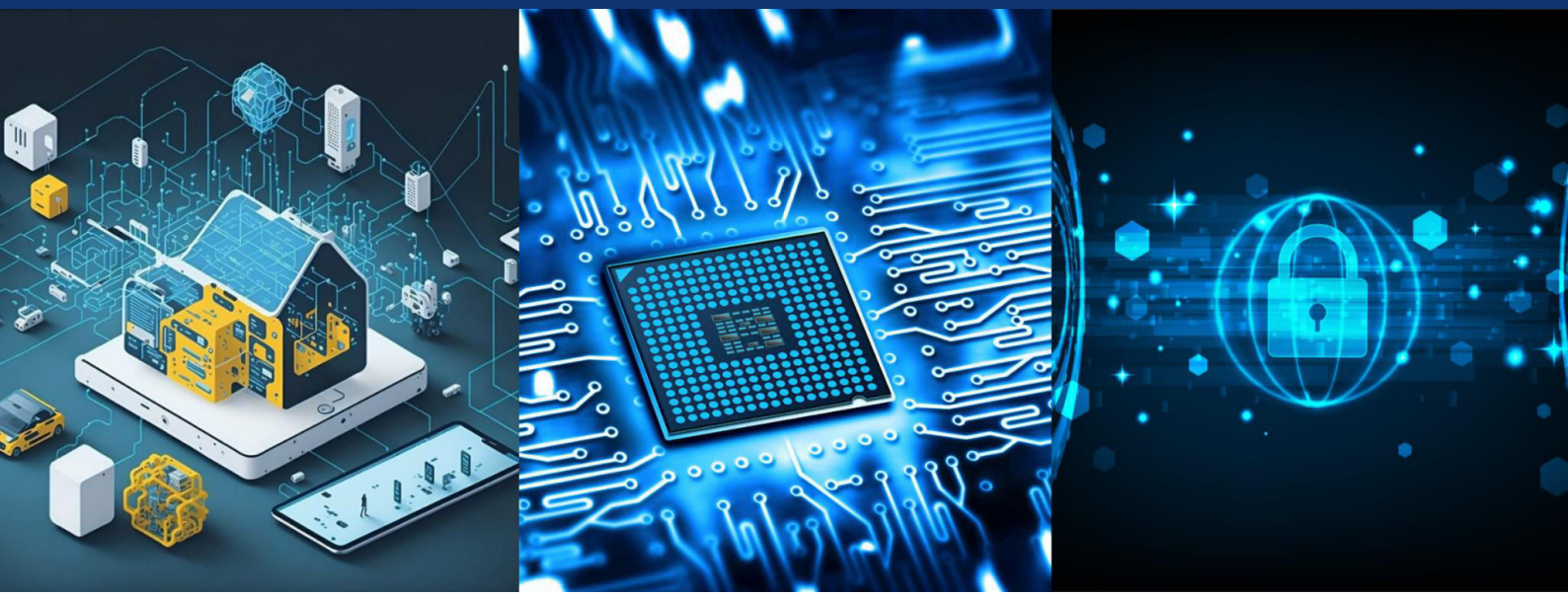
IJIRCCE

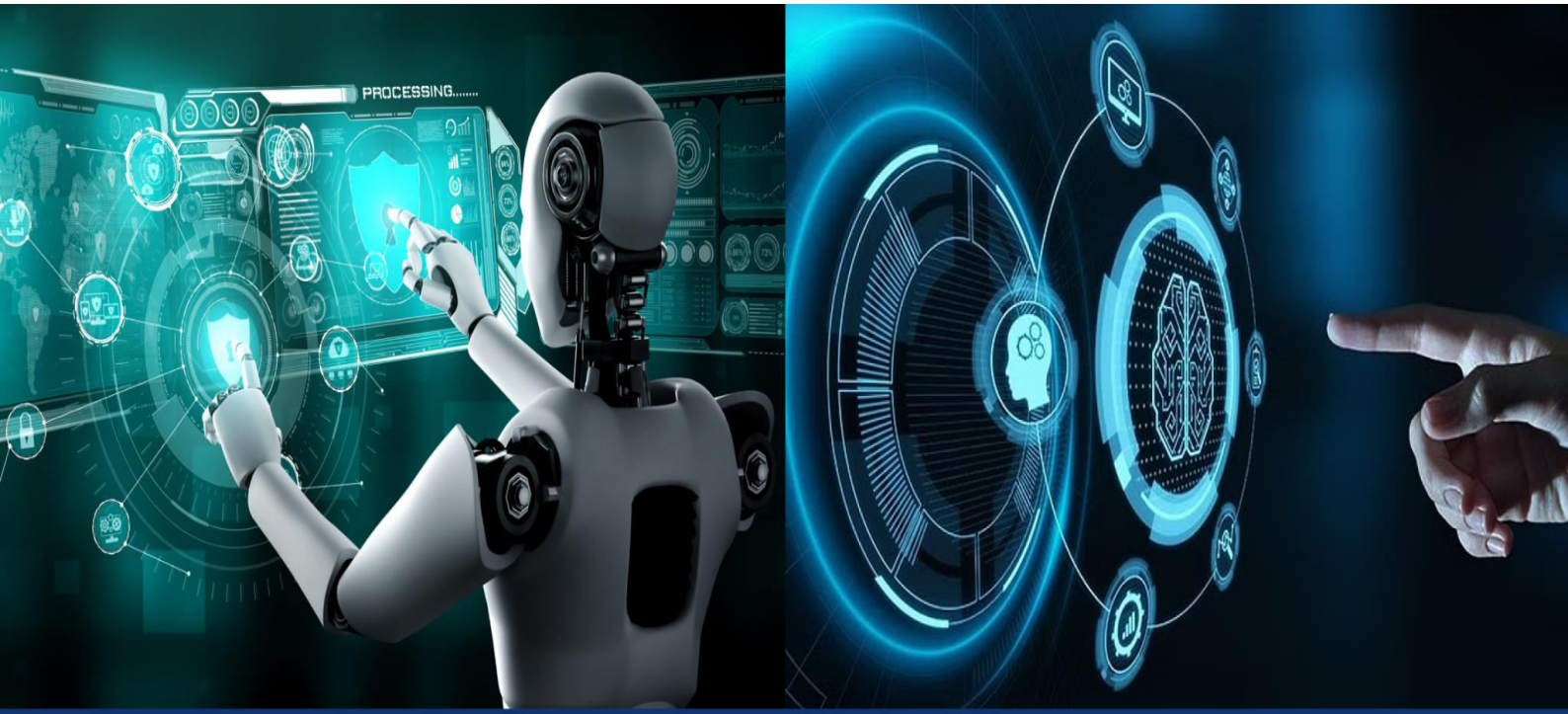
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.927

Volume 14, Issue 8, August 2026

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408016

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

An Efficient Deep Learning based Prediction Model for Botnet Attack Detection

Suraj Kumar, Dr. Dayashankar Pandey

M. Tech Scholar, Department of CSE, Sarvepalli Radhakrishnan University, Bhopal, India

Professor, Department of CSE, Sarvepalli Radhakrishnan University, Bhopal, India

ABSTRACT: Botnet attacks are a major threat to computer networks, allowing hackers to control many infected devices at once to launch attacks like DDoS, steal data, and spread spam. Older detection methods, such as signature-based systems and basic machine learning models, often fail to detect new and hidden botnet attacks because these attacks keep changing their patterns. To solve this problem, this paper presents an efficient deep learning model using Convolutional Neural Networks (CNN) to detect botnet attacks accurately. The proposed method first cleans and prepares the network traffic data, then extracts important features that help identify unusual or harmful traffic. The CNN model is then used to automatically learn patterns from this data and classify network traffic as either normal or botnet-related. The model is tested on a standard botnet dataset, and the results show that it achieves high accuracy, precision, recall, and F-measure, along with a low error rate. This proves that the proposed CNN-based model is an effective and reliable solution for detecting botnet attacks in real-world network systems.

KEYWORDS: IOT, CNN, Cyber, Botnet, Attack, Security.

I. INTRODUCTION

The rapid growth of internet-connected devices and the increasing reliance on digital networks have made cybersecurity one of the most pressing challenges of the modern era [1]. Among the various forms of cyberattacks, botnet attacks have emerged as one of the most dangerous and widespread threats, capable of compromising thousands or even millions of devices simultaneously to carry out coordinated malicious activities [2]. A botnet is essentially a network of infected devices, often referred to as "bots" or "zombies," that are remotely controlled by an attacker through a command-and-control (C&C) server without the knowledge of the device owners [3]. Once formed, botnets can be used to launch large-scale attacks such as Distributed Denial-of-Service (DDoS) attacks, spam campaigns, data theft, click fraud, and unauthorized cryptocurrency mining, causing severe financial and operational damage to individuals, organizations, and critical infrastructure [4].

The growing sophistication of botnet architectures has made their detection increasingly difficult. Modern botnets often use encrypted communication channels, peer-to-peer (P2P) structures instead of centralized servers, and dynamic domain generation algorithms (DGAs) to hide their C&C traffic and avoid detection by traditional security systems [5]. Furthermore, the sheer diversity of devices connected to the internet, including Internet of Things (IoT) devices with limited security protections, has significantly expanded the attack surface available to botnet operators, making it easier for attackers to recruit new bots and harder for defenders to secure every endpoint [6]. As a result, botnets have evolved from simple, easily detectable networks into highly adaptive and resilient threats capable of evading conventional defense mechanisms.

Traditional intrusion detection systems (IDS) have primarily relied on signature-based detection, where known attack patterns are stored in a database and compared against incoming network traffic [7]. While effective against previously identified threats, this approach is fundamentally limited when facing new or modified botnet variants, as it cannot detect attacks whose signatures are not already known [8]. Additionally, signature-based systems require frequent manual updates to remain effective, making them poorly suited to the fast-changing nature of botnet behavior. These limitations have pushed researchers toward more intelligent and adaptive detection techniques capable of identifying previously unseen attack patterns.

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408016

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Machine learning techniques have been widely explored as an alternative to traditional detection methods, offering the ability to learn patterns directly from network traffic data and generalize to new, unseen threats [9]. By analyzing features such as packet size, connection duration, traffic volume, and communication frequency, machine learning models can classify network flows as either normal or malicious with reasonably good accuracy [10]. However, many conventional machine learning approaches depend heavily on manual feature engineering, which requires domain expertise and may fail to capture the complex, hidden relationships present in large-scale network traffic data.

Deep learning, a subset of machine learning based on artificial neural networks, has gained significant attention in recent years due to its ability to automatically learn hierarchical feature representations directly from raw data, eliminating the need for manual feature extraction [11]. Among various deep learning architectures, Convolutional Neural Networks (CNNs) have shown particularly strong performance in identifying spatial and structural patterns within data, making them well-suited for analyzing network traffic represented in structured or image-like formats [12]. By applying convolutional filters across traffic features, CNNs can capture local patterns and correlations that are often indicative of botnet communication behavior, even when such patterns are subtle or partially hidden within normal-looking traffic.

II. METHODOLOGY

The methodology and work flow shows as per following flow chart-

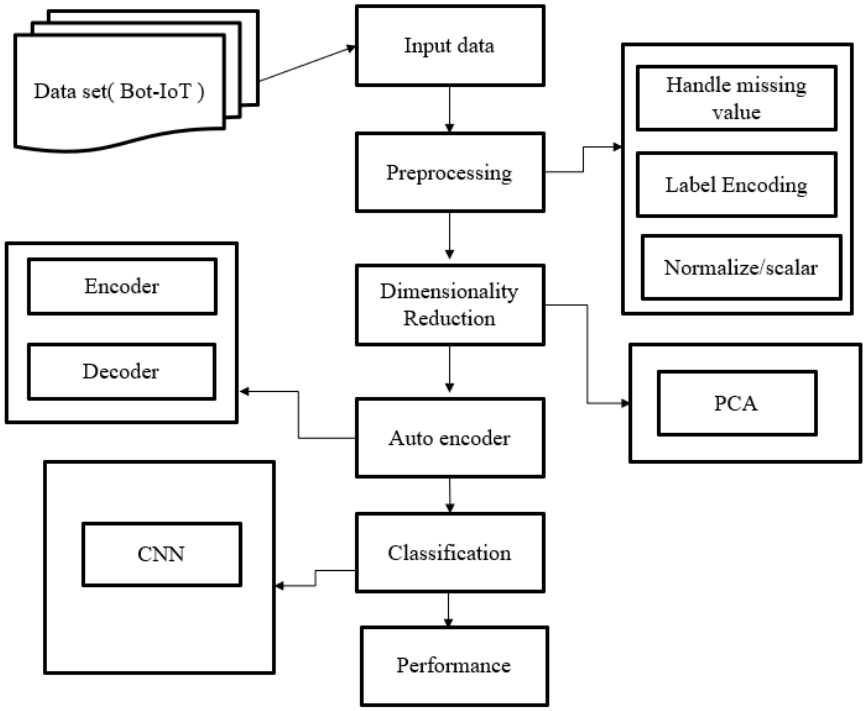


Figure 1: Flow chart

The proposed botnet attack detection framework follows a structured pipeline consisting of data acquisition, preprocessing, dimensionality reduction, autoencoder-based feature learning, CNN-based classification, and performance evaluation, as depicted in the flowchart. Each stage of this pipeline plays a critical role in transforming raw network traffic data into a refined representation suitable for accurate and efficient botnet detection. The following sections describe each stage in detail, along with the relevant mathematical formulations.

1. Dataset (Bot-IoT) and Input Data

The framework begins with the Bot-IoT dataset, a widely used benchmark dataset containing realistic network traffic captured from an IoT environment, comprising both normal traffic and various categories of botnet attacks, including

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408016

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

DDoS, DoS, reconnaissance, and information theft. Each record in the dataset is represented as a feature vector $x_i \in \mathbb{R}^n$, where $n$denotes the number of extracted traffic attributes such as packet size, protocol type, flow duration, and byte count, along with a corresponding class label $y_i$indicating whether the traffic is benign or malicious. This raw data is fed into the pipeline as the initial input for further processing.

2. Preprocessing

Raw network traffic data is often incomplete, inconsistent, or expressed in non-numeric formats, making preprocessing an essential step before model training. This stage consists of three sub-processes:

Handling Missing Values: Missing or corrupted entries in the dataset are addressed through imputation techniques, where missing values are typically replaced with the mean, median, or mode of the corresponding feature:

$$x_{ij}^{imputed} = \frac{1}{k}\sum_{p=1}^{k} x_{pj},\ x_{pj} \neq \text{NaN}$$

where $x_{ij}$is the missing value in the j-th feature of the i-th sample, and $k$is the number of available (non-missing) values for that feature.

Label Encoding: Since many traffic features, such as protocol type or connection state, are categorical in nature, they must be converted into numerical form for the model to process. Label encoding assigns a unique integer to each category:

$$x_{ij}^{encoded} = LE(x_{ij}),\ LE: C \rightarrow \{0, 1, 2, \ldots, c-1\}$$

where Crepresents the set of categorical values and $c$is the total number of distinct categories.

Normalization/Scaling: To ensure that all features contribute equally during model training and to prevent features with larger numeric ranges from dominating the learning process, min-max normalization is applied:

$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

This scales all feature values into the range $[0, 1]$, improving both convergence speed and model stability during training.

3. Dimensionality Reduction using PCA

Following preprocessing, Principal Component Analysis (PCA) is applied to reduce the high dimensionality of the traffic feature space while retaining the maximum possible variance. Given the standardized data matrix $X \in \mathbb{R}^{m \times n}$, the covariance matrix is first computed as:

$$C = \frac{1}{m-1} X^T X$$

The eigenvectors $v_k$and corresponding eigenvalues $\lambda_k$of $C$are then obtained by solving:

$$C v_k = \lambda_k v_k$$

The top $k$eigenvectors, corresponding to the largest eigenvalues, are selected to construct the projection matrix $W \in \mathbb{R}^{n \times k}$, and the reduced feature representation is computed as:

$$X_{PCA} = XW$$

This step reduces computational overhead and eliminates redundant or correlated features, allowing subsequent stages to focus on the most informative traffic characteristics.

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408016

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

4. Autoencoder (Encoder–Decoder)

After dimensionality reduction, the reduced feature set is passed through an autoencoder, an unsupervised neural network designed to learn compact and meaningful feature representations. The autoencoder consists of two components:

Encoder: The encoder compresses the input data X_{PCA} into a lower-dimensional latent representation z:

$$z = \sigma(W_e X_{PCA} + b_e)$$

where W_e and b_e are the weight matrix and bias of the encoder, and $\sigma(\cdot)$ is a non-linear activation function such as ReLU or sigmoid.

Decoder: The decoder attempts to reconstruct the original input from the latent representation:

$$\hat{X} = \sigma(W_d z + b_d)$$

where W_d and b_d are the decoder's weight matrix and bias. The autoencoder is trained to minimize the reconstruction error between the original input and its reconstructed version, typically using mean squared error:

5. Classification using CNN

The latent features extracted by the autoencoder are then fed into a Convolutional Neural Network (CNN) for final classification. The CNN applies convolutional filters to detect local patterns and correlations within the feature representation:

$$f_l = \sigma\left(\sum_i W_c * z_i + b_c\right)$$

where W_c represents the convolutional kernel weights, $*$ denotes the convolution operation, b_c is the bias term, and $\sigma(\cdot)$ is typically a ReLU activation function, $\sigma(x) = \max(0, x)$. Pooling layers are subsequently applied to reduce spatial dimensionality while retaining dominant features, followed by fully connected layers that map the learned features to the final output classes. The output layer uses a softmax or sigmoid activation function to produce the predicted class probability:

$$\hat{y} = \sigma\left(\sum_j W_o f_l + b_o\right)$$

which classifies the input traffic as either normal or belonging to a specific botnet attack category.

6. Performance Evaluation

In the final stage, the predictions generated by the CNN classifier are compared against the ground-truth labels to evaluate the model's effectiveness. Standard performance metrics, including accuracy, precision, recall, and F-measure, are computed to assess the overall detection capability of the proposed model. These results provide a quantitative basis for comparing the proposed autoencoder-CNN hybrid framework against existing botnet detection approaches, validating its effectiveness in accurately identifying malicious traffic within IoT network environments.

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408016

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

III. SIMULATION AND RESULTS

The simulation of the proposed methodology is done over python spyder 3.7.

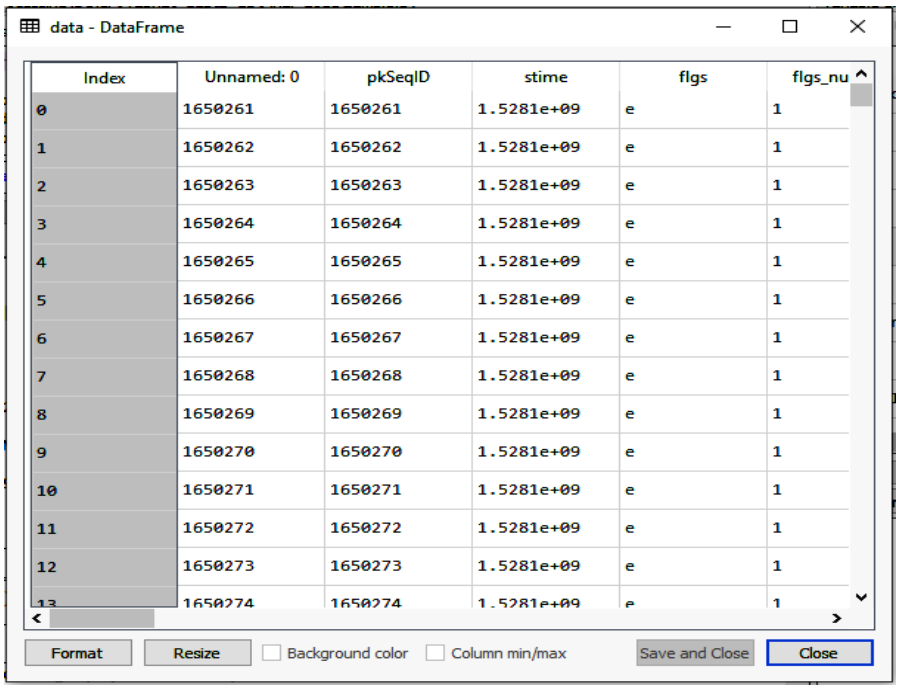


Figure 2: Dataset

Figure 2 is showing the dataset in the python environment. The dataset has various numbers of rows and column. The features name is mention in each column.

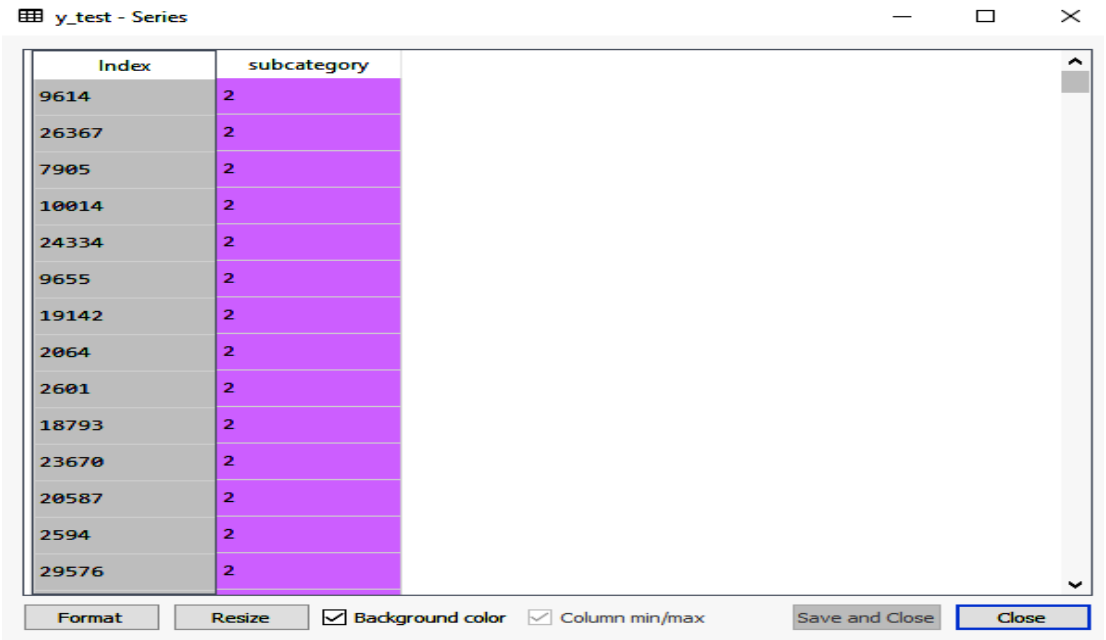


Figure 3: Y test

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408016

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Figure 3 is showing the y test of the given dataset. The given dataset is divided into the 20-30% part into the train dataset.

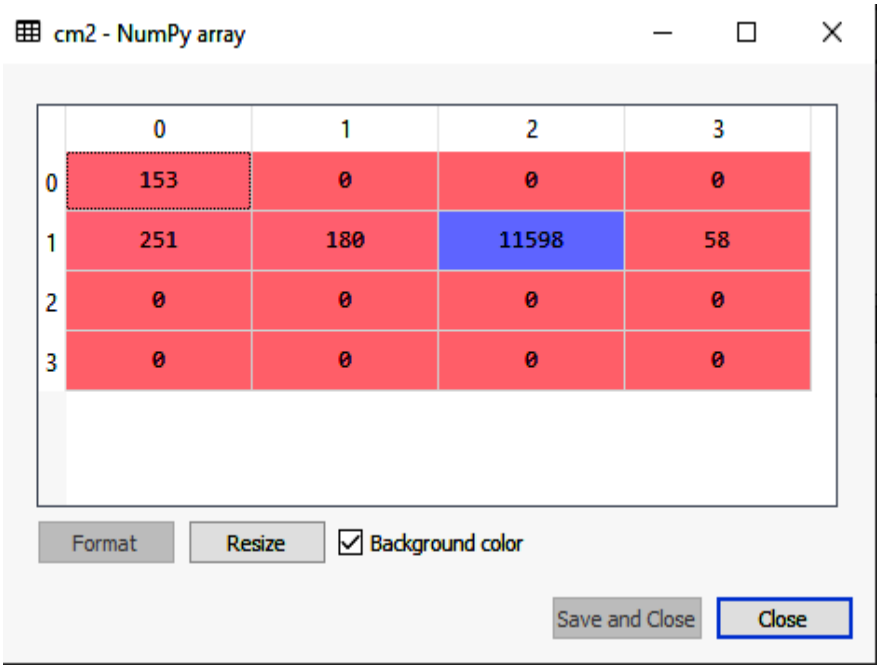


Figure 4: Confusion matrix of CNN

Figure 4 is presenting the confusion matrix of CNN for the given dataset after the training and the testing. It is matrix to identify the prediction of the given dataset.

Table 1: Result Comparison

Sr. No.	Parameters	Previous Work [1]	Proposed Work
1	Method	ACLR	CNN
2	Precision	97 %	99.99 %
3	F_Measure	96 %	99 %
4	Accuracy	97.49 %	99.99 %
5	Error Rate	2.51%	0.01%

As shown in the table 1, the proposed CNN-based model achieves a precision of 99.99%, a substantial improvement over the ACLR method's 97%, indicating a near elimination of false positives in botnet traffic classification. Similarly, the F-measure improves from 96% to 99%, reflecting a more balanced and consistent trade-off between precision and recall in the proposed approach. In terms of overall accuracy, the proposed model attains 99.99%, outperforming the previous work's 97.49% by a margin of 2.50%, demonstrating its superior ability to correctly distinguish between normal and botnet-infected traffic.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IV. CONCLUSION

This paper presented an efficient deep learning-based model for botnet attack detection, addressing the growing threat posed by increasingly sophisticated and evasive botnet architectures that traditional signature-based and shallow machine learning methods struggle to detect. The proposed methodology followed a systematic pipeline involving preprocessing of raw network traffic through missing value handling, label encoding, and normalization, followed by dimensionality reduction using PCA and feature refinement through an autoencoder, before final classification using a Convolutional Neural Network (CNN). This combination allowed the model to automatically learn discriminative and noise-reduced feature representations directly from IoT network traffic, eliminating the need for extensive manual feature engineering. Experimental results demonstrated that the proposed CNN-based model significantly outperformed the existing ACLR approach, achieving a precision of 99.99%, F-measure of 99%, and accuracy of 99.99%, while reducing the error rate to just 0.01%. These results confirm that integrating PCA-based dimensionality reduction with autoencoder-driven feature learning and CNN classification substantially enhances detection performance and reliability. Overall, the proposed framework offers a robust and highly accurate solution for real-time botnet attack detection in IoT environments, with future work aimed at validating the model on larger and more diverse datasets and extending it to detect emerging botnet variants.

REFERENCES

1. M. Ali, M. Shahroz, M. F. Mushtaq, S. Alfarhood, M. Safran and I. Ashraf, "Hybrid Machine Learning Model for Efficient Botnet Attack Detection in IoT Environment," in IEEE Access, vol. 12, pp. 40682-40699, 2024, doi: 10.1109/ACCESS.2024.3376400.
2. A. K. Kumar et al., "Enhanced Hybrid Deep Learning Approach for Botnet Attacks Detection in IoT Environment," 2024 7th International Conference on Signal Processing and Information Security (ICSPIS), Dubai, United Arab Emirates, 2024, pp. 1-6, doi: 10.1109/ICSPIS63676.2024.10812621.
3. M. Alshehri J. Ahmad, S. Almakdi, M. Qathrady, Y. Ghadi and w. Buchanan, “SkipGateNet: A Lightweight CNN-LSTM hybrid model with learnable skip connections for efficient botnet attack detection in IoT,” IEEE Access, vol. 12, pp. 35521–35538, March 2024 https://doi.org/10.1109/access 3371992.
4. M. Al-Fawa’reh, J. Abu-Khalaf, P. Szewczyk, and J.J Kang, MalbotDRL: Malware botnet detection using deep reinforcement learning in IOT Networks. IEEE Internet of Things Journal, vol. 11, no. 6, pp. 9610–9629, October 2023, https://doi.org/10.1109/jiot.2023.3324053
5. R. Kalakoti, H. Bahsi, and S. No mm (2024). Improving IOT security with explainable AI: Quantitative evaluation of explainability for IOT botnet detection. IEEE Internet of Things Journal, vol. 11, no. 10, pp. 18237–18254. January 2024, https://doi.org/10.1109/jiot.2024.3360626
6. X. Yan, X. Yu, S. Yao, and Y. Sun (2024). A domain embedding model for botnet detection based on Smart Blockchain. IEEE Internet of Things Journal, vol. 11, no. 5, pp. 8005–8018, February 2024, https://doi.org/10.1109/jiot.2023.3320046
7. S. Saravanan, and U.M. Balasubramanian, “An adaptive scalable data pipeline for multiclass attack classification in large-scale IOT Networks ”. Big Data Mining and Analytics, vol. 7, no. 2, pp. 500–511, April 2024, https://doi.org/10.26599/bdma.2023.9020027
8. P.V. Dinh, Q.U. Nguyen, D.T Hoang, D.N. Nguyen, S.P. Bao and E. Dutkiewicz, “Constrained twin variational auto-encoder for intrusion detection in IOT Systems ”. IEEE Internet of Things Journal, vol. 11, no. 8, pp. 14789–14803,2024, https://doi.org/10.1109/jiot.2023.3344842
9. J. Kabdjou, and N. Shinomiya, “Improving quality of service and HTTPS DDoS detection in MEC environment with a cyber-deception based architecture,” IEEE Access, vol. 12, pp. 23490–23503,2024, https://doi.org/10.1109/access.2024.3361476
10. A.A. Mohammed, and A.A. Ibrahim, “Malware detection in Adhoc EGovernment Network using machine learning,” 5 th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 20 24, https://doi.org/10.1109/hora58378.2023.10156724
11. S. I. Popoola, B. Adebisi, M. Hammoudeh, G. Gui and H. Gacanin, "Hybrid Deep Learning for Botnet Attack Detection in the Internet-of-Things Networks," in IEEE Internet of Things Journal, vol. 8, no. 6, pp. 4944-4956, 15 March15, 2021, doi: 10.1109/JIOT.2020.3034156.
12. S. I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh and O. Jogunola, "Federated Deep Learning for Zero-Day Botnet Attack Detection in IoT Edge Devices," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2021.3100755.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details