International Journal of Computer and Engineering Optimization (IJCEO)
Volume 04, Issue 01, January – February (2026)

RESEARCH ARTICLE

REAL-TIME INTRUSION DETECTION IN IOT WITH DEEP LEARNING-BASED MULTI-HEAD ATTENTION BIGRU

Christal Anto V * and Ahilan A

Department of Information Technology, DMI Engineering College, Aralvaimozhi, Tamil Nadu, India
Department of Electrical and Electronic Engineering, PSN College of Engineering and Technology, Tirunelveli, Tamil Nadu, India
*Corresponding e-mail: chrisanto0124@gmail.com

Abstract The Internet of Things (IoT) links a wide range of physical devices to the Internet, facilitating cutting-edge applications in fields like the military, healthcare, agriculture, and transportation. These IoT applications have grown in popularity due to their ability to tackle real-time challenges effectively. However, despite the benefits they provide IoT systems are notably susceptible to security vulnerabilities, making them targets for a range of cyberattacks. These threats include DDoS, MiTM, sinkhole attacks, eavesdropping, and DoS attacks. In this work, a novel Real-Time Intrusion Detection in IoT with a Deep Learning-based Multi-Head Attention BiGRU (RIGRU) approach has been proposed to accurately classify IoT attacks. Data is collected from IoT sensors on network devices. It goes through data cleaning and standardization. Dingo Optimization is used to select relevant features iteratively for classification tasks. These features are then fed into a Multi-Head Attention BiGRU Network to detect MiTM, DDoS attack and normal. The proposed RIGRU approach was calculated utilizing various metrics, namely precision, f1score, recall, and accuracy. The RIGRU model advances the overall accuracy by 0.87%, 0.95%, and 0.75%, over the PCC-CNN, DIDS and GNN, respectively.

Keywords *– Deep Learning, Dingo Optimization Algorithm, Intrusion Detection, Internet of Things*

1. INTRODUCTION

The IoT has seen a significant rise in specialized applications, including healthcare, smart agriculture, transportation systems, and industrial automation, all contributing to socio-economic development [1]. IoT systems comprise actuators, numerous network-enabled devices and interconnected sensors that exchange data across both the Internet and private networks [2-3]. As the use of IoT or "smart" devices rises, the importance of Intrusion Detection Systems (IDS) as a key tool to protect against various security threats becomes paramount [4]. At the core of IoT is the integration of smart sensors that link physical objects via a unified platform [5].

These smart sensors are designed to operate with minimal or no human interaction [6-7]. Physical objects and smart devices communicate and collaborate through specific addressing schemes [8]. The term "Smart," when applied to various physical objects, signifies the integration of IoT technology [9]. However, current IDS used in IoT environments are insufficient to address the complexities and diversity of these systems [10]. Developing effective IDS for IoT is a challenging and demanding task, necessitating a thorough and comprehensive analysis in this field [11].

Large quantities of resources are essential for IDS which restricts their usage on IoT devices [12]. Common attacks on IoT networks include DDoS attacks, where attackers overwhelm the network with traffic to render it inaccessible, and unauthorized access attempts that exploit weak authentication mechanisms or unsecured interfaces. However, the increasing usage of smart devices and systems has also brought about new security challenges [13]. In this research, a novel RIGRU approach has been proposed to accurately classify IoT attacks. The key objectives of the RIGRU approach are as follows.

- Initially, the raw data is collected from IoT sensors deployed on network devices. This raw data undergoes pre-processing, including data cleaning to remove inconsistencies and errors, and stemming/lemmatization to standardize features.
- The Dingo Optimization is used for selecting relevant features and optimizing the feature set iteratively for classification tasks.
- These features are then fed into a Multi-Head Attention BiGRU Network for real-time intrusion detection into categories, namely DDoS, MiTM, and normal.

The remaining section of the proposed approach are detailed below. The literature review is covered in detail in Section 2. The RIGRU methodology is described in Section

3. The outcome and discussion are covered in Section 4. The conclusion is covered in Section 5.

2. LITERATURE SURVEY

Many recent works on Intrusion Detection in IoT have made use of DL methods. The following section provides an overview of a few recent studies.

In 2023, Awajan, A., [14] proposed an IDS based on deep learning for IoT devices. This smart system uses a four-layer deep Fully Connected framework architecture to detect harmful traffic that may initiate assaults on linked IoT devices. The DL-based IDS demonstrates robust performance in identifying different types of attacks, achieving an accuracy of 93.74% for both evaluated and intrusions. Additionally, it records an average precision, F1-score and recall of 93.71%, 93.47%, and 93.82%.

In 2023, Bhavsar, M., et al., [15] suggested a DL-based IDS known as PCC-CNN to detect network anomalies. The KNN and CART approaches demonstrated similar accuracy rates of 98% and 99% across three datasets. However, the PCC-CNN model outperformed both, achieving a low misclassification rate of 0.001 and an accuracy of 99.89%.

In 2023, Altunay, H.C. and Albayrak, Z., [16] proposed three different methods to identify breaches in the Industrial IoT network using deep learning (DL) frameworks: CNN, a combined CNN+LSTM, and an LSTM model. The CNN+LSTM method achieved an accuracy of 92. 9% on the UNSW-NB15 dataset. For the X-IIoTID dataset, the same method displayed even greater accuracy, reaching 99. 80%.

In 2023, Bakhsh, S.A., et al., [17] proposed a deep learning-based Intrusion Detection System employing Feed Forward Neural Networks, Long Short-Term Memory networks, and Random Neural Networks to protect Internet of Things networks from cyber threats. This approach surpasses the existing approach when tested on the CIC-IoT22 dataset, with FFNN reaching an accuracy of 99. 93%, LSTM attaining 99. 85%, and RandNN achieving 96. 42% in identifying intrusions.

In 2023, Madhu, B., et al., [18] developed a DL framework called DIDS to handle unknown attacks and higher throughput with a low false alarm rate in large networks. After being evaluated with standard algorithms, the proposed algorithm showed earlier attack detection, reduced computational time, and achieved 99% accuracy.

In 2023, Sanju, P., [19] suggested a metaheuristic- DL strategy to advance intrusion detection in IoT systems. All things considered, the suggested study offers a viable way to enhance IDS in IoT, and it may provide a starting point for further studies in this area.

In 2023, Altaf, T., et al., [20] developed a GNN-based network IDS that optimizes the likelihood of incorporating structural elements of both legitimate and malevolent network traffic. In the datasets under consideration, the suggested model offers a 2% to 5% gain in precision, accuracy, recall, and F1, while requiring less training time and approach size. Across the four benchmark datasets, the suggested framework has demonstrated excellent accuracy.

3. PROPOSED METHODOLOGY

In this section, a RIGRU technique has been proposed to accurately classify IoT attacks. Initially, the raw data is collected from IoT sensors deployed on network devices. This raw data undergoes pre-processing, including data cleaning to remove inconsistencies and errors, and stemming/lemmatization to standardize features. The Dingo Optimization is used for selecting relevant features and optimizing the feature set iteratively for classification tasks. The selected features are then fed into a DL architecture comprising a Multi-Head Attention BiGRU Network. This network leverages multi-head attention for simultaneous feature focus and Bi-GRU for sequential data analysis, enabling sophisticated pattern recognition and classification of intrusion into categories such as DDoS, MiTM, and normal in real-time, empowering timely security responses. Figure 1 demonstrates the overall flow of suggested RIGRU,

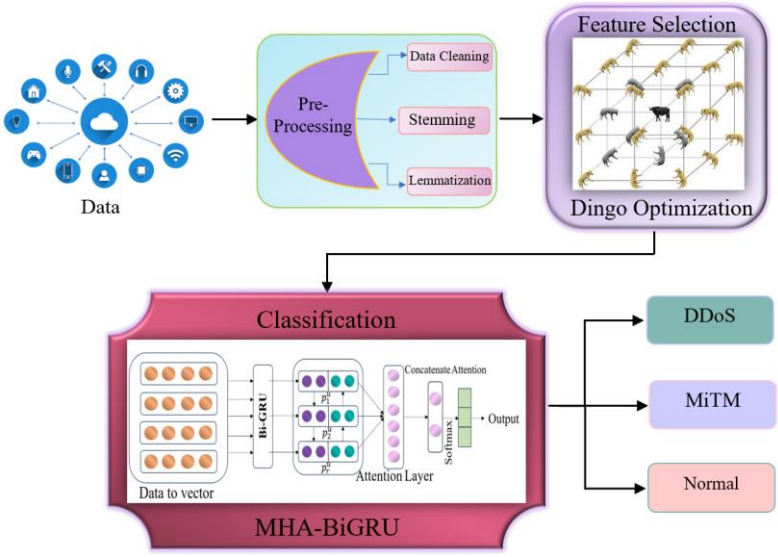

Figure 1. Schematic Representation of the Proposed RIGRU Framework

3.1 Data Collection

IoT sensors are embedded in network-connected devices and collect diverse data like temperature, motion, and environmental conditions. This unprocessed information supports immediate tracking, anticipatory upkeep, and decisions based on data across different industries.

3.2 Pre-Processing

Preprocessing plays an essential role in analyzing data, consisting of various methods that transform unprocessed data into a tidy, organized format suitable for examination. The preprocessing process typically includes the following steps:

Data Cleaning: It is the method of recognizing and handling null values, outliers, and errors in the dataset. This includes approaches such as imputation to replace missing values with estimated ones, filtering out outliers that may skew the analysis, and correcting errors in the data to ensure its quality and accuracy.

Stemming: Stemming is the method of reducing words to their simplest form by removing prefixes and suffixes. It utilizes heuristic guidelines to eliminate suffixes according to a set list of derivational affixes. For example, the terms "trouble," "troubled," and "troubles" could all be simplified to the base form "troubl. " Although this technique may occasionally produce words that are misspelled or not fully formed, it is quicker and easier than lemmatization, which requires a deeper linguistic examination to provide the accurate base form of a word.

Lemmatization: Lemmatization is the procedure of dropping a word to its most basic form, or replacing its suffix. Lemma is always a meaningful word, unlike a stemmed word. Lemmatization is also a method of integrating many words into one.

3.3 Dingo Optimization

Improved Dingo Optimization's optimization model is made for feature selection, which optimizes data and offers pertinent features. Finding the best real-world system solutions is a difficult task. because there are multiple excellent solutions available. (1) – (3) strengthen the persecution, scavenging, group assault, and survival.

$$\partial = (\mathrm{BF} - \mathrm{WF}) \tag{1}$$

$$\mathrm{Y} = \frac{\mathrm{mean(D(F-WF))}}{\mathrm{mean(D(F-BF))}} \tag{2}$$

$$\mathrm{UB} = \partial \times \left(\frac{\mathrm{q}}{\mathrm{Y}}\right) \tag{3}$$

The best and worst objective functions are denoted by the terms BF and WF, respectively, in this instance. Fitness is represented by F, distance by D, generality by Y and q, and arbitrarily by UB, which is a number between 0 and 1.

Step 1 - Group Attack: Dingoes hunt in packs, encircling their target using their tracking abilities. There is an equation that represents this.

$$\overrightarrow{\mathrm{D_b}}(\mathrm{l}+1) = \propto_1 \sum_{\mathrm{v}=1}^{\mathrm{ui}} \left(\frac{\overrightarrow{\emptyset \mathrm{k(d)}} - \mathrm{d}\vec{\mathrm{p}}(\mathrm{l})}{\mathrm{ui}}\right) - \vec{\mathrm{d}} \times (\mathrm{l}) \tag{4}$$

The expression $\overrightarrow{\mathrm{D_b}}(\mathrm{l}+1)$ represents the position of the feature, ui is a random number, $\mathrm{d}\vec{\mathrm{p}}$ represents the best iteration constructed utilizing the preceding iteration, $\vec{\mathrm{d}} \times(\mathrm{z})$ tends the search agent, $\emptyset$k(d) is the sub-set provided for all search agents and $\propto_1$ is a randomly generated number equally produced and displayed in a range of [-2, 2]. It has been updated in Equation (4).

Step 2 - Persecution: Dingoes frequently hunt tiny animals until they are within reach. Equation (5) provides an illustration of this behaviour.

$$\vec{\mathrm{d}}_\mathrm{p}(\mathrm{l}+1) = \vec{\mathrm{d}} \times (\mathrm{l}) + \propto_1 + \mathrm{f}^{\propto_2} \times (\vec{\mathrm{d}}_{\mathrm{q}_1}(\mathrm{l}) - \vec{\mathrm{d}}_\mathrm{p}(\mathrm{l}) \tag{5}$$

Here, the term $\vec{\mathrm{d}}_\mathrm{p}(\mathrm{l}+1)$ denotes a dingo movement within the search engine $\vec{\mathrm{d}}_\mathrm{p}(\mathrm{l})$.The derived random variable falls within [−1, 1] is represented as $\propto_2$, arbitrary number q_1 is updated in Equation (3) and $\vec{\mathrm{d}}_{\mathrm{q}_1}(\mathrm{l})$ represents the search agent for the qth interval.

Step 3 - Scavenger: Equation (6) describes the scavenging behavior of dingoes, which is thought of as an activity when they locate their meal.

$$\vec{\mathrm{d}}_\mathrm{p}(\mathrm{l}+1) = \frac{1}{2}[\mathrm{f}^{\propto_2} \times \vec{\mathrm{d}}_{\mathrm{q}_1}(\mathrm{l}) - (-1)^\sigma \times \vec{\mathrm{d}}_\mathrm{p}(\mathrm{l})] \tag{6}$$

The binary number, represented by the symbol σ, is produced spontaneously and is upgraded in Equation (3).

Step 4: The dingo's survival rate is given in Equation (7).

$$\mathrm{surv(D)} = \frac{\mathrm{BF-F(D)}}{\mathrm{BF-WF}} \tag{7}$$

In the current generation, the least fitness function is represented as WF and BF represents the highest level of fitness. Fs(D) is the fitness value provided for Dth search features and also Equation (8) determines the minimal survival rate.

$$\vec{\mathrm{d}}_\mathrm{p}(\mathrm{l}+1) = \vec{\mathrm{d}} \times (\mathrm{l}) + \frac{1}{2}[\vec{\mathrm{d}}_{\mathrm{q}_1}(\mathrm{l}) - (-1)^\sigma \times \vec{\mathrm{d}}_{\mathrm{q}_2}(\mathrm{l})] \tag{8}$$

The chosen search agent for the numerical value q_1 and q_2 is stated as $\vec{\mathrm{d}}_{\mathrm{q}_1}(\mathrm{l})$ and $\vec{\mathrm{d}}_{\mathrm{q}_2}(\mathrm{l})$, correspondingly, and Equation (4) updates the random integers. Here, the lowest survival rate is indicated as $\vec{\mathrm{d}}_\mathrm{p}(\mathrm{l})$.

3.4 Intrusion Detection

The selected features are then fed into a DL architecture comprising a Multi-Head Attention BiGRU Network. This network leverages multi-head attention for simultaneous feature focus and Bi-GRU for sequential data analysis, enabling sophisticated pattern recognition and classification of intrusion into categories such as DDoS, MiTM, and normal in real-time, empowering timely security responses.

BiGRU Layer: GRU networks fall under the category of recurrent neural networks (RNNs). To tackle the problem where standard RNNs update their memory in increments and face challenges with gradient decay, they are built on RNN principles. GRU can be seen

as a streamlined version of LSTM neural networks, allowing for easier determination while still preserving the effectiveness of LSTM models. LSTM networks utilize input and forget gates. The vector p_r denotes the input, m_{r-1} reflects the hidden state at the previous time step $r-1$, and m_r indicates the output vector of the current GRU. During step r, both p_r and m_{r-1} are fed into the GRU, which produces the output q_r. Equations 11, 12, 13, and 14 illustrate how q_r is defined:

$$b_r = \sigma(Z_b p_r + S_b m_{r-1} + f_b) \quad (11)$$

$$d_r = \sigma(Z_d p_r + S_d m_{r-1} + f_d) \quad (12)$$

$$\tilde{m}_r = \tanh\left(Z_m p_r + S_m (m_{r-1} \otimes b_r) + f_r\right) \quad (13)$$

$$m_r = (1 - d_r) \otimes m_{r-1} + d_r \otimes \tilde{m}_r \quad (14)$$

Where σ denotes the Sigmoid function, which assists GRU networks in managing memory by enabling them to retain or discard information. The elementwise multiplication is represented by ⊗, and the reset and update gates are indicated as d_r and b_r, respectively. Moreover, $\tilde{m}_r$ indicates the presumed state of the candidate at time r. The architecture of BiGRU features two hidden layers: one that processes data from the beginning to the end and another that processes it in reverse. Each data sample is inputted into both the forward and backward GRU networks, yielding a pair of symmetric hidden layer state vectors. By symmetrically combining these state vectors, we can derive an overall encoded representation of the input text, as illustrated below:

$$M_r = [\overrightarrow{M_r} \oplus \overleftarrow{M_r}] \quad (15)$$

The attention process associates a query (S) with a collection of key and value pairs. The assessment of the attention mechanism is divided into three steps. To begin with, a similarity function $n(S, L_i)$ is developed to assess how similar S is to each L. As illustrated in equation (12), there are various similarity functions, such as dot, general, and concat, that yield a corresponding attention score c:

$$n(S, L_i) = \begin{cases} S^D L_i & \text{Dot} \\ S^D V L_i & \text{General} \\ V[S; L_i] & \text{Concat} \end{cases} \quad (16)$$

Following that, apply a softmax function to derive the weighted vector, β. Finally, the combination of β and E produces the context vector r:

$$\beta_i = \text{softmax}(n_i) \quad (17)$$

$$r = \sum_{d=1}^{m} \beta_i W_i \quad (18)$$

The sizes of the queries (S) and keys (L) are represented by x_l, while the size of the values (E) is indicated by x_e. The scaled dot-product attention technique determines attention scores through the following formula:

$$\text{Atten}(S, L, E) = \text{Softmax}\left(\frac{SL^D}{\sqrt{x_l}}\right)E \quad (19)$$

Multi-head (MH) attention consists of several attention layers that function together, enabling the framework to focus on details from different appearance subspaces at various locations at the same time. In the attention mechanism, the queries, keys, and value vectors are processed through distinct linear projections before they are evaluated for importance through the scaled dot product attention method. The operations performed in the attention mechanism are called "heads" after every "q" operation, and the vectors are combined to form a single vector.

$$hea_i = \text{Atten}(SV_i^S, LV_i^L, EV_i^E) \quad (20)$$

$$MH(S, L, E) = \text{Conca}(hea_1, \cdots, hea_q)V \quad (21)$$

4. RESULT AND DISCUSSION

The experimental findings of the RIGRU approach are examined in this section. The performance is examined using several assessment metrics. The effectiveness of the RIGRU framework is compared to that of PCC-CNN, DIDS, and GNN in relation to recall, precision, accuracy, and F1 score.

4.1 Performance Metrics

The suggested method is evaluated using performance measures, including specificity, precision, mean squared error (MSE), recall, and accuracy.

$$\textbf{\textit{Accuracy}} = \frac{TrP + TrN}{FalN + TrP + FalP + TrN} \quad (22)$$

$$\textbf{\textit{F1 score}} = 2 \times \frac{PR.RC}{PR + RC} \quad (23)$$

$$\textbf{\textit{Precision}} = \frac{TrP}{TrP + FalN} \quad (24)$$

$$\textbf{\textit{Recall}} = \frac{TrP}{TrP + FalN} \quad (25)$$

$$\textbf{\textit{Mean Squared Error}}\ (MSE) = \frac{1}{M}\sum_{l=1}^{M}(q_{input} - q_{output})^2 \quad (26)$$

4.2 Comparison Analysis

This part contains the simulation to assess the efficiency of the suggested RIGRU model. The methods PCC-CNN, DIDS, and GNN are evaluated against the suggested technique.

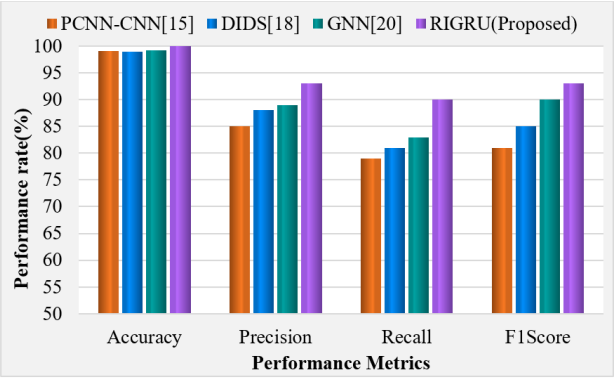

Figure 2. Performance Comparison

Figure 2 demonstrates the effectiveness of recall, accuracy, precision, and f1score between the new and current methods. Each classification method is assessed based on its f1score (F1S), recall (RC), precision (PR), accuracy (AC), and recall (RC) using the TrP, TrN, FalP, and FalN metrics.

The f1score, recall, precision, and accuracy for the introduced RIGRU approach are 99. 95%, 93%, 90%, and 92%, respectively, surpassing the performance of existing methods.

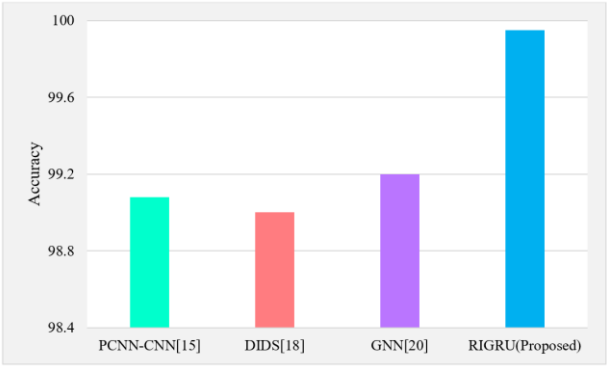

Figure 3. Accuracy Performance

Figure 3 compares the accuracy of four different methods: PCC-CNN [15], DIDS [18] and GNN [20] and a RIGRU approach. The RIGRU approach illustrate a strong performance with an accuracy of 99.95%. The proposed approaches achieve 99.95% accuracy, over PCC-CNN [15], DIDS [18] and GNN [20] which achieve 99.08%, 99%, and 99.2% respectively.

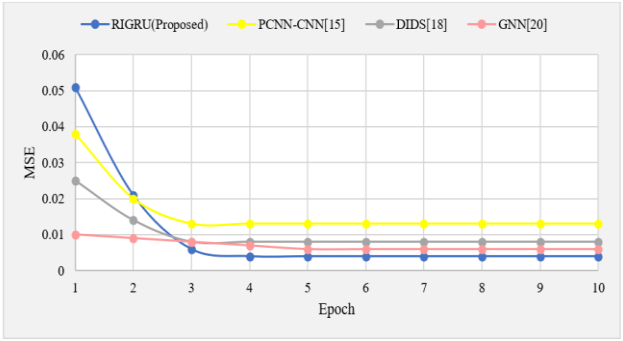

Figure 4. Error Comparison

Figure 4 illustrates a comparison of the error rates between the new RIGRU method and the current methods. The error rate for the proposed method stands at 0. 004%, which is lower than the errors for the existing PCC-CNN [15], DIDS [18], and GNN [20] methods, which have error rates of 0. 013%, 0. 008%, and 0. 006% respectively.

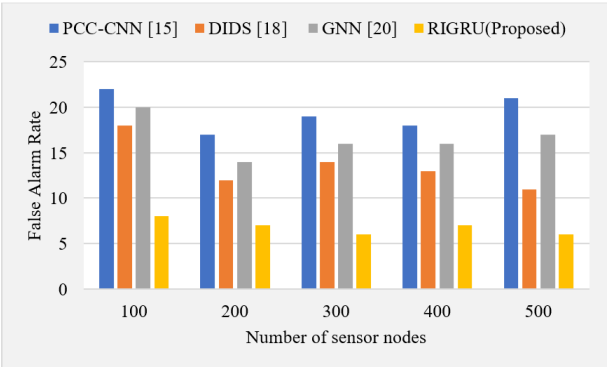

Figure 5. False alarm rate Comparison

Figure 5 illustrates the false alarm rates of different methods in IoT systems that use between 100 and 500 sensor nodes. The RIGRU method shows a reduced false alarm rate when compared to the current PCC-CNN [15], DIDS [18], and GNN [20]. This suggests that RIGRU is better at minimizing false alarms in varying sizes of IoT implementations.

5. CONCLUSION

In this paper, a RIGRU technique is proposed for classifying intrusion accurately in IoT. Initially, the data is collected from IoT sensors that are installed on network devices. This data goes through a process of cleaning and standardization. To perform classification tasks, Dingo Optimization is used to select relevant features iteratively. These selected features are then utilized as input for a Multi-Head Attention BiGRU Network, which performs real-time intrusion classification into different categories such as DDoS, MiTM, and normal attacks. Various assessments, including recall, precision, sensitivity, accuracy, and specificity, were utilized to evaluate the suggested RIGRU model. The proposed RIGRU techniques achieve 99.95% accuracy than PCC-CNN, DIDS and GNN which obtains 99.08%, 99%, and 99.2%. In comparison to PCC-CNN, DIDS, and GNN, the RIGRU model improves overall accuracy by 0.87%, 0.95%, and 0.75%, respectively. The future goal is to create an IDS for protecting IoT networks against routing attacks and improving security.

CONFLICTS OF INTEREST

The authors declare that there is no conflict of interest.

FUNDING STATEMENT

Authors did not receive any funding.

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] R. Yang, H. He, Y. Xu, B. Xin, Y. Wang, Y. Qu and W. Zhang, “Efficient intrusion detection toward IoT networks using cloud–edge collaboration (Journal Article)”, Computer Networks, vol. 228, pp. 109724, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] A. Rizzardi, S. Sicari and A.C. Porisini, “Deep reinforcement learning for intrusion detection in Internet of Things: Best practices, lessons learnt, and open challenges (Journal Article)”, Computer Networks, vol. 236, pp. 110016, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] R. Lazzarini, H. Tianfield and V. Charissis, “A stacking ensemble of deep learning models for IoT intrusion detection (Journal Article)”, Knowledge-Based Systems, vol. 279, pp. 110941, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] M. Mohy-Eddine, A. Guezzaz, S. Benkirane, M. Azrour and Y. Farhaoui, “An ensemble learning based intrusion detection model for industrial IoT security (Journal Article)”, Big Data Mining and Analytics, vol. 6, no. 3, pp. 273–287, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] C. Hazman, A. Guezzaz, S. Benkirane and M. Azrour, “lIDS-SIoEL: intrusion detection framework for IoT-based smart environments security using ensemble learning (Journal

Article)", Cluster Computing, vol. 26, no. 6, pp. 4069–4083, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[6] G.P. Fernando, A.A.H. Brayan, A.M. Florina, C.B. Liliana, A.M. Héctor-Gabriel and T.S. Reinel, "Enhancing intrusion detection in IoT communications through ML model generalization with a new dataset (IDSAI) (Journal Article)", IEEE Access, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[7] O. Elnakib, E. Shaaban, M. Mahmoud and K. Emara, "EIDM: Deep learning model for IoT intrusion detection systems (Journal Article)", The Journal of Supercomputing, vol. 79, no. 12, pp. 13241–13261, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[8] 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 (Journal Article)", Information Sciences, vol. 639, pp. 119000, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[9] N. Tekin, A. Acar, A. Aris, A.S. Uluagac and V.C. Gungor, "Energy consumption of on-device machine learning models for IoT intrusion detection (Journal Article)", Internet of Things, vol. 21, pp. 100670, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[10] N. Saran and N. Kesswani, "A comparative study of supervised machine learning classifiers for intrusion detection in Internet of Things (Journal Article)", Procedia Computer Science, vol. 218, pp. 2049–2057, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[11] O.A. Alghanam, W. Almobaideen, M. Saadeh and O. Adwan, "An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning (Journal Article)", Expert Systems with Applications, vol. 213, pp. 118745, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[12] A. Mahalingam, G. Perumal, G. Subburayalu, M. Albathan, A. Altameem, R.S. Almakki, A. Hussain and Q. Abbas, "ROAST-IoT: a novel range-optimized attention convolutional scattered technique for intrusion detection in IoT networks (Journal Article)", Sensors, vol. 23, no. 19, pp. 8044, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[13] R.A. Elsayed, R.A. Hamada, M.I. Abdalla and S.A. Elsaid, "Securing IoT and SDN systems using deep-learning based automatic intrusion detection (Journal Article)", Ain Shams Engineering Journal, vol. 14, no. 10, pp. 102211, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[14] A. Awajan, "A novel deep learning-based intrusion detection system for IoT networks (Journal Article)", Computers, vol. 12, no. 2, pp. 34, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[15] M. Bhavsar, K. Roy, J. Kelly and O. Olusola, "Anomaly-based intrusion detection system for IoT application (Journal Article)", Discover Internet of Things, vol. 3, no. 1, pp. 5, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[16] H.C. Altunay and Z. Albayrak, "A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks (Journal Article)", Engineering Science and Technology, an International Journal, vol. 38, pp. 101322, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[17] S.A. Bakhsh, M.A. Khan, F. Ahmed, M.S. Alshehri, H. Ali and J. Ahmad, "Enhancing IoT network security through deep learning-powered intrusion detection system (Journal Article)", Internet of Things, vol. 24, pp. 100936, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[18] B. Madhu, M.V.G. Chari, R. Vankdothu, A.K. Silivery and V. Aerranagula, "Intrusion detection models for IoT networks via deep learning approaches (Journal Article)", Measurement: Sensors, vol. 25, pp. 100641, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[19] P. Sanju, "Enhancing intrusion detection in IoT systems: A hybrid metaheuristics-deep learning approach with ensemble of recurrent neural networks (Journal Article)", Journal of Engineering Research, vol. 11, no. 4, pp. 356–361, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[20] T. Altaf, X. Wang, W. Ni, G. Yu, R.P. Liu and R. Braun, "A new concatenated multigraph neural network for IoT intrusion detection (Journal Article)", Internet of Things, vol. 22, pp. 100818, 2023. [CrossRef] [Google Scholar] [Publisher Link]

AUTHORS

Christal Anto V is an Assistant Professor in the Department of Information Technology at DMI Engineering College, Aralvaimozhi, India, where he has been serving since 2019. With expertise in information technology and related fields, his research interests include. He has contributed to numerous publications and is committed to advancing innovative solutions in IT education and applications.

Ahilan A received Ph.D. from Anna University, India, and working as an Associate Professor in the Department of Electronics and Communication Engineering at PSN College of Engineering and Technology, India. His area of interest includes FPGA prototyping, Computer vision, the Internet of Things, Cloud Computing in Medical, biometrics, and automation applications. TCS, Bangalore, where he has guided many computer vision projects and Bluetooth Low Energy projects. Meanwhile, special Guest lectures, Practical workshops, Hands-on programming in MATLAB, Verilog, and python at various technical institutions around India.

Arrived: 24.01.2026
Accepted: 20.02.2026