International Journal of Computer and Engineering Optimization (IJCEO)
Volume 01, Issue 02, November – December (2023)

RESEARCH ARTICLE

IOT BASED MALWARE ATTACK DETECTION USING LAYERED DEEP LEARNING FRAMEWORK

Vishnu Karthik Ravindran[1, *]

[1]Independent Researcher.

[*]*Corresponding e-mail: vkacharaya@gmail.com*

Abstract – Internet of Things (IoT) is the network connected among physical devices with various sensors which plays a key role in modern and smart societies. However, the IoT paradigm is prone to security concerns as many attackers try to hit the network and make it vulnerable. To overcome these issues, a novel Firefly optimization Algorithm combined SNN with Conv-BiLSTM for attack termination in IoT (FAST-IoT) approach is proposed for detecting attacks in IoT environment. Initially, the IoT devices generate data packets, which undergo data pre-processing. The processed data is then passed through a Spiking Neural Network model (SNN) for feature extraction. These features are selected by using the Firefly Optimization Algorithm (FOA) which is fed into a Convolutional-Bidirectional Long Short-Term Memory (Conv-BiLSTM) model. The Conv-BiLSTM model classifies the input as either Attack or No-Attack and if an attack is detected, it is blocked otherwise, the user proceeds without interruption. The FAST-IoT approach is evaluated by using the N-BaIoT dataset and it is simulated by a cloud simulator (Cloudsim). The proposed FAST-IoT approach achieves the highest accuracy of 93.2%, compared to 65.8% for DeepAK-IoT, 78.6% for MalBoT-DRL, and 87.4% for Hierarchical Cloud DNN respectively.

Keywords – *Malware Attack Detection, Internet of Things, Spiking Neural Network model, Firefly Optimization Algorithm.*

1. INTRODUCTION

The Internet of Things can be connected through smart terminals and mechanisms such as industrial systems, machine systems, autonomous vehicles, and basic wearables [1]. Numerous widespread attacks and threats have highlighted the security flaws in these networked IoT devices because of the easy accessibility of the Internet [2]. By 2025, an average of 30.9 billion devices are anticipated, with an emphasis on IoT devices. Malware assaults are among the risks and difficulties associated with cybersecurity in these networks and connected devices on the Internet [3].

Because it may be used to launch nearly any kind of cyberattack on linked devices—which, in extreme circumstances, might result in significant losses—the Internet of Things is a platform that attracts hackers [4]. Attackers may utilize current malware and massive attack strategies on harmful platforms. There isn't yet a protocol for the Internet of Things that can guarantee total security for linked devices [5]. Because no one standard supports all smart devices and integrated security procedures, IoT infrastructure is vulnerable to a wide range of assaults that pose serious security risks. In the end, strengthening defenses against the increasing amount of cyber threats and attacks will be required [6].

IoT application security risks can be reduced with the use of intrusion detection systems applied within the network [7]. Incoming and outgoing internet traffic gathered by Internet of Things (IoT) devices is continuously monitored by IDS, which detects any indications of a cyberattack [8]. Intrusion detection systems that rely on signatures identify potential threats by applying standards generated from previously identified assaults to recorded occurrences [9]. Conversely, anomaly-based approaches use a state's regular behavior to create simulations. Using this strategy, hostile actions can then be distinguished by their differences from acquired behaviors [10].

The complexity and difficulty of security monitoring are increasing due to the IoT infrastructure's rapid development [11]. Threats like as malware, viruses, and hackers are significant because they may compromise data integrity. IoT security is further diminished by insecure data, increasing dangers. For this reason, enhancing and modernizing security is crucial for the Internet of Things [12]. However, there are other IoT obstacles at every layer brought about by the interaction of numerous items and humans, such as privacy and security concerns, architectural responsibilities, and interoperability issues [13]. The major contributions of the proposed model are as follows,

- Initially, the IoT devices generate data packets, which undergo data pre-processing.
- The processed data is then passed through a Spiking Neural Network model (SNN) for feature extraction.
- These features are selected by using the Firefly Optimization Algorithm (FOA) which is fed into a

Convolutional-Bidirectional Long Short-Term Memory (Conv-BiLSTM) model.

- The Conv-BiLSTM model classifies whether the input as either Attack or Non-Attack.
- The accuracy, sensitivity, precision, recall, F1 score, malware detection rate and computational time are among the criteria used to assess the effectiveness of the proposed FAST-IoT method.

This is how the rest of the document is structured. In the second section, current research on malware categorization and detection in the Internet of Things is presented, including machine learning and deep learning models using various datasets. The suggested research approach is presented in the third section. In Section IV, the outcomes are examined and contrasted using performance metrics. The fifth and final section wraps up the suggested work and makes some recommendations for further research.

2. LITERATURE SURVEY

The development of efficient frameworks to identify malware attacks and safeguard IoT environments has drawn the attention of numerous academics. A strong defense against malicious activity on a network is an intrusion detection system. By connecting novel attacks to recurring attacks, the suggested method can aid in the detection of new threats. Anomaly-based and signature-based detection are the two primary methods for detecting intrusions into networks. They identify attacks by taking patterns from the datasets.

In 2021 Sahu, et al. [14] suggested utilizing a hybrid Deep Learning model to identify Internet of Things attacks. By utilizing CNN and LSTM models to bridge the gaps, the suggested approach creates a security framework and an attack detection mechanism that aids in the efficient identification of hostile devices. The suggested approach obtains a comparable accuracy of 96% when tested on 20 Internet of Things devices that have been infected with the Raspberry Pi malware.

In 2021 Palla, and Tayeb, [15] suggested to use of IOT devices to detect Mirai infections intelligently. The suggested approach detects Mirai malware by utilizing ML approaches that are built with MATLAB 2018b. When the suggested technique was tested using ANN on the Benign and Mirai datasets, it obtained a 0.3 false negative rate and 92.9% accuracy.

In 2022 Ali et al. [16] provide a practical multi-task deep learning method that uses behavioral traffic analysis to identify and detect IoT malware. When malicious network traffic is detected, the suggested LSTM-based model recognizes the type of malware and assesses whether the delivered traffic is malicious or benign. The accuracy of the suggested approach to identify hostile networks was 92.63%, 88.45%, and 95.83%, respectively.

In 2022 Yang et al. [17] suggested a deep learning and knowledge graph-based intrusion detection system for Internet of Things networks. A CNN-BiLSTM model that is attention-based is created to detect malicious query attacks in the suggested method, which turns queries from the Internet of Things network into word vectors. 90.01% intrusion detection accuracy is attained with the suggested approach.

In 2023 Ding et al. [18] Identify cyberattacks in the Internet of Things networks by putting forth a successful deep learning model. Remaining-based spatial representation block (RSR), temporal representation block (TRB), and detection block (DB) are the three blocks of the suggested DeepAK-IoT approach for identifying cyberattacks on IoT devices. Edge-IIoTset, UNSW-NB15, and TON-IoT are used to assess the DeepAK-IoT approach. 90.57%, 94.96%, and 98.41%, respectively, were the accuracy rates attained by the DeepAK-IoT technique for TON IoT, Edge-IIoTset, and UNSW NB15.

In 2023 Al-Fawa'reh, et al. [19] suggested detecting malicious botnets on IoT networks using deep reinforcement learning. The MalBoT-DRL technique combines amortized incremental statistics with an attention reward system that dynamically adapts to the constantly evolving malware models in the Internet of Things environment. The MalBoT-DRL approach was evaluated using the MedBIoT and N-BaIoT datasets, yielding average detection rates of 99.80% and 99.40%, respectively.

In 2024 Mosleh, and Sharifian, [20] suggested a productive approach for IoT malware classification using distributed deep neural networks connected with the cloud. To expedite malware detection and minimize resource consumption and maintenance, the suggested Hierarchical Cloud DNN technique may efficiently expand from IIoT devices to the edge and cloud. Adjustable degree of malware detection accuracy. With a comparable accuracy of 98.90%, the suggested approach is assessed using the BIG 2015 dataset [21].

Authentication, access control, data privacy, user privacy, and intrusion detection systems are critical security concerns. Feature extraction from packages is also essential for abnormality detection.

3. PROPOSED FAST-IOT METHODOLOGY

In this section, a novel Firefly optimization Algorithm combined SNN with Conv-BiLSTM for attack termination in IoT (FAST-IoT) approach is proposed for detecting attacks in IoT environment. Initially, the IoT devices generating a data packets, which undergoes data pre-processing. The processed data is then passed through a Spiking Neural Network model (SNN) for feature extraction. These features are selected by using Firefly Optimization Algorithm (FOA) which is fed into a Convolutional-Bidirectional Long Short-Term Memory (Conv-BiLSTM) model. The Conv-BiLSTM model classifies the input as either Attack or No-Attack and if an attack is detected, it is blocked otherwise, the user proceeds without interruption. The block diagram of the proposed FAST-IoT model is depicted in Figure 1.

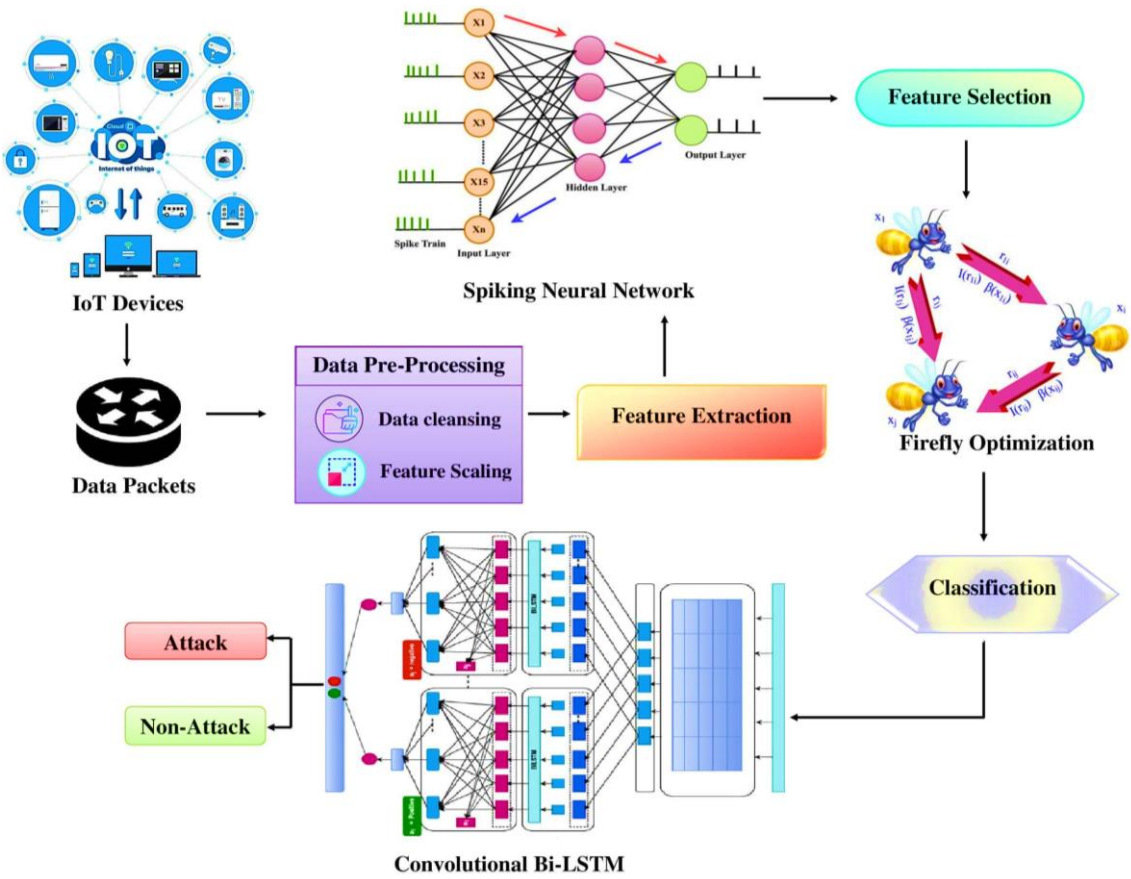

Figure 1. Block diagram of the Proposed FAST-IoT Model

3.1. Data Cleansing

Data cleaning, which comes before analysis, is taking out information that is redundant or inaccurate. By continually reiterating misconceptions, this kind of data undermines the model or algorithm that it supports. Missing data processing allows for the identification and removal of repetitive, duplicate, and NULL data. Through comparison with known causes, data anomalies can be located and eliminated. To facilitate an adequate evaluation of data quality, enter new data by the five requirements of validity, accuracy, consistency, uniformity, and completion.

3.2. Feature Scaling

Scaling features is another crucial aspect of data preparation. All features are separated within a specific, predictable range by applying feature scaling to the data. Common feature scaling methods include the conventional scaler. A learning measure for popular science is used to assess the suggested strategy.

3.3. Feature Extraction Using Spiking Neural Network

A specific kind of spike neural network (SNN) model is a neuron-to-neuron communication system that uses brief electrical impulses or pulses. SNN systems belong to the third generation of neural network techniques in terms of their dynamics. incorporates the ideas of processing individual items and storing data at intervals between vertices that are connected. The building blocks of the SNN architecture are a collection of delay-integrating neurons. A neuron denoted as j belongs to a set in which the connection weight w_ij between i and j is defined as follows:

$$x_j(t) = \sum_{i\in\Gamma_j} w_{ij}\, \varepsilon(t - t_i) \quad (1)$$

The synaptic potential can be calculated using equation (2):

$$\varepsilon(t) = \frac{t}{T} e^{1-\frac{t}{T}} \quad (2)$$

The constant τ determines the pulse width. Consequently, the number of synaptic connections of neuron j is described by equation (3):

$$x_j(t) = \sum_{i\in\Gamma_j} \sum_{k=1}^{m} w_{ij}^k\, y^k(t) \quad (3)$$

The output of the neuron is described as follows:

$$y_j^k(t) = \varepsilon(t - t_i - d_k) \quad (4)$$

The time interval d_k represents the difference between the post-synaptic neuron's waking time and the pre-synaptic neuron's active connectivity (k). The output potential u_j(t) of neuron j is shown in the final equation 5:

$$u_j(t) = \sum_{t_j^{(f)}\in F_j} \eta\,(t - t_j^{(f)}) + \sum_{i\in\Gamma_j} \sum_{t_i^{(g)}} w_{ij}\, \varepsilon(t - t_i^{(g)} - d_{ij}) \quad (5)$$

Were,

$$F_j = \{t^{(f)}; 1 \le f \le n\} = \{t | u_j(t) = \vartheta\} \quad (6)$$

The number of recorded pulses is n. It is necessary to program the kernel in three dimensions while dealing with SNN. In contrast, the most basic one-way or two-way architecture is used by the perceptron approach. Put another way, this basic activation function form allows all of the neuron outputs from each layer to be approximated at once. But the SNN activation function is more challenging because of its exponential structure, simultaneous mathematical approximations for temporal variables, and changing neuronal output.

Feature Selection Using Firefly Optimization Algorithm

The Firefly Algorithm (FA) is based on the behavioral and visual models of Firefly. FA considers two variables, such as variations in attractiveness and light intensity. Additionally, the lens function modifies the brightness. The firefly's brightness at a given position x is represented by I(x) ∘ f(x). On the other hand, attraction varies and is dependent upon the separation between firefly i and j. Moving away from the light source will also result in a drop in light intensity. Typically, the change in light intensity I(r) is governed by the inverse square law, as expressed in equation (7):

$$I(r) = \frac{I_s}{r^2} \tag{7}$$

where Is stands for the source's intensity. Equation (8) assumes that a particular medium has a constant light absorption coefficient (γ), and defines I as a function of distance r.

$$I(r) = I_0 e^{-\gamma r} \tag{8}$$

where I_0 is the initial intensity of light at r = 0. To prevent the singularity at r = 0 in the Is r2 command, equation (21) is evaluated in place of equations (9) and (10):

$$I(r) = I_0 e^{-\gamma r^2} \tag{9}$$

The ω of fireflies can also be found using equation (22):

$$\beta = \beta_0 e^{-\gamma r^2} \tag{10}$$

In this case, β0 represents the gravitational force at r = 0. As the exponential function computation requires more time than 1,1+r2, equation (11) is assessed similarly to equation (23):

$$\beta = \frac{\beta_0}{1+\gamma r^2} \tag{11}$$

Equation (12) expresses the distance between fireflyi and fireflyj as zi and zj, respectively:

$$r_{ij} = \left\| z_i - z_j \right\| = \sqrt{\sum_{k=1}^{d} (z_{i,k} - z_{j,k})^2} \tag{12}$$

where, in spatial coordinates zi,k stands for the kth element of the ith firefly. According to equation (13): Fireflies are drawn to more attractive fireflies.

$$z_i^{t+1} = z_i^t + \beta_0 e^{-\gamma r_{ij}^2} (z_j^t - z_i^t) + \alpha \varepsilon_i^t \tag{13}$$

Attraction is a part of the second stage. The third sentence is also employed for random generation. This is accomplished by using the random parameter α. I also show a vector of randomly generated numbers. FA also offers a faster convergence speed and more accuracy. It can strike a balance between local and global search and is also easy to deploy.

3.4. Classification Using Conv-BiLSTM Network

The network data stream T∈R l*d provides input to the 1D convolutional layer. N filters and the convolution kernel's width q are used to expand the convolution matrix in the 1D convolution layer, which produces the local features of n-grams. Filtering F_n yields the following feature maps, where 1 ~ n ~ N:

$$C_i^n = f(W^n \otimes X_{i:i+w-1} + b^n) \tag{14}$$

In this case, d stands for the vector dimension, ⊗ for the convolution operation, and b^n for the filter bias, F_n, the filter weight matrix. The formula for F_n is w∈ R q*d. Filter F_n's characteristic mapping yields the output C_i^n, where i th is an element of C_i^n. X_i's nominal filtering feature F_n X_(i:i+w-1) symbolizes the process of extracting features. In nonlinear activation, the symbol f stands for it. There is a nonlinear activation function (f) that comes following the RELU function. The network data is utilized to generate a feature map with a length of l.

$$C = [C1, C2, \ldots, Ci, Cl] \tag{15}$$

Max-Pooling Layer

The feature map generated by the convolution method is filtered by the pooling layer to eliminate the most significant features (c=max{c}), which are subsequently utilized to compute all local statistics. By integrating every input kernel size into a single maximum order output, a maximum amount of subsamples of input instances can be obtained.

In contrast with LSTM, Bi-LSTM uses two hidden states to allow bidirectional data flow. The classic RNN model must then decompose in order to add future information, using these bidirectional directions to hold input data of past and potential knowledge. Bi-LSTM is primarily achieved by concatenating two LSTM networks with opposing outputs. First, the forward LSTM state is used to collect the first dataset, and the reverse LSTM state is used to collect the second dataset. The network's structure allows it to remember both current and historical data. When employing a Bi-LSTM, the output of the first layer is used as input and the sequential output of the second layer is mixed with the final unit output of the preceding and subsequent levels. Upon stacking the BiLSTM layers, the outcome is h.

$$h = \left[h_{fd}, h_{bd}\right] \tag{16}$$

Where, h_{fd} denotes forward LSTM state and h_{bd} denotes the backward LSTM state.

Dense Layer

In the dense layer of the model, weights are utilized to link each input to each output. The output of the last layer is produced by the sigmoid function. The numerals 1 and 0 are used to denote the average of arbitrary results. The expected outcome of the sigmoid function is indicated in equation (8). To categorize the intrusion outcome as 0 or 1, binary cross

entropy is applied. In this illustration, the normal data is displayed as 1, and the assault data as 0.

$$Y = Sigmoid(wh + b) \quad (17)$$

Regularisation And Batch Normalisation

To prevent overfitting, mass regularization and standardization are required. Regularization can enhance the implemented procedure and improve the model's capacity to generalize. To prevent overfitting, the model will be updated during training. L2 elimination procedures and regularization are the most widely used methods. Put another way, they lose their short-term excitatory activity on downstream neurons during feedforward transmission, and weight changes are not communicated to downstream neurons. The Bi-LSTM layer regularity in this instance is set at 0.001 L2. According to a 10% drop probability, classes that are dropped happen both before and after the busiest class. Components are rearranged following each batch's processing as part of the batch standardization procedure. Consequently, the training process for deep networks will be more stable and require a great reduction in the number of training steps. To boost convergence and lessen the internal covariate bias, a batch normalization layer was added after the max pooling layer. The length of Conv BiLSTM is shown by equations (18) and (19).

$$T{\sim}O(M^2 * Q^2 * C_{in} * C_{out}) \quad (18)$$

$$T{\sim}O(M^2 * Q^2 * 2C_{in} * 2C_{out}) \quad (19)$$

where M is the histogram's output size, Q is the convolution kernel's size, and Cin, Cout, etc. are the number of input and output channels.

4. RESULT AND DISCUSSION

To achieve the best classification results, we have leveraged the best resources available to achieve our goals. As a result, we employed an Intel Core i7-6700HQ 2.6 processor with an Nvidia GTX 960 GPU to speed up the model training procedure. Tensor flow, OpenCV, SK learn, Num py, and Matplotlib libraries were utilized, along with Jupyter Notebook and Python 3.8 software packages.

4.1. Dataset Description

In the IoT context, port mirroring of switches yielded 155 features, which were gathered from the network data. The N-Ba IoT dataset is created using actual network traffic and contains nine commercial IoT devices as well as 23 essential attributes that are extracted at various intervals, including one minute, ten seconds, five minutes, and one millisecond. Numerous access points allow these gadgets to be linked to the WiFi network. The switches have been set up with port mirroring to gather and identify actual network traffic. The software Wireshark was used to record the data sets. Two well-known botnet assaults, BASHLITE and Mirai, are included in the N-Ba IoT dataset.

4.2. Performance Analysis

The performance measures F1 score, precision, recall, rate, and others are used to assess and compare various categorization models, each of which has unique benefits and drawbacks. The following is a description of accuracy and false positives (FPR):

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

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

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

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

The attack layer is the layer used in the detection task. In terms of precision, recall, and F1 score, the attack class is considered positive.

4.3. Comparative Analysis

A comparative analysis of the performance of four different models such as DeepAK-IoT, MalBoT-DRL, Hierarchical Cloud DNN, and the Proposed model across five evaluation metrics which are accuracy, specificity, precision, F1 Score, and recall.

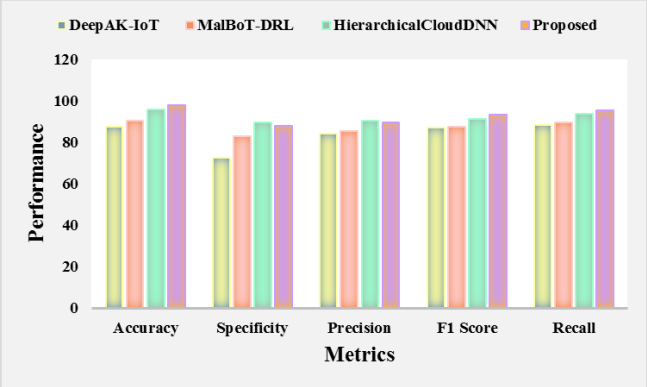

Figure 2. Performance comparison of the proposed and existing approach

The Performance comparison of the proposed and existing approach is shown in the Figure 2. The Proposed model and MalBoT-DRL show the highest accuracy, followed by Hierarchical Cloud DNN and then DeepAK-IoT. All four models show relatively high precision, with the Proposed model and MalBoT-DRL being slightly higher than Hierarchical Cloud DNN and DeepAK-IoT. Overall, the Proposed model exhibits strong and consistent performance across all metrics, indicating its effectiveness in the given task compared to the other models.

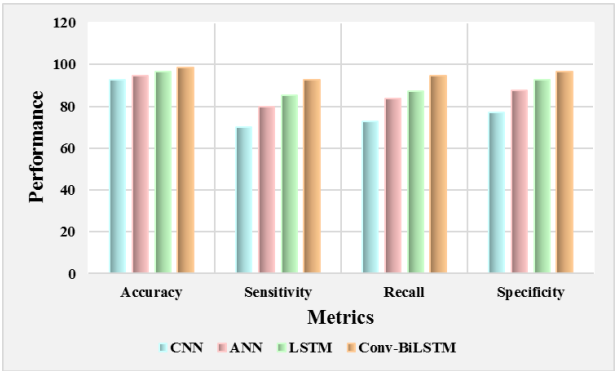

Figure 3. Performance comparison of networks

Figure 3. compares the performance of four models such as CNN, ANN, LSTM, and Conv-BiLSTM across four metrics which are accuracy, sensitivity, recall, and specificity. The Conv-BiLSTM model consistently outperforms others, especially in accuracy, recall, and specificity. ANN shows high accuracy and specificity but lower sensitivity and recall compared to Conv-BiLSTM. This suggests Conv-BiLSTM as the most effective model for the given task.

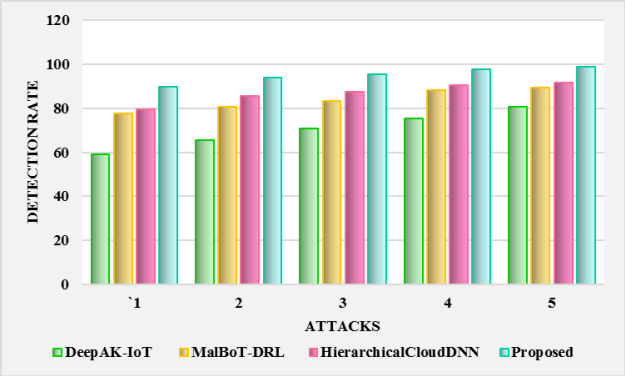

Figure 4. Attack detection rate comparison

Figure 4. illustrates the detection rate of different models such as DeepAK-IoT, MalBoT-DRL, Hierarchical Cloud DNN, and the Proposed model across the different attack scenarios. The Proposed model consistently achieves the highest detection rate across all attacks, indicating its superior performance in identifying various threats. The MalBoT-DRL and DeepAK-IoT perform moderately, with DeepAK-IoT having the lowest detection rates in most cases. This highlights the effectiveness of the Proposed model in handling diverse attack types.

5. CONCLUSION

This paper proposed a novel FAST-IoT approach for detecting attacks in an IoT environment. The performance of the FAST-IoT approach is simulated by using a cloud simulator (Cloud sim) and it is evaluated by using the N-Ba IoT dataset to detect attack events. The performance of this technique was assessed using accuracy, recall, F1 score, sensitivity, and detection rate. The proposed FAST-IoT approach achieves the highest accuracy of 93.2%, compared to 65.8% for DeepAK-IoT, 78.6% for MalBoT-DRL, and 87.4% for Hierarchical Cloud DNN respectively. To create a robust malware detection technique, machine learning, and deep learning techniques will be combined in the future to expand the proposed FAST-IoT method.

CONFLICTS OF INTEREST

This paper has no conflict of interest for publishing.

FUNDING STATEMENT

Not applicable.

ACKNOWLEDGEMENTS

The author would like to express his heartfelt gratitude to the supervisor for his guidance and unwavering support during this research for his guidance and support.

REFERENCES

[1] A. Hemmati, and A.M. Rahmani, "The Internet of Autonomous Things applications: A taxonomy, technologies, and future directions," *Internet of Things*, vol. 20, pp. 100635, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] O.I. Abiodun, E.O. Abiodun, M. Alawida, R.S. Alkhawaldeh, and H. Arshad, "A review on the security of the internet of things: Challenges and solutions," *Wireless Personal Communications*, vol. 119, pp. 2603-2637, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] A. Goudarzi, F. Ghayoor, M. Waseem, S. Fahad, and I. Traore, "A survey on IoT-enabled smart grids: emerging, applications, challenges, and outlook," *Energies*, vol. 15, no. 19, pp. 6984, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[4] A.E. Omolara, A. Alabdulatif, O.I. Abiodun, M. Alawida, A. Alabdulatif, and H. Arshad, "The internet of things security: A survey encompassing unexplored areas and new insights," *Computers & Security*, vol. 112, pp. 102494, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[5] F. Chantzis, I. Stais, P. Calderon, E. Deirmentzoglou, and B. Woods, "Practical IoT hacking: the definitive guide to attacking the internet of things," 2021. No Starch Press. [CrossRef] [Google Scholar] [Publisher Link]

[6] M. Shafiq, Z. Gu, O. Cheikhrouhou, W. Alhakami, and H. Hamam, "The Rise of "Internet of Things": Review and Open Research Issues Related to Detection and Prevention of IoT-Based Security Attacks," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, pp. 8669348, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] N. Mishra, and S. Pandya, "Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review," *IEEE Access*, vol. 9, pp. 59353-59377, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] J.B. Madavarapu, S. Nachiyappan, S. Rajarajeswari, N. Anusha, N. Venkatachalam, R.C.B. Madavarapu, and A. Ahilan, "Hot Watch: IOT based Wearable Health Monitoring System," *IEEE Sensors Journal*, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Y. Otoum, and A. Nayak, "As-ids: Anomaly and signature-based ids for the internet of things," *Journal of Network and Systems Management*, vol. 29, no. 3, pp. 23, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] S. Gokul Pran, and S. Raja, "An efficient feature selection and classification approach for an intrusion detection system using Optimal Neural Network," *Journal of Intelligent & Fuzzy Systems*, vol. 44, no. 5, pp. 8561-71, 1 2023 Jan. [CrossRef] [Google Scholar] [Publisher Link]

[11] L. Jenifer, X. Cheng, A. Ahilan, and P.J. Shermila, "A Novel Internet Of Things-Based Electrocardiogram Denoising Method Using Median Modified Weiner And Extended Kalman Filters". [CrossRef] [Google Scholar] [Publisher Link]

[12] E. Rehman, M. Haseeb-ud-Din, A.J. Malik, T.K. Khan, A.A. Abbasi, S. Kadry, M.A. Khan, and S. Rho, "Intrusion detection based on machine learning in the internet of things, attacks and counter measures," *The Journal of Supercomputing*, pp. 1-35, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[13] S.S. Albouq, A.A. Abi Sen, N. Almashf, M. Yamin, A. Alshanqiti, and N.M. Bahbouh, "A survey of interoperability challenges and solutions for dealing with them in IoT

environment," *IEEE Access*, vol. 10, pp. 36416-36428, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[14] A.K. Sahu, S. Sharma, M. Tanveer, R. Raja, "Internet of Things attack detection using hybrid Deep Learning Model," *Computer Communications*, vol. 176, pp. 146-154, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] T.G. Palla, and S. Tayeb, "Intelligent Mirai malware detection in IOT devices," In *2021 IEEE World AI IoT Congress (AIIoT)*, pp. 0420-0426, 2021, May. [CrossRef] [Google Scholar] [Publisher Link]

[16] S. Ali, O. Abusabha, F. Ali, M. Imran, and T. Abuhmed, "Effective multitask deep learning for iot malware detection and identification using behavioral traffic analysis," *IEEE Transactions on Network and Service Management*, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] X. Yang, G. Peng, D. Zhang, and Y. Lv, "An enhanced intrusion detection system for IoT networks based on deep learning and knowledge graph," *Security and Communication Networks*, vol. 2022, no. 1, pp. 4748528, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] W. Ding, M. Abdel-Basset, and R. Mohamed, "DeepAK-IoT: An effective deep learning model for cyberattack detection in IoT networks," *Information Sciences*, vol. 634, pp. 157-171, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[19] M. Al-Fawa'reh J., Abu-Khalaf, P. Szewczyk, and J.J. Kang, "MalBoT-DRL: Malware Botnet Detection Using Deep Reinforcement Learning in IoT Networks," IEEE Internet of Things Journal, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] M.R.B. Mosleh, and S. Sharifian, "An efficient cloud-integrated distributed deep neural network framework for IoT malware classification," Future Generation Computer Systems, vol. 157, pp. 603-617, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[21] S. Raja, S. Pran, N. Pandeeswari, P. Kiruthiga, D. Nithya, and G. Muthu Pandi, "Contemporary PCA and NBA based Hybrid Cloud Intrusion Detection System," EAI Endorsed Transactions on Energy Web, vol. 8, no. 36, 2021 Feb 19. [CrossRef] [Google Scholar] [Publisher Link]

AUTHORS

Vishnu Karthik Ravindran earned his Bachelor of Engineering in Computer Science and Engineering from Sri Ramakrishna Institute of Technology, India, in 2008. He later pursued a Master of Science in Computer Science at Syracuse University, New York, USA. Currently, he is a Software Engineer at a leading cloud-based company and actively contributes to open-source projects.

Arrived: 02.11.2023
Accepted: 04.12.2023