Journal of
Sensor and Actuator Networks

MDPI

Article

A Hybrid CNN–GRU Deep Learning Model for IoT Network Intrusion Detection

Kuburat Oyeranti Adefemi [1,*], Murimo Bethel Mutanga [1] and Oyeniyi Akeem Alimi [2]

[1] Department of Information and Communication Technology, Mangosuthu University of Technology, Umlazi, Durban 4026, South Africa; mutangamb@mut.ac.za
[2] Department of Information Systems, Durban University of Technology, Durban 4001, South Africa; oyeniyia@dut.ac.za
* Correspondence: adefemi.kuburat@mut.ac.za

Abstract

Internet of Things (IoT) networks are constantly exposed to various security challenges and vulnerabilities, including manipulative data injections and cyberattacks. Traditional security measures are often inadequate, overburdened, and unreliable in adapting to the heterogeneous yet diverse nature of IoT networks. This emphasizes the need for intelligent and effective methodologies. In recent times, deep learning models have been extensively used to monitor and detect intrusions in complex applications. The models can effectively learn and understand the dynamic characteristics of voluminous IoT datasets to prompt efficient decision-making predictions. This study proposes a hybrid Convolutional Neural Network and Gated Recurrent Unit (CNN-GRU) algorithm to enhance intrusion detection in IoT environments. The proposed CNN-GRU model is validated using two benchmark datasets: the IoTID20 and BoT-IoT intrusion detection datasets. The proposed model incorporates an effective technique to handle the class imbalance issues that are peculiar to voluminous datasets. The results demonstrate superior accuracy, precision, recall, F1-score, and area under the curve, with a reduced false positive rate compared to similar models in the literature. Specifically, the proposed CNN–GRU achieved up to 99.83% and 99.01% accuracy, surpassing baseline models by a margin of 2–3% across both datasets. These findings highlight the model's potential for real-time cybersecurity applications in IoT networks and general industrial control systems.

Keywords: cyberattacks; convolutional neural network; deep learning; gated recurrent unit; Internet of Things; IoT intrusion detection system

check for updates

Academic Editors: Elis Kulla and Donald Elmazi

Received: 31 August 2025
Revised: 21 September 2025
Accepted: 23 September 2025
Published: 26 September 2025

Citation: Adefemi, K.O.; Mutanga, M.B.; Alimi, O.A. A Hybrid CNN–GRU Deep Learning Model for IoT Network Intrusion Detection. *J. Sens. Actuator Netw.* **2025**, *14*, 96. https://doi.org/10.3390/jsan14050096

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, information and communication technologies (ICTs) have advanced rapidly, with innovative paradigms such as the Internet of Things (IoT) becoming household terms [1]. These IoT technologies have transformed human lifestyles and interactions with the environment by enabling efficient communication and seamless data exchange across devices and their operators [1]. IoT applications span diverse domains, including smart homes, intelligent transportation, industrial automation, healthcare monitoring, and financial security [2,3].

Despite the surge in adoption, IoT networks face persistent challenges, with security and privacy concerns remaining among the most critical. Due to factors including end devices' resource constraints (particularly in terms of security measures), heterogeneity

of communication protocols, large-scale deployments with some nodes in isolated and exposed environments, the ubiquity of connected nodes, and so on, IoT systems, networks, and associated devices regularly face significant vulnerabilities, threats, and attacks [4,5]. Common IoT-network-targeted attacks, such as eavesdropping, sinkhole attacks, jamming, Sybil attacks, and Distributed Denial of Service (DDoS) attacks, can compromise the confidentiality, integrity, and availability of IoT services and networks. The consequences of these intrusions can be severe and catastrophic [6–8]. For example, in 2016, the Mirai botnet used weak IoT security to launch DDoS attacks, disrupting major platforms such as Amazon, Netflix, and Twitter [9,10]. Meanwhile, a data breach at Mars Hydro, a Chinese IoT firm, exposed 2.7 billion records, including sensitive consumer information and passwords [11].

Various studies have identified conventional security requirements, which include authentication and authorization, access control, firewalls, tamper-proofing, and so on, for securing IoT networks [8,10,12]. However, these approaches are often inadequate, computationally inept, and unreliable in adapting to the heterogeneous yet diverse nature of IoT networks [13]. Furthermore, due to the sophistication of modern-day intrusion and attack scenarios, coupled with the voluminous data being generated from IoT network devices, conventional measures struggle to provide adequate security for these networks. For instance, conventional cryptographic solutions impose high resource demands and lack the ability to detect novel intrusions [13]. Also, it should be considered that the earlier the presence of cyberattacks is detected, the quicker the mitigation responses can be activated to prevent avoidable disasters.

In the literature, several formulations and machine learning approaches have been proposed for detecting intrusions in IoT networks [14,15]. However, considering that IoT applications typically produce voluminous data, conventional intrusion detection system (IDS) models are often inefficient for modern-day IoT applications [13]. Advanced artificial intelligence subsets, particularly deep learning techniques, have demonstrated strong performance in handling large-scale IoT data due to their automatic feature extraction capabilities, ability to learn complex patterns, and adaptability to evolving threats. Various deep learning architectures have been applied for IoT network IDSs with promising results. However, some of these proposed models do not adequately cater to the class imbalance issues associated with IoT datasets. Also, it is expected that an ensemble deep learning model will present much better performance compared to singular models that were proposed in the previous studies, since the strength of one model will be strengthened by the second model. Therefore, in this study, we investigate the effectiveness of different deep learning algorithms, feature extraction techniques, and normalization approaches within the IoT network context. Furthermore, we propose a robust hybrid deep learning model using a convolutional neural network (CNN) and a gated recurrent unit (GRU) for detecting intrusions in an IoT network. The following summarizes the main contributions of this study:

- This article focuses on the use of various deep learning techniques to detect intrusion in IoT networks.
- Using two benchmark datasets, the Bot-IoT and IoTID20 datasets, we develop a hybrid deep learning model using a convolutional neural network (CNN) and a gated recurrent unit (GRU) in conjunction with various regularization and optimization techniques to increase prediction accuracy while reducing computational complexity.
- We compare and evaluate the proposed models and other models in the literature.

The remainder of the paper is structured as follows: The related work is presented in Section 2. Section 3 discusses the proposed model, datasets, data preprocessing, and performance metrics. Section 4 presents the analysis of the results. Section 5 presents

the discussion and interpretation of key findings, and Section 6 presents the conclusion, limitations, and suggestions for future work.

2. Related Works

Due to advances in computing technology, internet infrastructure and associated technologies are evolving at a geometric rate. However, these advances have raised some security concerns, particularly vulnerabilities in adopted technologies and standards. Recent works in the literature have proposed the use of different models including statistical and data-driven models for intrusion detection in IoT networks. Albanbay et al. [16] compared deep neural network (DNN), CNN, and CNN+BiLSTM approaches for IoT network intrusion detection using the CICIoT2023 dataset. Similarly, in [17], Susilo et al. modeled an IDS based on a hybrid autoencoder, LSTM, and a CNN for the classification of IoT attacks, using the same dataset. The proposed model achieved remarkable results using popular metrics. In another related work, Alhasawl et al. [18] proposed a federated learning approach that uses convolutional neural networks to efficiently detect a decentralized DDoS attack (FL-DAD) in an IoT network. Using CICIDS2017 datasets for evaluation, the findings showed that the model performed exceptionally well across popular metrics. Using a different dataset for evaluation, Sahu et al. [19] modeled an IDS based on a convolutional neural network and long short-term memory for the classification of DDoS, using the IoT-23 dataset. Similarly, Saba et al. [20] proposed a convolutional neural network model for the anomaly detection of IoT attacks, using the Bot-IoT and Network Intrusion datasets. Similarly, Villegas Ch et al. [21] proposed a graph-based intrusion detection model (Graph Neural Networks) to detect DoS, spoofing, and Man-in-the-Middle attacks in an IoT network. Using the TON-IoT dataset and a custom dataset generated from a controlled IoT testbed for evaluation, the results showed that the model performed relatively well across all considered classification metrics. In a related comparison work, Alomari et al. [22] modeled variational autoencoder and deep neural network algorithms for detecting attacks in IoT networks. Using DDoS datasets for evaluation, the developed models performed exceptionally well across all considered metrics. Similarly, Emec et al. [23] proposed a hybrid BLSTM-GRU model to detect eight different IoT network attacks. Using the CICIDS-2018 and BoT-IoT datasets for the evaluation of the proposed model, the results showed that the model performed relatively well across all considered classification metrics. Similarly, Azumah [24] proposed an LSTM intrusion model for the detection and classification of IoT device networks, focusing on the smart home environment. The authors deployed the IoT-NI dataset for the evaluation of the model. Experimental results from the model presented an accuracy of 98%, a precision of 85%, a recall of 84%, and an F1-score of 83%. Shirley and Priya [25] presented a specialized IDS model for IoT networks, combining an autoencoder for feature extraction and dimensionality reduction and a feedforward neural network for classification. Using the CICIoT2023 dataset for evaluation, the model achieves 99.55% accuracy, 100% precision, 99% recall, and an F1-score of 100% in binary classification and 90.91% in multiclass classification. Omarov et al. [26] developed a hybrid model combining CNN and BiLSTM for anomaly detection in IoT networks, using the UNSW-NB dataset for evaluation. Table 1 presents a summary of the existing methods using deep learning techniques for IoT networks intrusion detection.

Despite the promising results achieved by these various proposed data-driven models for intrusion detection in IoT environments, several key challenges remain unresolved. Many existing models focus on isolated aspects of feature extraction, limiting their ability to fully capture the complex nature of IoT network traffic. Additionally, commonly used benchmark datasets are often characterized by significant class imbalance, which negatively impacts classification performance. Furthermore, previous studies tend to overempha-

size accuracy while overlooking other essential performance metrics such as precision, recall, and F1-score. To bridge these gaps, this study proposes a novel hybrid CNN-GRU model. This architecture is designed to integrate spatial and temporal feature learning more efficiently. We apply the Synthetic Minority Oversampling Technique (SMOTE) to mitigate data imbalance and conduct a comprehensive evaluation using multiple key metrics. Finally, the model is rigorously evaluated on multiple datasets to demonstrate its generalizability and effectiveness against a wide range of attack variants.

Table 1. Comparison of existing methods.

Article	Dataset Used	Model	Performance
Albanbay et al. [16]	CICIoT23	DNN, CNN, BiLSTM	94.84%
Susilo et al. [17]	CICIoT23	AE, LSTM, CNN	99.15%
Alhasawl et al. [18]	CICIoT17	CNN	98.7%
Sahu et al. [19]	CICIoT23	CNN+LSTM	96%
Saba et al. [20]	Bot-IoT and IoT-IN	CNN	95.55%
Villegas Ch et al. [21]	ToN-IoT	GNN	95%
Alomari et al. [22]	DDoSDataset	AE, DNN	92.8%
Emec et al. [23]	CIC2018 and BoT-Iot	BLSTM-GRU	98.58%
Azumah et al. [24]	IoT-NI	LSTM	98%
Shirley et al. [25]	CICIoT23	AE and FFNN	99.55%
Omarov et al. [26]	UNSW-NB	CNN-BiLSTM	96.28%

3. Methodology

This section presents the development of the proposed hybrid deep learning model (CNN-GRU) for detecting intrusions in IoT networks. The aim of this study is to classify network traffic into normal and attack categories. Our approach involves data description, data preprocessing, model development, model evaluation, and result interpretation, as illustrated in Figure 1.

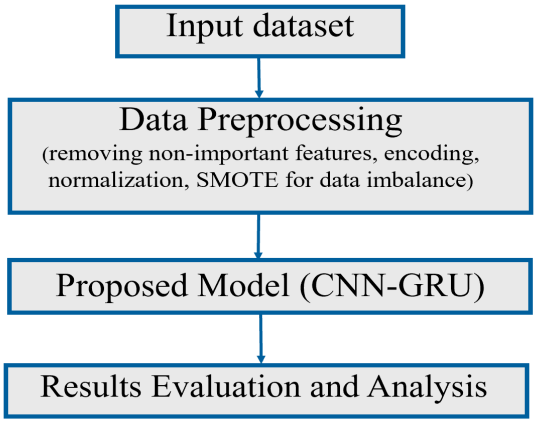


Figure 1. Workflow of the proposed model.

3.1. Dataset Description

The study deployed modified versions of two IoT benchmark datasets: the Bot-IoT and IoTID20 datasets. The two datasets were generated from realistic IoT network environments, incorporating both benign and malicious traffic patterns. The choice of these datasets ensures the proposed CNN-GRU model is trained and evaluated on diverse traffic scenarios, enabling it to generalize effectively across different IoT network frameworks.

The Bot-IoT dataset was developed by Koroniotis et al. [27] at the Cyber Range Lab of UNSW-Canberra in 2019. The Bot-IoT dataset combines real and simulated IoT network traffic, including various types of modern-day network attacks. With 46 features, the generated dataset contains more than 72 million records of normal and attacked traffic. The Bot-IoT dataset comprises four types of attacks: Normal, Scan, Data Theft, DoS, and DDoS attacks. The summary of the Bot-IoT dataset is presented in Table 2. Out of the 72 million instances, 10% of the original dataset was extracted for training and testing the developed model.

Table 2. Summary of Bot-IoT dataset.

Category	Subcategory	Number of Instances
Normal	Normal	105,202
DoS	HTTP	34,057
	TCP	19,111,830
	UDP	37,881,485
DDoS	HTTP	51,934
	TCP	15,975,894
	UDP	21,049,846
Scan	OS fingerprinting	350,093
	Service scanning	1,481,465
Data theft	Data exfiltration	5003
	Key logging	1387
Total		96,048,196

The IoTID20 dataset is a publicly available intrusion detection dataset generated by Ullah and Mahmoud [28]. The dataset was generated from modeling an IoT testbed focusing on smart home networks. IoTID20 captures network traffic between various IoT devices. The testbed setup includes AI Speakers, WiFi cameras, laptops, smartphones, wireless access points, and a router. The cameras and speakers serve as the victim devices, while the remaining devices are deployed as the attacking devices. The dataset includes 625,783 instances and 83 features that comprise both benign traffic and four types of attacks: denial of service (DoS), distributed denial of service (DDoS), data theft, and reconnaissance/scanning, as shown in a table. Table 3 presents a summary of the dataset. In this study, 80% of the original dataset (500, 625 instances) was used for training and testing the model.

Table 3. Summary of IoTID20 dataset.

Category	Number of Instances
Normal	40,073
DoS	59,391
MITM	35,377
Mirai	415,677
Scan	75,265

3.2. Data Preprocessing

The performance of a deep learning-based intrusion detection system is fundamentally dependent on the quality and characteristics of its input data. To ensure model reliability, the raw IoT network traffic data must be accurately cleaned and transformed for training models. This section describes the comprehensive preprocessing steps undertaken, which

include the removal of irrelevant features, conversion of categorical data into numerical values, scaling through normalization, rebalancing of the dataset, and splitting into training and testing sets. The first step involves removing non-informative features such as Flow ID, Source IP, Source Port, Destination IP, and Timestamp. These features do not contribute meaningfully to distinguishing between normal and malicious traffic. Removing them helps reduce noise and redundancies in the voluminous datasets. We used a low-variance filter to remove features where 99% of the values were identical, as they have no useful information for the model to learn from. Identifiers and session descriptors were removed as they lack discriminatory power for classifying traffic behavior. Utilizing these features would cause the model to overfit on specific network moments in time rather than learning generalizable patterns of malicious activity. This removal reduces dataset dimensionality and noise. Since deep learning algorithms require numerical input, categorical features were first converted into a numerical representation using the One-Hot Encoding method. Following this, the data was normalized to ensure all features are on a similar scale, thereby enhancing training stability. This was achieved using Min-Max Scaling, which maps the original data to a fixed range between 0 and 1, as defined in Equation (1).

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

where x is the original value, x' is the normalized value, $min(x)$ is the minimum value, and $max(x)$ is the maximum value of the feature. Furthermore, addressing data imbalance is a crucial part of preprocessing as it significantly influences model performance. In many IoT benchmark datasets, class imbalance is a major issue. This challenge makes classification models more likely to favor the majority class while overlooking rare classes. To mitigate this issue, this study applied the Synthetic Minority Oversampling Technique (SMOTE). SMOTE helps by selecting instances from the minority class, identifying their nearest neighbors, and creating new, similar samples, thus contributing to a more balanced dataset. The final step in the preprocessing stage was to divide the dataset into training and testing subsets. More information on SMOTE can be found in [29]. The training set was used to train the model, while the testing set was used to evaluate the performance of the developed model. We applied an 80:20 split, using 80% of the data for training and 20% for testing.

3.3. Proposed CNN-GRU Model

This study proposes a novel CNN-GRU hybrid model designed to enhance intrusion detection for IoT networks. The rationale for this design capitalizes on the complementary strengths of both models as CNNs excel at extracting spatial features, while GRUs are adept at capturing temporal dependencies in sequential network traffic. This integration significantly strengthens a robust and generalizable detection system that is less prone to overfitting and bias. As illustrated in Figure 2, the architecture sequentially incorporates an input layer, convolutional and pooling layers, a GRU layer, a dense layer, a dropout layer, and a final output layer.

The preprocessed IoT network traffic data is first fed into the input layer, which formats the sequential input for subsequent processing. This data is then processed by a one-dimensional convolutional layer, where filters are applied to extract spatial features indicative of malicious activity. The convolutional outputs are passed through a Rectified Linear Unit (ReLU) activation function to introduce non-linearity into the model, promoting faster learning and improved gradient flow [30]. The convolution operation is defined in Equation (2):

$$y[n] = (x * w)[n] = \sum_{k=0}^{P-1} x[n+k] \cdot w[k] \quad (2)$$

where $y[n]$ the output at position n, $x[n]$ is the input signal at position n, $w[k]$ is the filter or kernel value at position k and P is the size of the filter or kernel. Subsequently, a max-pooling layer downsamples the feature maps, reducing their spatial dimensionality. This operation serves to decrease computational complexity, control overfitting, and enhance translational invariance by retaining the most prominent features from each region. The condensed features are then passed to the GRU layer. The GRU layer is designed to model temporal dynamics and long-range dependencies within the sequence data, which is critical for identifying attack patterns that unfold over time. The GRU addresses the vanishing gradient problem common in standard RNNs through the use of gating mechanisms [31]. Each GRU cell contains a reset gate and an update gate that regulate information flow. The mathematical formulation of the GRU is given by Equations (3)–(6):

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (3)$$

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (4)$$

$$\tilde{h}_t = tanh(W \cdot [r_t \times h_{t-1}, x_t]) \quad (5)$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t \quad (6)$$

In these equations, x_t is the input, h_t is the output vector, $\tilde{h}_t$ is the activation vector, z_t denotes the update gate vector, r_t represents the reset gate, W denotes the weight matrices, and σ is the sigmoid function. The resulting sequential features are then forwarded to a fully connected (dense) layer for high-level feature integration. To further combat overfitting, a dropout layer is incorporated, which randomly omits a subset of neurons during training to improve generalization. Finally, the output layer uses a softmax activation function to map the transformed features into a probability distribution over the two output classes, benign and malicious, producing the final classification decision.

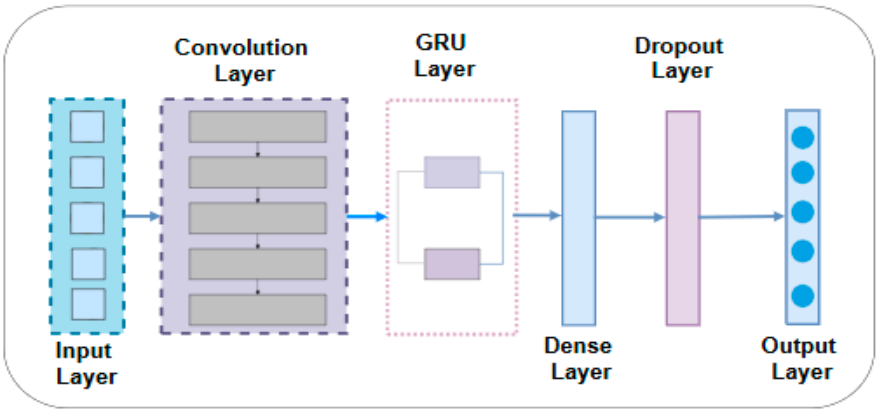


Figure 2. Architecture of the proposed CNN-GRU model.

3.4. Experimentation Setup

All experiments were performed on an Intel Core i5-4300 CPU (2.50 GHz) and 8 GB of RAM. The proposed model was developed using TensorFlow, Keras, Pandas, and Scikit-learn. The dataset underwent a preprocessing approach that included categorical encoding, normalization, and balancing of class distributions before splitting into training and testing sets. Through iterative experimentation, the proposed CNN-GRU architecture was determined to begin with an input layer, followed by a 1D convolutional layer (64 filters, kernel

size of 3, ReLU activation), a max-pooling layer, and a GRU layer with 64 units. To mitigate overfitting, a dropout layer (rate = 0.5) is incorporated, followed by a dense layer (32 units, ReLU) and a final softmax-activated output layer for binary classification. This design enables the CNN to identify spatial features while the GRU analyzes temporal sequences. The model was compiled with the Adam optimizer and binary cross-entropy loss function. A comprehensive grid search was conducted to optimize key hyperparameters, including the number of GRU units (32, 64, 96), learning rate (0.001 to 0.0002), dropout rate (0.2 to 0.5), batch size (32, 64, 128), and number of epochs (50, 100, 150). The optimal configuration, which achieved stable convergence, is 64 GRU units, a learning rate of 0.001, a dropout rate of 0.5, and a batch size of 64 and was trained for 50 epochs. Performance was assessed using standard metrics: accuracy, precision, recall, F1-score, and AUC. For a robust comparison, the proposed hybrid model was evaluated against baseline architectures, including a standalone CNN, GRU, Feedforward Neural Network (FFNN), and LSTM. All results presented in Section 4 reflect the model's performance on the completely unseen test set, which was rigorously excluded from all development and tuning phases.

3.5. Baseline Methods

This sub-section briefly discusses the baseline models used to compare the performance of the proposed model, including a feedforward neural network, LSTM, a standalone CNN, and a standalone GRU. These models were chosen for their automatic feature extraction, sequential dependencies, and computational efficiency.

3.5.1. Feedforward Neural Network (FFNN)

The foundational feedforward neural network is developed to be simple and computationally efficient with fully connected layers. The model is made up of only an input layer, a hidden layer, and an output layer. The model provides good performance without requiring a significant number of resources.

3.5.2. Long Short-Term Memory (LSTM)

LSTM was designed to overcome the limitations of traditional RNNs in capturing long-term dependencies. Its architecture is made up of a memory cell and three gates, input, forget, and output, that regulate the flow of information [32]. The memory cell acts as long-term storage, retaining important values over time, while the gates determine how this information is updated and used. The input gate decides how much new information should enter the memory cell, typically using sigmoid and tanh activation functions. The forget gate controls which parts of the previous state should be discarded by assigning values between 0 (forget) and 1 (retain). Finally, the output gate decides what information is passed forward as output.

3.6. Performance Metrics

The performance of the proposed models was evaluated using a comprehensive set of metrics that includes accuracy, precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (AUC). These metrics are mathematically defined below.

Accuracy measures the overall proportion of correct predictions made by the model across all classes, as calculated by Equation (7):

$$\% \text{ Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Samples}} \tag{7}$$

Precision quantifies the fraction of correctly identified positive instances out of all instances predicted as positive. A high precision indicates a low rate of false alarms. It is computed using Equation (8):

$$\text{Precision} = \frac{\text{True Positives}}{\text{False Positives} + \text{True Positives}} \tag{8}$$

where a True Positive (TP) is a correct prediction of the positive class, and a False Positive (FP) is an incorrect outcome of the positive class.

Recall (or Sensitivity) measures the model's ability to correctly identify all actual positive instances. It represents the fraction of true positives that were correctly identified, as given by Equation (9):

$$\text{Recall} = \frac{\text{True Positives}}{\text{False Negatives} + \text{True Positives}} \tag{9}$$

A False Negative (FN) occurs when the model fails to predict the positive class.

F1-score provides a single metric that balances both precision and recall by calculating their harmonic mean as defined in Equation (10):

$$\text{F1} - \text{score} = 2 \times \frac{(\text{precision} \times \text{recall})}{(\text{precision} + \text{recall})} \tag{10}$$

4. Experimental Results

This section discusses and analyzes the experimental results of the proposed deep learning model that is specifically developed for binary classification of intrusive IoT network traffic.

4.1. The Proposed CNN-GRU Results on the Two Datasets

The performance of the proposed CNN-GRU model was rigorously evaluated on the completely unseen test set. As shown in Figure 3, the CNN-GRU model achieved an accuracy of 99.83% on the modified versions of the IoTID20 dataset and 99.01% on Bot-IoT dataset. On the IoTID20 dataset, the model obtained a precision of 99.83%, a recall of 99.82%, and an F1-score of 99.83%, while on the Bot-IoT dataset, it achieved a precision of 98.94%, a recall of 98.53%, and an F1-score of 98.73%. These results indicate that CNN-GRU can effectively capture complex spatial–temporal features of IoT traffic, enabling highly accurate detection of both benign and malicious instances. The slightly lower performance on the Bot-IoT dataset likely reflects the higher variability and complexity of attacks in this dataset. Overall, the consistently high accuracy, precision, recall, and F1-score highlight CNN-GRU as a reliable and generalizable approach for intrusion detection in diverse IoT network environments.

4.2. Performance Comparison of the Proposed Model with Baseline Methods

The experimental results in Table 4 demonstrate that the proposed CNN–GRU model consistently outperforms all baseline methods across both modified versions of the IoTID20 and Bot-IoT datasets. On the modified version of the IoTID20 dataset, CNN-GRU achieved the highest accuracy (99.83%), precision (99.83%), recall (99.82%), and F1-score (99.83%), surpassing the CNN model by 0.93% in accuracy. On the modified version of the Bot-IoT dataset, the model also achieved superior performance with an accuracy of 99.01%, a precision of 98.94%, a recall of 98.53%, and an F1-score of 98.73%, outperforming the CNN and GRU by 0.60% and 0.85% in accuracy, respectively. Among the baseline models, the FFNN achieved the lowest performance with 90.47% accuracy and 90.23% F1-score using IoTID20 and 89.24% accuracy and 89.43% F1-score on Bot-IoT, indicating that simple

feed-forward networks are insufficient for capturing the complex spatial–temporal patterns of IoT traffic. The LSTM and GRU models showed significantly improved results, with accuracies above 96% on both datasets. The CNN performed slightly better than the GRU on IoTID20, with 98.90% accuracy and an F1-score of 98.80%. Using the Bot-IoT, the model achieved 98.41% accuracy and 98.16% F1-score, demonstrating the advantage of convolutional layers in extracting spatial features. The superior performance of the CNN-GRU hybrid can be attributed to its ability to simultaneously extract spatial features via convolutional layers and capture temporal dependencies through the GRU component. The relatively higher scores on IoTID20 data reflect their lower complexity, with fewer features, fewer attack categories, and a more balanced class distribution. In contrast, Bot-IoT presents a more challenging classification problem due to higher variability and class imbalance, making the model's strong performance on this dataset particularly notable. Overall, the balanced improvement in precision and recall across datasets indicates that the proposed CNN–GRU model effectively detects both majority and minority attack classes, reducing bias and ensuring reliable intrusion detection across diverse IoT network environments.

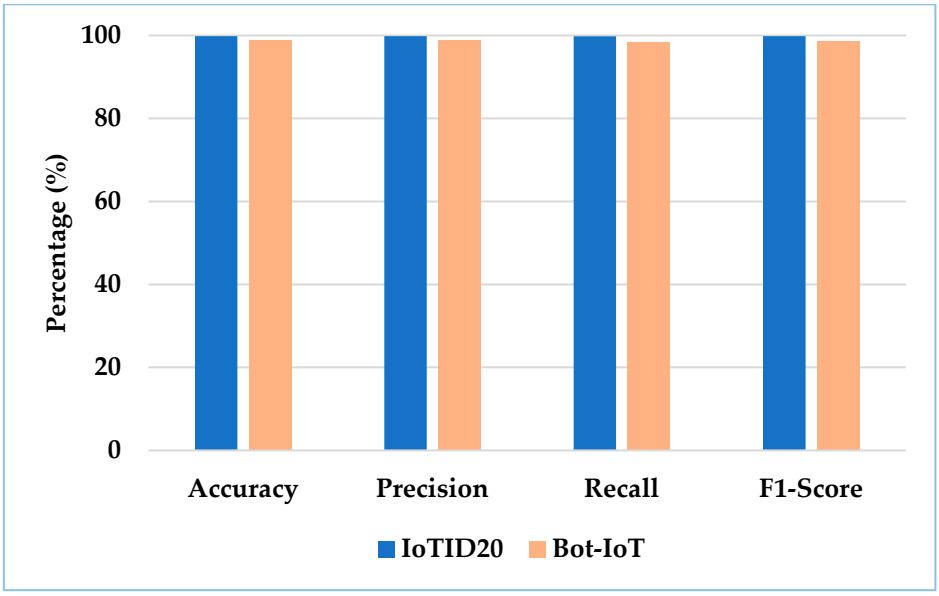


Figure 3. The CNN-GRU results on the two datasets.

Table 4. Performance comparison of the proposed model with baseline methods.

Dataset	Model	Accuracy	Precision	Recall	F1-Score	AUC
IoTID20	FFNN	90.47	90.23	90.23	90.23	0.90
	LSTM	97.26	97.69	97.19	97.43	0.97
	CNN	98.90	98.93	98.90	98.91	0.99
	GRU	98.8	98.80	98.80	98.80	0.99
	CNN-GRU	99.83	99.83	99.82	99.83	1.00
Bot-IoT	FFNN	89.24	89.45	89.42	89.43	0.89
	LSTM	96.12	96.12	97.03	97.57	0.96
	CNN	98.41	98.33	98.30	98.31	0.98
	GRU	98.16	98.18	98.09	98.13	0.98
	CNN-GRU	99.01	98.94	98.53	98.73	0.99

The Area Under the Receiver Operating Characteristic Curve (AUC) is a widely used performance metric that quantifies the overall ability of a model to discriminate between positive and negative classes across all possible decision thresholds. The AUC provides a comprehensive measure of model robustness, as it accounts for the trade-off

between the true positive rate (TPR) and false positive rate (FPR). As shown in Table 3, the IoTID20 dataset results indicate that both the CNN and GRU models demonstrated strong performance, AUC = 0.99, while the LSTM achieved slightly lower performance (AUC = 0.97). The FFNN achieved the lowest performance with an AUC of 0.90, indicating its limited ability to capture complex temporal dependencies in IoT data. In contrast, the CNN-GRU model attained an AUC of approximately 1.0, reflecting its effectiveness in combining the spatial feature extraction capability of a CNN with the sequential learning strength of a GRU.

Similarly, for the Bot-IoT dataset, the CNN-GRU model outperforms the baseline models, achieving an AUC of 0.99. Although CNN and GRU achieved competitive results (AUC = 0.98), their performance was still marginally lower than that of the proposed CNN-GRU model. LSTM achieved an AUC of 0.96, while the FFNN again achieved the lowest AUC of 0.89. Overall, the AUC analysis highlights the robustness and generalizability of the proposed CNN-GRU model across the IoT datasets. By achieving strong AUC values, the model demonstrates its reliability in distinguishing between benign and malicious traffic, thereby confirming its potential as a practical intrusion detection solution in real-world IoT environments.

4.3. Statistical Significance Analysis

To validate the superiority of the proposed CNN-GRU model over the baseline models, a series of paired sample *t*-tests were conducted. The tests compared the CNN-GRU model against each baseline model (FFNN and LSTM) across the five performance metrics (accuracy, precision, recall, F1-score, and AUC). The resulting t-statistics and *p*-values are summarized in Table 5.

Table 5. Statistical significance analysis result.

Dataset	Comparison	t-Statistic	*p*-Value
IoTID20	CNN-GRU vs. FFNN	4.05	0.0155
	CNN-GRU vs. LSTM	4.00	0.0161
BoT-IoT	CNN-GRU vs. FFNN	4.05	0.0155
	CNN-GRU vs. LSTM	4.05	0.0169

On the IoTID20 dataset, the CNN-GRU model achieved significantly higher results than the baseline methods, with *p*-values below 0.05. Similarly, on the Bot-IoT dataset, the CNN-GRU model significantly outperformed the FFNN and LSTM models. These findings demonstrate that the proposed CNN-GRU model consistently yields superior performance, particularly over simpler architectures, thereby reinforcing its robustness and generalization across diverse IoT intrusion detection scenarios.

5. Discussion and Interpretation of Key Findings

This study proposed a hybrid CNN-GRU model for intrusion detection in IoT networks. The experimental results demonstrate the model's robust performance, achieving an accuracy of 99.83% on the IoTID20 dataset and 99.01% on the BoT-IoT dataset. The proposed model's high levels of accuracy, precision, recall, and F1-score highlight its efficiency. The model's performance can be attributed to its hybridization design combining a CNN component, which excels at extracting spatial features and local patterns from individual network packets, and a GRU component, which effectively captures long-range temporal dependencies and sequential behaviors within the network flow. This dual-capability architecture allows the model to discern complex attack signatures that are both spatially malicious and temporally anomalous, a significant advantage over standalone models.

5.1. Comparison with Results from Similar Studies

The performance of the proposed CNN-GRU model was compared with recent similar studies in the literature, as summarized in Table 6.

Table 6. Performance comparison with previous studies.

Article	Model	Accuracy	Precision	Recall	F1-Score
[17]	AE+LSTM+CNN	99.15	99.39	99.00	99.19
[18]	CNN	98.7	98.8	98.9	98.8
[20]	CNN	95.55	-	-	-
[21]	GNN	-	96.00	94.00	95.00
[24]	LSTM	98.00	85.00	84.00	83.00
Our Work	**CNN-GRU**	**99.83**	**99.83**	**99.82**	**99.83**

- Comparison with hybrid models: The model outperformed the hybrid Autoencoder-LSTM-CNN architecture presented in [17], which achieved an accuracy of 99.15% on the CICIoT2023 dataset. While this result is commendable, our CNN-GRU model's higher performance (99.83%) suggests that the combination of a CNN with a computationally efficient GRU can be effective.
- Comparison with standalone models: The proposed hybrid model consistently outperforms standalone models. For instance, it exceeded the accuracy of a standalone CNN model (98.7%) on the CIC-IoT2017 dataset [18], another standalone CNN model that obtained 95.5% on Bot-IoT and IoT-NI [20], and a standalone LSTM model (98.00%) on the IoT-NI dataset [24]. This performance gap highlights the inherent limitation of models that specialize in only one type of feature (spatial or temporal) when faced with the complex nature of network intrusion data. While the studies in [18–20] indeed report excellent results, the advantages of the proposed CNN-GRU model extend beyond a marginal increase in accuracy; the proposed model also demonstrated robustness. The hybrid CNN-GRU architecture provides an advantage over the CNN and LSTM standalone models presented in [18,20,24]. Network intrusion data possesses both spatial patterns within packets and temporal dependencies across traffic flows. A standalone model is primarily limited to capturing only one type of feature. Our model's ability to learn both feature types provides a more holistic analysis, which is likely the reason for its improved performance. Secondly, a key strategic advantage is the choice of GRU over the more frequently used LSTM for the temporal component. While the hybrid CNN-LSTM model in [24] performs well, GRUs are known to achieve comparable accuracy with lower computational overhead due to their simpler gating mechanism. This makes the proposed CNN-GRU model not just accurate, but also more efficient and better suited for the computational constraints of real-world IoT deployments, a significant practical advantage. Our model's ability to achieve state-of-the-art results on two distinct datasets (IoTID20 and BoT-IoT) strongly indicates that it learns generalizable features of malicious traffic.
- Comparison with advanced architectures: The model also demonstrates strong performance against other approaches. It exceeded the F1-score (95.00%) reported for a Graph Neural Network (GNN) on the ToN-IoT dataset [21]. While GNNs are powerful for modeling network topology, the CNN-GRU approach offers a potent alternative that achieves high detection rates by focusing on the spatio-temporal patterns within the traffic flows themselves, without requiring explicit graph-structure data.

5.2. Critical Analysis and Limitations of the Comparison

It is important to interpret the comparisons with caution. Direct numerical comparison of accuracy metrics is challenging due to critical methodological variations:

- Dataset differences: Studies utilize different datasets (e.g., CICIoT2023, CIC-IoT2017, BoT-IoT, ToN-IoT), each with unique characteristics, class distributions, and attack profiles. A model highly tuned to one dataset may not generalize directly to another.
- Data preprocessing and feature selection: The specific features extracted, normalization techniques applied, and handling of class imbalance vary significantly across studies and greatly impact model performance.
- Data volume: Some studies use full datasets, while others use modified or fractional components [18,20]. Training on a larger, more diverse volume of data, as in this study, often leads to a more robust and generalizable model. Therefore, the most significant evidence of the CNN-GRU model's strength is not solely its superior metrics, but its demonstrated ability to achieve higher performance across different datasets (IoTID20 and BoT-IoT), indicating a degree of robustness and generalizability.

The high accuracy and robustness of the CNN-GRU model position it as a practicable model for real-world IoT security applications, for potential deployment at network gateways for real-time intrusion detection. Its design offers a favorable balance between performance and computational efficiency, which is crucial for IoT environments. In conclusion, the hybrid CNN-GRU model represents a powerful and effective framework for IoT intrusion detection.

6. Conclusions and Limitations

This study introduces a novel hybrid CNN-GRU architecture that demonstrates a significant advancement in deep learning-based intrusion detection for IoT networks. The novelty of this work lies in the integration of convolutional and recurrent layers, which enables the simultaneous extraction of critical spatial features from network traffic and the modeling of their long-range temporal dependencies. This integrated approach is uniquely appropriate for identifying complex, multi-stage cyberattacks that evolve over time, a challenge that standalone models often fail to address comprehensively. The results show an exceptional performance, achieving accuracies of 99.83% and 99.01% and outperforming standalone CNN, GRU, LSTM, and FFNN models. These results highlight the significant potential of hybrid deep learning approaches for accurately detecting complex and evolving cyber threats in IoT environments, thereby enhancing network resilience and protecting user privacy. However, this study has certain limitations that point toward valuable directions for future research. A primary limitation is the lack of interpretability techniques, such as LIME or SHAP, leaving the model's decision-making process as a "black box", which may limit understanding of how specific features influence anomaly detection. Another limitation lies in the use of a modified version of the benchmark datasets, which, although effective for experimentation, may not fully represent all real-world traffic patterns. In addition, the datasets employed were collected more than two years prior to this study, which implies that recent shifts in cyberattack strategies may not be adequately captured. Future work will focus on integrating explainable AI methods, such as LIME and SHAP, to enhance the transparency of the model and provide insights into its predictions, thereby increasing trust and practical applicability in real-world IoT security settings. Furthermore, future work will consider using recent IoT traffic data that reflects evolving attack trends, thereby improving the practical applicability of the proposed model in real-world IoT security settings.

Author Contributions: Conceptualization, K.O.A., M.B.M. and O.A.A.; methodology, K.O.A. and O.A.A.; software, K.O.A.; validation; K.O.A. and O.A.A.; formal analysis, K.O.A. and O.A.A.; investigation, K.O.A.; resources, K.O.A., M.B.M. and O.A.A.; data curation, K.O.A.; writing—original draft preparation, K.O.A. and O.A.A.; writing—review and editing, K.O.A. and O.A.A.; visualization,

K.O.A., M.B.M. and O.A.A.; supervision, M.B.M.; project administration, M.B.M.; funding acquisition, M.B.M. and O.A.A. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by Durban University of Technology.

Data Availability Statement: Publicly available open-source datasets were analyzed in this study. This data can be accessed at https://sites.google.com/view/iot-network-intrusion-dataset/home (accessed on 1 June 2025) and https://research.unsw.edu.au/projects/bot-iot-dataset (accessed on 1 June 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sayed, M. The Internet of Things (IoT), Applications and Challenges: A Comprehensive Review. *J. Innov. Intell. Comput. Emerg. Technol.* **2024**, *1*, 20–27.
2. Adefemi Alimi, K.O.; Ouahada, K.; Abu-Mahfouz, A.M.; Rimer, S.; Alimi, O.A. Refined LSTM Based Intrusion Detection for Denial-of-Service Attack in Internet of Things. *J. Sens. Actuator Netw.* **2022**, *11*, 32. [CrossRef]
3. Das, S.; Namasudra, S. Introducing the Internet of Things: Fundamentals, Challenges, and Applications. *Adv. Comput.* **2025**, *137*, 1–36.
4. Alzahrani, A.I. Exploring AI and Quantum Computing Synergies in Holographic Counterpart Frameworks for IoT Security and Privacy. *J. Supercomput.* **2025**, *81*, 1194. [CrossRef]
5. Sharma, S.B.; Bairwa, A.K. Leveraging AI for Intrusion Detection in IoT Ecosystems: A Comprehensive Study. *IEEE Access* **2025**, *13*, 66290–66317. [CrossRef]
6. Dritsas, E.; Trigka, M. A Survey on Cybersecurity in IoT. *Future Internet* **2025**, *17*, 30. [CrossRef]
7. Yaras, S.; Dener, M. IoT-Based Intrusion Detection System Using New Hybrid Deep Learning Algorithm. *Electronics* **2024**, *13*, 1053. [CrossRef]
8. Zahid, M.; Bharati, T.S. Enhancing Cybersecurity in IoT Systems: A Hybrid Deep Learning Approach for Real-Time Attack Detection. *Discov. Internet Things* **2025**, *5*, 73. [CrossRef]
9. Wei, C.; Xie, G.; Diao, Z. A Lightweight Deep Learning Framework for Botnet Detecting at the IoT Edge. *Comput. Secur.* **2023**, *129*, 103195. [CrossRef]
10. Al-Shurbaji, T.; Anbar, M.; Manickam, S.; Hasbullah, I.H.; ALfriehate, N.; Alabsi, B.A.; Alzighaibi, A.R.; Hashim, H. Deep Learning-Based Intrusion Detection System for Detecting IoT Botnet Attacks: A Review. *IEEE Access* **2025**, *13*, 11792–11822. [CrossRef]
11. Ali, W.; Amin, M.; Alarfaj, F.K.; Al-Otaibi, Y.D.; Anwar, S. AI-Enhanced Differential Privacy Architecture for Securing Consumer Internet of Things (CIoT) Data. *IEEE Trans. Consum. Electron.* **2025**, *71*, 5201–5215. [CrossRef]
12. Shah, Z.; Ullah, I.; Li, H.; Levula, A.; Khurshid, K. Blockchain Based Solutions to Mitigate Distributed Denial of Service (DDoS) Attacks in the Internet of Things (IoT): A Survey. *Sensors* **2022**, *22*, 1094. [CrossRef] [PubMed]
13. Alimi, O.A. Data-Driven Learning Models for Internet of Things Security: Emerging Trends, Applications, Challenges and Future Directions. *Technologies* **2025**, *13*, 176. [CrossRef]
14. Singh, N.J.; Hoque, N.; Singh, K.R.; Bhattacharyya, D.K. Botnet-Based IoT Network Traffic Analysis Using Deep Learning. *Secur. Priv.* **2024**, *7*, e355. [CrossRef]
15. Kamal, H.; Mashaly, M. Robust Intrusion Detection System Using an Improved Hybrid Deep Learning Model for Binary and Multi-Class Classification in IoT Networks. *Technologies* **2025**, *13*, 102. [CrossRef]
16. Albanbay, N.; Tursynbek, Y.; Graffi, K.; Uskenbayeva, R.; Kalpeyeva, Z.; Abilkaiyr, Z.; Ayapov, Y. Federated Learning-Based Intrusion Detection in IoT Networks: Performance Evaluation and Data Scaling Study. *J. Sens. Actuator Netw.* **2025**, *14*, 78. [CrossRef]
17. Susilo, B.; Muis, A.; Sari, R.F. Intelligent Intrusion Detection System Against Various Attacks Based on a Hybrid Deep Learning Algorithm. *Sensors* **2025**, *25*, 580. [CrossRef]
18. Alhasawi, Y.; Alghamdi, S. Federated Learning for Decentralized DDoS Attack Detection in IoT Networks. *IEEE Access* **2024**, *12*, 42357–42368. [CrossRef]
19. Sahu, A.K.; Sharma, S.; Tanveer, M.; Raja, R. Internet of Things Attack Detection Using Hybrid Deep Learning Model. *Comput. Commun.* **2021**, *176*, 146–154. [CrossRef]
20. Saba, T.; Rehman, A.; Sadad, T.; Kolivand, H.; Bahaj, S.A. Anomaly-Based Intrusion Detection System for IoT Networks through Deep Learning Model. *Comput. Electr. Eng.* **2022**, *99*, 107810. [CrossRef]
21. Villegas Ch, W.; Govea, J.; Maldonado Navarro, A.M.; Játiva, P.P. Intrusion Detection in IoT Networks Using Dynamic Graph Modeling and Graph-Based Neural Networks. *IEEE Access* **2025**, *13*, 65356–65375. [CrossRef]

22. Alomari, E.S.; Manickam, S.; Anbar, M. Adaptive Hybrid Deep Learning Model for Real-Time Anomaly Detection in IoT Networks. *J. Adv. Res. Des.* **2026**, *137*, 278–289.
23. Emeç, M.; Özcanhan, M.H. A Hybrid Deep Learning Approach for Intrusion Detection in IoT Networks. *Adv. Electr. Comput. Eng.* **2022**, *22*, 3–12. [CrossRef]
24. Azumah, S.W.; Elsayed, N.; Adewopo, V.; Zaghloul, Z.S.; Li, C. A Deep LSTM Based Approach for Intrusion Detection IoT Devices Network in Smart Home. In Proceedings of the 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 20–24 June 2021; pp. 836–841.
25. Shirley, J.J.; Priya, M. An Adaptive Intrusion Detection System for Evolving IoT Threats: An Autoencoder-FNN Fusion. *IEEE Access* **2025**, *13*, 4201–4217. [CrossRef]
26. Omarov, B.; Auelbekov, O.; Suliman, A.; Zhaxanova, A. CNN–BiLSTM Hybrid Model for Network Anomaly Detection in Internet of Things. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, 436–444. [CrossRef]
27. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Slay, J. Towards Developing Network Forensic Mechanism for Botnet Activities in the IoT Based on Machine Learning Techniques. In *Mobile Networks and Management, Proceedings of the 9th International Conference, MONAMI 2017, Melbourne, Australia, 13–15 December 2017*; Springer: Cham, Switzerland, 2018; pp. 30–44.
28. Ullah, I.; Mahmoud, Q.H. A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks. In *Advances in Artificial Intelligence*; Goutte, C., Zhu, X., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12109, pp. 508–520. [CrossRef]
29. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-Sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
30. Adefemi, K.O.; Mutanga, M.B. A Robust Hybrid CNN–LSTM Model for Predicting Student Academic Performance. *Digital* **2025**, *5*, 16. [CrossRef]
31. Ullah, I.; Mahmoud, Q.H. Design and Development of RNN Anomaly Detection Model for IoT Networks. *IEEE Access* **2022**, *10*, 62722–62750. [CrossRef]
32. Halbouni, A.; Gunawan, T.S.; Habaebi, M.H.; Halbouni, M.; Kartiwi, M.; Ahmad, R. CNN-LSTM: Hybrid Deep Neural Network for Network Intrusion Detection System. *IEEE Access* **2022**, *10*, 99837–99849. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.