International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Hybrid Machine Learning Model for Efficient Botnet Attack Detection in IoT Environment

J. JyothiBai, M. Sai Madhav, P. Soumya Srudeep, P. Kaushik
Department Of Computer Science and Engineering(AI&ML),CMRIT
Hyderabad, Telangana, India
Department Of Computer Science and Engineering (AI&ML), CMRIT
Hyderabad, Telangana, India
Department Of Computer Science and Engineering (AI&ML), CMRIT
Hyderabad, Telangana, India
Department Of Computer Science and Engineering (AI&ML), CMRIT
Hyderabad, Telangana, India

Abstract - The recent proliferation of IoT devices has made it much more difficult to secure network infrastructures against cyber attacks by botnets. The classical intrusion detection mechanisms do not always find it easy to detect new patterns of the botnets due to their dynamic nature and encrypted communication. This paper introduces a hybrid stacked deep learning network to identify botnet attacks based on the UNSW-NB15 dataset. It is an architecture that integrates Artificial and Convolutional Neural Networks (ANN), Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN) into a stacking ensemble learning model known as ACLR. The system consists of the structured preprocessing of data, encoding features and ensemble learning to enhance predictive performance. The proposed ACLR model achieved a testing accuracy of 96.42%, outperforming individual model which is higher than individual ANN (95.38%), CNN (95.35%), RNN (95.12%), and LSTM (94.44) models. The model also scores 0.9951 in ROC-AUC and 0.9963 in PR-AUC, meaning that it has a high level of separation of classes and stability when faced with imbalance of the classes. The ACLR framework presents a dependable and scalable solution to improving the security of the IoT network by efficiently modeling the spatial and temporal traffic trends.

Keywords: Botnet Detection, IoT Security, Deep Learning, Stacking Ensemble, UNSW-NB15, Cybersecurity Analytics

I. INTRODUCTION

The fast development of internet technologies has transformed the way individuals, organizations and industries work. The Internet of Things (IoT) is one of the biggest technological releases in recent years as connected systems become more and more relied upon, allowing smart homes, healthcare monitoring systems, industrial automation, and intelligent transportation. However, this quick integration of IoT devices into everyday life has also widened the attack surface for cyber threats. In particular, attacks by botnets have turned out to be dangerous and catastrophic to network based infrastructures.

A botnet is a set of compromised computers which are remotely operated to conduct systematic malicious acts These are DDoS attacks, spamming, phishing, data theft, spreading ransomware, and system unauthorized access. Many IoT devices remain highly vulnerable due to limited processing power, weak authentication mechanisms, fragile authentication, obsolete software, and ineffective security configurations. When they are compromised they will be able to silently be incorporated into a botnet and participate in a massive attack without their owners being aware. Traditional intrusion detection systems are based on signature-based and rule-based detection.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Although these techniques are effective in terms of the familiar threats, they are not very effective in detecting the new or evolving attack patterns. Correspondingly, the ever-changing character of modern botnets, encryption methods, as well as ingenious attack schemes contributes to the problem. As network traffic increases in volume and variety, manual feature engineering and static detection models fall short.

In recent years, machine learning and deep learning methods have shown promising results in network intrusion detection. Deep learning models can automatically identify complex patterns from raw network traffic data without needing extensive manual feature selection. Architectures such as Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Recurrent Neural Networks (RNN) have been widely studied for traffic classification. CNNs are used in finding spatial patterns in structured data, while LSTM and RNN models capture sequential dependencies and time-related relationships in network flows. Despite their benefits, individual deep learning models may face issues like overfitting, high variance, or difficulty generalizing across different attack types. This leads to the use of ensemble learning strategies. Stacking, a strong ensemble technique, combines several base learners to create a more accurate and stable prediction model. By merging the strengths of different architectures, stacking can enhance detection performance and reduce bias in the model.

This research proposes a hybrid stacked deep learning model for effective multi-class botnet attack detection. The model combines ANN, CNN, LSTM, and RNN in a layered ensemble framework to improve classification ability. The UNSW-NB15 dataset is used for experiments, which includes several attack categories along with normal network traffic. The main goal of this work is to boost detection accuracy while staying robust across various attack types.

The key contributions of this study are as follows. First, a hybrid stacking model that includes four deep learning architectures is designed to utilize both spatial and temporal traffic features. Second, thorough preprocessing methods are applied to get the network traffic data ready for deep learning training. Third, the proposed model is tested using several performance metrics such as accuracy, precision, recall, F1-score, ROC-AUC, and cross-validation to confirm its reliability. Finally, the performance of the hybrid model is compared with individual deep learning models to show its advantages.

The rest of this paper is organized as follows. Section II discusses related work in botnet detection and deep learning methods. Section III presents the proposed methodology and model structure. Section IV describes the experimental results and comparison analysis. Section V concludes the study and suggests potential future research directions.

II. RELATED WORK

The use of deep learning methods has gained popularity because they can be used to extract meaningful patterns directly out of raw data. Convolutional Neural Networks (CNN) are often used for network intrusion detection because they effectively capture spatial features from structured traffic data. They are particularly good at handling imbalanced datasets through normalization techniques[1]. Traditional detection systems that rely on signatures and rules were effective for known threats but had trouble with new attacks. This limitation led to the use of anomaly-based methods that spot deviations in network behavior[2]. Recurrent Neural Networks (RNN) and their more advanced versions have been employed to model the sequential dependencies in network traffic. Stacked RNN architectures have shown strong performance in detecting botnets by learning time-dependent patterns in data[3]. In addition, Artificial Neural Networks (ANN) have been used to recognize complex non-linear relationships among traffic features, leading to better classification accuracy than traditional methods[4].

Random Forest algorithms have been popularly used to select features and enhance detection, along with Gini impurity-based weighted feature selection, have been widely used to improve intrusion detection accuracy. They do this by picking the most relevant features from high-dimensional datasets[5]. Furthermore, deep learning-based intrusion detection systems have been used in specific areas, such as agriculture IoT. These systems can effectively detect DDoS attacks by learning the particular traffic behavior of their environment. However, their effectiveness may decrease when used in different or more generalized network settings[6]. Protocol-based deep learning models that use datasets like UNSW-NB15 and Bot-IoT have performed well in detecting DoS and DDoS attacks. They do this by focusing on protocol-level information and structured traffic patterns[7].

The union of several learning structures will allow to achieve a high detection rate exploiting the variety of

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

possible model designs, but this is accompanied by increased computational complexity with greater resource requirements[8]. Deep Neural Network (DNN) technologies such as DNNBot focus on the detection and classification of botnets through the use of deep hierarchical representations, achieving high accuracy in the detection process[9]. Recent developments in intrusion detection systems include the use of economic deep learning technologies for the detection of IoT botnets, which is aimed at achieving optimal resource consumption with high computational efficiency while ensuring high accuracy in the detection process[10].

The systematic literature reviews on software-defined networking-based IoT frameworks have emphasized the need for integrating SDN with IoT to achieve flexibility, scalability, and security management, though most of these literature reviews are of a conceptual nature[11]. The software-defined networking-based IoT frameworks have been found to achieve dynamic control of traffic, network visibility, and security policy implementation, which are essential for managing large-scale IoT networks. The hybrid CNN-LSTM models have been found to achieve high performance in intrusion detection by integrating spatial feature extraction and sequence learning in a single framework, though it has been found to increase the complexity of the model, training time, and computational cost[12]. In addition, the hybrid model has been found to require efficient optimization techniques to achieve a balance between accuracy and resource consumption.

In this regard, the use of hybrid deep learning techniques that combine the capabilities of various architectures, such as the integration of CNN, RNN, and ANN, has been suggested to improve the robustness, accuracy, and generalization of the results in the process of botnet detection. The machine learning-based frameworks for the detection of DDoS attacks in software-defined IoT networks have been proposed to enhance the accuracy and flexibility of the results, but they might not be effective in dealing with the high degree of diversity associated with the various attack patterns in different network environments, especially when the training data is limited[13][14]. In addition, the frameworks might need to be constantly retrained to keep the results effective against the newly emerging threats.

Attack graph-based hybrid deep learning models offer a systematic approach for the analysis of security in cyber-physical systems by depicting possible scenarios of attacks and dependencies. This helps improve the detection of threats and decision-making capabilities. The models are helpful in identifying critical nodes and assessing the progression of attacks among connected components of a network. The use of deep learning and attack graphs helps analyze complex network behaviors and improve the detection of complex attacks. The models are also helpful in implementing a proactive approach to defending against possible attacks by predicting possible scenarios of attacks before they actually happen. This will assist in implementing security measures prior to the real attacks occurring. The models are useful too in enhancing the visualization of network security. Although these approaches have advantages, these models are specific to the application and must have a solid understanding of how the system architecture works in creating a valid attack graph. The use of these approaches is constrained as well by the fact that they are difficult to scale and update them in a dynamic environment. The graph based analysis and deep learning models are also computationally expensive, which adds to the computational cost of these approaches. In addition, researchers have embraced k-fold cross-validation so that models are stable and in order to avoid overfitting. Although these reports demonstrate large accuracy levels, it is challenging to compare the performance of various works because of the dissimilarity of preprocessing, feature selection, and evaluation measures. Hence, the development of an effective preprocessing, balanced training and ensemble based deep learning framework is a significant field of study in the realization of credible multi-class botnet detection.

Title	Problem Statement	Method	Limitations
Attack Graph Hybrid Deep Learning Model (Presekal etal., 2023)	Security analysis in cyber-physical systems	Hybrid deep learning with attack graph model	Application-specific implementation
ML Framework for DDoS Detection	Detecting DDoS attacks in software-	Machine learning detection framework	Limited attack diversity

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

in SD-IoT (Bhayo et al., 2023)	defined IoT networks		
BlocNet Hybrid IDS (Bowen et al., 2023)	Need for dataset-independent intrusion detection system	Hybrid deep learning architecture	Higher computational complexity
CNN-LSTM Hybrid Intrusion Detection (Halbouni et al., 2022)	Capturing spatial and temporal patterns in network traffic	CNN-LSTM hybrid deep learning model	Increased model complexity
SDN-based IoT Framework Survey (Siddiqui et al., 2022)	Improving IoT network security using SDN frameworks	Systematic literature review and taxonomy	Conceptual analysis without practical implementation
DNNBot (Haq & Khan, 2022)	Classification of botnet attacks in network traffic	Deep Neural Network (DNN) model	Requires large training dataset
Random Forest with GIWRF Feature Selection (Disha & Waheed, 2022)	Improving intrusion detection accuracy	Random Forest with Gini impurity weighted feature selection	Requires manual feature engineering
Protocol-Based Deep IDS (Zeeshan et al., 2021)	Detecting DoS and DDoS attacks in network traffic	Protocol-based deep learning using UNSW-NB15 and Bot-IoT datasets	Limited generalization across attack types
Deep Learning IDS for Agriculture IoT (Ferrag et al., 2021)	Detecting DDoS attacks in agriculture IoT systems	Deep learning-based intrusion detection system	Focused on specific IoT environment
Stacked Recurrent Neural Network for Botnet Detection (Popoola et al., 2021)	Botnet detection in smart home IoT networks	Stacked Recurrent Neural Network (RNN)	High computational complexity

III. METHODOLOGY

The suggested framework adheres to a multi-step pipeline that includes dataset preprocessing, training of individual deep learning models, and the ensemble learning of the stacking.

SYSTEM ARCHITECTURE

The system proposed has adopted a layered architecture to convert the raw network traffic data into meaningful predictions. Figure 1 illustrates that the proposed approach will be based on the following approach. The methodology is founded on the model stacking, in which the ANN, CNN, LSTM, RNN outputs serve to make the final prediction. The UNSW-NB15 dataset is initially fed in the preprocessing module. The processed feature set is then cleaned and encoded and sent to four separate deep learning models: ANN, CNN, LSTM, and RNN. These models are capable of learning various Network-traffic behavior, including non-linear correlations between the features of the spatial locations, and time-relations.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

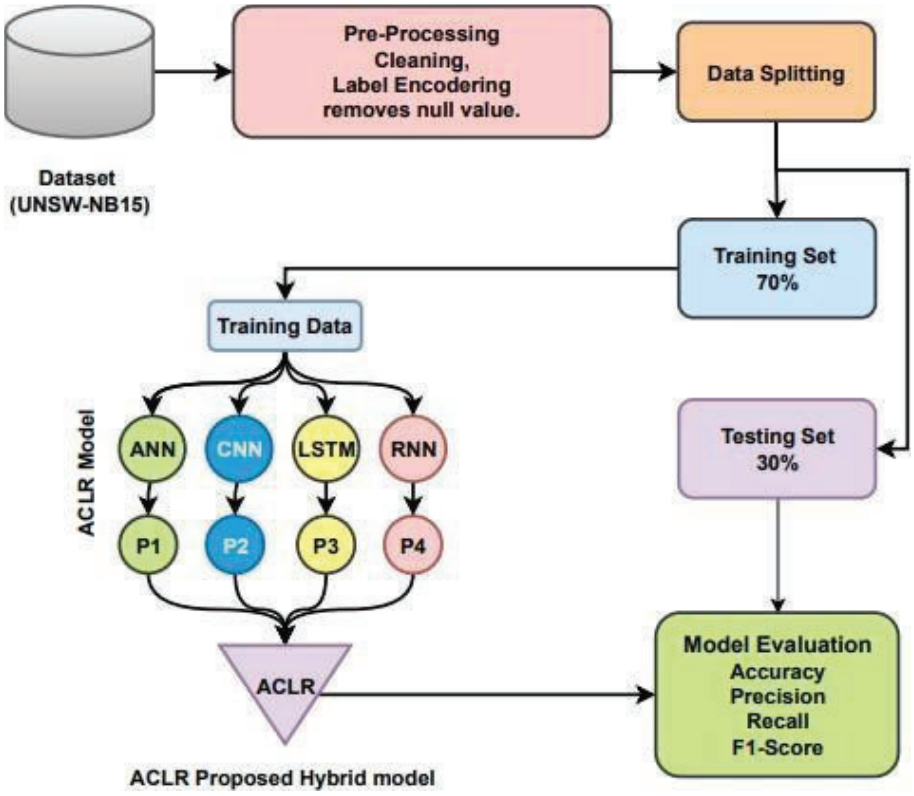


Figure 1: Architecture of the proposed approach.

3.1 Dataset Description

Experimental assessment is carried out with UNSW-NB15. It consists of 82,332 network traffic records with 45 features extracted which are flow-based attributes. This collection presents a constant proportion of multi-class classification.

Attack Types	Absolute Frequencies	Absolute Percentages
Normal	37000	0.449400
Generic	18871	0.229206
Exploits	11132	0.135209
Fuzzers	6062	0.073629
DoS	4089	0.049665
Reconnaissance	3496	0.042462
Analysis	677	0.008223
Backdoor	583	0.007081
Shellcode	378	0.004591
Worms	44	0.000534

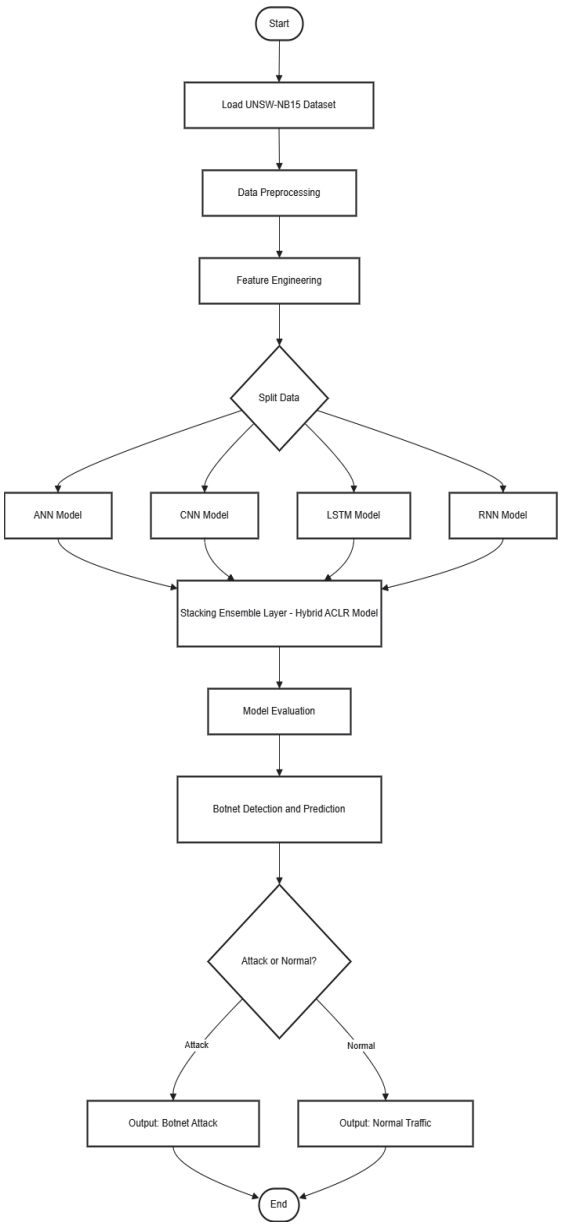
Table 1: Distribution of attack categories in the UNSW-NB15 dataset.

3.2 Data Preprocessing

Raw network datasets typically have missing data, categorical data and an uneven scales, which itself cannot be directly learned in deep learning. Preprocessing is thus extremely significant to enhance the performance of the model. First, unstable or inconsistent entries were dropped to prevent e.g. training instability. Label encoding was used to translate categorical values, like the type of protocols and types of services, into numerical values. Preprocessing implies processing missing values, categorizing label attributes, and normalizing the numerical ones. Assume that the feature vector is denoted X = {x 1, x 2, ., x n } and that the labels are Y = {y 1, y 2, ., y n).

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

WORKFLOW



The workflow diagram represents the workflow of the proposed hybrid deep learning model to detect botnet attacks using the UNSW-NB15 dataset, which includes both normal and attack traffic that have been labeled. Preprocessing of data cleans it of missing data flips categorical variables, and normalizes numerical ones to enhance quality. The meaningful patterns are extracted by feature engineering and the dataset is split into training and testing sets. ANN, CNN, LSTM, and RNN are a few of the models that are trained individually. A stacking ensemble is used to combine their outputs with the aim of producing the Hybrid ACLR model. Performance is measured using standard metrics and the system categories traffic to be either Botnet Attack or Normal Traffic.

MATHEMATICAL EQUATIONS

Measurement schemes like accuracy, precision, and recall are typically applied to measure classification performance. These measures imply that the model can be used to predict various classes or category of input data. Accuracy is sometimes used but not sufficient especially when databases are imbalanced. Recall (also called sensitivity or true positive rate) and is the percentage of predicted positive cases that are accurately recalled. It lends an idea of how effectively the model detects all the positive cases in question. The fact that recall values are higher indicates that this is an effective model to minimize false negatives. Precision also known as positive predictive value is the ratio of the predicted positive cases that are correct to the expected positive cases. It is a measure of the effectiveness of the model at identifying a positive sample at a minimum number of false positives. The high values of precision reflect that the model generates very few false positives. The calculation of the evaluation parameters is done by the following equations.

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

Where TP is true positive; that is, the number of attacks correctly predicted. False positive is referred to as FP, the normal packet is guessed to be an attack. TN is true negative indicating that there is a normal packet predicted to be a normal packet. In its turn, FN implies an attack which is predicted as a normal packet.

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

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

Another widely-used indicator for analyzing the effectiveness of the classifier system is F1 score. This measure makes it possible to evaluate the model by taking into account both its precision and recall. Here is the formula of the F1 score calculation:

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

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

The ROC-AUC metric determines the area under the curve for the Receiver Operating Characteristics. In other words, this is a metric that visualizes the performance of a binary classification algorithm. Specifically, this metric depicts the True Positive Rate against the False Positive Rate at varying threshold values. The value of ROC-AUC can vary between 0 and 1, where a high value is indicative of good performance. A value of 0.5 suggests random classification, while a value of 1 indicates perfect classification.

$$\text{ROC-AUC} = \int_0^1 \text{TPR}(\text{FPR})\, d(\text{FPR})$$

PR-AUC is an acronym for the area under the Precision-Recall curve. It assesses the performance of a model on a binary classification task by analyzing how accurately it predicts the positive class. The PR-AUC ranges from 0 to 1, where higher values show better model performance. A value of 1 indicates perfect precision and recall, while a value of 0 indicates poor performance.

$$PR\text{-}AUC = \int_0^1 Precision(Recall)\, d(Recall)$$

The integral here defines the area underneath the curve. The values of Precision and Recall depend on the threshold used for the predicted values.

SAMPLE CALCULATIONS

To show how these calculations are done, an example calculation is given using the values of the confusion matrix generated from the ANN model.

From the confusion matrix:

TP = 8677

TN = 7029

FP = 389

FN = 372

The total number of samples is computed as follows:

Total Samples = TP + TN + FP + FN

= 8677 + 7029 + 389 + 372

= 16467

The number of correctly classified samples is:

Correct Predictions = TP + TN

= 8677 + 7029

= 15706

Thus, the accuracy can be calculated as:

Accuracy = 15706 / 16467

= 0.9538 (95.38%)

Likewise, the precision and recall are computed as:

Precision = 8677 / (8677 + 389) = 0.9534

Recall = 8677 / (8677 + 372) = 0.9532

On the basis of these figures, the F1 score can be calculated as:

F1 Score = (2 × 0.9534 × 0.9532) / (0.9534 + 0.9532)

= 0.9533

ALGORITHMS

1) Artificial Neural Networks

ANNs play a very important role in the field of deep learning and work on the principles based on human brain structure and functioning. ANNs have gained recognition because of their ability to discover complicated connections and patterns in data. This makes ANNs useful in many different applications, including time series analysis, image recognition, and natural language processing. ANNs have successfully been applied in many fields, including financial sectors, healthcare, and self-driving vehicles. However, one of the limitations associated with ANNs is that they are often computational-intensive and require large amounts of training datasets. Yet, significant progress in hardware and algorithms development helped ANN become an integral part of modern machine learning. Table 2 highlights some of the hyperparameters in ANNs.

Hyperparameter	Values
Number of Layers	3
Layer 1 Dense	64
Layer 2 Dense	32
Layer 3 Dense	1
Activation function	relu
Activation function	sigmoid
Optimizer	adam
loss	binary_crossentropy
metrics	accuracy
batch_size	32
Number of Epochs	30

Table 2: Hyperparameter of ANN model.

2) Convolutional Neural Network

A sophisticated version of an ANN is CNN. This type of neural network was designed specifically to process visual data like images and videos efficiently. The main advantage of CNNs is that they are good at identifying patterns and features in images. The input data goes

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

through several layers of filtering within the convolutional layers. In doing so, CNNs become capable of capturing local patterns and relationships. The feature map then shrinks due to the pooling layers; therefore, computational complexity decreases. Finally, the fully connected layers help extract more complex data and create the network's final output.image classification

Hyperparameter	Values
Number of Convolutional Layers	2
Number of Filters (Layer 1)	64
Number of Filters (Layer 2)	32
Activation function	relu
Activation function	sigmoid
Kernel Size	(3,3)
Pooling	MaxPooling (2x2)
Optimizer	adam
loss	binary_crossentropy
metrics	accuracy
batch_size	32
Number of Epochs	30

Table 3: Hyperparameter of CNN model.

3) Long Short-Term Memory Network

LSTM is one of the variants of RNN used for modeling sequential data. Unlike normal RNNs, LSTM networks can remember data for long periods and retain long-term dependencies. The memory units within LSTM models are responsible for storing the required information and updating the same. The input, forget, and output gates decide which information should be retained and which should be forgotten. There are three gates present in LSTM models that regulate the flow of information within the LSTM units: the input gate, the forget gate, and the output gate.

Hyperparameter	Values
Number of Layers	2
LSTM (Layer 1)	64
Dense (Layer 2)	32
Activation function	sigmoid
Optimizer	adam
loss	binary_crossentropy
metrics	accuracy
batch_size	32
Number of Epochs	30

Table 4: Hyperparameter of LSTM model.

4) Recurrent Neural Network

Recurrence relations make RNN useful for various tasks such as natural language processing, speech recognition, and handwriting recognition. However, it is good at modeling short-range dependencies, while its performance on long-range dependencies may be poor due to vanishing gradient problems. Vanishing gradient refers to a situation where the gradients diminish with time. This may limit the ability of traditional RNNs to detect long-range dependencies.

IV. RESULT ANALYSIS

The deep learning techniques of ANN, CNN, LSTM, and RNN were used for experimentation because of the ability of the mentioned algorithms to recognize the non-linear, spatial, and temporal nature of the network traffic. The allocation of datasets involved the division of 70% data for training purposes while the rest of 30% was kept aside for testing purpose. The additional tests were conducted by random sampling of data. Accuracy, precision, recall, and F1-score were considered the key parameters for evaluating the performance of the proposed models. Figure 3 shows the graphical user interface of the hybrid technique that may be helpful for botnet detection on IoT.

Figure 3: Home screen

As illustrated in Figure 4 below, the performances of ANN, CNN, LSTM, RNN, and Hybrid ACLR models are evaluated using accuracy, precision, recall, and F1-score metrics. Both ANN and CNN showed equal accuracy, which was 95.38% and 95.35%, respectively. On the other hand, RNN model performed at 95.12% accuracy, whereas LSTM model obtained 95.44% accuracy. Nonetheless, the Hybrid ACLR model showed outstanding performance with 96.42% accuracy.

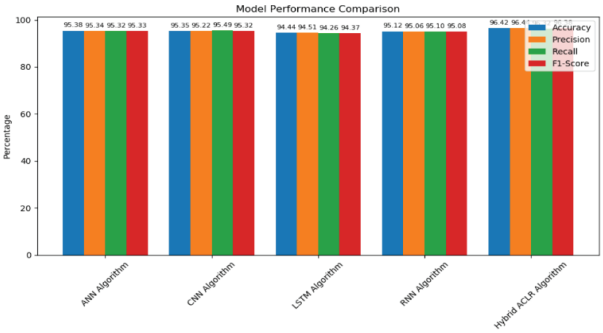


Figure 4: Performance analysis of deep learning models

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

The ROC and PR curves for the developed hybrid ACLR model are shown in Figure 5. The ROC curve remains very close to the top left corner of the diagram. This implies that the model has a very high True Positive rate with a very low False Positive rate irrespective of the threshold level. It attained an AUC of 0.9951 for the ROC curve, signifying good separability. The PR curve is also very close to the upper edge with a PR-AUC of 0.9963.

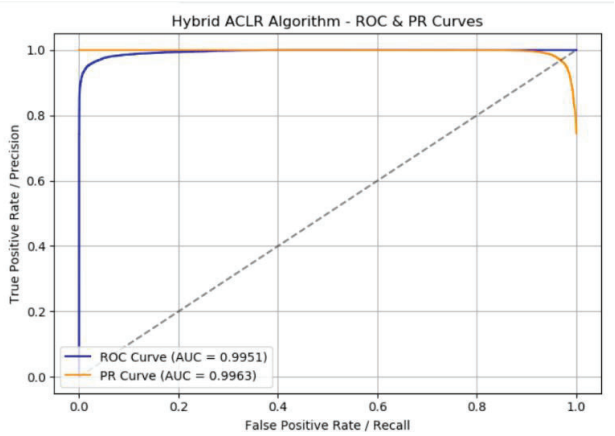


Figure 5: Performance regarding ROC-PR curve

Table 5 compares models based on accuracy, precision, recall, and F1-score. The Hybrid ACLR model outperforms all standalone models. Among individual models, LSTM achieved the highest accuracy at 0.9544, followed closely by ANN at 0.9538, CNN at 0.9535, and RNN at 0.9512. All these models exceeded 94%. The Hybrid ACLR reached an accuracy of 0.9642, showing better generalization and prediction. Improved precision, recall, and F1-score confirm that stacking effectively balances false positives and false negatives by using the strengths of several deep learning models.

Models	Accuray	Precision	Recall	F1Score
ANN	0.9538	0.9534	0.9532	0.9533
CNN	0.9535	0.9522	0.9549	0.9532
LSTM	0.9544	0.9451	0.9426	0.9437
RNN	0.9512	0.9506	0.9510	0.9508
ACLR	0.9642	0.9644	0.9632	0.9638

Table 5: Analysis of all employed models.

V. DISCUSSION

This research introduces ACLR, a hybrid deep learning model designed for detecting botnet attacks in IoT networks. The proposed architecture integrates four types of architecture, ANN, CNN, LSTM, and RNN, through a stacked ensemble method. Using the combination of the best attributes of spatial features extraction, nonlinear mapping, and sequential learning, the ACLR model is able to develop a robust hybrid detection mechanism that enhances the accuracy of classifications. The accuracy of the model was measured using a training and testing set of 70:30. ACLR gave an impressive accuracy result of 0.9642 compared to each model, which is superior to all other models. ANN gave 0.9538, CNN gave 0.9535, RNN gave 0.9512, and LSTM gave 0.9544. The model's performance is validated further through the use of metrics such as the ROC-AUC score and PR-AUC score. The ACLR model has a ROC-AUC score of 0.9951 and a PR-AUC score of 0.9963. The performance demonstrated by these AUC scores signifies a high level of separability among classes despite possible class imbalance in the data.

VI. CONCLUSION

The rise in frequency and complexity of cyber attacks has made detecting botnets a key issue in IoT networks. Deep learning models have shown they can effectively identify malicious traffic, but using ensemble methods often leads to more stable and accurate predictions than individual models. In this research, we proposed a hybrid stacking model that combines ANN, CNN, LSTM, and RNN, called ACLR, to improve botnet detection performance. Alongside this hybrid model, we also implemented individual models of ANN, CNN, LSTM, and RNN to ensure a fair comparison. The experimental results show that the Hybrid ACLR model achieved the highest accuracy of 0.9642, surpassing ANN (0.9538), CNN (0.9535), RNN (0.9512), and LSTM (0.9544). This improvement confirms that stacking multiple deep learning architectures boosts generalization by using different feature extraction methods. Additionally, the proposed model reached a ROC-AUC value of 0.9951 and a PR-AUC value of 0.9963, indicating excellent class separability and strength under conditions of potential class imbalance. The evaluation metrics, including accuracy, precision, recall, and F1-score, collectively show how effective the hybrid approach is at reducing false positives and false negatives.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

We noticed that increasing the model's depth and complexity can enhance predictive performance; however, it also leads to higher computational costs and longer training times. While individual models like ANN, CNN, LSTM, and RNN achieve good results, the stacked ACLR model consistently outperforms them by combining spatial and temporal learning methods. Still, there remain several difficulties in employing deep learning for botnet identification systems, which include the requirement for labeled data, the danger of being subject to an adversarial attack, and challenges with implementing the model in the real-world Internet of Things (IoT). Possible future studies may focus on scalability, computational efficiency, or reinforcement learning techniques that would further help in developing efficient automated threat detectors.

VII. FUTURE WORK

This hybrid approach has exhibited good performance, however it still needs to be further developed in order to be used on the internet of things platforms. Scalability, efficiency, and flexibility of this

The following improvements can be made in future

- Test the performance of the model based on real time network traffic by assessing its scalability, latency and dynamic adaptation capabilities.
- Optimize the model using techniques such as pruning, compression and knowledge distillation to reduce computational overhead.
- Use adaptive learning techniques to tackle changes in network environment and attacks.
- Apply transfer learning to improve accuracy with less data in different IoT settings.
- Utilize reinforcement learning for continuous learning and intelligent decision making.

REFERENCES

[1] A. Presekal, A. Stefanov, V. S. Rajkumar, and P. Palensky, “Attack graph model for cyber-physical power systems using hybrid deep learning,” IEEE Transactions on Smart Grid, 2023.

[2] J. Bhayo, S. A. Shah, S. Hameed, A. Ahmed, J. Nasir, and D. Draheim, “Towards a machine learning-based framework for ddos attack detection in software-defined iot (sd-iot) networks,” Engineering Applications of Artificial Intelligence, vol. 123, p. 106432, 2023.

[3] B. Bowen, A. Chennamaneni, A. Goulart, and D. Lin, “Blocnet: a hybrid, dataset-independent intrusion detection system using deep learning,” International Journal of Information Security, pp. 1–25, 2023.

[4] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, “Cnn-lstm: hybrid deep neural network for network intrusion detection system,” IEEE Access, vol. 10, pp. 99837–99849, 2022.

[5] S. Siddiqui, S. Hameed, S. A. Shah, I. Ahmad, A. Aneiba, D. Draheim, and S. Dustdar, “Towards software-defined networking-based iot frameworks: A systematic literature review, taxonomy, open challenges and prospects,” IEEE Access, 2022.

[6] M. A. Haq and M. A. Rahim Khan, “DNNBot: Deep neural network-based botnet detection and classification,” Computers, Materials & Continua, vol. 71, no. 1, pp. 1–15, 2022.

[7] R. A. Disha and S. Waheed, “Performance analysis of machine learning models for intrusion detection system using gini impurity-based weighted random forest (giwrf) feature selection technique,” Cybersecurity, vol. 5, no. 1, p. 1, 2022.

[8] M. Zeeshan, Q. Riaz, M. A. Bilal, M. K. Shahzad, H. Jabeen, S. A. Haider, and A. Rahim, “Protocol-based deep intrusion detection for dos and ddos attacks using unsw-nb15 and bot-iot datasets,” IEEE Access, vol. 10, pp. 2269–2283, 2021.

[9] M. A. Ferrag, L. Shu, H. Djallel, and K.-K. R. Choo, “Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0,” Electronics, vol. 10, no. 11, p. 1257, 2021.

[10] S. I. Popoola, B. Adebisi, M. Hammoudeh, H. Gacanin, and G. Gui, “Stacked recurrent neural network for botnet detection in smart homes,” Computers & Electrical Engineering, vol. 92, p. 107039, 2021.

[11] M. Azizjon, A. Jumabek, and W. Kim, “1d cnn based network intrusion detection with normalization on imbalanced data,” in 2020 international conference on artificial intelligence in information and communication (ICAIIC), pp. 218–224, IEEE, 2020.

[12] I. H. Sarker, “Deep cybersecurity: a comprehensive overview from neural network and deep learning perspective,” SN Computer Science, vol. 2, no. 3, p. 154, 2021.

[13] A. A. Ahmed, W. A. Jabbar, A. S. Sadiq, and H. Patel, “Deep learning-based classification model for botnet attack detection,” Journal of Ambient Intelligence and Humanized Computing, vol. 11, no. 1, pp. 1–14, 2020.

[14] N. Elsayed, Z. ElSayed, and M. Bayoumi, “IoT botnet detection using an economic deep learning model,” arXiv preprint, 2023.

[15] M. Y. Alzahrani and A. M. Bamhdi, “Hybrid deep-learning model to detect botnet attacks over internet of things environments,” Soft Computing, vol. 26, no. 16, pp. 7721–7735, 2022.