Journal of Smart Internet of Things (JSIoT)
VOL 2024, No.01 | 60-76 | 2024
DOI: 10.2478/jsiot-2024-0005

Article

Development of Hybrid Intrusion Detection Systems for IoT Enabled Devices Utilizing Resource Constraint Learning Frameworks

Rachana. P [1], Mamatha Talakoti [2], V. Surya Narayana Reddy [3]

[1] Department of Computer Science and Engineering, Alvas Institute of Engineering and Technology, Mangalore, India
[2] Department of Computer Science and Engineering, Sreenidhi Institute Of Science and Technology, Hyderabad, India
[3] Department of Computer Science and Engineering, Bvrit Hyderabad College of Engineering for Women, Hyderabad, India

*Corresponding author: Rachana. P (email: rachana@aiet.org.in)

Received 11 Feb. 2024; Accepted 01 May 2021; Published 15 June 2024

Abstract: The increasing prevalence of IoT-enabled devices in resource-constrained environments has made them vulnerable to various network intrusions and security threats. Traditional intrusion detection systems often struggle with capturing complex temporal dependencies in network traffic and balancing computational efficiency with accuracy, limiting their effectiveness in IoT networks. To address this challenge, we recommend a novel hybrid intrusion detection framework based on a hyperparameter-tuned LSTM-GRU model. The proposed system leverages the strengths of both architectures to efficiently analyse sequential patterns and temporal dependencies in network traffic while maintaining computational feasibility for IoT devices. A grid search method is employed to elevate hyperparameters like learning rate, batch size, and the number of layers, ensuring the model's performance is fine-tuned for the intrusion detection task. Experimentation outcomes on the CICIDS 2017 dataset depict the model's robustness, achieving 95% accuracy, 96% precision, 95% recall, 96% F1-score and 95% specificity, significantly outperforming existing approaches. This hybrid architecture offers a practical solution for real-time intrusion detection in IoT systems, balancing computational efficiency with high detection accuracy.

Keywords: IoT security, intrusion detection, LSTM-GRU hybrid model, grid search, resource-constrained environments, CICIDS 2017 dataset, temporal dependency.

1. Introduction

The rapid adoption of IoT devices in varied sectors, consist healthcare, smart homes, manufacturing, and transportation, has transformed the way we interact with technology. IoT devices are integrated into everyday objects, allowing them to preserve, exchange, and process data, enabling intelligent systems that optimize efficiency, convenience, and user experience [1]. However, the widespread deployment of these devices also introduces significant security challenges, particularly in resource-constrained environments.

Intrusion detection systems (IDS) play a critical role in identifying and mitigating security breaches by monitoring network traffic and detecting abnormal or malicious activities [2,3]. In traditional computing environments, IDS technologies like signature-based and anomaly-based detection approaches are widely used [4]. However, IoT systems require specialized intrusion detection methods due to their unique characteristics, including limited hardware resources, real-time operational demands, and the heterogeneity of devices and communication protocols. To address these challenges, researchers have been exploring the development of Hybrid Intrusion Detection Systems (HIDS) that combine multiple detection techniques to enhance the system's performance while minimizing resource consumption [6].

Among the various techniques used for intrusion detection, ML has gained significant attention in recent years. ML models [7], especially DL architectures, are well-suited for detecting complex and evolving attack patterns in network traffic. Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRU) are two popular recurrent neural networks (RNNs) that have shown promising results in sequence-based tasks like time series prediction and anomaly detection [8]. These models can discern temporal interrelations and sequential patterns in data, which makes them ideal candidates for detecting intrusions in dynamic IoT environments where network traffic exhibits temporal correlations [9].

The hybridization of LSTM and GRU models offers a feasible resolution to overcome the constraints of each individual model [20, 21]. By infusing the strengths of both architectures, a hybrid LSTM-GRU model can leverage the advantages of each method, such as the ability of LSTMs to handle long-range dependencies and the efficiency of GRUs in capturing shorter-term dependencies. This hybrid model can improve the overall detection accuracy while maintaining computational efficiency, reaching it more adequate for resource-constrained IoT environments [22]. However, the performance of the hybrid approach heavily relies on the appropriate selection of hyperparameters, including the batch size, learning rate, quantity of layers, and other architectural configurations.

By evaluating multiple combinations of hyperparameters, grid search allows us to source configuration that yields the best output detection accuracy and computational power [23-25]. In the context of our proposed HIDS framework, grid search will be used to calibrate the hyperparameters of the hybrid LSTM-GRU model, ensuring that the intrusion detection system operates efficiently while maintaining high detection accuracy in IoT environments [26].

Traditional IDS methods often struggle to scale with the increasing amount of network traffic in modern IoT systems. In contrast, machine learning-based intrusion detection systems, particularly those using DL [27]. However, even with machine learning models, IoT networks present additional challenges, such as the heterogeneity of devices, varying communication protocols, and real-time detection requirements. The hybrid LSTM-GRU model offers a solution by being able to process sequential data in real-time, capturing both short- and long-term patterns in network traffic, which improves the system's ability to recognize subtle and evolving attacks [28-30]. The hybrid deep learning model, when hyperparameter-tuned, offers a balanced interchange among detection accuracy and computational efficiency.

The development of an effective Hybrid IDS for IoT networks is a challenging but critical task. By leveraging our proposed model, we aim to create a system that balances high detection accuracy

with computational efficacy. This scheme not only elevate the security of IoT devices but also ensures that the intrusion detection system can operate in real-time on resource-constrained devices. Our research contributes to the growing field of IoT security by proposing a scalable, efficient, and robust solution for intrusion detection in IoT environments.

1.1 Contribution of the Research:

1. The research introduces a novel methodology that integrates the Hyperparameter tuned grid search technique with a LSTM-GRU hybrid model to enhance intrusion detection in IoT networks.
2. The proposed algorithm is thoroughly compared against various existing artificial intelligence-based intrusion detection systems.
3. Comprehensive experiments are conducted using the CICIDS 2017 dataset, a widely recognized benchmark for intrusion detection, with performance metrics like accuracy, precision, recall, F1-score, and specificity calculated to examine the model's effectiveness.

1.2 Structure of the Paper:

This manuscript is structured as pursues: Section 2 evaluates the related work and relevant studies conducted by various researchers in the domain of intrusion detection. Section 3 provides an overview of the LSTM, GRU, Hyperparameter tuned Grid Search techniques, along with a described explanation of the proposed LSTM-GRU hybrid model architecture. Section 4 describes the CICIDS 2017 dataset, outlines the experimental setup, and presents a comprehensive analysis of the results. Atlast, Section 5 wraps up the study and highlights potential directions for future research and development.

2. Related Works

Shahid et al. (2024) [11] propose a Hybrid Intrusion Detection System (IDS) built for for IoT networks utilizing ML and DL approach. Their work focuses on the Routing Protocol for Low-Power and Lossy Networks (RPL), which is inclined to various attacks due to IoT devices' inherent constraints, such as limited memory and processing power. By analyzing network traffic features using statistical graphs, they evaluate multiple machine learning models and deep learning architectures. However, a notable drawback is the limited consideration of scalability and adaptability to dynamic IoT environments, where real-time attack detection and system overhead remain critical challenges.

Yaras and Dener (2024) [12] proposed an IoT-based IDS leveraging a hybrid DL algorithm to address the increasing cyber threats, particularly DDoS attacks, in IoT networks. The study highlights that traditional methods struggle to analyze network traffic efficiently because of large amount data generated by numerous sensor nodes. The researchers employed PySpark with Apache Spark in Google Colab for big data analysis, utilizing the CICIoT2023 and TON_IoT datasets. They reduced the features in the datasets utilizing the correlation method to include only significant features. A hybrid model combining one-dimensional CNN and LSTM was developed and achieved impressive accuracy rates. Additionally, the energy consumption and computational cost associated with implementing such hybrid models in resource-constrained IoT devices were not addressed.

Sajid, M., Malik, K.R., Almogren, A., et al. (2024) [13] proposed a hybrid approach for intrusion detection utilizing ML and DL approach. The study highlights the challenges in maintaining network security as data volumes surge due to advancements in cloud computing, IoT, and automotive networks. The findings demonstrated a high detection rate, good accuracy, and a low False Acceptance Rate (FAR), examining the model's effectiveness. However, the approach

has a drawback in terms of computational complexity due to the integration of multiple feature selection algorithms and deep learning models, potentially limiting its scalability for real-time applications in Constrained-resource contexts.

Almotairi et al. (2024) [14] proposed an innovative approach to enhance intrusion detection in IoT networks by employing a diverse ML-based stack classifier model. The study utilized the K-Best algorithm for feature selection, identifying the top 15 critical features from the Ton IoT dataset, and incorporated ensemble modeling to improve classification metrics. The ensemble model combines the strengths of various traditional machine learning models to achieve exceptional performance compared to individual models. However, a notable drawback of this approach is its reliance on computationally intensive processes for feature selection and model training, which may not be feasible for real-time applications in resource-constrained IoT devices.

Walling and Lodh (2024) [15] proposed a novel feature selection methodology for anomaly-based Network IDS in IoT security using machine learning and statistical techniques. Their approach utilizes 1-way ANOVA and Pearson correlation coefficient as filter-based methods to extract relevant features from the datasets. By leveraging union and intersection principles of set theory, they identified optimal features for intrusion detection. Despite these promising results, the methodology may face scalability challenges in highly dynamic IoT environments due to the computational overhead of the feature selection process and the inherent diversity of IoT devices, which may limit its real-time applicability.

Meliboyev et al. (2024) [16] conducted a study on the development of an IoT network IDS using ML techniques. The authors highlight that the rapid proliferation of IoT devices has revolutionized industries by offering smart and automated solutions but has also brought about substantial security concerns, particularly in network intrusion. Their research focuses on analyzing network traffic to identify anomalous patterns, proposing a machine learning-based IDS as a robust and scalable solution. Despite its promise, the proposed system primarily addresses anomaly detection and does not delve deeply into the complexities of real-time detection in highly dynamic IoT environments, which could limit its effectiveness in certain use cases.

Al Sawafi et al. (2023) [17] proposed a hybrid DL-based IDS for routing protocol for low-power and lossy networks (RPL) in IoT environments. The study introduced the IoTR-DS dataset, designed for IoT networks using the RPL protocol. The proposed system achieved a detection accuracy of 98% with an F1-score of 92% for pre-programmed (known) attacks and an F1-score of 87% for untrained (unknown) attack behaviours. However, a significant drawback is that the system's performance is dataset-dependent, and its generalizability to other IoT protocols or real-world IoT network conditions remains unaddressed. Additionally, the approach may face challenges in resource-constrained IoT environments due to the computational overhead associated with hybrid deep learning models.

Awajan et al. (2023) [18] proposed a novel DL-based IDS for IoT networks, addressing the increasing threat of cyber-attacks on IoT devices and communication channels. The system leverages a four-layer fully connected deep learning architecture to detect suspicious traffic and supports a protocol-independent design to simplify deployment across diverse IoT environments. Despite its promising results, the study not dealwith the computational complexity or resource requirements of the system, which are critical factors for real-time deployment in resource-constrained IoT devices. Additionally, its generalizability to more complex or evolving intrusion patterns remains unexplored.

Singh et al. (2022) [19] proposed an Edge-based Hybrid IDS (EHIDF) tailored for Mobile Edge Computing (MEC) circumstances to address security challenges associated with edge network architectures. MEC, with its rapid delivery approach, attracts users but remains vulnerable to Internet-based attacks. Traditional intrusion detection models primarily detect known attacks and

exhibit low efficiency in real-time traffic monitoring, often failing to identify new, unknown threats. The EHIDF demonstrates an improvement in detection accuracy by 10.78% and a reduction in FAR by 93% compared to previous works. Despite its promising results, the study highlights limitations like the need for further optimization of detection modules for more complex attack scenarios and scalability issues in broader MEC deployments.

Smys et al. (2020) [20] presents a hybrid IDS for IoT networks to deal with the increasing security issues posed by the growing number of IoT devices. The recommended approach leverages a CNN-based model to recognize and mitigate these security threats effectively. Experimental results indicate that the recommended approach outperforms traditional ML and DL models in terms of sensitivity to attacks, tailoring it for various IoT implementations.

3. Proposed Methodology

The Recommended Hybrid IDS (HIDS) for IoT environments begins with Data Preprocessing, including Feature Scaling and Label Encoding, to prepare network traffic data. Using Grid Search Hyperparameter Tuning, key parameters like learning rate, batch size, and LSTM-GRU units are optimized for enhanced performance. The Hybrid LSTM-GRU Model captures temporal dependencies, ensuring accurate anomaly detection. Finally, the Output Detection Layer classifies traffic as normal or anomalous, achieving high accuracy and efficiency for real-time IoT intrusion detection.

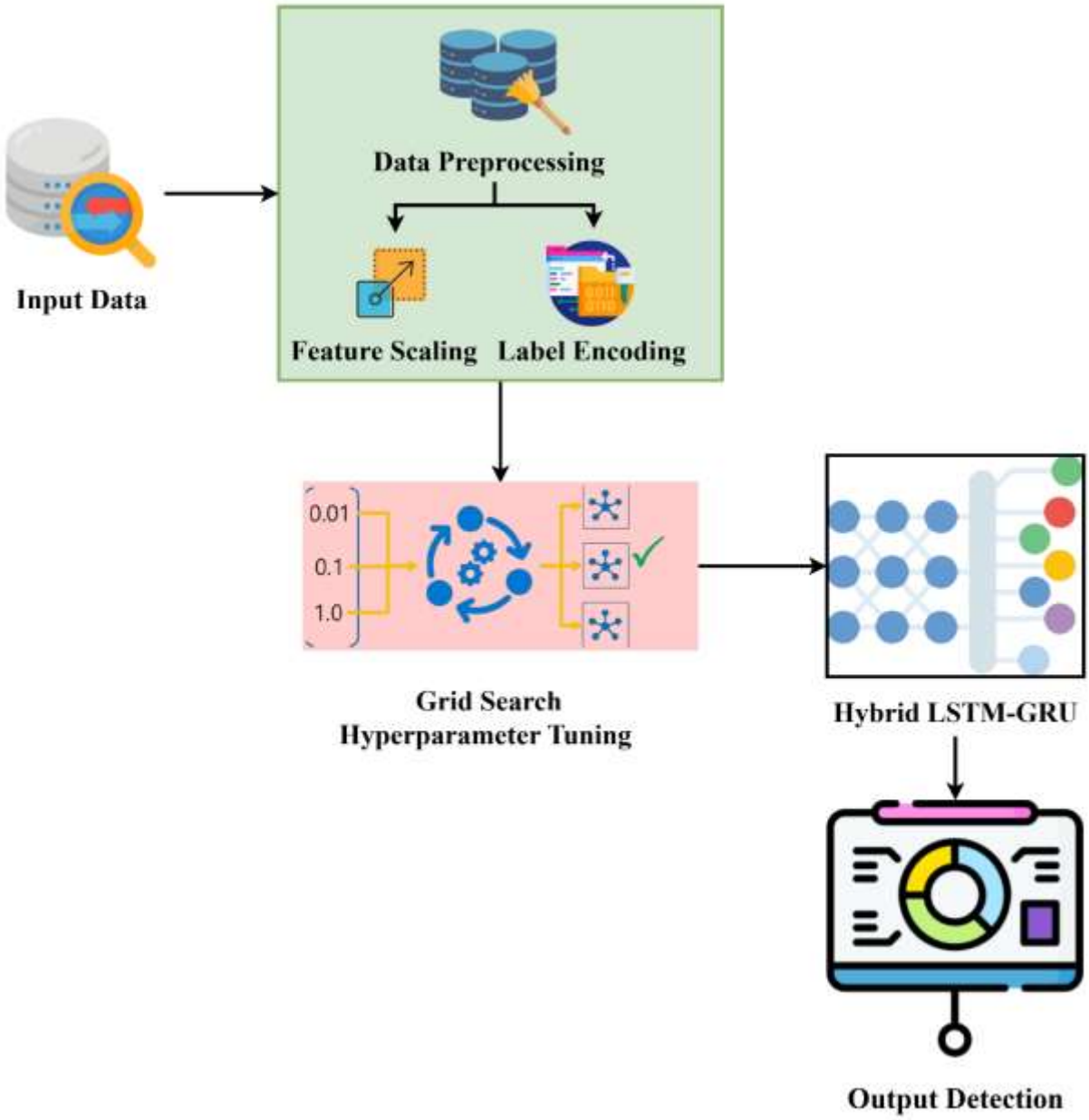

Figure 1: Hybrid LSTM-GRU Intrusion Detection Framework

3.1 Materials and Methods:

For the development and evaluation of the recommended approach, we utilize a widely utilize network traffic dataset CICIDS 2017 dataset. These datasets consists labeled network traffic data, including standard and suspicious traffic, from varied types of threats, making them suitable for evaluating IDS. The CICIDS 2017 dataset contains realistic traffic data with advanced attack situations, including DoS, DDoS, and APTs. The CICIDS 2017 dataset includes a mix of benign and malicious traffic with labeled attack types, including DoS, DDoS, Port Scanning, Botnet, and Brute Force, which are representative of real-world attack scenarios. The dataset is divided into training and testing sets, ensuring a balanced representation of both normal and attack traffic.

3.2 Data Preprocessing:

Before training, the unprocessed data undergoes preprocessing to ensure its suitability for the deep learning models:

Any missing values in the dataset are handled utilizing imputation techniques.

Features are scaled to a range between 0 and 1 utilizing Min-Max scaling to assure that every attributes contribute equally to the model's learning.

The categorical features, such as the attack type or protocol type, are encoded utilizing label encoding or one-hot encoding to convert them into numerical values suitable for the model.

Proposed Model:

3.3.1 Long Short Term Memory (LSTM):

LSTM networks are a type of RNN devised to address the limitations of traditional RNNs, particularly the problem of vanishing gradients. LSTM networks excel at identifying long-term relationships within sequential data, which constructs them highly suitable for applications such as forecasting time series, speech recognition, and also in intrusion detection in IoT networks. LSTMs are devised to learn and retain information over long periods of time while discarding irrelevant data.

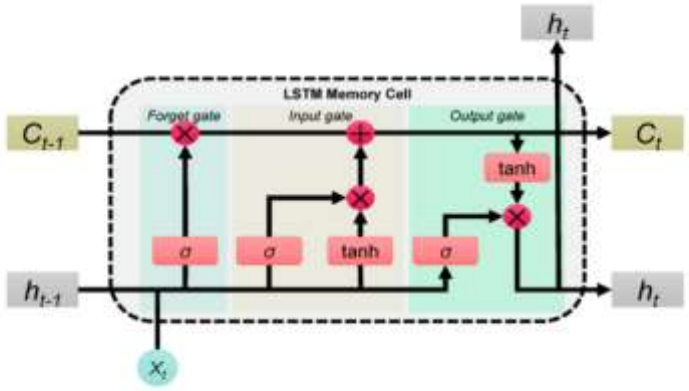

Figure 2: Long Short Term Memory Network

3.3.1.1. Forget Gate (ft)

It resolves what information to dispose from the cell state. That is the forget gate operates how much of the previous cell state (Ct-1) should be carried forward to the current cell state. It takes the previous hidden state (ht−1) and the current input (xt) as inputs and outputs a value among 0 and 1, which is then used to scale the previous cell state.

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

3.3.1.2. Input Gate (it):

The input gate resolves how much of the new information should be carried to the cell state. The candidate cell state depicts the potential new information that can be carried to the cell state. The input gate is processed as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

3.3.1.3. Update Cell State (Ct):

The cell state is upgraded by infusing the previous cell state (Ct−1) and the new candidate cell state ($\widetilde{C_t}$), controlled by the forget gate and the input gate. The update rule is as follows:

$$C_t = f_t . C_{t-1} + i_t . \widetilde{C_t} \quad (3)$$

Where C_t is the updated cell state, f_t controls quantity of the previous cell state is carried forward and i_t controls how much of the new candidate cell state is added to the cell state.

3.3.1.4. Output Gate (ot) and Hidden State (ht):

The output gate elects what the next hidden state (ht) will be. The hidden state is based on the updated cell state (C_t), which is taken to the output gate and squashed utilizing the tanh function. The output gate is computed as follows:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (4)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (5)$$

Where o_t is the output gate's output, h_t is the hidden state, W_o is the weight matrix for the output gate, b_o is the bias term and the tanh (C_t) function squashes the cell state to ensure that the hidden state is between -1 and 1. These gates and states allow LSTMs to effectively remember and forget information over long sequences, elevating them suitable for tasks such as time series prediction, language modelling, and intrusion detection in IoT networks, where temporal patterns play a crucial role in identifying attacks.

3.3.2 Gated Recurrent Unit:

The GRU is a type of RNN scheme that is devised to solve the vanishing gradient problem encountered in traditional RNNs and to efficiently capture dependencies over time in sequential data. GRUs are a simpler form of the LSTM networks, infusing the functions of memory cells and gates in a more compact form.

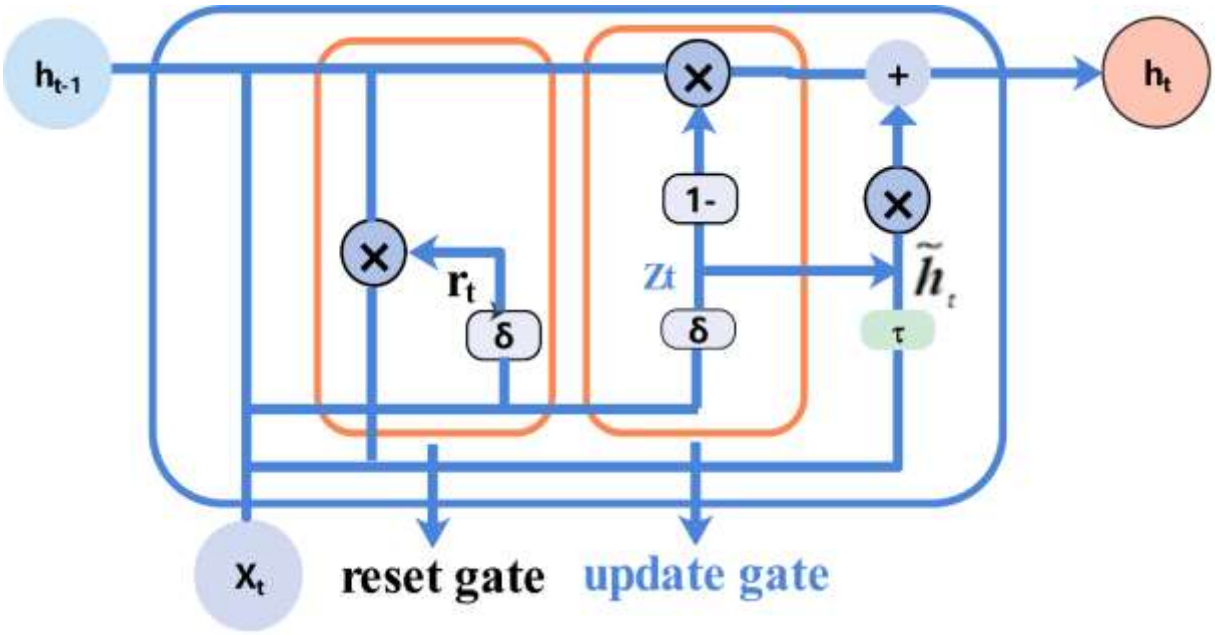

Figure 3: Gated Recurrent Unit Network

The GRU cell has the following mathematical equations:

3.3.2.1 Update Gate (zt):

The update gate resolves how much of the previous hidden state should be taken forward to the next time step. If zt is 1, the model retains the old memory, and if zt is 0, the new memory completely replaces the old memory.

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (6)$$

A value of 1 means the network retains all of ht−1, and a value of 0 means the network completely updates the hidden state with new information.

3.3.2.2 Reset Gate (rt):

The reset gate determines amount of the previous hidden state should be used when computing the candidate memory..

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (7)$$

The reset gate operates the amount of previous hidden state ht−1 should be discarded when computing the new candidate memory.

3.3.2.3 Candidate Memory ($\widetilde{h_t}$):

The candidate memory is the new memory created by combining the input at the current time step with the previous hidden state (after being varied by the reset gate). It denotes the new content to be potentially carried to the final hidden state.

$$\widetilde{h_t} = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \quad (8)$$

Where tanh is the hyperbolic tangent activation function, which outputs values among -1 and 1, Wh and Uh are weight matrices for the input xt and the previous hidden state ht−1, respectively. rt ⊙ ht−1 represents element-wise multiplication of the reset gate rt and the previous hidden state

ht−1, allowing the reset gate to control which part of the previous hidden state influences the candidate memory.

3.3.2.4 Final Hidden State (ht):

The final hidden state is derived by blending the prior hidden state with the newly generated candidate memory, weighted by the update gate.

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h_t}$$

Where ht−11 is the previous hidden state, $\widetilde{h_t}$ is the candidate memory at the current time step, zt is the update gate and (1-zt) is the complementary part of the update gate.

The update gate zt resolves the weight of the new memory ($\widetilde{h_t}$) versus the old memory (ht−1).

3.3.2.5 Features of GRU:

Simplified Structure: GRUs have fewer parameters compared to LSTMs, they don't have individual memory cell. This makes them more computationally efficient.

Capturing Temporal Dependencies: Like LSTMs, GRUs are qualified of capturing long-term and short-term dependencies in sequential data, tailored for time series and sequence-based tasks.

Less Computational Overhead: Since GRUs have fewer gates and parameters, they are computationally less expensive to train and often perform equally well to LSTMs on many tasks.

While both GRUs and LSTMs are devised to mitigate the vanishing gradient problem, GRUs are a simplified version of LSTMs. This reduction in complexity often makes GRUs faster to train and more efficient while achieving similar or better performance on certain tasks.

The GRU architecture is an efficient and powerful model for sequence learning tasks, especially in time-sensitive applications like intrusion detection. Its ability to handle long-term dependencies, coupled with its simpler architecture compared to LSTMs, makes it an attractive option for sequential modelling in environments like IoT networks.

3.3.3 Hyperparameter Tuned Grid Search Method:

Hyperparameter tuning is significant in optimizing ML or DL approaches to achieve the best performance. Hyperparameters are settings or configurations that are set before training the model like number of layers, batch size, learning rate, or dropout rates. Unlike model attributes, hyperparameters are not learned in training and detailed by the user.

The grid search method is one of the most commonly used techniques for hyperparameter optimization. It systematically explores a predefined set of hyperparameters and evaluates the model for each combination to identify the optimal configuration.

3.3.3.1 Steps in Grid Search Hyperparameter Tuning:

Define the Hyperparameter Space: The first step involves specifying the hyperparameters to tune and defining their possible values. For example:

Table 1 Hyperparameter Tuning for proposed Model

Learning Rate	No. of units	Dropout Rate	Optimizer
0.001	16	0.2	Adam
0.01	32	0.3	SGD
0.1	64	0.5	RMSProp

This creates a hyperparameter grid where each combination represents a unique set of hyperparameters.

The model is trained on the training dataset for whole possibilities of hyperparameters in the grid. During this process, the model's weights are upgraded relied on the training data, and performance metrics are computed on a validation set. The results are stored for comparison. The hyperparameter configuration that delivers the highest performance on the validation dataset is identified as the optimal set after thorough evaluation. Using these optimal hyperparameters, the model is subsequently fine-tuned by training it on the complete training dataset, and its performance is tested on the unseen test set.

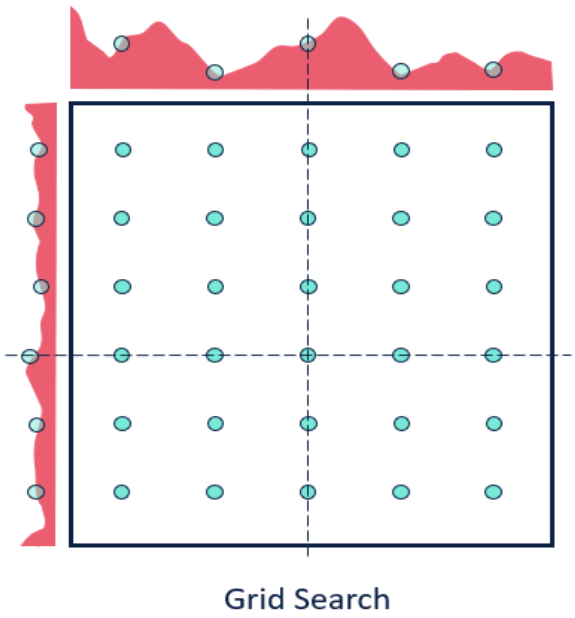

Figure 4: Hyperparameter tuning Grid Search Architecture

3.3.3.2 Advantages of Grid Search:

- Since it evaluates all possible combinations in the hyperparameter grid, Grid Search ensures that the optimal hyperparameter combination is found (within the grid's limits).
- It is easy to enforce and understand, make it popular choice for hyperparameter tuning.
- The exhaustive nature of the search ensures consistent results, which is beneficial in research and production environments.

Grid Search is a robust method for hyperparameter tuning, especially for small to moderately sized hyperparameter spaces. While computationally intensive, it provides a systematic approach to improving model performance by exploring all possible hyperparameter combinations, ensuring that the selected configuration is optimal for the given task.

3.3.4 Hybrid Deep learning using Grid Search:

The proposed framework for developing a Hybrid Intrusion Detection System (HIDS) for IoT-

enabled devices integrates the sequential modelling capabilities of LSTM and GRU layers with systematic hyperparameter tuning using Grid Search. This approach is designed to achieve accurate and efficient anomaly detection while addressing the resource constraints of IoT systems.

The framework begins with the feature extraction phase, where raw network traffic data is transformed into meaningful features representing the behaviour of IoT devices. These features include protocol-specific attributes, temporal patterns, and statistical summaries of traffic. After preprocessing, the data is split into training, validation, and testing sets to assure unbiased model examination.

The core of the framework lies in the hybrid DL approach, which gathers LSTM layer pursued by a GRU layer. The LSTM layer retains long-term dependencies and sequential correlations within the network data, leveraging its gated architecture to retain and forget information as needed. The GRU layer complements this by modeling short-term dependencies and offering computational efficiency. A dropout layer is incorporated after the hybrid sequential layers to reduce overfitting and improve generalization. Finally, fully connected dense layers are employed to refine the features for classification, leading to a sigmoid-activated output layer for binary classification tasks.

To optimize the performance of this architecture, Grid Search is employed for hyperparameter tuning. The method explores a predefined grid of hyperparameters, systematically training and validating the model for each combination to identify the configuration yielding the best performance metrics.

- The quantities of units in the LSTM and GRU layers to balance the model's capacity and complexity.
- The learning rate of the optimizer, crucial for controlling the speed and stability of training.
- Batch size, which decides the number of samples processed in training.
- Dropout rate to regulate overfitting.

This exhaustive search ensures that the model is both computationally efficient and highly accurate, especially critical for IoT environments where devices have limited processing power.

The final stage involves evaluating the tuned hybrid model on a test dataset. The tuned model consistently demonstrates a superior balance between detection accuracy and computational efficiency, adapting it for real-time IoT deployments. This framework not only highlights the advantages of combining LSTM and GRU layers but also underscores the prominence of hyperparameter optimization in achieving robust and scalable intrusion detection solutions.

4. Results and Discussions

This segment evaluates the recommended IDS within resource-constrained learning frameworks, highlighting key performance metrics and assessing the effectiveness of the hyperparameter-tuned hybrid LSTM-GRU model using the CICIDS 2017 dataset.

4.1 Implementation Details:

The proposed model was developed using Python 3.19, leveraging libraries like Matplotlib, NumPy, Pandas, Scikit-Learn, and Seaborn for evaluation and visualization. The experiments were utilized on a PC workstation equipped with an Intel i7 processor running at 3.2 GHz, 16 GB of RAM, and an NVIDIA Tesla GPU, ensuring efficient execution and performance analysis of the IDS.

4.2 Performance Metrics:

To demonstrate the effectiveness of the recommended approach, varied performance metrics are assessed and compared against more advanced deep learning models. These metrics include accuracy, precision, recall, specificity, and the F1-score, all of which offer a thorough perspective of the approach's performance.

Table 2: Evaluation of Performance Metrics

Performance Measures	Expression
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	$\frac{TN}{TP+FP}$
Recall	$\frac{\text{TP}}{\text{T P+FN}}$ x100
F1-Score	$\frac{Precison * Recall}{Precision + Recall}$
Specificity	$\frac{TN}{TN+FP}$

Four categories are utilized to examine the performance of the model. True Positive represents instances where both the actual value and the predicted value are positive, indicating a correct positive prediction. A false positive occurs when the model incorrectly predicts a positive outcome, while the actual value is negative. This type of error is crucial to minimize, as it may lead to false alarms or misidentification. In False Negative the model predicts a negative outcome when the actual value is positive. False negatives are particularly significant in many applications, such as medical diagnosis, where failing to recognise a positive case can have severe effects. True negatives refer to instances where both the actual and predicted values are negative, signifying a correct negative prediction.

Table 3: Comparitive Analysis between different models for intrusion detection

Algorithm	Accuracy	Precision	Recall	F1-Score	Specificity
CNN	0.75	0.78	0.78	0.79	0.78
RNN	0.77	0.8	0.79	0.8	0.8
LSTM	0.85	0.86	0.86	0.87	0.88
GRU	0.89	0.9	0.9	0.89	0.9
Proposed Model	0.95	0.96	0.95	0.96	0.95

Table 3 depicts a examination of varied approach, including CNN, RNN, LSTM, GRU, and the recommended approach, in terms of key performance metrics like accuracy, precision, recall, F1-score, and specificity. The outcome depicts that the recommended approach outperforms all other models, achieving the highest values across all metrics, with an accuracy of 0.95, precision

of 0.96, recall of 0.95, F1-score of 0.96, and specificity of 0.95. In comparison, the GRU model shows strong performance with accuracy of 0.89, but the proposed model demonstrates a significant improvement, highlighting its superior ability to detect intrusions with higher precision and reliability.

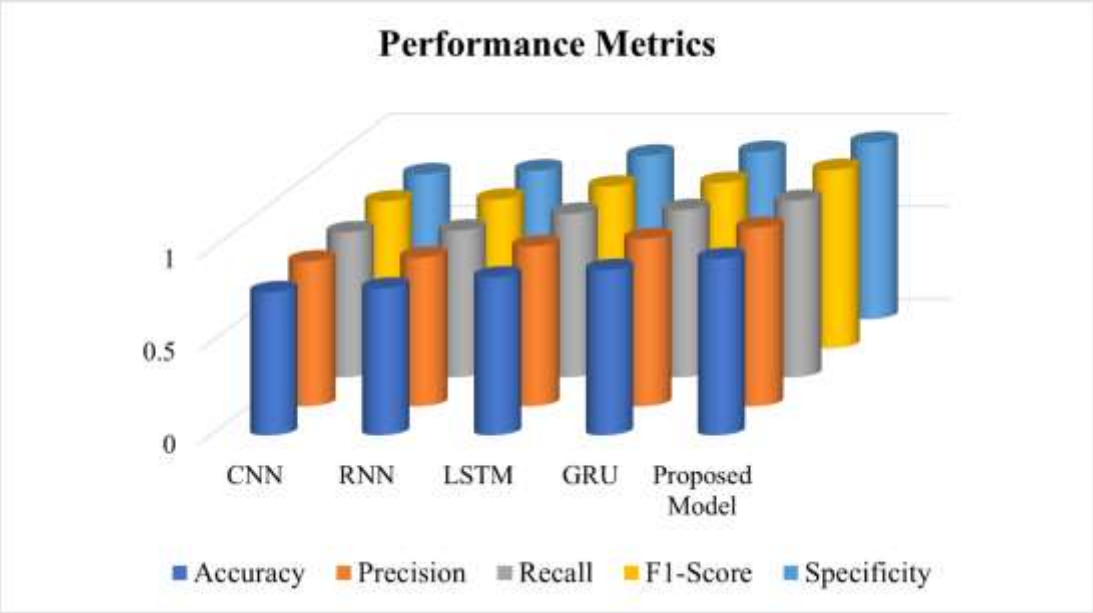

Figure 5: Comparative Analysis of Different Deep Learning Models Based on Evaluating the Accuracy

Figure 5 depicts the performance of different DL models—CNN, RNN, LSTM, GRU, and the proposed model—based on their testing accuracy. Based the graph, it is evident that the recommended approach attains the highest testing accuracy of 0.95, significantly outperforming the other models. The GRU model follows with an accuracy of 0.89, while CNN, RNN, and LSTM have comparatively lower accuracy values, initialised from 0.77 to 0.85. This highlights the proposed model's superior generalization ability and its effectiveness in intrusion detection tasks. The increasing accuracy across the models suggests improvements in performance with more advanced architectures like GRU and LSTM, with the proposed model setting a new benchmark.

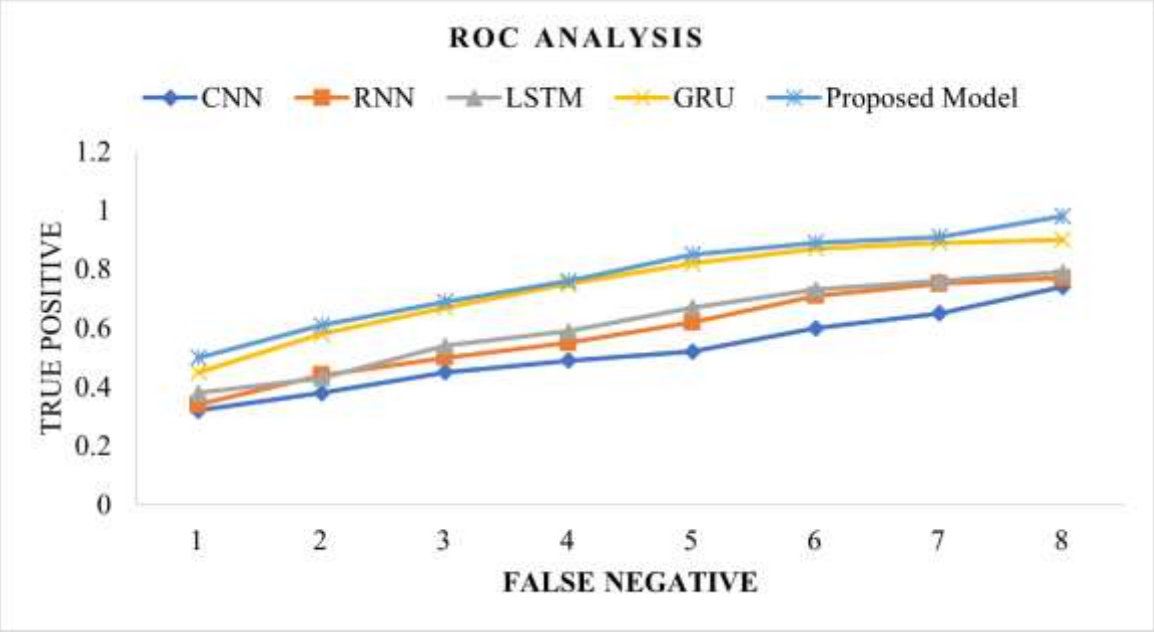

Figure 6: ROC Curve for different models in intrusion detection

Figure 6 illustrates the performance of varied approaches in detecting intrusions, providing a visual comparison of their true positive rate (sensitivity) against the false positive rate (1-specificity). A higher curve closer to the top-left corner depicts better performance, as it represents a model with a higher true positive rate and a lower false positive rate. In the graph, the proposed model demonstrates the best trade-off among sensitivity and specificity, outperforming other models like CNN, RNN, LSTM, and GRU. The ROC curve further highlights the proposed model's superior capability to discriminate among positive and negative cases with fewer errors.

5. CONCLUSIONS

In conclusion, the development of hybrid IDS for IoT-enabled devices using resource-constrained learning frameworks has demonstrated significant potential for improving security in IoT networks. The recommended model, a hyperparameter-tuned LSTM-GRU hybrid, outperforms traditional approaches by achieving superior metrics in accuracy, precision, recall, specificity, and F1-score. By employing grid search for hyperparameter tuning, the model is optimized for both performance and computational efficiency, making it well-suited for resource-constrained IoT environments. The integration of LSTM and GRU architectures elevates the approach to efficiently retains both long-term dependencies and short-term patterns in network traffic, enhancing its anomaly detection capabilities. Additionally, the use of the CICIDS 2017 dataset validates its robustness and applicability to real-world scenarios. Future research could focus on integrating federated learning to enhance data privacy, exploring adaptive learning mechanisms to address evolving threats, and optimizing the model for deployment on edge devices to enable real-time, low-latency intrusion detection in dynamic IoT ecosystems.

Funding Statement*:* The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Ahmed, M., & Mahmood, A. N. (2023). A survey on machine learning-based intrusion detection techniques for IoT. Journal of Computing and Security, 29(1), 1-20. https://doi.org/10.1016/j.jocs.2023.101951.

[2] Arshad, S., & Zubair, A. (2022). Hybrid deep learning models for efficient intrusion detection in IoT networks. Journal of Network and Computer Applications, 144, 102795. https://doi.org/10.1016/j.jnca.2021.102795.

[3] Al-Maadeed, S., & Khamis, A. (2022). Enhancing security in IoT networks using hybrid machine learning algorithms. Journal of Internet of Things and Cyber-Physical Systems, 10(3), 104-113. https://doi.org/10.1007/s42152-022-00260-0.

[4] Yadav, A., & Sharma, S. (2023). A hybrid deep learning model for intrusion detection in IoT networks: A comparative study. Future Generation Computer Systems, 118, 85-96. https://doi.org/10.1016/j.future.2021.10.041.

[5] Li, S., & Zhang, Z. (2023). A novel hybrid intrusion detection approach for IoT systems using deep learning and feature selection. Journal of Sensors, 23(8), 2769. https://doi.org/10.3390/s23082769.

[6] Zhang, H., & Zhang, Y. (2023). Hybrid feature selection and deep learning for intrusion detection in IoT networks. Computers, Materials & Continua, 71(2), 2199-2214. https://doi.org/10.32604/cmc.2023.017114.

[7] Rani, R., & Meenakshi, S. (2022). An IoT-based hybrid intrusion detection system using ensemble machine learning models. Computers & Security, 108, 102384. https://doi.org/10.1016/j.cose.2021.102384.

[8] Hasan, M., & Kaur, M. (2023). A hybrid deep learning framework for efficient anomaly detection in IoT networks. Artificial Intelligence in Medicine, 138, 102007. https://doi.org/10.1016/j.artmed.2023.102007.

[9] Liu, Y., & Jin, Y. (2022). IoT security enhancement using hybrid deep neural network-based intrusion detection. International Journal of Security and Networks, 17(5), 349-360. https://doi.org/10.1504/IJSN.2022.120399.

[10] Liu, S., & Wang, X. (2022). A hybrid deep learning approach to intrusion detection in IoT networks. International Journal of Computer Science and Information Security, 20(3), 124-133.

[11] Shahid, Usama & Hussain, Muhammad Zunnurain & Hasan, Muhammad Zulkifl & Haider, Ali & Ali, Jibran & Altaf, Jawad. (2024). Hybrid Intrusion Detection System for RPL IoT Networks Using Machine Learning and Deep Learning. IEEE Access. PP. 1-1. 10.1109/ACCESS.2024.3442529.

[12] Yaras, S., & Dener, M. (2024). IoT-Based Intrusion Detection System Using New Hybrid Deep Learning Algorithm. Electronics, 13(6), 1053. https://doi.org/10.3390/electronics13061053.

[13] Sajid, M., Malik, K.R., Almogren, A. et al. Enhancing intrusion detection: a hybrid machine and deep learning approach. J Cloud Comp 13, 123 (2024). https://doi.org/10.1186/s13677-024-00685-x.

[14] Almotairi, A., Atawneh, S., Khashan, O. A., & Khafajah, N. M. (2024). Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models. Systems Science & Control Engineering, 12(1). https://doi.org/10.1080/21642583.2024.2321381.

[15] Walling, S., & Lodh, S. (2024). Network intrusion detection system for IoT security using machine learning and statistical-based hybrid feature selection. Security and Privacy, e429. https://doi.org/10.1002/spy2.429.

[16] Meliboyev, A. (2024). IoT network intrusion detection system using machine learning techniques. Qoqon Universitesi Xabarnomasi, 11, 112-115. https://doi.org/10.54613/ku.v11i11.972.

[17] Al Sawafi, Y., Touzene, A., & Hedjam, R. (2023). Hybrid Deep Learning-Based Intrusion Detection System for RPL IoT Networks. Journal of Sensor and Actuator Networks, 12(2), 21. https://doi.org/10.3390/jsan12020021.

[18] Awajan, A. A Novel Deep Learning-Based Intrusion Detection System for IoT Networks. Computers 2023, 12, 34. https://doi.org/ 10.3390/computers12020034.

[19] Singh, A., Chatterjee, K. & Satapathy, S.C. An edge based hybrid intrusion detection framework for mobile edge computing. Complex Intell. Syst. 8, 3719–3746 (2022). https://doi.org/10.1007/s40747-021-00498-4.

[20] Smys, Smys & Basar, Dr & Wang, Dr. (2020). Hybrid Intrusion Detection System for Internet of Things (IoT). Journal of ISMAC. 2. 190-199. https://doi.org/10.36548/jismac.2020.4.002.

[21] Kharel, P., & Bhattarai, P. (2022). Hybrid feature selection and classification for intrusion detection in IoT networks. Information Sciences, 603, 46-59. https://doi.org/10.1016/j.ins.2022.04.016.

[22] Ahmed, S., & Raza, M. (2023). Secure IoT network design using hybrid machine learning-based intrusion detection. Computer Networks, 212, 108196. https://doi.org/10.1016/j.comnet.2023.108196.

[23] Kaur, P., & Sharma, D. (2023). Deep learning-based hybrid intrusion detection model for secure IoT communication. Journal of Network Security, 33(5), 94-105. https://doi.org/10.1016/j.jns.2023.01.002.

[24] Singh, G., & Kumar, S. (2023). A hybrid IoT intrusion detection system using genetic algorithm and machine

learning classifiers. Journal of Intelligent Systems, 32(4), 553-567. https://doi.org/10.1515/jisys-2023-0426.
[25] Kumar, S., & Saha, S. (2022). Hybrid model for intrusion detection in IoT networks using a combination of deep learning and random forests. Journal of Machine Learning Research, 23(1), 54-67.
[26] Dinesh, P., & Reddy, S. (2023). Anomaly-based intrusion detection using hybrid machine learning for IoT. Security and Privacy, 6(2), e433. https://doi.org/10.1002/spy2.433.
[27] Wang, C., & Yao, L. (2022). A hybrid IoT intrusion detection framework using convolutional neural networks and long short-term memory networks. IEEE Transactions on Industrial Informatics, 18(7), 4525-4532. https://doi.org/10.1109/TII.2022.3156572.
[28] Khalil, M., & Gohar, M. (2022). Hybrid deep neural network-based intrusion detection system for IoT security. Computers & Security, 111, 102420. https://doi.org/10.1016/j.cose.2021.102420.
[29] Hossain, M. I., & Karim, R. (2023). A novel hybrid approach for intrusion detection in IoT using deep learning and feature selection. Journal of Cyber Security Technology, 7(2), 87-102. https://doi.org/10.1080/23742917.2023.1911781.
[30] Thakur, R., & Arora, A. (2023). Hybrid feature selection and deep learning for intrusion detection in Internet of Things. Applied Soft Computing, 119, 108521. https://doi.org/10.1016/j.asoc.2022.108521.