Review of Computer Engineering Research
2026 Vol. 13, No. 4, pp. 241-267
ISSN(e): 2410-9142
ISSN(p): 2412-4281
DOI: 10.18488/76.v13i4.5372
© 2026 Conscientia Beam. All Rights Reserved.

A hybrid temporal deep learning framework using TCN–GRU and attention for intrusion detection in IoT networks

Anamul Haque Sadial[1+]
Atowar ul Islam[2]
Sankar Haloi[3]
Sikdar Md Sultan Askari[4]

[1,2]*Department of Computer Science, University of Science and Technology, Meghalaya, 9th Mile, Ri-Bhoi, Meghalaya, India.*
[1]*Email: sadialanam@gmail.com*
[2]*Email: atowar91626@gmail.com*
[3]*Department of Mathematics, Cotton University, Panbazar, Guwahati, Assam, India.*
[3]*Email: sankarcottonghy@rediffmail.com*
[4]*Department of Computer Science and Engineering, Rajiv Gandhi University, Arunachal Pradesh, India.*
[4]*Email: sikdar.askar@rgu.ac.in*



(+ Corresponding author)

Article History
Received: 9 March 2026
Revised: 22 July 2026
Accepted: 11 August 2026
Published: 20 November 2026

Keywords
Anomaly detection
Cybersecurity
Deep learning
Internet of things
Intrusion detection system
Lightweight security models
Network traffic analysis.

ABSTRACT

The swift expansion of Internet of Things applications has significantly increased the number and complexity of cyberattacks, creating a pressing need for reliable and efficient intrusion detection systems. This study aims to develop a lightweight hybrid deep learning framework that improves intrusion detection. The proposed model combines TCN, GRU, and an attention mechanism to learn important temporal features from network traffic. To address class imbalance in the BoT-IoT dataset, SMOTE is applied during training to improve the representation of minority attack classes. The proposed framework is evaluated using standard classification metrics, including accuracy, precision, recall, F1-score, and computational efficiency. Experimental results demonstrate that the model achieves 99.92% accuracy, along with high precision and F1-score, while requiring only 15K trainable parameters, making it computationally efficient. The findings indicate that combining temporal feature extraction with attention-based feature weighting improves the detection of diverse IoT attacks without increasing model complexity. The proposed framework offers a practical and scalable solution for deployment in resource-constrained IoT environments, where accurate and efficient real-time intrusion detection is essential.

Contribution/Originality: This study contributes to the existing literature by proposing a hybrid TCN-GRU-Attention framework for IoT intrusion detection. This study uses an integrated temporal learning methodology. The paper's primary contribution is finding that the proposed model improves attack-detection accuracy and reduces false alarms while maintaining computational efficiency.

1. INTRODUCTION

The rapid advancement of modern computing, wireless communication, embedded systems, and sensing technologies has accelerated the widespread adoption of the Internet of Things (IoT). IoT enables interconnected smart devices to communicate, exchange information, and perform automated operations with limited human intervention. These intelligent systems are increasingly deployed in healthcare monitoring, industrial automation, smart transportation, smart agriculture, intelligent surveillance, and smart city infrastructure due to their ability to improve efficiency, scalability, and real-time decision-making capabilities [1-3]. Despite these advantages, the

© 2026 Conscientia Beam. All Rights Reserved.

exponential growth of IoT devices has introduced serious cybersecurity concerns, as many connected devices operate in open, heterogeneous network environments with limited built-in security protections.

The distributed and resource-constrained nature of IoT networks makes them susceptible to a wide range of cyberattacks, including distributed denial-of-service (DDoS) attacks, botnet activities, malware injection, ransomware, spoofing, and zero-day intrusions [4-6]. Attackers frequently exploit insecure communication protocols, weak authentication mechanisms, and software vulnerabilities to compromise IoT devices and disrupt network services. Furthermore, IoT devices generally possess limited computational power, memory capacity, storage resources, and battery life, making the deployment of conventional security mechanisms difficult in practical environments [7, 8]. As a result, designing lightweight, intelligent, and adaptive intrusion detection systems has become a critical research area in modern IoT security.

Intrusion detection systems (IDSs) play an important role in identifying malicious activities and unauthorized access attempts within network infrastructure. Conventional signature-based IDS approaches are capable of detecting previously known attack patterns with relatively high precision; however, they fail to identify new or evolving threats that do not match predefined signatures. To overcome this limitation, anomaly-based intrusion detection methods using machine learning (ML) techniques have received considerable attention because they can learn normal network behaviour and detect abnormal traffic patterns categorised into network-based intrusion detection systems (NIDS) and host-based intrusion detection systems (HIDS). Nevertheless, the effectiveness of IDS models in IoT environments is significantly affected by heterogeneous traffic distributions, high-dimensional feature spaces, noisy data, and severe class imbalance problems observed in real-world network datasets [1, 5, 9].

Previous intrusion detection research has primarily relied on conventional machine learning techniques such as decision Trees (DT), Random Forests (RF), Support Vector Machines (SVM),k-Nearest Neighbour (KNN), Naïve Bayes, and clustering algorithms for attack classification [2, 10, 11]. Although these approaches achieved satisfactory performance in relatively controlled environments, they often struggled to perform effectively in large-scale and dynamic IoT networks due to their limited capability in extracting complex non-linear relationships from sequential traffic data [9, 12, 13]. To improve performance evaluation and benchmarking consistency, publicly available datasets such as CICIDS2017 [14], UNSW-NB15 [15], and BoT-IoT [16] have been extensively utilised in IDS research. However, many existing studies address feature extraction, attack detection, and class imbalance independently, resulting in limited robustness when handling highly imbalanced IoT traffic distribution and minority attack categories [1, 7, 17].

To address the shortcomings, recent studies have increasingly adopted (DL)-based intrusion detection approaches because of their ability to automatically learn hierarchical and abstract feature representations directly from raw network traffic [1, 18, 19]. Deep learning architectures such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-term Memory (LSTM), and Gated Recurrent Units (GRU) have demonstrated promising results in capturing complex spatial and temporal dependencies in IoT traffic patterns [12, 20, 21]. In particular, hybrid deep learning frameworks that combine convolutional and recurrent architectures have shown improved intrusion detection performance by simultaneously learning local spatial correlations and long-term sequential behaviours from network traffic data [18, 20, 22].

Among recurrent architectures, Gated Recurrent Units (GRUs) have gained significant attention because they provide performance comparable to traditional recurrent networks while requiring fewer trainable parameters and lower computational complexity, making them highly suitable for resource-constrained IoT devices [7, 21, 23]. Furthermore, attention mechanisms have recently emerged as an effective strategy for improving feature learning by enabling deep learning models to focus selectively on the most informative parts of sequential data [5, 23, 24]. Attention-based frameworks can significantly enhance intrusion detection accuracy by prioritising critical temporal patterns associated with malicious network behaviour while suppressing irrelevant information.

Despite substantial progress in intelligent intrusion-detection research, several limitations remain unresolved.

© 2026 Conscientia Beam. All Rights Reserved.

First, many existing IDS frameworks rely independently on either convolutional architectures or recurrent learning models, limiting their ability to simultaneously capture fine-grained local traffic patterns and long-range dependencies present in IoT network data [12, 18].

Second, severe class imbalance remains a major challenge in datasets such as BoT-IoT, where minority attack classes are underrepresented, causing biased model learning and reduced detection performance for rare attacks [1, 16, 17].

Third, although attention mechanisms have demonstrated promising results, relatively few studies have integrated attention-based feature prioritization with Temporal Convolutional Networks (TCN) and GRU architecture within a unified hybrid framework for IoT intrusion detection [5, 23, 24].

Additionally, many existing deep learning models involve high computational overhead and memory consumption, limiting their deployment feasibility in lightweight IoT environments [7, 25].

In response to these research challenges, the present study is guided by the following research questions:

RQ1: How can temporal convolution and recurrent learning be effectively integrated to improve intrusion-detection performance in IoT traffic?

RQ2: How can attention mechanisms enhance the identification of critical attack-related temporal features?

RQ3: What is the impact of addressing class imbalance on the detection performance of minority attack classes?

RQ4: Can a lightweight hybrid deep-learning model achieve high detection accuracy while remaining computationally efficient for resource-constrained environments?

To address these questions, this study proposes a hybrid intrusion detection framework that integrates Temporal Convolutional Networks (TCN), Gated Recurrent Units (GRU), and an attention mechanism for intelligent IoT attack detection. The TCN component is employed to capture long-range temporal dependencies and parallel sequential patterns efficiently, while the GRU layer models dynamic sequential behaviour with reduced computational complexity.

The attention mechanism further enhances feature learning by assigning greater importance to relevant temporal features associated with malicious activities. In addition, the Synthetic Minority Over-sampling Technique (SMOTE) is applied to mitigate the class imbalance problem within the BOT-IoT dataset and improve minority attack classification performance. The proposed model is evaluated using standard performance metrics such as accuracy, precision, recall, F1-score, and confusion matrix analysis, and its performance is compared with existing machine learning and deep learning–based intrusion detection approaches to demonstrate its effectiveness and robustness.

Major contributions of the proposed research work are:

1. Proposed an intelligent Hybrid IDS architecture

This research introduces a hybrid deep learning–based intrusion detection framework by combining Temporal Convolutional Networks (TCN), Gated Recurrent Units (GRU), and an attention mechanism for accurate and adaptive IoT attack detection. The integration of these techniques enables efficient extraction of both temporal and sequential traffic characteristics.

2. Improved Temporal Dependency Learning in IoT Traffic

The proposed framework effectively captures long-range temporal relationships and hidden sequential dependencies present in IoT network traffic. The TCN component enhances parallel feature-extraction capability, while GRU efficiently models dynamic behavioural patterns with reduced computational complexity.

3. Enhanced Detection Accuracy Using Attention -Based Learning

An attention mechanism is incorporated to identify and prioritise the most relevant attack-related features from large volumes of network traffic data. This selective feature learning improves the detection of complex and evolving cyberattacks while minimising the influence of irrelevant information.

© 2026 Conscientia Beam. All Rights Reserved.

4. Address severe Class imbalance Problem

To overcome biased model learning caused by highly imbalanced IoT datasets, the Synthetic Minority Oversampling Technique (SMOTE) is employed. The balancing process significantly improves minority attack classification and strengthens the model's overall detection capability.

5. Designed a Lightweight and Computationally Efficient Model

Unlike many existing deep learning IDS frameworks with high computational overhead, the proposed architecture is designed to remain lightweight with fewer trainable parameters. This makes the model more suitable for deployment in real-time and resource-constrained IoT environments.

6. Comprehensive Experimental Validation on BoT -IoT Dataset

The effectiveness of the proposed framework is validated using the benchmark BoT-IoT dataset through multiple evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrix analysis to ensure reliable performance assessment.

7. Improved Detection of Diverse IoT Cyberattacks

The proposed model demonstrates strong capability in identifying multiple categories of malicious activities, including DDoS attacks, botnet traffic, spoofing behaviour, and anomalous network activities commonly observed in IoT ecosystems.

8. Comparative Performance Analysis with Existing Models

The study conducts a detailed comparative analysis of existing machine learning- and deep learning–based intrusion detection approaches to demonstrate the superiority of the proposed framework in detection performance, robustness, and computational efficiency.

9. Contribution Towards Secure and Intelligent IoT Systems

This research advances intelligent IoT cybersecurity by presenting an adaptive, scalable intrusion-detection solution that handles heterogeneous traffic patterns and evolving cyberthreats in modern smart environments.

10. Practical Applicability for Edge and Real-Time IoT Security

Due to its lightweight architecture and efficient learning capability, the proposed IDS framework offers practical deployment potential in edge computing and real-time IoT monitoring applications where low latency and reduced resource consumption are essential.

2. LITERATURE REVIEW

The rapid growth of the Internet of Things (IoT) has significantly transformed modern communication systems by enabling seamless connectivity among smart devices, sensors, cloud platforms, and intelligent applications. However, the increasing interconnection of IoT devices has also exposed networks to various cyber threats such as distributed denial-of-service (DDoS) attacks, botnets, ransomware, spoofing, and zero-day intrusions. Traditional security mechanisms are often unable to handle the heterogeneous, dynamic, and resource-constrained nature of IoT environments. Consequently, researchers have increasingly focused on machine learning and deep learning-based intrusion detection systems (IDS) to enhance IoT network security.

Previous studies primarily focused on applying machine learning techniques to IoT security. Conti et al. highlighted the growing importance of IoT security and digital forensics in connected environments and discussed the limitations of conventional security architectures in handling evolving cyberthreats [8]. Similarly, Xiao et al. examined machine-learning-driven IoT security techniques and demonstrated that intelligent algorithms could improve attack-detection accuracy and adaptive defense mechanisms [3]. These studies established the foundation for intelligent intrusion-detection research in IoT systems.

Several benchmark datasets were later introduced to support IDS model development and evaluation. Sharafaldin et al. [14] proposed the CICIDS dataset, which contains modern attack scenarios and realistic network traffic patterns suitable for evaluating intrusion detection systems Sharafaldin et al. [14]. Moustafa and Slay [15] introduced the

© 2026 Conscientia Beam. All Rights Reserved.

UNSW-NB15 dataset with diverse attack categories and balanced traffic features for cybersecurity research [15]. Furthermore, Koroniotis et al. [16] developed the Bot-IoT dataset specifically for IoT botnet traffic analysis, enabling researchers to investigate large-scale malicious IoT activities [16]. These datasets have become widely adopted in deep learning-based IDS research. With the advancement of deep learning, researchers began investigating neural network architectures for intrusion detection. Javaid et al. [26] demonstrated that deep neural networks could automatically learn complex traffic patterns and achieve higher detection accuracy compared to traditional machine learning techniques Javaid et al. [26]. Kasongo and Sun [13] further improved intrusion detection performance by integrating feature engineering with deep learning methods, thereby enhancing classification efficiency and reducing false-alarm rates Kasongo and Sun [13]. Roopak et al. [19] also explored deep learning models for cybersecurity applications in IoT networks and showed that automated feature learning significantly improved anomaly-detection capability [19]. Survey-based studies have provided comprehensive insights into the strengths and limitations of machine learning and deep learning approaches for IoT security. Al-Garadi, et al. [2] reviewed various machine learning and deep learning methods for IoT intrusion detection and identified challenges related to data imbalance, computational overhead, scalability, and feature extraction Al-Garadi, et al. [2]. Hindy et al. [6] presented a taxonomy of network threats and intrusion detection systems, categorizing attacks and detection mechanisms based on operational characteristics and deployment environments [6]. More recently, Zhang et al. [1] reviewed deep learning applications in intrusion detection systems and emphasized the importance of overcoming spatiotemporal feature-extraction limitations and class-imbalance problems in modern IDS frameworks [1].

Feature extraction and representation learning have also received considerable attention in recent research. Thaker et al. investigated feature extraction strategies for machine learning-based intrusion detection in IoT networks and reported that optimized feature representation significantly improves classification accuracy while reducing computational complexity Thaker et al. [11]. Ullah and Mahmoud [9] proposed a deep learning-based anomaly detection framework capable of identifying abnormal traffic behavior in IoT environments with improved robustness against sophisticated attacks Ullah and Mahmoud [9]. Banaamah and Ahmad [17] further demonstrated that deep learning architectures can efficiently detect malicious activities in IoT systems by learning hierarchical traffic representations [17].

Hybrid deep learning architectures have emerged as one of the most promising directions for IoT intrusion detection. Khan et al. [18] proposed a hybrid deep learning-based IDS for IoT networks and achieved improved detection performance by combining multiple neural learning mechanisms Khan et al. [18]. Cui et al. [5] developed an IoT-based intrusion detection system using a hybrid deep learning algorithm and reported significant improvements in attack classification efficiency and detection reliability [4]. Similarly, Cui et al. [5] introduced deep residual networks integrated with attention mechanisms to improve feature learning and attack recognition in IoT intrusion detection systems [5].

Recent studies have increasingly focused on lightweight and computationally efficient IDS models suitable for edge-enabled IoT environments. Mutleg et al. [7] analyzed the trade-off between detection accuracy and computational complexity in deep learning-based IoT intrusion detection systems and emphasized the need for resource-efficient architectures Mutleg et al. [7]. Nandanwar and Katarya [12] proposed a deep learning-enabled IDS for industrial IoT environments capable of achieving reliable intrusion detection under real-time operational constraints Nandanwar and Katarya [12]. Gupta et al. [10] conducted a comparative analysis of machine learning and deep learning models in fog-enabled IoT networks and concluded that deep learning models generally outperform traditional methods in terms of detection accuracy and adaptability [10].

Advanced neural architectures combining convolutional, recurrent, and attention-based mechanisms have shown substantial improvements in intrusion detection performance. Afraji et al. [20] introduced an integrated CNN–LSTM–GRU framework for IoT and Industrial IoT (IIoT) intrusion detection, demonstrating enhanced temporal feature extraction and improved classification stability Afraji et al. [20]. Sagu et al. [21] proposed a GRU–CNN

© 2026 Conscientia Beam. All Rights Reserved.

model trained using SUCMO optimization and reported notable improvements in attack detection efficiency and learning convergence Sagu et al. [21]. Kareem et al. [23] developed a robust GRU model with an attention mechanism and fuzzy weighting for SDN-IoT intrusion detection, achieving strong detection capability against complex network attacks [23]. These studies indicate that recurrent and attention-driven architectures are highly effective in capturing sequential traffic behavior in dynamic IoT environments.

Recent research has also explored scalable and intelligent IDS frameworks incorporating advanced optimization and language learning capabilities. Hossain [25] proposed a scalable deep learning-based IDS framework capable of handling large-scale IoT traffic while maintaining high detection efficiency and reduced processing overhead Hossain [25]. Ma et al. [24] integrated feature selection techniques with large language model fine-tuning for IoT intrusion detection and demonstrated that intelligent feature adaptation can significantly improve attack classification performance. Ma et al. [24]. Zhang [27] developed an intelligent intrusion detection system using deep learning methods to enhance adaptive learning and attack identification in modern IoT infrastructures Zhang [27]. Chithra Rani and Baalaji [22] further proposed an ensemble stacking-based deep learning model for IoT-edge platforms, achieving enhanced intrusion detection accuracy and improved model generalization [22]. Although substantial progress has been achieved in IoT intrusion detection research, several challenges remain unresolved. Existing studies frequently report issues related to class imbalance, overfitting, high computational requirements, limited generalization capability, and inadequate performance against emerging zero-day attacks. In addition, many deep learning models achieve high detection accuracy at the expense of greater complexity and resource consumption, making deployment on lightweight IoT devices and edge environments difficult. Therefore, there remains a strong need for efficient, lightweight, and hybrid deep learning frameworks capable of extracting temporal and contextual traffic features while maintaining low computational overhead and high detection reliability in real-time IoT applications. The remainder of this paper is organised as follows: Section 3 presents the proposed methodology, including dataset description, preprocessing techniques, feature engineering, and model architecture. Section 4 discusses the experimental setup, performance evaluation metrics, and comparative analysis of results. Section 5 highlights the Novelty and contribution, major findings, and limitations of the proposed framework. Finally, Section 6 concludes the paper and outlines potential directions for future research in intelligent IoT intrusion detection systems.

3. METHODOLOGY

This section describes the methodological design of the proposed Hybrid TCN-GRU-Attention Intrusion Detection System developed for securing IoT networks. The whole approach has been structured to address several practical challenges reported in various existing studies, particularly the need to capture temporal dependencies in network traffic, manage severe class imbalance, reduce false-alarm rates, and ensure that the model remains suitable for deployment in environments with limited computational resources. The proposed intrusion-detection framework is organised in a layered and modular manner so that each component performs a specific function within the overall system. Such a design allows efficient processing of network traffic while keeping computational complexity under control. Initially, raw IoT traffic data is obtained from standard benchmark datasets. This data is then passed through a sequence of preprocessing steps followed by feature learning and classification stages. Each stage is designed to progressively refine the input data and extract meaningful patterns related to normal and malicious behaviour. Figure 1 presents the overall architecture of the proposed Hybrid TCN-GRU-Attention framework.

© 2026 Conscientia Beam. All Rights Reserved.

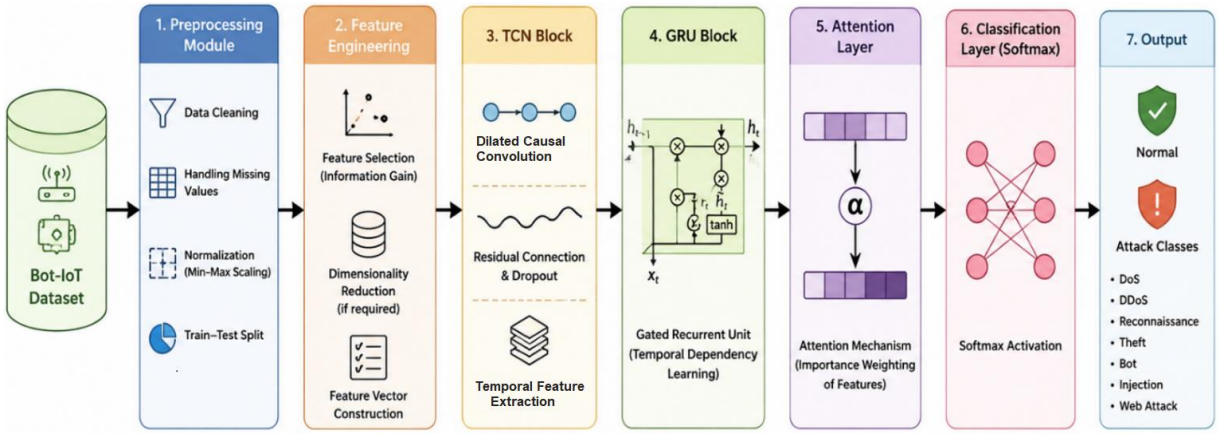


Figure 1. Overall architecture of the proposed TCN-GRU-Attention IDS.

As shown in Figure 1, the workflow begins with data acquisition and preprocessing, followed by SMOTE-based class balancing, feature learning using the TCN and GRU modules, attention-based feature refinement, and the final intrusion classification stage. The figure illustrates how information flows through each module to produce accurate and computationally efficient intrusion detection for IoT networks.

3.1. Dataset

The evaluation of the proposed framework is carried out using the BoT-IoT dataset, which is widely accepted as a realistic representation of IoT network traffic and associated attack behaviour. This dataset contains multiple categories of cyber-attacks, including distributed denial-of-service (DDoS), denial-of-service (DoS), reconnaissance activities, and information theft. A notable characteristic of this dataset is a highly imbalanced distribution, where attack instances significantly outnumber normal traffic. Such an imbalance reflects real-world IoT scenarios and presents a challenging environment for intrusion detection models. By selecting this dataset, the study ensures that the proposed approach is assessed under practical and demanding conditions rather than on artificially balanced data. Table 1 presents the main characteristics of the BoT-IoT dataset, including the traffic type, attack categories, data volume, imbalance ratio, and overall realism.

Table 1. Presents a summary of the key characteristics of the BoT-IoT dataset.

Attribute	Description
Total records	Large-scale (millions)
Traffic type	IoT network traffic
Attack classes	DDoS, DoS, Reconnaissance, Theft
Imbalance ratio	Extremely skewed
Realism	High

As shown in Table 1, the dataset contains large-scale IoT network traffic with multiple attack types and an extremely skewed class distribution, making it appropriate for validating intrusion-detection models under practical operating conditions.

3.2. Data Preprocessing

Before training, the raw network traffic data is carefully prepared to enhance both the reliability and effectiveness of the learning process. This preparation involves removing irrelevant and noisy features, treating missing or undefined values, and normalizing numerical attributes using Min-Max scaling to maintain consistency across inputs. Categorical variables are converted into numerical form through label encoding to make them suitable for model processing. Since the dataset exhibits a high degree of class imbalance, the Synthetic Minority Over-sampling

© 2026 Conscientia Beam. All Rights Reserved.

Technique (SMOTE) is applied exclusively to the training data. This approach helps in increasing the representation of minority classes while preventing any information leakage into the testing phase, thereby ensuring a fair evaluation of the model. Figure 2 presents the complete data preprocessing and SMOTE-based balancing workflow adopted in this study. As illustrated in Figure 2, the process begins with the collection of raw BoT-IoT network traffic data, followed by data cleaning to eliminate redundant records, missing values, and irrelevant features. The cleaned data are then normalized using Min-Max scaling to transform all numerical features into the range 0 to 1.

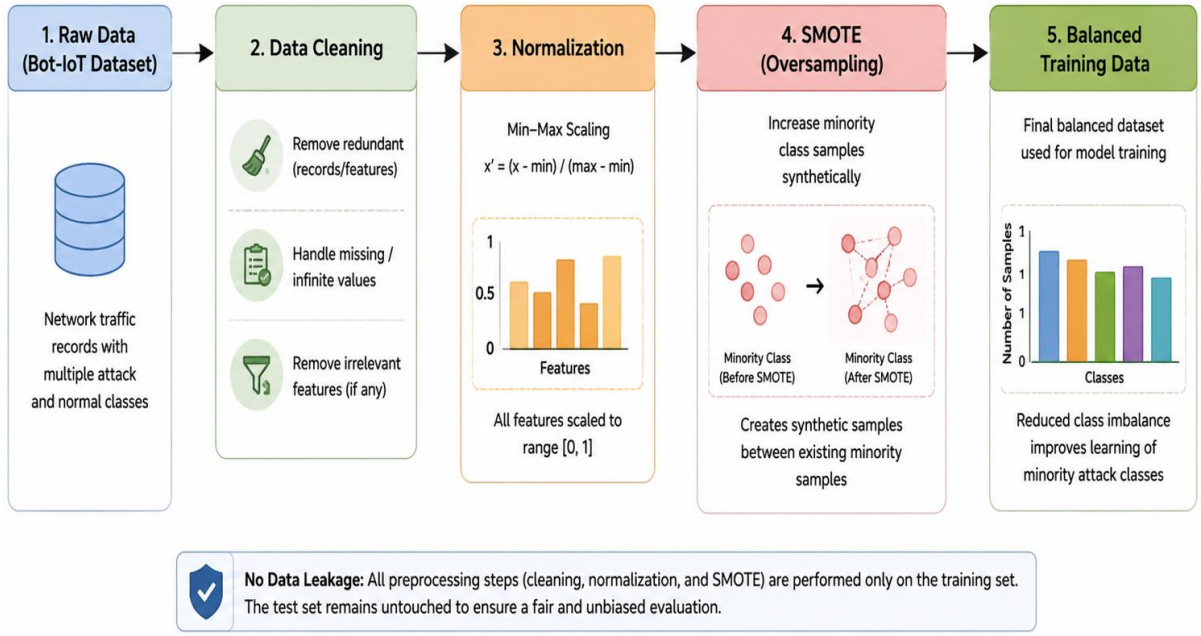


Figure 2. Data preprocessing and SMOTE-based balancing Workflow.

Figure 2 further shows that SMOTE is applied only to the training dataset to generate synthetic samples for minority attack classes, thereby reducing class imbalance without introducing information leakage. Finally, the figure presents the balanced training dataset used for model learning, while the test dataset remains unchanged to ensure an unbiased evaluation of the proposed intrusion detection framework.

3.3. Temporal Feature Learning Using TCN

In the proposed framework, Temporal Convolutional Networks (TCNs) are used as the initial deep learning components to model temporal relationships present in IoT network traffic. Unlike conventional convolutional neural networks, TCNs operate using causal convolutions, ensuring that predictions at any time step depend only on past information. In addition, the use of dilated convolutions allows the network to capture long-range dependencies without increasing computational complexity. Residual connections are also incorporated to facilitate stable training and improve information flow across layers. Together, these characteristics enable the model to learn both short-term variations and long-term patterns in network traffic, making TCNs well-suited for analyzing sequential data in intrusion detection tasks. Figure 3 presents the internal architecture of the Temporal Convolutional Network (TCN). It illustrates causal padding, stacked dilated convolution layers with increasing dilation factors, and a residual skip connection, which together enable efficient learning of short, medium, and long-term temporal dependencies from sequential IoT traffic data.

© 2026 Conscientia Beam. All Rights Reserved.

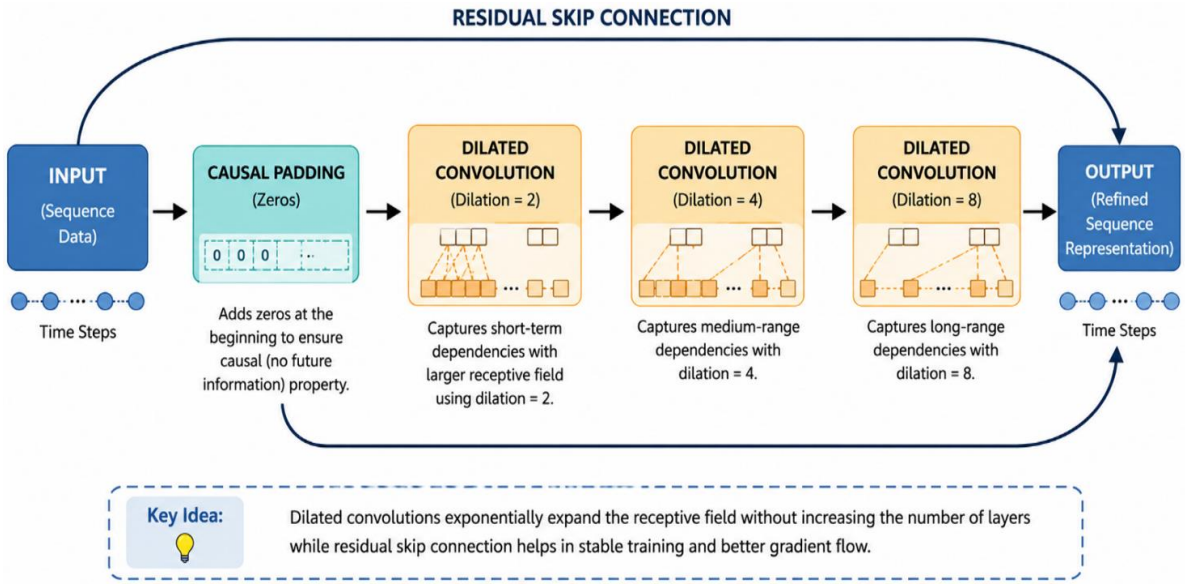


Figure 3. Internal Structure of the Temporal Convolutional Network (TCN).

3.4. Sequential Dependency Modelling with GRU

While the TCN component captures long-range temporal patterns, Gated Recurrent Units (GRUs) are incorporated to learn short-term and contextual dependencies within network traffic sequences. GRUs are designed to retain relevant information from previous time steps while efficiently updating the current state, allowing the model to better understand sequential behaviour in data. In comparison to Long Short-Term Memory (LSTM) networks, GRUs require fewer parameters, which leads to faster training and reduced memory usage. These characteristics make them particularly suitable for IoT and edge-based environments where computational resources are limited. The internal working structure of the GRU used in this framework is presented in Figure 4.

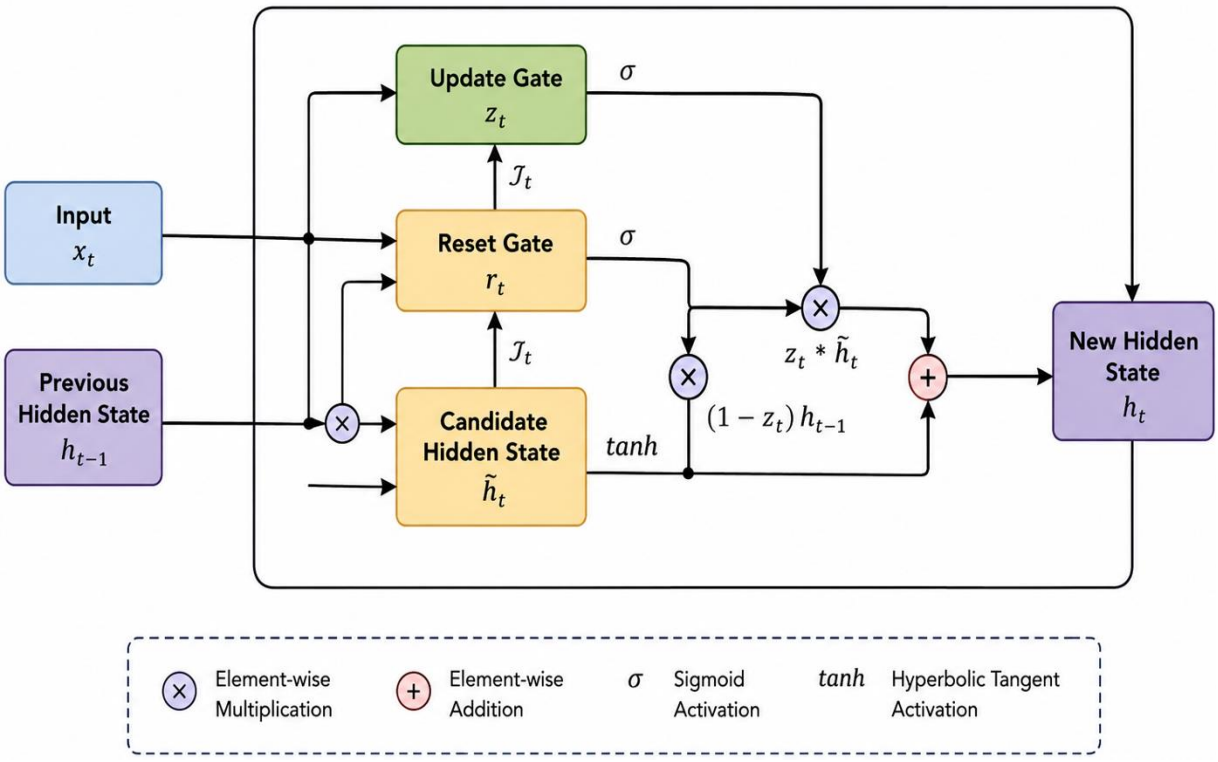


Figure 4. GRU cell structure for IoT traffic modeling.

© 2026 Conscientia Beam. All Rights Reserved.

3.5. Attention Mechanism for Feature Prioritization

Following the GRU layer, an attention mechanism is incorporated to refine the feature learning process by assigning different levels of importance to temporal features. Instead of treating all time steps equally, the model selectively emphasises those parts of the traffic sequence that carry stronger indications of malicious activity. This selective focus helps in reducing the influence of irrelevant or repetitive information while highlighting meaningful patterns associated with intrusions. As a result, the model becomes more effective in distinguishing between normal and attack behaviour, particularly in cases where minority attack classes are present. The operational concepts of this attention mechanism, as integrated with the GRU layer, are illustrated in Figure 5.

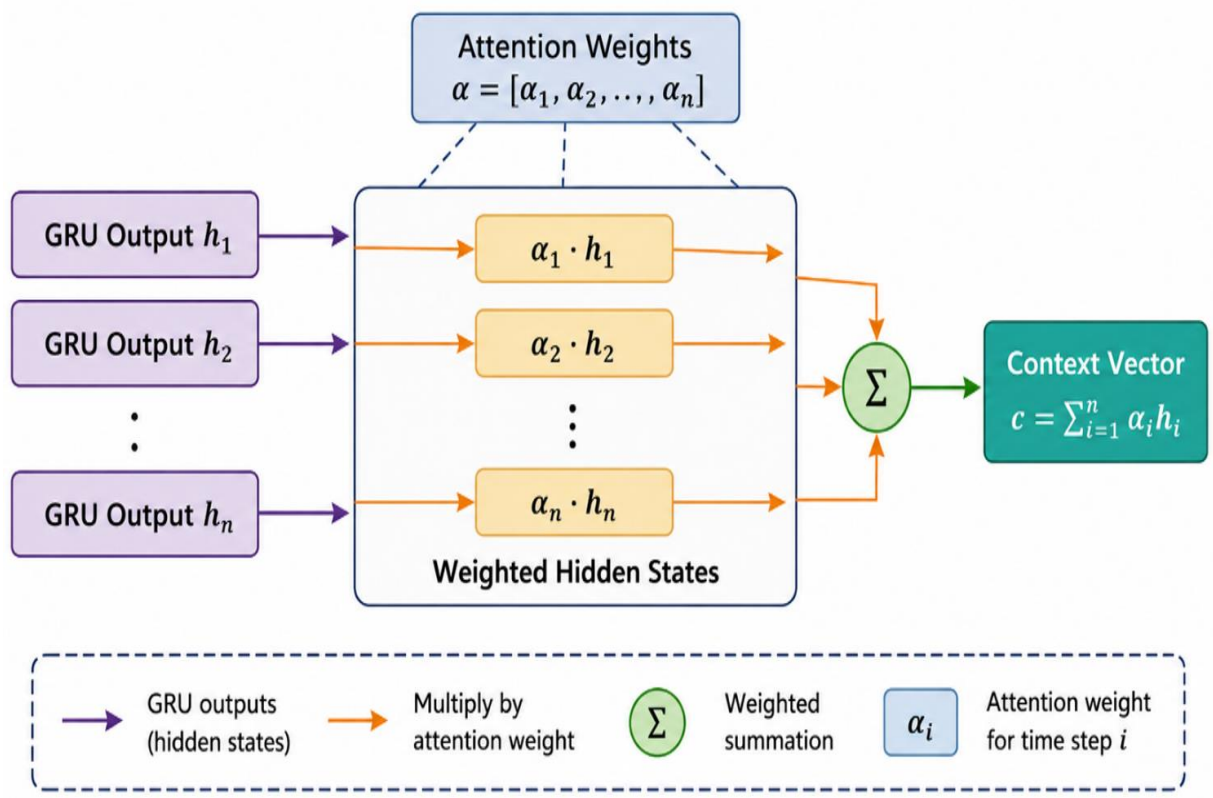


Figure 5. Attention mechanism applied to GRU outputs.

3.6. Classification and Decision Layer

The feature representation obtained after the attention stage is forwarded to a fully connected layer, where the learned patterns are combined and transformed into a suitable form for classification. This is followed by a SoftMax layer that assigns probability scores to each input instance, indicating whether the observed traffic corresponds to normal behaviour or a potential attack. The overall decision-making process, as depicted in the corresponding figure, reflects how the model translates learned features into final classification outcomes. Figure 6 presents the analytical workflow of the proposed Hybrid TCN–GRU–Attention Intrusion Detection System. It illustrates the sequential flow from pre-processed IoT data through TCN, GRU, attention, fully connected, and SoftMax layers to accurately classify network traffic as normal or malicious.

3.7. Model Training and Evaluation Metrics

The model is trained using the Adam optimizer, chosen for its stable and efficient convergence, while categorical cross-entropy is used to guide the learning process for classification. To prevent overfitting and improve generalization, early stopping is applied based on validation performance.

© 2026 Conscientia Beam. All Rights Reserved.

The model's effectiveness is evaluated using standard metrics such as accuracy, precision, recall, and F1-score, ensuring a balanced assessment, especially for minority attack classes in imbalanced IoT data. In addition, a comparative analysis with existing intrusion detection approaches is conducted to demonstrate the strengths, limitations, and improvements of the proposed methods, as summarized in Table 2.

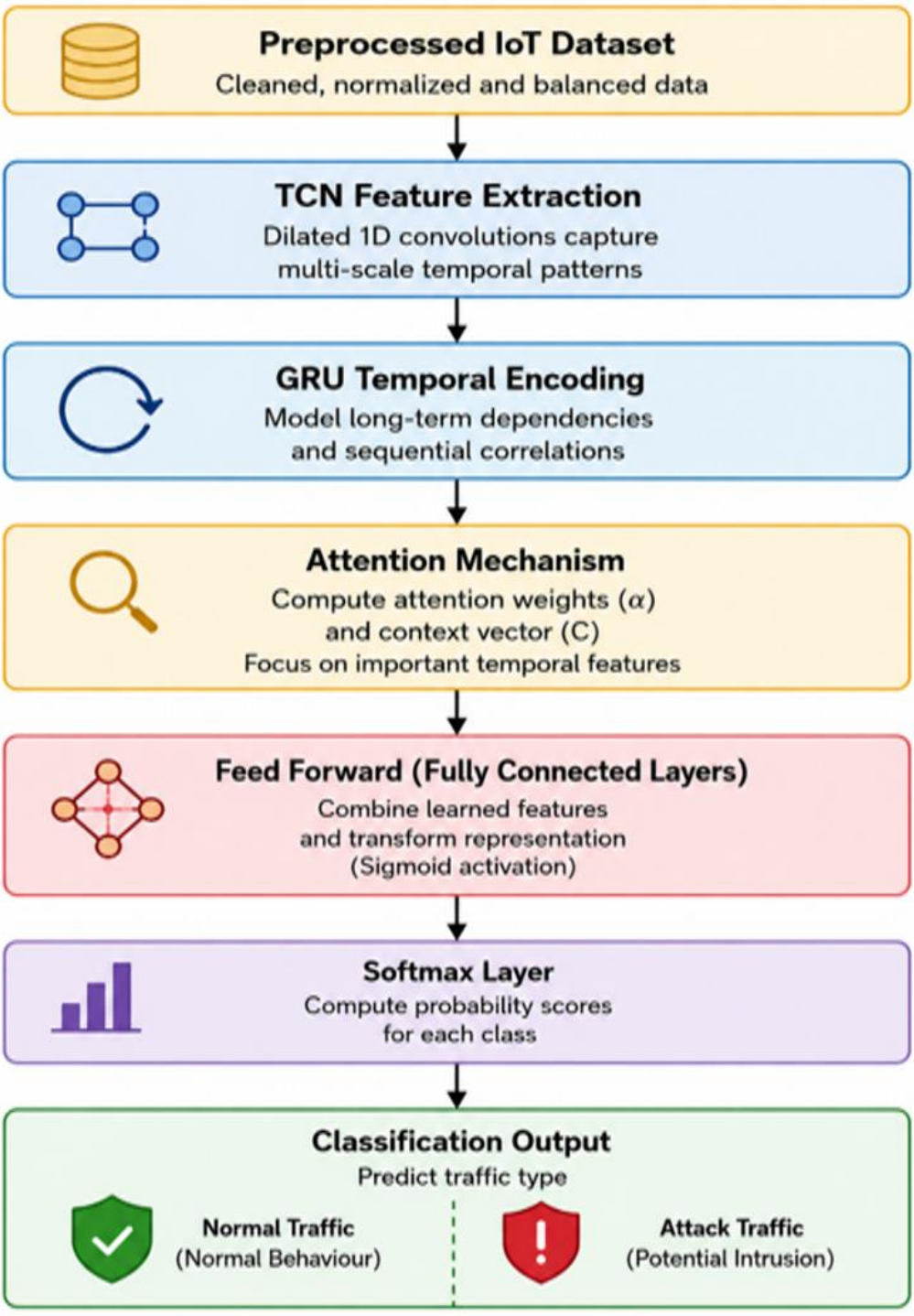


Figure 6. Analytical workflow of the proposed Hybrid TCN-GRU-Attention Intrusion Detection System.

© 2026 Conscientia Beam. All Rights Reserved.

Table 2. Comparative Analysis of Existing IDS Approaches and Proposed System.

Reference	Method	Dataset	Key Findings	Limitations	Research Gap
Zhang, et al. [1]	Deep Learning Survey	Multiple datasets	Identified challenges related to spatiotemporal learning and data imbalance	No unified detection framework has been proposed	Need for lightweight hybrid temporal IDS
Yaras and Dener [4]	Hybrid Deep Learning	IoT Traffic	Improved intrusion detection accuracy	High computational complexity	Lightweight implementation required
Cui, et al. [5]	ResNet + Attention	IoT Dataset	Enhanced feature representation and attention learning	Did not address class imbalance	Minority attack detection remains weak
Khan, et al. [18]	Hybrid Deep Learning	IoT Networks	Achieved high classification accuracy	Limited temporal dependency modelling	Long-range sequence learning needed
Afraji, et al. [20]	CNN–LSTM–GRU	IoT and IIoT	Strong sequential feature extraction	Increased model complexity	Efficient hybrid optimisation required
Mutleg, et al. [7]	Deep Learning IDS	IoT Dataset	Balanced accuracy and security analysis	High resource consumption	Lightweight IoT deployment challenge
Gupta, et al. [10]	ML vs DL Comparison	Fog-IoT	DL models outperformed ML methods	Computational overhead not analysed	Need efficient real-time IDS
Banaamah and Ahmad [17]	Deep Learning IDS	IoT Dataset	Improved attack detection performance	Limited scalability analysis	Scalability in large IoT networks
Nandanwar and Katarya [12]	DL-enabled IDS	Industrial IoT	Effective industrial intrusion detection	Lack of imbalance handling	Robust minority attack learning
Ullah and Mahmoud [9]	DL-based Anomaly Detection	IoT Traffic	Improved anomaly detection capability	Higher training complexity	Optimised lightweight architecture
Ma, et al. [24]	Feature Selection + LLM	IoT Dataset	Improved contextual learning	Large computational requirements	Resource-efficient intelligent IDS
Sagu, et al. [21]	GRU–CNN + SUCMO	IoT Traffic	Enhanced optimisation and accuracy	Model complexity remains high	Compact hybrid framework needed
Hossain [25]	Scalable Deep Learning IDS	IoT Networks	Improved scalability and efficiency	Limited temporal attention integration	Attention-guided temporal learning
Kareem, et al. [23]	GRU + Attention + Fuzzy Logic	SDN-IoT	Better attack classification accuracy	Increased architectural complexity	Lightweight unified framework
Zhang [27]	Intelligent DL-based IDS	IoT Networks	Adaptive intrusion detection	Limited imbalance mitigation	Balanced learning required
Chithra Rani and Baalaji [22]	Ensemble Stacking DL	IoT-Edge Platforms	Improved generalisation capability	High training overhead	Efficient hybrid edge deployment
Proposed System	TCN–GRU–Attention + SMOTE	BoT-IoT	High accuracy, improved minority attack detection, low false alarm rate, lightweight architecture	—	Addresses temporal learning, imbalance handling, attention-based prioritisation, and computational efficiency simultaneously

© 2026 Conscientia Beam. All Rights Reserved.

3.7.1. Problem Definition

Let the IoT network traffic datasets be represented as a collection of input-output pairs as follows:

$$D = \{(X_i, y_i)\}_{i=1}^{N} \quad (1)$$

where,

- D - represents the IoT network traffic dataset.
- N —is the total number of samples in the dataset.
- X_i -denotes the input feature vector of the ith network traffic sample.
- y_i -represents the corresponding class label or output.
- i -is the sample index, where i = 1, 2, 3…N.

The objective of the proposed intrusion detection framework is to learn a mapping function defined as:

$$f_\Theta: X \rightarrow y \quad (2)$$

Where,

- f_Θ - denotes the predictive mapping or classification function learned by the deep-learning model.
- X - represents the input traffic feature sequence provided to the model.
- y — denotes the predicted output class label generated by the model.

3.7.2. Data Balancing Using SMOTE

In intrusion detection datasets, the number of normal traffic samples is usually much larger than the number of attack samples. This unequal distribution of data is known as the class-imbalance problem. Due to this imbalance, the learning model may become biased toward the majority class and fail to correctly identify rare attack categories. To overcome this limitation, the Synthetic Minority Over-sampling Technique (SMOTE) is applied during the preprocessing stage. SMOTE artificially generates new minority-class samples instead of simply duplicating the existing records. This improves the learning capability of the model and helps in achieving better classification performance.

The minority class samples are represented as:

$$X_{\min} = (x_1, x_2, \ldots, x_m) \quad (3)$$

Where,

$X_{\min}$ - represents the complete set of minority class samples present in the training dataset. In the context of IoT intrusion detection, these samples generally correspond to rare attack instances that appear less frequently in comparison to normal traffic.

$x_1, x_2, \ldots, x_m$– denotes the individual data samples belonging to the minority class.

x_i —Refers to the i^{th} minority class sample.

m —Represents the total number of minority class instances available in the dataset.

The synthetic samples generated using SMOTE are mathematically expressed as:

$$x_{\text{new}} = x_i + \lambda(x_{nn} - x_i),\ \lambda \in (0,1) \quad (4)$$

Where,

x_{new} – represents the newly generated synthetic minority sample created by the SMOTE algorithm.

$x_i$denotes an existing minority class sample selected from the dataset.

x_{nn} – represents the nearest neighbour of xi. This neighbour is selected from the same minority class based on distance-similarity measures such as Euclidean distance.

$(x_{nn} - x_i)$ -indicates the difference vector between the nearest neighbour sample and the selected sample. This vector determines the direction in which the new synthetic sample will be generated.

λ —is a random interpolation factor used to control the position of the newly generated sample between $x_i$and x_{nn}.

$\lambda \in (0,1)$-means that the value of λ lies between 0 and 1. Since the value remains within this interval, the generated sample is created along the line segment connecting x_i and x_{nn}.

© 2026 Conscientia Beam. All Rights Reserved.

- The expression $x_i + \lambda(x_{nn} - x_i)$ – performs interpolation between two minority samples to generate a realistic artificial datapoint rather than an exact duplicate.

After applying SMOTE, the dataset becomes more balanced, which can be represented as:

$$| X_{major} | \approx | X_{minor} | \quad (5)$$

Where,

- $| X_{major} |$ – represents the total number of majority-class samples in the dataset. These are usually normal traffic records or highly frequent classes.
- $| X_{minor} |$ – represents the total number of minority-class samples after synthetic oversampling.
- The symbol $|\ \ |$ – denotes the size or cardinality of a dataset.
- The symbol $\approx$ indicates that the number of majority and minority samples becomes approximately equal after balancing.

3.7.3. Temporal Feature Extraction Using TCN

To effectively analyze sequential IoT network traffic data, a Temporal Convolutional Network (TCN) is used in the proposed framework. TCN is designed to capture long-range temporal relationships and sequential patterns from time-series traffic records. Unlike conventional convolutional networks, TCN applies dilated convolution operations, which allow the model to observe a wider temporal context without significantly increasing computational complexity. This helps the model learn both short-term and long-term behavioural patterns present in IoT traffic flows.

The dilated convolution operation is mathematically expressed as:

$$H_t = \sum_{k=0}^{K-1} W_k \cdot X_{(t-d \cdot k)} \quad (6)$$

Where,

H_t – represents the hidden feature output generated by the TCN at time step t.

$\sum$ –denotes the summation operation, where the convolution process accumulates contributions from multiple input positions.

k – represents the kernel index used during convolution.

K – denotes the kernel size of the convolution filter. It determines the number of neighbouring input values considered during feature extraction.

W_k – represents the learnable convolution weight associated with the k^{th}kernel position.

$X_{(t-d \cdot k)}$ – represents the input sequence value at a delayed temporal position.

X – denotes the sequential input traffic data fed into the TCN model.

t – represents the current time step within the sequence.

d – denotes the dilation factor. It controls the spacing between input elements used in convolution and enables the model to capture long-range temporal dependencies.

The term $(t - d \cdot k)$ – determines the previous temporal positions selected during dilated convolution.

The residual connection introduced in the TCN is represented as:

$$Z_t = H_t + X_t \quad (7)$$

Where,

Z_t – represents the final residual output of the TCN block at time step t.

H_t – denotes the transformed feature representation obtained after convolution.

X_t –represents the original input value at time step t.

The overall output of the TCN layer is expressed as:

$$Z = TCN(X) \quad (8)$$

Where,

Z – represents the complete output feature sequence generated by the TCN layer.

© 2026 Conscientia Beam. All Rights Reserved.

$TCN(X)$ – denotes the transformation performed by the Temporal Convolutional Network on the input sequence X.

X – represents the original sequential IoT traffic input data.

3.7.4. Sequential Modelling Using GRU

After temporal feature extraction using TCN, the resulting sequence is processed using a Gated Recurrent Unit (GRU). GRU is a recurrent neural network architecture designed to capture sequential dependencies efficiently while reducing the vanishing-gradient problem commonly observed in traditional recurrent networks. GRU uses gating mechanisms to selectively retain or discard information from previous time steps.

The update gate is defined as:

$$z_t = \sigma(W_z Z_t + U_z h_{t-1}) \qquad (9)$$

Where,

z_t – represents the update gate vector at time step t.

σ – denotes the sigmoid activation function, which produces values between 0 and 1.

W_z – represents the weight matrix associated with the current input sequence.

Z_t – denotes the TCN output at time step t.

U_z – represents the recurrent weight matrix connected to the previous hidden state.

h_{t-1} – denotes the hidden state from the previous time step.

The update gate determines how much information from the previous hidden state should be retained.

The reset gate is mathematically represented as:

$$r_t = \sigma(W_r Z_t + U_r h_{t-1}) \qquad (10)$$

Where,

r_{t-} represents the reset gate vector at time step t.

W_r- denotes the input-weight matrix for the reset gate.

U_r – represents the recurrent weight matrix associated with previous hidden information.

h_{t-1} – denotes the hidden state from the previous time step.

The reset gate controls how much previous information should be ignored while generating the candidate hidden state.

The candidate hidden state is computed as:

$$\tilde{h}_t = tanh\ (W_h Z_t + U_h(r_t \odot h_{t-1})) \qquad (11)$$

Where,

$\tilde{h}_t$ – represents the candidate hidden state generated at time step t.

$tan\,h$ –denotes the hyperbolic tangent activation function, which maps values between $-1$and 1.

W_{h-}represents the input weight matrix used for hidden state generation.

U_h – denotes the recurrent weight matrix.

$r_t \odot h_{t-1-}$ represents element-wise multiplication between the reset gate and the previous hidden state.

$\odot$ – denotes the Hadamard product, or element-wise multiplication operation.

The final hidden state of the GRU is expressed as:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \qquad (12)$$

Where,

h_t –represents the final hidden state at time step t.

$(1-z_t)$ – determines the proportion of previous information retained.

z_{t-} Controls the amount of newly generated information incorporated into the current state.

h_{t-1-} denotes the previous hidden state.

$\tilde{h}_t$ – Represents the candidate hidden state.

© 2026 Conscientia Beam. All Rights Reserved.

$\odot$ – denotes element-wise multiplication.

The sequence of hidden states generated by the GRU is represented as:

$$H = (h_1, h_2, \dots, h_T) \qquad (13)$$

Where,

H – represents the complete sequence of hidden states generated by the GRU.

$h_1, h_2, \dots, h_T$ – denote hidden states from the first to the final time step.

T – represents the total sequence length or total number of time steps.

The GRU effectively models sequential behavioural patterns within IoT traffic flows and preserves important temporal information for intrusion detection.

3.7.5. Attention Mechanism

To identify the most informative temporal features, an attention mechanism is applied over the GRU hidden states. The attention layer assigns higher importance to significant time steps that contribute more strongly to intrusion detection.

- The relevance score for each hidden state is calculated as:

$$e_t = v^{\top} tanh\,(W_a h_t + b_a) \qquad (14)$$

Where,

e_t – represents the attention relevance score at time step t.

$v^{\top}$ – denotes the transpose of the trainable attention weight vector.

$tanh$ – represents the hyperbolic tangent activation function.

W_a – denotes the attention weight matrix.

h_{t-n} represents the GRU hidden state at time step t.

b_a – represents the bias vector associated with the attention layer.

The attention weights are computed using the softmax function.

$$\alpha_t = \frac{exp\,(e_t)}{\sum_{j=1}^{T} exp\,(e_j)} \qquad (15)$$

Where,

α_t – represents the normalized attention weight assigned to the hidden state at time step t .

$exp(e_t)$ – denotes the exponential transformation of the relevance score.

$\sum_{j=1}^{T} exp\,(e_j)$ – calculates the total exponential score across all time steps.

T – denotes the total sequence length.

The SoftMax function converts raw attention scores into probability values whose total sum equals 1.

The context vector is then computed as:

$$c = \sum_{t=1}^{T} \alpha_t h_t \qquad (16)$$

Where,

c – represents the context vector generated by the attention mechanism.

α_t – denotes the attention weight assigned to the hidden state h_t.

h_t – Represents the GRU hidden state at time step t.

The weighted summation operation combines important temporal features into a single representative vector.

3.7.6. Classification Layer

The final classification is performed using a SoftMax output layer. The context vector generated by the attention mechanism is transformed into class-probability scores.

© 2026 Conscientia Beam. All Rights Reserved.

The classification process is expressed as follows:

$$\hat{y} = Softmax(W_c c + b_c) \qquad (17)$$

Where,

- $\hat{y}$ − represents the predicted output probability vector.
- $So\,ftmax(\cdot) - denotes\ the\ software\ activation\ function\ used\ for\ multiclass\ classification.$
- W_c −represents the classification weight matrix.
- c −denotes the context vector generated by the attention layer.
- b_c −represents the bias term used in the classification layer.
- The SoftMax function converts output scores into probability values corresponding to different attack classes.

3.7.7. Loss Function

To optimize the learning process, the model uses the categorical cross-entropy loss function. This loss function measures the difference between the actual class labels and predicted probability values.

The categorical cross-entropy loss is expressed as:

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \qquad (18)$$

Where:

- L −denotes the categorical cross-entropy loss.
- C −represents the total number of output classes.
- i −Denotes the class index.
- y_i_is the ground-truth label for class i, where $y_i \in \{0,1\}$.
- $\hat{y}_i$ −is the predicted probability of class i, where $0 \leq \hat{y}_i \leq 1$.
- $\log(\hat{y}_i)$ −represents the natural logarithm of the predicted probability.
- The negative sign ensures that the loss remains non-negative because $\log(\hat{y}_i)$is non-positive for probability values in the range $0 < \hat{y}_i \leq 1$.

The categorical cross-entropy loss minimizes prediction error and improves the classification capability of the proposed IoT intrusion detection framework.

3.8. Model Integration and Learning

The proposed intrusion detection framework integrates Temporal Convolutional Networks (TCN), Gated Recurrent Units (GRU), and an attention mechanism in a sequential manner to effectively analyze IoT network traffic. Initially, the TCN captures both short-term variations and long-range temporal patterns using causal and dilated convolutions. The extracted features are then processed by the GRU to model sequential dependencies efficiently with reduced computational cost. An attention layer is subsequently applied to highlight the most relevant temporal features, improving the quality of learned representations. Finally, the refined features are passed through fully connected layers with a sigmoid function for classification, while training is optimized using Adam along with regularization techniques to ensure robust performance.

3.9. Classification Metrics

To evaluate the effectiveness of the proposed hybrid intrusion detection framework under imbalanced IoT traffic conditions, multiple classification metrics are employed. These performance measures help analyze the model from different perspectives, including prediction accuracy, attack detection capability, and minority class recognition performance.

The accuracy of the model is computed as:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \qquad (19)$$

© 2026 Conscientia Beam. All Rights Reserved.

Where,

- TP – represents true positives. These are attack samples correctly classified as attacks.
- TN – represents true negatives. These are normal traffic samples correctly classified as normal traffic.
- FP – represents false positives. These are normal samples incorrectly classified as attacks.
- FN – represents false negatives. These are attack samples incorrectly classified as normal traffic.
- Accuracy measures the overall proportion of correctly classified samples among all observations.

The precision metric is defined as:

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (20)$$

Where,

- Precision measures how accurately the model identifies malicious traffic among all predicted attack instances.
- TP – Represents correctly predicted attack samples.
- FP – Represents normal traffic samples incorrectly identified as attacks.
- A higher precision value indicates fewer false alarm predictions.

The recall metric is expressed as:

$$\text{Recall} = \frac{TP}{TP+FN} \qquad (21)$$

Where,

- Recall measures the ability of the model to correctly detect actual attack instances.
- TP- denotes correctly identified attack traffic.
- FN – denotes attack traffic incorrectly classified as normal traffic.
- High recall indicates strong attack detection capability with fewer missed intrusions.

The F1 score is calculated as:

$$\text{F1-score} = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \qquad (22)$$

Where,

- The F1-score provides a balanced evaluation by combining both precision and recall into a single metric.
- Precision represents prediction exactness.
- Recall represents detection completeness.
- The multiplication factor 2 ensures harmonic balancing between precision and recall.
- The F1-score becomes high only when both precision and recall are high.

Together, these evaluation metrics provide a comprehensive analysis of the proposed intrusion detection model by considering overall classification performance as well as minority attack detection capability.

Algorithm 1: Hybrid TCN–GRU–Attention Intrusion Detection Model

Input:

Pre-processed dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$, sequence length T, number of features F, learning rate η, batch size B, and number of epochs E

Output:

Trained intrusion detection model M capable of classifying IoT traffic into normal and attack categories.

Steps:

Step 1: Load the pre-processed IoT network traffic dataset D and perform data normalization, encoding, and feature preparation.

Step 2: Apply the Synthetic Minority Over-sampling Technique (SMOTE) to balance minority attack classes and generate a balanced dataset.

© 2026 Conscientia Beam. All Rights Reserved.

Step 3: Reshape the processed dataset into a tensor representation of size (N, T, F), where N denotes the number of samples, T represents sequence length, and F indicates feature dimension.

Step 4: Split the balanced dataset into training, validation, and testing datasets for model learning and performance evaluation.

Step 5: Initialise the trainable model parameters Θ using Xavier initialisation and configure the Adam optimiser with a learning rate η.

Step 6: Pass the input sequence through the Temporal Convolutional Network (TCN) layer using dilation factors $d \in \{1,2\}$to extract temporal traffic features through causal one-dimensional convolution.

Step 7: Apply batch normalization and Rectified Linear Unit (ReLU) activation to improve feature stability and introduce nonlinearity.

Step 8: Introduce residual connections by combining extracted temporal features with the original input sequence to preserve information and improve gradient flow.

Step 9: Apply dropout regularisation with probability $p = 0.45$ to reduce overfitting during training.

Step 10: Feed the extracted temporal features into the Gated Recurrent Unit (GRU) layer with 32 hidden units to model sequential dependencies.

Step 11: Update the GRU hidden state iteratively at each time step using update gates, reset gates, and candidate hidden state representations.

Step 12: Compute attention scores for each hidden state and assign importance weights using the softmax function.

Step 13: Generate the context vector by computing the weighted sum of hidden states to retain important temporal information.

Step 14: Pass the context vector through a fully connected layer and apply the ReLU activation function to transform learned feature representations.

Step 15: Generate the final prediction output using a sigmoid activation function to classify network traffic as normal or malicious.

Step 16: Compute the focal loss function with an additional false-negative penalty to improve minority attack detection capability.

Step 17: Optimise model parameters using the Adam optimisation algorithm for Eepochs through forward propagation, backpropagation, and parameter updates.

Step 18: Monitor validation ROC–AUC performance and apply early stopping if no performance improvement is observed.

Step 19: Evaluate the trained model on the testing dataset using accuracy, precision, recall, F1 score, and ROC–AUC metrics.

Step 20: Return the trained intrusion detection model M for IoT traffic classification.

The final trained model M is obtained, achieving optimal validation performance and is capable of accurately detecting intrusion patterns.

Figure 7 presents the proposed Hybrid TCN–GRU–Attention intrusion detection workflow, illustrating data preprocessing, SMOTE-based balancing, hybrid feature extraction, attention-guided learning, and performance evaluation. The study employs a hybrid TCN–GRU–Attention framework to analyze IoT network traffic with improved temporal learning and feature prioritization. The BOT-IoT dataset is preprocessed through cleaning, normalization, and encoding, followed by an 80:20 train–test split with stratified sampling. To address class imbalance, SMOTE is applied only to the training data. The TCN extracts long-range temporal patterns, the GRU models sequential dependencies, and the attention layer highlights important features. Model performance is validated using cross-validation and tested on unseen data, ensuring robustness and generalization beyond a single experimental setup.

© 2026 Conscientia Beam. All Rights Reserved.

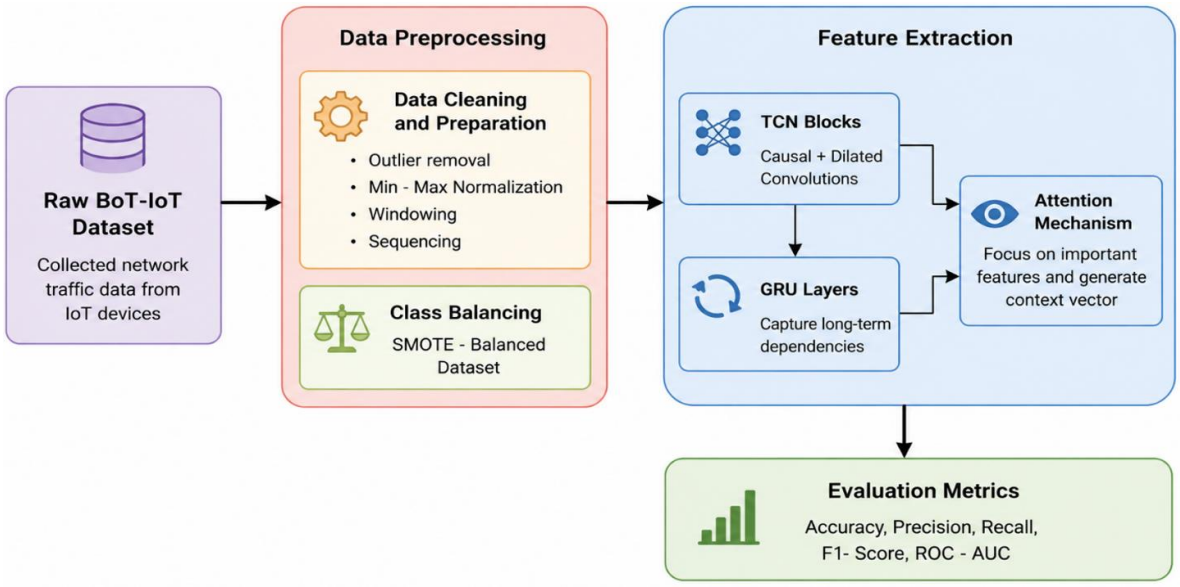


Figure 7. Proposed Flowchart of the Hybrid TCN–GRU–Attention Intrusion Detection System.

4. RESULTS AND DISCUSSION

This section presents a comprehensive evaluation of the proposed TCN–GRU–Attention-based intrusion detection framework using the Bot-IoT dataset. The Bot-IoT dataset is widely recognized as an effective benchmark for IoT intrusion detection research because it contains realistic network traffic behaviour, multiple attack categories, and highly imbalanced class distributions. The analysis primarily focuses on the proposed model's capability to address class imbalance, achieve high detection accuracy, maintain classification stability, and outperform existing machine-learning and deep-learning-based intrusion detection approaches.

A major challenge associated with the Bot-IoT dataset is the severe imbalance between normal and attack traffic classes. As presented in Table 3, attack traffic constitutes 739,735 samples (98.82%), whereas normal traffic contains only 8,834 samples (1.18%). Such an imbalanced distribution can significantly bias the learning process toward the majority class, resulting in poor minority class detection and increased false alarm rates. Similar observations regarding imbalance-related performance degradation have been reported in IoT intrusion detection.

Table 3. Initial Class Distribution in the Bot-IoT Dataset.

Class	Count	Percentage
Attack	739,735	98.82%
Normal	8,834	1.18%
Total	748,569	100%

Nevertheless, in the present study, the integration of SMOTE with the hybrid TCN–GRU–Attention architecture significantly improves minority-class learning without degrading overall model stability. The effect of SMOTE-based balancing is illustrated in Figure 8, which compares network-traffic distributions before and after oversampling. The figure clearly demonstrates that the imbalance problem is substantially reduced, leading to improved training stability and balanced classification behaviour.

4.1. Overall Detection Performance

The proposed TCN-GRU Attention model is evaluated using an unseen testing dataset consisting of 224,571 network traffic instances. The obtained performance metrics are summarized in Table 4.

© 2026 Conscientia Beam. All Rights Reserved.

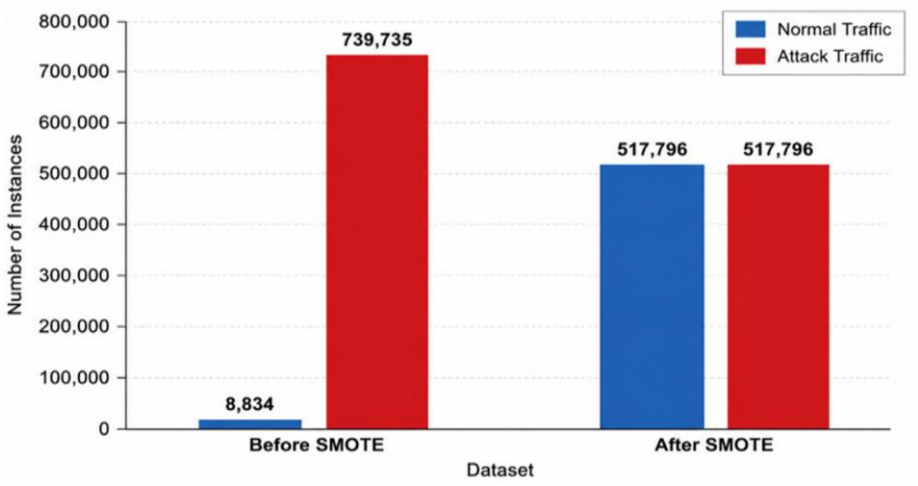


Figure 8. Distribution of network traffic instances before and after SMOTE.

Table 4. Overall Performance Metrics of the Proposed Model.

Metric	Value
Accuracy	99.92%
Precision (Attack)	100.00%
Recall (Attack)	99.92%
F1-Score (Attack)	99.96%
ROC–AUC	99.99%
Test Loss	0.0029

The experimental results demonstrate that the proposed framework achieves exceptionally high intrusion detection performance. The precision value of 100% indicates that no legitimate traffic samples are incorrectly classified as attacks, which is highly important for practical IoT deployments where false alarms can interrupt network operations and increase administrative overhead. Furthermore, the very low test loss indicates stable convergence and strong generalization capability. However, unlike the present study, their models involved comparatively larger architectures with higher computational requirements. In contrast, the proposed framework achieves competitive performance while maintaining lower computational complexity and improved suitability for edge-assisted IoT systems.

4.2. Classification Reliability and Confusion Matrix Analysis

A more detailed understanding of the model's behaviour is obtained from the confusion matrix shown in Figure 9.

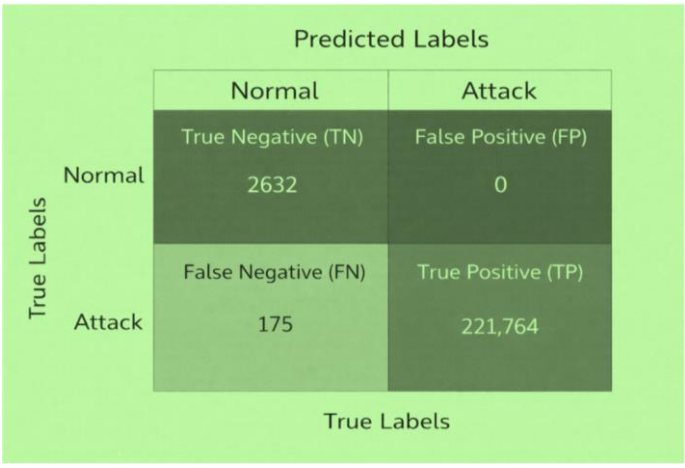


Figure 9. Comprehensive Test Set Performance (Confusion Matrix).

© 2026 Conscientia Beam. All Rights Reserved.

The model correctly identifies 221,764 attack samples (true positives) and 2,632 normal samples (true negatives). Importantly, no false positives are observed, indicating that legitimate traffic is never misclassified as malicious. Only 175 instances are recorded as false negatives, representing a very small fraction (0.08%) of missed attacks.

Based on these values, specificity and sensitivity can be expressed as follows:

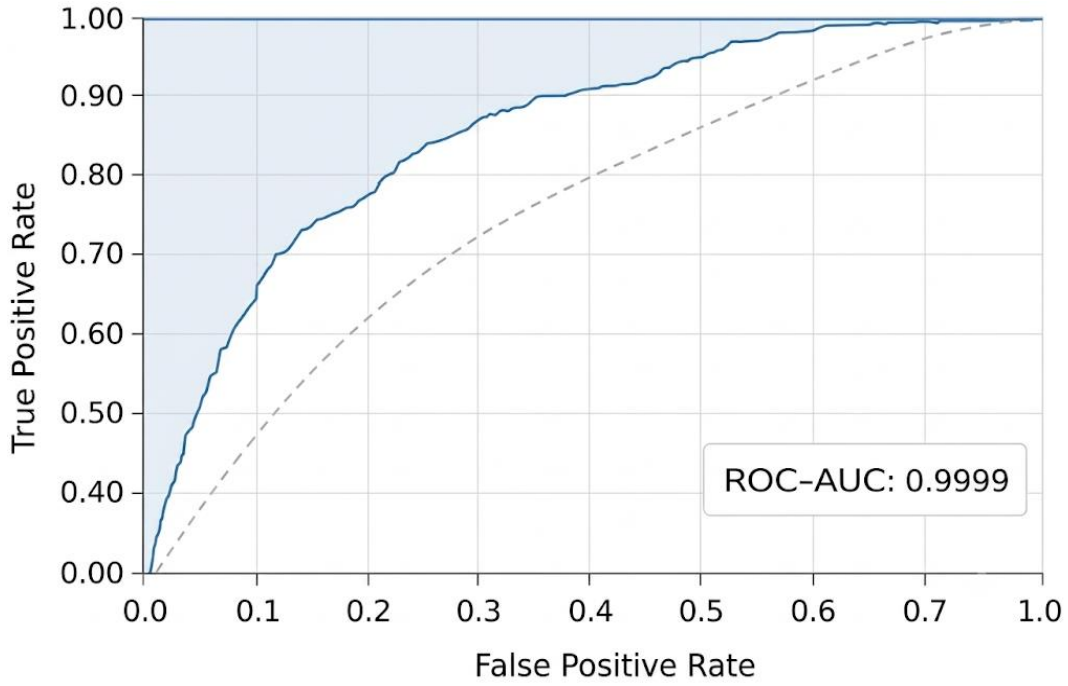
$$\text{Specificity} = \frac{TN}{TN + FP} = 100\%$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} = 99.92\%$$

These results demonstrate that the model maintains both high detection capability and reliable false-alarm control, which are critical requirements for real-time IoT security systems.

4.3. ROC–AUC Analysis

The discriminative ability of the model is further illustrated by the ROC–AUC curve presented in Figure 10. With an AUC value of 99.99%, the model achieves near-perfect separation between normal and attack classes. The curve indicates a very low false-positive rate even at high true-positive rates, confirming that the model remains stable across different classification thresholds. In contrast, the proposed TCN–GRU–Attention framework maintains stable performance even under severe class-imbalance conditions. Figure 10 presents the ROC–AUC performance of the proposed IDS, where the solid blue curve represents the model's ROC performance and the grey dotted diagonal line denotes the random-classifier baseline (AUC = 0.5).



Note: Figure 10 presents the ROC–AUC performance of the proposed IDS, where the solid blue curve represents the model's ROC performance and the grey dotted diagonal line denotes the random-classifier baseline (AUC = 0.5).

Figure 10. ROC–AUC Performance of the Proposed IDS.

4.4. Comparison with Traditional Machine Learning Models

To evaluate the effectiveness of the proposed framework, its performance is compared with that of commonly used machine learning models. The results are presented in Table 5.

© 2026 Conscientia Beam. All Rights Reserved.

Table 5. Performance Comparison with ML Baselines.

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	95.3%	96.1%	94.8%	95.4%
SVM	92.7%	93.2%	91.9%	92.5%
Logistic Regression	88.4%	89.1%	87.6%	88.3%
Proposed Model	99.92%	100.0%	99.92%	99.96%

The proposed framework consistently outperforms all baseline machine learning models across every evaluation metric. In addition to improved classification performance, the proposed model requires approximately 35 minutes of training time, which is substantially lower than SVM-based training while remaining computationally efficient for scalable IoT applications. The superior performance of the proposed architecture confirms the effectiveness of combining temporal convolution, recurrent learning, and attention mechanisms within a unified framework.

4.5. Statistical Significance and Stability Analysis

To validate model reliability and performance consistency, repeated sampling and bootstrapping techniques are applied to compute 95% confidence intervals. The results are highlighted and summarized in Table 6.

Table 6. Performance Confidence Intervals (95%).

Metric	Point Estimate	Lower Bound	Upper Bound
Accuracy	99.92%	99.90%	99.94%
Precision	100.00%	99.99%	100.00%
Recall	99.92%	99.90%	99.94%
F1-Score	99.96%	99.95%	99.97%

The narrow confidence intervals indicate that the proposed models produce highly stable and reliable predictions with minimal variation across repeated evaluations. The robustness of the framework is further strengthened through attention-weight visualisation, feature-importance analysis, and cross-validation performance assessment. An ablation study additionally confirms the contribution of each architectural component to overall intrusion-detection performance. Recent and contradictory findings from related literature have also been incorporated to position the proposed study within the broader IoT intrusion-detection research landscape.

5. NOVELTY AND CONTRIBUTIONS

The primary novelty of this study lies in the development of a unified hybrid intrusion detection framework that combines Temporal Convolutional Networks (TCN), Gated Recurrent Units (GRU), and an attention mechanism within a lightweight architecture for IoT security applications. Unlike many existing approaches that rely on standalone convolutional or recurrent models, the proposed framework simultaneously captures long-range temporal dependencies, sequential traffic behaviour, and feature importance, thereby improving intrusion detection capability in complex IoT environments. Another significant contribution of this work is the integration of SMOTE-based class balancing to address the severe imbalance present in the Bot-IoT dataset, which enhances the detection of minority traffic classes and reduces classification bias. The proposed model maintains a compact architecture of approximately 15K trainable parameters, making it computationally efficient and suitable for real-time IoT and edge-assisted deployments.

Extensive experimental evaluation, comparative analysis, confidence interval assessment, and confusion matrix analysis further confirm the robustness, reliability, and practical applicability of the framework for modern IoT intrusion detection systems. This section highlights the major findings of the study while also discussing the strengths, limitations, and practical implications of the proposed approach in real-world cybersecurity scenarios.

© 2026 Conscientia Beam. All Rights Reserved.

5.1. Limitations of the Proposed Work

Although the proposed framework achieved promising results, certain limitations remain within the current investigation.

1. The experimental evaluation was conducted mainly using the Bot-IoT Dataset, which may not completely represent the diversity of real-world IoT traffic conditions.
2. Dataset imbalance was addressed using SMOTE-based synthetic oversampling, which may generate artificial traffic samples that differ from naturally occurring network behaviour.
3. The present work mainly focused on binary intrusion detection rather than detailed classification of multiple attack categories.
4. The framework has not yet been extensively tested within large-scale, real-time IoT infrastructures or highly dynamic industrial environments.
5. Similar to many deep learning models, the interpretability of the hybrid architecture remains limited because its decision-making process operates largely as a black-box system.

6. CONCLUSION

The present study developed a lightweight hybrid intrusion detection framework by combining Temporal Convolutional Networks (TCN), Gated Recurrent Units (GRU), and an attention mechanism to strengthen cybersecurity in Internet of Things (IoT) environments. The proposed framework was designed to analyse sequential network traffic behaviour, identify temporal attack patterns, and determine the importance of relevant traffic features from highly imbalanced IoT datasets. Experimental analysis performed on the Bot-IoT Dataset showed that the model achieved highly satisfactory detection performance, with 99.92% accuracy, 100% precision, 99.92% recall, and a 99.96% F1-score. The framework also produced zero false-positive predictions and very few false negatives, indicating its capability to detect malicious activities with high reliability and stability in real-time IoT environments. The application of SMOTE-based balancing contributed significantly to improving minority attack detection and reducing classification bias caused by dataset imbalance. In addition, the proposed architecture maintained relatively low computational complexity, which makes it suitable for implementation in resource-constrained IoT devices and edge computing systems. The integration of TCN, GRU, and attention layers enabled efficient extraction of temporal and contextual traffic information, thereby improving the overall intrusion detection capability against modern cyber threats.

6.1. Policy Implications

The outcomes of this research may provide useful guidance for cybersecurity planning, IoT governance, and intelligent threat management systems.

1. The proposed framework can assist organizations in establishing automated intrusion detection systems for smart healthcare, industrial IoT, transportation systems, and critical infrastructure applications.
2. Regulatory authorities and cybersecurity policymakers may utilize AI-driven intrusion detection approaches while formulating IoT security guidelines and cybersecurity compliance standards.
3. Since the framework requires comparatively fewer computational resources, it may support secure deployment in edge computing and distributed IoT environments where lightweight security mechanisms are essential.
4. The reduction of false alarms and improvement in attack detection accuracy can help organizations minimize operational risks and improve incident response efficiency.
5. The framework may also contribute to strengthening national cyber defense strategies by enabling faster identification of botnet attacks, DDoS attacks, and other emerging IoT-based cyber threats.

© 2026 Conscientia Beam. All Rights Reserved.

6.2. Future Research Directions

Several improvements can be explored in future studies to enhance the robustness and practical applicability of the proposed framework.

1. Future experiments may validate the model using additional benchmark and real-world IoT datasets, such as the CICIDS2017 Dataset, UNSW-NB15 Dataset, and TON-IoT Dataset.
2. The framework may be extended toward multiclass intrusion detection to classify different categories of cyberattacks more effectively.
3. Federated learning and distributed artificial intelligence approaches can be integrated to support privacy-preserving collaborative intrusion detection across decentralized IoT networks.
4. Explainable Artificial Intelligence (XAI) techniques may be incorporated to improve model transparency, interpretability, and trustworthiness for cybersecurity professionals.
5. Additional optimization techniques can be applied to reduce power consumption and improve deployment efficiency on lightweight edge and fog-computing devices.
6. Future studies may also investigate adaptive online learning approaches for handling continuously evolving attack patterns and unknown zero-day threats in dynamic IoT ecosystems.

Funding: This study received no specific financial support.
Institutional Review Board Statement: Not applicable.
Transparency: The authors state that the manuscript is honest, truthful, and transparent, that no key aspects of the investigation have been omitted, and that any differences from the study as planned have been clarified. This study followed all writing ethics.
Competing Interests: The authors declare that they have no competing interests.
Authors' Contributions: All authors contributed equally to the conception and design of the study. All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Y. Zhang, R. C. Muniyandi, and F. Qamar, "A review of deep learning applications in intrusion detection systems: Overcoming challenges in spatiotemporal feature extraction and data imbalance," *Applied Sciences*, vol. 15, no. 3, p. 1552, 2025. https://doi.org/10.3390/app15031552

[2] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (IoT) security," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1646-1685, 2020. https://doi.org/10.1109/COMST.2020.2988293

[3] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT security techniques based on machine learning: How do IoT devices use AI to enhance security?" *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 41-49, 2018. https://doi.org/10.1109/MSP.2018.2825478

[4] S. Yaras and M. Dener, "IoT-based intrusion detection system using new hybrid deep learning algorithm," *Electronics*, vol. 13, no. 6, p. 1053, 2024. https://doi.org/10.3390/electronics13061053

[5] B. Cui, Y. Chai, Z. Yang, and K. Li, "Intrusion detection in IoT using deep residual networks with attention mechanisms," *Future Internet*, vol. 16, no. 7, p. 255, 2024. https://doi.org/10.3390/fi16070255

[6] H. Hindy *et al.*, "A taxonomy of network threats and the effect of current datasets on intrusion detection systems," *IEEE Access*, vol. 8, pp. 104650–104675, 2020. https://doi.org/10.1109/ACCESS.2020.3000179

[7] M. L. Mutleg, A. M. Mahmood, and M. M. J. Al-Nayar, "Deep learning based intrusion detection system of IoT technology: Accuracy versus computational complexity," *International Journal of Safety and Security Engineering*, vol. 14, no. 5, pp. 1547–1558, 2024. https://doi.org/10.18280/ijsse.140522

[8] M. Conti, A. Dehghantanha, K. Franke, and S. Watson, "Internet of Things security and forensics: Challenges and opportunities," *Future Generation Computer Systems*, vol. 78, pp. 544-546, 2018. https://doi.org/10.1016/j.future.2017.07.060

© 2026 Conscientia Beam. All Rights Reserved.

[9] I. Ullah and Q. H. Mahmoud, "Design and development of a deep learning-based model for anomaly detection in IoT networks," *IEEE Access*, vol. 9, pp. 103906–103926, 2021. https://doi.org/10.1109/ACCESS.2021.3094024

[10] H. Gupta, A. Jadhav, and A. S. Bisht, "Comparative analysis of machine and deep learning models for intrusion detection in fog-enabled IoT networks," *International Journal of Networked and Distributed Computing*, vol. 14, no. 1, p. 1, 2026. https://doi.org/10.1007/s44227-025-00079-8

[11] S. Thaker, H. Sarhan, and N. Portmann, "Feature extraction for machine learning-based intrusion detection in IoT networks," in *Proceedings of the 2022 IEEE/IFIP Network Operations and Management Symposium (NOMS 2022)*, 2022, pp. 1–9.

[12] H. Nandanwar and R. Katarya, "Deep learning enabled intrusion detection system for Industrial IOT environment," *Expert Systems with Applications*, vol. 249, p. 123808, 2024. https://doi.org/10.1016/j.eswa.2024.123808

[13] S. M. Kasongo and Y. Sun, "A deep learning method with filter based feature engineering for wireless intrusion detection system," *IEEE Access*, vol. 7, pp. 38597–38607, 2019. https://doi.org/10.1109/ACCESS.2019.2905633

[14] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, vol. 1, Funchal, Madeira, Portugal, 2018, pp. 108-116. https://doi.org/10.5220/0006639801080116

[15] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, ACT, Australia, 2015, pp. 1-6. https://doi.org/10.1109/MilCIS.2015.7348942

[16] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779-796, 2019. https://doi.org/10.1016/j.future.2019.05.041

[17] A. M. Banaamah and I. Ahmad, "Intrusion detection in IoT using deep learning," *Sensors*, vol. 22, no. 21, p. 8417, 2022. https://doi.org/10.3390/s22218417

[18] N. W. Khan *et al.*, "A hybrid deep learning-based intrusion detection system for IoT networks," *Mathematical Biosciences and Engineering*, vol. 20, no. 8, pp. 13491-13520, 2023. https://doi.org/10.3934/mbe.2023602

[19] M. Roopak, G. Y. Tian, and J. Chambers, "Deep learning models for cyber security in IoT networks," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, 2019, pp. 452-457. https://doi.org/10.1109/CCWC.2019.8666588

[20] D. M. A. A. Afraji, J. Lloret, and L. Peñalver, "An integrated hybrid deep learning framework for intrusion detection in IoT and IIoT networks using CNN-LSTM-GRU architecture," *Computation*, vol. 13, no. 9, p. 222, 2025. https://doi.org/10.3390/computation13090222

[21] A. Sagu, N. S. Gill, P. Gulia, N. Alduaiji, P. K. Shukla, and M. A. Shah, "Advances to IoT security using a GRU-CNN deep learning model trained on SUCMO algorithm," *Scientific Reports*, vol. 15, no. 1, p. 16485, 2025. https://doi.org/10.1038/s41598-025-99574-9

[22] P. R. Chithra Rani and K. Baalaji, "Deep learning-based ensemble stacking for enhanced intrusion detection in IoT-edge platforms," *Discover Applied Sciences*, vol. 7, no. 8, p. 928, 2025. https://doi.org/10.1007/s42452-025-06871-z

[23] O. S. Kareem, Y. A. Muhammed, Z. A. A. Shwany, T. S. Mustafa, M. U. Salur, and N. A. Othman, "Intrusion detection in SDN-IoT using robust GRU with attention mechanism and fuzzy weighting," *Intelligent Data Analysis: An International Journal*, vol. 30, no. 5, pp. 1452–1469, 2026. https://doi.org/10.1177/1088467X251340184

[24] H. Ma, W. Zhang, D. Zhang, and B. Chen, "An IoT intrusion detection framework based on feature selection and large language models fine-tuning," *Scientific Reports*, vol. 15, no. 1, p. 21158, 2025. https://doi.org/10.1038/s41598-025-08905-3

[25] M. A. Hossain, "Deep learning-based intrusion detection for IoT networks: A scalable and efficient approach," *EURASIP Journal on Information Security*, vol. 2025, no. 1, p. 28, 2025. https://doi.org/10.1186/s13635-025-00202-w

© 2026 Conscientia Beam. All Rights Reserved.

[26] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," *EAI Endorsed Transactions on Security and Safety*, vol. 3, no. 9, pp. 1-6, 2016.

[27] H. Zhang, "Development of an intelligent intrusion detection system for IoT networks using deep learning," *Discover Internet of Things*, vol. 5, no. 1, p. 74, 2025. https://doi.org/10.1007/s43926-025-00177-7

Views and opinions expressed in this article are the views and opinions of the author(s), Review of Computer Engineering Research shall not be responsible or answerable for any loss, damage or liability etc. caused in relation to/arising out of the use of the content.

© 2026 Conscientia Beam. All Rights Reserved.