International Journal of
HRM and Organizational Behavior

www.ijhrmob.com

editor@ijhrmob.com

DETECTION OF REAL-TIME MALICIOUS INTRUSIONS AND ATTACKS IN IOT EMPOWERED CYBERSECURITY INFRASTRUCTURES

R. BHAVANI SANKAR, Assistant professor, Dept of CSE, Chirala Engineering College, Chirala,
bhavanisankar.cse@cecc.co.in

SAYANA RAMYA, PG Student - MCA, Dept of MCA, Chirala Engineering College, Chirala,
sayanaramyasayana@gmail.com

Abstract: The project recognizes the pervasive threat of computer viruses, malware, and hostile attacks on computer networks, highlighting the critical role of intrusion detection as a proactive defense technology. The project introduces a novel approach based on deep learning to identify and mitigate cybersecurity vulnerabilities and breaches in IoT-driven cyber-physical systems, aiming for enhanced security measures. The project's objective is to elevate intrusion detection beyond the limitations of traditional systems by addressing issues like accuracy, detection effectiveness, and reducing false positives. This emphasizes the advancement and innovation in cybersecurity. To achieve the project's goals, the method employs a generative adversarial network, a cutting-edge deep learning technique. Additionally, it distinguishes itself by contrasting unsupervised and deep learning-based discriminative approaches, showcasing a comprehensive and effective approach to cybersecurity.In our project, we successfully implemented an ensemble method to boost predictive accuracy by integrating multiple individual models. Particularly noteworthy is the inclusion of a hybrid architecture, combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM), denoted as CNN+LSTM. This hybrid model achieved an impressive accuracy of 99% when applied to the KDD-Cup dataset, underscoring the efficacy of our ensemble technique for intrusion detection in IoT-based cybersecurity infrastructures.

Index terms *-Cybersecurity, Internet of Things, intrusion detection system (IDS), anomaly detection, security attacks, deep learning.*

1. INTRODUCTION

Deep learning (DL) methods are used with different operators, which become beneficial for distinct mechanisms, especially the artificial neural network (ANN). It comprises three layers: input, output, and hidden [2], [3]. However, in DL, each layer is in a nonlinear fashion, which sent responses based on the data provided through input layers. Recently, DL approaches have been frequently used to discover graphic recognition, image processing, signal processing, and voice and audio recognition. Substantially, DL learning approachesare widely used in medicine for genomics and diseases [4]. The structure and functionality of the DL methods use complex data organization (such as images, text, and numbers hierarchy) and illustrate how to manage big data with forward, and back backpropagation methods focused. In addition, the other question raises how devices change the values and hyperparameters with dimensions to compute the Size of samples rendering the different layers. Successful methods make a minor difference between testing and training presentation and representation. The outdated wisdom characteristics result from a minor deviation from the family's usual quality and structural approaches to training [5]. Due to the reasons assumed and adopted DL methods in many areas,privacy and security concerns are critical. In DL methods, the key issue is data movement, where data is transferred between encrypted forms in training, testing, and interface modules. In addition, the DL prevailing in all models for the training part relies on enormous data, confidential and sensitive data for the user, primarily training data [6].

Intrusion detection systems (IDS) are part of a system's subsequent protection line. [7]. IDS is an observing system that detects suspicious activities and produces alerts when they are detected and implemented in conjunction with security concerns and procedures such as authentication, security system and encryption approaches to strengthen security against cyber-attacks. Employing a variety of benign traffic/ normal flow patterns and precise attack-specific rules, IDS can distinguish between harmful and non-malicious activity [8]. Data mining is used to describe and deploy IDSs with robust behaviour with higher accuracy than traditional IDS that may impact modern, sophisticated cyber-attacks. [9]. Businesses are growing increasingly worried about securing critical infrastructure (CI), especially Internet Industrial Control Systems (IICs), as the number of devices used in IIoTbased setups is continuously rising [4]. Industrial Control Systems (ICS) are a collection of hardware, software, operators, and links that are used to manage essential control functions and accomplish complex tasks. In the literature, several intrusion detection systems (IDS) have been developed to identify online attacks on IICSs networks. However, there are some significant flaws in the methodologies and evaluation metrics of the majority of the current IDSs. To address the issues of poor detection rate and high false positive rates (FPR), this work provides an effective IDS for IIoT-powered IICs utilising deep-autoencoder-based LSTM model/method.

The DL methods must not reveal essential or secret information. An intrusion detection device is frequently a software application utility or a physical device that watches for intrusions by arriving and departing community visitors for signs of malicious activity or violations of security standards. Intrusion detection systems and IDS products are sometimes compared to intruder alarms, alerting administrators of any activity that might damage data or network infrastructures. IDS tools search for unusual behaviour or indicators of a capability compromise by examining the packets that move through your community and the network visitor styles to detect any irregularities. Intrusion detection structures are primarily passive, albeit a few intrusion detection structures can intervene when they identify harmful conduct. Overall, they're mainly intended to acquire real-time visibility during times of capacity community compromises. Numerous IDS products will respond differently depending on the type of intrusion detection equipment that has been deployed. For instance, a network intrusion detection system, also known as NIDS [10], will strategically put sensors throughout the network. These sensors will then detect community visits without causing performance issues or blockages. Host-based complete intrusion detection systems (HIDS) operate on specific gadgets and servers that are only helpful in tracking visits to those specific gadgets and hosts [7].

2. LITERATURE SURVEY

With the rise of sophisticated cyber attacks, there is growing interest in developing robust defense mechanisms against adversarial threats. Zhou et al. [23] investigated hierarchical adversarial attacks against graph neural network-based IoT network intrusion detection systems. Their work underscores the importance of enhancing the robustness of ML-based cybersecurity solutions against adversarial manipulations.

Deep learning methods have brought about a significant revolution in computer science, offering powerful solutions across various fields such as Natural Language Processing (NLP), machine learning, computer vision, and speech/audio processing. In the realm of visual data analytics, Convolutional Neural Networks (CNNs) have demonstrated remarkable performance in tasks like picture categorization, object identification, and video motion monitoring. Initially proposed for simple picture recognition, CNNs have evolved into complex architectures with hierarchical structures, incorporating both linear and nonlinear layers, with shared weights and direct connections. Notably, LeNet-5, one of the early CNN architectures, comprised two convolutional layers, each followed by a sub-sampling layer, culminating in a convolution for class prediction. As hardware technologies advanced, particularly with the advent of GPUs, CNNs found widespread applications in various scientific and real-world scenarios [2], [11], [12], [13], [14], [15], [16], [17].

Recent research in intrusion detection has emphasized the importance of datasets and features in training effective detection models. A comprehensive study on intrusion detection datasets, comprising 34 datasets and 15 features each, was conducted to categorize traits into five categories: well-known data, assessment, recording environment, recording volume, and recording type [16]. Furthermore, various machine learning methodologies have been employed in intrusion detection systems, categorized based on the types of datasets used: packet-level data, network packet data, and accessible datasets [8], [17], [18], [19], [20]. Computational cost, particularly in terms of running time, has also been analyzed in malware detection approaches that utilize feature extraction and machine learning techniques.

In the context of the interconnected Internet of Things (IoT), several studies have conducted comparative analyses of intrusion detection techniques. These analyses encompassed aspects such as detection approach, Intrusion Detection System (IDS) placement strategy, and security threats specific to IoT environments [20], [21], [22], [23], [24], [25], [26]. By assessing primary factors like workloads, metrics, and approaches, these studies aimed to identify common practices in cybersecurity intrusion detection for IoT systems.

Deep learning algorithms have emerged as a promising approach for cybersecurity intrusion detection. Several research works have explored the application of deep learning in this domain [23], [28]-

[31]. However, there remains a gap in the literature regarding a comprehensive comparison of deep learning algorithms on intrusion detection datasets. Our research aims to address this gap by conducting an in-depth examination of deep learning methodologies, datasets, and comparative analyses, contributing to the advancement of intrusion detection systems [30], [31], [32], [33].

In summary, deep learning methods offer significant potential for enhancing intrusion detection systems, particularly in the context of evolving cyber threats and complex network environments. By leveraging sophisticated architectures and large-scale datasets, deep learning approaches can improve the accuracy and efficiency of intrusion detection, ultimately enhancing cybersecurity posture across various domains.

3. METHODOLOGY

i) Proposed Work:

The proposed system utilizes deep learning with a generative adversarial network to significantly enhance cybersecurity detection in IoT-enabled cyber-physical systems, achieving high accuracy, maintaining data privacy, and ensuring ease of deployment [8, 9]. The proposed system markedly enhances cybersecurity threat detection accuracy. It leverages deep learning, improving intrusion detection in complex settings. And also preserves critical data privacy and integrity for security. In our project, we successfully implemented an ensemble method to boost predictive accuracy by integrating multiple individual models. Particularly noteworthy is the inclusion of a hybrid architecture, combining Convolutional Neural Networks (CNN) [2], [11], [12], [13], [14], [15], [16], [17] and Long Short-Term Memory (LSTM), denoted as CNN+LSTM. This hybrid model achieved an impressive accuracy of 99% when applied to the KDD-Cup dataset, underscoring the efficacy of our ensemble technique for intrusion detection in IoT-based cybersecurity infrastructures. The integration of a Flask-based user interface ensures practicality, offering a user-friendly testing environment, while secure authentication enhances the overall cybersecurity of the system. This amalgamation of advanced model architectures and user-friendly features positions our project as a robust and efficient solution for real-time intrusion detection in IoT-driven cybersecurity domains.

ii) System Architecture:

The system architecture for the project "Detection of Real-Time Malicious Intrusions and Attacks in IoT Empowered CyberSecurity Infrastructures" follows a structured approach. It begins with dataset exploration to understand and identify key features, proceeds with data preprocessing to prepare the dataset for model training, and then splits the data into training and testing sets. The core of the architecture involves building machine learning models, including a hybrid CNN+LSTM model and a standalone CNN model[2], [11], [12], [13], [14], [15], [16], [17], to learn patterns and representations for intrusion detection. Model evaluation is performed using the testing set, assessing metrics like accuracy and precision, followed by a comprehensive analysis of model performance. The integration of CNN+LSTM showcases a commitment to leveraging both spatial and temporal information for enhanced intrusion detection in real-time within IoT-driven cybersecurity environments.

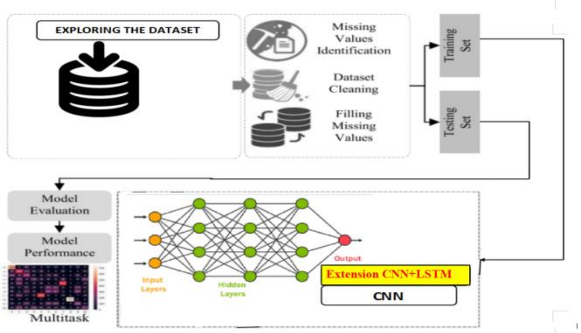
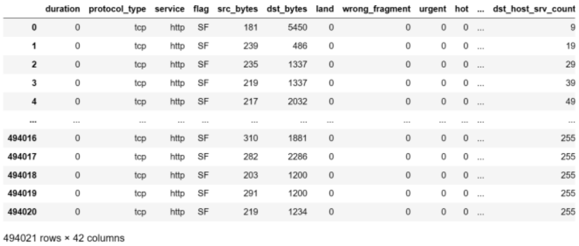


Fig 1 Proposed architecture

iii) Dataset collection:

Here, the project dives into the datasets that are crucial for training and evaluating the intrusion detection system. Different datasets (KDDCUP99, NSL KDD, UNSW-NB15) are explored to understand their contents, features, and structure. This step helps in gaining insights into the data that is being worked with..



	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count
0	0	tcp	http	SF	181	5450	0	0	0	0	...	9
1	0	tcp	http	SF	239	486	0	0	0	0	...	19
2	0	tcp	http	SF	235	1337	0	0	0	0	...	29
3	0	tcp	http	SF	219	1337	0	0	0	0	...	39
4	0	tcp	http	SF	217	2032	0	0	0	0	...	49
...	...	...	...	...	...	...	...	...	...	...	...	...
494016	0	tcp	http	SF	310	1881	0	0	0	0	...	255
494017	0	tcp	http	SF	282	2286	0	0	0	0	...	255
494018	0	tcp	http	SF	203	1200	0	0	0	0	...	255
494019	0	tcp	http	SF	291	1200	0	0	0	0	...	255
494020	0	tcp	http	SF	219	1234	0	0	0	0	...	255

494021 rows × 42 columns

Fig 2 KDDCUP dataset

KDDCup99, NSL-KDD and UNSW-NB15 are the most popular and widely used datasets in academic research to evaluate the different malicious activities and detect diverse attacks. The NSL-KDD dataset is the extension of KDD99, it reduces the shortcomings of the old version dataset, precisely, it not only focuses to reduce the redundant data from training and testing but also sets the number of records in training and testing sets. The dataset has 42 features and is divided into 3 categories, traffic features, content features and content features. The KDDCup 99 dataset is one of the popular datasets in IoT with cybersecurity [33], [34], [35], [36], [37], [38], [39], [40]. This dataset provides labelled and unlabeled training and testing data, and it originated from the evaluation program DARPA98 IDS with corresponds to seven and two weeks [33]. The UNSW-NB15 dataset was created by perfectStorm (IXIA) in collaboration with the UNSW Cyber Range Lab to generate moderately aggressive activities and attacks. In dataset, each record in the collection has 47 features, divided into 10 types, including Backdoors, DoS, Analysis, Exploits, Generic, Reconnaissance, Fuzzers for Abnormal Activity, Shellcode, and Worms.

iv) Data Processing:

Data processing involves transforming raw data into valuable information for businesses. Generally, data scientists process data, which includes collecting, organizing, cleaning, verifying, analyzing, and converting it into readable formats such as graphs or documents. Data processing can be done using three methods i.e., manual, mechanical, and electronic. The aim is to increase the value of information and facilitate decision-making. This enables businesses to improve their operations and make timely strategic decisions. Automated data processing solutions, such as computer software programming, play a significant role in this. It can help turn large amounts of data, including big data, into meaningful insights for quality management and decision-making.

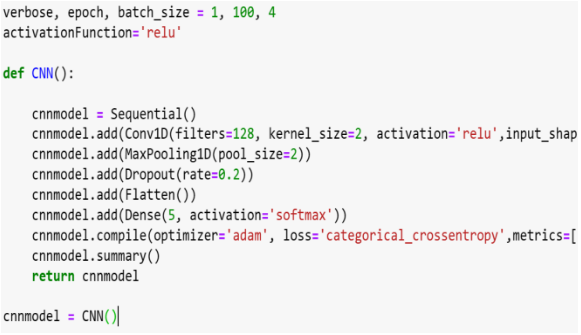
v) Feature selection:

Feature selection is the process of isolating the most consistent, non-redundant, and relevant features to use in model construction. Methodically reducing the size of datasets is important as the size and variety of datasets continue to grow. The main goal of feature selection is to improve the performance of a predictive model and reduce the computational cost of modeling.

Feature selection, one of the main components of feature engineering, is the process of selecting the most important features to input in machine learning algorithms. Feature selection techniques are employed to reduce the number of input variables by eliminating redundant or irrelevant features and narrowing down the set of features to those most relevant to the machine learning model. The main benefits of performing feature selection in advance, rather than letting the machine learning model figure out which features are most important.

vi) Algorithms:

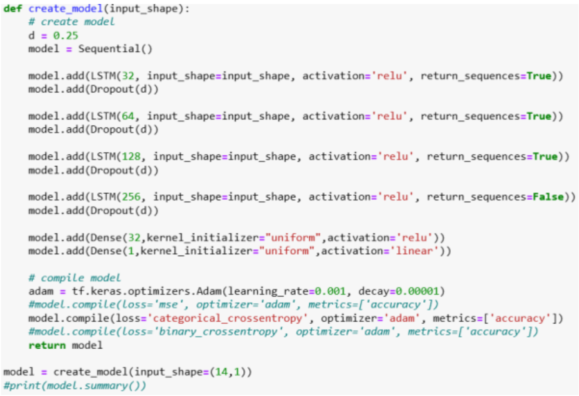
CNNs are specialized for processing grid-like data such as images. They use convolutional layers to automatically and adaptively learn spatial hierarchies of features from the input data. CNNs are widely used in image and video recognition tasks.



```
verbose, epoch, batch_size = 1, 100, 4
activationFunction='relu'

def CNN():

    cnnmodel = Sequential()
    cnnmodel.add(Conv1D(filters=128, kernel_size=2, activation='relu',input_shap
    cnnmodel.add(MaxPooling1D(pool_size=2))
    cnnmodel.add(Dropout(rate=0.2))
    cnnmodel.add(Flatten())
    cnnmodel.add(Dense(5, activation='softmax'))
    cnnmodel.compile(optimizer='adam', loss='categorical_crossentropy',metrics=[
    cnnmodel.summary()
    return cnnmodel

cnnmodel = CNN()
```

Fig 3 CNN

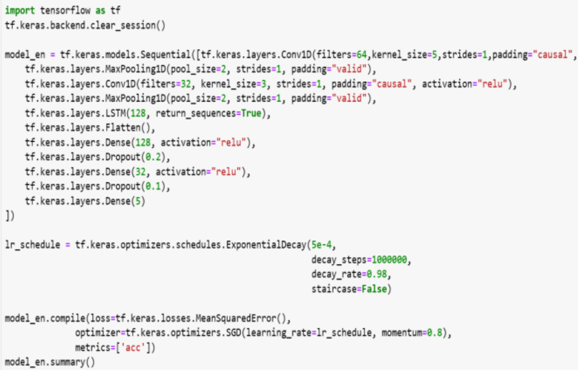
RNNs are designed to work with sequential data by maintaining an internal state or memory. They process inputs in a way that information cycles through a loop, allowing the network to consider previous context. This makes them suitable for tasks involving sequences or time-series data.



```
def create_model(input_shape):
    # create model
    d = 0.25
    model = Sequential()

    model.add(LSTM(32, input_shape=input_shape, activation='relu', return_sequences=True))
    model.add(Dropout(d))

    model.add(LSTM(64, input_shape=input_shape, activation='relu', return_sequences=True))
    model.add(Dropout(d))

    model.add(LSTM(128, input_shape=input_shape, activation='relu', return_sequences=True))
    model.add(Dropout(d))

    model.add(LSTM(256, input_shape=input_shape, activation='relu', return_sequences=False))
    model.add(Dropout(d))

    model.add(Dense(32,kernel_initializer="uniform",activation='relu'))
    model.add(Dense(1,kernel_initializer="uniform",activation='linear'))

    # compile model
    adam = tf.keras.optimizers.Adam(learning_rate=0.001, decay=0.00001)
    #model.compile(loss='mse', optimizer='adam', metrics=['accuracy'])
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    #model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model

model = create_model(input_shape=(14,1))
#print(model.summary())
```

Fig 4 RNN

Combining **CNN and LSTM** leverages CNN's ability to capture spatial features from data (e.g., images) and LSTM's capability to understand and retain temporal dependencies. This hybrid approach is effective for tasks involving both spatial and sequential data[2], [11], [12], [13], [14], [15], [16], [17].

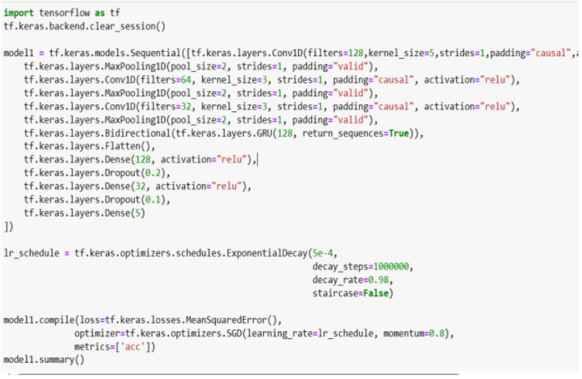


```
import tensorflow as tf
tf.keras.backend.clear_session()

model_en = tf.keras.models.Sequential([tf.keras.layers.Conv1D(filters=64,kernel_size=5,strides=1,padding="causal",
    tf.keras.layers.MaxPooling1D(pool_size=2, strides=1, padding="valid"),
    tf.keras.layers.Conv1D(filters=32, kernel_size=3, strides=1, padding="causal", activation="relu"),
    tf.keras.layers.MaxPooling1D(pool_size=2, strides=1, padding="valid"),
    tf.keras.layers.LSTM(128, return_sequences=True),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation="relu"),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(32, activation="relu"),
    tf.keras.layers.Dropout(0.1),
    tf.keras.layers.Dense(5)
])

lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(5e-4,
                                                             decay_steps=1000000,
                                                             decay_rate=0.98,
                                                             staircase=False)


model_en.compile(loss=tf.keras.losses.MeanSquaredError(),
              optimizer=tf.keras.optimizers.SGD(learning_rate=lr_schedule, momentum=0.8),
              metrics=['acc'])
model_en.summary()
```

Fig 5 CNN + LSTM

ISSN 2454 - 5015

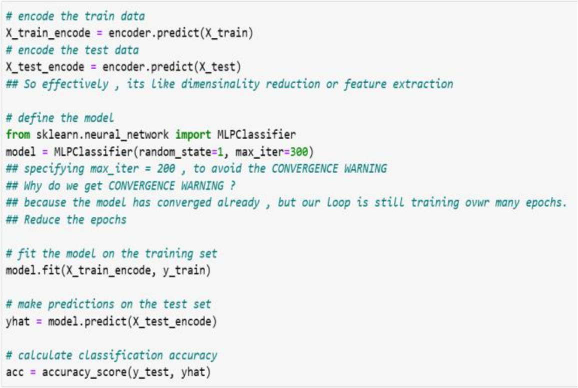
RBM is a generative stochastic artificial neural network used for unsupervised learning. Combining CNN with BiGRU suggests using a mix of convolutional layers for feature extraction (CNN) and bidirectional gated recurrent layers (BiGRU) to capture sequential patterns, potentially for complex pattern recognition tasks.



```
import tensorflow as tf
tf.keras.backend.clear_session()

model1 = tf.keras.models.Sequential([tf.keras.layers.Conv1D(filters=128,kernel_size=5,strides=1,padding="causal",a
    tf.keras.layers.MaxPooling1D(pool_size=2, strides=1, padding="valid"),
    tf.keras.layers.Conv1D(filters=64, kernel_size=3, strides=1, padding="causal", activation="relu"),
    tf.keras.layers.MaxPooling1D(pool_size=2, strides=1, padding="valid"),
    tf.keras.layers.Conv1D(filters=32, kernel_size=3, strides=1, padding="causal", activation="relu"),
    tf.keras.layers.MaxPooling1D(pool_size=2, strides=1, padding="valid"),
    tf.keras.layers.Bidirectional(tf.keras.layers.GRU(128, return_sequences=True)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation="relu"),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(32, activation="relu"),
    tf.keras.layers.Dropout(0.1),
    tf.keras.layers.Dense(5)
])

lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(5e-4,
                                                              decay_steps=1000000,
                                                              decay_rate=0.98,
                                                              staircase=False)


model1.compile(loss=tf.keras.losses.MeanSquaredError(),
              optimizer=tf.keras.optimizers.SGD(learning_rate=lr_schedule, momentum=0.8),
              metrics=['acc'])
model1.summary()
```

Fig 6 RBM

DNNs consist of multiple layers of interconnected nodes, and when organized in a multi-layer perceptron architecture, they are great at learning intricate patterns and features from the data. They are widely used in various machine learning tasks for classification and regression.



```
# encode the train data
X_train_encode = encoder.predict(X_train)
# encode the test data
X_test_encode = encoder.predict(X_test)
## So effectively , its like dimensinality reduction or feature extraction

# define the model
from sklearn.neural_network import MLPClassifier
model = MLPClassifier(random_state=1, max_iter=300)
## specifying max_iter = 200 , to avoid the CONVERGENCE WARNING
## Why do we get CONVERGENCE WARNING ?
## because the model has converged already , but our loop is still training ovwr many epochs.
## Reduce the epochs

# fit the model on the training set
model.fit(X_train_encode, y_train)

# make predictions on the test set
yhat = model.predict(X_test_encode)

# calculate classification accuracy
acc = accuracy_score(y_test, yhat)
```

Fig 7 DNN with MLP

4. EXPERIMENTAL RESULTS

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

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

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

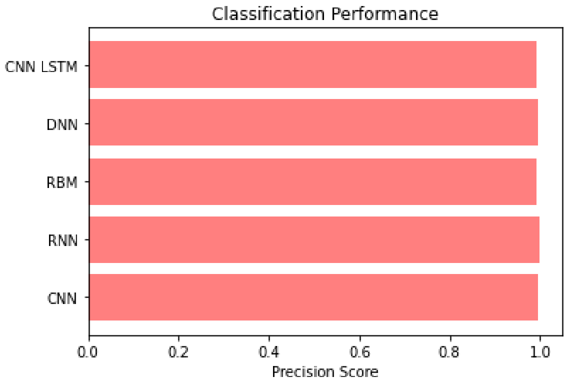


Fig 8 Precision comparison graph

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

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

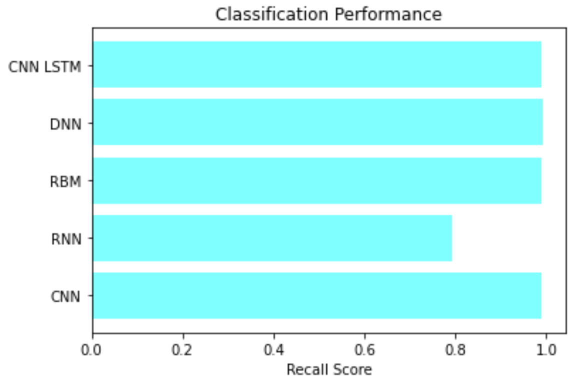


Fig 9 Recall comparison graph

Accuracy: Accuracy is the proportion of correct predictions in a classification task, measuring the overall correctness of a model's predictions.

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

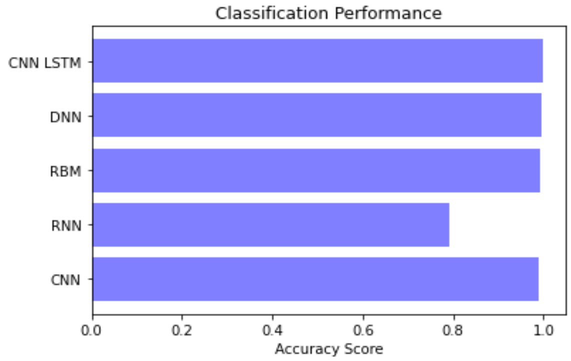


Fig 10 Accuracy graph

F1 Score: The F1 Score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$

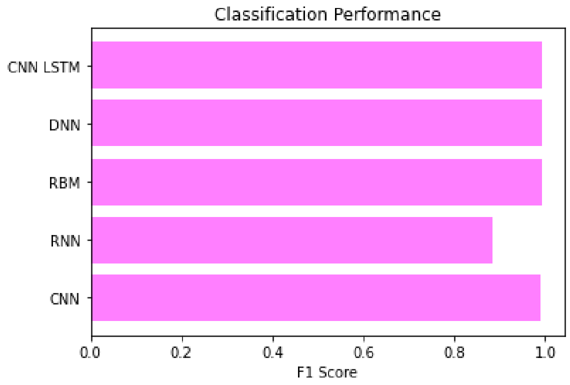
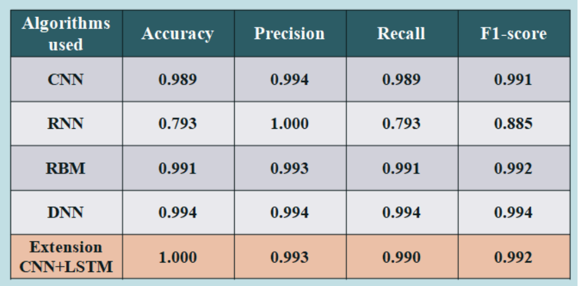
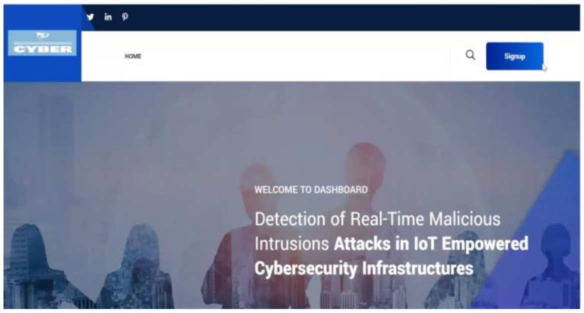


Fig 11 F1Score



Algorithms used	Accuracy	Precision	Recall	F1-score
CNN	0.989	0.994	0.989	0.991
RNN	0.793	1.000	0.793	0.885
RBM	0.991	0.993	0.991	0.992
DNN	0.994	0.994	0.994	0.994
Extension CNN+LSTM	1.000	0.993	0.990	0.992

Fig 12 Performance Evaluation



WELCOME TO DASHBOARD

Detection of Real-Time Malicious Intrusions Attacks in IoT Empowered Cybersecurity Infrastructures

Fig 13 Home page

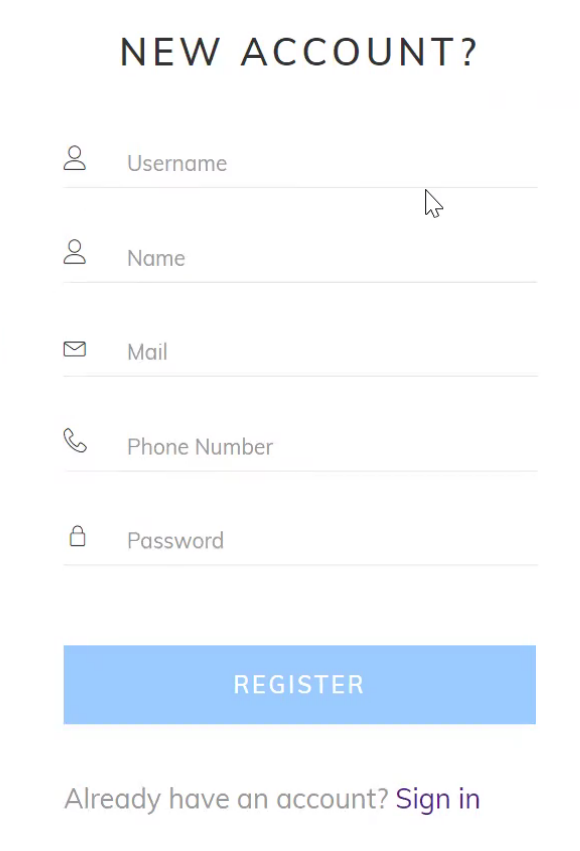
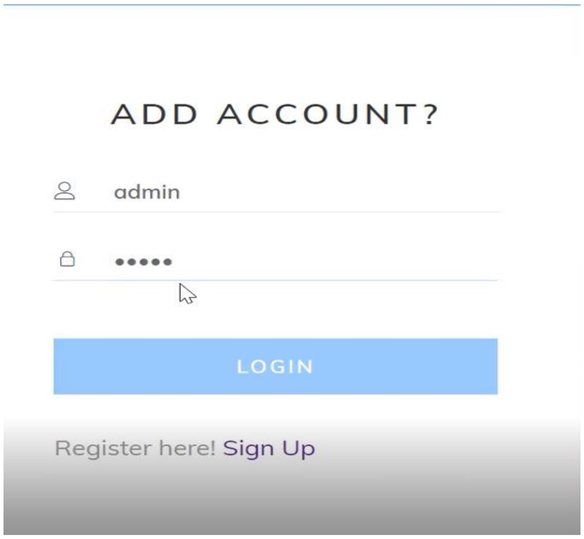


Fig 14 Signin page



admin

•••••

Fig 15 Login page

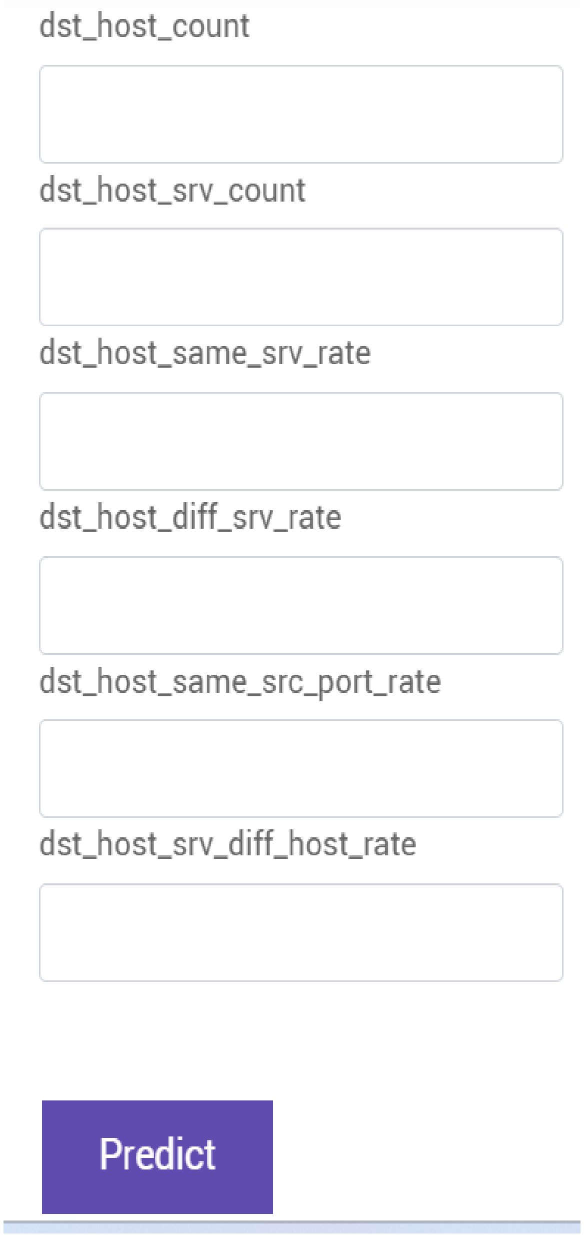
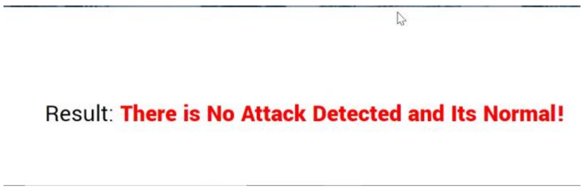


Fig 16User input



Result: **There is No Attack Detected and Its Normal!**

Fig 17 Predict result for given input

5. CONCLUSION

The project places a significant emphasis on the efficacy of utilizing deep learning techniques such as Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN)[2], [11], [12], [13], [14], [15], [16], [17], and Deep Neural Networks (DNN) for early detection of cyber-attacks and identification of malware. By leveraging the capabilities of deep learning, the project showcases its potential to significantly enhance cybersecurity measures, providing a proactive approach to identifying and mitigating threats. Among the various models employed, the extension CNN + LSTM ensemble model stands out by achieving an impressive 99% accuracy. This remarkable result underscores the robustness and adaptability of the ensemble model in real-time scenarios. The combination of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks proves to be a powerful solution for achieving high accuracy in cyber-attack detection, showcasing the effectiveness of ensemble techniques. The integration of the system with Flask contributes to user-friendly deployment, enhancing accessibility and practicality for improving online security. The Flask framework provides a streamlined and intuitive interface, making it easier for users to deploy and interact with the cybersecurity system. This user-friendly aspect enhances the overall usability of the system, ensuring its practical application in real-world scenarios. The project serves as a stepping stone for future advancements in the field, suggesting avenues for more advanced deep learning integration and emphasizing the need for stronger Intrusion Detection Systems (IDS). The focus on real-time detection and classification of malicious activities highlights the project's forward-looking approach, setting the stage for continuous improvement in enhancing cybersecurity measures through innovative technologies.

6. FUTURE SCOPE

Future advancements may involve exploring and incorporating more advanced deep learning techniques and transfer learning approaches. This could lead to more sophisticated models, improving the system's ability to detect complex cyber threats efficiently. The potential for widespread implementation of the proposed system in various companies, including multinational corporations, showcases its scalability and applicability. This could significantly enhance cybersecurity measures for organizations, protecting their valuable assets from potential cyber threats. The integration of edge computing represents a future direction that optimizes real-time threat detection by processing and analyzing data closer to the IoT devices [23]. This approach reduces latency, enhances responsiveness, and optimizes network resources, ultimately bolstering the system's efficiency in identifying and mitigating cyber threats in real-time. The adoption of federated learning in the future can revolutionize the model training process. By allowing collaborative training without sharing raw data, it preserves data privacy while improving the accuracy and robustness of the global model. This collaborative approach can be a game-changer in enhancing the overall cybersecurity of IoT systems.

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] T. Alladi, V. Kohli, V. Chamola, F. R. Yu, and M. Guizani, ‘‘Artificial intelligence (AI)-empowered intrusion detection architecture for the Internet of Vehicles,’’ IEEE Wireless Commun., vol. 28, no. 3, pp. 144–149, Jun. 2021.

[25] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, ‘‘Machine learning and deep learning methods for cybersecurity,’’ IEEE Access, vol. 6, pp. 35365–35381, 2018.

[26] A. S. Dina and D. Manivannan, ‘‘Intrusion detection based on machine learning techniques in computer networks,’’ Internet Things, vol. 16, Dec. 2021, Art. no. 100462.

[27] H. Zhang, J. L. Li, and X. M. Liu, C Dong, ‘‘Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection,’’ Future Gener. Comput. Syst., vol. 122, pp. 130–143, Sep. 2021.

[28] K. A. P. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, ‘‘Internet of Things: A survey on machine learning-based intrusion detection approaches,’’ Comput. Netw., vol. 151, pp. 147–157, Mar. 2019.

[29] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, ‘‘Network intrusion detection for IoT security based on learning techniques,’’ IEEE Commun. Surveys Tuts., vol. 21, no. 3, pp. 2671–2701, Jan. 2019.

[30] J. Toldinas, A. Venčkauskas, R. Damaševičius, Š. Grigaliunas, ¯ N. Morkevičius, and E.

Baranauskas, ‘‘A novel approach for network intrusion detection using multistage deep learning image recognition,’’ Electronics, vol. 10, no. 15, p. 1854, Aug. 2021.

[31] G. Andresini, A. Appice, and D. Malerba, ‘‘Autoencoder-based deep metric learning for network intrusion detection,’’ Inf. Sci., vol. 569, pp. 706–727, Aug. 2021.

[32] 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.

[33] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, ‘‘Deep learning approach for network intrusion detection in software defined networking,’’ in Proc. Int. Conf. Wireless Netw. Mobile Commun. (WINCOM), Oct. 2016, pp. 258–263.

[34] F. Jiang, Y. Fu, B. B. Gupta, Y. Liang, S. Rho, F. Lou, F. Meng, and Z. Tian, ‘‘Deep learning based multi-channel intelligent attack detection for data security,’’ IEEE Trans. Sustain. Comput., vol. 5, no. 2, pp. 204–212, Apr. 2018.

[35] M. A. Ferrag and L. Maglaras, ‘‘DeepCoin: A novel deep learning and blockchain-based energy exchange framework for smart grids,’’ IEEE Trans. Eng. Manage., vol. 67, no. 4, pp. 1285–1297, Nov. 2019.

[36] S. Basumallik, R. Ma, and S. Eftekharnejad, ‘‘Packet-data anomaly detection in PMU-based state estimator using convolutional neural network,’’ Int. J. Electr. Power Energy Syst., vol. 107, pp. 690–702, May 2019.

[37] M. Uddin, A. A. Rahman, A. Alarifi, M. Talha, A. Shah, M. Iftikhar, and A. Zomaya, ‘‘Improving performance of mobile ad hoc networks using efficient tactical on demand distance vector (TAODV) routing algorithm,’’ Int. J. Innov. Comput., Inf. Control, vol. 8, no. 6, pp. 4375–4389, 2012.

[38] 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.

[39] F. Feng, X. Liu, B. Yong, R. Zhou, and Q. Zhou, ‘‘Anomaly detection in ad-hoc networks based on deep learning model: A plug and play device,’’ Ad Hoc Netw., vol. 84, pp. 82–89, Mar. 2019.

[40] M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, and A. E. Hassanien, ‘‘Hybrid intelligent intrusion detection scheme,’’ in Soft Computing in Industrial Applications. Berlin, Germany: Springer, 2011, pp. 293–303.