International Journal of Electrical and Computer Engineering (IJECE)
Vol. 14, No. 5, October 2024, pp. 5874~5883
ISSN: 2088-8708, DOI: 10.11591/ijece.v14i5.pp5874-5883

Deep learning model for elevating internet of things intrusion detection

Nitu Dash[1], Sujata Chakravarty[2], Amiya Kumar Rath[1]
[1]Department of Computer Science and Engineering, Faculty of Engineering, Biju Patnaik University of Technology, Rourkela, India
[2]Department of Computer Science and Engineering, Centurion University of Technology and Management, Odisha, India

Article Info

Article history:

Received Mar 31, 2024
Revised Jul 7, 2024
Accepted Jul 11, 2024

Keywords:

Bidirectional long short-term memory
Gated recurrent unit
Internet of things
Intrusion detection system
JAYA optimization

ABSTRACT

The internet of things (IoT) greatly impacts daily life by enabling efficient data exchange between objects and servers. However, cyber-attacks pose a serious threat to IoT devices. Intrusion detection systems (IDS) are vital for safeguarding networks, and machine learning methods are increasingly used to enhance security. Continuous improvement in accuracy and performance is crucial for effective IoT security. Deep learning not only outshines traditional machine learning methods but also holds untapped potential in fortifying IDS systems. This paper introduces an innovative deep learning framework tailored for anomaly detection within IoT networks, leveraging bidirectional long short-term memory (BiLSTM) and gated recurrent unit (GRU) architectures. The hyper parameters of the proposed model are optimized using the JAYA optimization technique. These models are validated using IoT-23 and MQTTset datasets. Several performance metrics including accuracy, precision, recall, F-score, true negative rate (TNR), false positive rate (FPR), and false negative rate (FNR), have been selected to assess the effectiveness of the suggested model. The empirical results are scrutinized and juxtaposed with prevailing approaches in the realm of intrusion detection for IoT. Notably, the proposed method emerges as showcasing superior accuracy when contrasted with existing methods.

Corresponding Author:

Sujata Chakravarty
Department of Computer Science and Engineering, Centurion University of Technology and Management
Odisha, India
Email: chakravartys69@gmail.com

1. INTRODUCTION

The internet of things (IoT) is a paradigm shift in which devices, vehicles, physical objects and appliances are interlinked via sensors and software, allowing them to share data and operate autonomously without human involvement. facilitating data collection and exchange [1]. The expansive IoT network extends Internet connectivity beyond traditional devices to include household gadgets, vehicles, and industrial equipment, fostering convenience, smarter cities, enhanced healthcare, and increased industry efficiency [2]. The heterogeneous nature of the IoT ecosystem presents challenges for security due to varying manufacturers and protocols, leaving devices vulnerable to known threats [3]. Data privacy concerns arise from continuous data collection [4]. Addressing these challenges is crucial for realizing the full potential of IoT while ensuring user trust and safety. Intrusion detection systems and responsive threat mitigation are essential security measures [5]. Consequently, the IoT research community have put forth several machine learning (ML) and deep learning (DL) techniques aimed at enhancing IoT security [6]. DL-based security methods autonomously learn heterogeneous features from unstructured data in IoT environments, effectively detecting mutated attacks and reducing the need for frequent patches, thus enhancing system resilience [7],

[8]. Kim *et al.* [9] proposed convolutional neural network and long short-term memory network (CNN-LSTM) intrusion detection systems (IDS) model utilizes normalized UTF-8 character encoding for spatial feature learning, achieves 91.54% and 93% accuracy on CSIC-2010 and CICIDS2017 datasets respectively. Susilo and Sari [10] used random forest (RF), CNN, and multi-layer perceptron (MLP) for IDS classification on the Bot-IoT dataset, with CNN achieving the highest accuracy at 91%. Aldhaheri *et al.* [11] developed an IDS using IoT-Bot dataset, spiking neural networks (SNN) for signal categorization, and dendritic cell algorithm (DCA) for classification, achieving over 98.73% accuracy. An LSTM-CNN hybrid for IoT intrusion detection in smart home networks achieves 98% accuracy, outperforming existing models with fewer false alarms, based on [12]. In study [13], a lightweight intrusion detection method for IoT networks, utilizing a dense random neural network (DnRaNN), demonstrates excellent performance on binary and multiclass categorization using the *ToN_IoT* dataset. The XGBoost classifier was implemented in a paper [14] to identify intrusions in IoT networks. Device-based intrusion detection system (DIDS), a novel deep learning model, excels in large networks with 99% accuracy, low false alarms, and superior performance [15]. The proposed approach for the IoT [16], deep integrated stacking-IoT (DIS-IoT), integrates four diverse deep learning techniques to achieve superior accuracy and low false positive rates. A novel LSTM-based IDS [17] for IoT networks provides explainable model conclusions by utilizing distinct input features from the SPIP framework, achieving high accuracy on various datasets.

Existing ML and DL-based security mechanisms have limitations, including outdated datasets, specific attack focus, emulated data, and underfitting due to limited training samples. This paper is motivated by the realization that selecting the right deep learning approach along with optimization technique and dataset can greatly improve accuracy in IoT IDS networks. The significant contributions in this paper are: i) Developing a BiLSTM and GRU-based deep learning model for IoT anomaly detection, optimized with JAYA algorithm for hyperparameter tuning, ii) Performs IoT attacks analysis using benchmark datasets to improve accuracy and reduce false alarm rate, and iii) Conducts systematic comparative experiments with contemporary research in the field.

The paper delineates the framework of the proposed model in section 2. Section 3 elaborates on the methodologies, including the utilization of IoT-2023 and MQTTset datasets, preprocessing steps, implementation of BiLSTM, GRU, and JAYA optimization. Section 4 showcases the results and analysis, followed by the conclusion in Section 5.

2. PROPOSED FRAMEWORK

This paper introduces deep learning models for IoT network anomaly detection. The model development involves four main steps. Firstly, IoT-2023 and MQTTset datasets are chosen and preprocessed through data cleaning, digitization, and normalization. Secondly, BiLSTM and GRU are employed for building the IDS model, with JAYA optimization technique to fine-tune the hyperparameters. Thirdly, the optimized BiLSTM and GRU model are trained to establish detection rules. Finally, the model's performance is evaluated with a testing dataset to ensure generalizability. Figure 1 shows the various stages of the proposed framework. Outlined below are the procedural steps undertaken to implement the proposed model for IDS in IoT network. IoT-2023 and MQTTset dataset are taken and preprocessed.

a. The architecture of BiLSTM and GRU based IDS model in IoT network is defined along with the hyperparameter optimization function.
b. An objective function that evaluates the performance of BiLSTM and GRU based IDS based on the chosen hyperparameters is calculated.
c. Jaya is used with an initial population of solutions. These solutions represent different configurations of the BiLSTM and GRU network.
d. Jaya iteratively improves the population by updating the solutions. The primary steps are:
   - Evaluation: The fitness (objective function value) for each solution is calculated in the population based on their hyperparameters and the BiLSTM and GRU network's performance.
   - Exploration: Potential solutions are explored by generating new hyperparameter configurations.
   - Update: Old solutions is replaced with new ones if they are better (lower fitness). Jaya updates the solutions by comparing each pair of solutions and selecting the one with the better fitness.
   - Termination: A termination criterion is decided, such as a maximum number of iterations or a target fitness value, to stop the optimization process.
e. After optimization, the best hyperparameters are extracted and the final JAYA-BiLSTMIDS and JAYA-GRUIDS model for IoT network is developed.
f. The final models are trained on the training dataset.
g. Lastly, the JAYA-BiLSTMIDS and JAYA-GRUIDS models undergo testing on the test dataset to evaluate their classification performance.

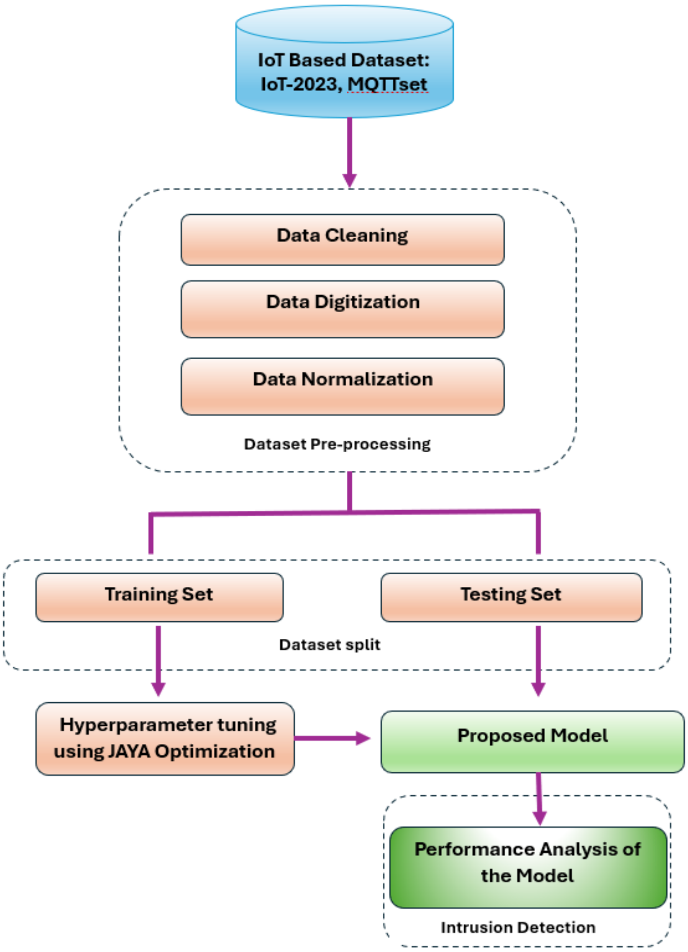

Figure 1. Proposed model

3. METHOD
3.1. Datasets description and preprocessing
3.1.1. IoT-2023 dataset

The IoT-23 dataset, offers researchers a substantial collection of labeled IoT security data, comprising 23 labeled events with 20 malicious and 3 non-malicious scenarios. It includes nine types of attack, reflecting real-world IoT network conditions [18]. Table 1 displays unique instances of both normal and malicious attacks in the IoT-23 dataset after redundancy removal.

3.1.2. MQTTset dataset

The MQTTset dataset captures data from real-world IoT networks using the MQTT protocol, featuring information from different IoT sensors linked to an MQTT broker and gathered through the IoT-Flock tool [19]. It includes details such as message payloads, timestamps, topic hierarchies, and other relevant metadata associated with MQTT-based interactions. The dataset encompasses various aspects of MQTT communication and contains records of normal and attack network behavior. It includes five distinct attacks pertaining to the IoT networks' MQTT communication protocol. Table 2 shows unique instances of normal and malicious attacks post redundancy removal.

3.1.3. Data preprocessing

It aims to clean and transform the data to make it suitable for training and testing models [20].

- Data cleaning: Duplicate records are removed, and missing values are either imputed or incomplete records are discarded.
- Data digitization: The dataset encompasses both character and numeric attributes. Character-based attributes have been transformed into their corresponding numeric representations for consistency and analysis.

- Data normalization: All the features are scaled within the range of [0-1] using (1).

$$f = \frac{f - min}{max - min} \tag{1}$$

where *f* is feature value, *max* is maximum value, and *min* is minimum value of the feature.

Table 1. Class distribution of IoT-2023 dataset

Category	No. of instances after removing redundancy
Benign	4,253,672
Attack	1,699,608
C&C	20,612
DDoS	4,619,869
File Download	7,707
HeartBeat	12,648
Mirai	756
Okiru	12,908,506
Port Scan	2,999,999
Torii	24,492
Total	26,547,869

Table 2. Class distribution of MQTTset dataset

Category	No. of Instances after removing redundancy
Benign	420,136
Bruteforce	4,513
MQTTFlood	77,756
MalariaDoS	11,265
Malformed	3,535
SlowITe	3,044
Total	520,249

3.2. Bidirectional long short-term memory

Bidirectional long short-term memory (BiLSTM) [21] is a sequence processing architecture utilizing two LSTM units [22], each consisting of input, forget, and output gates regulated by sigmoid neural network layers, enabling effective information flow and retention across long sequences. BiLSTM networks connect two distinct hidden LSTM layers in opposing directions while directing them towards the same output as presented in Figure 2. In this configuration, the input sequence undergoes processing in a forward manner by one LSTM layer, while the inverted version of the input sequence is simultaneously fed into another LSTM layer as a backward state layer in time [23]. At a specific timestep, denoted by t, the input is represented by $x_t = (x_1, x_2, x_3, \dots \dots, x_n) \in R^{nxd}$. The hidden states that are forward and backward are represented as $\vec{h} \in R^{nxd}$ and $\overleftarrow{h} \in R^{nxd}$. The computation is given in (2) to (4).

$$\vec{h}_t = \sigma \left(W_{\vec{h}} x_t + W_{\vec{h}\vec{h}} \vec{h}_{t-1} + b_{\vec{h}} \right) \tag{2}$$

$$\overleftarrow{h}_t = \sigma(W_{\overleftarrow{h}} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}) \tag{3}$$

$$y_t = W_{\vec{h}y} \vec{h}_t + W_{\overleftarrow{h}y} \overleftarrow{h}_t + b_y \tag{4}$$

The hidden state of the forward layer and the backward layer are merged in the output layer. The BiLSTM generates a sequence of hidden states as its output.

$$y_t = \sigma[\vec{h}_t, \overleftarrow{h}_t] \tag{5}$$

The σ function merges output sequences from both forward and backward LSTM layers, combining them based on their hidden states. The resulting final hidden state h_t encapsulates the complete sentence, where h_t is equal to $[\vec{h}_t, \overleftarrow{h}_t]$.

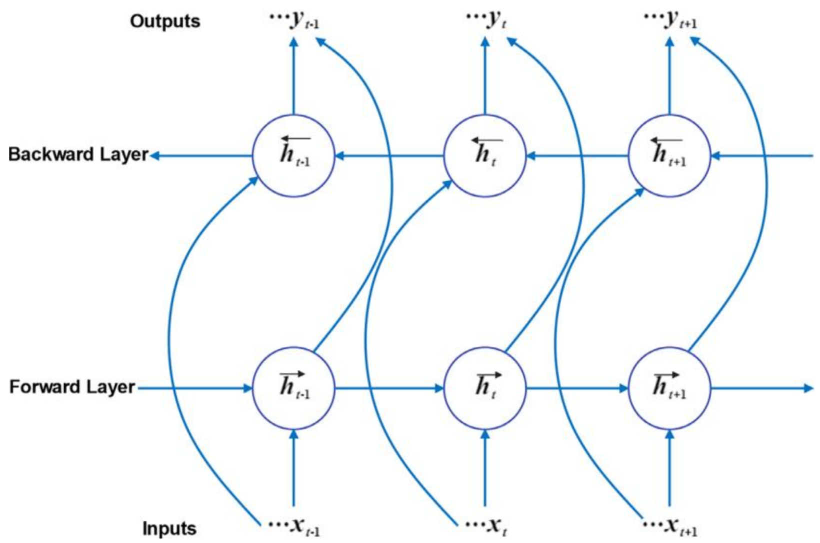

Figure 2. Bidirectional LSTM

3.3. Gated recurrent unit

A gated recurrent unit (GRU) simplifies LSTM by merging the "forget" and "input" gates into a single "update gate" and combining the hidden and cell states [24]. The GRU architecture includes reset and update gates, both supported by a single hidden state, using sigmoid for information flow and tanh for computing the output. Figure 3 illustrates a GRU cell. The reset and update gates are shown mathematically as (6) and (7).

$$r_t = \sigma((w_{xr}\, x_t + w_{hr}\, h_{t-1} + b_r)) \tag{6}$$

$$u_t = \sigma((w_{xu}\, x_t + w_{ur}\, h_{t-1} + b_u)) \tag{7}$$

where 'r_t' signifies the reset gate for a time stamp 't' and 'u_t' signifies the update gate. 'h_{t-1}' signifies the GRU's earlier hidden state, 'w' stands for the weight value, and 'b' represents the biases associated with the reset and update gates. The hidden state value is calculated utilizing (8) and (9).

$$\tilde{h}_t = tanh\,(w_{hx}\, x_t + w_h\, h\,(r_t\, h_{t-1}) + b_u)) \tag{8}$$

$$h_t = (1 - u_t)h_{t-1} + u_t\, \tilde{h}_t \tag{9}$$

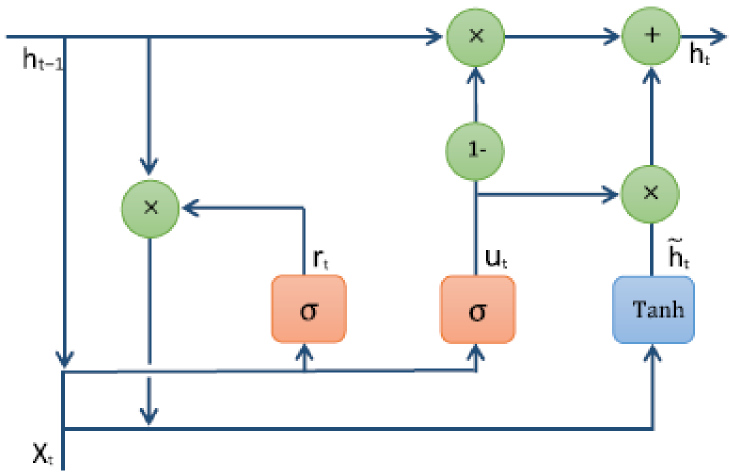

Figure 3. GRU cell operation

3.4. JAYA optimization

It a gradient-free metaheuristic inspired by natural selection, iteratively improves candidate solutions through exploration and exploitation without hyperparameters, aiming to distance from the worst and approach the best solution iteratively [25]. During exploration, solutions are compared and improved based on their fitness values, while during exploitation, the algorithm exploits the best solution found so far. This process continues until a stopping criterion is met or a satisfactory solution is found, as depicted in (10).

$$X_{j,k}^{i+1} = X_{j,k}^{i} + r_1(X_{j,best}^{i} - |X_{j,k}^{i}|) - r_2(X_{j,worst}^{i} - |X_{j,k}^{i}|) \tag{10}$$

where $X_{j,k}^{i}$ signifies the value of the j^{th} variable of the k^{th} particle at the i^{th} generation. $X_{j,best}^{i}$ denotes the value of the j^{th} variable of the best solution found within the i^{th} generation. $X_{j,worst}^{i}$ represents the value of the j^{th} variable of the worst solution identified in the i^{th} generation, r_1 and r_2 are two random numbers drawn from the uniform distribution $U(0,1)$. $X_{j,k}^{i+1}$ refers to the j^{th} variable of X_{k}^{i+1} i.e., the new solution or position to be evaluated. If the fitness improves, X_{k}^{i+1} replaces X_{k}^{i}.

4. RESULTS AND DISCUSSION

The experiments presented in this paper are carried out by integrating the TensorFlow backend with the Keras framework. Google Colab served as the platform for conducting these experiments. The proposed JAYA-BILSTMIDS and JAYA-GRUIDs models are subjected to validation using an extensive array of performance metrics. The evaluation metrics listed in (11) to (17), ensuring a thorough and complete assessment of model performance. Similarly, Table 3 outlines the parameters and hyperparameters utilized in BiLSTM, and GRU architecture for classification.

$$Accuracy\ (ACC) = \frac{(\mathrm{TP+TN})}{(\mathrm{TP+TN+FP+FN})} \tag{11}$$

$$Precision = \frac{\mathrm{TP}}{(\mathrm{TP+FP})} \tag{12}$$

$$Recall\ or\ TPR = \frac{\mathrm{TP}}{(\mathrm{TP+FN})} \tag{13}$$

$$F - score = 2 * \frac{(\mathrm{Precision} * \mathrm{Recall})}{(\mathrm{Precision} + \mathrm{Recall})} \tag{14}$$

$$TNR = Specificity = \frac{\mathrm{TN}}{(\mathrm{TN+FP})} \tag{15}$$

$$FPR = \frac{\mathrm{FP}}{(\mathrm{FP+TN})} \tag{16}$$

$$FNR = \frac{\mathrm{FN}}{(\mathrm{FN+TP})} \tag{17}$$

The hyperparameters are fine-tuned by Jaya optimization algorithm. The value of max iterations is set to 20 with lower bound as 50 and upper bound as 150, dimensionality as 1 and the number of search agents is 1. The optimizer adjusts the parameters like the batch size, epochs, and window size, leading to an improvement in the classification accuracy of the model.

Table 3. BiLSTM and GRU model parameters and hyperparameters for classification

Layer	Name	Configuration
Input	Input layer	Input features of IoT-2023 and MQTTset dataset
Hidden	BILSTM or GRU	Neuron Units=512, Bias, Kernel and Activity regularizer
	Activation	LeakyReLU (alpha = 0.2)
	Layer normalization	Center = True, Scale = True, Axis = 1
	Regularization	11 = 0.0001, 12 = 0.0001
	Dropout	Dropout rate = 0.2
Classification	Dense	Neuron = 512, Activation = ReLU
Output	Output layer	Teo neurons, Activation = SoftMax
Hyper parameters	Early stopping (monitor = 'loss', verbose = 1, patience = 6), optimizers = Adams, loss function = binary_crossentrophy, Learning rate = 0.001, Batch size = 120, epoch = 200 to 500.	

This paper performs binary classification on IoT-2023 and MQTTset datasets utilizing the JAYA-BiLSTMIDS and JAYA-GRUIDS model in IoT network and the outcomes are given in Table 4. The accuracy of JAYA-BiLSTMIDS for IoT-23 and MQTTset dataset are 99.65% and 99.88%. Similarly, the findings of JAYA-GRUIDS model on IoT-23 dataset are 99.42% accuracy and 99.45% accuracy on MQTTset dataset. Figures 4 and 5 show the graphical performance comparison of JAYA-BiLSTMIDS and JAYA-GRUIDS on IoT-23 and MQTTset dataset. It is clearly observed that JAYA-BiLSTMIDS shows better performance than JAYA-GRUIDS on both the datasets.

Table 4. JAYA-BiLSTMIDS and JAYA-GRUIDS model classification using IoT-2023 and MQTTset dataset

Model	Dataset	Class	Accuracy	Precision	Recall	F1-Score	TNR	FPR	FNR
JAYA-BiLSTMIDS	IoT-23	Normal	99.65	99.31	99.40	99.36	99.88	0.12	0.60
		Anomaly		99.89	99.88	99.89	99.40	0.60	0.12
	MQTTset	Normal	**99.88**	98.74	99.84	99.29	99.94	0.06	0.16
		Anomaly		99.99	99.94	99.96	99.84	0.16	0.06
JAYA-GRUIDS	IoT-23	Normal	99.42	98.96	99.11	99.03	99.82	0.18	0.89
		Anomaly		99.84	99.82	99.83	99.11	0.89	0.18
	MQTTset	Normal	99.45	99.15	98.99	99.07	99.85	0.15	1.01
		Anomaly		99.82	99.85	99.84	98.99	1.01	0.15

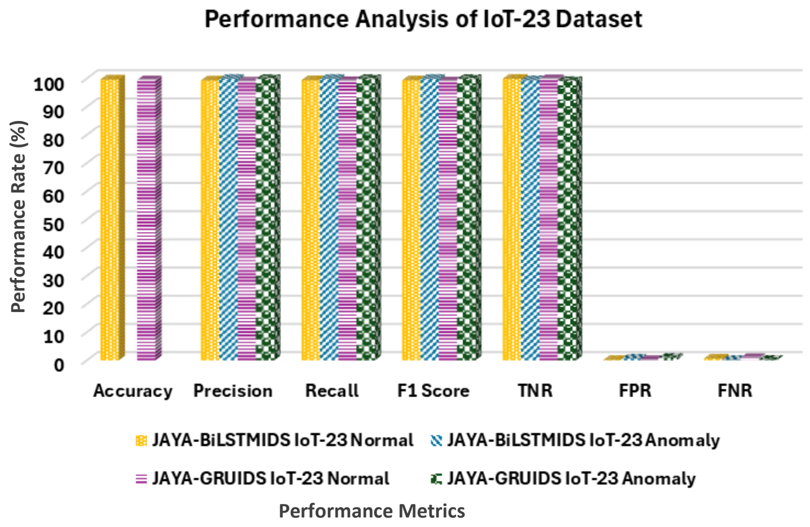

Figure 4. Performance analysis of JAYA-BiLSTMIDS and JAYA-GRUIDS on IoT-23 dataset

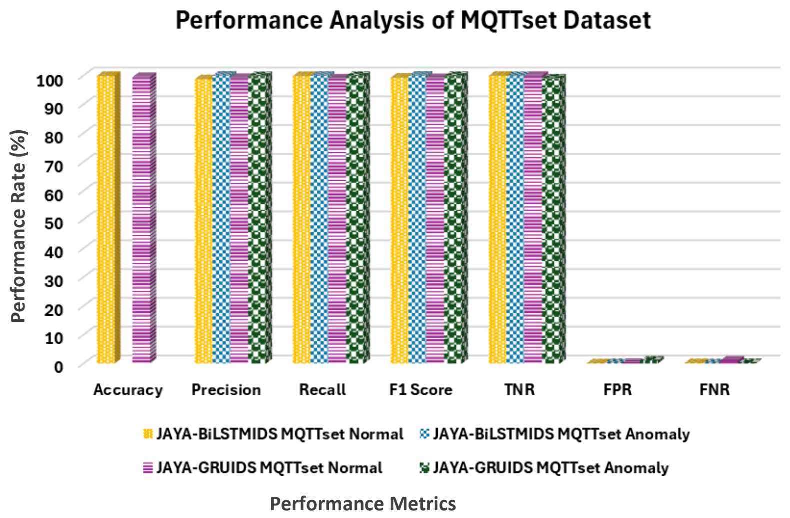

Figure 5. Performance analysis of JAYA- BiLSTMIDS and JAYA-GRUIDS on MQTTset dataset

A low false alarm rate FAR ensures that the IDS accurately identifies genuine security threats while minimizing false alerts. Figure 6. depicts the false alarm rate (FAR) of the proposed model, showcasing remarkable results with 0.55% and 0.51% on the IoT-23 and MQTTset datasets for JAYA-BiLSTMIDS, and 0.62% and 0.57% for JAYA-GRUIDS on the same datasets. Figure 7 shows the classification performance using receiver operating characteristic (ROC).

The key findings indicate that the proposed JAYA-based IoT IDS models showed high performance in anomaly detection across two datasets, proving their robustness. High accuracy, precision, recall, and F1-scores for classification highlight their effectiveness in identifying malicious activities in IoT networks. This research stands out for using JAYA optimization technique for exploring deep learning architectures. Strengths include developing a lightweight binary classification model. Limitation include scalability to larger datasets and real-time IoT deployment. As shown in Table 5, the effectiveness of the suggested model has been confirmed by comparison with other relevant papers.

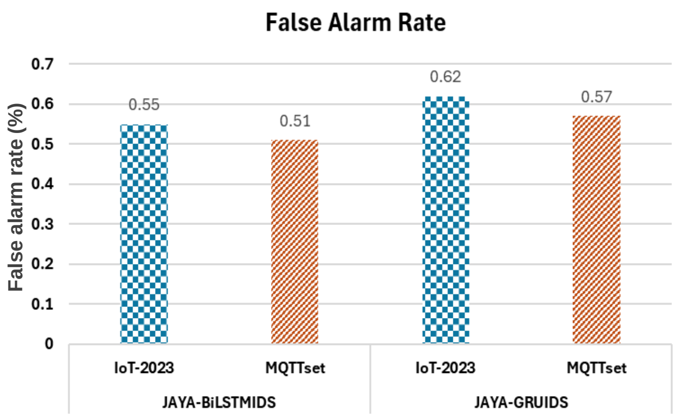

Figure 6. False alarm of JAYA-BiLSTMIDS and JAYA-GRUIDS on IoT-23 and MQTTset dataset

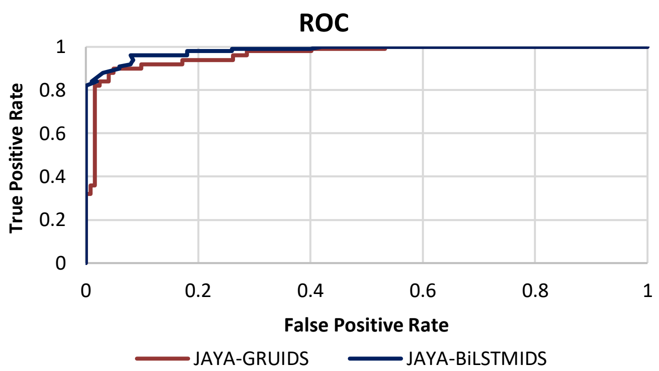

Figure 7. ROC of JAYA-BiLSTMIDS and JAYA-GRUIDS

Table 5. Performance Comparison of the proposed model with other articles

Article	Year	Model	Dataset	Accuracy	Precision	Recall	F1-Score	FAR
Kim *et.al.* [9]	2020	CNN-LSTM	CSIC-2010	91.54	98.54	68.26	80.65	-
			CICISC-2017	93.00	86.47	76.83	81.36	-
Aldhaheri *et.al.* [11]	2020	DeepDCS	BoT-IoT	98.73	99.17	98.36	98.77	-
Azumah *et.al.* [12]	2021	LSTM	IoT	98	83	84	83	-
Latif *et.al.* [13]	2022	DnRaNN	ToN_IoT	99.15	99.23	99.07	99.27	-
Madhu *et.al.* [15]	2023	DIDS	Real time data of an IoT network	99	97	96	97	-
Lazzarini *et.al.* [16]	2023	DIS-IoT	CICIDS2017	98.70	95.90	97.60	96.75	-
Proposed Model	2024	JAYA-GRU	IoT-2023	99.42	99.40	99.47	99.43	0.62
			MQTTset	99.45	99.48	99.42	99.40	0.57
		JAYA-BiLSTM	IoT-2023	99.65	99.60	99.64	63.00	0.55
			MQTTset	99.88	99.37	99.89	99.63	0.51

5. CONCLUSION

In this paper, a lightweight deep learning model, rooted in recurrent neural networks is presented to identify anomalies in IoT networks, highlighting the cybersecurity importance with a focus to increase the accuracy and reduce the FAR. The proposed model encompasses BiLSTM and GRU methodologies optimized by JAYA optimization technique, forming a comprehensive structure for analyzing anomalous activities aimed at intrusion detection in IoT networks. IoT-2023 and MQTTset datasets are used to assess the efficacy of the proposed model. The performance evaluation of JAYA-BiLSTMIDS on the IoT-23 dataset reveals an accuracy of 99.65%, while achieving 99.88% accuracy on the MQTTset dataset. Similarly, the JAYA-GRUIDS model attains an accuracy of 99.42% on the IoT-23 dataset and 99.45% on the MQTTset dataset. Notably, both proposed models demonstrate a low FAR, showcasing outstanding results with 0.55% and 0.51% on the IoT-23 and MQTTset datasets for JAYA-BiLSTMIDS, and 0.62% and 0.57% for JAYA-GRUIDS on the same datasets. It is observed that JAYA-BiLSTM yields better results than JAYA-GRUIDS in terms of accuracy and FAR. These findings highlight the potential of employing simpler architectures to attain comparable levels of IDS classification performance in IoT network. Future studies may investigate integrating these models into real world IoT systems and exploring ensemble methods to boost detection abilities.

REFERENCES

[1] M. M. Noor and W. H. Hassan, "Current research on internet of things (IoT) security: A survey," *Computer Networks*, vol. 148, pp. 283–294, Jan. 2019, doi: 10.1016/j.comnet.2018.11.025.
[2] S. Khadim, O. Ali Hassen, and H. Ibrahim, "A review on the mechanism mitigating and eliminating internet crimes using modern technologies: Mitigating internet crimes using modern technologies," *Wasit Journal of Computer and Mathematics Science*, vol. 1, no. 3, pp. 50–68, Sep. 2022, doi: 10.31185/wjcm.48.
[3] M. Aboubakar, M. Kellil, and P. Roux, "A review of IoT network management: Current status and perspectives," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, pp. 4163–4176, Jul. 2022, doi: 10.1016/j.jksuci.2021.03.006.
[4] B. K. Padhi, S. Chakravarty, B. Naik, R. M. Pattanayak, and H. Das, "RHSOFS: Feature selection using the rock hyrax swarm optimization algorithm for credit card fraud detection system," *Sensors*, vol. 22, no. 23, p. 9321, Nov. 2022, doi: 10.3390/s22239321.
[5] S. Bebortta, S. K. Das, and S. Chakravarty, "Fog-enabled intelligent network intrusion detection framework for internet of things applications," in *2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, IEEE, Jan. 2023, doi: 10.1109/confluence56041.2023.10048841.
[6] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (IoT) security," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020, doi: 10.1109/comst.2020.2988293.
[7] H. K. Channi, R. Sandhu, N. C. Giri, P. Singh, and F. A. Syam, "Comparison of power system flow analysis methods of IEEE 5-bus system," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 34, no. 1, p. 11, Apr. 2024, doi: 10.11591/ijeecs.v34.i1.pp11-18.
[8] L. Aversano, M. L. Bernardi, M. Cimitile, and R. Pecori, "A systematic review on deep learning approaches for IoT security," *Computer Science Review*, vol. 40, p. 100389, May 2021, doi: 10.1016/j.cosrev.2021.100389.
[9] A. Kim, M. Park, and D. H. Lee, "AI-IDS: Application of deep learning to real-time web intrusion detection," *IEEE Access*, vol. 8, pp. 70245–70261, 2020, doi: 10.1109/access.2020.2986882.
[10] B. Susilo and R. F. Sari, "Intrusion detection in iot networks using deep learning algorithm," *Information*, vol. 11, no. 5, p. 279, May 2020, doi: 10.3390/info11050279.
[11] S. Aldhaheri, D. Alghazzawi, L. Cheng, B. Alzahrani, and A. Al-Barakati, "DeepDCA: Novel network-based detection of IoT attacks using artificial immune system," *Applied Sciences*, vol. 10, no. 6, p. 1909, Mar. 2020, doi: 10.3390/app10061909.
[12] S. W. Azumah, N. Elsayed, V. Adewopo, Z. S. Zaghloul, and C. Li, "A deep LSTM based approach for intrusion detection IoT devices network in smart home," in *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, IEEE, Jun. 2021, doi: 10.1109/wf-iot51360.2021.9596033.
[13] S. Latif *et al.*, "Intrusion detection framework for the internet of things using a dense random neural network," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6435–6444, Sep. 2022, doi: 10.1109/tii.2021.3130248.
[14] T.-T.-H. Le, Y. E. Oktian, and H. Kim, "XGBoost for imbalanced multiclass classification-based industrial internet of things intrusion detection systems," *Sustainability*, vol. 14, no. 14, p. 8707, Jul. 2022, doi: 10.3390/su14148707.
[15] B. Madhu, M. Venu Gopala Chari, R. Vankdothu, A. K. Silivery, and V. Aerranagula, "Intrusion detection models for IOT networks via deep learning approaches," *Measurement: Sensors*, vol. 25, p. 100641, Feb. 2023, doi: 10.1016/j.measen.2022.100641.
[16] R. Lazzarini, H. Tianfield, and V. Charissis, "A stacking ensemble of deep learning models for IoT intrusion detection," *Knowledge-Based Systems*, vol. 279, p. 110941, Nov. 2023, doi: 10.1016/j.knosys.2023.110941.
[17] M. Keshk, N. Koroniotis, N. Pham, N. Moustafa, B. Turnbull, and A. Y. Zomaya, "An explainable deep learning-enabled intrusion detection framework in IoT networks," *Information Sciences*, vol. 639, p. 119000, Aug. 2023, doi: 10.1016/j.ins.2023.119000.
[18] S. R. García, A. Parmisano, and M. J. Erquiaga, "IoT-23: A labeled dataset with malicious and benign IoT network traffic," Stratosphere Lab., Czech Republic, 2020, doi: 10.5281/zenodo.4743746.
[19] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, "MQTTset, a new dataset for machine learning techniques on MQTT," *Sensors*, vol. 20, no. 22, p. 6578, Nov. 2020, doi: 10.3390/s20226578.
[20] M. Srikanth Yadav. and R. Kalpana., "Data preprocessing for intrusion detection system using encoding and normalization approaches," in *2019 11th International Conference on Advanced Computing (ICoAC)*, IEEE, Dec. 2019, doi: 10.1109/icoac48765.2019.246851.
[21] A. Al Hamoud, A. Hoenig, and K. Roy, "Sentence subjectivity analysis of a political and ideological debate dataset using LSTM and BiLSTM with attention and GRU models," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 7974–7987, Nov. 2022, doi: 10.1016/j.jksuci.2022.07.014.

[22] B. Lindemann, B. Maschler, N. Sahlab, and M. Weyrich, "A survey on anomaly detection for technical systems using LSTM networks," *Computers in Industry*, vol. 131, p. 103498, Oct. 2021, doi: 10.1016/j.compind.2021.103498.

[23] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," *Expert Systems with Applications*, vol. 185, p. 115524, Dec. 2021, doi: 10.1016/j.eswa.2021.115524.

[24] R. Koniki, M. D. Ampapurapu, and P. K. Kollu, "An anomaly based network intrusion detection system using LSTM and GRU," in *2022 International Conference on Electronic Systems and Intelligent Computing (ICESIC)*, IEEE, Apr. 2022, doi: 10.1109/icesic53714.2022.9783500.

[25] E. H. Houssein, A. G. Gad, and Y. M. Wazery, "Jaya algorithm and applications: A comprehensive review," in *Metaheuristics and Optimization in Computer and Electrical Engineering*, Springer International Publishing, 2020, pp. 3–24, doi: 10.1007/978-3-030-56689-0_2.

BIOGRAPHIES OF AUTHORS

Nitu Dash currently pursuing Ph.D. under Biju Patnaik University of Technology (BPUT) and holds MCA, an M.Tech. in computer science and engineering from BPUT. An accomplished IT professional with 22 Years of academic and industry experience. Participated in over 60 national and international conferences, workshops, faculty development programs (FDP), seminars, and webinars, she remains at the forefront of cutting-edge research and industry trends. She can be contacted at email: nitudash@gmail.com.

Sujata Charkravarty a Senior Member of IEEE, is currently working as dean School of Engineering and Technology (SoET), professor computer science and engineering and CEO of Data Science and Machine Learning Research Centre, Centurion University of Technology and Management, Odisha. There are about 180 publications, eight patents published, one patent granted, one book and 25 book chapters to her credit. Under her guidance, 8 PhD scholars have been awarded, and currently supervising 10 PhD scholars. She serves as a reviewer for numerous international journals and SERB research grant proposals. She received several prestigious awards including the Jhansi Rani Laxmibai Prativa Puraskar for Technical Education and Research in 2018, the Abdul Kalam Chair Professor Award in 2021, and a Medal of Honor and Certificate of Excellence in 2022. In 2024, she is honored as the best women volunteer in the IEEE Bhubaneswar Subsection. She can be reached at sujata.chakravarty@cutm.ac.in.

Amiya Kumar Rath a distinguished educationist and researcher, assumed the role of vice chancellor at Biju Patnaik University of Technology, Odisha, in April 2023, bringing with him extensive experience from prestigious positions such as Adviser (ICT) at NAAC, Bengaluru, and professor of CSE at VSSUT, Burla. Holds B.E. in CSE from Marathwada University, MBA in systems management from Shivaji University, M.Tech in computer science from Utkal University, and Ph.D. in computer science, specializing in embedded systems, also from Utkal University. He has mentored 13 doctoral scholars, published over 150 research papers, authored 9 books, chaired international conferences, and actively participated in professional development through international conferences. He can be contacted at amiyaamiya@rediffmail.com.