ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 11 Issue 5 May 2024

Detecting Cyber Threats – A Deep Learning Based Framework for Network Attack Detection

[1] R. Syed Ali Fathima, [2] S. Karthik Reddy, [3] K. Nikhil Chowdary, [4] D. Jaya Sai Manjunath, [5] P. Partha Saradhi Reddy

[1] Assistant Professor, Department of Computer Science and engineering Kalasalingam academy of research and education Virudhunagar, Tamil Nadu, India
[2] Department of Computer Science and engineering, Kalasalingam academy of research and education Virudhunagar, Tamil Nadu, India
Email ID: [1] syedalifathima207@gmail.com, [2] sirupakarthikreddy2002@gmail.com, [3] nikhilkollipati@gmail.com, [4] jayasaimanjunath678@gmail.com, [5] palleparthasaradhi@gmail.com

Abstract*— Computer viruses, bad software, and other aggressive acts can damage a computer network. Intrusion monitoring, which is an active defence system, is a key part of network security. Problems with traditional intrusion detection systems include low accuracy, missed threats, a lot of false alarms, and not being able to handle new types of breaches. In order to address these issues, we propose a novel approach for identifying vulnerabilities in cyber-physical systems using deep learning. Our suggested framework highlights the differences between uncontrolled and DL -based methods. We demonstrate the effectiveness of a generative adversarial network in detecting cyber risks in IoT-powered IICs networks. The results show that this system is able to identify various types of threats with higher accuracy, reliability, and efficiency. State-of-the-art DL classifiers successfully detected the most common attacks on NSL-KDD, KDDCup99, and UNSW-NB15 datasets, while also protecting sensitive user and system data during training and testing.

Index Terms*— Intrusion Detection, Deep Learning, Cyber security Vulnerabilities, Generative Adversarial Network, Network Security.

I. INTRODUCTION

There is a very important part of defense called an Intrusion Detection System (IDS) that keeps networks and systems safe from bad things happening. This software carefully watches network data, trying to find and stop any strange or illegal activity as soon as possible. IDS is one of the most important defenses against cyber dangers; it is the first line of defense in the fight to protect valuable digital assets.

An important part of an IDS is its ability to tell the difference between harmless, normal network data and behavior that could be harmful. It does this by using both safe travel patterns and unique rules that are only used for attacks. By looking at the data that moves through a network, IDS tries to find things that don't seem right or don't follow the usual trends. This could mean that security has been broken or rules have been broken. In the past few years, data mining methods have been used to make IDS systems stronger and more accurate. These high-tech systems use machine learning algorithms and complex data to find and stop modern, complex cyber risks.

Protecting important assets, especially Internet Industrial Control Systems (IICS), is becoming a bigger issue in the field of hacking. As the number of devices in Industrial Internet of Things (IIoT) networks keeps growing, hacks on these linked systems become more likely. As a result, making IDS that work especially with IICS networks has become more important.

Even though IDS technology has come a long way, there are still problems to solve. Many IDSs on the market today have low detection rates and high false positive rates (FPR), which can cause a lot of useless alerts and security breaches that are missed. A potential way to solve these problems is to use Long Short-Term Memory (LSTM) models that are based on deep autoencoders. This new method uses the strengths of DL and recurrent neural networks to make an IDS for IIoT-powered IICS that works better.

In conclusion, intrusion detection systems are important parts of modern cybersecurity because they keep an eye out for online dangers. This is important because technology is always changing and hacks are getting smarter. To keep vital infrastructure safe and secure in the digital age, we need to create IDS solutions like deep autoencoder-based LSTM models for IIoT-powered IICS. These more advanced intrusion detection systems might be able to improve the accuracy of detection, cut down on false positives, and offer a strong defense against online threats that target industrial systems.

II. LITERATURE REVIEW

Image segmentation worked well with convolutional neural networks. "AlexNet," "VGG," "Inception," and "ResNet" are all examples of networks. This will show which networks did better on the job with the "ImageNet" dataset. Then, to see how well it worked, they put movies into groups using the "Kinetics400" and "UCF101" datasets. Lastly, it was checked to see if their success with pictures would carry

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 11 Issue 5 May 2024

over to films. One way to do this is to compare the error margins of the networks above. To group movies, the two networks with the smallest error range are put to the test. If they are successful, these networks will be able to find people in videos recorded by devices. The fact that both "ResNet" and "Inception" networks had 70% success rates showed that the method worked.

The quick spread of Internet of Things (IoT) apps has made networks busier and created a lot of gadgets that need to do complicated calculations. IoT devices collect information that helps people and businesses make decisions that can change their lives. Most of the time, these IoT devices have slow CPUs, little memory, and storage that uses little power. This means hackers can get into these machines since they can't run general protection software. There is a new threat to IoT networks now. This problem can be fixed by multi-access edge computing (MEC), which takes complicated computer chores from IoT devices to the edge. The linked work has mostly been about finding the best ways to keep IoT gadgets safe. We think that distributed systems based on MEC should be given more attention. This piece goes into great depth about new network intrusion detection systems (NIDS) and the safety of the Internet of Things (IoT). Machine learning (ML) methods based on MEC were looked at. The study also looks at how to apply NIDS and compares information that are available to the public. Finally, we suggest IoT networks that are based on MEC and have NIDS.

As IT gets better, digital data is growing very quickly. This has led to new security issues that need to be fixed right away. The best way to stop harmful attacks and find out what's going on with a network is to use intrusion monitoring tools. A lot of IDS methods use machine learning (ML). There are more false alarms and data bias in ML-based IDSs because their training data sets are shorter. This study creates a hybrid network-based intrusion detection system (HNIDS) for data that isn't all the same. It makes use of better random forest (IRF) and enhanced genetic algorithm and particle swarm optimization (EGA-PSO) techniques. First, the suggested HNIDS makes small data sets better by using EGA and PSO. To learn about small sample traits, you can use a fair set of data. The vector in the suggested HNIDS is better with PSO. A multi-objective function makes GA better by picking the best features and making fitness results better so that the attention is on the most important features. It also shrinks the size, raises the TPR, and drops the FPR. Next, an IRF gets rid of traits that aren't important, uses a list of decision trees in each process that is repeated, keeps an eye on the classifier, and stops it from overfitting. Using NSL-KDD test datasets, the suggested method is compared to a number of machine learning algorithms. The HNIDS method has a success rate of 98.979% on BCC and 88.149% on MCC for the NSL-KDD dataset, which is significantly higher than the success rates of SVM, RF, LR, NB, LDA, and CART.

Health care is a great place for IoT to grow. New developments in the Internet of Medical Things (IoMT) will make medical care better. Even though it has benefits, cyberattacks on healthcare technology that is connected to the internet could put patients' health and privacy at risk. We need IoMT systems that work well and provide smooth medical services for a large group of people. To keep patients' privacy and safety in this network safe, we need a strong protected model. It is hard to come up with security methods for IoMT networks. This paper tries to come up with a tree-based IoMT network intruder detection model. The suggested method cuts down on the size of the raw data to speed up finding anomalies while keeping the accuracy at 94.23%.

Fog computing, an extension of cloud technology, has shifted the responsibility of managing UAV data from the internet to a more distributed approach. Previously, the internet served as the central hub for this task. The main goals are to make it easier for drones to figure out their resource limits and send that information to a computer node outside of the drone for scheduling, processing, management, optimization, and safety. The nodes are close to each other and are joined by a wireless sensor network, which makes this possible. Most drone systems are planned and made with standard technologies to make them reliable. This lowers the amount of power, resources, and delay needed for fast-response apps. But new study on fog-enabled drone-based data management and speed risks privacy, data safety, and even life. But Bitcoin and other cryptocurrencies use Blockchain Hyperledger Technology a lot. It is used in a lot of distributed systems because it is clear, reliable, available, safe, trustworthy, and has origin. Distributed drone control is becoming more popular because fog nodes and blockchain Hyperledger technology can collect, organize, analyze, improve, handle, and store data from drones. We suggested working together with blockchain Hyperledger fabric and B-Drone, a genetic program with metaheuristics for managing fog nodes. This is where drone data is planned, improved, handled, controlled, and safely kept in the fog node. Before transactions are sent, hash encryption (SHA-256) keeps them safe. In the private permissioned network, the blockchain smart contracts handle the communication and link protocols between drone-fog nodes automatically. The joint way cuts down on computing costs by 12.03% and raises performance by 73.11%, as shown in simulations. This makes the network more reliable by 54.29% and lowers the cost of preserving drone ledgers by 30.13% compared to previous cutting-edge methods.

III. EXISTING SYSTEM

The work that has already been done is mostly about finding and labeling hacks that come from places that aren't expected or planned.

While the current work looks at how to use both traditional machine learning algorithms and a deep neural network (DNN). Which could mean that cyberattack discovery works less well.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 11 Issue 5 May 2024

The work that has already been done, on the other hand, is mostly based on publicly available standard malware datasets, which might not fully show how complicated current cyber-physical systems are.

IV. METHODOLOGY

In the papers, a deep neural network (DNN), a type of DL model, is looked into to make an IDS that can adapt to different situations and find and label hacks that no one saw coming. Because threats and network activity are always changing, it's important to look at the different datasets that have been created over the years using both basic and dynamic methods. This kind of research helps to find the best program that can find future cyberattacks before they happen. A full analysis of tests using DNNs and other traditional machine learning models is shown on a number of freely available standard malware datasets.

We suggest a new way to find safety holes and breaches in cyber-physical systems that is based on DL. The suggested structure compares the uncontrolled and DL -based (RNN, CNN, and DNN) ways of telling the difference. We present a generative adversarial network (RBN, DBN, DBM, and DA) that can find cyber risks in IICs networks that are powered by IoT. This test looks at how well the suggested efficient IDS scheme works on IIoT IICs and outside networks using the NSLKDD, KDDCup99, and UNSW-NB15 datasets.

Benefits:

We target cyber-physical system cybersecurity breaches. This may help us pinpoint risks.

We demonstrate further DL techniques and GAN designs, including RBN, DBN, DBM, and DA. With multiple ways, recognition may be more effective and versatile.

We use datasets like NSL-KDD, KDDCup99, and UNSW-NB15 to test our suggested IDS system. In the area of breach detection study, these files are generally seen as standards.

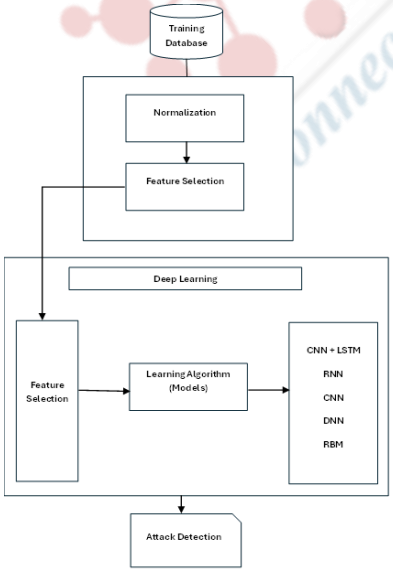

Fig. 1. Proposed Architecture

We talk about generative adversarial networks (GANs), which are a way to find computer threats. GANs have shown promise in many areas because they can create and sort data, which could make it easier for cyber-physical systems to find things.

V. IMPLEMENTATION

Procedures:

Step 1: Data Loading Module:

Purpose: Import the dataset from the provided links.

Dataset Links:

- NSL - KDD: NSL-KDD Dataset
- KDD-CUP: KDD-CUP Dataset
- UNSW-15-NB: UNSW-NB15 Dataset

Description: Provide information about the datasets, including their sources and the types of attacks and normal network traffic they contain.

Step 2: Data Preprocessing Module:

Purpose: Get the information ready and look through it.

Some tasks that might be needed are cleaning up the data, dealing with missing values, storing category features, and doing exploratory data analysis (EDA).

Step 3: Splitting Data into Train & Test Module:

Purpose: Make two sets of data: one for training and one for testing.

To make sure the split is fair, use methods like stratified selection.

Step 4: Model Generation Module:

Purpose: Build various machine learning models to detect network intrusions.

For each dataset (NSL-KDD, KDD-CUP, UNSW-NB15), you mentioned using different models:

KDD CUP: CNN, RNN (LSTM), DNN, RBM (CNN + BiLSTM), CNN + LSTM

NSL KDD: CNN, RNN (LSTM), DNN, RBM (CNN + BiLSTM), CNN + LSTM

UNSW-NB15: CNN, RNN (LSTM), DNN, RBM (CNN + BiLSTM), CNN + LSTM

Calculate the accuracy of each model using evaluation metrics.

Step 5: User Signup & Login Module:

Purpose: Users should be able to sign up and log in to the system.

Set up ways for users to prove who they are and what they can do.

Step 6: User Input Module:

Purpose: Get feedback from people to help you make predictions.

Get the network flow info or features that you need to find intrusions.

Step 7: Prediction Module:

Purpose: Based on the chosen machine learning models, make predictions.

Show the end expected result, which will tell you if a network attack was found or not.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 11 Issue 5 May 2024

Step 8: Ensemble Method Module:

Purpose: Combine the findings of more than one model to get more accurate results.

Look into group methods that have been shown to be very accurate, like CNN + LSTM. Use group methods to improve the system's performance. You can use computer languages and libraries that are right for your project to build each of these modules. For example, you could use Python with libraries like scikit-learn, TensorFlow, and Keras, and for the user registration and input parts, you could use web development tools. To do the things that are listed in each section, you would have to write code, process data, train and test models, and make a way for people to connect with the system.

Algorithms

CNN

CNNs, or Convolutional Neural Networks, are DL models designed to interpret visual data like images and videos. The network's convolutional layers allow it to learn from the input and detect patterns, shapes, and features. These layers use filters to find and pull out important data while keeping the connections between things in space. CNNs are very useful for tasks like image classification, object detection, and face recognition because they can learn and describe complex visual traits in a hierarchical way. This makes them important in computer vision applications and many other areas that involve understanding and analyzing images.

RNN (LSTM)

LSTM artificial neural networks handle sequential input using memory and feedback loops. Recurrent neural networks. They excel in time-sensitive tasks like time series analysis, natural language processing, voice recognition, and more.

Standard RNNs have the vanishing gradient issue, but LSTMs overcome it. They employ custom memory cells, switches, and cell states, making their design more sophisticated. Because they can store data throughout time, LSTMs can grasp and predict sequential data context.

In an LSTM network, information passes between input, forget, and output gates. These gates control how data enters and leaves the memory cell. They do this so that important data stays in the memory cell and useless data is thrown away. This lets LSTMs understand long-distance connections and correctly guess or group things based on sequential trends.

To sum up, LSTMs are a type of RNN that are great at dealing with sequential data because they can keep and change background information over time. They are used for many things, like analyzing text and speech to making predictions based on time series and more, where it's important to understand and use how time works.

DNN

Artificial neural networks with numerous layers of linked nodes are called deep neural networks (DNNs). DNNs solve complicated issues in computer vision, voice recognition, and natural language processing, making them vital to machine learning and DL. A typical DNN has an input layer, numerous hidden layers, and an output layer. The network's nodes alter and transform data at each layer. DL methods, which are often used to train DNNs, change the weights and biases of these neurons over and over again while they are being trained. This lets them learn complex patterns and representations in the data. Deep neural networks can easily pull-out hierarchical features because of their depth. This makes them good at learning features and abstracting them. Many jobs, such as picture classification, object recognition, machine translation, and voice synthesis, have been done very well by DNNs. However, training deep networks can be hard on computers and may need a lot of tagged data. Hardware improvements and better training methods have made DNNs much more useful and effective, strengthening their place as a mainstay of modern machine learning and artificial intelligence.

RBM (CNN + BiLSTM)

A DL design called a Recurrent Boltzmann Machine (RBM) combines a Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM). It is used for many things, such as handling images and sequences. The CNN takes in data like images and pulls out spatial features, while the BiLSTM finds trends that happen in a certain order. RBM adds the ability to generate and distinguish. These parts work together to make a strong model that can learn hierarchical structures from complex data. This model can handle both spatial and time information, which makes it useful for tasks like picture recognition, natural language processing, and more.

CNN + LSTM

A mixed DL design is made up of a Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM). CNN is great at taking out spatial traits from data, which makes it perfect for picture analysis. LSTM, on the other hand, is great at dealing with sequential data, like natural language. When CNN and LSTM work together, CNN can pull out useful features from the input, and LSTM handles these features in order, recording how they change over time. This combination is very helpful for tasks like video analysis, which needs both geographical and time information, and for natural language tasks, which need to understand the context and links between words in a string of text.

VI. EXPERIMENTAL RESULTS

Accuracy: How well a test can tell the difference between sick and healthy people is called its accuracy. To get an idea of how accurate a test is, we should figure out what percentage of cases are true positives and true negatives. In terms of math, this can be written as

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 11 Issue 5 May 2024

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Precision: Precision is the percentage of correctly classified cases or samples compared to those that were correctly classified as hits. So, here is the method to figure out the precision:

Precision = True positives/ (True positives + False positives) = TP/ (TP + FP)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive+False\ Positive}$$

Recall: In machine learning, recall measures how successfully a model finds all key examples of a class. It indicates how effectively a model captures class cases. Divide the number of accurately anticipated positive observations by the total genuine positives.

$$Recall = \frac{TP}{TP+FN}$$

F1-Score: The F1 score is a way to rate the correctness of a machine learning model. It takes a model's accuracy and memory scores and adds them together. The accuracy measure counts how many times, across the whole collection, a model made a correct guess.

$$\textbf{F1 Score} = \frac{2}{\left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}\right)}$$

$$\textbf{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

KDD-CUP DATASET GRAPHS

KDD-CUP

Accuracy Curve of CNN&LSTM:

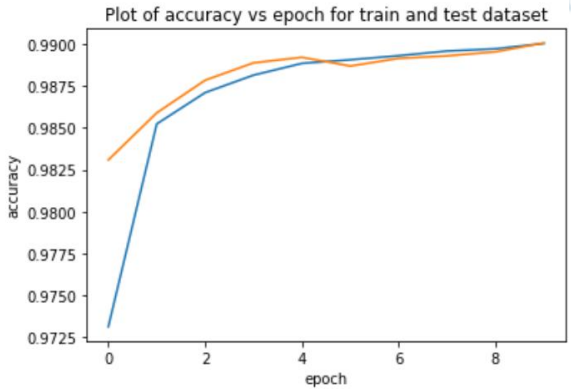

Fig. 2. Accuracy curve of CNN+LSTM

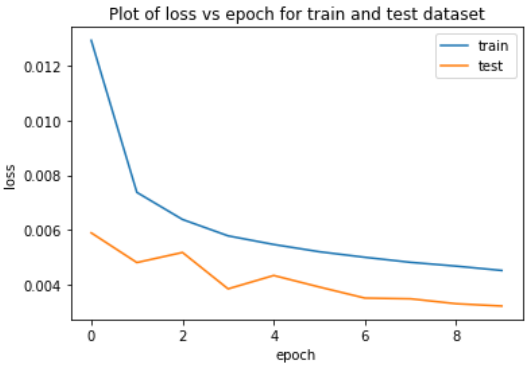

Fig. 3. Loss curve of CNN+LSTM

Accuracy:

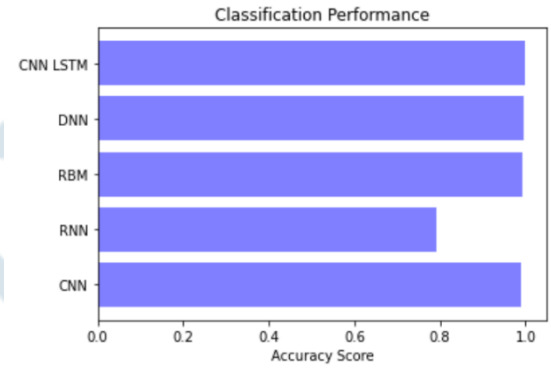

Fig. 4. Accuracy comparison graph

Precision:

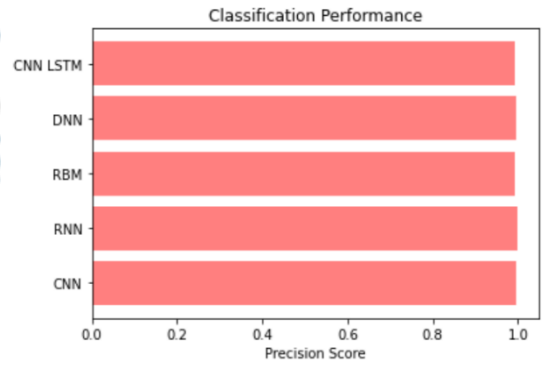

Fig. 5. Precision comparison graph

Recall:

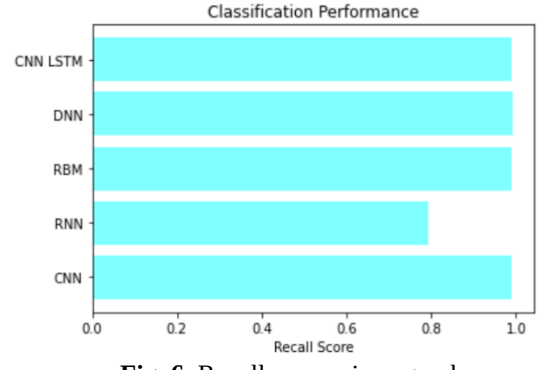

Fig .6. Recall comparison graph

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 11 Issue 5 May 2024

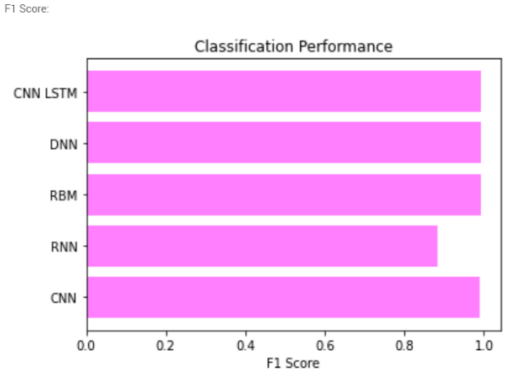

Fig. 7. F1-Score comparison graph

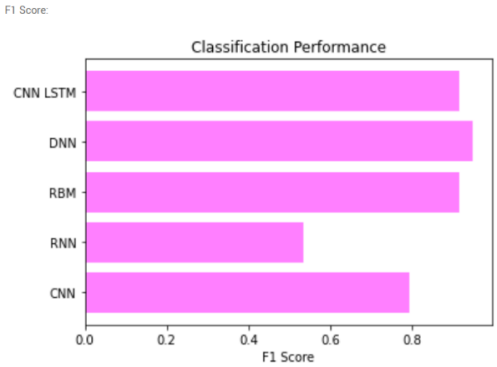

Fig. 11. F1-Score comparison graph

NSL-KDD DATASET GRAPHS

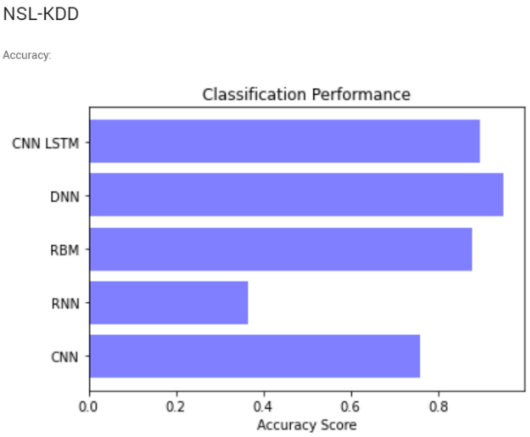

Fig. 8. Accuracy comparison graph

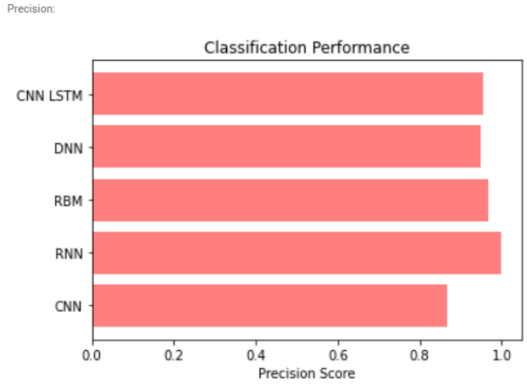

Fig. 9. Precision comparison graph

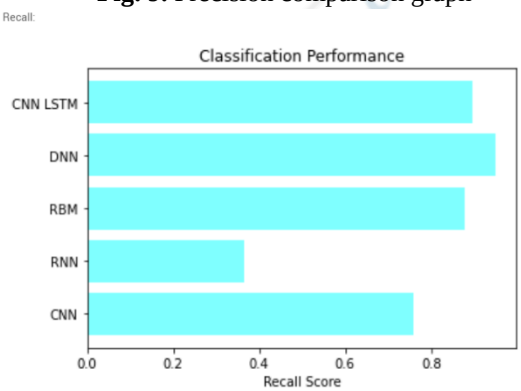

Fig. 10. Recall comparison graph

UNSW-NB15 DATASET GRAPHS

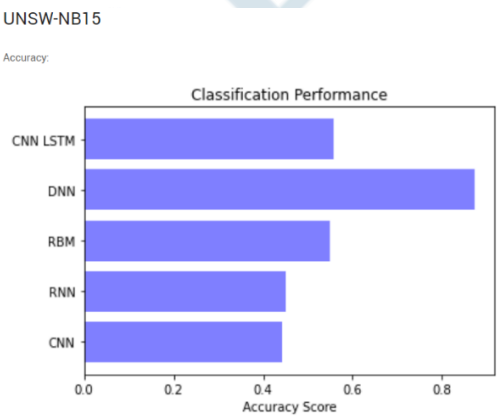

Fig. 12. Accuracy comparison graph

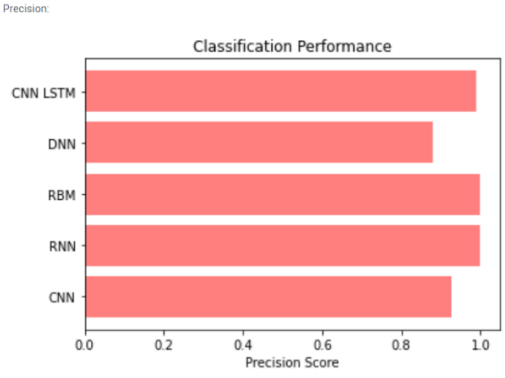

Fig. 13. Precision comparison graph

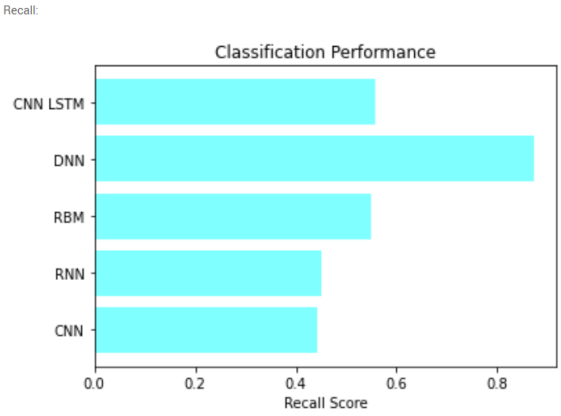

Fig. 14. Recall comparison graph

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 11 Issue 5 May 2024

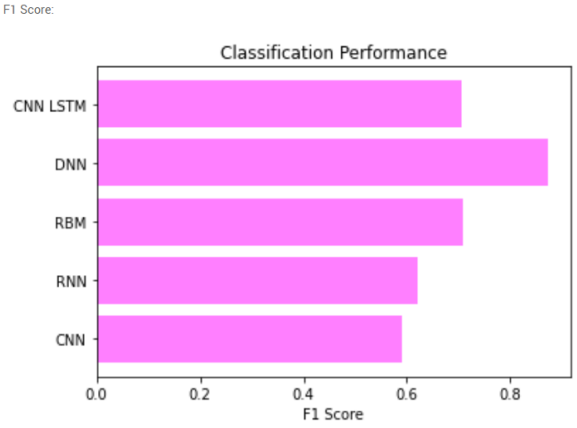

Fig. 15. F1-Score comparison graph

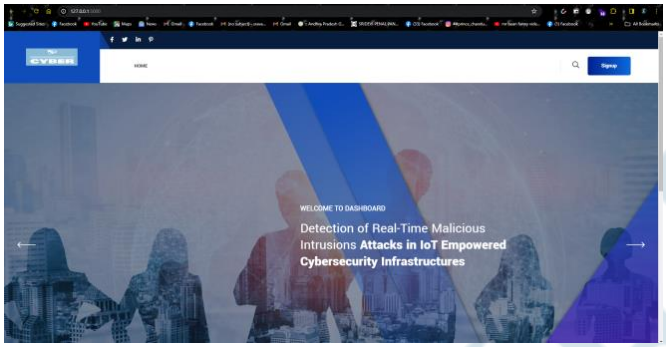

Fig. 16. Flask home page

VII. CONCLUSION

This essay talks about the problems and restrictions that came up in earlier research that tried to figure out how to use DL to find and get rid of online dangers quickly. DL methods, such as recognition and discrimination, are used to find cyberattack software. We did, however, make a list of the seven ways, which are DL (RNN, CNN, and DNN) and generative models/methods (RBN, DBN, DBM, and DA). Our study also looks at how accurate the texts are in the research area. The tests we did for our work show that IDS and cybersecurity threats can be found by working together in a technological setting. Besides that, we looked into which DL methods worked better than the others. Based on this study, using DL techniques raises the rate of classification attack investigations while maintaining the high performance of cutting-edge guided systems. To make this study more useful for future work, it was expanded to include more advanced deep learning methods and transfer learning techniques. IDS training is also used to confirm that the guided system is stable. When creating a creative new Intrusion Detection System (IDS), the features may be employed in real time to discover inside and outside hackers and their evil deeds.

In the future, researchers can add more advanced DL techniques, look into transfer learning, make IDS more reliable, and help find threats in real time.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, ‘‘Deep learning,’’ Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘‘ImageNet classification with deep convolutional neural networks,’’ Commun. ACM, vol. 60, no. 2, pp. 84–90, Jun. 2017.

[3] M. K. Islam, M. S. Ali, M. M. Ali, M. F. Haque, A. A. Das, M. M. Hossain, D. S. Duranta, and M. A. Rahman, ‘‘Melanoma skin lesions classification using deep convolutional neural network with transfer learning,’’ in Proc. 1st Int. Conf. Artif. Intell. Data Analytics (CAIDA), Apr. 2021.

[4] A. Ahmim, M. Derdour, and M. A. Ferrag, ‘‘An intrusion detection system based on combining probability predictions of a tree of classifiers,’’ Int. J. Commun. Syst., vol. 31, no. 9, p. e3547, Jun. 2018.

[5] A. Ahmim, L. Maglaras, M. A. Ferrag, M. Derdour, and H. Janicke, ‘‘A novel hierarchical intrusion detection system based on decision tree and rules-based models,’’ in Proc. 15th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS), May 2019, pp. 228–233.

[6] Z. Dewa and L. A. Maglaras, ‘‘Data mining and intrusion detection systems,’’ Int. J. Adv. Comput. Sci. Appl., vol. 7, no. 1, pp. 1–10, 2016.

[7] B. Stewart, L. Rosa, L. A. Maglaras, T. J. Cruz, M. A. Ferrag, P. Simoes, and H. Janicke, ‘‘A novel intrusion detection mechanism for SCADA systems which automatically adapts to network topology changes,’’ EAI Endorsed Trans. Ind. Netw. Intell. Syst., vol. 4, no. 10, p. e4, 2017.

[8] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, ‘‘Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study,’’ J. Inf. Secur. Appl., vol. 50, Feb. 2020, Art. no. 102419.

[9] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, ‘‘A bidirectional LSTM deep learning approach for intrusion detection,’’ Expert Syst. Appl., vol. 185, Dec. 2021, Art. no. 115524.

[10] A. A. Salih, S. Y. Ameen, S. R. Zeebaree, M. A. Sadeeq, S. F. Kak, N. Omar, I. M. Ibrahim, H. M. Yasin, Z. N. Rashid, and Z. S. Ageed, ‘‘Deep learning approaches for intrusion detection,’’ Asian J. Res. Comput. Sci., vol. 9, no. 4, pp. 50–64, 2021.

[11] 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.

[12] K. He, X. Zhang, S. Ren, and J. Sun, ‘‘Deep residual learning for image recognition,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[13] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, ‘‘How transferable are features in deep neural networks?’’ in Proc. Adv. Neural Inf. Process. Syst., vol. 27, 2014, pp. 1–9.

[14] G. Awad, C. G. Snoek, A. F. Smeaton, and G. Quénot, ‘‘Trecvid semantic indexing of video: A 6-year retrospective,’’ ITE Trans. Media Technol. Appl., vol. 4, no. 3, pp. 187–208, 2016.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, ‘‘Rethinking the inception architecture for computer vision,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2818–2826.

[16] 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.

[17] R. L. Haupt and S. E. Haupt, Practical Genetic Algorithms. Wiley, 2004, doi: 10.1002/0471671746.

[18] D. Hossain, G. Capi, and J. M., ''Optimizing deep learning parameters using genetic algorithm for object recognition and robot grasping,'' J. Electron. Sci. Technol., vol. 16, no. 1, pp. 11–15, 2018.

[19] O. E. David and I. Greental, ''Genetic algorithms for evolving deep neural networks,'' in Proc. Companion Publication Annu. Conf. Genetic Evol. Comput., Jul. 2014, pp. 1451–1452.

[20] 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.

[21] 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.

[22] A. K. Balyan, S. Ahuja, U. K. Lilhore, S. K. Sharma, P. Manoharan, A. D. Algarni, H. Elmannai, and K. Raahemifar, ''A hybrid intrusion detection model using EGA-PSO and improved random forest method,'' Sensors, vol. 22, no. 16, p. 5986, Aug. 2022.

[23] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, and K. I.-K. Wang, ''Hierarchical adversarial attacks against graph-neural-network-based IoT network intrusion detection system,'' IEEE Internet Things J., vol. 9, no. 12, pp. 9310–9319, Jun. 2021.

[24] A. A. Khan, A. A. Laghari, T. R. Gadekallu, Z. A. Shaikh, A. R. Javed, M. Rashid, V. V. Estrela, and A. Mikhaylov, ''A drone-based data management and optimization using Metaheuristic algorithms and blockchain smart contracts in a secure fog environment,'' Comput. Electr. Eng., vol. 102, Sep. 2022, Art. no. 108234.

[25] K. Gupta, D. K. Sharma, K. Datta Gupta, and A. Kumar, ''A tree classifier-based network intrusion detection model for Internet of Medical Things,'' Comput. Electr. Eng., vol. 102, Sep. 2022, Art. no. 108158.

[26] Sudar, K. M., Beulah, M., Deepalakshmi, P., Nagaraj, P., Chinnasamy, P. (2021). Detection of Distributed Denial of Service Attacks in SDN using Machine Learning Techniques. In Proceedings of the 2021 International Conference on Computer Communication and Informatics (ICCCI 2021), January 27-29, 2021, Coimbatore, India (Article number: 9402517, Category number: CFP2108R-ART, Code: 168476).