Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.033896
Article

An IoT Environment Based Framework for Intelligent Intrusion Detection

Hamza Safwan[1], Zeshan Iqbal[1], Rashid Amin[1], Muhammad Attique Khan[2], Majed Alhaisoni[3], Abdullah Alqahtani[4], Ye Jin Kim[5] and Byoungchol Chang[6,*]

[1]Deparment of Computer Science, UET Taxila, Taxila, 47080, Pakistan
[2]Department of Computer Science, HITEC University, Taxila, 47080, Pakistan
[3]Computer Sciences Department, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh, 11671, Saudi Arabia
[4]College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj, Saudi Arabia
[5]Department of Computer Science, Hanyang University, Seoul, 04763, Korea
[6]Center for Computational Social Science, Hanyang University, Seoul, 04763, Korea
*Corresponding Author: Byoungchol Chang. Email: bcchang@hanyang.ac.kr
Received: 30 June 2022; Accepted: 07 September 2022

Abstract: Software-defined networking (SDN) represents a paradigm shift in network traffic management. It distinguishes between the data and control planes. APIs are then used to communicate between these planes. The controller is central to the management of an SDN network and is subject to security concerns. This research shows how a deep learning algorithm can detect intrusions in SDN-based IoT networks. Overfitting, low accuracy, and efficient feature selection is all discussed. We propose a hybrid machine learning-based approach based on Random Forest and Long Short-Term Memory (LSTM). In this study, a new dataset based specifically on Software Defined Networks is used in SDN. To obtain the best and most relevant features, a feature selection technique is used. Several experiments have revealed that the proposed solution is a superior method for detecting flow-based anomalies. The performance of our proposed model is also measured in terms of accuracy, recall, and precision. F1 rating and detection time Furthermore, a lightweight model for training is proposed, which selects fewer features while maintaining the model's performance. Experiments show that the adopted methodology outperforms existing models.

Keywords: Software-defined networks; prediction modeling; machine learning; deep learning

1 Introduction

SDN (Software Defined Networks) is a new approach to routing traffic on the intranet or the internet. It is adaptable, dynamic, and cost-effective. Because the control and data planes are separated in the SDN environment, it promotes separation of concerns. The SDN control, data, and application planes are comprised of three planes. All network policies can be applied by the controller. SDN is also deployed on Internet of Things (IoT) networks, where many security issues arise during SDN implementation. Because the controller is a critical component of SDN, attackers focus their efforts

on it. If the controller is attacked, the entire network is under attack. Denial of Service (DoS) and Distributed Denial of Service (DDoS) is the most common attacks on SDN. For that purpose, an Intrusion Detection Systems (IDS) are used.

Currently, SDN is deployed in many big enterprise organizations like (Google, Amazon, Facebook, and Cisco). SDN capability helps us solve several security issues present in traditional networking. However, SDN introduces some new threats discussed in [1].

The performance of IDS increased by using Machine Learning techniques [2]. The adoption of new and emerging technologies like IoT, cloud, Network Function Virtualization (NFV), and AI has [3] increased the risk of cybersecurity because they produce an enormous amount of data and make information security challenges. There are various attacks like signature-based IDS and Anomaly-based IDS [4].

The intrusion detection system is critical in detecting malicious attacks. Because of their advantages in zero-day attacks, machine learning algorithms have grown in popularity [5]. The threat of DoS and DDoS is constantly increasing, disrupting many network services [6,7]. For all of these reasons, rapid and precise detection is required. Detecting DoS and DDoS attacks in the network, on the other hand, is difficult. DDoS attacks are classified as resource-consuming, application-layer, or volumetric. Machine learning-based NIDS techniques have been used in a variety of applications. Web-shell intrusion in IoT networks using the ensemble technique [8].

These challenges form the motivation of this work. There is a need to address all these problems and provide an adequate solution for these problems. Our work assumes the following:

- Proposed a hybrid machine learning model based on Random Forest and Long Short Term Memory (LSTM).
- Investigate the Machine Learning solutions for IDS and also discuss the research gaps.
- Proposed a combination of regularization technique based on L2 regularization and Dropout to solve the problem of overfitting.
- Several experiments are performed to evaluate the performance of the proposed model.
- Validate the performance of the model on SDN specific InSDN dataset.
- Choose the most relevant features for attack detection in SDN Environment.

2 Related Work

Since the beginning of computer architecture, there has been research in NIDS and HIDS. The application of machine learning and deep learning techniques to NIDS and HIDS is now critical. In [9] describes a detailed survey of existing techniques. Applying machine learning algorithms to SDN has attracted many researchers. A solution proposed in [10] solves some issues present in KDD-CUP 99 dataset by an experimental study using the NSL-KDD dataset and achieves the best performance. Five algorithms are trained on the NSL-KDD dataset resulting in the accuracy of 97% for the random forest, 83% for J48%, 94% for CART, 85% for SVM, and 70% for Naïve Bayes. In [11], the author used Deep Neural Network for the experiment. NSL-KDD dataset is used, and six features are used to train the proposed method. As a result, 75% accuracy is achieved through this method.

In [12], the authors extended their research using GRU-RNN and achieved an accuracy of 89%. In [13], the author uses PCA for IDS. NSL-KDD is used in this research. Min-Max normalization technique is being used. This method achieves an overall accuracy of 99%. In [14], the author presents a model with two stages of a Deep Neural Network designed for NIDS. A stacked auto-encoder is used with a SoftMax activation function. 89.134% accuracy is achieved on the UNSW-NB15 dataset.

In [15], the LSTM and Autoencoder approach classify the attacks. In [16], an Adversarial autoencoder neural network is used for NIDS with the combination of GANs and various auto-encoders. GAN consists of two networks generator and a discriminator. In the study, the generator generates fake packets, and the discriminator tells whether the packet is legitimate or not. In [17] ensemble technique for intrusion detection is introduced. In the study, four different machine learning algorithms are used SVM-SOM shows the best accuracy of 98.12%. In [18], the author used a CNN multi-channel deep learning feature algorithm. Two fully connected layers are used in the experiment, along with the SoftMax activation function. Three different datasets are used in this study, with an accuracy of 94%. Furthermore, a new Scale Hybrid-IDS AlertNet (SHIA) model is proposed in [19]. This framework effectively monitors the network traffic and detects any possible network attack. This study uses and analyzes several datasets: UNSW-NB 15, KDD-CUP99, and NSL-KDD.

In [20] UNSW-NB15 dataset was used. CNN algorithm is used in the research based on residual learning to learn more features. The modified focal loss function addresses the problem of class imbalance. To avoid overfitting, global average pooling and Batch normalization is used. In [21], a new method called CRNN (convolutional recurrent neural network is used for research. In this method, CNN performs convolution to capture local features. RNN fetches temporal features. This improves the accuracy and performance of IDS. CSC-CIC-DS 2018 dataset is used for the experiment.97.75% accuracy is achieved by using 10-fold cross-validation. In [22], three machine learning algorithms are used. SVM, Naïve Bayes, and KNN are used with UNSW-NB 15 dataset.95% accuracy is achieved through SVM. In [23] the author used SVM, naïve bayes, artificial intelligence and KNN. The wrapper feature selection technique is used in the study. KNN achieves an overall accuracy of 98.3%. This study also shows that feature selection methods can increase the model's accuracy. In [24] Decision Tree-Recursive Feature Elimination method is used. A decision tree is used with an accuracy of 98%.

In [25], evolutionary SVM and the KPCA feature selection method are used in a new proposed methodology. KPCA is used to reduce the dimensions of the data. In the study, the N-RBF algorithm is used to reduce noise. As a result, 98% accuracy is achieved in the study. In [26], the author used the NSL-KDD dataset to detect multi-classification attacks. XGBoost, Random Forest, and Decision Tree are used. XGBoost achieves an overall accuracy of 95%. The performance of XGBoost is compared with other algorithms to measure its performance. In [27] proposed a new technique to detect DDoS attacks in SDN Environment. A new technique called (EMSOM-KD) is proposed. In this technique, entropy is combined with SOM. The problem of suspicious and dead neurons is solved by using the SOM technique. Optimal accuracy is achieved in this research. In [28] real-time adaptive DDOS detection technique based on the RT-SAD technique is proposed. Optimal accuracy is achieved in this research.

Using machine learning and Deep Learning algorithms, the proposed methodology improves IDS implementation. The NSL-KD dataset is used in this paper to detect anomalies using enhanced Random Forest and MLP, with each class determining whether the traffic is an attack or normal. There are several steps we take to detect network intrusion. All of the steps are outlined below. In [29] the authors of the research applied the dimensionality reduction technique to the Internet of Things and reduced storage and communication cost. After applying the dimensionality reduction technique, classification algorithms are applied to IoT data. Good results are obtained after applying the dimensionality reduction technique to IoT data. In [30], the authors applied Deep Learning techniques to detect or predict road attacks. Deep Learning algorithms provide good results in determining road accidents. This technique can also be applied to the intrusion detection system.

3 Proposed Methodology

The proposed methodology improves the SDN implementation of the Intrusion Detection System. For anomaly detection, this study employs a hybrid machine learning model. For implementation, the In SDN dataset is used. There are several features in the dataset, including src bytes, dst bytes, pkt count, and labels. This section explains the dataset and features used. To determine the best features in this study, a feature selection algorithm is used. This will increase classification accuracy. The first step in classification is data preprocessing noise is removed in this process. Then, in preprocessing feature, scaling is also done. After preprocessing next step is feature selection. In feature selection, feature selection techniques are used. After the feature selection model building phase starts. In model building, the preprocessed dataset is given to the machine learning model. After model building, there is a prediction stage, and at last, we calculate the model's accuracy based on the confusion matrix and ROC Curve. In binary classification confusion, matric and ROC curves are used to determine the performance of the machine learning algorithm. In the figure below, data is collected and then preprocessed. Preprocessing is necessary as the dataset contains so much noise and outliers; we need to remove them from our dataset to achieve good results.

3.1 Dataset

Various datasets are used to analyze the intrusion in the network. Most of the datasets are IP-based, like DARPA98, NSL-KDD and KDD-CUP99. For SDN, a Flow-based and SDN-specific dataset is required. Most datasets are outdated and cannot perform well on SDN-related problems. InSDN dataset is used in this research. It is specifically used for SDN. The dataset was made publicly available to researchers for deep learning and Machine Learning research. There are 84 features in the dataset [31]. The dataset contains TCP, UDP, and ICMP traffic data. Its target labels are DDoS, probe, DoS, Normal, BFA, Web-Attack, Botnet and U2R. The dataset contains statistical features like src bytes, dst bytes, and packet per flow. This dataset is used in many research articles and is considered a standard dataset to find intrusion in the network.

Table 1: In SDN attack distribution

No	Features
Normal	18%
DDoS	34%
DoS	9.67%
Probe	27%
Web-attack	0.053%
U2R	0.0047%
Password guessing	0.388%

Several attack categories are present in the dataset. We only deal with binary classification in this research. Normal is labeled with 0, and all the attack categories are 0. The SDN dataset also contains SDN-related features that are not present in other datasets. Some datasets like NSL-KDD, KDD CUP 99 and DARPA are standard but only contain traces of traditional networks. They are not flow-based datasets and contain very few attacks, which are mostly insufficient for today's networks. In SDN contains flow based as well as SDN related features. Flow based datasets keeps track of all the contents of the packet. All the packet information is inspected in flow-based datasets. These features

are described below in Table 2. The features which are specifically used for SDN are only present in this dataset. We develop our model based on these features to detect anomalies in the network.

Table 2: SDN specific features

No	Features
1	Tot-Fwd-Pkts
2	Total-Bwd-Pkts
3	Totallen-Fwd-Pkts
4	Fwd-Pkts-len-min
5	Fwd-Pkts-len-max
6	Flow bytes

3.2 Data Visualization

Pie charts in Figs. 1 and 2 are used to visualize the distribution of abnormal and normal labels in dataset and multiclass attacks respectively.

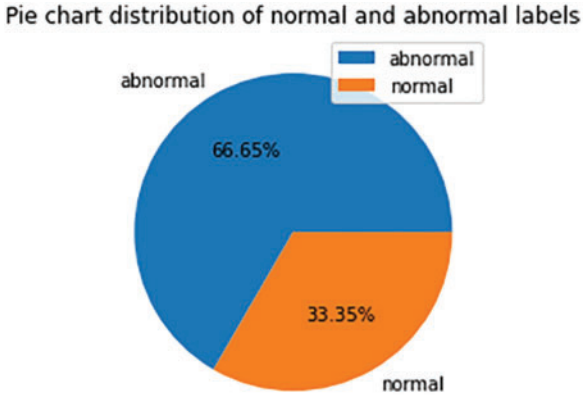

Figure 1: Dataset distribution

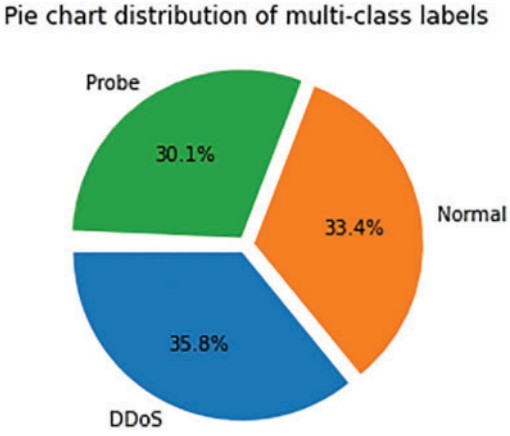

Figure 2: Multi-class attacks

3.3 Machine Learning Approaches

Machine Learning and Deep Learning approaches are used to analyze the system's performance and detect unusual events that are considered malicious. We have data with high density in network systems, and movements are detected through statistical machine learning models. In this section, machine learning algorithms are defined.

3.4 Proposed Model

The proposed model for this research is based on a hybrid technique that combines machine learning with deep learning, as shown in Fig. 3. In the proposed model Random Forest model is combined with the LSTM algorithm. Furthermore, the problem of underfitting and overfitting is solved by combining the l2 regularization technique with the Dropout technique. In this way, optimal accuracy is achieved. The features obtained from the Random Forest model are passed to Deep Neural Network for classification, thus making it a hybrid model; the use of Random Forest and the LSTM model makes it hybrid. The hybrid model uses the capabilities of two algorithms. First, the features obtained from the feature selection technique are passed to the Random Forest model, which generates a score for each feature based on the impurity value. The feature from the feature selection technique is passed to the Random Forest model for further selection. Then these selected features are passed to the Deep Learning model to make predictions, thus making our model hybrid. The Random Forest algorithm can calculate the features' importance and selects them based on their score. The features which have higher value got selected for classification.

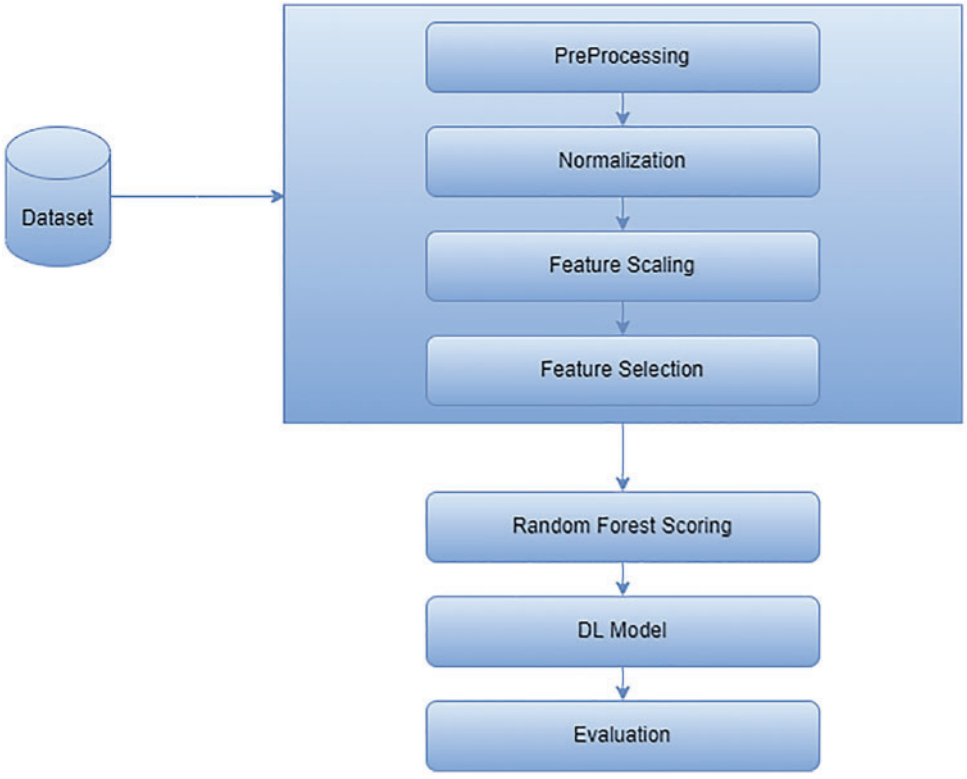

Figure 3: Proposed methodology

3.5 Regularizer

L2 regularizer gives fewer weights to the features that are not important. The drawback of using this regularizer is that it controls only weight values and does not consider the relationship between them. We proposed a hybrid regularizer to solve the problem, which combines L2 regularizer with Dropout. We are trying to solve the problem of overfitting and underfitting. Ensemble neural network solves the problem of overfitting, but it requires a lot of computational power and additional cost. The dropout method randomly drops some nodes during the model's training. This is more efficient and consumes less power and resources. We are combining L2 regularizer with Dropout regularizer and making it a hybrid regularization technique. By combining regularization and Dropout, the performance of the model got enhanced. The results obtained by this technique will be depicted in the results section.

Fig. 4 is about the combination of Deep Learning model with the regularization technique. Two regularization techniques are combined in this research to tackle the issue of overfitting faced by the DL models.

Figure 4: Deep learning with hybrid model

In Fig. 5, the neural network model is presented before applying the regularization technique. Here every node is connected to other neurons. The input nodes are connected to every other neuron in the hidden layer, and every hidden layer is then connected to the output layer. In this way, all the neurons participate in the model development. Fig. 6 illustrates the neural network model after applying the Dropout regularization technique. Here we drop some nodes during training. Random Forest is also one of the most important algorithms used in machine learning. It is a supervised learning algorithm built on an ensemble of decision trees. The bagging method is used to train Random Forest. In the bagging method, learning models are combined to give higher accuracy.

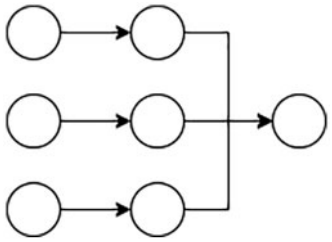

Figure 5: Before regularization

Random forest is also known as an ensemble classifier, as illustrated in Fig. 7. The decision tree with majority voting is selected for prediction. The random forest produces higher accuracy because it contains many decision trees. After the feature selection technique, data is fed to the model in our experimentation process, and optimal accuracy is achieved. Several advantages of random forest are discussed in [32]. In our proposed technique Random Forest scoring method is used for feature scoring. The features obtained from the feature selection technique are passed to the Random Forest model to calculate the feature importance. For classification problems, Gini impurity or information gain is

used. If the impurity decreases, then the feature importance is more. The average of impurity can be taken to determine the feature's importance.

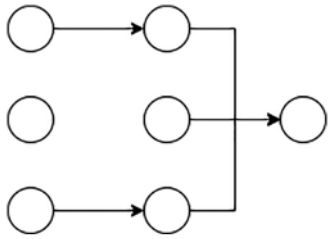

Figure 6: After regularization

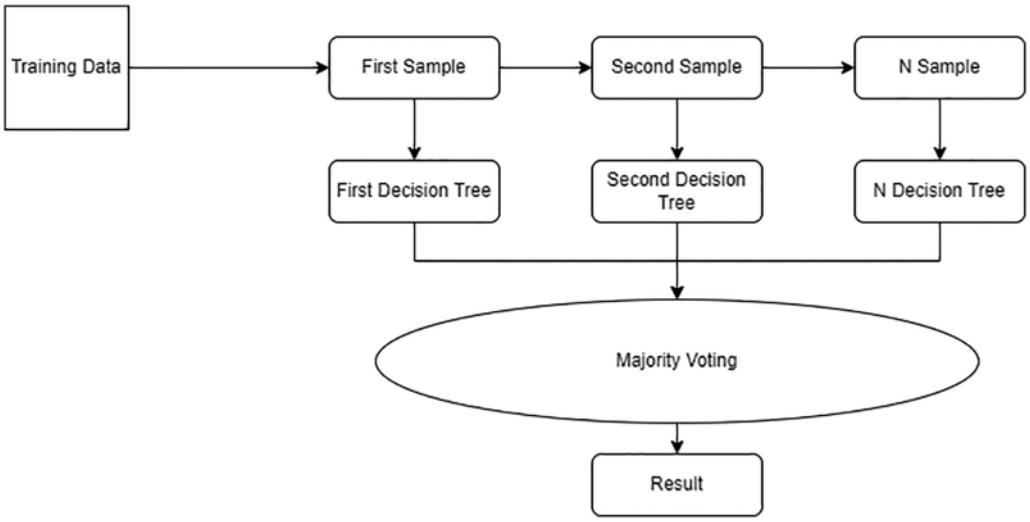

Figure 7: Random forest model

Convolutional Neural Network (CNN): CNN is one of the most important algorithms in Deep Learning, specifically for high-dimensional and image data. It also works well with numerical data. There are several steps involved in CNN like pooling, convolution, full connection, and output. CNN do automatic feature selection by using filters, thus making it efficient for any kind of data (illustrated in Fig. 8).

Recurrent Neural Networks (RNN): A recurrent Neural Network is a Deep Learning-based d algorithm that is a generalization of a feed-forward neural network. RNN has an internal memory that saves states in its memory. It is recurrent because it performs the same function for every input, and the output is dependent on the past function. RNN suffers from a vanishing gradient problem.

Long Short Term Memory (LSTM): LSTM is a special kind of RNN. LSTM is capable of learning long-term dependencies. The problem that RNN faces is solved using the LSTM model. they remember information for a very long time. LSTM has a chain-like structure. Many RNNs is combined to make the LSTM model. Gates are used in LSTM to remember the past state. Gates are composed of a sigmoid activation function, and a function is known as pointwise multiplications.

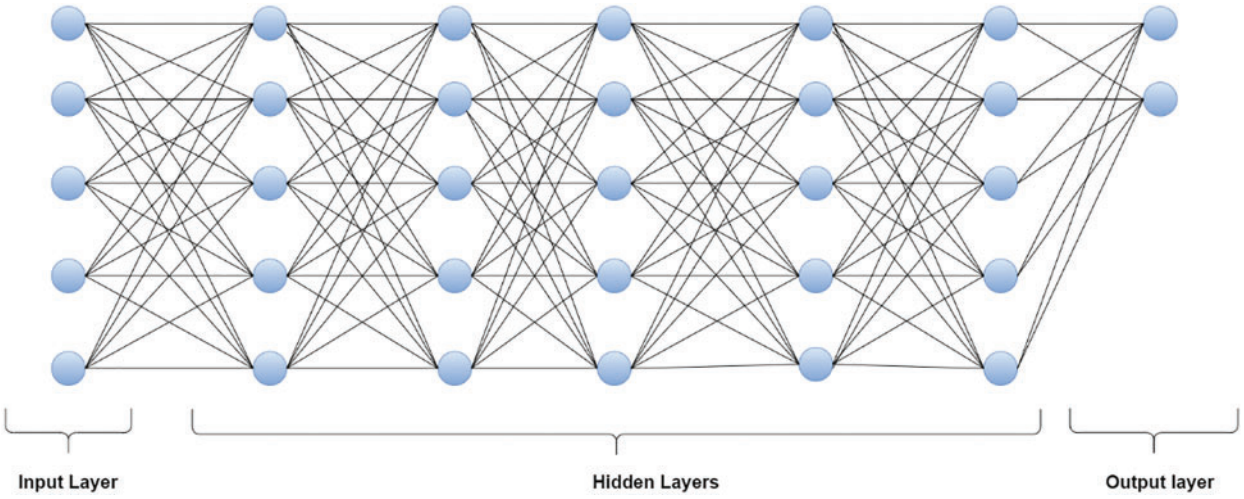

Figure 8: A simple CNN architecture

The equation of RNN is described below:

$$I_t = \sigma\,(W_i\,[h_{t-1},\, X_t] + b_i) \tag{1}$$

$$F_t = \sigma\,(W_f\,[h_{t-1},\, X_t] + b_f) \tag{2}$$

$$O_t = \sigma\,(W_o\,[h_{t-1},\, X_t] + b_o) \tag{3}$$

Above are the equations for LSTM gates. In these equations, I_t represents the input gate F_t represents the forget gate. O_t represents the output gate. σ represents the sigmoid function. W_x is the weight of the neurons. X_t is the input at the current timestamp, and b_x is the biases of the respective gate.

Different gates are present in LSTM for memory purposes (Fig. 9). These gates save the states of different cells. Mostly RNN is used for sequence modeling.

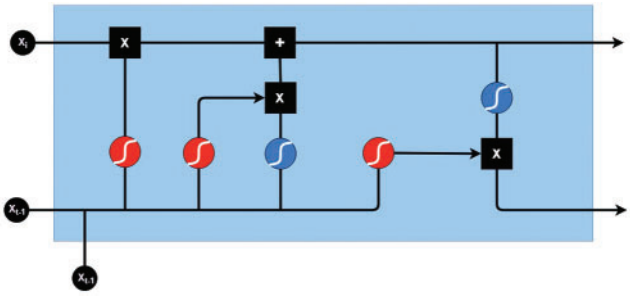

Figure 9: LSTM architecture

Feature selection is one of the most important tasks in the machine learning domain. The selection of wrong features leads to poor accuracy, so it is necessary to select optimal features for the model. The Pearson coefficient correlation technique is used in our model Pearson correlation coefficient measures the relationship between two values and variables. It ranges the values between -1 and $+1$. 0 indicates no relationship, $+1$ indicates a positive correlation, and -1 indicates a negative correlation among variables. It tells about the strength and direction of the relationship among variables by calculating

the variance and covariance. Only twenty features are selected for the training purpose in our proposed methodology. The features selected using the Pearson correlation coefficient are described in Table 3. These are all the features used in training the machine learning model. These features are then given to the Random Forest model for further selection. These features are then selected based on the Random Forest feature scoring method. This technique solves the issue of feature selection as we are using two techniques for efficient feature selection. The correlation score of every feature is given in Table 1. All the features have a positive correlation and help determine the network attacks; let's take an example of src port; its correlation is 0.323, and this feature is helpful in determining the network attack. The features which have a high impact on training are positively correlated, as seen in Table 4 (the calculated performance measures).

Table 3: Selected features

Feature	Score
Fwd Seg Size Avg	0.203
Fwd Pkt Len Mean	0.2
Bwd Pkt Len Max	0.20
Fwd IAT Mean	0.208
Flow Pkts	0.212
Bwd Pkts	0.212
Fwd IAT Std	0.225
Fwd Pkt Len Max	0.233
Pkt Len Max	0.264
Fwd IAT Max	0.264
Bed Pkt Len Std	0.267
Fwd Pkt Len Min	0.284
Fwd IT Tot	0.308
Pkt Len Std	0.311
Src Port	0.323

Table 4: Evaluation metrics

Metric	Formula	Definition
Accuracy	$TP + TN/TP + TN + FP + FN$	Overall accuracy of the model.
Precision	$TP/TP + FP$	The correctly predicted abnormal network traffic ratio to the total abnormal network traffic.
F-score	$2 * TP/2 * TP + FP + FN$	The accuracy of the model on the whole dataset.
Recall	$TP/TP + FN$	

4 Results

To experiment, python is used as a programming language with the In SDN dataset. The system consists of 16 GB RAM, an intel core processor of 2.30 GHz, and four logical processors. In the first phase, ten features selected by the feature selection technique are used for training. A 70:30 split ratio

is set for training and testing purposes. The accuracy obtained by different machine learning models is shown below.

It is evident from the Table 5 that the accuracy, precision, recall, and f1 score of our proposed technique is much better than the existing technique. The proposed technique's effectiveness lies in using the hybrid model and combined regularization technique. Accuracy alone cannot depict the model's effectiveness; we also used other parameters like precision, recall, and f1 score.

Table 5: Accuracy based analysis

Technique	Accuracy	Precision	Recall	F1-score
Hybrid + combined regularization	99.4%	99.3%	98%	99%
Hybrid	99%	99%	97%	98%

Several experiments are performed by changing the layers and epochs of the model. It is evident from the results given in Table 6 that the model cannot over fit or under fitted by increasing the hyperparameters. Optimal accuracy is achieved at each iteration.

Table 6: Layer wise accuracy

Epochs + layers	Accuracy	Precision	Recall	F1 score
50, 50	99.4%	99.3%	98%	99%
100, 50	99.6%	99.7%	99.8%	99%
125, 75	99.8%	99.7%	99.8%	99.8%
140, 90	99.8%	99.8%	99%	99%
150, 100	99.8%	99.8%	99.3%	99%

In Table 6 accuracy of the model with respect to number of layers are described. Accuracy of the Deep Learning model increased by increase in the number of hidden layers. Epochs and layers are important in determining the accuracy of the Deep Learning model. in order to achieve better results, the number of epochs are always set to optimal level.

4.1 Model Curves

In Deep Learning problems, model loss and accuracy curves depict the model's overall performance with respect to increase or decrease in hidden layers and several parameters (Fig. 10). Fig. 10 is the accuracy plot of the proposed model. It is evident from the figure that the accuracy of the model gets increases with the increase in the number of epochs. During the initial phases of the model training, the model's accuracy is low, but with the increase in the number of epochs, the accuracy increases. When dealing with deep learning problems, accuracy *vs.* epoch curves is an important parameter to judge the model performance with the increased number of epochs (Fig. 11).

Fig. 12 is the accuracy plot of the proposed model. It is evident from the figure that the accuracy of the model gets increases with the increase in the number of epochs. During the initial phases of the model training, the model's accuracy is low, but with the increase in the number of epochs, the accuracy increases.

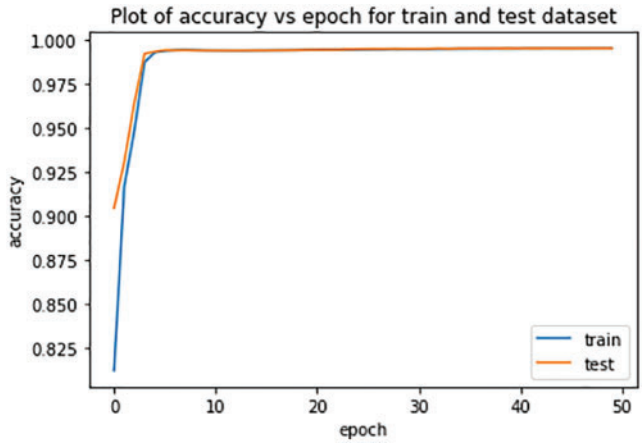

Figure 10: Accuracy curve

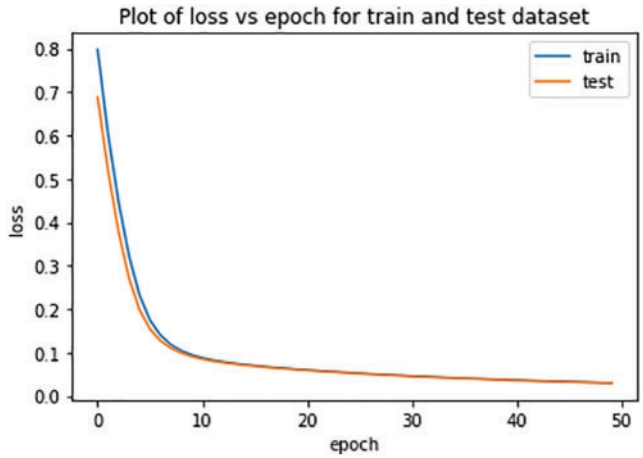

Figure 11: Loss curve

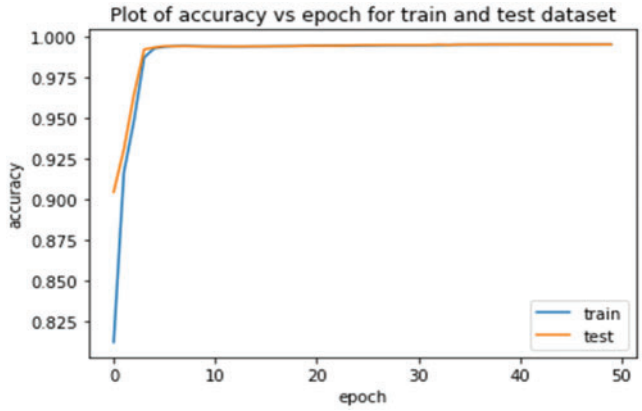

Figure 12: Proposed model accuracy curve

Fig. 13 is the model loss curve. It is evident from the figure that the model loss becomes low with the increase in the number of epochs. During the initial phases, the model loses more, but loss decreases as the number of epochs increases. The loss of the model is nearly zero at epoch 50.

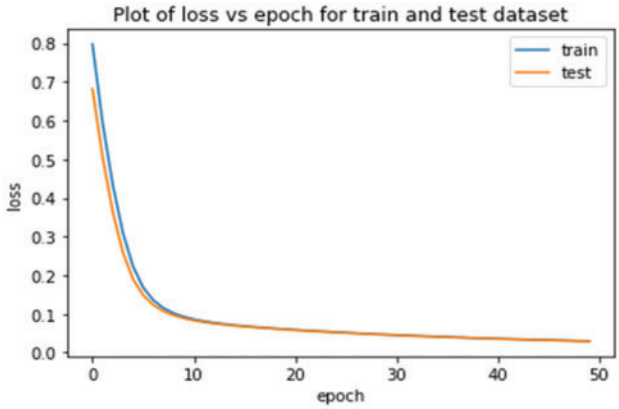

Figure 13: Proposed model loss curve

Fig. 10 is about the accuracy curve of the proposed model. As the number of epochs increases the accuracy of the model also increases. The number of epochs is set to an optimal level to get optimal accuracy. Sometimes more epochs lead to overfitting problems in Deep Learning tasks.

4.2 ROC Curves

The Roc curve is an important parameter in classification problems. It plots the results for true positive rates and true negative rates. The Roc curve of models is shown in Fig. 14. Fig. 14 represents the Roc curves of the machine learning model. It is inferred from the curves that the proposed model works best when working with ten features. The accuracy is nearly 100. Fig. 15 represents the Roc curves of the machine learning model. It is inferred from the curves that the proposed model works best when working with ten features. The accuracy of the proposed model is nearly 100. From the results, it is concluded that the performance of our proposed model is better than the simple hybrid model in terms of accuracy, precision, and recall.

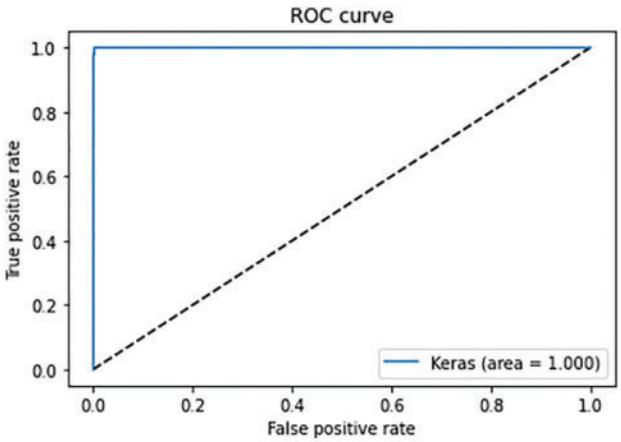

Figure 14: ROC curve

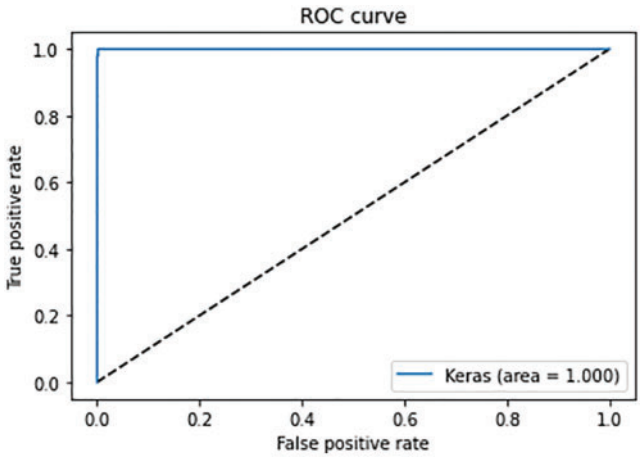

Figure 15: Proposed model ROC curve

4.3 Comparison

This section discusses our model performance with other algorithms like CNN, MLP, and RNN. The model's performance is evaluated based on accuracy, precision, and recall. The accuracy, precision, recall, and f1 score obtained by different algorithms are depicted in the table. It is evident from the table that the accuracy achieved by our proposed model is more than the other state-of-the-art algorithms. Our proposed methodology is also doing good in terms of precision. Recall and f1 score make it a good and feasible candidate for intrusion detection systems (as given in Table 7).

Table 7: Comparison with other models

Algorithm	Accuracy	Precision	Recall	F1 score
Proposed	99%	99%	99%	98%
MLP	98%	98.4%	98%	97%
RNN	99.2%	99.2%	99%	97%

We also examine the efficiency of our proposed models by comparing the test and training running time. Table 5 provides information about the parameters. The training time of CNN and MLP is more, but they also have less accuracy. In [33], unsupervised learning is used for anomaly detection. The model takes about 7.16 h to train, which is a very high time. In that respect, our model cost is very low. It is trained on only six features with 62,808 samples.

Table 8 describes about the training time of the proposed and other DL algorithms. It is evident from the table that the training time of the proposed model is low as compared to other algorithms. This training time is calculated by %time python function which calculates the time of the algorithm. Feature selection plays a very important role in getting lower training and testing time.

Table 8: Performance evaluation

Algorithm	Training time
Proposed	1 min 14 s
CNN	2 min 10 s
MLP	1 min 20 s

4.4 Discussion

Table 9 shows the studies on DDoS and intrusion detection systems using machine learning and Deep learning. It is evident from the table that different datasets are used to detect intrusion and DDoS attacks. In these studies, public datasets are used for evaluation. Mostly used datasets are NSL-KDD, KDD-CUP99, CICIDS 2017, CAIDA, and CIC-DDOS. The performance of machine learning algorithms is evaluated on these datasets. These all are the standard datasets used for intrusion detection. The increasing number of intrusions and cyber-attacks needs up-to-date datasets. For this reason, researchers used SDN datasets for their research. The dataset used in this research is NSL-KDD, a refined version of KDD.

Table 9: Comparison with other work

Datasets	Feature selection	Algorithm	Accuracy
NSL-KDD [24]	KPCA	SVM	98%
NSL-KDD [22]	No	K-means and KNN	98%
CICDOS [23]	RFE	Decision tree	98%
CIC-RN [20]	No	CRNN	97%
UNSW-NB15 [21]	No	SVM, NB, KNN	95%

Results and experiments show that machine learning and Deep Learning models successfully determine network attacks. Our research work aims to contribute to this field. Experimental results show that the proposed feature selection technique successfully determines the attacks as it selects those features from the datasets that are important and have higher weights than others. However, this is not the best feature selection technique, but it guarantees the selection of the optimal number of features for attacks like DoS and DDoS that needs to be investigated or mitigated without wasting any time. Therefore, the most effective features should be selected when creating a machine learning model.

It can also be seen from the table that the accuracy of the machine learning model is better than other studies, so it can be said that machine learning models contribute positively to the detection of machine learning attacks.

5 Conclusion

Traditional networking is less adaptable and scalable than SDN. As a result, the market share of SDN is rapidly increasing. Despite its many advantages over traditional networking, SDN is vulnerable to a variety of security threats such as DoS and DDoS attacks. As a result, the IDS for attack detection must be improved. In this study, the problem is addressed by proposing a hybrid machine learning

technology as well as a hybrid regularization technique. This is the goal of the IoT environment. To address the risk of over fitting and under fitting, a hybrid regularization technique is proposed; additionally, the problem of feature selection is solved using a hybrid technique and feature selection techniques. Experiments show that the proposed model enhances performance and efficiency. We will perform all these scenarios on other standard datasets in the future.

Funding Statement: This work was supported by "Human Resources Program in Energy Technology" of the Korea Institute of Energy Technology Evaluation and Planning (KETEP), granted financial resources from the Ministry of Trade, Industry & Energy, Republic of Korea (No. 20204010600090).

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

References

[1] M. C. Dacier, S. Dietrich, F. Kargl and H. König, "Network attack detection and defense: Security challenges and opportunities of software-defined networking," *Dagstuhl*, vol. 2, no. 1, pp. 1–6, 2016.
[2] J. Wan, D. Li and A. Vasilakos, "Security in software-defined networking: Threats and countermeasures," *Sensors*, vol. 1, no. 4, pp. 1–21, 2016.
[3] S. Moraboena, G. Ketepalli and P. Ragam, "A deep learning approach to network intrusion detection using deep autoencoder," *Artificial Intelligence Review*, vol. 34, no. 4, pp. 457–463, 2020.
[4] E. C. Ogu, O. A. Ojesanmi, O. Awodele and S. Kuyoro, "A botnets circumspection: The current threat landscape, and what we know so far," *Information*, vol. 10, no. 2, pp. 337, 2019.
[5] S. Sambangi and L. Gondi, "A machine learning approach for DDoS (distributed denial of service) attack detection using multiple linear regressioN," *Information*, vol. 1, no. 1, pp. 41–51, 2020.
[6] W. Nazih, W. S. Elkilani, H. Dhahri and T. Abdelkader, "Survey of countering DoS/DDoS attacks on SIP based VoIP networks," *Electronics*, vol. 9, no. 2, pp. 1827, 2020.
[7] T. Horak, P. Strelec, L. Huraj, P. Tanuska and M. Kebisek, "The vulnerability of the production line using industrial IoT systems under ddos attack," *Electronics*, vol. 10, no. 6, pp. 381, 2021.
[8] B. Yong, W. Wei, K. C. Li and Q. Zhou, "Ensemble machine learning approaches for webshell detection in internet of things environments," *Transactions on Emerging Telecommunications Technologies*, vol. 7, no. 2, pp. 1–12, 2020.
[9] H. Gajjar and Z. Malek, "A survey of intrusion detection system (IDS) using OpenStack private cloud," in *2020 Fourth World Conf. on Smart Trends in Systems, Security and Sustainability (WorldS4)*, Beijing, China, pp. 162–168, 2020.
[10] C. M. Hsu, M. Z. Azhari, H. Y. Hsieh and S. W. Prakosa, "Robust network intrusion detection scheme using long-short term memory based convolutional neural networks," *Mobile Networks and Applications*, vol. 26, no. 3, pp. 1137–1144, 2021.
[11] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi and M. Ghogho, "DeepIDS: Deep learning approach for intrusion detection in software defined networking," *Electronics*, vol. 9, no. 1, pp. 1533, 2020.
[12] M. Mittal, K. Kumar and S. Behal, "Deep learning approaches for detecting DDoS attacks: A systematic review," *Soft Computing*, vol. 6, no. 2, pp. 1–37, 2022.
[13] S. T. Ikram and A. K. Cherukuri, "Improving accuracy of intrusion detection model using PCA and optimized SVM," *Journal of Computing and Information Technology*, vol. 24, no. 7, pp. 133–148, 2016.
[14] F. A. Khan, A. Gumaei, A. Derhab and A. Hussain, "A novel two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, no. 3, pp. 30373–30385, 2019.
[15] M. P. Novaes, L. F. Carvalho, J. Lloret and M. L. Proença, "Long short-term memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment," *IEEE Access*, vol. 8, no. 2, pp. 83765–83781, 2020.

[16] G. Zhang, X. Wang, R. Li, Y. Song and J. Lai, "Network intrusion detection based on conditional wasserstein generative adversarial network and cost-sensitive stacked autoencoder," *IEEE Access*, vol. 8, no. 5, pp. 190431–190447, 2020.
[17] W. D. Nanda and F. D. S. Sumadi, "LRDDoS attack detection on SD-IoT using random forest with logistic regression coefficient," *Rekayasa Sistem Dan Teknologi Informasi*, vol. 6, no. 3, pp. 220–226, 2022.
[18] G. Andresini, A. Appice, N. Di Mauro and D. Malerba, "Multi-channel deep feature learning for intrusion detection," *IEEE Access*, vol. 8, no. 5, pp. 53346–53359, 2020.
[19] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman and A. Alazab, "Hybrid intrusion detection system based on the stacking ensemble of c5 decision tree classifier and one class support vector machine," *Electronics*, vol. 9, no. 2, pp. 160–173, 2020.
[20] M. A. Khan, "HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system," *Processes*, vol. 9, no. 6, pp. 834, 2021.
[21] S. Othman, F. Ba-Alwi, N. Alsohybe and A. Al-Hashida, "Intrusion detection model using machine learning algorithm on big data environment," *Journal of Big Data*, vol. 5, no. 1, pp. 1–12, 2018.
[22] H. A. Ahmed, A. Hameed and N. Z. Bawany, "Network intrusion detection using oversampling technique and machine learning algorithms," *PeerJ Computer Science*, vol. 8, no. 3, pp. e820, 2022.
[23] J. A. Perez-Diaz, I. A. Valdovinos, K. K. R. Choo and D. Zhu, "A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning," *IEEE Access*, vol. 8, no. 1, pp. 155859–155872, 2020.
[24] W. Lian, G. Nie, B. Jia, D. Shi and Y. Liang, "An intrusion detection method based on decision tree-recursive feature elimination in ensemble learning," *Mathematical Problems in Engineering*, vol. 20, no. 11, pp. 1–21, 2020.
[25] K. S. Sahoo, B. K. Tripathy, K. Naik, S. Ramasubbareddy and B. Balusamy, "An evolutionary SVM model for DDOS attack detection in software defined networks," *IEEE Access*, vol. 8, no. 6, pp. 132502–132513, 2020.
[26] A. O. Alzahrani and M. J. Alenazi, "Designing a network intrusion detection system based on machine learning for software defined networks," *Future Internet*, vol. 13, no. 5, pp. 111, 2021.
[27] D. Raumer, L. Schwaighofer and G. Carle, "Monsamp: A distributed sdn application for qos monitoring," in *2014 Federated Conf. on Computer Science and Information Systems*, NY, USA, pp. 961–968, 2014.
[28] H. C. Huang, P. S. Tsai, N. T. Hu and Z. Q. Yang, "Adaptive fuzzy cmac design for an omni-directional mobile robot," in *2014 Tenth Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, NY, USA, pp. 839–843, 2014.
[29] L. Rashid, S. Rubab, M. Alhaisoni, A. Alqahtani and S. Alsubai, "Analysis of dimensionality reduction techniques on internet of things data using machine learning," *Sustainable Energy Technologies and Assessments*, vol. 52, no. 2, pp. 102304, 2022.
[30] A. Azhar, S. Rubab, M. M. Khan, Y. A. Bangash and M. D. Alshehri, "Detection and prediction of traffic accidents using deep learning techniques," *Cluster Computing*, vol. 3, no. 1, pp. 1–17, 2022.
[31] M. S. Elsayed, N. A. Le-Khac and A. D. Jurcut, "InSDN: A novel SDN intrusion dataset," *IEEE Access*, vol. 8, no. 2, pp. 165263–165284, 2020.
[32] J. K. Jaiswal and R. Samikannu, "Application of random forest algorithm on feature subset selection and classification and regression," in *2017 World Congress on Computing and Communication Technologies (WCCCT)*, NY, USA, pp. 65–68, 2017.
[33] R. Braga, E. Mota and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," *IEEE Local Computer Network*, vol. 5, no. 3, pp. 408–415, 2010.