Computers, Materials & Continua
DOI: 10.32604/cmc.2023.032617
Article

A Fused Machine Learning Approach for Intrusion Detection System

Muhammad Sajid Farooq[1], Sagheer Abbas[1], Atta-ur-Rahman[2], Kiran Sultan[3], Muhammad Adnan Khan[4,*] and Amir Mosavi[5,6,7]

[1]School of Computer Science, National College of Business Administration and Economics, Lahore, 54000, Pakistan
[2]Department of Computer Science, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia
[3]Department of CIT, The Applied College, King Abdulaziz University, Jeddah, 21589, Saudi Arabia
[4]Department of Software, Gachon University, Seongnam, 13120, Korea
[5]John von Neumann Faculty of Informatics, Obuda University, Budapest, 1034, Hungary
[6]Institute of Information Engineering, Automation and Mathematics, Slovak University of Technology in Bratislava, Bratislava, 81107, Slovakia
[7]Faculty of Civil Engineering, TU-Dresden, Dresden, 01062, Germany
*Corresponding Author: Muhammad Adnan Khan. Email: adnan@gachon.ac.kr
Received: 23 May 2022; Accepted: 24 June 2022

Abstract: The rapid growth in data generation and increased use of computer network devices has amplified the infrastructures of internet. The interconnectivity of networks has brought various complexities in maintaining network availability, consistency, and discretion. Machine learning based intrusion detection systems have become essential to monitor network traffic for malicious and illicit activities. An intrusion detection system controls the flow of network traffic with the help of computer systems. Various deep learning algorithms in intrusion detection systems have played a prominent role in identifying and analyzing intrusions in network traffic. For this purpose, when the network traffic encounters known or unknown intrusions in the network, a machine-learning framework is needed to identify and/or verify network intrusion. The Intrusion detection scheme empowered with a fused machine learning technique (IDS-FMLT) is proposed to detect intrusion in a heterogeneous network that consists of different source networks and to protect the network from malicious attacks. The proposed IDS-FMLT system model obtained 95.18% validation accuracy and a 4.82% miss rate in intrusion detection.

Keywords: Fused machine learning; heterogeneous network; intrusion detection

1 Introduction

During the last few years, rapidly increasing network data transfer has created significant complications in network data management, which may lead to network intrusion in the future [1]. Anti-virus software, Intrusion detection methods and firewalls are main cyber security techniques

and such techniques are responsible to protect the net-work from external and internal attacks [2]. Various intrusion detection techniques are being used, but these methods face problems due to the high-speed of the networks and unidentified attacks. As a result, an effective network attack detection mechanism is required. Different intrusion detection systems based on machine learning methodologies have recently become popular [3]. Machine learning has an advanced branch called deep learning that consisted of multiple layers that indicate the learning process [4]. On the other hand deep learning techniques have been considered by some researchers as a way to develop more effective and efficient intrusion detection systems than the present machine learning methodologies [5]. The authors of [6] developed a deep learning-based solution for detecting network attacks and monitoring flow computations. A real-time detection method based on traffic calculations and common patterns, as well as a classification method based on Deep belief networks and support vector machines (DBN-SVM). The DBN-SVM approach was applied to improve classification accuracy, and real-time detection was achieved via Sliding window (SW) stream data processing.

A system is based on the CICDS2017 open-source dataset was used to verify the suggested method by implementing a series of experiments. The proficiency and real-time competency of the proposed method were greater than those of other typical machine learning algorithms.

In [7], the authors introduced an intrusion detection scheme that consists of Apache Spark to detect malicious traffic. A stack auto encoder network was applied to extract features, and a support vector machine was used as a classifier. Huang et al. [8] addressed intrusion detection system problems by proposing a novel model based on an extreme learning machine. The extreme learning machine consisted of a single hidden layer based on a feed forward neural network that used random input weights, a bias for the hidden layer and output weights. Another study [9] was based on a fast learning network that consisted of a single-layer feed-forward neural network and a multilayer feed-forward neural network.

Ali et al. [10] presented an anomaly detection system that combined a quick learning network with particle swarm optimization. The findings revealed that the proposed mod-el outperformed other algorithms in terms of accuracy.

The content of this article is organized in the following manner. The second section provides a quick overview of the relevant studies. The suggested detection framework is detailed in Section 3. The experimental data and analysis are presented in Section 4 and summarize the study with limitations and future work.

2 Related Work

The excessive use of the internet and other communication technologies has become a great challenge for the network and it creates the need to secure from malicious traffic [11]. Various intrusion detection systems have been offered to enhance malicious traffic detection in heterogeneous networks. An efficient technique is presented in [12] based on a hybrid network, namely, RULA-intrusion detection system (RULA-IDS), for intrusion detection. RULA-IDS performed intrusion detection at a large scale of statistical data in the network traffic. The global attention mechanism layer, completely linked layer, feature extraction layer, and Support vector machine (SVM) classification layer are normally four layers. The temporal and spatial characteristics in network traffic attributes were extracted using the feature extraction layer, so the global attention layer was used to preserve essential information from the components. Finally, an SVM classifier was applied for output for the RULA-IDS system.

Similarly, another study [13] presented a novel system for intrusion detection based on weighted class classification. The intrusion detection scheme consisted of two parts. In the first part, the supervised Machine learning (ML) algorithm worked on the past information in the network. The ML algorithm created a classifier that segregated the investigated attacks, and these decisions were stored in a database. Second, a specifically designed iterative algorithm was used to increase the accuracy of detectable attacks and enhance class' weights. This system improved the performance and maximized the number of correctly discernible types. The University of new south wales network (UNSW) dataset was utilized to analyse the system's performance.

In [14], the researchers presented a model based on the Stacked contracted auto encoder (SCAE) approach and SVM classification algorithm. The deep learning approach was applied to extract features automatically and evaluate the performance. The SCAE method was used for both low-dimensional features and high-dimensional features of the network traffic. The SVM and SCAE techniques combined shallow and deep learning approaches and helped decrease the systematic overhead significantly. The proposed model used the NSL-KDD and KDD Cup 99 datasets, and the analysis of the experiments demonstrated that the proposed model obtained 95% accuracy. The authors [15] explored data set characteristics and evaluated cyber security data set characteristics for the detection of anomalies in the network.

The detailed analysis of numerous detection techniques based on machine learning technologies was conducted to examine or defend network traffic from malicious assaults [16]. In order to reduce computational complexity during detection performance, a novel multistage optimized machine learning-based network intrusion detection system was introduced. The effect of the oversampling strategy on the training sample size of the proposed model was investigated in order to find the minimum suitable training sample size. Moreover, the temporal complexity and detection effectiveness of two feature selection algorithms based on information gain and correlation were examined. Moreover, various machine learning hyper parameter optimization methods were examined to increase the performance of the proposed system.

Deep and shallow learning strategies were both used in the particular learning approach for intrusion detection systems [17]. However, these approaches encountered various problems in recognizing complicated intrusion patterns. In a lack no of samples, the single deep learning model was not useful for detecting intrusion attacks. In order to enhance the act of a machine learning-based intrusion detection system, the researchers proposed a Big data-based hierarchical deep learning system (BDHDLS). The suggested model used content and behaviour based features to recognize information in the payload and network traffic characteristics. Each model in the BDHDLS focused on learning a characteristic data circulation in one cluster. This approach improved the detection ratio of intrusion attacks in the network. In another study [18] introduced two stages hierarchical hybrid approach for intrusion detection in IoT scenarios. In the first stage, a multimodal deep auto encoder is applied for the anomaly detection, and in the second stage soft-output classifiers are used for attack classification.

With extensive efforts to enhance the accuracy of an intrusion detection system, the authors in [19] described that the rapid increase in data and excess usage of the internet created a need to segregate network traffic in the form of routine flow and anomalies. Therefore, intrusion detection systems with a machine learning approach could play a dynamic role in protecting national security and the economy. Moreover, the authors presented a Convolutional neural network and long short-term memory network (CNN-LSTM) model for intrusion detection. The study normalized spatial feature learning through UTF-8 character encoding and extracted the features from real-time Hypertext

transfer protocol (HTTP) traffic without compression, encryption, and entropy calculations. Fixed real-time data and two CSIC-2010 and CICIDS2017 public datasets were used to train and validate the model. During the training phase, the network traffic was analyzed and labelled as true or false. As a result, artificially intelligent intrusion detection systems separated unknown patterns and obfuscated attacks from network traffic.

The system is calculated the intrusion probability through continuous training that lead to accurate analysis of malicious traffic. Since thresh holding and deviation play an essential role in intrusion detection, the authors in [20] introduced a novel distance function that could detect similarity in two patterns. The dimension of features is a significant issue in machine learning. A feature transformation technique was used to reduce the number of features through the Gaussian distance function in the proposed model. The new computation expression was applied to determine the threshold and deviation in Gaussian space. The NSL-KDD and KDD datasets were used for training and validation purposes. Since K-fold cross-validation is used in machine learning models to collect attractive performance parameters such as accuracy, recall, and precision, the performance evaluation was based on these metrics in the aforementioned research; which demonstrated that the feature transformation technique was better. Feature engineering hitches and low accuracy in intrusion detection systems are the leading causes of obtaining the unwanted results. To solve these issues, the study in [21], the authors proposed a model to detect network traffic anomalies and to discover and fix the issues with feature engineering and low intrusion detection accuracy. Bidirectional long short-term memory (BLSTM) and an attention mechanism were combined in the suggested framework. The major goal of the attention method was to manage packet vector network flow. In order to capture the characteristics of network traffic data, many convolutional layers were employed, and Softmax was used for network traffic categorization. The proposed method did not use any feature skills or important features to train automatically. By defining network traffic behaviour the abovementioned system efficiently increased the performance of intrusion detection.

In another study [22], the researchers suggested a novel model based on a Whale optimization algorithm (WOA) and generic variables for a wireless network's intrusion detection system. To avoid being caught in the local optimum, the mutation operator was utilized, and the crossover operator was used to assist whales increase their search space. First, the model chose the beneficial features that aided in the detection of invasions. Then, to recognize distinct forms of invasions, a support vector machine was deployed. The datasets from Linux and the Canadian Institute of Cyber Security were used to measure the model's performance. When compared to the traditional whale optimization technique and the current evolutionary algorithm, the experimental findings demonstrated that the system had a higher attack detection rate. In [23], researchers concentrated on intrusion detection in network traffic with flow-based data by using two different deep learning techniques, unsupervised deep learning and semi-supervised learning. Variation auto encoder and auto encoder approaches were applied to classify unidentified network traffic attacks using flow features. The model was tested on standard traffic and anomaly data, and the variation auto encoder performance was found to be better than the auto encoder. The literature review found that intrusion detection methods based on artificial intelligence approaches have attracted the attention of researchers. Nevertheless, all existing models need standard/normal traffic patterns to compare with anomalies to detect anomalous traffic. In [24], a Deep extreme learning machine (DELM), neural network, decision tree, and support vector machine were applied to detect malicious traffic in an intelligent home network. The DELM model achieved 93.91% accuracy for the NSL-KDD dataset and 94.6% accuracy for the CUP-99 dataset. For intrusion detection in the networks, the proposed IDS-FMLT in our study focuses on an intrusion

finding structure in a diverse network enabled with a joined machine learning system model. The IDS-FMLT model uses fused machine learning approaches to protect the networks from malicious attacks. Various dataset such as KDD dataset [25], CUP-99 dataset [26], and NetML-2020 dataset [27] and NSL-KDD dataset [28] were used in RNN [29], adaptive voting algorithm [30], ANN [31], DELM [32], SVM [33], SOMNN [34], ANN [35], NAÏVE BAYES + RP [36] and GANS [37].

Tab. 1 shows that most of the researchers used the different state-of-the-art approaches like RNN, adaptive voting algorithm, ANN, DELM, SVM, SOMNN and GANs for the detection of intrusion in networks. It observe that the mostly researchers used single machine learning technique without pre-processing in their proposed intrusion detection models.

Table 1: Summary of state-of-the-art literature survey

Authors	Approaches	Data splitting during training and validation	Accuracy (%)	Limitations
Farhan [29]	RNN	Training (60%) Validation (40%)	93.8	No pre-processing
Chen et al. [30]	Adaptive voting algorithm	Training (85%) Validation (15%)	84.5	No pre-processing and low accuracy
Khan et al. [31]	ANN	Training (85%) Validation (15%)	79.9	No pre-processing and low accuracy
Avallaee et al. [32]	DELM	Training (70%) Validation (30%)	91.3	Less number of samples are used during training
Ibrahim et al. [33]	SVM	Training (85%) Validation (15%)	92.1	No pre-processing
Panda et al. [34]	SOMNN	Training (65%) Validation (35%)	73.1	No pre-processing, less number of samples are used during training and low accuracy
Alshinina et al. [35]	ANN	Training (45%) Validation (55%)	79.9	No pre-processing, less number of samples are used during training and low accuracy

(Continued)

Table 1: Continued

Authors	Approaches	Data splitting during training and validation	Accuracy (%)	Limitations
Rahman et al. [36]	Discriminative Multinomial Naïve Bayes + RP	Training (93%) Validation (7%)	80.6	Less samples are used in validation and low accuracy
Saleem et al. [37]	GANs	Training (85%) Validation (15%)	81.1	No pre-processing and low accuracy

The following are the proposed study key contributions:

a. The main goals are increase intrusion detection accuracy in heterogeneous networks and reduce the miss rate as well.
b. A novel approach Fused machine learning is presented for a better approximation of intrusion traffic in heterogeneous networks.
c. The suggested model also makes it possible to assess network performance in order to increase the accuracy of the existing model.
d. Finally, the proposed fused machine learning algorithm is assessed on three well-known datasets, KDD, CUP-99, and NetML-2020, which include two classes. Simulation outcomes have revealed that the proposed fused ML method reports better results than other existing techniques, such as Artificial Neural Networks [24], Support Vector Machines [24], Decision Trees [24], and DELM [24].

3 Proposed IDS-FMLT Model

The automated detection process is an essential part of any intrusion detection system in a network. However, it is difficult to determine whether the network traffic is anomalous or legitimate. Therefore, automated detection systems that detect malicious network traffic are primarily based on machine learning methods [16]. The suggested model empowered with fused machine learning methods for intrusion detection in a heterogeneous network is shown in Fig. 1.

The training and validation phases of the proposed model are divided into two parts. First, the data repository layer has been used to gather data from various datasets, such as KDD [25], CUP-99 [26], and NetML-2020 [27]. After storing the data, the pre-processing layer activates and handles repeated and missing values using different techniques, such as the mean, mode, and moving average. All three datasets are used at the application layer to train the model using a real-time sequential deep extreme learning machine approach. A mathematical model of Real-time sequential deep extreme learning machine (RTS-DELM) [``````````] model is given below. A DELM is applied to train a feed-forward neural network with hidden layers. Initially, the DELM consists of back-propagation input burdens that are updated casually. The RTS-DELM method involves multiple feed forward hidden layers which contains k hidden nodes neurons as well as a training dataset of K records ($¥_a$, W_a), where $¥_a$ represent the input features and W_a represents the output features.

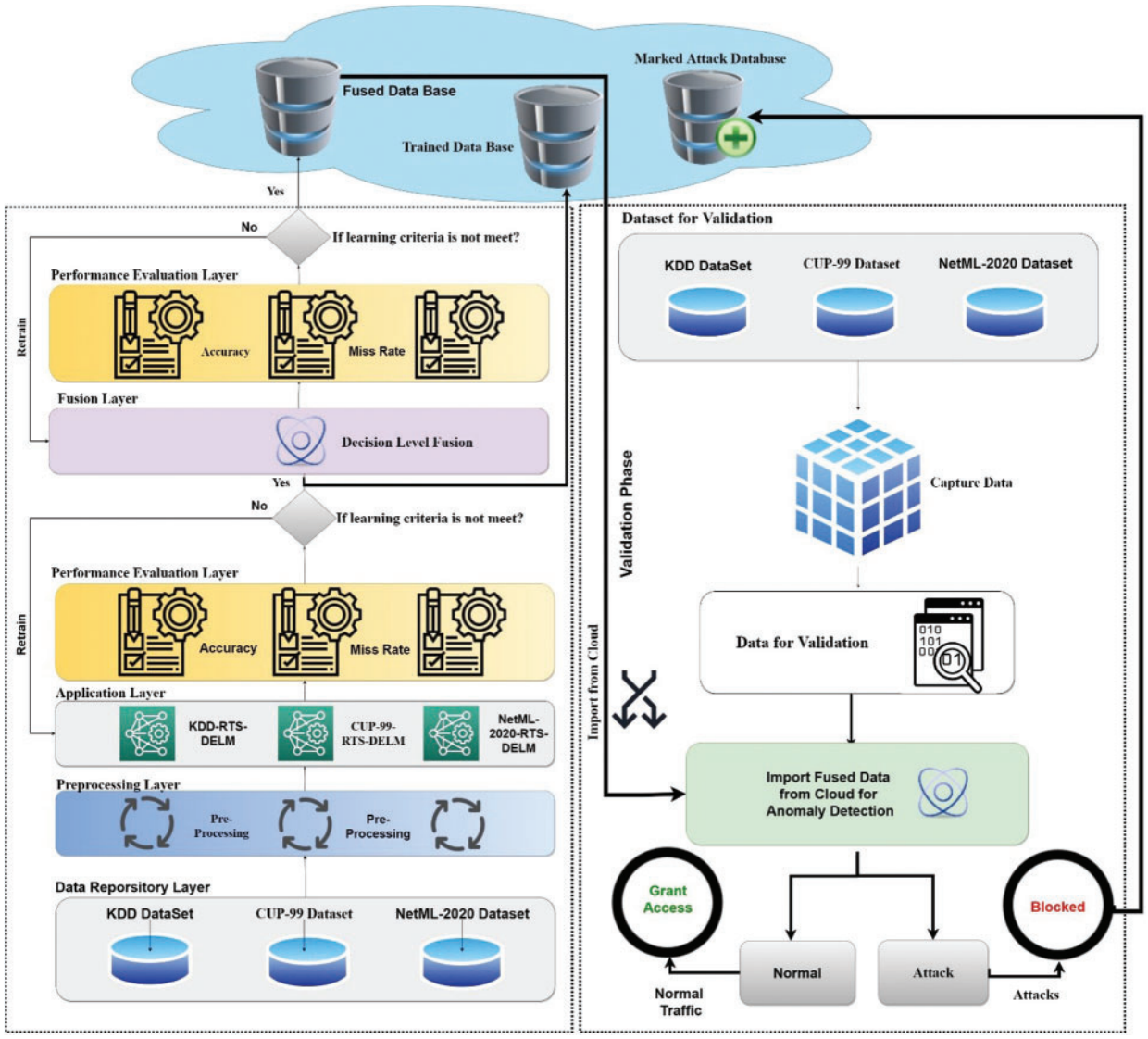

Figure 1: The proposed IDS-FMLT system model

The RTS-DELM operates as follows: ¥$_a$ $\in$ N_m and W_a $\in$ N_l. The results of multiple hidden layers are represent in Eq. (1) [24]:

$$\mathrm{W}_a = \sum_{z=1}^{k} \beta_z \delta \ (\mathrm{Q}_z \mu_a + v_z), \quad a \in [1, K] \tag{1}$$

The learning variables are μ_a and x_z, the output is β_z at weight z, and δ:N $\rightarrow$ N represents the activation function (AF).

An ideal reconciliation of feed-forward neural network based on the multiple hidden layers with zero error simplifies that distinct intervals Q_z and v_z, β_z occur [24] such that:

Eq. (2) [28] shown as:

$$\mathrm{R}\beta = \mathrm{Y} \tag{2}$$

where

$$\mathrm{R} = \begin{bmatrix} \delta\left(\mathrm{Q}_1\mu_1 + v_1\right) & & \delta\left(\mathrm{Q}_k\mu_1 + v_k\right) \\ \vdots & & \vdots \\ \cdots & & \cdots \\ \vdots & & \vdots \\ \delta\left(\mathrm{Q}_1\mu_k + v_1\right) & & \delta\left(\mathrm{Q}_k\mu_k + v_k\right) \end{bmatrix} \tag{3}$$

And the output function is representing in Eq. (4) [28]

$$\beta = \left(\beta_1^T \ldots \ \beta_k^T\right)^T, \ \mathrm{Y} = \left(\mathrm{w}_1^T \ldots \ \mathrm{w}_k^T\right)^T \tag{4}$$

The outcome weights are solved by hidden layer neurons with the following relationship [32] Eq. (5):

$$\beta = \mathrm{R}^{-1} Y \tag{5}$$

R^{-1} Represents the inverse matrix of R. The RTS-DELM [24] shows the computational model applicable for this study.

The deep extreme learning machine model depends on batch learning. An IDS-RTS-DELM is provided a method for dealing with information that arrives in a logical order. When additional information becomes available, the IDS-RTS-DELM model upgrades the batch learning and operates on it.

Let $\{\mu_a, \mathrm{w}_a)\}_{a=1}^{k_0}$ represent the training dataset. R_0 and β_0 show the performance matrix of the hidden layer [28]. The output layer weights are as shown in Eq. (6):

$$\mathrm{R}_0 = \begin{bmatrix} \delta\left(\mathrm{Q}_1\mu_1 + v_1\right) & & \delta\left(\mathrm{Q}_k\mu_1 + v_k\right) \\ \vdots & & \vdots \\ \cdots & & \cdots \\ \vdots & & \vdots \\ \delta\left(\mathrm{Q}_1\mu_{K_0} + v_1\right) & & \delta\left(\mathrm{Q}_k\mu_{K_0} + v_k\right) \end{bmatrix} \tag{6}$$

$$\beta_0 = \mathrm{T}_0\mathrm{R}_0^T\mathrm{Y}_0 \tag{7}$$

where

$$\mathrm{T}_0 = \left(\mathrm{w}_1^T \ \mathrm{R}_0\right)^{-1} \text{ And } \mathrm{Y}_0 = \left(\mathrm{w}_1^T \ \ldots \mathrm{w}_{K_0}^T \right)^T \tag{7a}$$

The creation of the $(j + 1)^{th}$ record with K^{j+1} records and the performance of the partially hidden layer are as follows Eq. (8) [28]:

$$R_{j+1} = \begin{bmatrix} \delta\left(Q_1 \mu_{\left(\sum_{z=0}^{j} Kz\right)+1} + v_1\right) & \delta\left(Q_k \mu_{\left(\sum_{z=0}^{j} Kz\right)+1} + v_k\right) \\ \vdots & \vdots \\ \cdots & \cdots \\ \vdots & \vdots \\ \delta\left(Q_1 \mu_{\left(\sum_{z=0}^{j} Kz\right)+1} + v_1\right) & \delta\left(Q_k \mu_{\left(\sum_{z=0}^{j} Kz\right)+1} + v_k\right) \end{bmatrix} \tag{8}$$

The performance weights can be adjusted by using the following Eq. (9):

$$\beta_{j+1} = \beta_j + T_{j+1} R_{j+1}^{T} \left(Y_{j+1} - R_{j+1} \beta_j\right) \tag{9}$$

$$T_{j+1} = T_j + T_j R_{j+1}^{T} \left(BG + R_{j+1} T_j R_{j+1}^{T}\right)^{-1} R_{j+1} T_j \tag{10}$$

$$cx_z = g\left(T_{j+1}\right)$$

where $g\left(T_{j+1}\right)$ represents the sigmoid activation function in the hidden layer [32].

$$Error = \frac{1}{2} \sum_{k} (mz - cx_z)^2 \tag{11}$$

where

$m_z =$ desired outcome

$cx_z =$ calculated outcome

Eq. (11) represents a back propagation error. The weights can be updated to minimize this error [28]. The change in weight is shown in Eq. (12) as follows:

$$\Delta F_{a,z}^{r=n} \propto -\frac{\partial T}{\partial F^{r=n}} \tag{12}$$

$r = 1, 2, 3 \ldots n$ (*no.of neurons*)

And z = Layer of Output Value

Eq. (13) demonstrates the weight update and biases, where σ represents the learning rate [24–28].

$$F_{a,k}^{r}(h+1) = F_{a,k}^{r}(h) + \sigma \Delta F_{a,z}^{r} \tag{13}$$

Eq. (13) shows the updated weights of the RTS-DELM with ith weight. The standard evaluation layer is again activated and assesses the performance of the recommended system model. If the learning standards do not satisfy the proposed model's requirements, the model must be retrained; otherwise, the decision level fusion layer is activated and the trained model is kept in a cloud database. In the fusion layer, decision-level fusion empowered with fuzzy logic is applied to train the model. In this article we used mamdani fuzzy inference system for decision level fusion. As we know that the performance of any decision based system is rely on knowledgebase. The proposed decision level fusion is shown below: IF (KDD-RTS-DELM is no and CUP-99-RTS-DELM is no and NetML-2020 is no) THEN (Intrusion Detection is no)

IF (KDD-RTS-DELM is no and CUP-99-RTS-DELM is no and NetML-2020 is yes) THEN (Intrusion Detection is yes)

IF (KDD-RTS-DELM is no and CUP-99-RTS-DELM is yes and NetML-2020 is no) THEN (Intrusion Detection is yes)

IF (KDD-RTS-DELM is no and CUP-99-RTS-DELM is yes and NetML-2020 is yes) THEN (Intrusion Detection is yes)

IF (KDD-RTS-DELM is yes and CUP-99-RTS-DELM is no and NetML-2020 is no) THEN (Intrusion Detection is yes)

IF (KDD-RTS-DELM is yes and CUP-99-RTS-DELM is no and NetML-2020 is yes) THEN (Intrusion Detection is yes)

IF (KDD-RTS-DELM is yes and CUP-99-RTS-DELM is yes and NetML-2020 is no) THEN (Intrusion Detection is yes)

IF (KDD-RTS-DELM is yes and CUP-99-RTS-DELM is yes and NetML-2020 is yes) THEN (Intrusion Detection is yes)

The performance evaluation of the fusion layer is activated to evaluate the fused machine learning approaches. If the learning criteria do not meet the requirements, then retune the fusion layer parameters, otherwise, the fused trained model are stored in the fused database in the cloud for future use.

In the validation phase, the KDD, CUP-99 and NetML-2020 datasets are used to evaluate the proposed IDS-FMLT model. The fused model is imported from the cloud for the prediction of network traffic. The proposed IDS-FMLT model predicts two types of network traffic, namely, normal and malicious attacks. If the proposed IDS-FMLT model predicts normal traffic, then traffic is granted access. If the proposed IDS-FMLT model predicts malicious traffic, then traffic is blocked and stored as a marked attack in the database in the cloud.

4 Results and Discussion

In this research, MATLAB 2020a tool is used for the simulation purpose. The (RTS-DELM) is applied to input KDD data, CUP-99 data, and NetML-2020 data. The data are separated into two stages: training & validation. Seventy percent of the data (855600 fused samples) and 30% (366685 fused samples) data are used for validation in the training phase which are shown in Tabs. 2 and 3.

Table 2: Training data for the proposed intrusion detection scheme in heterogeneous networks empowered with fused machine learning model

Proposed fusion based IDS-FMLT (training)			
fused samples (FS = 855600)		Output results ($\in_A$, $\in_B$)	
Input	Expected output (I_A, I_B)	$\in_A$ (Access)	$\in_B$ (Block)
	I_A = 784077 access	776738	7339
	I_B = 71523 block	20639	50884

Table 3: Validation data for the proposed intrusion detection scheme in heterogeneous networks empowered with a fused machine learning model

Proposed fusion IDS-FMLT (validation)			
Total number of fused samples (FS = 366685)		Output results ($\in_A$, $\in_B$)	
Input	Expected output (I_A, I_B)	$\in_A$ (Access)	$\in_B$ (Block)
	I_A = 336033 (Access)	330109	5924
	I_B = 30652 (Block)	11751	18901

The various statistical analysis parameters [38–41] like accuracy, miss rate, probability of detection, true negative rate, incidence, positive predicted value, and negative projected importance of the proposed IDS-FMLT model are all evaluated in the following way:

$$\text{Accuracy} = \frac{\sum \text{True positive (access)} + \sum \text{True negative (block)}}{\sum \text{Total population}} \tag{14}$$

$$\text{Miss rate} = \frac{\sum \text{False positive (access)} + \sum \text{False negative (block)}}{\sum \text{Total population}} \tag{15}$$

$$\text{Probability of detection (PD)} = \frac{\sum \text{True positive (access)}}{\sum \text{True positive (access)} + \sum \text{False negative (block)}} \tag{16}$$

$$\text{True Negative Rate (TNR)} = \frac{\sum \text{True negative (block)}}{\sum \text{True negative (block)} + \sum \text{False Positive (access)}} \tag{17}$$

$$\text{Prevelance} = \frac{\sum \text{Condition Positive}}{\sum \text{Condition Positive} + \sum \textit{Condition Negative}} \tag{18}$$

$$\text{Positive Predicted Value} = \frac{\sum \text{True positive (accesss)}}{\sum \text{True Positive (access)} + \sum \text{False Positive (access)}} \tag{19}$$

$$\text{Negative Predicted Value} = \frac{\sum \text{True Negative (block)}}{\sum \text{True Negative (block)} + \sum \text{False Negative (block)}} \tag{20}$$

$$\text{False Alarm Rate} = 1 - \frac{\sum \text{True Negative (block)}}{\sum \text{True Negative (block)} + \sum \text{False Negative (block)}} \tag{21}$$

$$\text{Specificity} = \frac{\sum \text{True Negative (block)}}{\sum \text{True Negative (block)} + \sum \text{False Negative (block)}} \tag{22}$$

$$\text{Sensititvity} = \frac{\sum \text{True Negative (block)}}{\sum \text{True Negative (block)} + \sum \text{False Negative (block)}} \tag{23}$$

Fig. 2 shows the training statistical measures of different parameters, such as accuracy, miss rate, and probability of detection, specificity, prevalence, positive predicted value, and negative predicted

value. The proposed IDS-FMLT model has obtained an accuracy of 96.73% and 95.18% during the training and validation phases, respectively. Other statistical parameters, such as the miss rate, probability of detection, specificity, prevalence, positive predicted value, specificity, sensitivity and negative predicted value are reported as 3.27%, 97.41%, 87.40%, 93.20%, 99.06%, 87.40%, 97.41% and 71.14%, respectively, in the training phase.

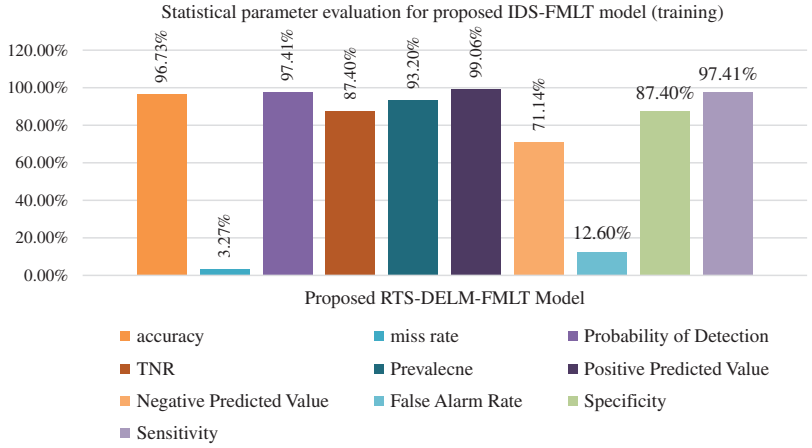

Figure 2: Performance evaluation with statistical parameters for proposed IDS-FMLT model (training)

Fig. 3 illustrates the proposed model's validation performance for a variety of statistical parameters, including miss rate, rate of detection, specificity, prevalence, positive projected value, specificity, sensitivity and negative predicted value. The suggested model obtained 4.82%, 96.56%, 76.14%, 93.20%, 98.24%, 76.14%, 96.56% and 61.70% for miss rate, probability of detection, specificity, prevalence, positive predicted value, and negative projected value, respectively, during the validation stage.

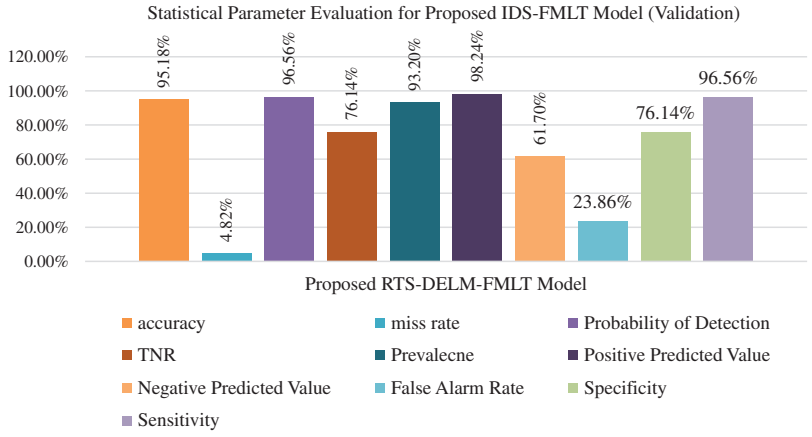

Figure 3: Performance evaluation with statistical parameters for the proposed IDS-FMLT model (validation)

Tabs. 3 and 4 demonstrates the overall performance of the proposed IDS-FMLT model and compare it with other state-of-the-art techniques. The experimental results of the IDS-FMLT model

achieved an accuracy of 95.18% and a missing rate of 4.82%, which are better than those of different existing approaches, such as a neural network [24], support vector machine [24], decision tree [24], deep extreme learning machine [24], RNN [29], adaptive voting algorithm [30], ANN [31], DELM [32], SVM [33], SOMNN [34], ANN [35], NAÏVE BAYES + RP [36] and GANS [37].

Table 4: Comparison of the proposed IDS-FMLT system model with various techniques across multiple datasets

Method	Accuracy (Dataset KDD) [25]	Accuracy, (Dataset CUP-99) [26]	Accuracy (Dataset NetML-2020) [27]
Neural network [24]	81.2%	90.39%	91.23%
Support vector machine [24]	69.52%	89.94%	90.07%
Decision tree [24]	81.5%	91.12%	92.79%
Deep extreme learning machine [24]	93.91%	94.6%	94.93%
Proposed IDS-FMLT model	**Fused** [(RTS-KDD) + (RTS-CUP-99) + (RTS-NetML-2020)] **95.18%**		
Method	Miss rate (Dataset KDD) [25]	Miss rate (Dataset CUP-99) [265]	Miss rate (Dataset NetML-2020) [27]
Neural network [24]	18.8%	9.61%	8.77%
Support vector machine [24]	30.48%	10.06%	9.93%
Decision tree [24]	18.5%	8.88%	7.21%
Deep extreme learning machine [24]	6.09%	5.4%	5.07%
Proposed IDS-FMLT model	**Fused** [(RTS-KDD) + (RTS-CUP-99) + (RTS-NetML-2020)]**4.82%**		

Tab. 5 shows the in-contrast of proposed IDS-FMLT model with various state-of-the-art existing approaches [29–37]. The performance the proposed IDS-FMLT model in terms of different statistical parameters like "accuracy and miss rate", during the training the proposed IDS-FMLT model is achieved 96.73% and validation phase the proposed IDS-FMLT model achieved the accuracy 95.18%. The summary of existing literature results analysis that clearly shows the accuracy is enhanced as compared to various state-of-the-art existing ML approaches.

Table 5: Performances of state-of-the-art approaches and the proposed IDS-FMLT system model

Authors	Pre-processing	Approaches	Training/Validation	Accuracy (%)	Miss-rate (%)
Farhan [29]	No	RNN	Training (60%) Validation (40%)	94.1 93.8	5.9 6.2
Chen et al. [30]	No	Adaptive voting algorithm	Training (85%)	85.2	14.8

(Continued)

Table 5: Continued

Authors	Pre-processing	Approaches	Training/Validation	Accuracy (%)	Miss-rate (%)
			Validation (15%)	84.5	15.5
Khan et al. [31]	No	ANN	Training (85%)	81.2	18.8
			Validation (15%)	79.9	20.1
Avallaee et al. [32]	Yes	DELM	Training (70%)	92.1	7.9
			Validation (30%)	91.3	8.7
Ibrahim et al. [33]	No	SVM	Training (85%)	93.5	6.5
			Validation (15%)	92.1	7.9
Panda et al. [34]	No	SOMNN	Training (65%)	75.4	24.6
			Validation (35%)	73.1	26.9
Alshinina et al. [35]	No	ANN	Training (45%)	81.2	18.8
			Validation (55%)	79.9	20.1
Rahman et al. [36]	No	Discriminative multinomial Naïve Bayes + RP	Training (93%)	81.5	18.5
			Validation (7%)	80.6	19.4
Saleem et al. [37]	No	GANs	Training (85%)	86.5	13.5
			Validation (15%)	81.1	18.9
Proposed IDS-FMLT model	Yes	Fused ML (RTS-DELM)	Training (70%)	96.73	3.27
			Validation (30%)	95.18	4.82

5 Conclusion

The rapid growth of data necessitates securing networks from malicious attacks. The proposed IDS-FMLT model is used to predict malicious and regular traffic in the networks. Moreover, the proposed approach is tested on various heterogeneous datasets, namely, KDD, CUP-99, and NetML-2020. Statistical significance is analyzed and compared to state-of-the-art methods for malicious attacks in networks.

The experimental results of the proposed IDS-FMLT model demonstrated an accuracy of 96.73% for training and 95.18% for validation. The proposed IDS-FMLT model achieved satisfactory performance as compared to other state-of-the-art studies.

6 Limitations and Future Work

The proposed IDS-FMLT classifies the network traffic into two categories: malicious and regular. The proposed Fused Machine Learning algorithm was used to increase the system performance in terms of detection of malicious and regular traffic. On the other hand, the IDS-FMLT framework may increase the computational complexity of the system. In future work, Long short-term memory (LSTM), federated learning, and hybrid computational intelligence can be applied, which may improve the system's accuracy. In future, we may use the other latest published intrusion detection data set like IoT datasets for intrusion detection systems such as CIC IoT Dataset 2022 and other bench marks dataset as well like NSL-KDD or UGR16 or UNSW-NB15 or CICDS-17, 18, and 19 during training phase, which make our system more reliable in real time application

Acknowledgement: We thank our families and colleagues who provided us with moral support.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] T. Zhang, D. Han, M. D. Marino, L. Wang and K. C. Li, "An evolutionary-based approach for low-complexity intrusion detection in wireless sensor networks," *Wireless Personal Communications*, vol. 119, no. 4, pp. 1–24, 2021.

[2] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Applied Sciences*, vol. 9, no. 20, pp. 4396–4405, 2019.

[3] I. F. Kilincer, F. Ertam and A. Sengur, "Machine learning methods for cyber security intrusion detection: Datasets and comparative study," *Computer Networks*, vol. 188, pp. 107840–107848, 2021.

[4] A. Aldweesh, A. Derhab and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issue," *Knowledge-Based Systems*, vol. 189, pp. 105124–105132, 2020.

[5] G. Kocher and G. Kumar, "Machine learning and deep learning methods for intrusion detection systems: Recent developments and challenges," *Soft Computing*, vol. 25, no. 15, pp. 9731–9763, 2021.

[6] H. Zhang, Y. Li, Z. Lv, A. K. Sangaiah and T. Huang, "A real-time and ubiquitous network attack detection based on a deep belief network and support vector machine," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 3, pp. 790–799, 2020.

[7] S. N. Mighan and M. Kahani, "A novel scalable intrusion detection system based on deep learning," *International Journal of Information Security*, vol. 20, no. 3, pp. 387–403, 2021.

[8] G. B. Huang, Q. Y. Zhu and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[9] G. Li, P. Niu, X. Duan and X. Zhang, "Fast learning network: A novel artificial neural network with a fast learning speed," *Neural Computing and Applications*, vol. 24, no. 7, pp. 1683–1695, 2014.

[10] M. H. Ali, B. A. D. A. Mohammed, A. Ismail and M. F. Zolkipli, "A new intrusion detection system based on a fast learning network and particle swarm optimization," *IEEE Access*, vol. 6, pp. 20255–20261, 2018.

[11] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, pp. 4150–4179, 2021.

[12] W. Chen, H. Cao, X. Lv and Y. Cao, "A hybrid feature extraction network for intrusion detection based on a global attention mechanism," in *Int. Conf. on Computer Information and Big Data Applications*, Atlanta, GA, USA, pp. 481–485, 2020.

[13] Z. Chkirbene, A. Erbad, R. Hamila, A. Gouissem, A. Mohamed *et al.,* "Machine learning-based cloud computing anomalies detection," *IEEE Network*, vol. 34, no. 6, pp. 178–183, 2020.
[14] 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. early access, pp. 1–14, 2020.
[15] X. A. L. Novo, M. V. Barbas, V. A. Villagrá and M. S. Rodrigo, "Evaluation of cybersecurity data set characteristics for their applicability to neural networks algorithms detecting cybersecurity anomalies," *IEEE Access*, vol. 8, pp. 9005–9014, 2020.
[16] M. Injadat, A. Moubayed, A. B. Nassif and A. Shami, "Multi-stage optimized machine-learning framework for network intrusion detection," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1803–1816, 2021.
[17] W. Zhong, N. Yu and C. Ai, "Applying big data-based deep learning system to intrusion detection," *Big Data Mining and Analytics*, vol. 3, no. 3, pp. 181–195, 2020.
[18] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico and A. Pescapé, "A hierarchical hybrid intrusion detection approach in IoT scenarios," in *IEEE Global Communications Conf.*, Taipei, Taiwan, pp. 1–7, 2020.
[19] A. Kim, M. Park and D. H. Lee, "Ai-ids: Application of deep learning to real-time web intrusion detection," *IEEE Access*, vol. 8, pp. 70245–70261, 2020.
[20] A. Nagaraja, U. Boregowda, K. Khatatneh, R. Vangipuram, R. Nuvvusetty *et al.,* "Similarity-based feature transformation for network anomaly detection," *IEEE Access*, vol. 8, pp. 39184–39196, 2020.
[21] 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.
[22] R. Vijayanand and D. Devaraj, "A novel feature selection method using whale optimization algorithm and genetic operators for intrusion detection system in wireless mesh network," *IEEE Access*, vol. 8, pp. 56847–56854, 2020.
[23] S. Zavrak and M. İskefiyeli, "Anomaly-based intrusion detection from network flow features using variational autoencoder," *IEEE Access*, vol. 8, pp. 108346–108358, 2020.
[24] M. A. Khan, S. Abbas, A. Rehman, Y. Saeed, A. Zeb *et al.,* "A machine learning approach for blockchain-based smart home networks security," *IEEE Network*, vol. 35, no. 3, pp. 223–229, 2021.
[25] A. Inam, A. Sarwar, A. Atta, I. Naaseer, S. Y. Siddiqui *et al.,* "Detection of COVID-19 enhanced by a deep extreme learning machine," *Intelligent Automation and Soft Computing*, vol. 27, no. 3, pp. 701–712, 2021.
[26] M. Sheikhan, Z. Jadidi and A. Farrokhi, "Intrusion detection using reduced-size RNN based on feature grouping," *Neural Computing and Applications*, vol. 21, no. 6, pp. 1185–1190, 2012.
[27] X. Gao, C. Shan, C. Hu, Z. Niu and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," *IEEE Access*, vol. 7, no. 3, pp. 82512–82521, 2019.
[28] B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ," in *IEEE Int. Conf. on Signal Processing and Communication Engineering Systems*, Guntur, India, pp. 92–96, 2015.
[29] A. M. Farhan, "Effect of rotation on the propagation of waves in hollow poroelastic circular cylinder with magnetic field," *Computers, Materials & Continua*, vol. 53, no. 2, pp. 129–156, 2017.
[30] X. Chen and J. H. Jiang, "A method of virtual machine placement for fault-tolerant cloud applications," *Intelligent Automation & Soft Computing*, vol. 22, no. 4, pp. 587–597, 2016.
[31] M. A. Khan, A. Rehman, K. M. Khan, M. A. A. Ghamdi and S. H. Almotiri, "Enhance intrusion detection in computer networks based on deep extreme learning machine," *Computers, Materials & Continua*, vol. 66, no. 1, pp. 467–480, 2021.
[32] M. Avallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the kdd cup-99 dataset," in *IEEE Symp. on Computational Intelligence for Security and Defense Applications*, Guntur, India, pp. 1–6, 2009.
[33] L. M. Ibrahim, D. T. Basheer and M. S. Mahmod, "A comparison study for intrusion database based on self-organization map artificial neural network," *Journal of Engineering Science and Technology*, vol. 8, no. 1, pp. 107–119, 2013.

[34] M. Panda, A. Abraham and M. R. Patra, "Discriminative multinomial naïve Bayes for network intrusion detection," in *Sixth Int. Conf. on Information Assurance and Security*, Atlanta, GA, USA, pp. 5–10, 2010.
[35] R. Alshinina and K. Elleithy, "A highly accurate machine learning approach for developing wireless sensor network middleware," in *Wireless Telecommunications System*, Phoenix, AZ, USA, pp. 1–7, 2018.
[36] A. Rahman, S. Abbas, M. Gollapalli, R. Ahmed, S. Aftab *et al.,* "Rainfall prediction system using machine learning fusion for smart cities," *Sensors*, vol. 22, no. 6, pp. 3504–3519, 2022.
[37] M. Saleem, S. Abbas, T. M. Ghazal, M. A. Khan, N. Sahawneh *et al.,* "Smart cities: Fusion-based intelligent traffic congestion control system for vehicular networks using machine learning techniques," *Egyptian Informatics Journal*, vol. 23, no. 1, pp. 1–10, 2022.
[38] M. W. Nadeem, H. G. Goh, M. A. Khan, M. Hussain, M. F. Mushtaq *et al.,* "Fusion-based machine learning architecture for heart disease prediction," *Computers, Materials & Continua*, vol. 67, no. 2, pp. 2481–2496, 2021.
[39] S. Y. Siddiqui, A. Athar, M. A. Khan, S. Abbas, Y. Saeed *et al.,* "Modelling, simulation and optimization of diagnosis cardiovascular disease using computational intelligence approaches," *Journal of Medical Imaging and Health Informatics*, vol. 10, pp. 1005–1022, 2020.
[40] H. Sun and R. Grishman, "Lexicalized dependency paths based supervised learning for relation extraction," *Computer Systems Science and Engineering*, vol. 43, no. 3, pp. 861–870, 2022.
[41] I. Naseer, B. S. Khan, S. Saqib, S. N. Tahir, S. Tariq *et al.,* "Diagnosis heart disease using mamdani fuzzy inference expert system," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 7, no. 26, pp. 1–9, 2020.