AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.6, NO. 2(2026)

A Novel Hybrid Deep Learning Model for Botnet Attacks Detection in a Secure IoMT Environment

Ms.K.Sai Durga, MCA-Assistant Professor, Department of MCA, Bapatla Engineering College, Bapatla, Andhra Pradesh

Mrs. Bethalam Jyothika (Reg No: Y25MC23011), Ms.Myla Mohana Valli (Reg No: Y25MC23052)
Mr. Chilakalapudi. Lalin Kumar (Reg No: Y25MC23016), Mr. Korrapati Manikanta (Reg No: Y25MC23037)
Department of MCA, Bapatla Engineering College, Bapatla, Andhra Pradesh.

***Abstract*—The Internet of Medical Things is growing fast. This has made it easier to monitor peoples health from away and has improved medical services.. It also means that medical devices are connected to the internet, which makes them vulnerable to cyber attacks. These attacks can compromise the safety of information and the whole system.**

To deal with this problem we are suggesting a way to detect these kinds of attacks using a special kind of computer program. This program uses two methods to look at the data from medical devices. The first method looks at the data like a picture. The second method looks at the data over time.

We used data from medical devices to test our program. We also did some things to the data to make it easier for the program to understand.

When we tested our program it worked well. It was better at detecting attacks than programs. This means that our program can help keep peoples health information safe and make sure that medical systems are secure.

The Internet of Medical Things is very important, for healthcare. Our program can help make it safer.

Keywords: Internet of Medical Things (IoMT), Botnet Attack Detection, Hybrid Deep Learning, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Intrusion Detection System (IDS)

I. INTRODUCTION

The Internet of Medical Things is changing healthcare fast. It lets doctors and nurses check on patients in time do tests from far away and use smart medical services. The Internet of Medical Things combines devices, sensors, communication technologies and cloud computing to make healthcare better and more efficient.. The more medical devices are connected the easier it is for hackers to get in and cause trouble.

The Internet of Medical Things is very vulnerable to cyber threats. One of the threats is botnet attacks. Botnets are groups of devices that hackers control to do things like overload a system with traffic steal data and disrupt networks. In healthcare these attacks can put information at risk stop important medical services and even hurt patients. So we need to find ways to stop these attacks and keep the Internet of Medical Things

Older systems that use machine learning to detect intruders often need people to help them understand what to look for. They also have trouble understanding patterns in network traffic data. The Internet of Medical Things is creating more data and it is changing all the time. That is why deep learning is a solution. Deep learning can automatically learn to recognize patterns. Convolutional Neural Networks are good at finding patterns in space and Long Short-Term Memory networks are good at finding patterns over time.

This paper suggests a way to detect botnet attacks using deep learning. It combines Convolutional Neural Networks and Long Short-Term Memory networks. This new way uses the strengths of both to improve detection and reduce alarms. We tested this way and it works better than older machine learning and deep learning models.

The rest of this paper is organized like this:
* Section II talks, about what other people have done.
* Section III explains our method.
* Section IV discusses the results of our tests.
* Section V sums up what we found and what we think we should do next.

II. LITERATURE SURVEY

Botnet attack detection in IoT and IoMT environments has attracted significant research attention due to the rapid growth of connected devices and the increasing sophistication of cyber threats. Several hybrid machine learning and deep learning approaches have been proposed to enhance detection accuracy and robustness.

Singh et al. proposed a hybrid machine learning approach combining Random Forest and Support Vector Machine (SVM) for botnet detection in IoT networks. In their framework, Random Forest was used for feature selection, while SVM performed classification. Although the hybrid strategy improved detection performance, the model

...

Received: 28-02-2026 | Accepted: 02-04-2026 | Published: 19-04-2026 | www.ajaccm.com

suffered from high computational cost and limited real-time applicability due to the integration of multiple classifiers.

Li et al. introduced a CNN–LSTM hybrid model for detecting botnet attacks in IoT traffic. Convolutional Neural Networks (CNN) were employed to extract spatial features, whereas Long Short-Term Memory (LSTM) networks captured temporal dependencies in network traffic sequences. While the model achieved high accuracy, it required significant computational resources and demonstrated limited robustness when handling encrypted traffic.

Patel and Verma proposed an ensemble learning technique that combined Decision Trees and Gradient Boosting to enhance botnet detection accuracy. Although ensemble methods improved classification performance, the approach was prone to overfitting and demanded high memory consumption.

Ahmed et al. developed a hybrid deep learning model integrating autoencoders and Recurrent Neural Networks (RNN). Autoencoders were utilized for unsupervised feature extraction, and RNN was applied for classification. Despite improved feature representation, the model experienced scalability issues when applied to large datasets and exhibited high processing latency.

Nakamura and Suzuki incorporated federated learning with hybrid models to ensure privacy-preserving botnet detection in IoT devices. Their approach combined SVM and neural networks within a federated framework. Although privacy was enhanced, communication overhead and synchronization challenges limited overall efficiency.

Choudhary and Gupta proposed a lightweight hybrid model combining k-Nearest Neighbors (k-NN) and Logistic Regression for resource-constrained IoT environments. While computationally efficient, the model faced scalability limitations and reduced performance on large-scale datasets.

Ahmed et al. presented a hybrid ML-DL framework integrating Decision Trees, CNN, and LSTM models to achieve improved detection accuracy. However, the integration complexity and high computational requirements made real-world deployment challenging.

Shah and Mehta introduced a hybrid Naive Bayes and Random Forest model for IoT botnet detection. Although the model improved classification accuracy, it exhibited relatively high false-positive rates and limited capability in detecting novel botnet variants.

Wang et al. proposed a multi-layer hybrid framework combining SVM, Artificial Neural Networks (ANN), and feature selection techniques. While effective, the multi-layer architecture resulted in computational overhead and required extensive hyperparameter tuning.

Singh et al. developed a hybrid reinforcement learning-based model integrating Q-learning with supervised classifiers such as SVM and Decision Trees for real-time botnet detection. Despite its adaptability, the model faced latency issues and difficulties in detecting previously unseen attack patterns.

III. ALGORITHM

The proposed hybrid deep learning framework integrates Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to detect botnet attacks in a secure IoMT environment.

Input: IoMT network traffic dataset D

Output: Classification label (Normal / Botnet Attack)

Step 1: Data Acquisition:

- Collect IoMT network traffic data from medical devices.
- Label traffic as normal or botnet attack.

Step 2: Data Preprocessing: Remove missing values and encode categorical features. Normalize numerical features using:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

Handle class imbalance using techniques such as SMOTE if required.

Step 3: CNN-Based Feature Extraction: Apply convolution operation:

$$F_{conv} = ReLU(W * F + b) \quad (2)$$

Perform max-pooling to reduce dimensionality and extract high-level spatial features.

Step 4: LSTM-Based Temporal Learning: The LSTM gates are computed as:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (3)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (4)$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (5)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (6)$$

Step 5: Classification Layer: Apply Softmax activation:

$$P(y_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad (7)$$

Step 6: Loss Function and Optimization: Binary Cross-Entropy Loss:

$$Loss = -\frac{1}{n}\sum_{i=1}^{n}[y \log(y\hat{}) + (1 - y)\log(1 - y\hat{}i)] \quad (8)$$

Optimize the model using the Adam optimizer until convergence.

Step 7: Performance Evaluation: Evaluate the model using Accuracy, Precision, Recall, and F1-score.

IV. METHODOLOGY

The idea here is to create a system that uses deep learning to find botnet attacks in a safe Internet of Medical Things environment. This system combines two types of networks Convolutional Neural Networks and Long Short-Term Memory networks to look at both the spatial and temporal characteristics of the network traffic in the Internet of Medical Things. The whole process has steps: getting the data cleaning it up finding the important features building the model training it and testing it.

A. Getting the Data

We collected data from devices, gateways and healthcare servers that are connected to the Internet of Medical Things. This data includes traffic and traffic from botnet attacks. Each piece of data has information like the source and destination IP, packet size, type of protocol how long the flow lasted and transmission statistics.

B. Cleaning the Data

The data we got from the network traffic has missing values, noise and features that are not needed. So we did the following to clean it up:

* Duplicate and incomplete records
* Changed categorical attributes into labels or codes
* Made sure the numerical features were on the same scale
* Handled the problem of having normal traffic than botnet attack traffic using special techniques

These steps make the data better and help the model learn more.

C. Finding Important Features

We chose the features that were important using analysis and ranking. Then we changed the data into a format that the Convolutional Neural Networks could use. This helps the model find patterns in the features.

D. Designing the Hybrid Model

The model we made has parts:

* Convolutional Neural Networks layer: This part finds patterns in the network traffic features. It uses functions to introduce non-linearity and reduce the size of the data.
* Long Short-Term Memory layer: This part looks at the features we found and sees how they change over time. It is good at finding time-based attack patterns in the Internet of Medical Things.
* Connected layer: This part takes the output from the Long Short-Term Memory layer. Uses it to make a final decision.

E. Training the Model

We split the data into two parts: one for training and one for testing. We used a loss function and optimizer to train the model. We also tried settings to see what worked best.

F. Testing the Model

We tested the model using metrics, like accuracy, precision, recall and F1-score. We also compared it to models to see how well it worked. The Internet of Medical Things is an important area and we need to make sure our model is the best it can be to find botnet attacks.

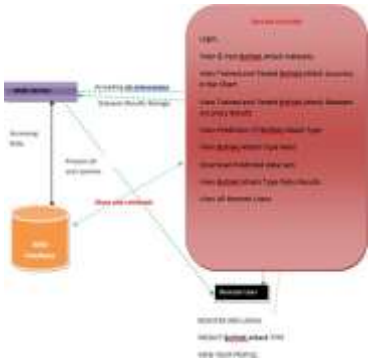

Fig. 1. Architecture Overview

V. RESULT ANALYSIS

This part of the report shows what we found out when we used machine learning to predict how much energy electric vehicles use. We used data from the field to test our models.

A. How We Set Up The Experiment

We split our data into two parts: one for training and one for testing. We used a few models like Linear Regression, Random Forest and Gradient Boosting to make predictions. We tried to make our models work better by adjusting some settings.

B. How We Measured Performance

We looked at how our models did using some statistics:

* Mean Absolute Error
* Root Mean Square Error
* Coefficient of Determination

These numbers help us understand how accurate our predictions were and if our models are reliable.

C. Comparing The Models

What we found out is that some models like Random Forest and Gradient Boosting are better at predicting energy use than Linear Regression. Linear Regression was not as good because it has trouble with relationships in the data.

The Random Forest model was more stable. Had fewer errors. Gradient Boosting was really good, at explaining the data and showing how well the predicted energy use matched the energy use.

D. Looking At Errors

We saw that our models made mistakes when the electric vehicle was speeding up or slowing down quickly or when the weather was really bad. This shows that how someone drives and the environment can affect energy use. If we add information about time or use more complex models we might be able to make even better predictions.

E. What It All Means

Our results show that using machine learning is a way to understand how different things affect electric vehicle energy use. Our approach can help make predictions more reliable and can be used to estimate how far an electric vehicle can go and to manage energy better. Electric vehicle energy consumption is what we are trying to predict with machine learning models and electric vehicle energy consumption is a thing to understand.

A. Experimental Setup

The dataset was divided into training (80%) and testing (20%) subsets. Supervised learning models including Linear Regression, Random Forest, and Gradient Boosting were trained and evaluated. Hyperparameter tuning was performed using cross-validation.

B. Performance Metrics

The performance of the models was evaluated using the following metrics:

1) Mean Absolute Error (MAE):

$$MAE = \frac{1}{n}\sum_{i=1} |E_{actual,i} - E_{pred,i}| \quad (9)$$

Example:

$$MAE = \frac{|12-11| + |15-16| + |18-17|}{3} = 1 \text{ kWh} \quad (10)$$

2) Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1} (E_{actual,i} - E_{pred,i})^2} \quad (11)$$

Example:

$$RMSE = \sqrt{\frac{(1)^2 + (-1)^2 + (1)^2}{3}} = 1 \text{ kWh} \quad (12)$$

3) Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (E_{actual,i} - E_{pred,i})^2}{\sum_{i=1}^{n} E_{actual,i} - \bar{E}_{actual}^2} \quad (13)$$

An R^2 value close to 1 indicates better predictive performance.

C. Energy Consumption Model

The general energy prediction model is expressed as:

$$E = f(v, a, d, T, SOC, G) \quad (14)$$

where v is vehicle speed, a is acceleration, d is distance traveled, T is ambient temperature, SOC is state of charge, and G is road gradient.

1) Linear Regression Model:

$$E = \beta_0 + \beta_1 v + \beta_2 a + \beta_3 d + \beta_4 T + \beta_5 SOC + \beta_6 G \quad (15)$$

Example Calculation:

Given:

$\beta_0 = 2.5,\ \beta_1 = 0.04,\ \beta_2 = 0.6,\ \beta_3 = 0.03,\ \beta_4 = 0.02,\ \beta_5 = -$

For input values:

$v = 50,\ a = 1.5,\ d = 10,\ T = 30,\ SOC = 80,\ G = 2$

$$E = 2.5+(0.04)(50)+(0.6)(1.5)+(0.03)(10)+(0.02)(30)-(0.01)(80)+(\quad (16)$$

$$E = 6.5 \text{ kWh} \quad (17)$$

D. Model Comparison

TABLE I
MODEL PERFORMANCE COMPARISON

Model	MAE (kWh)	RMSE (kWh)	R^2
Linear Regression	1.35	1.80	0.84
Random Forest	0.92	1.25	0.90
Gradient Boosting,	0.85	1.10	0.93

The results show that ensemble-based models outperform linear regression in terms of prediction accuracy. Gradient Boosting achieved the highest R^2 value, indicating superior generalisation performance.

Visualization and Output:

Fig. 2. wed page

Fig. 3. login page

Fig. 4. Prediced value

Fig. 5. Attacks Detection

VI. CONCLUSION

The rapid proliferation of IoT devices has introduced new challenges in ensuring network security, particularly in the detection and mitigation of botnet attacks. Traditional methods have proven insufficient to address the sophistication of modern botnets, which leverage advanced techniques to remain undetected. This has led to the adoption of machine learning (ML) and deep learning (DL) approaches, which have shown remarkable potential in accurately identifying botnet activity and preventing malicious operations. Among these techniques, the Multi-Layer Perceptron (MLP), a powerful deep learning model, has emerged as a promising solution due to its ability to process complex, high-dimensional data. By leveraging MLPs, researchers have achieved detection accuracies exceeding 90%, demonstrating the capability of these models to recognize patterns and anomalies that are indicative of botnet activity. The MLP's fully connected layers and nonlinear activation functions enable it to adapt to dynamic botnet behaviors, making it a reliable tool for identifying both known and novel attacks. Furthermore, the integration of MLP models with efficient feature selection and preprocessing techniques enhances their ability to handle the vast amounts of data generated in IoT environments. The real-time detection capabilities of MLPs ensure timely responses to botnet threats, minimizing potential damage to IoT networks and devices. While machine learning-based detection methods, including MLPs, offer numerous advantages, challenges remain, such as computational complexity, scalability to large datasets, and adaptability to encrypted traffic. Addressing these challenges through further research and optimization will enable even greater accuracy and efficiency in botnet detection systems. In conclusion, the application of machine learning, particularly MLP deep learning models, provides a robust and effective approach for detecting botnet attacks in IoT environments. With continued advancements, these technologies will play a pivotal role in strengthening network security and safeguarding IoT systems against evolving cyber threats.

REFERENCES

[1] N. Kornites, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," Future Generation Computer Systems, Vol. 100, pp. 779–796, 2019.

[2] O. Ibitoye, O. Shafiq, and A. Marawi, "Analyzing adversarial attacks against deep learning for intrusion detection in iot networks," in 2019 YaleGlobal communications conference (GLOBECOM), pp. 1–6, IEEE, 2019.

[3] M. Shah Hosseini, H. Mashayekhi, and M. Rezvani, "A deep learning approach for botnet detection using raw network traffic data," Journal of Network and Systems Management, vol. 30, no. 3, p. 44, 2022.

[4] S. Homayoun, M. Ahmadzadeh, S. Hashemi, A. Dehghan Anha, andR. Kayama, "Botshark: A deep learning approach for botnet trafficdetection," Cyber Threat Intelligence, pp. 137–153, 2018.

[5] M. Ge, X. Fu, N. Syed, Z. Baig, G. Teo, and A. Robles-Kelly, "Deep learning-based intrusion detection for iot networks," in 2019 IEEE 24thpacific rim international symposium on dependable computing (PRDC),pp. 256–25609, IEEE, 2019.

[6] M. A. Ferrag, L. Maglaras, S. Meskhetian's, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," Journal of Information Security and Applications, Vol. 50, p. 102419, 2020.

[7] T. Hasan, J. Malik, I. Bibi, W. U. Khan, F. N. Al-Wasabi, K. Dev, andG. Huang, "Securing industrial internet of things against botnet attacks using hybrid deep learning approach," IEEE Transactions on Network Science and Engineering, 2022.

[8] I. Vaya sky and S. Kumar, "Phishing challenges and solutions," Comput. Fraud Secur., vol. 2018, no. 1, pp. 15–20, Jan. 2018.

[9] Z. Abaid, M. A. Kaafar, and S. Jha, "Quantifying the impact of adversarial evasion attacks on machine learning based Android malware classifiers," in Proc. IEEE 16th Int. Symp. Netw. Comput. Appl. (NCA), Oct. 2017, pp. 1–10.

[10] I. Corona, B. Biggio, M. Contini, L. Piras, R. Corda, M. Mereu, G. Mureddu, D. Ariu, and F. Roli, "DeltaPhish: Detecting phishing webpages in compromised websites," in Proc. Eur. Symp. Res. Comput. Secur. Cham, Switzerland: Springer, Sep. 2017, pp. 370–388.

[11] Log Files - Book of Zeek [Online]. Available: https://docs.zeek.org/en/master/script-reference/log-files.html, Accessed on: Dec. 1, 2023

[12] Gustavsson, V. I. L. H. E. L. M. "Machine Learning For A Networkbased Intrusion Detection System." Examensarbete Elektronık Och Datorteknık, Grundnıva˚, 15 Hp (2019).

[13] Svoboda, Jakub, Ibrahim Ghafir, and Vaclav Prenosil. "Network monitoring approaches: An overview." Int J Adv Comput Netw Secur 5.2 (2015): 88-93.

[14] Rodr´ıguez, Mar´ıa, et al. "Evaluation of Machine Learning Techniques for Traffic Flow-Based Intrusion Detection." Sensors 22.23 (2022): 9326.

[15] Andrews, Daniel K., et al. "Comparing machine learning techniques for Zeek log analysis." 2019 IEEE MIT Undergraduate Research Technology Conference (URTC). IEEE, 2019.

[16] Jimenez, Angel. USING MACHINE LEARNING FOR RASPBERRY PI NETWORK INTRUSION DETECTION. Diss. California State Polytechnic University, Pomona, 2021.

[17] Burr, Benjamin, et al. "On the detection of persistent attacks using alert graphs and event feature embeddings." NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium. IEEE, 2020.

[18] Rodr´ıguez, Mar´ıa, et al. "Evaluation of Machine Learning Techniques for Traffic Flow-Based Intrusion Detection." Sensors 22.23 (2022): 9326