International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *15 August 2023* ***Revised:*** *12 October 2023* ***Accepted:*** *20 October 2023*

Cloud-based Near Real-Time Multiclass Interruption Recognition and Classification using Ensemble and Deep Learning

Minhaj Khan[1], Mohd. Haroon[2]
[1]Department of Computer Science and Engineering
Integral University
Lucknow-226026,Uttar Pradesh, India
minhajkhan7786@gmail.com
[2]Department of Computer Science and Engineering
Integral University
Lucknow-226026,Uttar Pradesh, India
mharoon@iul.ac.in

Abstract—Due to speedy development in internet facilities, detecting intrusions in a real-time cloud environment is challenging via traditional methods. In this case, advanced machine or deep learning methods can be efficiently used in anomaly or intrusion detection. Thus, the present study focuses on designing and developing the intrusion detection scheme using an ensemble learning-based random forest method and deep convolutional neural networks in a near real-time cloud atmosphere. The proposed models were tested on CSE-CICIDS2018 datasets in Python (Anaconda 3) environment. The proposed models achieved 97.73 and 99.91 per cent accuracy using random forest and deep convolutional neural networks, respectively. The developed models can be effectively utilised in real-time cloud datasets to detect intrusions.

Keywords-Random Forest, Ensemble learning, Deep convolutional neural network, Intrusion detection, Cloud environment, CSE-CICIDS2018 dataset.

I. INTRODUCTION

With the fast-growing of internet usage and the growing dependence on computer networks for various tasks, securing these networks against cyber threats has become crucial [1]. Cyber threats, such as network intrusions, pose significant risks to data and services' confidentiality, integrity, and availability. Intrusion Detection Systems (IDS) play a serious role in protection networks by detecting and alerting against potential malicious activities [1], [2]. The key objective of IDS is to see strange behaviours in network traffic that may indicate potential security breaches [3].

Traditional IDS methods, which rely on rule-based or statistical techniques, may need to be more effective in detecting complex and evolving cyber threats [1], [3], [4]. In this case, advanced artificial intelligence-based approaches such as machine and deep learning are playing an essential role in detecting anomalies in the form of intrusions from complex cloud sources [1], [2], [5]. In addition, these machine and deep learning methods improve the accuracy rate and efficiency in detecting intrusions. However, machine learning needs extra effort to extract the features and train the models. In this case, deep learning models can automatically learn patterns and elements from large amounts of data, making them well-suited for intrusion detection tasks [1], [5].

Recently, machine and deep learning models such as decision trees [6], random forests, support vector machines, naïve Bayes classifiers [7], k-nearest neighbour method, CNNs, Recurrent Neural Networks (RNNs) or hybrid methods have shown significant possibility in intrusion detection studies [1], [2]. These methods can accurately detect abnormalities and automatically learn complicated patterns from enormous quantities of data. This enables them to adjust and develop to varying threat landscapes, making them well-suited for active and evolving cyber threat systems [4], [6].

Several investigations have been performed on the machine and deep learning in interruption discovery. For instance, the authors [6] worked on network intrusion detection using a decision tree classifier and achieved 99.42% and 98.80% accuracy with NSL-KDD and CICIDS2017 datasets, respectively. Similarly, the work conducted by [8] uses SVM, Naïve Bayes and ANN algorithms on KKDDCUP99 and NLSKDD datasets with a maximum of 95 per cent accuracy. Another study by [9] used a fuzziness-based semi-supervised learning method via ensemble learning (FSSL-EL) to detect network intrusion on the NSL-KDD dataset with a maximum of 84% accuracy. Furthermore, the studies [4], [10], [11], [12] have also used several machine learning methods on NSL-

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *15 August 2023* ***Revised:*** *12 October 2023* ***Accepted:*** *20 October 2023*

KDD datasets with some limits in the form of less accuracy or binary classification.

Conversely, the research is recently more focused on using deep learning methods for detecting intrusions. For instance, the Deep Neural Networks (DNN) and Long Short-Term Memory (LSTM) models were applied to the UNSW-NB15 dataset with 95 and 88 per cent accuracy [13]. Another study by the authors [14] utilized CNN-based IDS for detecting network attacks. The model was trained on a UNSW-NB15 dataset and achieved 95% accuracy in detecting intrusions, showcasing the potential of deep learning in seeing specific types of cyber threats. In addition, research has also been conducted on using DNN for intrusion detection. For instance, a real-time intrusion detection scheme by [2] is proposed using DNN on the NSL-KDD dataset with 81% correctness.

Overall, the literature on intrusion detection via machine and deep learning highlights the potential of these models in improving the accuracy and efficiency of IDS. The capability of deep learning measures to automatically learn patterns and representations from data, adapt to changing threat landscapes, and perform feature extraction and anomaly detection make them promising approaches for enhancing the capabilities of IDS in detecting complex and evolving cyber threats [1]. Further research in this field can contribute to developing more robust and effective IDS for safeguarding computer networks against intrusion attempts.

However, earlier studies either focused on the machine or deep learning methods on simple datasets for binary classification. The data sets used in previous studies are too small. In addition, this earlier research needs to show more accuracy for complex datasets like CSE-CICIDS2018 with multiclass labels. Furthermore, as mentioned earlier, the analysis must provide the highest accuracy with less training or testing time. Therefore, the present study focuses on designing and implementing machine and deep learning models to spot multiple intrusions from a composite near real-time cloud environment.

The primary aims are (1) to collect and pre-process near real-time data, (2) to extract and select the essential features, (3) to develop the machine and deep learning models using random forest and deep convolutional neural networks models, (4) to estimate the performance of the presented models, and (5) to compare the present studies results with standard literature. We will also highlight the potential benefits of using deep convolutional neural networks in intrusion detection, such as improved accuracy, scalability, and adaptability to changing threat landscapes. This research's results can contribute to developing more robust and effective IDS for securing computer networks against cyber threats.

We organize the rest article in the following order. Section II shows the detail of the CSE-CICIDS2018 dataset. Section III focuses on the proposed and implemented methodology. In Section IV, we designed an experiment to authenticate the success of our process and discussed the results obtained from the proposed methods. Finally, Section V concludes the present study.

II. THE DATASETS

The near real-time CSE-CICIDS2018 dataset [15] is used in the present study to detect the intrusion, which was collected at the University of New Brunswick (UNB)by the Canadian Institute for Cybersecurity (CIC) [15]. The dataset contains network traffic that was captured in a simulated enterprise environment.

The "Wednesday-14-02-2018_TrafficForML_CICFlowMeter" is one of the files included in the dataset. It has network traffic in the form of flow records generated by the CICFlowMeter software. These flow records offermaterial about the communication between network hosts, including source and destination IP addresses, port numbers, protocol types, and the number of packets and bytes exchanged. The dataset includes benign and malicious traffic, and the intrusions are classified into different categories based on the type of attack [15].

III. THE PROPOSED METHODOLOGY

The proposed and implemented methodology contains data pre-processing, feature selection for ensemble learning, and model development with the evaluation of the results. The roadmap of the projected method is represented in Figure 1.

A. Data pre-processing

In the pre-processing, we eliminated the unwanted or duplicate and null or zero data and cleaned, corrected any errors in data, and prepared the data before it was used for analysis. Some missing values were also removed from the data. Latterly, the dataset is converted into the float data format for further processing [5].

B. Feature Selection

Selecting essential features is critical to building a predictive model for detecting network intrusions [5], [16]. Therefore, we set the most significant features from the Wednesday-14-02-2018_TrafficForML_CICFlowMeter dataset using information gain, gain-ratio, and chi-squared methods. These methods have provided nineteen relevant components from a total of seventy-nine.

C. Ensemble-based Machine Learning Random Forest Classifier

Random Forest is a popular ensemble learning method often used in intrusion detection systems because it can handle large amounts of information and detect patterns that may be difficult for other algorithms to identify [3]. The basic idea of a Random Forest is to build multiple decision trees, each trained on a random subset of the input data and a random subset of the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *15 August 2023* ***Revised:*** *12 October 2023* ***Accepted:*** *20 October 2023*

input features. The algorithm's output combines the predictions of all the individual decision trees. The random nature of the algorithm helps to reduce over fitting, which can occur when a machine learning algorithm is trained too closely on a specific set of data and needs to generalize better to new data [3], [16]. In this study, we built an ensemble of decision trees to handle large and complex data and identify the patterns. We generated 1000 decision trees with forty-two random states. The 'sklearn' library was imported to build the random ensemble forest classifier.

D. Deep convolutional neural network

The CNNs are a popular type of neural network that has been effectively applied to datasets. The CNN architecture is defined using the number of layers and the activation functions [1], [5]. For the CSE-CICIDS2018 Wednesday-14-02-2018 dataset, the input data consists of network traffic features. Therefore, the CNN architecture involved the 1D convolutional layers, pooling, and fully connected layers. An output layer is several nodes equal to the number of classes in the dataset, with each node representing a different class. The activation function utilized in the output layer was ReLU. The sequential 1D model is developed with 64 filters, six kernel sizes, three max-pooling layers, the same padding, and flattened layers. The proposed CNN algorithm is provided here.

Proposed Algorithm for CNN

Input: Training data D, validation data V, number of epochs E, learning rate alpha, batch size B Initialize weights and biases of the CNN
For e = 1 to E:
Shuffle and split the training data into batches of size B
For each batch of training data:
Forward propagate the data through the CNN
Compute the loss using a suitable loss function (e.g., cross-entropy)
Backpropagate the loss and update the weights and biases using gradient descent with learning rate alpha
End for
Evaluate the performance of the CNN on the validation data
End for
Output: Trained CNN model.

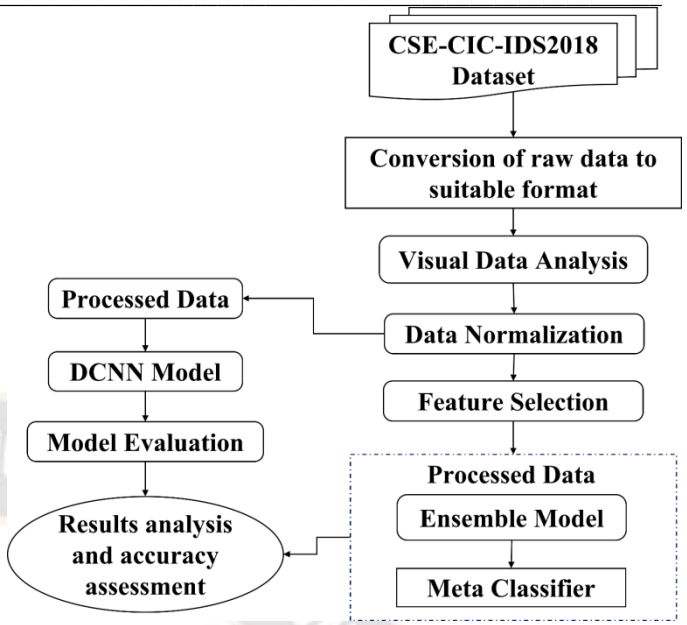

Figure 1. The workflow of the proposed methodology.

E. Performance evaluation of the proposed models

The proposed models' performances were evaluated using a confusion matrix, 10-fold cross-validation, and metrics such as accuracy, precision, recall, and the F1-score implemented via Eq.'s (1), (2), (3), and (4), respectively [2], [16].

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

$$\text{F1 score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

Where TP, FP, TN, and FN are truly positive, predict one, and the actual is also one. A false positive indicates one, but the essential is zero. True negative predicts zero, and actual is also zero, and false negative predicts zero, but really is one, correspondingly [2], [16].

IV. RESULTS AND DISCUSSION

The proposed methods were implemented on Windows 11 Home 22H2 operating systems, AMD Ryzen 5-5500U, Radeon Graphics-2.10 GHz with 16 GB RAM and Python 3.8 version. The raw data were pre-processed, and feature selection approaches such as information gain, gain-ratio, and chi-squared [16] were applied to the processed data. In the original processed data, seventy-nine total features were present. Though, all these features could have been more helpful for intrusion detection. Thus, nineteen prominent features were

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *15 August 2023* ***Revised:*** *12 October 2023* ***Accepted:*** *20 October 2023*

selected and provided to the proposed models. In the original data, three attacks, such as Benign, FTP-Brute Force, and SSH-Brute Force, were present.

The ensemble learning-based random forest method [3] was first implemented with the selected features. The dataset is divided into seventy per cent for training and thirty per cent for testing purposes, and then the models were trained and tested accordingly. The random forest model has trained on the training data with 1000 estimators (decision trees) and forty-two random states. The random forest model took 13 minutes and 2.7 minutes for training and testing, achieving 98.37 and 97.72 per cent accuracy on training and testing data, respectively. Table 1 highlights the accuracy, precision, recall, and F1-score performed on training and testing datasets using the random forest method.

TABLE I. THE PERFORMANCE RESULTS FOR TRAINING AND TESTING DATASETS USING THE RANDOM FOREST METHOD

Performance metrics for random forest	Training Results (%)	Testing Results (%)
Accuracy	98.28	97.16
Precision	98.29	97.67
Recall	97.99	98.93
F1-Score	98.98	97.13

The random forest method applied to training data has achieved the highest accuracy of 98.28%, precision of 98.29%, recall of 97.99%, and f1-score of 98.98%, respectively. The random forest model predicted superior accuracy of 97.16%, a precision of 97.67%, a recall of 98.93% and an f1-score of 97.13% using the testing dataset (Table 1). The confusion matrix (Figure 2) is also generated from the random forest algorithm for evaluating the correctly and incorrectly classified features through testing. It is observed from the confusion matrix that the classes are wrongly identified with each other (Figure 2) with some limits through the random forest method. There are 300 instances mismatched with the Benign type.

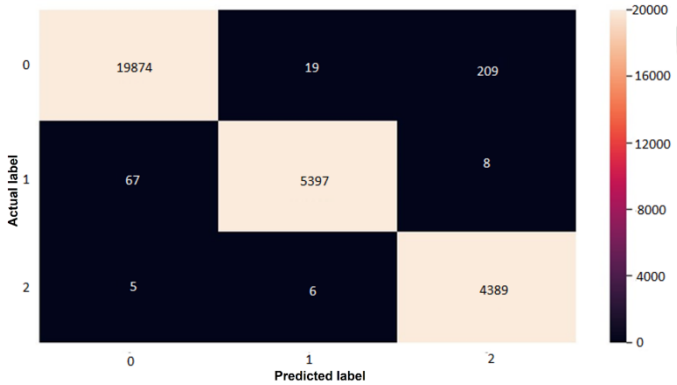

Figure 2. Confusion matrix of random forest method implemented on the testing dataset.

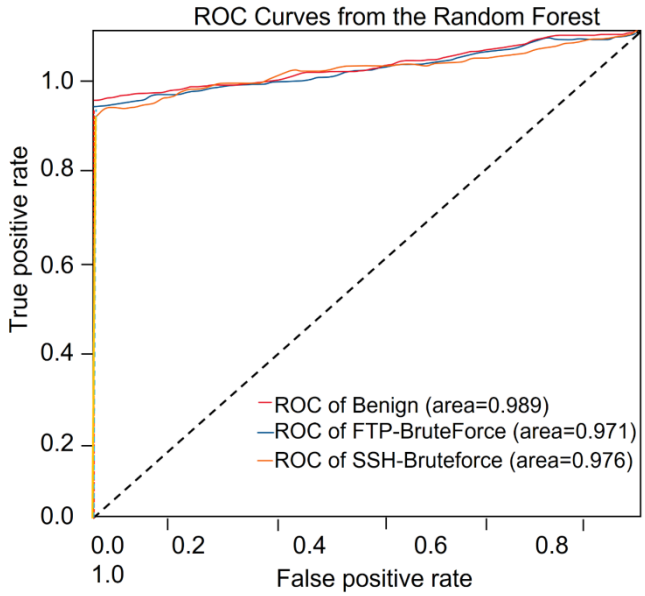

Figure 3. The ROC curve for the random forest method.

The receiver operating characteristic (ROC) is demonstrated in Figure 3, generated using the random forest method. The ROC is a 2-D curve of the random forest model that shows the graphical representation of the classification results and the various attacks from the datasets. It is seen from the ROC curve (Figure 3); all the classes have accurate classes with satisfactory performances.

On the other hand, the deep convolutional neural network has been proposed and implemented using a similar dataset and achieved more than 99 per cent accuracy on training and testing datasets. In this study, the set hyper parameters for the CNN model are hidden nodes "72", batch size "32", loss "categorical_crossentropy", "SoftMax" activation function, and "Adam" optimizer. Like the random forest method, the CNN model is also trained on seventy per cent of datasets, and the remaining thirty per cent of data is used for testing purposes. The CNN model took 26 and 1.2 minutes for training and testing, respectively. The training time is more in CNN than in the random forest. However, the testing time is better with the highest accuracy than the random forest. The performance matrices implemented for CNN are shown in Table 2 for training and testing datasets. The CNN model's accuracy is 99.99 and 99.91 per cent on training and testing datasets, respectively, the highest accuracy for the used dataset. The precision (99.91%), recall (99.93%), and f1-score (99.98%) are also the highest on the testing data. The confusion matrix (Figure 4) derived via the CNN model shows that the correctly classified instances are more than random forests. Only eighteen specimens were wrongly classified with each other via the CNN model seems that the results are more promising and efficient than random forest. It can be noticed from the confusion matrix developed via the CNN model (Figure 4) has

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *15 August 2023* ***Revised:*** *12 October 2023* ***Accepted:*** *20 October 2023*

resulted best in detecting the correct classes of various attacks compared to the random forest model.

TABLE II. THE PERFORMANCE RESULTS FOR TRAINING AND TESTING DATASETS USING THE CNN MODEL

Performance metrics for CNN	Training Results (%)	Testing Results (%)
Accuracy	99.99	99.91
Precision	99.92	99.93
Recall	99.99	99.93
F1-Score	99.99	99.98

Figure 4. Confusion matrix of CNN model implemented on the testing dataset.

The CNN model's training, validation accuracy, and log loss are also implemented for each epoch's train and test data. These parameters were used to verify the accurately sensed classes in the testing set relied on the training information. Conversely, loss denotes the mistakenly trained representatives. In our case, the testing success rate was high, and loss was negligible (Figures 5 and 6). The loss plot shows that the inaccurate prediction is significantly less, with only five epochs. The data is trained with five epochs in the present experiment due to the limited size. The CNN model has resulted best for all the classes as multi-classification with less time to train and validate. Figure 5 shows the model's accuracy with used epochs throughout the training and testing stage. The CNN model has shown the maximum accuracy near the second and fifth epochs (Figure 5). Correspondingly, the testing loss was low near the fifth epoch. Figure 6 demonstrates the loss curve over the consecutive epochs throughout the training and testing stage.

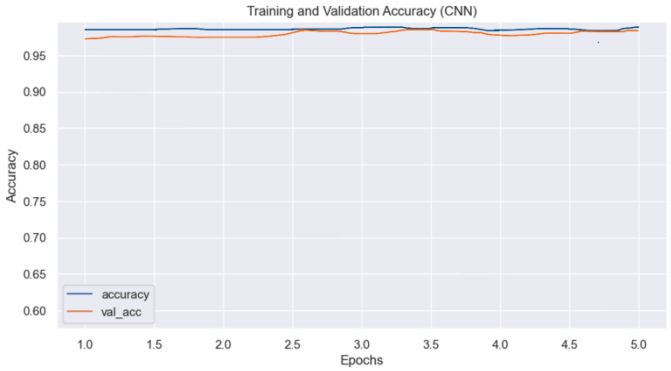

Figure 5. Training and validation accuracy graph for CNN model.

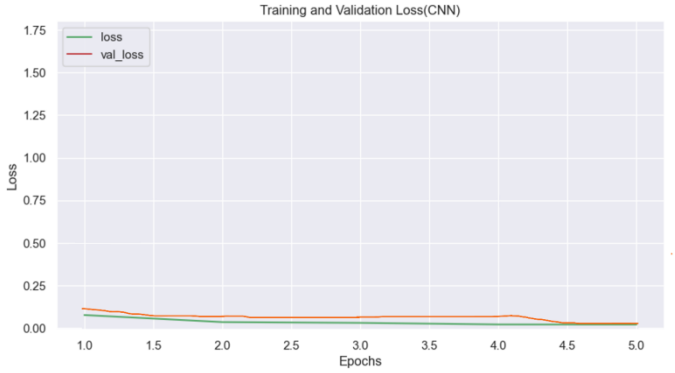

Figure 6. Training and validation loss graph for CNN model.

The random forest and CNN model have resulted accurately per their working strategy. Figure 7 compares random forest and CNN models for accuracy, precision, recall, and F1-score. The proposed CNN model resulted best in all metrics than random forest (Figure 7).

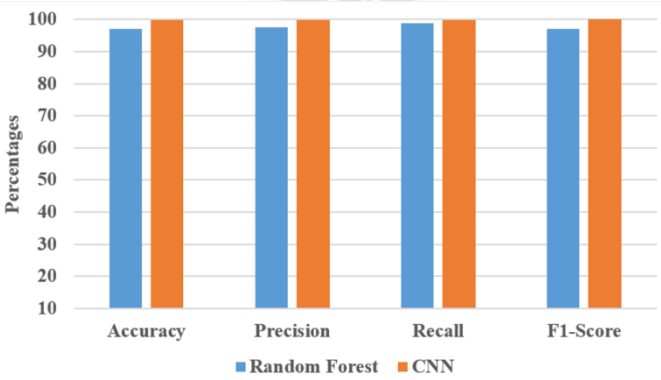

Figure 7. Comparison of the implemented models for their performances.

A. Comparative study

The present study's results are compared with earlier studies, noting that the proposed models performed well on the used dataset. Table 3 shows the proposed CNN model's accuracy and earlier studies' performance. The highest accuracy in an earlier study [13] was 95.02% for CNN and 86% for random forest implemented on UNSW-NB15 data. Similarly,

the study used UNSW-NB15 data for detecting intrusions with 94.4 per cent accuracy [14] using the CNN model. The accuracy was 99% using the random forest for the NSL-KDD dataset [17]. In addition, the study [2] focused on NSL-KDD data and the CNN model and achieved 81.87 per cent accuracy. The research achieved 97.06% accuracy based on the ensemble learning approach and 99.91% accuracy using the CNN model via the UNSW-NB-15 dataset [3]. The results achieved in the present research are better than the earlier studies for either the random forest or CNN model. Our promising models can efficiently detect intrusions in the cloud platform.

TABLE III. THE COMPARISON OF RESULTS OF THE PRESENT STUDY WITH EARLIER STUDIES

Algorithms	**Accuracy (%)**	**Reference**
Decision tree	98.90	[6]
SVM, Naïve Bayes and ANN	95	[8]
fuzziness-based semi-supervised learning	84	[9]
CNN	81.87	[2]
CNN	95.02	[13]
CNN	94.4	[14]
Random forest and CNN	**97.72 and 99.91**	**Present study**

V. CONCLUSIONS

In the present study, we proposed random forest and deep convolutional neural network models to accurately detect and predict intrusions from the real-time complex cloud environment data. Before model development, the essential nineteen features were selected from a total of seventy-nine based on the importance of the unique feature. According to the results, the present study is reliable and accurate for complex data compared to earlier studies. It is concluded that an ensemble-learning-based random forest model needs more precise and sufficient features than a deep convolution neural network. In addition, the deep convolution neural network model also provides the highest accuracy for complex real-time datasets. The proposed models are accurate and sufficient as per the results from the current study that is efficient and accurate.

ACKNOWLEDGEMENTS

This work is acknowledged under Integral University Manuscript No IU/R&D/2023-MCN0002079.

REFERENCES

[1] Lansky, J., Ali, S., Mohammadi, M., Majeed, M. K., Karim, S. H. T., Rashidi, S., ... & Rahmani, A. M. (2021). Deep learning-based intrusion detection systems: a systematic review. IEEE Access, 9, 101574-101599.

[2] Thirimanne, S. P., Jayawardana, L., Yasakethu, L., Liyanaarachchi, P., & Hewage, C. (2022). Deep neural network based real-time intrusion detection system. SN Computer Science, 3(2), 145.

[3] Shahzad, F., Mannan, A., Javed, A. R., Almadhor, A. S., Baker, T., & Al-Jumeily OBE, D. (2022). Cloud-based multiclass anomaly detection and categorization using ensemble learning. Journal of Cloud Computing, 11(1), 1-12.

[4] Megantara, A. A., & Ahmad, T. (2021). A hybrid machine learning method for increasing the performance of network intrusion detection systems. Journal of Big Data, 8(1), 1-19.

[5] Hagar, A. A., &Gawali, B. W. (2022). Apache Spark and Deep Learning Models for High-Performance Network Intrusion Detection Using CSE-CIC-IDS2018. Computational Intelligence and Neuroscience, 2022.

[6] Guezzaz, A., Benkirane, S., Azrour, M., & Khurram, S. (2021). A reliable network intrusion detection approach using decision tree with enhanced data quality. Security and Communication Networks, 2021, 1-8.

[7] Masoodi, F. (2021). Machine learning for classification analysis of intrusion detection on NSL-KDD dataset. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(10), 2286-2293.

[8] Rokade, M. D., & Sharma, Y. K. (2021, March). MLIDS: A machine learning approach for intrusion detection for real time network dataset. In 2021 International Conference on Emerging Smart Computing and Informatics (ESCI) (pp. 533-536). IEEE.

[9] Gao, Y., Liu, Y., Jin, Y., Chen, J., & Wu, H. (2018). A novel semi-supervised learning approach for network intrusion detection on cloud-based robotic system. IEEE Access, 6, 50927-50938.

[10] Abrar, I., Ayub, Z., Masoodi, F., &Bamhdi, A. M. (2020, September). A machine learning approach for intrusion detection system on NSL-KDD dataset. In 2020 international conference on smart electronics and communication (ICOSEC) (pp. 919-924). IEEE.

[11] Yedukondalu, G., Bindu, G. H., Pavan, J., Venkatesh, G., &SaiTeja, A. (2021, September). Intrusion detection system framework using machine learning. In 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 1224-1230). IEEE.

[12] Yihunie, F., Abdelfattah, E., & Regmi, A. (2019, May). Applying machine learning to anomaly-based intrusion detection systems. In 2019 IEEE Long Island Systems, Applications and Technology Conference (LISAT) (pp. 1-5). IEEE.

[13] HP, C., Nandini, P., &Honnavalli, P. (2021, August). Cloud-based Network Intrusion Detection System using Deep Learning. In The 7th Annual International Conference on Arab Women in Computing in Conjunction with the 2nd Forum of Women in Research (pp. 1-6).

[14] Ashiku, L., & Dagli, C. (2021). Network intrusion detection system using deep learning. Procedia Computer Science, 185, 239-247.

[15] https://www.unb.ca/cic/datasets/ids-2018.html.

[16] Krishnaveni, S., Sivamohan, S., Sridhar, S. S., & Prabakaran, S. (2021). Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing. Cluster Computing, 24(3), 1761-1779.

[17] Arjunan, K., & Modi, C. N. (2017, January). An enhanced intrusion detection framework for securing network layer of cloud computing. In 2017 ISEA Asia Security and Privacy (ISEASP) (pp. 1-10). IEEE.