JIRBDAI Journal of Institutional Research, Big Data Analytics and Innovation

AN IMPROVED DEEP LEARNING BASED INTRUSION DETECTION SYSTEM FOR IOT NETWORKS

Aishatu Yakubu Aliyu, Aminu Adamu, Mamman Maharazu

Computer Science Department, Umaru Musa Yaradua University, Katsina
*Corresponding Author: ayshayaqb@gmail.com

ABSTRACT

The Internet of Things (IoT) has transformed daily life by enabling seamless connectivity among smart devices across various domains, including healthcare, smart homes, and industrial automation. However, the rapid expansion of IoT has led to a massive increase in connected devices and data generation, with an estimated 41.6 billion devices expected to produce 79.4 zettabytes of data by 2025. Deep learning-based intrusion detection systems (IDS) have shown strong potential for learning complex traffic patterns, Most existing models often suffer from overfitting, poor generalization, and high computational cost, making them unsuitable for resource-constrained IoT environments. This study proposes an improved Intrusion Detection System that integrates a Recurrent Neural Network (RNN) with ensemble classifier combining Support Vector Machine (SVM) and Logistic Regression (LR) and Convolutional Neural Network (CNN) for feature extraction .The RNN captures temporal dependencies within network traffic, while the ensemble classifier enhances generalization and reduces overfitting. The model is evaluated using the CICIDS2017 dataset. Experimental results demonstrate an accuracy of 88%, with reduced computational complexity compared to traditional methods. The proposed approach offers a practical solution for real-time IoT security applications, particularly in resource-constrained environments.

KEYWORDS: Intrusion Detection System (IDS); Internet of Things (IoT); Deep Learning (DL); Recurrent Neural Network (RNN); Convolutional Neural Network (CNN); Ensemble Classifiers'; CICIDS2017 dataset.

I. INTRODUCTION

Our lives are being revolutionized by the new computing paradigm that the Internet of Things (IoT) has sparked [1]. Without the need for human intervention, these devices can gather and send data over wireless media by communicating and exchanging data with one another. IoT is widely used in many different applications, including smart home apps, transportation, agriculture, healthcare, and environmental monitoring. [1]. International Data Corporation (IDR) IoT analytics projects that by 2025, there will be over 41 billion connected devices worldwide, generating 74.5 zettabytes of data. Cybercriminals have become interested in this field due to its rapid growth, reliance on IoT devices, and business potential [1], while innovators and entrepreneurs are making it even more appealing by creating and promoting various IoT device applications to improve and simplify our lives [2] Medical IoT (MIoT), which is sensitive due to its direct connection to healthcare, is one of the fastest-growing IoT sectors. Recent trends in cyber-attacks indicate that MIoT device intrusion is at an all-time high and is growing dangerously [3]. Attackers can carry out extensive attacks on internet resources by taking advantage of the flaws in a large number of IoT devices. DDoS attacks happen when a number of compromised devices or machines bombard

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19416531

a targeted device or machine with requests, overloading its server and causing it to crash. As demonstrated by attacks on cloud services and virtual machines, DDoS attacks are very effective because even highly configured machines experience reduced performance [4]. When an attack on an IoT device is successful, it not only damages the device but also gives the attacker access to additional services it offers. Additionally, these devices serve as the control unit for various facilities; in the event of a security breach, an attacker can access the entire control system [5] .IoT devices are systems with limited processing power, it is not feasible to secure vulnerable IoT environments using conventional cyber-security techniques [6]. Large amounts of data are continuously produced by IoT environments, and it can be difficult to identify anomalous data from these data streams. Large amounts of data can be effectively analyzed by deep neural networks, which can also identify patterns and relationships between the data and classify it based on predetermined characteristics [7] .Deep learning models were found to give better accuracy, however most of the models proposed often suffer from overfitting, poor generalization and are computationally intensive. To address these challenges, this study proposed a recurrent neural network and ensemble classifiers to mitigate overfitting, enhance generalization and to improve the detection accuracy of IDS in IoT Networks. The main contributions of this study are as follows:

(a). Integration of RNN for sequential analysis, CNN for feature extraction, and ensemble classifiers (SVM and LR) for improved classification.
(b). Evaluation of the proposed model using the CICIDS2017 dataset.
(c). Comparative analysis with existing models to demonstrate improved detection performance and reduced computational cost.
(d). Achievement of 88% detection accuracy, highlighting the model's suitability for real-time IoT environments.

The rest of this paper is organized as follows: Section 2 presents the relevant literature review; Section 3 presents the proposed methodology and model design; Section 4 discusses implementation and results; and Section 5 concludes with limitations and future directions.

II. RELATED WORKS

The number of devices connected to the internet is rapidly growing, and it has become an indispensable aspect of human life. Specifically, Internet of Things (IoT) devices are now a commonplace aspect of daily life [8]. Despite the impressive growth rate of IoT devices, cyber security concerns have emerged as a major obstacle to the technology's quicker adoption [1]. To reduce the quantity and impact of attacks on IoT systems, it is more crucial than ever to create an intrusion detection system based on deep learning. Khraisat et al. have discussed various methods of intrusion detection in their survey paper by emphasizing the effectiveness of these methods [9]. The authors in [10–13] exhibit promising performance in intrusion detection. Several recent works have proposed deep learning approaches for IoT security. Susilo et al., (2020) developed an algorithm combining Random Forest, Multilayer Perceptron, and Convolutional Neural Networks (CNN) to detect denial-of-service attacks using the Bot-IoT dataset. Although the model achieved strong accuracy, it lacked detailed preprocessing steps and comparisons with state-of-the-art methods [14]. Ravi et al. (2022) introduced a recurrent deep learning-based feature fusion model incorporating an ensemble meta-classifier, achieving 99% accuracy [15]. [16] Otoum et al. (2019) proposed DL-IDS, which integrate Spider Monkey Optimization for feature selection and a Stacked Deep Polynomial Network for classification. Although their results are superior to the proposed method from the accuracy perspective, the network architecture simplicity makes the proposed system a unique contribution to the intrusion detection research domain.

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19416531

Other researchers have investigated ensemble and hybrid models. Abbas et al. (2022) proposed a voting-based ensemble IDS that integrates Logistic Regression, Naïve Bayes, and Decision Trees, reporting an accuracy of up to 98% on the CICIDS2017 dataset. However, their work primarily focused on classification accuracy without addressing issues of model scalability or resource efficiency [17]. Similarly, Oseni et al. (2023) focus specifically on the Internet of Vehicles (IoV) using an explainable deep learning-based intrusion detection framework, but their frameworks are limited in generalizability to broader IoT applications [18]. Recent studies have also investigated hybrid neural network architectures. Morshedi et al. (2024) proposed a neural network for IoT security which comprises denseNet, CNN and hybrid of CNN-LSTM for detecting DoS, DDos and other attacks, the results show outstanding accuracy, reaching 0.997 on the test data. The study aimed to compare the effectiveness of different neural network architectures using the CIC-IDS2017 dataset in the context of deploying machine learning models in IoT infrastructure The research focused on identifying suitable architectures that could achieve high performance while minimizing computational requirements, rather than increasing the depth of the neural networks However, to address the slight variations seen in the CNNLSTM and DenseNet architectures during mid-training, more investigation and optimization might be necessary. However, most of the deep learning-based models proposed in the afore-mentioned researches are computationally intensive and difficult to implement on resource-limited devices in IoT networks. In contrast, our study presents a simple, lightweight and suitable for small scale binary classification task [5].

Table 1: Summary of some Related Works

Author/Year	Problem solved	Method	Achievements	Gaps
susilo et al., (2020)	DoS	Designed DoS attack detection using Random forest, MLP and CNN on bot-IoT dataset	CNN and RF produced best detection results for DoS attacks (91.27%).	Focused mainly on DoS attacks; limited generalization
Idrissi et al., (2021)	Botnet attacks	Built a CNN model to detect botnet attacks	Fast and accurate system that can detect attacks in real time.	Limited to botnet attack.
Ravi et al., (2022)	Dos,DDoS and other attacks	The authors developed an intelligent network intrusion detection system Using a recurrent deep learning combined PCA for feature extraction incorporating meta-ensemble classifier.	99% accuracy achieved	Very complex architecture, hard to deploy on IoT edge devices.
Oseni et al., (2023)	_	The author proposed security model for internet of vehicle (IoV) using Explainable deep learning-based IDS	99.15% accuracy achieved	There is problem of overfitting which has yet to be addressed through different datasets. Also not

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19416531

				suitable for general IoT networks.
Morshedi et al., (2024)	DoS, DDoS and other attacks	Compared densenet,CNN, and hybrid CNN-LSTM using CICIDS2017	Achieved up to 99.7% accuracy	Instability during model training ; consumed a lot of energy.
Awajan, 2023		Designed six-layer DNN + SVM for IoT networks.	Achieved 93.7% accuracy across 5 attacks.	Overfitting and poor generalization across different datasets
Banaama et al.(2022)	survey	Many researchers are now using DL models to improve the detection efficiency and reduce false alarms in IoT environments	Demonstrated that DL can enhance IoT threat detection speed and accuracy.	

Overall, the literature highlights that while deep learning models provide high detection Accuracy, however most of the models proposed often suffer from ovefitting and poor generalization and are computationally intensive difficult to implement on resource-limited devices in IoT networks, limiting real-time applicability. To address these gaps, this paper proposes IDS that combine RNN for sequential pattern learning, and an ensemble classifier integrating Support Vector Machine and Logistic Regression.

III. METHODOLOGY

The proposed methodology used in this research includes data collection, preprocessing, data splitting, model training and testing, and evaluation. The dataset utilized in this study is CICIDS2017 [5] which were created by the Canadian institute for cybersecurity in 2017.The dataset is labeled meaning each record is annoted as either benign or malicious. It is widely used in the research community developing and testing ML and DL models and other cybersecurity solution. To prepare the dataset, missing values are checked and eliminated. Categorical variables were encoded to numerical values also the dataset underwent standardization which eliminates the mean and scales the data to achieve unit variance,64 features were automatically extracted from 78 features in the dataset using Convolutional Neural Network (CNN) algorithm. For training the models, dataset was divided into 80% for training set and 20% for testing. Figure 2 illustrates the proposed model for intrusion detection in the IoT network. SVM and LR were used for ensemble machine learning anomalous classification to improve the overall performance. Ensemble method reduces the risk of over fitting and increases generalization.

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19416531

Figure 2: The Proposed IDS Model

A. Dataset Collection

The CICIDS 2017 dataset, a comprehensive collection of network traffic data intended for research and development in the field of cyber security, specifically in the area of intrusion detection systems (IDS), was utilized in this study. It is a labeled dataset with 78 features. The class label, which identifies the sample as an attack or normal traffic, is the dataset's final feature. This dataset contains 14 different types of attacks. In terms of IoT detection, some of the attack types are important. The dataset contains a variety of attacks, including port scans, DoS and DDoS attacks, and botnet attacks. Additionally, the dataset contains complete packet payloads in pcap format, which can be utilized for in-depth network traffic analysis. Additionally, the dataset contains profiles of every network flow, which enables researchers to learn more about the features of the traffic, captured in a controlled environment.

B. Preprocessing of Dataset

This section presents the first stage of the methodology, before applying any ML approach to the dataset it is essential to preprocess the data to ensure high accuracy and improve the model. In this study we performed several preprocessing steps on the CICIDS2017 dataset, including handling missing values, scaling the data to ensure all features are on the same scale and transforming categorical values into numerical values. These steps are applied to make the dataset suitable for training and testing our propose model. This procedure includes:

(a). Data cleaning: In data cleaning we leveraged python libraries including pandas and numpy to verify the dataset for missing values or NaN /Null values, as part of cleaning process.

(b). Data encoding: Label Encoder is employed to convert labels in the data Frame to numeric values.

(c). Data exploration: Data exploration is the process of analyzing and summarizing a dataset to understand its structure, content, and main features. Figure 3 is the description of the dataset.

```
df.describe()
```

	Destination Port	Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Total Length of Bwd Packets	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	Fwd Packet Length Std	...	act_data_pkt_fwd	min_seg_size_forward	Active Mean
count	66237.000000	6.623700e+04	66237.000000	66237.000000	66237.000000	6.623700e+04	66237.000000	66237.000000	66237.000000	66237.000000	...	66236.000000	66236.000000	6.623600e+04
mean	6454.605553	1.388736e+07	5.180262	5.119223	695.622567	6.716837e+03	351.442064	23.465797	110.715335	135.438642	...	3.447793	22.101455	1.585840e+05
std	16710.894455	2.815843e+07	16.873659	23.797124	3230.046907	4.222698e+04	1435.510506	136.608403	400.376692	613.421563	...	13.108006	4.844533	8.489147e+05
min	0.000000	0.000000e+00	1.000000	0.000000	0.000000	0.000000e+00	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000e+00
25%	80.000000	4.995600e+04	2.000000	1.000000	26.000000	0.000000e+00	6.000000	0.000000	6.000000	0.000000	...	1.000000	20.000000	0.000000e+00
50%	80.000000	9.949050e+05	3.000000	4.000000	30.000000	2.020000e+02	20.000000	0.000000	8.666667	0.000000	...	2.000000	20.000000	0.000000e+00
75%	80.000000	7.639831e+06	5.000000	5.000000	64.000000	1.160100e+04	38.000000	6.000000	35.000000	10.263203	...	4.000000	20.000000	1.005000e+03
max	61538.000000	1.199983e+08	1681.000000	2942.000000	120783.000000	4.991419e+06	11680.000000	1472.000000	3867.000000	6692.644993	...	1680.000000	52.000000	1.000000e+08

8 rows × 78 columns

Figure 3: Data Exploration

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19416531

C. Feature Extraction with Convolutional Neural Network (CNN)

CNN is a kind of neural network used for classifying and recognizing images and videos. It is very good at identifying patterns in data, including edges, textures, and objects. The most pertinent features from the dataset are extracted in this study using CNN, which enhances the model's performance and lowers computational complexity. Prior to classification, the 1DCNN architecture created in this study extracts hierarchical features from the CICIDS dataset. The proposed architecture consists of multiple convolutional layers, pooling layer, dropout and dense layer. The first CNN layer consists of 64 filters with a kernel size of 3 along with ReLU as an activation function. This layer transforms the input into a higher dimensional feature map. Followed by a Max-pooling1D having a pool size of 2, which reduces the feature map dimensionality and computational cost while retaining the most significant features. It is then followed by second convolutional layer consisting of 128 filters with a kernel size of 3. Another max-pooling is introduced; the output is then flattened into a single vector to enable integration with dense layers. A dense layer with 128 neurons is used to learn high levels feature interactions, followed by a dropout layer with a dropout rate of 0.5 to prevent overfitting and enhance model generalization. Figure 4 shows the proposed 1D-CNN.

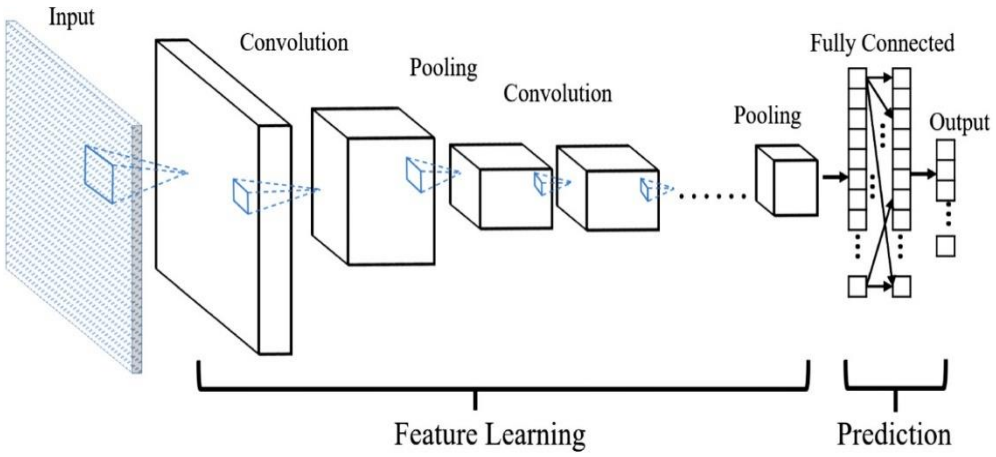

Figure 4: Proposed 1D-CNN Architecture [19]

D. Ensemble Method

When two or more machine learning algorithms is combined, ensemble learning yields better results than when the individual algorithms are used separately. Rather than depending on a single model, the predictions made by each learner are combined using a combination rule to produce a single, more accurate prediction. The accuracy and diversity of the base learners are the primary factors that determine how well ensemble learning techniques work. A machine learning model is deemed accurate if it can effectively generalize to new situations [21]. When combining ensemble base models, majority voting is the most popular method. When it comes to classification and regression tasks, majority voting is the most widely used and logical combination method. The class with the majority vote is returned as the ensemble prediction in classification problems after the predictions for each class are added together.

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19416531

In this study to improve prediction performance, an ensemble model was created using a Voting Classifier. Two base learners Support Vector Machine and Logistic Regression were combined using soft voting. In soft voting, the predicted class probabilities from each classifier are averaged, and the class with the highest probability is selected as the final output. Soft voting typically yields better generalization as it considers the confidence level of each classifier rather than relying on a simple majority vote.

Support Vector Machines (SVM): SVM is a supervised machine learning method that maximizes the distance between the hyperplane and the nearest sample points of each class by using data attributes to create a splitting hyperplane between two or more classes. SVMs are particularly well-suited for datasets with a large number of feature attributes but a small number of sample points because of their remarkable capacity for generalization [8]. Figure 5 shows the working of SVM

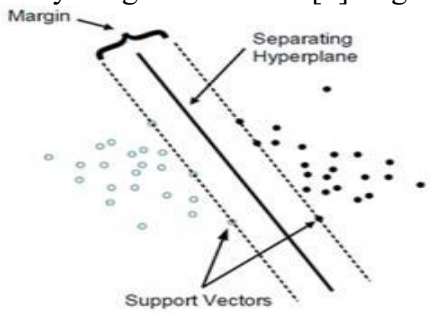

Figure 5: Working of Support Vector Machine

Logistic Regression Logistic (LR) is a powerful statistical technique based on probability that is frequently used in supervised machine learning to solve classification problems. An input instance is given a probability value by the binary classification; values greater than 0.50 classify it as "class A," and values less than 0.50 classify it as "class B." as shown in Figure 6.

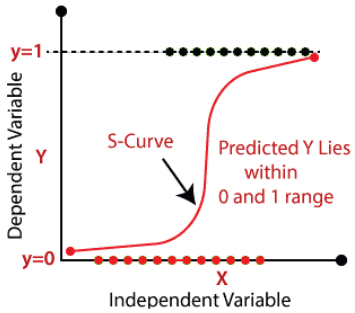

Figure 6: Logistic regressions [21]

The sigmoid function, a logistic feature, is used to calculate the probability that a new instance will belong to a specific class. LR is most commonly applied to classification problems, though it can also be applied to regression problems [21].

E. Recurrent Neural Network (RNN) for Intrusion Detection

The RNN layer processes the features extracted by the CNN and helps in detecting patterns over time that indicate DDoS and other attacks. A three-layer recurrent neural network has been

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19416531

designed and optimized to detect intrusion in this experiment. The first layer is SimpleRNN where input sequence data is fed into, this layer Processes sequential data and outputs a hidden state representing the sequence. The hidden state serves as the input to dense layer which extract high level features from the RNN output. The output is then passed to last dense layer to produce the final prediction. The figure 7 illustrates the proposed RNN.

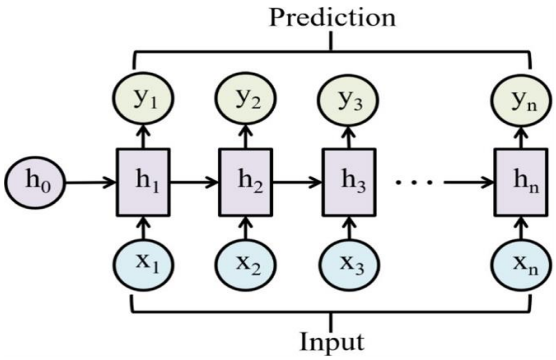

Figure 7: The Proposed Model (RNN) [19].

The Rectified Linear Activation Unit (ReLU) function is used for the hidden layers and the sigmoid for the output layer [1]. These two activation functions are defined by Equation (1) and (2)

$$\text{ReLU}(z) = \max(0, z) \quad (1)$$

$$y = \alpha(z) = \frac{1}{1+e^{-z}} \quad (2)$$

Binary cross entropy loss has been used for loss function to minimize the error defined by Equation (3). The model minimizes the loss function using the Adam optimizer by updating the weights during training.

$$\text{Cross Entropy} = -\sum_i p(y_i) \log q(y_i) \quad (3)$$

F. Basic RNN Architecture

Standard RNNs' basic architecture and working principle: RNNs are made to process sequential data by storing information about prior inputs in a hidden state. An input layer, a hidden layer, and an output layer make up the fundamental architecture. RNNs have recurrent connections, which enable information to cycle within the networks, in contrast to feed forward neural networks. At each time step t, the RNN takes an input vector x_t and updates its hidden state ht, using the following equation:

$$h_t = \sigma_h(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (4)$$

Where W_{xh} is the weight matrix between the input and hidden layer, w_{hh} is the weight matrix for the recurrent connection, b_h is the bias vector, and σ_h is the activation function, typically the hyperbolic tangent function (tanh) or the rectified linear unit [19]. The output at each time step, t, is given by the following:

$$y_t = \sigma_y(W_{hy}h_t + b_y) \quad (5)$$

Where w_{hy} is the weight matrix between the hidden and output layers, by is the bias vector, and σ_y is the activation function for the output layer. The core of RNN operations involves the recurrent computation of the hidden state, which integrates the current input with the previous hidden state because of this recurrent computation, RNNs can display dynamic temporal behavior. The network's behavior is greatly influenced by the selection of activation function σh, which

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19416531

introduces non-linearity and allows the network to recognize and represent intricate patterns in the data [19]. Rectified linear units (ReLUs) are a popular activation function in RNNs. If the input is positive, the ReLU function outputs it directly; otherwise, it outputs zero. Because gradients can move through the network more efficiently thanks to this simplicity, the vanishing gradient issue is somewhat mitigated [19].

The input values are squashed by the sigmoid function to fall between 0 and 1. Similar to tanh, it produces values in a different range, which makes it helpful in situations where the output must be interpreted as probabilities. For classification tasks, cross-entropy is a popular loss function that works well. The dissimilarity between the actual and predicted class probabilities is effectively captured by cross-entropy, which calculates the difference between the true label distribution p and the predicted label distribution q. When the output is a probability representing two classes (0 or 1), the cross-entropy loss reduces to binary cross-entropy loss in the context of binary classification [19].

G. Implementation Approach

The tools used for experimental programs are google Colab with Python Programming language with CIC-IDS-2017 dataset a Canadian institute for cybersecurity developed in 2017. These tools help to preprocess the data, train the model, and evaluate the performance of the model. The model is trained and evaluated on the CICIDS 2017 dataset. The dataset was split into training (80%) and testing (20%) sets. The model achieved a detection accuracy of 88% for DDoS and others attacks. In comparison with traditional methods, our proposed model significantly reduces computational complexity while maintaining high detection performance.

H. Evaluation Metrics

To ensure a comprehensive evaluation of the proposed model, multiple performance metrics and validation techniques were employed. A confusion matrix was used to analyze the classification results in terms of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). This provides a detailed understanding of the model's prediction capability.

(a). Accuracy: It is defined as the percentage of correctly classifying the records over the total number of records.

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

(b). Precision (P): Defined as the ratio of the number of True Positives (TP) records divided by the sum of True Positives (TP) and False Positives (FP) Classified records.

$$\text{Precision} = \frac{TP}{TP+FP}$$

(c). Recall (R): Defined as the ratio of number of True Positives records divided by the sum of True Positives and False Negatives (FN) classified records.

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

(d). F1 measure: Defined as the harmonic mean of Precision and Recall and represents a balance between them.

$$\text{F1 score} = 2 * \frac{Preccision*recall}{precision+recall}$$

Additionally, the Receiver Operating Characteristic (ROC) curve and Area under the Curve (AUC) were used to evaluate the model's ability to distinguish between attack and normal traffic across different thresholds. Furthermore, computational performance was evaluated using training time and inference time to support the claim of reduced computational complexity.

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19416531

IV. RESULTS AND DISCUSSION

Table 4.1 presents the performance of the proposed model (RNN) for Attack detection.

Table 2: Performance Evaluation Metrics

Metrics	Value (%)
Accuracy	0.88
Precision	0.81
Recall	0.99
F1score	0.89

The results demonstrate that the proposed model achieves high recall, indicating its strong capability to detect DDoS and other attacks. Precision is slightly lower, indicating the presence of false positives, but the overall F1-score confirms balanced performance.

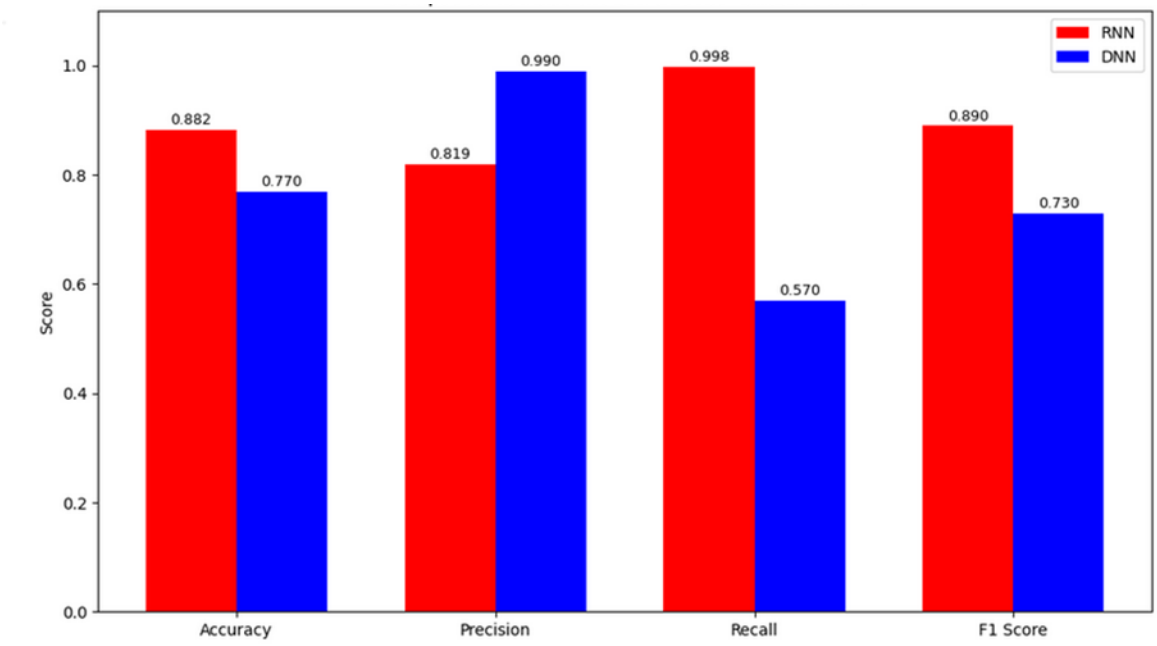

Figure 8: Comparison between the Proposed and the Benchmark Paper

The results demonstrate that the proposed model significantly improves recall (0.99), demonstrating excellent ability to correctly detect attacks such as DDoS in IoT networks. Although the precision value (0.81) shows that some false positives still occur, the high F1-score (0.89) reflects a balanced trade-off between precision and recall. Overall, the proposed RNN-based IDS with ensemble learning outperform the existing DNN approach in all evaluated performance metrics. This confirms that integrating RNN architecture with ensemble classifiers leads to more robust detection, better generalization, and s reduced overfitting compared to traditional deep learning models.

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19416531

A. Confusion Matrix Analysis

The confusion matrix in Figure 9 demonstrates the effectiveness of the proposed model, correctly identifying 7010 attack instances and 4569 normal instances. The model achieves a very low number of false negatives (15), resulting in a high recall of 0.99, which indicates excellent capability in detecting malicious traffic. However, 1654 false positives were observed, leading to a lower precision of 0.81. This suggests that some normal traffic is misclassified as attacks. Overall, the model prioritizes attack detection, minimizing missed threats at the cost of increased false alarms, a trade-off that is acceptable in intrusion detection systems.

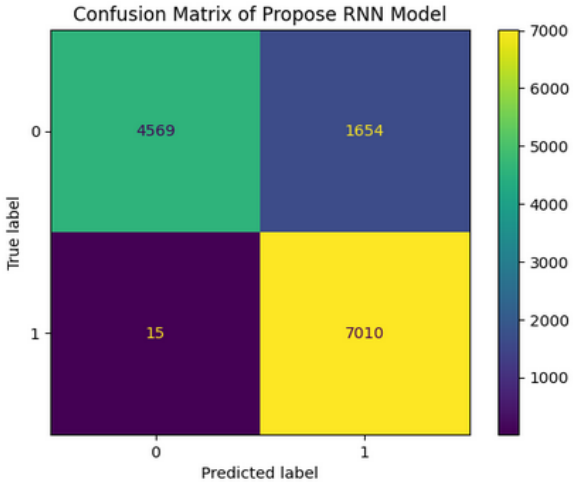

Figure 9: The Confusion Matrix of Proposed RNN Model

B. ROC Curve Analysis

The Receiver Operating Characteristic (ROC) curve was used to evaluate the classification performance of the proposed model. As shown in Figure 10 the curve rises steeply towards the top-left corner, indicating a high true positive rate with a relatively low false positive rate. The Area Under the Curve (AUC) is 0.95, which demonstrates excellent discriminative capability of the model in distinguishing between normal and attack traffic [14]. This result confirms that the proposed RNN-Ensemble model performs significantly better than random classification and is highly effective for intrusion detection in IoT environments. Although the model achieves high recall, the slightly lower precision suggests the presence of false positives. However, the ROC curve indicates that the model can be further optimized by adjusting the decision threshold to achieve a better balance between precision and recall.

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19416531

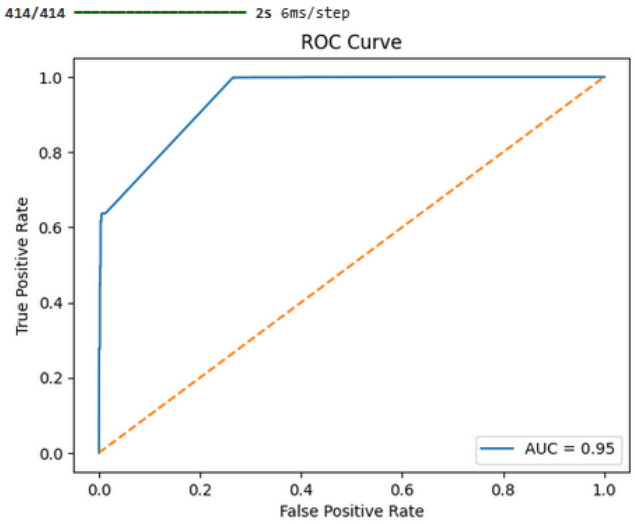

Figure 10: ROC curve of RNN Algorithm

C. Computational Performance

The computational efficiency of the model was evaluated using training and inference time. The results indicate that the proposed model achieves reduced computational cost compared to deeper architectures, due to the use of RNN- based ensemble and CNN-based feature extraction. This makes the model suitable for deployment in resource-constrained IoT environments.

Table 3: Computational Performance

Metrics	Time (s)
Training Time	318.94 seconds
Inference Time	2.96 seconds
Current Memory Usage	0.27 M

D. Comparative Analysis with Existing Models

To validate the effectiveness of the proposed model, a comparison was conducted with several state-of-the-art intrusion detection approaches evaluated on the CICIDS2017 dataset.

Table 4: Comparative Analysis with Existing Models

Article/ Year	Approach	Accuracy	Precision	recall	F1 score
Abbas et al., (2022)	Ensemble (LR, NB, DT)	0.98	0.95	0.94	0.94
Ravi et al., (2022)	RNN + Meta Ensemble	0.99	0.98	0.98	0.98
Morshedi et al., (2024)	CNN-LSTM	0.997	0.99	0.99	0.99

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19416531

Awajan (2023)	DNN with SVM	0.77	0.99	0.53	0.73
Proposed Model	RNN + CNN + Ensemble	0.88	0.81	0.99	0.89

Although some models achieve higher accuracy, they often require high computational resources. The proposed model provides a balanced trade-off between detection performance and computational efficiency, making it suitable for resource-constrained IoT environments.

E. Limitation and Future Works

The improved intrusion detection using RNN based IDS demonstrate excellent experimental results and analysis. The experimental findings back up the assertion that it is a workable and efficient way to identify DDoS and other attacks in Internet of Things networks. Nevertheless, the suggested intrusion detection system has certain flaws, just like any computer system. These restrictions make it possible to carry out additional research and improve the suggested system. One dataset is the focus of the model we suggested. To improve detection accuracy and system performance, future research may use different kinds of RNNs or sophisticated techniques in the ensemble model. To further validate its performance, the suggested framework might also be put into practice in an actual IoT environment. Using massive datasets with different attacks is another crucial objective.

V. CONCLUSION

The ease of use that the Internet of Things (IoT) provides in every aspect of our daily lives has made it an integral part of our lives. This has made it possible for malicious individuals and attackers to jeopardize the availability, confidentiality, and integrity of these IoT networks and devices. In this study the proposed Recurrent Neural Network with ensemble classifier had been a simple Intrusion Detection System (IDS) suitable for small scale binary classification tasks with sequential data. In order to achieve extremely high detection accuracy, the ensemble approach is based on machine learning. Additionally, the study used CNN to extract features, which reduced computational complexity. Also it that when compared to the simple approach, the ensemble approach yields higher detection accuracy.

REFRENCES

[1] A. Awajan, “A novel deep learning-based intrusion detection system for IoT networks,” 2023.
[2] V. G. Saranya Vaishalini, A. Ramathilagam, R. Palanikumar, P. Raghavan, P. Gopikannan, and K. Manikandan, “Comprehensive survey of deep learning-based intrusion detection and prevention systems for secure communication in the Internet of Things,” 2024.
[3] J. Soldatos, S. Gusmeroli, P. Malo, and G. Di Orio, “Internet of Things applications in future manufacturing,” in *Digitising the Industry Internet of Things: Connecting the Physical, Digital and Virtual Worlds*. Delft, Netherlands: River Publishers, 2022, pp. 153–183.
[4] N. Faruqui, M. A. Yousuf, M. Whaiduzzaman, A. Azad, A. Barros, and M. A. Monsi, “LungNet: A hybrid deep-CNN model for lung cancer diagnosis using CT and wearable sensor-based medical IoT data,” *Computers in Biology and Medicine*, vol. 139, p. 104961, 2021.

https://doi.org/10.5281/zenodo.19416531

[5] R. Morshedi, M. Matinkhah, and M. T. Sadeghi, "Intrusion detection for IoT network security with deep learning," *Journal of Artificial Intelligence and Data Mining*, vol. 12, no. 1, pp. 37–55, 2024, doi: 10.22044/jadm.2023.13539.2471.
[6] N. Harada, N. Shibata, S. Kaneko, K. Honda, J. Terada, Y. Ishida, K. Akashi, and T. Miyachi, "Quick suppression of DDoS attacks by frame priority control in IoT backhaul with construction of Mirai-based attacks," *IEEE Access*, vol. 10, pp. 22392–22399, 2022.
[7] Y. Zheng, Z. Li, X. Xu, and Q. Zhao, "Dynamic defenses in cyber security: Techniques, methods and challenges," *Digital Communications and Networks*, vol. 8, pp. 422–435, 2022.
[8] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for Internet of Things (IoT) security," *IEEE Communications Surveys & Tutorials*, vol. 22, pp. 1646–1685, 2020.
[9] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Journal of Big Data*, vol. 2, no. 20, 2019, doi: 10.1186/s42400-019-0038-7.
[10] A. Alazab, A. Khraisat, M. Alazab, and S. Singh, "Detection of obfuscated malicious JavaScript code," *Future Internet*, vol. 14, no. 217, 2022.
[11] B. I. Farhan and A. D. Jasim, "Survey of intrusion detection using deep learning in the Internet of Things," *International Journal of Computer Science and Mobile Computing*, 2022, doi: 10.52866/ijcsm.2022.01.01.009.
[12] N. A. Khan, A. Awang, and S. A. A. Karim, "Security in Internet of Things: A review," *IEEE Access*, 2022, doi: 10.1109/ACCESS.2022.3209355.
[13] Y. H. Reddy, A. Ali, P. V. Kumar, M. H. Srinivas, K. Netra, V. J. Achari, and R. Varaprasad, "A comprehensive survey of Internet of Things applications, threats, and security issues," *SAR Journal of Engineering and Technology*, vol. 4, no. 4, 2022, doi: 10.36346/sarjet.2022.v04i04.0032.
[14] B. Susilo and R. F. Sari, "Intrusion detection in IoT networks using deep learning algorithm," Information, vol. 11, no. 5, p. 279, May 2020, doi: 10.3390/info11050279.
[15] V. Ravi, R. Chaganti, and M. Alazab, "Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system," *Computers & Electrical Engineering*, vol. 102, p. 108156, 2022.
[16] Y. Otoum, D. Liu, and A. Nayak, "DL-IDS: A deep learning–based intrusion detection framework for securing IoT," *Transactions on Emerging Telecommunications Technologies*, article e3803, 2019.
[17] Abbas, A., Khan, M. A., Latif, S., Ajaz, M., Shah, A. A., & Ahmad, J. (2022). A new ensemble-based intrusion detection system for Internet of Things. Arabian Journal for Science and Engineering, 47(2), 1805–1819. https://doi.org/10.1007/s13369-021-06086-5
[18] A. Oseni et al., "An explainable deep learning framework for resilient intrusion detection in IoT-enabled transportation networks," *IEEE b Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 1000–1014, Jan. 2023, doi: 10.1109/TITS.2022.3188671.
[19] Mienye, I.D.; Swart, T.G. A Comprehensive Review of Deep Learning: Architectures, Recent Advances, and Applications. Information **2024**, 15, 755.https://doi.org/10.3390/ info15120755.
[20] Mienye, I. D., & Sun, Y. (2022). A survey of ensemble learning: Concepts, algorithms, applications, and prospects. IEEE Access, 10, 99129–99149. https://doi.org/10.1109/ACCESS.2022.3207287