follows.

$$L = b\theta_2(G) = \delta(Z_{qr}G + \alpha_2) \quad (3)$$

The equation represents, the input vectors by $Y = (Y_1, Y_2, \ldots, Y_n)$, the output vectors are represented as $L = (L_1, L_2, \ldots, L_n)$, and the low dimensional vector output from the hidden layer can be represented by $G = (G_1, G_2, \cdots, G_m)$; $Y \in T^n, L \in T^n, and\, G \in T^m$. The dimensions of the input vector are represented as n, the number of the hidden unit is denoted by *m*, and The notation used to represent the weights of the connection matrix between the input and concealed layers is $Z_{pq} \in T^{m \times n}$, the hidden layer and the output layer is represented as $Z_{qr} \in T^{n \times m}$. The hidden layer and output layer for the activation function are represented as $b\theta_1(\cdot)$ and $b\theta_2(\cdot)$. The bias vectors of the hidden layer and input are represented as $\alpha_2 \in T^{m \times 1}$ and $\alpha_1 \in T^{n \times 1}$. The model employs the sigmoid as an activation function, which is mathematically represented as follows.

$$b\theta_1(\cdot) = b\theta_2(\cdot) = \frac{1}{1+e^{-Y}} \quad (4)$$

Adjusting encoder and decoder parameters reduces the error between input data and output. Once the features have been extracted, they are inputted into the CNN classifier model in order to enhance the model's classification accuracy. This CNN model contains different layers: the input, output, convolutional, dense, max-pooling and drop-out layer.

a) Input layer: The input layer receives the abstracted feature set and processes it using artificial neurons. The initial data serves as the input for the subsequent layers. The artificial weights of the neurons are randomly calculated by transferring them to other subsequent layers.

b) Convolutional layer: The input image features are declared as output to the convolutional layer. Here, the original raw data from the input layer is filtered, and convolution operations are performed. The subsequent layer receives the filtered data and constructs the output layer's feature map using the input layer's feature maps.

c) Max-pooling layer: The major task of max-pooling layer is to convert feature representation into meaningful information. Here, the compatibility of the image dimensions, the parameter decrement and the calculation are carried out by adopting these layers. Pooling helps improve the stability of the input data.

d) Drop-out and dense layer: The drop-out layer is important when training the CNN layer to protect the model from overfitting problems. The dense layer receives data from each neuron of the aforementioned layer.

e) Fully connected layer: FC Fully connected layer use activation function softmax. It insists that the previous layer neurons is linked to the next layer for making several classes.

f) Output layer: The output layer is obtained from the previous input layer with calculations of neurons. The input layer is compressed to produce output into different classes. The final results are obtained from the output layer and refined for higher classification accuracy. The CNN model is used to classify intrusion in cloud environments [26]. The schematic architecture for an autoencoder-based convolutional neural network is shown in Figure 2.

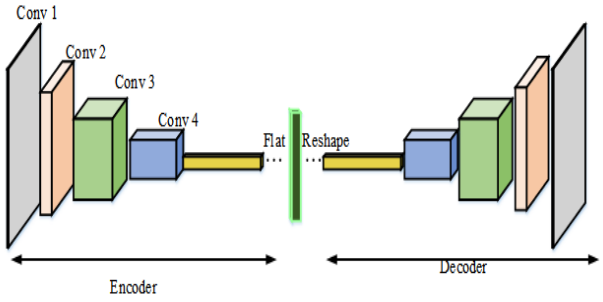

Figure 2: Schematic architecture for AE-CNN

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

2) *Random K-means clustering assisted deep neural network:* Random Forest is the type of unsupervised machine learning algorithm used to gather features from original images. Based on given constraints, the subset for dividing the internal nodes for a predictor variable is selected. During the classification process, the lower bound of the random variable is ascertained using entropy. The mathematical expression for the internal node and entropy of the decision tree is as follows:

$$H = -\sum_{q-1}^{c} g_q \times \log\left(g_q\right) \tag{5}$$

Here, priority probability of the class and amount of unique classes is represented as g_q and C. In each division of the decision tree, the value should be increased to gather a large amount of information. After feature extraction, the clustering is performed by K-Means clustering algorithm [27]. Here, cluster formation is determined by the centroid value. Reducing the distance between a data instance and a group that is a part of the cluster is the primary objective of the clustering algorithm. This process has built a loop. This loop is used to find the K centre. Finally, after the successful completion of the process, a 'K' number of the cluster is selected by using this algorithm. The square error function of the K-Means clustering is mathematically expressed in the following equation.

$$\sum_{a=1}^{p}\sum_{b=1}^{q}\left(\|ca - db\|\right)^2 \tag{6}$$

Here, P number of objects is denoted as C_a and K-Means clustering process is explained in the following Table 2.

TABLE II: K-MEANS CLUSTERING

Step 1: Set the clusters as an input **Step 2:** Select the first K centroids randomly $db; b = 1; 2; \ldots; K$ from data objects. **Step 3:** Length between data objects and the K-cluster is evaluated by the following equation. $\sqrt{\sum_{a=1}^{p}\left(c_a - d_b\right)^2}$ **Step 4:** Allocate cluster data objects and search for minimum distance **Step 5:** Centroids values are updated by the following equation. $d_b = \frac{1}{P}\sum_{a=1}^{p} b_a$

In the following condition, if one is satisfied, the K-Means algorithm is terminated. The conditions are described in the following Table 3.

TABLE III: CONDITIONS FOR K-MEANS CLUSTERING ALGORITHM

Step 1. The centroid average is a change **Step 2**. If the iteration has reached the maximum limit and
Step 3. The cluster object is ideal.

Once the datasets have been clustered using the K-Means clustering algorithm, DNN is utilized to conduct the classification phase. The DNN is the most popular computational network; it contains many hidden layers and interconnecting nodes. This DNN algorithm contains three main steps. The initial step is topology model, which explains number of layers and evaluates connection for neurons in each layer. Second step is forward propagation layer with activation function and perceptron classifier used by artificial neurons. Finally, the third step is the backward propagation contains loss function and optimizer.

First step is the topology model, which contains an input, hidden, and output layer. The input layers contain 125 nodes, which are represented by features of pre-processed data. The hidden layer is placed in between input and output layers. It contains two hidden layers with 50 neural nodes. Finally, at output layer, it provides results for classification based on the type of intrusion into the cloud environment. The next process is forward propagation, which targets to calculate the results like attack or normal using a perceptron classifier. ANN is used as a multi-layer perceptron and is the basis of DNN. The perceptron layer is mathematically expressed in the following equation.

$$x = \sum\nolimits_{a=1}^{p} Ya\, Za + c \tag{7}$$

Here, the number of nodes in the layer and the values of the nodes are represented as p and y. The weights and the bias of the nodes are denoted by z and C. Existing techniques rely primarily on activation functions including Sigmoid, ReLU, Softmax, and Tanh for intrusion detection mechanisms. ReLU is utilized in the hidden layer of this model; its expression is as follows.

$$g(y) = for\, y \le 0 = for\, y > 0 \tag{8}$$

The equation representing the softmax activation function utilized in the output layer is as follows.

$$\sigma(w)_b = \frac{e^{w_b}}{\sum_{l=1}^{l} e^{w_l}} \quad for\, b = 1,2,3\ldots\ \ldots, l \tag{9}$$

Here, the softmax function is represented by σ, and the input vector is denoted by W. The standard exponential of the input vector and the total number of classes are represented as e^{w_b} and l. The entire number of classes and the standard exponential function of the input vector are represented by e^{w_l}. Finally, the third stage is backward propagation, the popular way to train the DNN model. It contains optimizers and loss functions. The loss function causes the system values to be reduced in order to attain the best model parameter values. Thus, minimize the loss function for each attribute to obtain optimal value. The optimizer is used to get the best parameter value. The various loss functions are used in all applications, such as RMSprop, Adam, stochastic gradient decenter, and batch gradient decenter optimizer. Among these optimizers, Adam is the best optimizer. Therefore, the schematic architecture of the deep neural network supported by random K-means clustering is shown in Figure 3.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

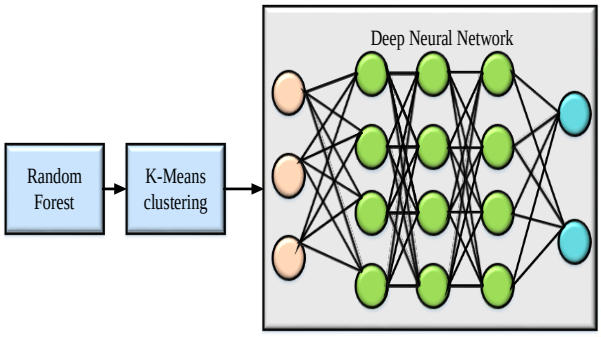

Figure 3: Architecture for RF+K-means+DNN

3) *Auto encoder K-means clustering assisted long short-term memory:* In this model, auto-encoders are used to extract features, and K-means clustering is utilized to cluster datasets. Finally, intrusion detection can be performed using the LSTM network model to enhance efficiency and accuracy of the model. LSTM reduces the vanishing and exploding gradients in the RNN model [28]. The main goal of the LSTM is to regulate the cell state by using three gates, namely forget gate, input gate, and output gate. Forget gate (h_s) keep the information of the previous gate (b_{s-1}) by using the values of the input (y_s) hidden state (g_{s-1}), and the output may vary from 0 or 1. The cell state b_s is generated by using mathematical operations b_{s-1}, h_s *and* p_s. The flow of information should be controlled by the output gate (o_s), which has been flowing from the current cell state to the hidden state, and the values maybe 0 or 1. The mathematical equation is represented in the below equation.

$$h_s = sigmoid\left(Z_{hy} y_s + Z_{hg} g_{s-1} + c_h\right) \quad (10)$$

$$p_s = sigmoid\left(Z_{py} y_s + Z_{pg} g_{s-1} + c_p\right) \quad (11)$$

$$b_s = b_{s-1} \otimes h_s + p_s \otimes \tanh\left(Z_{by} y_s + Z_{bg} g_{s-1} + c_b\right) \quad (12)$$

$$o_s = sigmoid\left(Z_{oy} y_s + Z_{og} g_{s-1} + c_o\right) \quad (13)$$

$$g_s = o_s \otimes \tanh\left(b_s\right) \quad (14)$$

Here, the input vector is represented by $y_s \in M^t$. The superscripts such as t and u in the $Z \in M^{u*t}$, $c \in M^u$, which denotes the dimensions of the input vector and number of words in the dataset. At any time S, the input vectors are represented as (y_s), the previous hidden state and the cell state is denoted by (g_{s-1}) and (b_{s-1}). The output of the current hidden state and cell state is represented as g_s and (b_s). The element-wise vector multiplication is represented as $\otimes$.

Here, the features are selected from the dataset by autoencoder model, and the clustering of data is performed by K-means clustering. Finally, intrusion detection can be performed by LSTM as a classifier model. and feature extraction is done by the K-Means clustering model. The proposed AE+K-Means+ LSTM is represented in the following Figure 4.

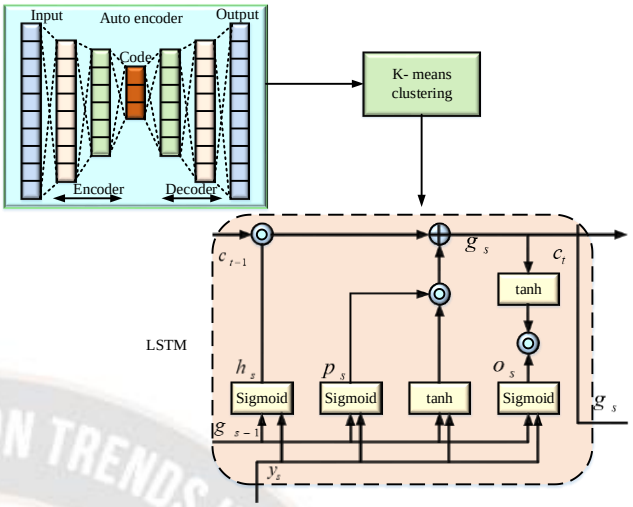

Figure 4: Block diagram of AE+K-means+LSTM

4) Wild horse AlexNet assisted Bi-directional Gated Recurrent Unit models: The proposed Wild Horse Optimization (WHO) is the Meta heuristic algorithm inspired by wild horses. The wild horse optimization method is used to minimize the dimensionality of huge data and enhance classification accuracy. It aids in reducing the number of input variables in order to increase the efficiency of the suggested model. To reduce the calculation's complexity, the best characteristics are chosen from the pre-processed photos. The optimal features are chosen based on accurate fitness functions, and the mathematical expression is given in the following equation.

$$Fitness = \delta * f + (1-\mu) * \frac{|M_p|}{|M_q|} \quad (15)$$

Here, the parameter affecting the intrusion detection results is represented by δ, the error value in classification is denoted by f, and features obtained from pre-processed data is represented by M_p. The total features presented in the dataset are denoted by M_q. This WHO optimization contains five stages, which are described as follows.

The first stage is to generate the initial population, in this stage, the population of input features are initialized, and the random population is denoted by $(y) = \{y_1, y_2, \ldots, y_m\}$. The random population of features continuously evaluates the target values , which are represented by $(\vec{s}) = \{S_1, S_2, \ldots, S_n\}$. The amount of features and amount of groups in the population is expressed by N and $H = \lceil N \times T_a \rceil$. Here, the search agent percentage is represented by T_a. Optimal feature search agent is chosen as a leader in the WHO approach; based on this optimal search agent, the remaining agent explores essential features. The second stage is the grazing strategy; the leader is placed in the middle of the exploration region. The ideal search agent assumes the role of a central leader positioned within the feature exploration space. This is expressed mathematically in the following equation.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

$$\vec{Y}_{s,H}^{r} = 2\alpha\cos(2\pi r\alpha)\times\left(Lead\ Searchagent^{s} - Y_{s,H}^{r}\right) + Lead\ searchagent^{r} \quad (16)$$

Here, the position of the group member and the search agent position is represented by $Y_{s,H}^{r}$ and $Lead\ searchagent^{s}$. The adaptive mechanism is represented by α, and updated position of search agent during the grazing strategy is represented as $\vec{Y}_{s,H}^{r}$. The uniform distribution is denoted by α, and it ranges from -2 to 2. The third stage is the mating strategy; the particular features are equated with other group of features to identify optimal feature easily. The mathematical expressions are given in the following equation.

$$Y_{H,B}^{a} = Crossover\left(Y_{H,r}^{l}, Y_{H,s}^{z}\right) \quad r \neq s \neq b, a = l = end \quad (17)$$
$$Crossover = mean$$

Here, the position of the feature a from the group b is represented as $Y_{H,B}^{a}$. The fourth stage is the group leadership strategy; this strategy is used to guide the other group members. The leading search agents select features that provide more information to detect intrusions. This strategy is mathematically expressed in the following equation.

$$\overline{Goodsearchagent_{H_r}} = \begin{cases} 2\alpha\cos(2\pi r\alpha)\times\left(\mu - Goodsearchagent_{H_r}\right) + \mu & if\ r_3 > 0.5 \\ 2\alpha\cos(2\pi r\alpha)\times\left(\mu - Goodsearchagent_{H_r}\right) - \mu & if\ r_3 \le 0.5 \end{cases} \quad (18)$$

$$Goodsearchagent_{H_r} = \begin{cases} Y_{H,r} & if\ \cos t\left(Y_{H,r}\right) < \left(Goodsearchagent_{H_r}\right) \\ Goodsearchagent_{H,r} & if\ \cos t\left(Y_{H,r}\right) > \left(Goodsearchagent_{H_r}\right) \end{cases} \quad (19)$$

Here, the upcoming position of the lead search agent of a group *r* is represented as $\overline{Goodsearchagent_{H_r}}$, the suitable region position is represented as μ, the current position of the lead search agent is denoted by $Goodsearchagent_{H_r}$, and the adaptive mechanism is represented as α.

Finally, the last step is to exchange and select a suitable search agent. Based on the fitness function, the leader is randomly selected from search agent groups. From the above equation, if the other search agent is better than the leader, the leader's position should be changed. The optimal features are then selected and passed on to the classification phase. This classification uses two neural network models, AlexNet and BiGRU, to detect intruders in the cloud environment. In these models, AlexNet is used to reduce the vanishing gradient problem and classification errors. The second model, BiGRU, is used to improve the convergence rate and high classification accuracy. Based on these advantages, these models are used to classify intrusions in cloud environment.

In the first phase, AlexNet is used to detect network intrusions. This type of CNN model includes 1 softmax layer, 5 convolutional layers, 2 fully connected layers and 3 max-pooling layers. The selected features are passed as input to convolutional layer. The input size is then reduced, and the number of calculations in max-pooling layer is minimized. The prominent features are extracted in pooling layer through down-sampling. In all layers, the activation function ReLU is used to train the process, the drop-out layer is used to reduce the overfitting problems, and finally, the output layer is used to identify intrusion into the cloud network.

To identify various attack types, the BiGRU model [29] is integrated with AlexNet at the second level. It is employed to accurately produce classification results by extracting contextual information from input data. This BiGRU can be performed in both forward and reverse directions. This BiGRU is obtained from the updated version of GRU. The most often used Gated Recurrent Neural Network (GRU) is an LSTM that has been simplified. GRU minimizes network parameters, streamlines gating units, and prevents overfitting issues as compared to LSTM. This model can achieve better results in the classification phase with the same number of iterations. The retention and deletion information is determined by GRU, including update gate and reset, which are expressed in the following equation.

$$V_s = \sigma\left(X_y\left[G_s - 1, Y_s\right]\right) \quad (20)$$

$$F_s = \sigma\left(X_F\left[G_s - 1, Y_s\right]\right) \quad (21)$$

$$\tilde{G}_s = \tanh\left(X_G\left[F_s \otimes G_s - 1, Y_s\right]\right) \quad (22)$$

$$G_s = \left(1 - V_s\right)\otimes G_s - 1 + V_s \otimes \tilde{G}_s \quad (23)$$

Here, the update gate and reset gate at the time step S is represented as V_s, and F_s. At the time step S, the hidden layer unit and input for the next step are denoted by $\tilde{G}_s$ and G_s. The hidden layer unit at the previous moment is denoted by $G_s - 1$. This GRU model can read only the data in one direction. Therefore, the BiGRU model is proposed to improve the transfer of data in both forward and backward directions. Two units with the same input and a connection to the same output are found in the hidden layer. By using better learning functions, the time series training can be increased by treating forward and backward time series separately. The classification model's accuracy is increased by using this BiGRU model.

$$U_s = \left[\vec{G} : \bar{G}\right] \quad (24)$$

Here, the forward-gated recurrent units are represented by $\vec{G}$, and the backward-gated recurrent unit is denoted by $\bar{G}$. The softmax is an activation function used in the BiGRU model. This softmax function identifies the types of intrusion, and the mathematical expressions are provided in the following equation.

$$Soft\max\left(x_i\right) = \frac{e^{x_i}}{\sum_{j=1}^{i} e^{z_j}} \quad (25)$$

Here, x_i represents input of a softmax layer and is represented as,

$$x_i = \omega^{i} z + c_i \quad (26)$$

Where, the bias is denoted as c and weight is represented as ω. In the process of training, the weight and bias are determined. Thus, the proposed WABi-GRU model can effectively detect intrusion into the cloud environment. With the use of the flow Directional Algorithm (FDA) technique, the hyperparameters of this suggested model can be adjusted. The loss function is minimized by adjusting the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

hyperparameters. The following equation mentions the fitness function's mathematical expression.

$$Fitness = Min[K_h] \tag{27}$$

$$here, K_h = \frac{1}{A}\sum_{a=1}^{A}(x_p - x_a')^2 \tag{28}$$

Here, the loss function is denoted by K_h , and number of iterations in training set is represented by A . The actual value and the predicted value are represented as x_p *and* x_a' . The layer details of the auto-encoder-based convolutional neural network are described in Figure 5

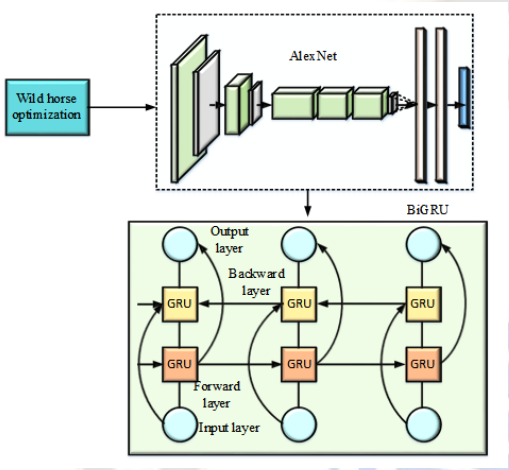

Figure 5: Schematic architecture of WABi-GRU

This schematic architecture explains the classification of intrusion detection in the cloud environment.

5) AlexNet Bi-GRU model : The AlexNet model can be divided into multiple levels and different layers. These layers can be used for different levels of abstraction. Three datasets are used here: CICDS 2018, UNSW-NB 15 datasets and NSL-KDD, which are used to classify intrusion in cloud environment. These data sets act as training data sets, which can obtain original data according to data collection models. The cloud environment's intrusion and assault types are classified by AlexNet using convolutional, max-pooling, fully connected, and softmax layers. The activation function ReLu can be used here to simplify the training process. The input size can be reduced by the max-pooling layer, and the calculations can be reduced. The second model is a Bi-GRU model and a kind of RNN model. The attack types are determined by this hybrid model that combines Bi-GRU and AlexNet. In Figure 6, the AlexNet Bi-GRU model is displayed.

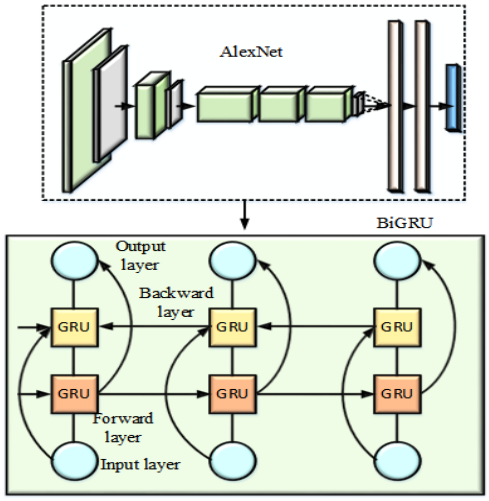

Figure 6: Architecture for AlexNet-Bi-GRU model

The AlexNet-based Bi-GRU model can be used as a classifier model for detecting intrusion in many applications like cloud environments. The Bi-GRU model contains both forward and backward directions and is represented by the following equation.

$$\overline{A}_f = \overrightarrow{GRU}(L_{ft}) \qquad n \in [1, N] \tag{29}$$

$$\overline{A}_b = \overleftarrow{GRU}(L_{ft}), \qquad n \in [N, 1] \tag{30}$$

The input feature sequences are represented by ft , the forward hidden and backward hidden state is denoted by $\overline{A}_f$ and $\overline{A}_b$. The contextual information can be fetched by L_{ft} .

6) Autoencoder based AlexNet Bi-GRU: Based on the popularity of AE, which is used as a suppression mechanism to detect attacks or intrusions in various applications. It contains an input, hidden, and output layer based on unsupervised learning. This Autoencoder has two phases, encoder and decoder, and the most important layer is the coding layer, which is between encoder and decoder. In the classification phase, the BiGRU AlexNet model is used to identify the types of attacks. The schematic architecture for the AE+AlexNet+Bi-GRU model is shown in Figure 7.

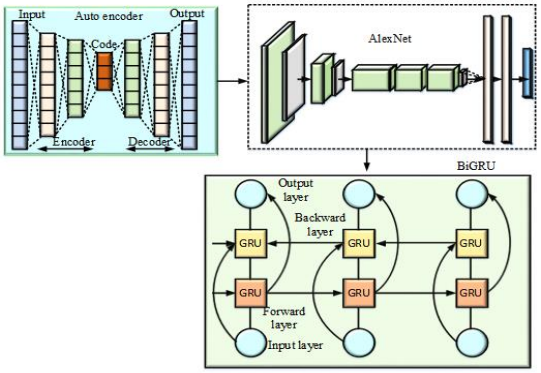

Figure 7: Architecture for AE+AlexNet+Bi-GRU models

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

This architecture explains the autoencoder-based AlexNet BiGRU model for intrusion detection. The AlexNet-based Bi-GRU model can obtain better results in all three datasets, namely CICIDS 2018, NSL-KDD and UNSW-NB 15 dataset. Thus, the intrusion can be detected by the following AlexNet-based BiGRU models. Hyperparameter configuration is explained in the following Table 4.

TABLE IV: HYPERPARAMETER DETAILS FOR PROPOSED MODELS

Methods	**Hyperparameter**	**Details**
Autoencoder	Number of neurons of the first encoded layer	32
	Number of neurons of the second encoded layer	16
	Number of neurons of the third encoded layer	7
	Number of neurons of first decoded layer	16
	Number of neurons of the second decoded layer	32
	First, second, and third layers activation function (E, D)	RELU
	Optimizer	Adam
	Learning rate	0.001
	Loss function	Mean Absolute Error
	Number of epochs	100
	Batch size	10000
CNN	Number of convolutional kernels of the first layer	64
	Size of convolutional kernels	3×3
	Strides size	2
	Number of convolutional layers	1
	Number of fully connected layers	2
	Activation function for convolutional and first fully connected layer	ReLU
	Activation function of the last layer	Sigmoid
	Learning rate	0.001
	Loss function	Binary Cross Entropy (BCE)
	Number of neurons of the first (fully connected) layer	128
	Number of neurons of output (fully connected) layer (CICIDS 2018)	Binary-2 Multi-class- 4
	Drop-out probability	0.5
	Number of epochs	100
	Batch size	10000
	Number of neurons of output (fully connected) layer (NSL-KDD)	5
	Number of neurons of output (fully connected) layer (UNSW-NB 15)	10
LSTM	LSTM layers	128
	Number of fully connected layer	3
	First fully connected layers	64
	First fully connected layers	32
	Final fully connected layers	5
	Activation function of first and second fully connected layers	ReLU

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

	Activation function of final fully connected layers	Softmax
	Optimizer	Adam
	Learning rate	0.001
	Loss function	Mean Absolute Error
	Number of epochs	100
	Batch size	10000
	Drop out	0.5
AlexNet + BiGRU		
	Size of convolutional kernels	3×3
	Strides size	2
	Number of convolutional layers	1
	BiGRU layers	128
	Number of fully connected layer	2
	Activation function of first fully connected layers	ReLU
	Activation function of the last layer	Softmax
	Learning rate	0.001
	Loss function	BCE
	Number of neurons of first fully connected layer	128
	Number of neurons of output (fully connected) layer (CICIDS 2018)	Binary-2 Multi-class- 4
	Number of neurons of output (fully connected) layer (NSL-KDD)	5
	Number of neurons of output (fully connected) layer (UNSW-NB 15)	10
	Drop-out probability	0.5
	Number of epochs	100
	Batch size	10000

The table provides hyperparameter and layer information for the approaches employed in the suggested models.

IV. RESULTS AND DISCUSSION

Several parameters are studied and compared in this part to existing and new models. To demonstrate that the suggested model can improve accuracy and performance on a variety of criteria. Accuracy, Precision, Recall, F1 Score, False Negative Rate (FNR), False Positive Rate (FPR), and False Discovery Rate (FDR) measures are studied and compared to the current and suggested models. These metrics are used for comparison and demonstrate the efficiency of the proposed model. This comparison demonstrates the most effective model for intrusion detection in a cloud setting.

A. Dataset Description

The descriptions are provided for three datasets, namely CICIDS2018, NSL-KDD, and UNSW-NB15.

1) CICIDS2018 dataset: The CICIDS2018 dataset contains seven different types of attacks, namely Denial of Services (DoS), Distributed Dos (DDoS), Bot, DoS attacks-SlowHTTPTest, Benign, DoS attacks Hulk, Heart Bleed, Brute Force, Botnet, APT (Advanced Persistent Threat Detection). , web attacks and infiltration. The attack infrastructure contains 50 machines, 420 machines and 30 servers and organization has 5 departments. It includes network traffic capture; the system logs off each workstation after extracting 80 features from the captured traffic with CICFlowMeter-V3. The distribution of the dataset is described in Table 5.

TABLE V: SAMPLE DISTRIBUTION

Type	**Number of Samples**	
	Training	**Testing**
Benign	646603	161601

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

Bot	86811	21709
DoS attacks-SlowHTTPTest	34	7
DoS attacks-Hulk.	87117	21811

2) NSL-KDD dataset: The KDD Cup 1999 dataset is expanded to include the NSL-KDD, which is utilized as an intrusion detection mechanism. This dataset comprises the following five distinct classes: DoS, normal, R2L, probe, and U2R. The description of the sample classes in the NSL-KDD dataset can be found in Table 6.

TABLE VI: SAMPLE DISTRIBUTION

Type	**Number of records**	
	Training	**Testing**
Normal	67343	9711
Denial of Services	45927	7456
Probe	11656	2421
Remote to Local	995	2756
User to Root	52	200

3) *UNSW-NB15 dataset*: Analysis, backdoors, fuzzes, exploits, dos, generic, shellcode, reconnaissance, and worms are among the nine categories of attacks in the UNSW-NB15 dataset. Bro-IDS tools and Argus are utilized, as well as twelve algorithms, to create a total of 49 features with class names. The UNSW-NB15_features.csv file describes these features. Table 7 describes the distribution of the NSL-KDD data set.

TABLE VII: SAMPLE DISTRIBUTION

Type	**Number of records**	
	Training	**Testing**
Normal	56000	37000
Analysis	2000	677
Backdoor	1746	583
DoS	12264	4089
Exploits	33393	11132
Fuzzers	18184	6062
Generic	40000	18871
Reconnaissance	10491	3496
Shell code	1133	378
Worms	130	44

B. Performance metrics

The following section describes several performance metrics and explains the expressions.

1) Accuracy: Accuracy is defined as the ratio of the total number of precisely noticed anomalous and normal records to all records, which may be stated mathematically in the following equation.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (31)$$

Here, TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative

2) Precision: Precision is used to assess the accuracy of a classification classifier. It is the proportion of anticipated to anomalous records. It can be determined numerically using the equation below.

$$\Pr ecision = \frac{TP}{TP + FP} \quad (32)$$

3) Recall: The completeness or sensitivity of the classifier is measured by recall using the false negative rate. It is the number of right classifications penalized by the number of lost objects, and it may be calculated using the equation below.

$$\mathrm{Re}\, call = \frac{TP}{TP + FN} \quad (33)$$

4) F1-Score: In the F1-score, the harmonic mean of the precision and recall score is calculated and mathematically analyzed using the following equation.

$$F1 - Score = 2 \times \frac{\Pr ecision \times \mathrm{Re}\, call}{\Pr ecision + \mathrm{Re}\, call} \quad (34)$$

5) False Positive Rate (FPR): FPR is the proposition of normal records that were not correctly predicted to anomaly records, and the expression is given below

$$FPR = \frac{FP}{FP + TN} \quad (35)$$

6) Detection Rate (DR): DR is the proportion of correct irregularity records that are correctly identified, and the mathematical expressions are given below.

$$DR = \frac{TP}{TP + FN} \quad (36)$$

7) False Detection Rate (FDR): FDR is defined as the predicted ratio of false positive classifications (false discoveries) to total positive classifications. The following equation expresses the mathematical calculations.

$$FDR = \frac{FP}{(TP + FP)} \quad (37)$$

8) False Negative Rate (FNR): FNR is defined as the ratio of the real number of false negatives to the total number of false negatives, and its expression is given below.

$$FNR = \frac{FN}{(FN + TP)} \quad (38)$$

C. Performance analysis

The performances are analyzed for three datasets: CICIDS 2018, NSL-KDD and UNSW-NB15.

1) Evaluation of the CICIDS 2018 Dataset: Several Performances are analyzed in the following section. Figure 8

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

represents the binary classification analysis for several existing and proposed models.

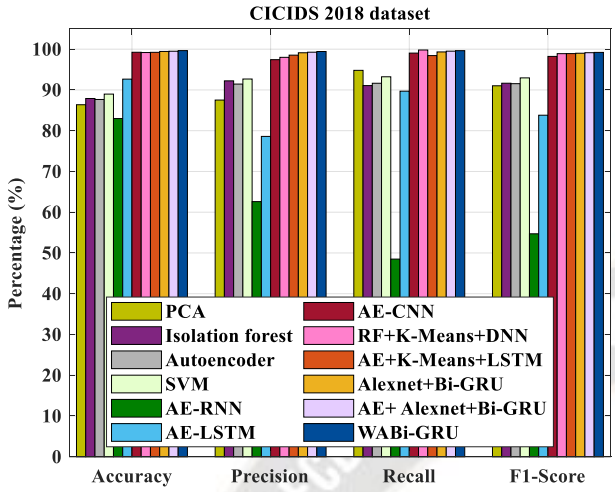

Figure 8: Performance comparison Binary classification

The graph depicts the accuracy, precision, recall, and F1-score measures for existing and proposed models. The existing models, such as PCA, Autoencoder, and AE-RNN, can obtain the accuracy of 86.37%, 87.66% and 82.978%. The proposed models such as AE-CNN, RF+K-means+DNN, AE+K-means+LSTM, Alexnet+Bi-GRU, AE+Alexnet+Bi-GRU and WABi-GRU can attain accuracy values of 99.246%, 99.178%, 99.23%, 99.45%, 99.50% and 99.65%, Precision values of 97.43%, 98% 98.54%, 99.123%, 99.26% and 99.432%, and F1-Score of 98.23%, 98.9%, 98.92%, 99.01%, 99.14% and 99.012%. In this comparison, the AlexNet-based Bi-GRU model can obtain better values than other existing models. Figure 9 shows the analyzed accuracy of existing and proposed models.

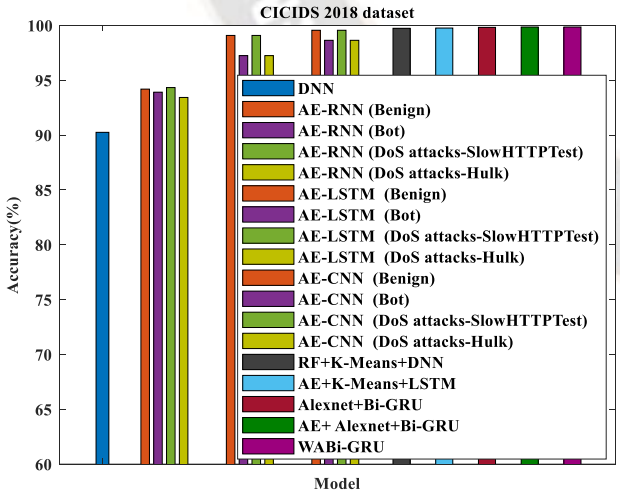

Figure 9: Performance outcomes for multi-class (a) Accuracy.

Figure 9 shows that accuracy for multiple models is analyzed based on multiple class classifications. In this analysis, AlexNet-based models such as Alexnet+Bi-GRU, AE+AlexNet+Bi-GRU and WABi-GRU can achieve 99.819%, 99.852% and 99.890% accuracy. From this comparison, based on multiple models, AlexNet-based Bi-GRU can achieve higher accuracy than other models. Figure 10 (a) and (b) shows FRP and FDR measured for different models.

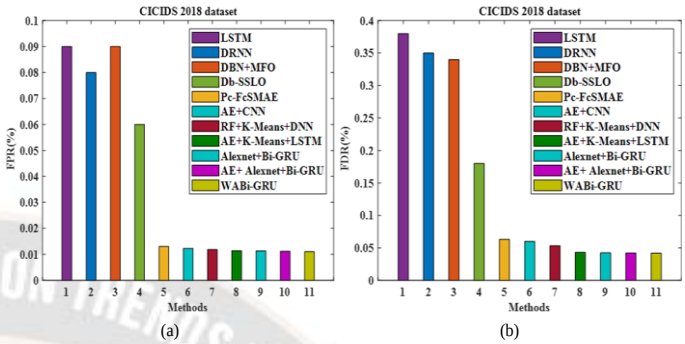

Figure 10: Performance outcomes (a) FPR (b) FDR

Figure 10 (a) shows that the values of the existing models, such as DRNN and Pc-FcSMAE, are analyzed based on the FPR parameter. It can obtain values of 0.080, 0.060 and 0.013. This parameter is analyzed for proposed models to prove which model can perform better. In this analysis, AlexNet-based models such as Alexnet+Bi-GRU, AE+AlexNet+Bi-GRU and WABi-GRU models can obtain FPR values of 0.0113, 0.0112 and 0.0110. Figure 10 (b) shows the FDR analysis for the existing and proposed models. As a result, AlexNet-based Bi-GRU can outperform other existing models. Figure 11 shows the analysis of the FNR rate for several existing and proposed models.

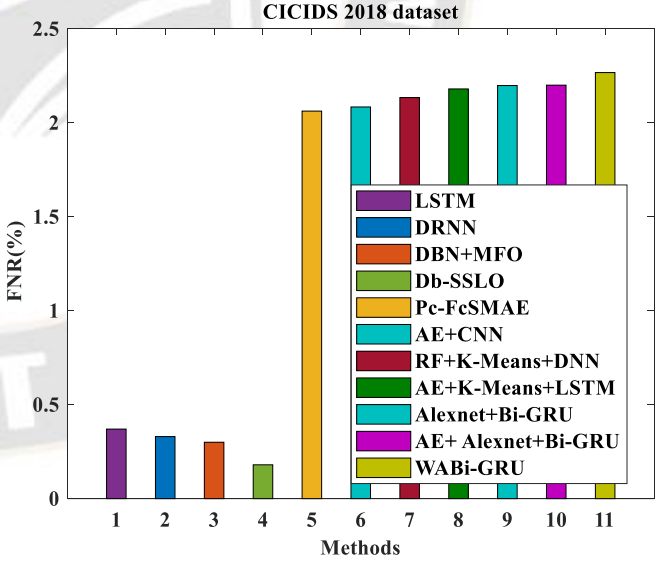

Figure 11: Performance analysis for FNR

The proposed models, such as AE-CNN, RF+K-means+DNN, AE+K-means+LSTM, Alexnet+Bi-GRU, AE+AlexNet+Bi-GRU and WABi-GRU can achieve values of 2.084, 2.134, 2.18, 2,198, 2,200 and 2,267. In this analysis, WABi-GRU can achieve higher values than other proposed models. These comparisons are made to prove which model can perform better. Figure 12 (a), (b) and (c) shows that the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

confusion matrix is calculated for suggested models such as AE+CNN, RF+K-means+DNN, and AE+K-means+LSTM.

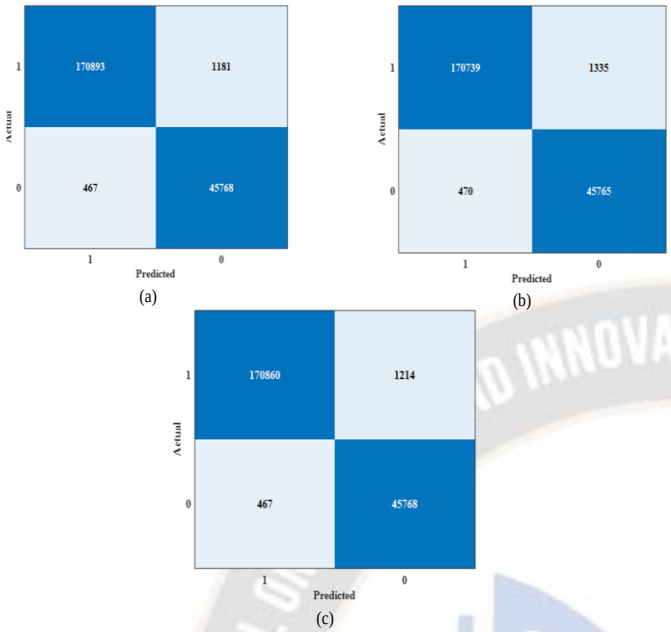

Figure 12: Confusion matrix (a) AE+CNN (b) RF+K-means+DNN (c) AE+K-Means+LSTM

In this Figure, the confusion matrix is evaluated for the proposed models, such as AE+CNN, RF+K-Means and AE+K-Means+LSTM. The error rate can be minimized in the matrix shown. Figure 13 (a), (b) and (c) shows that the confusion matrix is evaluated for the proposed models, such as Alexnet+Bi-GRU, AE+AlexNet+Bi-GRU and WABi-GRU models.

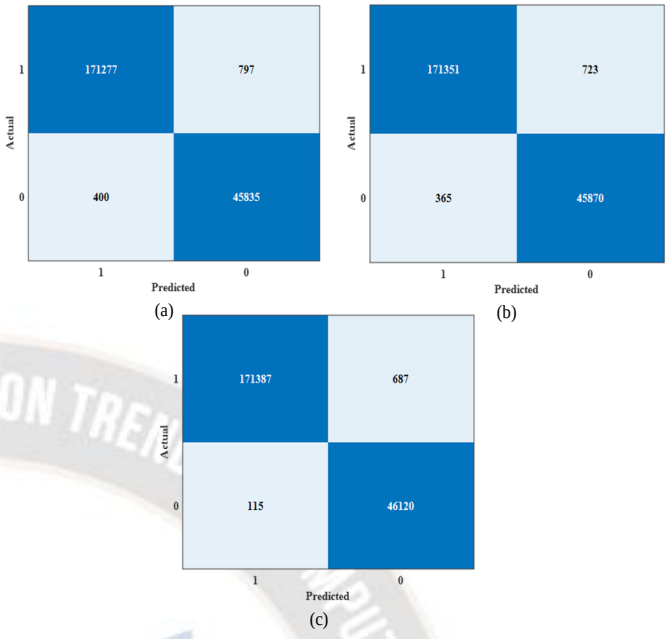

Figure 13: Confusion matrix (a) Alexnet+Bi-GRU (b) AE+AlexNet+Bi-GRU (c) WABi-GRU

In this Figure, the proposed WABi-GRU model can better obtain correct predicted values and fewer unpredicted values. Thus, the Bi-GRU model based on AlexNet can achieve better efficiency than other models and achieve a much lower number of error rates than other models. Table 8 below compares several existing and proposed models based on binary classification.

TABLE VIII: PERFORMANCE COMPARISON FOR BINARY CLASSIFICATION

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-measure (%)**
PCA [30]	86.37	87.52	94.81	91.02
Isolation forest [30]	87.90	92.23	91.07	91.65
Autoencoder [30]	87.66	91.44	91.64	91.54
SVM [30]	88.98	92.68	93.23	92.96
AE-RNN [31]	82.978	62.6	48.5	54.7
AE-LSTM [31]	92.653	78.6	0.897	83.8
AE-CNN [31]	99.246	97.43	99.04	98.23
RF+K-Means+DNN	99.278	0.980	0.998	0.989
AE+K-Means+LSTM	99.33	98.54	98.43	98.92
Alexnet+Bi-GRU	99.45	99.123	99.325	99.01
AE+ Alexnet+Bi-GRU	99.50	99.26	99.513	99.14
WABi-GRU	**99.65**	**99.432**	**99.638**	**99.212**

Accuracy is analyzed for several existing and proposed models for multi-class classification is represented in the following Table 9.

TABLE IX: COMPARISON OF ACCURACY FOR MULTI-CLASS CLASSIFICATION

Model	**Classes**	**Accuracy (%)**
DNN [33]	**All**	**90.25**
AE-RNN [31]	Benign	94.19
	Bot	93.91

	DoS attacks-SlowHTTPTest	94.33
	DoS attacks-Hulk	93.43
AE-LSTM [31]	Benign	99.084
	Bot	97.243
	DoS attacks-SlowHTTPTest	99.084
	DoS attacks-Hulk	97.243
AE-CNN [31]	Benign	99.555
	Bot	98.639
	DoS attacks-SlowHTTPTest	99.555
	DoS attacks-Hulk	98.639
RF+K-Means+DNN[32]	All	99.735
AE+K-Means+LSTM	All	99.756
Alexnet+Bi-GRU	All	99.819
AE+ Alexnet+Bi-GRU	All	99.852
WABi-GRU	All	99.890

The metrics such as recall, precision and F1-score are analyzed for an existing and proposed model, which is in Table 10.

TABLE X: PERFORMANCE COMPARISON OF PRECISION, RECALL AND F1-SCORE

Models	**Classes**	**Precision (%)**	**Recall (%)**	**F1-score (%)**
XGB [33]	All	84.5	83.4	83.9
DT [33]	All	87.33	88.5	87.9
AE-RNN [31]	Benign	0.99	**0.99**	**0.99**
	Bot	0.99	0.95	0.97
	DoS attacks-SlowHTTPTest	0	0	0
	DoS attacks-Hulk	0.96	0.97	0.96
AE-LSTM [31]	Benign	0.97	0.95	0.96
	Bot	0.99	**0.99**	**0.99**
	DoS attacks-SlowHTTPTest	0	0	0
	DoS attacks-Hulk	0.78	0.82	0.80
AE-CNN [31]	Benign	0.98	**0.99**	**0.99**
	Bot	**1.00**	0.98	**0.99**
	DoS attacks-SlowHTTPTest	0	0	0
	DoS attacks-Hulk	0.97	0.87	0.91
RF+K-Means+DNN [32]	All	0.997	0.997	0.997
AE+K-Means+LSTM	All	99.712	99.767	99.798
Alexnet+Bi-GRU	All	99.85	99.799	99.822
AE+ Alexnet+Bi-GRU	All	99.87	99.832	99.854
WABi-GRU	All	99.89	99.865	99.896

FPR, FDR and FNR performances are analyzed for several models and are described in Table 11.

TABLE XI: COMPARISON OF PERFORMANCE ANALYSIS

Methods	**FPR (%)**	**FDR (%)**	**FNR (%)**
LSTM [34]	0.090	0.380	0.370
DRNN [34]	0.080	0.350	0.330
DBN+MFO [34]	0.090	0.340	0.300
Db-SSLO [34]	0.060	0.180	0.180
Pc-FcSMAE [34]	0.013	0.063	2.062
AE+CNN	0.0123	0.060	2.084
RF+K-Means+DNN	0.0118	0.0532	2.134
AE+K-Means+LSTM	0.0114	0.043	2.18
Alexnet+Bi-GRU	0.0113	0.0423	2.198

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

AE+ Alexnet+Bi-GRU	0.0112	0.0420	2.200
WABi-GRU	**0.0110**	**0.0418**	**2.267**

The AUC values, testing and training time for the existing and proposed model are described in Table 12.

TABLE XII: PERFORMANCES ANALYSIS FOR AUC SCORE

Binary Classification				**Multi-class classification**			
Models	**AUC**	**Training time (s)**	**Testing time (s)**	**Models**	**AUC**	**Training time (s)**	**Testing time (s)**
AE+RNN [31]	0.703	816.36	5.78	AE+RNN 31	0.83	1152.964	7.8
AE+LSTM [31]	0.915	1248.943	13.645	AE+LSTM 31	0.845	1308.344	9.938
AE+CNN [31]	0.991	1427.824	7.804	AE+CNN 31	0.852	1335.935	9.303
RF+K-means+DNN	0.993	1290.43	7.034	RF+K-means+DNN	0.902	1234.56	9.021
AE+K-means+LSTM	0.995	1103.54	6.984	AE+K-means+LSTM	0.95	1154.67	8.954
Alexnet+Bi-GRU	0.996	1000.678	6.554	AE+Alexnet+Bi-GRU	0.9643	1132.89	8.673
AE+ Alexnet+Bi-GRU	0.997	998.78	6.234	WABi-GRU	0.97	1102.21	8.435
WABi-GRU	**0.998**	943.87	6.045	**WABi-GRU**	0.985	1043.43	8.213

This table describes the performances such as FRP, FDR, FNR and AUC scores for existing and proposed models. In this comparison examination, the proposed model outperforms other existing models.

2) Evaluation of the NSL-KDD Dataset: Accuracy, Precision, Recall, and F1-Score are performance indicators that are examined for many existing models including GRU, RNN, CNN, LSTM, CNN-LSTM and proposed models like AE+CNN, RF+K-means+DNN, AE+K-Means+LSTM, AlexNet-Bi-GRU, AE+Alexnet+BiGRU and WABi-GRU. The performances are compared and analyzed with proposed and existing models to choose the best method for intrusion detection. Figures 14 (a) and 14(b) represent the accuracy and precision of existing and proposed models.

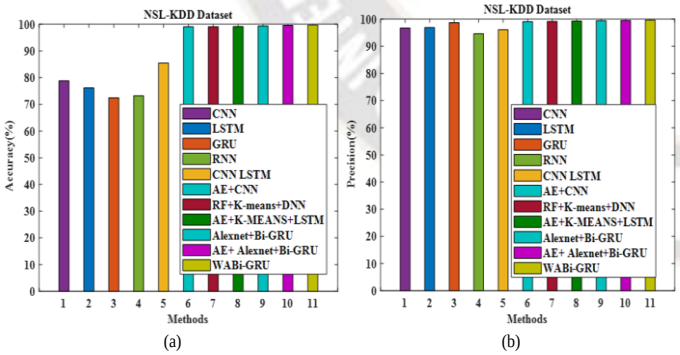

Figure 14: Evaluation of (a) Accuracy (b) Precision

Figure 14 (a) shows that the existing models, such as GRU, RNN and CNN-LSTM, can obtain an accuracy of 72.5%, 73.2% and 85.5%. The proposed models such as AE+CNN, RF+K-means+DNN, AE+K-Means+LSTM, Alexnet+Bi-GRU, AE+ Alexnet+Bi-GRU and WABi-GRU can attain values of 99.03%, 99.092%, 99.137%, 99.34%, 99.546% and 99.7%. Figure 14 (b) represents the precision parameter analyzed for various models. The WABi-GRU can obtain values of 99.63%. Here, the proposed AlexNet-Bi-GRU and AE+Alexnet+BiGRU models are also analyzed. It can obtain values of 99.46% and 99.521%. Thus, AlexNet-based Bi-GRU models can obtain better values than other proposed models. This analysis is performed to prove which proposed models can obtain better accuracy than other models. Figure 15 (a) represents the recall and f1-score analyzed for the existing and proposed models.

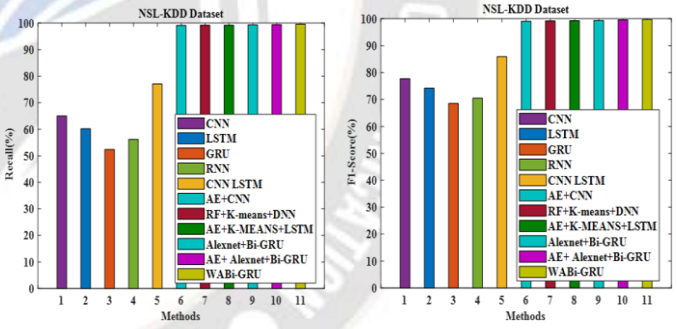

Figure 15: Performance evaluation of (a) Recall (b) F1-Score

Figure 15 (a) shows that the recall metrics are analyzed for existing and proposed models. The existing models like CNN, LSTM and CNN-LSTM can gain recall values of 65%, 60.2% and 77.1%. The model WABi-GRU can achieve recall values of 99.587% more than other proposed models such as AE+CNN, RF+K-means+DNN and AE+K-Means+LSTM. Thus, AlexNet-based Bi-GRU models can obtain better values than other proposed models. Figure 15 (b) represents the analysis of F1-Score for existing and proposed models. In this comparison, the AlexNet-based Bi-GRU model can attain higher values than other proposed models. Figure 16 (a) and (b) represents the comparison of FRP and DR for existing and proposed models.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

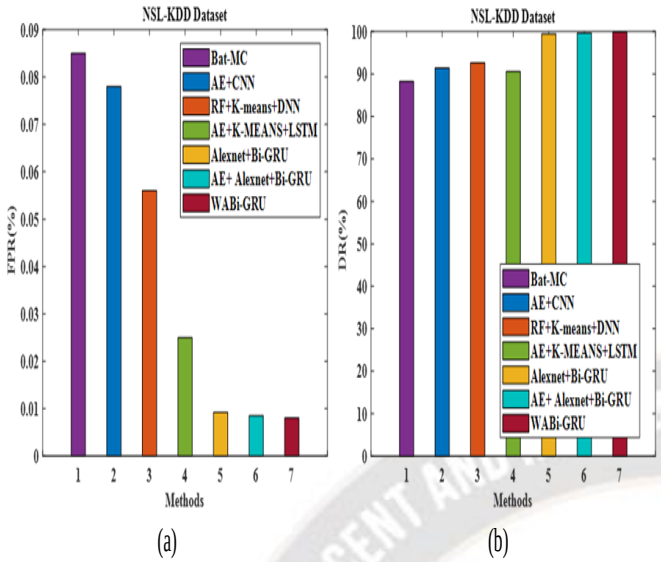

Figure 16: Performance outcomes (a) FPR (b) DR

Figure 16 (a) shows the analysis of FPR metrics for existing and several proposed models. The existing model can obtain the last position by obtaining higher values than other proposed models. From the four proposed models AE+CNN, RF+K-means+DNN, AE+K-Means+LSTM and WABi-GRU, WABi-GRU can obtain less FPR than other models. Figure 16 (b) shows that DR is analyzed for multiple models. In this analysis, AlexNet-based Bi-GRU can attain better values than other models. This comparison can be used to prove that WABi-GRU is more efficient than other models. Figure 17 (a), (b) and (c) shows that the confusion matrix is analysed for the proposed models such as AE+CNN, RF+K-means+DNN, and AE+K-Means+LSTM.

In this Figure, the proposed model can be evaluated under various classes like Normal, DoS, Probe, R2L and U2R. AE-CNN can attain normal class values of 28840, and AE+K-Means+LSTM can obtain values of 29092 in the R2L class. Figure 18 (a), (b) and (c) shows that the confusion matrix is evaluated for the proposed models, such as Alexnet+Bi-GRU, AE+AlexNet+Bi-GRU and WABi-GRU models.

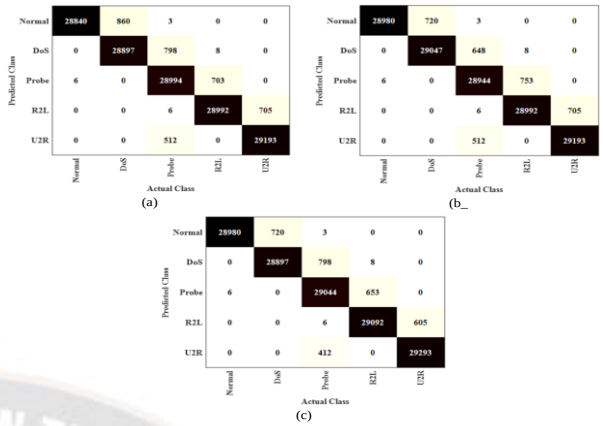

Figure 17: Confusion matrix (a) AE+CNN (b) RF+K-means+DNN (c) AE+K-Means+LSTM

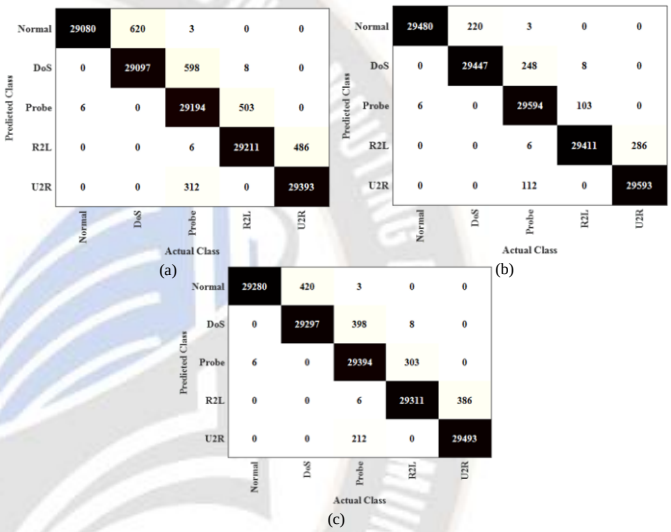

Figure 18: Confusion matrix (a) Alexnet+Bi-GRU (b) AE+AlexNet+Bi-GRU (c) WABi-GRU

In Figure 18, the proposed model WABi-GRU can obtain better correct predicted values and less number of unpredicted values. Thus, the AlexNet-based Bi-GRU model can obtain better efficiency than other models and obtain very low error rates than other models. The performance analyzed for the NSL-KDD dataset is shown in the following Table 13.

TABLE XIII: PERFORMANCE EVALUATION OF THE NSL-KDD DATASET

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
CNN [35]	78.8	96.7	65	77.7
LSTM [35]	76.2	96.9	60.2	74.2
GRU [35]	72.5	98.7	52.4	68.5
RNN [35]	73.2	94.6	56.2	70.5
CNN LSTM [35]	85.5	96.1	77.1	85.9
AE+CNN	99.03	99.09	99.12	99.089
RF+K-means+DNN	99.092	99.14	99.18	99.178

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

AE+K-MEANS+LSTM	99.137	99.299	99.201	99.298
Alexnet+Bi-GRU	99.34	99.46	99.345	99.378
AE+ Alexnet+Bi-GRU	99.546	99.521	99.432	99.567
WABi-GRU	**99.7**	**99.63**	**99.587**	**99.724**

In this comparison, the methods using the AlexNet model can achieve better values than other existing models. In this analysis, only minor differences can be observed between the performances of the Alexnet+Bi-Gru and AE+Alexnet+Bi.Gru and WABi-GRU models. Overall, WABi-GRU proves to be comparatively better than the existing methods. The WABi-GRU model achieved better results due to effective feature learning and better data training in terms of larger epoch size. By effectively setting hyperparameters, the WABi-GRU model achieves 99.7% accuracy, which is better than previous models. The FPR and DR performance is analyzed for existing and proposed models and described in Table 14.

TABLE XIV: PERFORMANCE METRICS COMPARISON

Models	FPR (%)	DR (%)
Bat-MC [36]	0.085	88.23
AE+CNN	0.078	91.4
RF+K-means+DNN	0.056	92.6
AE+K-means+LSTM	0.025	90.56
Alexnet+Bi-GRU	0.0092	99.45
AE+ Alexnet+Bi-GRU	0.0085	99.67
WABi-GRU	0.008	99.87

The AUC scores are analyzed for existing and proposed models using the NSL-KDD dataset shown in Table 15.

TABLE XV: ANALYSIS OF THE AUC SCORE FOR NSL-KDD DATASET

Methods	AUC score
Vanila RNN [37]	0.50
DNN [37]	0.80
GRU-RNN [37]	0.90
AE+CNN	0.914
RF+K-means+DNN	0.94
AE+K-means+LSTM	0.967
Alexnet+Bi-GRU	0.9721
AE+ Alexnet+Bi-GRU	0.9799
WABi-GRU	**0.989**

From the tables above, it can be seen that the analysis of existing models may achieve lower performance than the WABi-GRU model. This performance comparison helps select the efficient intrusion detection model to achieve better results.

3) Evaluation of UNSW-NB15 dataset: The performance metrics such as accuracy, precision, recall, f1-score, FPR and confusion matrix are analyzed for the proposed models. Figure 19 (a) and (b) represent the accuracy and precision that are analyzed and compared with existing and proposed models.

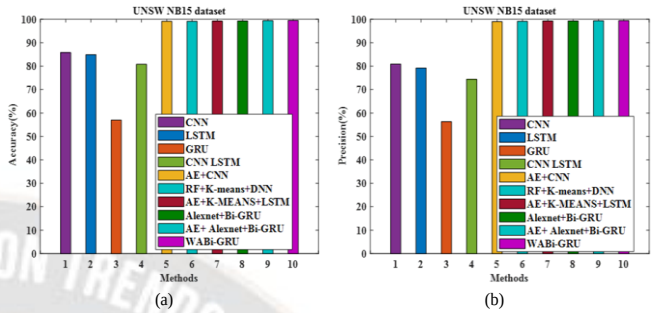

Figure 19: Performance results (a) Accuracy (b) Precision

Figure 19 (a) shows that the accuracy is measured for existing models such as CNN, LSTM and GRU. It can obtain an accuracy of 85.8%, 84.9% and 57%. The proposed model like AE+CNN, RF+K-means+DNN, AE+K-Means+LSTM, Alexnet+Bi-GRU, AE+Alexnet+Bi-GRU and WABi-GRU are analyzed and obtained the accuracy of 99.128%, 99.145%, 99.247%, 99.313%, 99.399% and 99.53%. In this comparison, AlexNet-based Bi-GRU models can attain better accuracy than other models. Figure 19 (b) represents that the precision is analyzed for several models, such as AE+CNN, RF+K-means+DNN, AE+K-Means+LSTM and WABi-GRU. In these models, WABi-GRU can obtain precision values of 99.45% than other models. Figure 20 (a) and (b) represent recall, and the f1-score is evaluated for several models.

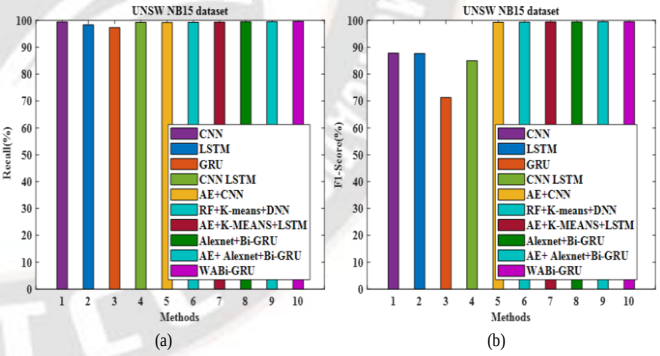

Figure 20: Evaluation of (a) Recall (b) F1-Score

Figure 20 (a) shows the analysis of the recall parameter for various existing and proposed models. The existing models can obtain fewer values than the proposed models. WABi-GRUI can obtain first rank position based on the performances. Figure 20 (b) shows that the f1-score is analyzed for several models. In this analysis, WABi-GRU can obtain values of 99.50% than other proposed and existing models. In this analysis, AlexNet-based Bi-GRU models can obtain better results than other existing ones. Figure 21 (a), (b) and (c) shows that the confusion matrix for the proposed models such as AE+CNN, RF+K-means+DNN, and AE+K-Means+LSTM.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

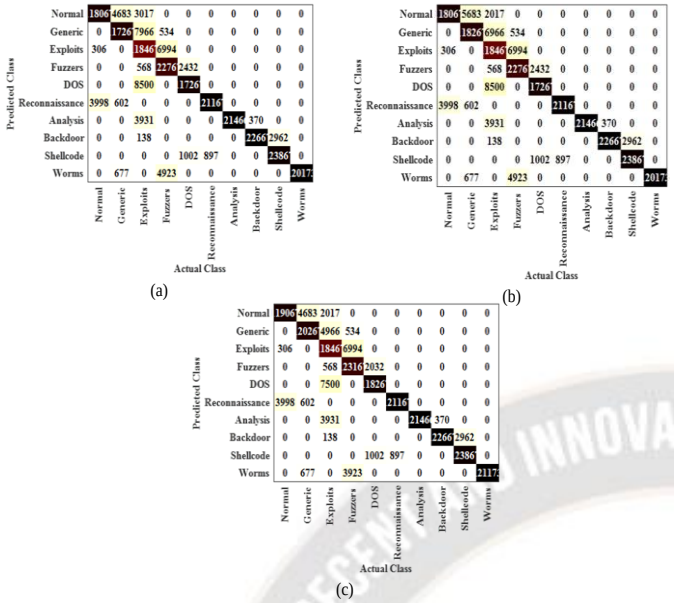

Figure 21: Confusion matrix (a) AE+CNN (b) RF+K-means+DNN (c) AE+K-Means+LSTM

In this Figure, the confusion matrix is evaluated for the proposed models, such as AE+CNN, RF+K-Means and AE+K-Means+LSTM. The classes are accurately predicted when compared to other existing models. Figure 22 (a), (b) and (c) shows that the confusion matrix is evaluated for the proposed models, such as Alexnet+Bi-GRU, AE+AlexNet+Bi-GRU and WABi-GRU models.

In Figure 22, the proposed model WABi-GRU can obtain better correct predicted values and less number of unpredicted values. Thus, the AlexNet-based Bi-GRU model can obtain better efficiency and also obtain very low error rates than other models. The performance evaluation for various models is described in the following Table 16.

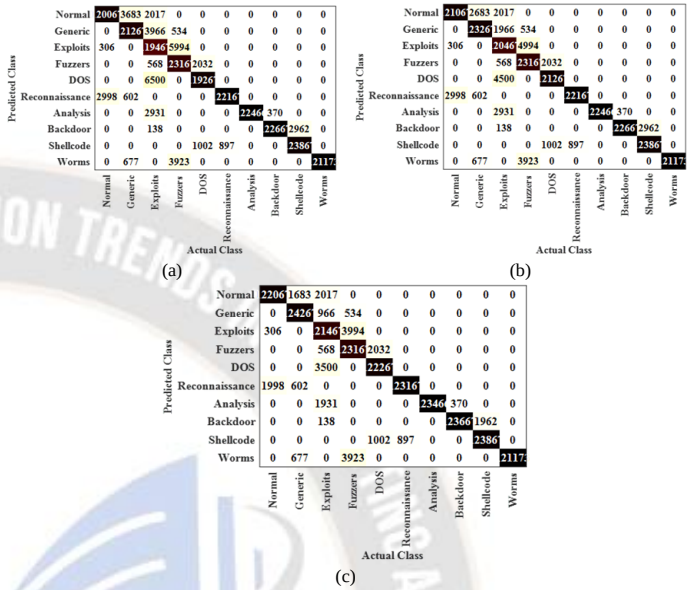

Figure 22: Confusion matrix (a) Alexnet+Bi-GRU (b) AE+AlexNet+Bi-GRU (c) WABi-GRU

TABLE XVII: PERFORMANCE COMPARISON FOR EXISTING AND PROPOSED MODELS USING UNSW-NB 15 DATASET

Methods	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
CNN [35]	85.8	80.9	99.4	87.8
LSTM [35]	84.9	79.2	98.3	87.7
GRU [35]	57	56.3	97.3	71.3
CNN LSTM [35]	80.8	74.4	99.3	85
AE+CNN	99.128	99.012	99.23	99.298
RF+K-means+DNN	99.145	99.12	99.298	99.323
AE+K-means+LSTM	99.247	99.278	99.34	99.396
Alexnet+Bi-GRU	99.313	99.312	99.489	99.401
AE+ Alexnet+Bi-GRU	99.399	99.391	99.501	99.486
WABi-GRU	**99.53**	**99.45**	**99.67**	**99.50**

The parameter FPR is analyzed for an existing and proposed model, which is given in the Table 17.

TABLE XVIII: COMPARISON OF VARIOUS EXISTING AND PROPOSED MODELS.

Methods	**FPR (%)**
RNN [38]	56.38
LSTM [38]	40.02
DNN [38]	40.02
SAE-DNN [38]	0.17
AE+CNN	0.15
RF+K-means+DNN	0.13
AE+K-means+LSTM	0.12
Alexnet+Bi-GRU	0.11
AE+ Alexnet+Bi-GRU	0.10

WABi-GRU	**0.09**

This table is used to analyze the metrics and display the efficient intrusion detection model. In analysis and comparison of different existing and proposed models for identifying intrusion in cloud environment using three different datasets namely, CICIDS 2018, NSL-KDD and UNSW-NB15 dataset. From this analysis, the proposed models such as AE+CNN, RF+K-means+DNN, AE+K-Means+LSTM, Alexnet+Bi-GRU AE+Alexnet+Bi-GRU and WABi-GRU are analyzed with several performance metrics. The proposed models can obtain better accuracy and develop an efficient model by increasing the epoch values in the training dataset. By optimizing the hyperparameters in the model, the WABi-GRU model can achieve higher accuracy than the previous model. Based on this comparison, the WABi-GRU model can attain better accuracy and enhanced performances than other models. Thus, the WABi-GRU model is efficient for intrusion detection.

V. CONCLUSION

This paper compares the performance of classification models for efficient intrusion detection systems. The effective model was demonstrated in the following stages: In the first phase, data cleaning is performed, and in the pre-processing phase, data normalization is performed to remove unwanted noise from the data. The next step is to use the four models to compare and prove which is better for detection. The proposed models, such as RF+K-means+DNN, AE+K-Means+LSTM, AlexNet Bi-GRU, AE+Alexnet+Bi-GRU and WABi-GRU can obtain an accuracy of 99.278%, 99.33%, 99.45%, 99.50%, 99.65% for the CICIDS dataset 2018 for binary classification. In multi-class classification, the AlexNet Bi-GRU, AE+Alexnet+Bi-GRU and WABi-GRU can attain accuracy of 99.819%, 99.852% and 99.890%. In NSL-KDD, the AlexNet Bi-GRU, AE+Alexnet+Bi-GRU and WABi-GRU achieve accuracy of 99.34%, 99.546% and 99.7%. In the UNSW-NB 15 dataset, the AlexNet Bi-GRU, AE+Alexnet+Bi-GRU and WABi-GRU achieve accuracy of 99.313%, 99.399% and 99.53%. AlexNet Bi-GRU-based models can obtain better performances than other existing models. The other performance is also analyzed, such as FPR, FDR, DR and compared with existing models. In this analysis, AlexNet Bi-GRU based models can obtain better results when compared to other existing models. From this comparison, small changes are noticed in the AlexNet Bi-GRU, AE+Alexnet+Bi-GRU and WABi-GRU models. There is no huge variation in accuracy among AlexNet Bi-GRU-based models. In the future, the proposed work can be improved by creating the model in a real-time cloud environment, such as VMware workstation, to test its reliability. Furthermore, the model's performance can be increased by utilizing hyperparameter tuning technologies.

ACKNOWLEDGMENT

None

REFERENCES

[1] A. Aldallal, and F. Alisa. “Effective intrusion detection system to secure data in cloud using machine learning”. Symmetry Vol. 13(12), 2306, 2021.

[2] A. Thangasamy, B. Sundan, and L. Govindaraj. “Dynamic phad/ahad analysis for network intrusion detection and prevention system for cloud environment”. In 2021 4th International Conference on Computing and Communications Technologies (ICCCT), pp. 273-279, 2021. IEEE.

[3] A. Aldallal. “Toward efficient intrusion detection system using hybrid deep learning approach”. Symmetry Vol. 14(9), 1916, 2022.

[4] A. Javaid, Q. Niyaz, W. Sun, and M. Alam. “A deep learning approach for network intrusion detection system”. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), pp.21-26, 2016.

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

[6] G. Karatas, O. Demir, and O. K. Sahingoz. “Deep learning in intrusion detection systems”. In 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), pp.113-116, 2018. IEEE.

[7] K. Alrawashdeh, and C. Purdy. “Toward an online anomaly intrusion detection system based on deep learning”. In 2016 15th IEEE international conference on machine learning and applications (ICMLA), pp.195-200, 2016. IEEE.

[8] N. T. Van, and T. N. Thinh. “An anomaly-based network intrusion detection system using deep learning”. In 2017 international conference on system science and engineering (ICSSE), pp.210-214, 2017. IEEE.

[9] M. Mayuranathan, M. Murugan, and V. Dhanakoti. “Best features based intrusion detection system by RBM model for detecting DDoS in cloud environment”. Journal of Ambient Intelligence and Humanized Computing Vol. 12, 3609-3619, 2021.

[10] J. K. Samriya, and N. Kumar. “A novel intrusion detection system using hybrid clustering-optimization approach in cloud computing”. In Materials Today: Proceedings, Vol. 2(1), pp.23-54, 2020.

[11] S. M. Kasongo. “A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework”. Computer Communications Vol.199, pp.113-125, 2023.

[12] W. Wang, X. Du, D. Shan, R. Qin, and N. “Wang. Cloud intrusion detection method based on stacked contractive auto-encoder and support vector machine”. IEEE transactions on cloud computing Vol. 10(3), pp.1634-1646, 2020.

[13] J. Gao. “Network intrusion detection method combining CNN and biLSTM in cloud computing environment”. Computational Intelligence and Neuroscience 2022, 2022.

[14] R. Nayak, M. M. Behera, U. C. Pati, and S. K. Das. “Video-based real-time intrusion detection system using deep-learning for smart city applications”. In 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), pp.1-6, 2019. IEEE.

[15] K. N. Rao, K. Venkata Rao, and P. Reddy PVGD. “A hybrid intrusion detection system based on sparse autoencoder and deep neural network”. Computer Communications Vol.180, pp.77-88, 2021.

[16] H. Attou, M. Mohy-eddine, A. Guezzaz, S. Benkirane, M. Azrour, A. Alabdultif, and N. Almusallam. “Towards an Intelligent Intrusion Detection System to Detect Malicious

Activities in Cloud Computing". Applied Sciences Vol. 13(17), p.9588, 2023.

[17] S. A. Nallamuthu. "A hybrid genetic-neuro algorithm for cloud intrusion detection system". Journal of Computational Science and Intelligent Technologies Vol. 1(2), pp.15-25, 2020.

[18] P. K. Keserwani, M. C. Govil, and E. S. Pilli. "An optimal intrusion detection system using GWO-CSA-DSAE model". Cyber-Physical Systems Vol. 7(4), pp.197-220, 2021.

[19] S. S. Chakravarthi, R. Jagadeesh Kannan, V. A. Natarajan, and X. Z. Gao. "Deep Learning Based Intrusion Detection in Cloud Services for Resilience Management". Computers, Materials & Continua Vol. 71(3), 2022.

[20] P. Ghosh, D. Sarkar, J. Sharma, and S. Phadikar. "An intrusion detection system using modified-firefly algorithm in cloud environment". International Journal of Digital Crime and Forensics (IJDCF) Vol. 13(2), pp.77-93, 2021.

[21] K. Sethi, R. Kumar, N. Prajapati, and P. Bera. "Deep reinforcement learning based intrusion detection system for cloud infrastructure". In 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS), pp.1-6, 2020. IEEE.

[22] A. Abusitta, M. Bellaiche, M. Dagenais, and T. Halabi. "A deep learning approach for proactive multi-cloud cooperative intrusion detection system". Future Generation Computer Systems Vol. 98, pp.308-318, 2019.

[23] V. Balamurugan, and R. Saravanan. "Enhanced intrusion detection and prevention system on cloud environment using hybrid classification and OTS generation". Cluster Computing Vol. 22(6), pp.13027-13039, 2019.

[24] M. B. Umair, Z. Iqbal, M. A. Faraz, M. A. Khan, Y. D. Zhang, N. Razmjooy, and S. Kadry. "A network intrusion detection system using hybrid multi-layer deep learning model". Big data 2022.

[25] A. S. Almogren, "Intrusion detection in Edge-of-Things computing". Journal of Parallel and Distributed Computing Vol. 137, pp.259-265, 2020.

[26] V. Prabhakaran, and A. Kulandasamy. "Hybrid semantic deep learning architecture and optimal advanced encryption standard key management scheme for secure cloud storage and intrusion detection". Neural Computing and Applications Vol. 33(21), pp.14459-14479, 2021.

[27] J. Zhu, L. Huo, M. D. Ansari, and M. A. Ikbal. "Research on data security detection algorithm in IoT based on K-means". Scalable Computing: Practice and Experience Vol. 22(2), pp.149-159, 2021.

[28] M. D. Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi. "LSTM-based intrusion detection system for in-vehicle can bus communications". IEEE Access Vol. 8, pp.185489-185502, 2020.

[29] D. Javeed, T. Gao, P. Kumar, and A. Jolfaei. "An Explainable and Resilient Intrusion Detection System for Industry 5.0". IEEE Transactions on Consumer Electronics 2023.

[30] M. Verkerken, L. D'hooge, T. Wauters, B. Volckaert, and F. D. Turck. "Towards model generalization for intrusion detection: Unsupervised machine learning techniques". Journal of Network and Systems Management Vol. 30, pp.1-25, 2022.

[31] R. Bingu, S. Jothilakshmi, and N. Srinivasu. "An intelligent multi-class deep classifier-based intrusion detection system for cloud environment". Concurrency and Computation: Practice and Experience: e7840.

[32] R. Bingu, S. Jothilakshmi, and N. Srinivasu. "Design of Intrusion Detection System using Ensemble Learning Technique in Cloud Computing Environment". International Journal of Advanced Computer Science and Applications Vol. 14(5), 2023.

[33] M. A. Khan. "HCRNNIDS: hybrid convolutional recurrent neural network-based network intrusion detection system". Processes. Vol. 9 (5), 834, 2021.

[34] R.M. Balajee, and J. Kannan MK. "Intrusion detection on AWS cloud through hybrid deep learning algorithm". Electronics Vol. 12(6), 1423, 2023.

[35] A. Meliboev, J. Alikhanov, and W. Kim. "Performance evaluation of deep learning based network intrusion detection system across multiple balanced and imbalanced datasets". Electronics Vol. 11(4), 515, 2022.

[36] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li. "BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset". IEEE Access Vol. 8, pp.29575-29585, 2020.

[37] T. A. Tang, L. Mhamdi, D. McLernon, S. A. Raza Zaidi, and M. Ghogho. "Deep recurrent neural network for intrusion detection in sdn-based networks". In 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), pp.202-206, 2018. IEEE.

[38] K. N. Rao, K. V. Rao, and P. Reddy PVGD. "A hybrid intrusion detection system based on sparse autoencoder and deep neural network". Computer Communications Vol. 180, pp.77-88, 2021.