As per UGC guidelines an electronic bar code is provided to seure your paper

International Journal for Modern Trends in Science and Technology
Volume 10, Issue 01, pages 64-71.
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol10issue01.html
DOI: https://doi.org/10.46501/IJMTST1001009

Optimizing Computer System Security Utilizing Anomaly Detection and Intrusion Aversion

Satti Pravallika[1] | D.Phani Kumar[2] | B.Sujatha[3]

[1]Department of Computer Science and Engineering, Godavari Institute of Engineering and Technology, Rajahmundry, Andhra Pradesh, India. pravallikareddy2110@gmail.com
[2]Asst.Professor,Department of Computer Science and Engineering, Godavari Institute of Engineering and Technology, Rajahmundry, Andhra Pradesh, India. phanikumar@giet.ac.in
[3]Professor, Department of Computer Science and Engineering, Godavari Institute of Engineering and Technology, Rajahmundry, Andhra Pradesh, India. birudusujatha@gmail.com

To Cite this Article
Satti Pravallika, D.Phani Kumar and B.Sujatha, Optimizing Computer System Security Utilizing Anomaly Detection and Intrusion Aversion, International Journal for Modern Trends in Science and Technology, 2024, 10(01), pages. 64-71. https://doi.org/10.46501/IJMTST1001009

Article Info
Received: 02 January 2024; Accepted: 22 January 2024; Published: 22 January 2024.

ABSTRACT

Robust security is essential to protect Cyber Systems and the Internet of Things (IoT) from Cyber-assaults. This work explores the use of datasets, including KDD-CUP, to enhance intrusion detection systems. Recurrent Neural Networks (RNN), Multi-Layer Perceptron (MLP), Restricted Boltzmann Machines (RBM), and Convolutional Neural Networks (CNN) are among the deep learning models that we used to achieve exceptional accuracy; CNN emerged as the best performer with 96%.By combining the advantages of several models, an ensemble approach was utilized to strengthen the security posture even further. In particular, a study was carried out on how well CNN - LSTM worked together across all datasets. The promising results of the proposed ensemble approach underscore its potential in more robustly and accurately identifying various types of Cyber-attacks. In order to improve anomaly detection and intrusion prevention techniques in cyber-physical systems, the study findings presented here offer a solid basis for bolsteringIoT security against the dynamic cyber threat landscape.

Keywords—KDD-CUP,CyberSystems, IoT, CNN, RNN, Anomaly detection,Intrusion prevention, Security-Attacks.

1. INTRODUCTION

With the increasing technology the massive amount of data storage is also increasing. In order to save the data a high-level security is very essential for the organizations. Even though the data which is secured is also subjected to malicious attacks in many forms. In order to protect the data an effective tool called Intrusion Detection System is being utilized [1]. The next line of defense for a system is comprised of intrusion detection systems (IDS)[2]. Through the use of unique attack-specific rules and a range of benign traffic/normal flow patterns, intrusion detection systems are able to differentiate between malicious and benign activities. Compared to classical IDS, data mining can more accurately characterize and deploy IDSs with robust behavior that can thwart sophisticated, current Cyber-attacks.Businesses which are based on IIOt are increasing and the protection of the critical source is becoming more concern especially IICS. To detect online attacks against IICS networks, a number of intrusion detection systems (IDS) arecreated.The approaches and assessment metrics used by the bulk of the current IDSs, however, have several serious shortcomings. By the help of aauto-encoder-based LSTM model, we can develop an effective IDS for IIoT-powered IIC to resolve the problems of low detection rate and High False Positive Rates (FPR). Here the goal is to improve computer network security by creating a new intrusion detection technique for cyber-physical systems that is based on deep learning. Maintaining the integrity of sensitive information at all times while achieving excellent performance in terms of true rate and detection rate is our goal.

In the realm of computer network security, the persistent menace of malicious software, computer viruses, and hostile attacks poses significant challenges. Issues with traditional intrusion detection systems include low detection capabilities, high false positive rates, low accuracy, and limited ability to adapt to new types of intrusions. This research addresses these pressing concerns by suggesting a deep-learningbased approach to identify and stopsecurity breaches & vulnerabilities in cyber systems. The main issue at hand is the requirement for an improved intrusion detection system that is more effective and efficient in protecting sensitive information and systems and providing better performance in a variety of assault scenarios[3][4][5][6].

2. LITERATURE REVIEW

Convolutional Neural Network—Case Study:

Networks like "AlexNet", "VGG", "Inception" and "ResNet" were used to refer convolutional neural networks because of their great effectiveness in classifying images. In this manner, the goal is to confirm which networks performed better in the "Imagenet" dataset challenge. After that, the "Kinetics400" and "UCF101" datasets were used to confirm their success in categorizing videos. Lastly, it was determined whether the success in classifying photos may also indicate a potential success in classifying videos.In order to achieve this, a comparison is made between the error margins of the networks previously described. After selecting and analyzing both the networks with the least margin of error, then the videos are classified using these networks. If these networks are successful, they will be able to recognize human activities in videos that are input by sensors with accuracy. It should be highlighted that the networks "ResNet" & "Inception" demonstrated performance in the chosen strategy with extremely satisfactory success rates over 70%[7][8][9].

Intrusion Detection(ID) in Internet of Things Systems(IDS): An Overview of Design Methods Using ML, Multi-Access Edge Computing, and Databases:

The (IoT) has a rapid expansion in its applications, leading to a significant rise in networks and a highly competing complexity among the linked devices. IoT gather important data that was required for the businesses and individual consumers make vital decisions thoseeffect their daily lives. The majority of the Internet of Things devices have poor CPU speeds, limited space, and minimal storage. Because those devices cannot run current general-purpose security software, they are therefore susceptible to cyber-attacks[10][13][15]. IoT networks become inherently risky as a result. By moving sophisticated computing operations from IoT devices to the edge, the Multi-Access Edge computing (MEC) platforms has arisen to alleviate those limitations. The majority of connected works currently in existence concentrate on identifying the best security ways to safeguard IoT devices[14]. We think more focus should be on distributed systems that use MEC. Modern Network Intrusion Detection Systems (NIDS) and IoT Network Security procedures are thoroughly reviewed in this study. We have examined the methods that make use of machine learning (ML) techniques and are based on MEC platforms. Additionally, a comparative examination of the assessment criteria, deployment methodologies, and datasets that are publicly available that were used in the NIDS design is performed in this research. Lastly, we suggest a MEC-based NIDS framework for IoT networks.

An Enhanced Random Forest Method with EGA-PSO Hybrid Intrusion Detection Model:

As the IT technology is having a rapid growth the digital data is widely available, leading to new security risks that require quick action. With the help of the Intrusion Detection System these unwanted intrusions and unauthorized access can be. MLtechniques are frequently applied in IDS. As a result of a small training dataset,ML-based IDS experiences problems with data imbalance and produces a greater false detection ratio. To address the problem of data imbalance, this study creates an effective Hybrid Network-Based IDS(HNIDS) model that makes use of Improved Random Forest (IRF) and Enhanced Genetic Algorithm and Particle Swarm Optimization (EGA-PSO) techniques[15][16][17][18]. A PSO approach is used in the suggested HNIDS to enhance the vector. A multi-objective function is added to GA to increase its performance. It chooses the better features and produces better outcomes to investigate important characteristics, minimizing dimensions, increasing the True Positive Rate (TPR), and reducing the false positive rate (FPR). The following step involves IRF that removes less important features, adds a list of decision trees to each iterative process, monitors the classifier's performance, and guards against overfitting problems.The NSL-KDD benchmark datasets are used to evaluate the performance of the suggested approach and current machine learning techniques.

A Tree classifier-based network intrusion-detection model for Internet of Medical Things:

One of the main potential applications of the Internet of Things (IoT) is healthcare. The Internet of Medical Things has shown tremendous growth recently, enabling improved medical services. Inspite of all advantages, cyber threats on healthcare devices that are connected have the potential to harm a patient health as well as compromise privacy. An extensive protected model is needed to guarantee patient privacy in this network because to the enormous demand for IoMT devices that provide smooth and effective medical services for a broad population. It is quite difficult to create security models for IoMT networks, always. This work aims to develop a network intrusion detection model for IoMT networks using a tree classifier approach. With a very high accuracy of 94.23%, the suggested solution efficiently minimizes the dimension of the input data to expedite the error detection process.

Drone-based data management and optimization in an optimal fog environment with blockchain smart contracts and metaheuristic algorithms:

Unmanned Aerial Vehicle (UAV)-enabled control and management of drone-based data has replaced internet hub with the development of fog computing, an extension of cloud-enabling technology. Since they are connected to one another via a wireless sensor network, main goals are to enhance the computation and processing of drone-based resource limitations and transfer the data to the outsourced computational node for scheduling, processing, managing, optimizing, and preserving it. The standard for drone system design and execution is this cooperative technical approach to resilience, which lowers compute power, resource consumption, and latency for applications requiring quick responses. However, there is a significant privacy, security, and preservation concern presented by recent research into fog-enabled drone-based data management and optimization[14]. Conversely, Blockchain Hyperledger Technology, primarily associated with Bitcoin Cryptocurrencies, has found widespread application in numerous distributed applications because of its distributed features related to protection, security, provenance, availability, ledger integrity, transparency, and trustworthiness. Drone-based data can thus be captured, scheduled, processed, optimized, managed, and preserved through collaborative processes including fog-nodes and blockchain hyperledger technology. This has led to an increased focus on the distributed application of drone control. In this research, we suggested a collaborative solution for fog node management using B-Drone, a genetic algorithm with metaheuristic support enabled by blockchain hyperledger fabric. This method manages the processing, scheduling, optimization, processing, managing, and preservation of drone-based data in the fog node securely. The SHA-256 hash-encryption algorithm ensures the privacy of every transaction between the drone and the fog node prior to exchange[16][17][18][19][20]

3. IMPLEMENTATION

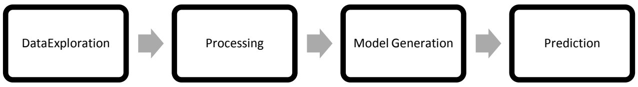


Fig 1: Implementation Process

Step.1: The data exploration module is used to import data into the system.

Step.2: The module will be used to read and process data.Data will be separated into training and tested using this module.

Datasets used:

KDD-CUP: -Since 1999, the most popular data set for evaluating anomaly detection methods has been KDD.This data collection, prepared by Stolfo et al., is built upon the data collected during the DARPA'98 IDS evaluation program[11]. DARPA'98 is the compressed raw (binary) tcpdump data, which is about 4 terabytes of network traffic from 7 weeks. After processing, this data may be divided into 5 million connection records, each with about 100 bytes. The test data collected over the course of two weeks contains nearly two million connection records. About 4,900,000 distinct connection vectors, each with 41 attributes and a label indicating whether it is a normal or an attack, with a single attack type, make up the KDD training dataset.
The datasets comprise 24 distinct attack types for training.An additional of 14 types present exclusively in the test data. The training attack types are listed in [12] and each is explained in great depth.

The three groups of KDD'99 features are as follows:
1) **Basic features**: This group includes anything that are removed from TCP/IP connection. Almost all of those traits result in an implicit detection delay.
2) **Traffic features**: Based on features that are computed with regard to a window interval, this category are divided into two groups:
a) Features labeled as "same host" compute statistics on protocol behavior, service, etc. and examine only connections established in the last two seconds with the same destination host as the connection
b) Only connections established in the recent two seconds are considered by the "same service" option.
3) **Content features:**R2L and U2R attacks do not follow any particularly frequent sequential patterns like DoS and probing attacks. We need certain parameters, like the quantity of unsuccessful login attempts, to be able to search the data section for suspicious activity in order to identify these types of assaults. We refer to these attributes as content characteristics[12].

Step.3: The model generation process involves constructing models.

CNN:

One of deep neural network called convolutional neural networks (CNNs) is made specifically to interpret structured grid data, like photographs. The fundamental principle of CNNs is the automatic and adaptive learning of spatial hierarchies of features from the input data through the use of convolutional layers. This model's operational procedures are as follows:

1. An input layer, which usually consists of one or more images, is where the process starts. Every image is represented as a grid of pixel values, with color information contained in each pixel.Convolutional operations are applied to the input image using learnable kernels. This process captures local patterns and features in the input.
2. After convolutionPooling layers are used to minimizedimensions. By keeping highest value among a set of values, max pooling, for example, efficiently down samples the feature maps while maintaining crucial information.
3. The output from the convolutional and pooling layers is flattened into a one-dimensional vector. This flattening process organizes the learned features and prepares them for input into fully connected layers.
4. One or more fully connected layers pass through the flattened vector. By identifying intricate patterns and connections between the retrieved information, these layers perform the role of classifiers.
5. In order to create probabilities for each class in image classification, the SoftMax activation function is frequently used. The class that has the highest

probability gets advanced to the training phase and is regarded as the projected class.

6. Labeled data is fed into the network during training, and a loss function is used to quantify the discrepancy between the expected and actual labels. The network's weights and biases are then modified using backpropagation and optimization techniques like stochastic gradient descent in order to minimize this loss.
7. The network modifies its parameters after every iteration in the iterative training process. The convolutional and fully connected layers' weights are updated in this way, enabling the model to improve and learn its internal representations.
8. The model can be assessed using fresh, untested data once it has been trained. An input image is passed through the trained network during inference, and the output offers predictions or classifications based on the attributes that were learned.

RNN – LSTM:

Long Short-Term recollection (LSTM) networks, in particular, are a type of Recurrent Neural Network (RNN) that manage sequential data by retaining a recollection of previous information. The model can update, discard, and selectively store data over a sequence thanks to the gates in the memory cell. In order to reduce the discrepancy between expected and actual outputs, the network's parameters are modified during the training phase. This model's operational procedures are as follows:

1. Sequential data, like time series or natural language, is a good fit for RNNs. Words in a sentence or timestamps in a time series are examples of sequences of items that make up the input data.
2. Recurrent connections in RNNs enable information to be transferred between steps in the sequence. The network can identify patterns and dependencies in sequential input because to this recurrent structure.
3. 4. The input gate selects the data to be stored in the memory cell from the current input. It calculates a potential value and establishes the appropriate addition quantity for the cell.
4. The forget gate determines which data from the prior state ought to be erased. It generates a forget factor that affects the memory cell by taking into account both the current input and the prior state.
5. The memory cell modifies its contents through the use of input and forget gates. It mixes the remembered data from the forget gate with the candidate value from the input gate.
6. Sequential input and matching target output are sent to the RNN-LSTM during training. A loss function is used to quantify the difference between the expected and actual output.
7. To reduce the loss,network's weights and biases are modified via Backpropagation. Since the gradients are calculated over the course of the entire sequence, this procedure is frequently referred to as Backpropagation Through Time (BPTT) in the context of RNNs.
8. Gradient clipping is sometimes used to address problems such as vanishing gradients in lengthy sequences. This entails restricting the gradients' magnitude during Backpropagation.
9. The model learns the sequential patterns and dependencies in the data through an iterative training phase, until the model reaches a performance level that is suitable, the network's parameters are modified.

DNN:

An artificial neural network is having several layers in the middle of both input and output layers called Deep Neural Network (DNN).[1][16] In order to introduce non-linearity, DNNs alter input data by applying activation functions to several layers of neurons. Through training, the network discovers ideal weights and biases to produce precise predictions for a given task. For the purpose of changing the model parameters in response to the calculated mistakes, the Backpropagation algorithm is essential. This model's operational procedures are as follows:

1. The DNN starts with an input layer that gets the input data's unprocessed features. In the input layer, every node denotes a feature.
2. There is a hidden layer or layers between input and output layers. Nodes (neurons) of each layer are linked to nodes in the layers above and below.

Weights connected with these linkages are acquired through training.

3. An activation function is applied to each node in a hidden layer to the weighted sum of its inputs. Rectified Linear Unit (ReLU) is a popular activation function for adding non-linearity to the model.
4. The input values are multiplied with the respective weights to determine the weighted sum. These products are then added to the total and a bias term is added.
5. The DNN discovers the ideal weights and biases during training in order to reduce the discrepancy between the goal values and its anticipated output. The error is propagated backward across the network via a technique known as Backpropagation, and the weights are changed appropriately.
6. A loss function is used to calculate the difference between actual target values and projected output. Reducing loss is main objective of training.
7. The weights& biases were adjusted based on the estimated gradients of the loss function with respect to these parameters using optimization algorithms like gradient descent, until the model performs at a level that is suitable, process is repeated.
8. An activation function which is suitable for the current task is applied by the output layer. A sigmoid function is typically used for binary classification, and a softmax function is typically used for multi-class classification.
9. DNN's prediction for the input is shown in its final output. For example, the output of a classification operation could be a probability distribution over many classes.
10. The DNN uses an iterative learning process, modifying its weights and biases in response to the relationships and patterns it discovers from the training set. Training doesn't stop until the model performs well enough on untested data.

RBM (CNN + BILGRU):

A particular kind of unsupervised generative artificial neural network is called a restricted Boltzmann machine (RBM). It acquires a probability distribution across its set of inputs through learning. It does this by minimizing the discrepancy between generated and observed data by modifying weights and biases. Deep learning can benefit from the use of the generated generative model in a number of applications. This model's operational procedures are as follows:

1. Visible units and hidden units are two levels of nodes that make up an RBM. There are no connections within a layer, but every node in one layer is connected toother layer's node.
2. The binary states of the nodes in both layers are typically 0 or 1. The input data is represented by the visible layer, where as the features or patterns are captured by the hidden layer.
3. RBMs function as models based on energy. Each visible and hidden unit configuration has a corresponding energy value. Readjusting the settings to make desirable configurations less energetic than undesired ones is the aim of the training process.
4. Weights are allocated to connections between visible and hidden units in RBMs. Furthermore, every visible and concealed unit has biases. The training procedure teaches these parameters.
5. To decrease the variationamong the data produced by the model and the observed data, weights and biases are adjusted throughout the training process of an RBM. Contrastive Divergence is a technique that is frequently used in the procedure.
6. Updating the hidden layers and adjusting states of the visible units to the input data constitute a positive phase of training. After updating the hidden units, the negative phase reconstructs the visible units using the updated hidden units.
7. The weights and biases are updated using the difference between the positive and negative phases.
8. An RBM can function as a generative model after it has been trained. The RBM creates new samples by updating the hidden layersand reconstructs the visible units giving a collection of visible units (input data).
9. Applications for RBMs include feature learning, dimensionality reduction, collaborative filtering, and more. They are frequently included into deeper neural network topologies as building components.

CNN + LSTM:

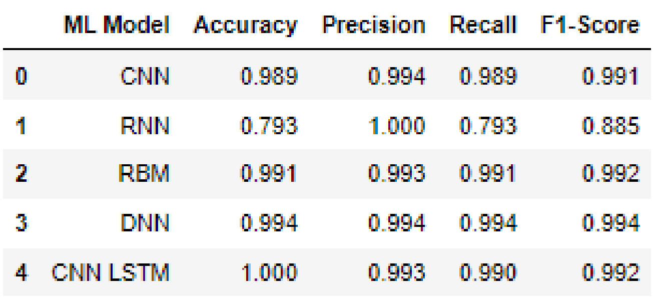
The CNN+LSTM design holds the advantages of both long short-term memory—which are useful for capturing temporal dependencies in sequential data—and convolutional neural networks (which are good at extracting spatial features). This combination is effective for jobs requiring both temporal and spatial comprehension, which makes it appropriate for a variety of sequence modeling and computer vision applications. This model's operational procedures are as follows: The input data for this architecture is typically sequential in nature, such as video frames, time series, or spatial-temporal data.

1. To extract spatial features from individual frames or spatial slices of the data, the network's first component usesCNN. Hierarchical representations of spatial patterns are captured by convolutional and pooling layers.
2. The LSTM network is then fed CNN's output. For capturing temporal dependencies in sequential data, the LSTM is a good choice. It analyses the CNN's sequential properties, taking into account their temporal interdependence and sequence..
3. The hidden states of an LSTM can be subjected to additional processing or applied directly to a range of tasks, including error detection, sequence prediction, and the generation of subsequent frames in a video sequence.
4. A suitable loss function is used throughout the CNN+LSTM architecture to train it from start to finish for the given objective. Based on the discrepancy between the expected and actual sequences, CNN and LSTM layers' parameters are modified in this way.
5. To compute gradients and update the weights of the LSTM layers, the network uses a Backpropagation variation known as Backpropagation Through Time (BPTT) during training. As a result, the model can simultaneously learn temporal and spatial representations.
6. When it comes to tasks like action recognition in films, frame prediction, or spatiotemporal pattern analysis in sequences, CNN+LSTM architecture is very helpful because it takes into account both spatial and temporal information.

   Step.4: Based on the best algorithm prediction is presented.

4. RESULTS

Table 1: 1st Dataset – KDD-CUP Comparison graphs



	ML Model	Accuracy	Precision	Recall	F1-Score
0	CNN	0.989	0.994	0.989	0.991
1	RNN	0.793	1.000	0.793	0.885
2	RBM	0.991	0.993	0.991	0.992
3	DNN	0.994	0.994	0.994	0.994
4	CNN LSTM	1.000	0.993	0.990	0.992

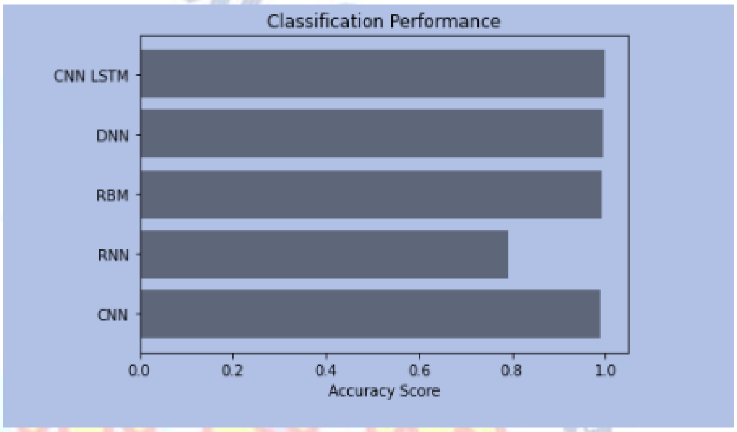


Fig 2.1: Accuracy comparative graph showing the differences between accuracy scores of deep learning algorithms using KDD-CUP dataset.

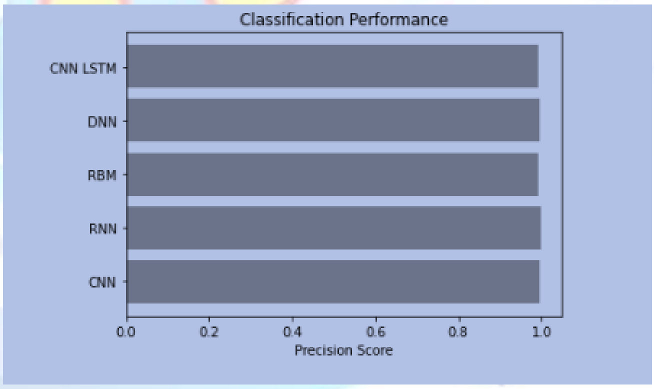


Fig 2.2: Precision comparativegraph showing the differences between the precision scores of deep learning algorithms using KDD-CUP dataset.

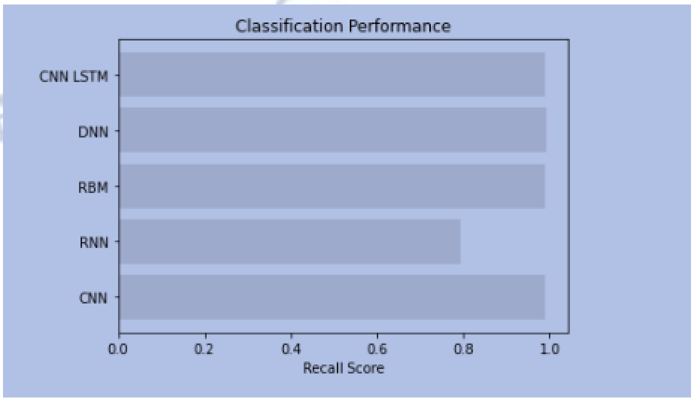


Fig 2.3: Recall comparative graph showing differences between recall scores of DL algorithms using KDD-CUP dataset.

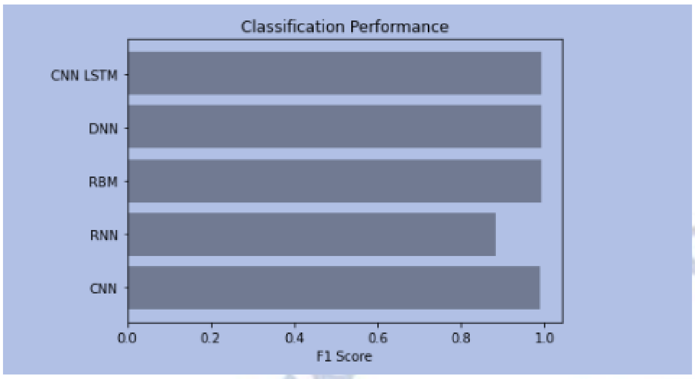


Fig 2.4: F1Score comparative graph showing the differences between F1 scores of DL algorithms using KDD-CUP dataset.

5. CONCLUSION

Deep learning was utilized in this framework to distinguish and classify malware that targets cyber-attacks. Seven strategies are employed by the system to improve efficiency: Generative models like RBN, DBN, DBM, and DAand Deep Learning models like RNN, CNN, and DNN. Our experiments demonstrates the possibilities identifying IDS and Online protection assaults using CNN-LSTM with more accuracy. Deep learning techniques advance the development of state-of-the-art guided frameworks and aid in categorization interruption requests. Future research in this area may focus on transfer learning and deep learning techniques. IDS preparation also verifies the integrity of the controlled framework.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] Mohammad Sazzadul Hoque, Md. Abdul Mukit and Md. Abu Naser Bikas,"AN IMPLEMENTATION OF INTRUSION DETECTION SYSTEM USING GENETIC ALGORITHM"

[2] A Chittur, "Model Generation for an Intrusion Detection System Using Genetic Algorithms". January 2005.

[3] Srinivas Mukkamala, Andrew H. Sung, Ajith Abraham, "Intrusion detection using an ensemble of intelligent paradigms", Journal of Network and Computer Applications, Volume 28, Issue 2, April 2005, Pages 167-182

[4] KDD-CUP-99 Task Description; http://kdd.ics.uci.edu/databases/kddcup99/task.html

[5] KDD Cup 1999: Tasks; http://www.kdd.org/kddcup/index.php?section=1999&method=task

[6] H. G. Kayacık, A. N. Zincir-Heywood, M. I. Heywood, "Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets", May 2005.

[7] X. Xu, and T. Xie, "A reinforcement learning approach for host-based intrusion detection using sequences of system calls," In Proc. of International Conference on Intelligent Computing, Lecture Notes in Computer Science, LNCS 3644, 2005, pp. 995-1003.

[8] Lee W., Stolfo S., and Mok K., "Adaptive Intrusion Detection: A Data Mining Approach," Artificial Intelligence Review, 14(6), December 2000, pp. 533-567.

[9] N. Toosi and M. Kahani, A New Approach to Intrusion detection Based on An Evolutionary Soft Computing Model Using Neuro-Fuzzy Classifiers.Computer Communications, vol. 30, Issue 10, pp. 2201-2212, 2007.

[10] Mahesh kumarsabhanani and gurselSerpen (2003), "Application of Machine learning algorithms to KDD intrusion detection dataset within misuse detection context" In Proceedings of the International Conference on Machine Learning, Models, Technologies and Applications (MLMTA), Vol. 1, pp. 209-215.

[11] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu and Ali A. Ghorbani (2009), "A detailed analysis of the KDD CUP 99 data set", in Proceedings of the Second IEEE international conference on Computational intelligence for security and defence applications, pp. 53-58, Ottawa, Ontario, Canada.

[12] KDDCUP,1999.Availableon:http://kdd.ics.uci.edu/databases/kddcup 99/kddcup99.html, Ocotber 2007.

[13] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani"A Detailed Analysis of the KDD CUP 99 Data Set"

[14] J. Azevedo and F. Portela, ''Convolutional neural network—A practical case study,'' in Proc. Int. Conf. Inf. Technol. Appl. Singapore: Springer, 2022, pp. 307–318.

[15] M. Uddin, R. Alsaqour, and M. Abdelhaq, ''Intrusion detection system to detect DDoS attack in Gnutella hybrid P2P network,'' Indian J. Sci. Technol., vol. 6, no. 2, pp. 71–83, 2013.

[16] J. Gu and S. Lu, ''An effective intrusion detection approach using SVM with Naïve Bayes feature embedding,'' Comput. Secur., vol. 103, Apr. 2021, Art. no. 102158.

[17] E. Gyamfi and A. Jurcut, ''Intrusion detection in Internet of Things systems: A review on design approaches leveraging multi-access edge computing, machine learning, and datasets,'' Sensors, vol. 22, no. 10, p. 3744, May 2022.

[18] Tara N. Sainath, Oriol Vinyals, Andrew Senior, Has¸imSak,"CONVOLUTIONAL, LONG SHORT-TERM MEMORY, FULLY CONNECTED DEEP NEURAL NETWORKS"

[19] Ayman Emam; M. Shalaby; Mohamed Atta Aboelazm; Hossam E. Abou Bakr,"A Comparative Study between CNN, LSTM, and CLDNN Models in The Context of Radio Modulation Classification"

[20] Hyun S. Kim ,Kyung M. Han,JinhyeokYuORCID,"Development of a CNN+LSTM Hybrid Neural Network"