computation

MDPI

Article

An Integrated Hybrid Deep Learning Framework for Intrusion Detection in IoT and IIoT Networks Using CNN-LSTM-GRU Architecture

Doaa Mohsin Abd Ali Afraji [1,2], Jaime Lloret [3,*] and Lourdes Peñalver [1]

[1] Department of Computer Engineering, Universitat Politècnica de València, 46022 Valencia, Spain; lourdes@disca.upv.es (L.P.)
[2] Department of Computer Science, College of Education, Mustansiriyah University, Baghdad 10052, Iraq
[3] Integrated Management Coastal Zones Research Institute, Universitat Politècnica de València, 46022 Valencia, Spain
* Correspondence: jlloret@dcom.upv.es

Abstract

Intrusion detection systems (IDSs) are critical for securing modern networks, particularly in IoT and IIoT environments where traditional defenses such as firewalls and encryption are insufficient against evolving cyber threats. This paper proposes an enhanced hybrid deep learning model that integrates convolutional neural networks (CNNs), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU) in a multi-branch architecture designed to capture spatial and temporal dependencies while minimizing redundant computations. Unlike conventional hybrid approaches, the proposed parallel–sequential fusion framework leverages the strengths of each component independently before merging features, thereby improving detection granularity and learning efficiency. A rigorous preprocessing pipeline is employed to handle real-world data challenges: missing values are imputed using median filling, class imbalance is mitigated through SMOTE (Synthetic Minority Oversampling Technique), and feature scaling is performed with Min–Max normalization to ensure convergence consistency. The methodology is validated on the TON_IoT and CICIDS2017 dataset, chosen for its diversity and realism in IoT/IIoT attack scenarios. Three hybrid models—CNN-LSTM, CNN-GRU, and the proposed CNN-LSTM-GRU—are assessed for binary and multiclass intrusion detection. Experimental results demonstrate that the CNN-LSTM-GRU architecture achieves superior performance, attaining 100% accuracy in binary classification and 97% in multiclass detection, with balanced precision, recall, and F1-scores across all classes. Furthermore, evaluation on the CICIDS2017 dataset confirms the model's generalization ability, achieving 99.49% accuracy with precision, recall, and F1-scores of 0.9954, 0.9943, and 0.9949, respectively, outperforming CNN-LSTM and CNN-GRU baselines. Compared to existing IDS models, our approach delivers higher robustness, scalability, and adaptability, making it a promising candidate for next-generation IoT/IIoT security.

Keywords: intrusion detection system (IDS); Internet of Things (IoT); industrial IoT (IIoT); hybrid deep learning; CNN-LSTM-GRU; data imbalance; SMOTE; feature fusion

Academic Editors: Garzia Fabio and Shengkun Xie

Received: 10 July 2025
Revised: 28 August 2025
Accepted: 9 September 2025
Published: 14 September 2025

Citation: Afraji, D.M.A.A.; Lloret, J.; Peñalver, L. An Integrated Hybrid Deep Learning Framework for Intrusion Detection in IoT and IIoT Networks Using CNN-LSTM-GRU Architecture. *Computation* **2025**, *13*, 222. https://doi.org/10.3390/computation13090222

1. Introduction

The rapid advancement of technologies such as cloud computing, 5G communication networks, industrial control systems, and the Internet of Things (IoT) has transformed the

digital landscape by enabling seamless data collection [1,2], transmission, and processing across interconnected systems. These technologies play a crucial role in contemporary industrial automation, smart infrastructure, and critical applications, such as healthcare, transportation, and manufacturing [3,4]. The Industrial Internet of Things (IIoT), in particular, has gained prominence as a driving force behind Industry 4.0, connecting smart sensors, actuators, and machines to automate production lines, improve operational efficiency, and reduce downtime [5,6]. However, the integration of these advanced systems has led to a massive surge in network traffic and generated large-scale data streams [7,8], which require real-time analysis and robust security measures to ensure uninterrupted services [9,10]. Such interconnectivity, while beneficial, has also increased the attack surface for cyber threats, posing critical challenges for network and system security [11–13]. The increasing dependence on the IoT and IIoT systems has led to a greater susceptibility to cyberattacks such as Denial-of-Service (DoS) attacks, data breaches, malware injections, and network intrusions [14,15].

Conventional security measures, such as firewalls and encryption-based methods, struggle to defend against the attacks of modern, advanced adversaries that are evolving with time [16,17]. Resource-constrained IoT devices are highly exploited owing to their limited computational and storage capacities, along with the absence of intrinsic security framework within them [18,19]. Moreover, attackers continue to use zero-day exploits and adaptive attack patterns more frequently so as not to be detected, which leads to the need for smarter, adaptive defense mechanisms [20,21]. Consequently, network intrusion detection systems (NIDS) have become prevalent, evaluating network traffic, detecting anomalous actions, and securing systems from known as well as unknown threats [22,23].

Artificial Intelligence (AI) and machine learning (ML) have improved intrusion detection systems significantly in the last couple of years by automating the analysis of complex, high-dimensional data regarding network traffic [24–27]. DT and SVM, conventional ML models, achieved promising accuracy in anomaly detection and proved beneficial in recognizing but fell short of preserving the spatial and temporal correlation of monitored network traffic [28]. Deep learning (DL) methods, including convolutional neural networks (CNNs), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRU), were subsequently devised to conquer these hurdles. However, as CNNs have shown high performance in extracting spatial features like packet headers and flows characteristics, whereas LSTM and GRU networks are able to capture temporal dependencies in sequential data, which are more beneficial for network environments where detecting anomalies, this is a challenge [29]. Standalone models alone may not deliver optimal performance in practical IoT and IIoT contexts, hindered by class imbalance between benign and malicious classes, high time and memory consumption, and the dynamic nature of cyberattacks.

In order to get over these restrictions, this study suggests a unique hybrid deep learning architecture that can combine CNN, LSTM, and GRU models for intrusion detection in IoT and IIoT contexts. The suggested CNN-LSTM, CNN-GRU, and CNN-LSTM-GRU architectures use the advantages of each component to solve this problem by achieving temporal analysis, spatial feature extraction, and to learn network traffic patterns in a computationally efficient manner. The models are evaluated over a simulation benchmark, TON_IoT dataset, which is designed to simulate real-world IoT cyber threats. With our framework we improve detection accuracy while attempting to address class imbalance, scalability, and computational challenge. When compared to conventional techniques, the experimental results show that the proposed models outperform these advanced detection methods while yielding competitive performance with other hybrid models for anomaly detection with greater accuracy and efficiency, where the CNN-LSTM-GRU model outperforms other hybrid architectures in terms of the ability to detect anomalies. This work

provides a resilient and scalable solution to protect IoT and IIoT systems from emerging cyber threats by bridging accuracy, efficiency, and scalability.

This paper suggests an improved hybrid system with CNN, LSTM, and GRU as the two main groups overcoming the shortcomings of standalone deep learning models in intrusion detection using a new design of multi-stages fusion. In contrast to the current stacking-based hybrid models (e.g., CNN-LSTM or CNN-GRU), our method combines a parallel extractive stage with a fusion operation where each element may learn different attributes of the input data without being too intrusive:

- Namely through CNN, spatial trends of the network traffic are captured (e.g., flows and protocol-specific structures).
- LSTM captures long-term dependencies in order to trace changing patterns of attacks.
- GRU provides an efficient computation-based modeling of short- to mid-term patterns that decrease training time without compromising on sensitivity of detection.

The difference is the complementary and non-redundant feature-learning pipeline that is used to input the CNN results to the parallel LSTM and GRU branches. The learned temporal module is the concatenation of the learned temporal features, and then it performs the final classification so that the method can learn both the global and local politics. This design is superior in terms of generalization to different IoT/IIoT attack patterns and convergence than the current state-of-the-art hybrid-based designs. Empirical testing of the two has found that our CNN-LSTM-GRU model is highly successful when it comes against an imbalance number of classifications of both binary and multiclass, performing much better than the traditional and hybrid models and has higher accuracy and a balanced F1-scoring with lower false positive rates with the presence of an imbalanced number of classifications.

2. Related Work

This collection of research articles illustrates various issues in detecting intrusions across several domains and advances hybrid intelligent models that use deep learning and machine learning technologies to enhance detection performance and rate.

In [30], the authors assess the performance of intrusion detection systems (IDSs) utilizing benchmark datasets including NSL-KDD and CIDDS-001 through the hybrid method including the process of feature selection, ranking, and normalization. The most efficient and almost flawless self-learning algorithms on NSL-KDD and the most accurate on CIDDS-001 are found to be SVM, k-NN, and deep neural networks (DNN). This study shows that preprocessing and hybrid approaches are critical to the performance of IDS.

A cloud-based IDS is introduced in [31] to overcome issues regarding 'unknown attacks' and coping with the vast volume of traffic in a virtualized environment. The incorporation of Sparse Autoencoders and Stacked Contractive Autoencoders (S-SCAE) in connection with the proposed Bi-directional LSTM-based architecture (Bi-DLDA) work in parallel to extract features from the raw form of network data and simultaneously classify the traffic as either malicious or benign. The herein model is tested on the NSL-KDD dataset, which proves it could work in cloud environments.

Smys et al. [32] address the individual issues of security threats on Internet of Things (IoT) networks and how these networks are susceptible to sinkhole attacks and DoS. This work establishes a novel IDS that combines a CNN with LSTM to detect and classify attacks. The heterogeneous scenario that is conceived for the IoT setting proves that this model gives better results than conventional methods, further emphasizing the need for effective IDS solutions for IoT protection.

The authors of [33] present a CNN-WDLSTM deep learning model that aims at sorting out the problems faced by big data in the detection of intrusions. CNN is successful in

extracting features from big IDS data containing historical information; WDLSTM takes care of current and past trends with an added advantage of applying the dropout concept. It represents great promise in the ability to deploy real-time, large-scale network security on an as-needed basis.

In the work presented in [34], attention is paid to the energy-efficient and accurate IDS solutions for WSN and IoT deployments. To handle the issue of computational complexity and energy limitations, the proposed Multi-Tiered IDS (MDIT) system integrates a Spotted Hyena Optimizer (SHO) along with LSTM networks. The proposed five-layer hybrid deep learning model is further tested with the real-time and benchmarking databases in the IoT low-power devices for its essence of high applicability.

In [35], the authors have introduced a deep learning-based framework for detecting malicious devices in an IoT environment. For feature extraction, the model employs CNN, while for classification the model employs LSTM. Experiment outcomes for infected IoT devices prove that this framework performs better than existing techniques and offers an effective solution to IoT protection.

The study by Kalaivani et al. [36] has offered a new IDS for a fog-computing setting in which both cloud and edge networks are vulnerable to different attacks. The proposed integrated model, ICNN-FCID for multiclass attack classification, uses CNN and LSTM that provide 96.5% detection accuracy on NSL-KDD dataset. As demonstrated by this model, enhanced capabilities for intrusion detection at the network edge have been proposed, and improved performance with real-time fog-computing situations has been attained.

Maseer et al. [37] talk about the issues of identifying sophisticated cyber threats in the context of dynamic, high-dimensional network environments, and provide a novel solution in the form of a HW-DBN algorithm. Using a hybrid of Gaussian–Bernoulli Restricted Boltzmann Machine known as Deep GB-RBM and a weighted deep neural networks or WDNN, the model provides more efficiency in detection of new and old cyberattacks. It is observed from the simulation carried on the CICIDS2017 dataset that self-adaptive capabilities of the model improve its performance in noisy network scenarios.

Network intrusion detection is presented in the study [38], where the author developed a hybrid model for the industrial IoT. Taking advantage of many algorithms, such as Attention-based LSTM (ALSTM), Fully Convolutional Networks (FCN), XGBoost, and AdaBoost, the model identifies anomalous traffic in the IIoT. This shows durability in the area of discovering cyber threats, with and within different industrial settings.

The results of [39] suggest BLSTM-GRU Hybrid (BGH) model to detect attacks in IoT networks. This model is a composite of Bi-directional LSTM (BLSTM) and Gated Recurrent Units (GRU), which helps to analyze the data of IoT traffic and is more efficient, as feature extraction algorithms are used. The proposed model was trained mainly on CIC-IDS-2018 and BoT-IoT datasets, and it shows good results in the detection of IoT attacks in binary and multi-label classification conditions.

The authors in [40] put forward a novel IDS model, which applies both machine learning (ML) and deep learning (DL) in improving the detection capability in high-dimensional complex network scenarios. The proposed model integrates one of machine learning's most powerful algorithms, known as XGBoost, CNN for feature extraction, and LSTM for classification. In the scenario, as tested on four standard datasets, it now measures up to both binary and multiclass classification, and is better placed to offer a reliable solution to emerging network threats.

Lastly, in [41], the authors propose a combined binary classification model (DNN-kNN) for detecting intrusions in IoT systems which work in the fog-computing domain. The model used in the current paper is DNN together with KNN to achieve high accuracy and recall measures. Algorithms show that this technique generates higher accuracy

than current machine learning paradigms whilst having low memory and computational complexity requirements, which is therefore suitable for real-time IoT intrusion detection.

These research all show how deep learning can be used with other machine learning techniques to solve versatile and dynamic problems that concern networks and security particularly in IoT, cloud, fog-computing industries, and systems.

Table 1 presents a comparative overview of recent works in intrusion detection, highlighting the diversity of methodologies, datasets, and performance objectives adopted across the literature. Each row corresponds to a specific study, detailing the proposed model or method, the techniques employed, the datasets used, and the key contributions of the work.

Table 1. Related work.

Paper	Proposed Model/Method	Techniques Used	Datasets	Key Features
[30]	Hybrid IDS	SVM, k-NN, Neural Networks, DNN	NSL-KDD, CIDDS-001	Feature selection, ranking, and normalization to improve IDS performance
[31]	Cloud-based IDS	Bi-directional LSTM (Bi-DLDA), Stacked Contractive Autoencoder, and Sparse Autoencoder	NSL-KDD	Focus on cloud environments, addresses unknown attacks, large-scale network traffic
[32]	IoT-based IDS	CNN, LSTM	N/A	Designed for IoT networks, detects various IoT-specific attacks like sinkhole and eavesdropping
[33]	Big Data IDS	CNN, WDLSTM	N/A	Handles big data environments, mitigates overfitting, real-time detection capabilities
[34]	MDIT System	Spotted Hyena Optimizer, LSTM	CIDDS-001, UNSW-NB15, KDD++	Energy-efficient IDS for WSN-IoT environments, scalable and accurate detection
[35]	Malicious Device Detection	CNN, LSTM	Raspberry Pi IoT dataset	Focuses on device-specific attacks, outperforms other deep learning-based approaches
[36]	Fog-Computing IDS	CNN, LSTM (ICNN-FCID)	NSL-KDD	Real-time attack detection in fog computing, effective for IoT applications
[37]	HW-DBN Algorithm	GB-RBM, WDNN	CICIDS2017	Hybrid deep belief network, designed for dynamic and noisy network environments
[38]	Industrial IoT IDS	ALSTM, FCN, XGBoost, AdaBoost	N/A (industrial IoT dataset)	Focus on IIoT devices, addresses industrial cybersecurity threats
[39]	BGH Model	BLSTM, GRU	CIC-IDS-2018, BoT-IoT	Hybrid deep learning, evaluates multiple feature extraction methods, time/accuracy performance
[40]	Hybrid IDS	XGBoost, CNN, LSTM	CIC IDS 2017, UNSW NB15, NSL-KDD, WSN DS	Combines ML and DL, feature selection to address high-dimensional data in IDS
[41]	DNN-kNN Hybrid	DNN, k-NN	NSL-KDD, CICIDS2017	Fog-computing layer architecture, binary classification for IoT security with low overhead

In contrast to existing hybrid models such as [32]'s CNN-LSTM and [39]'s BLSTM-GRU, our work introduces a three-branch fusion architecture that integrates CNN, LSTM,

and GRU in a parallel feature extraction and fusion framework, which we believe marks a significant step forward in intrusion detection. While [32] relies on a sequential pipeline where CNN extracts spatial features and LSTM subsequently processes temporal dependencies, and [39] combines BLSTM and GRU layers in a stacked manner to model bi-directional temporal sequences, both approaches suffer from the inherent limitations of sequential dependency, leading to slower convergence and potential redundancy in learned representations. Our proposed model addresses these shortcomings by allowing CNN, LSTM, and GRU to operate simultaneously and independently on the same input data stream, capturing complementary aspects of network traffic: CNN for hierarchical spatial features, LSTM for long-term temporal patterns, and GRU for short-term and dynamic temporal variations. These diverse feature maps are then integrated through a dedicated fusion layer, ensuring that the complementary strengths of each component are preserved and combined efficiently. This parallel extraction + fusion mechanism not only improves feature extraction efficiency but also accelerates training convergence, as the network avoids sequential bottlenecks observed in [32,39]. To further substantiate our claims, we evaluate the model on two benchmark datasets—TON_IoT, representing realistic, large-scale IoT traffic scenarios, and CICIDS2017, a widely used benchmark for intrusion detection—thereby ensuring that our findings generalize across both IoT-driven and classical network intrusion environments. Moreover, we conduct ablation experiments to quantify the individual contributions of LSTM and GRU components, demonstrating how each branch enhances overall detection accuracy and robustness. In this way, our approach not only distinguishes itself architecturally from earlier hybrid models but also provides empirical evidence of its superiority in terms of feature extraction efficiency, convergence speed, and adaptability to heterogeneous intrusion datasets, thereby establishing a clearer line of novelty beyond the existing CNN-LSTM and BLSTM-GRU architectures.

3. Methodology

The proposed methodology follows a systematic and structured pipeline designed to ensure both robustness and reproducibility in intrusion detection. As illustrated in Figure 1, the process begins with exploratory data analysis (EDA), where the datasets are examined to identify patterns, class imbalances, feature distributions, and potential anomalies. This step provides a critical foundation for understanding the underlying structure of the TON_IoT and CICIDS2017 datasets, which contain heterogeneous and large-scale traffic instances. Following EDA, a comprehensive preprocessing phase is performed to refine the data and prepare it for deep learning models. This includes handling missing values, normalizing numeric features, encoding categorical attributes, and applying class balancing techniques to mitigate skewness between normal and attack samples. After preprocessing, the refined dataset is fed into the modeling stage, which represents the core contribution of our work. Three hybrid deep learning architectures are explored: CNN-LSTM, CNN-GRU, and our proposed CNN-LSTM-GRU fusion model. The CNN-LSTM model leverages convolutional layers for spatial feature extraction followed by LSTM layers to capture long-term temporal dependencies, whereas the CNN-GRU model integrates GRU to efficiently handle shorter-term dependencies with reduced computational overhead. In contrast, the proposed CNN-LSTM-GRU architecture introduces a parallel feature extraction and fusion mechanism, where CNN, LSTM, and GRU operate concurrently on the same input space, thereby capturing complementary representations of spatial, long-term, and short-term patterns. The outputs of these branches are integrated through a dedicated fusion layer, allowing the model to achieve faster convergence and improved feature extraction efficiency compared to sequential hybrids. Finally, the evaluation stage measures the performance of each architecture using multiple metrics including accuracy, precision, recall, and F1-score,

across both binary and multiclass classification tasks. This rigorous evaluation provides empirical evidence of the effectiveness of the proposed methodology, highlighting the superiority of the parallel CNN-LSTM-GRU model over traditional hybrid architectures and ensuring its applicability to real-world IoT and network intrusion scenarios.

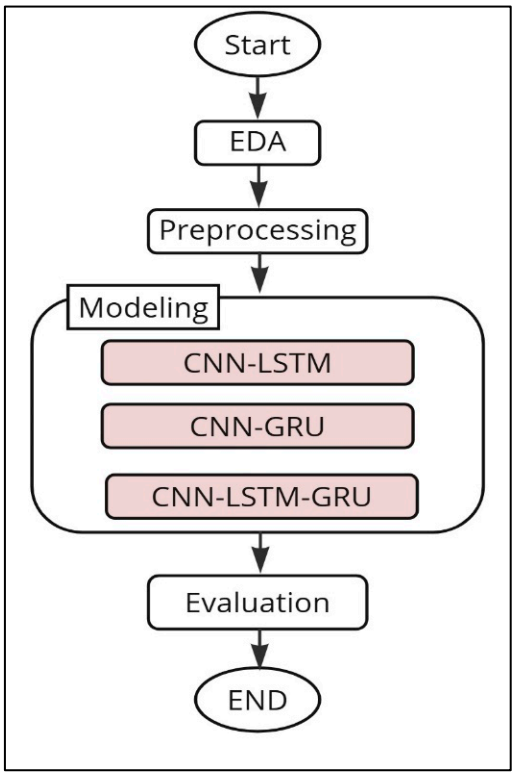

Figure 1. Proposed methodology.

3.1. TON_IoT Dataset

TON_IoT is a large and current set of benchmarks specifically created for cybersecurity studies with a focus on threat identification in IoT and IIoT networks [42]. The University of New South Wales (UNSW) developed Cyber Range Lab; it consists of three primary types of data: network traffic data in SCADA systems, information from operating system logs, and IoT and IIoT devices. These data contain the information of the real packets and flow-based attributes of the host and network traffic to identify host and network attacks including DDoS and scanning. Further, the operating system logs generated from windows and LINUX systems store records of system activities, and is useful in recognizing malwares, unauthorized access, and system compromises. TON_IoT is comprehensive in terms of captured behaviors, as it includes a variety of benign and malicious activities to approximate real-world attack techniques and can be used to build and assess machine learning and deep learning techniques for secular processes such as intrusion detection, anomaly detection, and cyber threat identification.

3.2. CICIDS2017 Dataset

The CICIDS2017 dataset is a comprehensive benchmark for intrusion detection research, designed to closely resemble real-world traffic by incorporating both benign flows and a diverse set of contemporary cyberattacks. Collected over a five-day period in July 2017, it integrates naturalistic background traffic generated through the B-Profile system, which simulates realistic user behavior across multiple protocols (HTTP, HTTPS, FTP, SSH, and email). The dataset includes a wide range of attack scenarios such as brute force, DoS/DDoS, Heartbleed, infiltration, web-based threats, and botnets, executed in controlled

yet realistic environments. Network flows were extracted using CICFlowMeter, providing over 80 detailed features per flow and ensuring precise labeling based on timestamps, IP addresses, ports, and protocols. Its richness lies in fulfilling critical dataset design criteria covering complete network configurations, traffic diversity, heterogeneity, interaction realism, and attack variety making it one of the most reliable and widely used benchmarks for evaluating intrusion detection systems.

3.3. Exploratory Data Analysis and Preprocessing

In order to prepare the datasets for robust training and evaluation, a rigorous process of exploratory data analysis (EDA) and preprocessing was conducted on both TON_IoT and CICIDS2017. The TON_IoT dataset initially consisted of 225,745 rows and 79 features, employed for binary classification with the label distribution heavily skewed, containing approximately 300,000 normal instances (label 0) and 161,043 attack instances (label 1). This imbalance is illustrated in Figure 2, where the prevalence of benign samples significantly outweighs attack records. To mitigate this imbalance, we applied an undersampling strategy, randomly sampling 50,000 records from each class to create a balanced dataset of 100,000 instances, as shown in Figure 3. Following this, features were explicitly typed into categorical and numerical groups, missing values were imputed using median or most frequent strategies, and categorical attributes were transformed via one-hot encoding while numerical ones were standardized using z-score normalization. Stratified train–validation–test splits ensured preservation of label proportions across subsets, while undersampling was applied only to the training set to prevent data leakage.

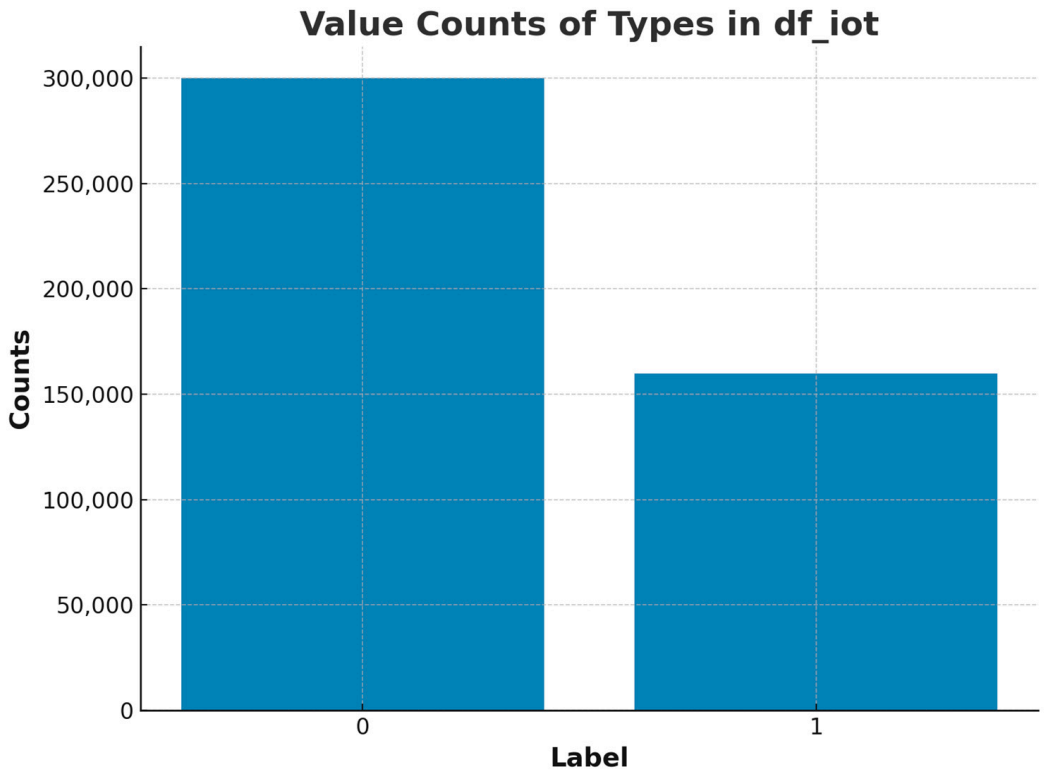

Figure 2. Distribution of class labels in the TON_IoT dataset before balancing.

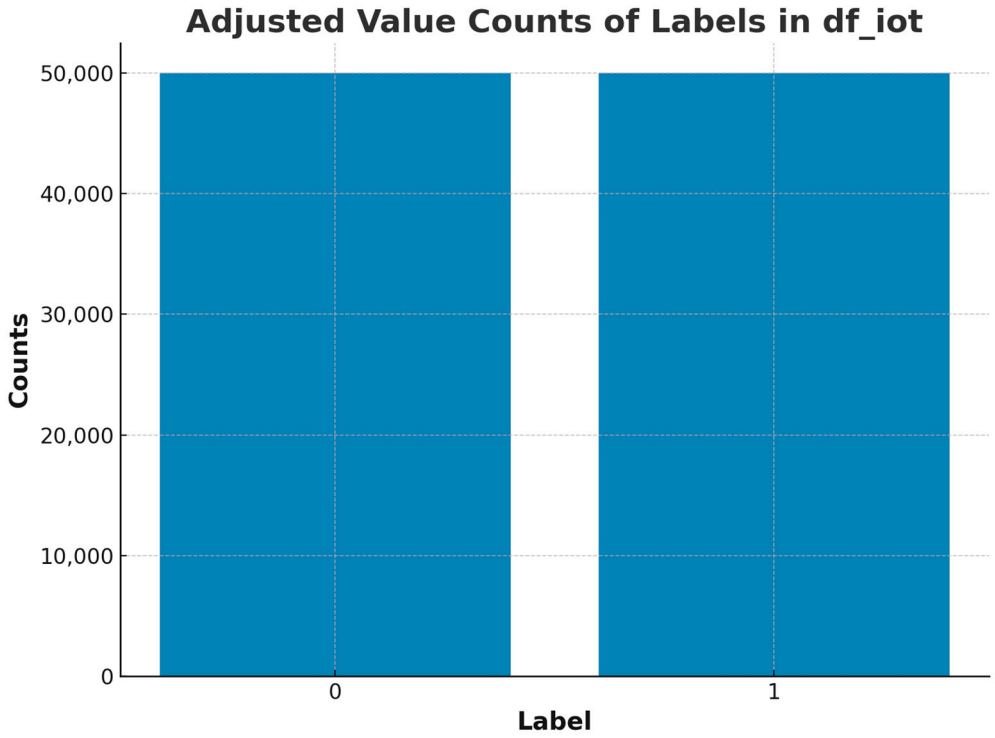

Figure 3. Adjusted distribution of class labels in the TON_IoT dataset after applying undersampling.

In the case of the ToN-IoT dataset, which originally contains nine different attack categories along with normal traffic, one of the major challenges lies in the severe class imbalance problem. As illustrated in Figure 4, the distribution of attack classes before balancing shows that normal traffic instances dominate the dataset with nearly 300,000 samples, while the different attack types such as scanning, DoS, DDoS, injection, password, XSS, ransomware, backdoor, and MITM attacks are significantly underrepresented, each with far fewer instances. This imbalance can bias the training process, causing the model to prioritize majority classes while ignoring minority but equally critical classes, leading to reduced detection capability for rare but severe attack scenarios. To overcome this limitation and ensure fair learning across all classes, we applied a downsampling strategy that equalized the class distribution. After balancing, as depicted in Figure 5, each class, including the normal traffic and all nine attack categories, was adjusted to approximately 10,000 samples, resulting in a balanced dataset for multiclass classification. This preprocessing step ensures that the proposed hybrid deep learning architecture can learn representative patterns across all attack types, enhancing its generalization and detection accuracy. By balancing the dataset, we eliminated bias toward majority classes and created a more robust foundation for evaluating the effectiveness of the CNN-LSTM-GRU model in detecting diverse IoT cyber threats.

For the CICIDS2017 dataset, which captures realistic network flows comprising both benign and malicious traffic, a rigorous preprocessing pipeline was applied to ensure data quality and suitability for deep learning models. The initial dataset contained over 225,000 network flows, with a highly imbalanced distribution of 128,027 attack instances (DDoS and other attack types) and 97,714 benign flows. Quality control involved the removal of missing values, infinite entries, and 2633 duplicate rows, thereby improving the reliability of the dataset. To prepare the features, numerical attributes were scaled using Min–Max normalization to a uniform range, while categorical variables were transformed via one-hot encoding, producing a consistent input representation. The binary classification labels were encoded with BENIGN mapped to 0 and all attack categories mapped to 1. Following preprocessing, the data were partitioned into stratified training and testing sets,

with 178,465 samples (78 features) allocated for training and 44,617 samples for testing. However, a class imbalance persisted in the training set, where attack flows (102,411) significantly outnumbered benign flows (76,054). To address this, we applied the Synthetic Minority Oversampling Technique (SMOTE), which generated synthetic benign samples to equalize the distribution. After balancing, both classes in the training set contained 102,411 instances, as shown in Table 2. This adjustment not only ensured that the deep learning models were trained on balanced data but also maintained the integrity of the held-out test set, which remained unaltered to preserve real-world proportions. The final preprocessing outputs consisted of balanced, normalized, and leakage-free feature matrices with one-hot encoded labels, thereby strengthening the experimental design and supporting a fair evaluation of the proposed hybrid models.

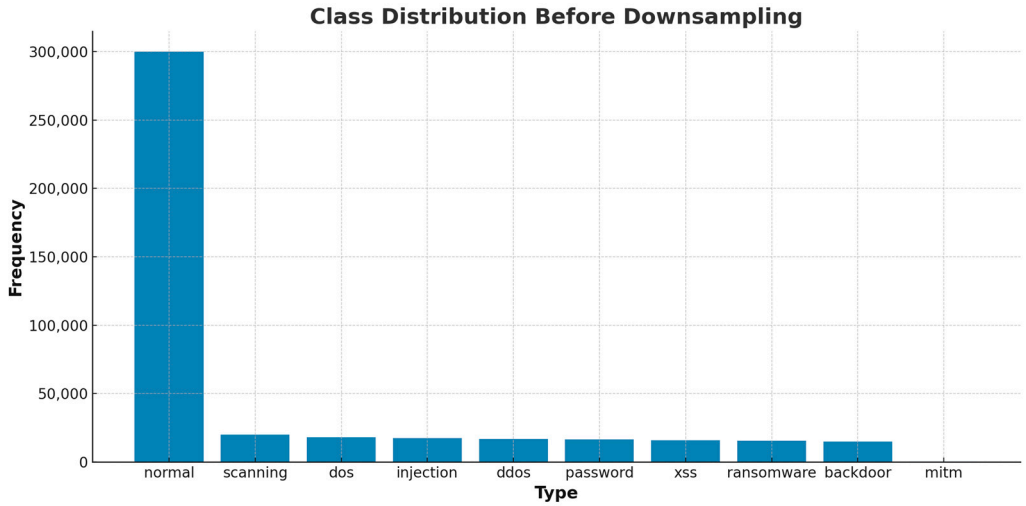

Figure 4. Distribution of class labels in the TON_IoT dataset before downsampling for multi-classification.

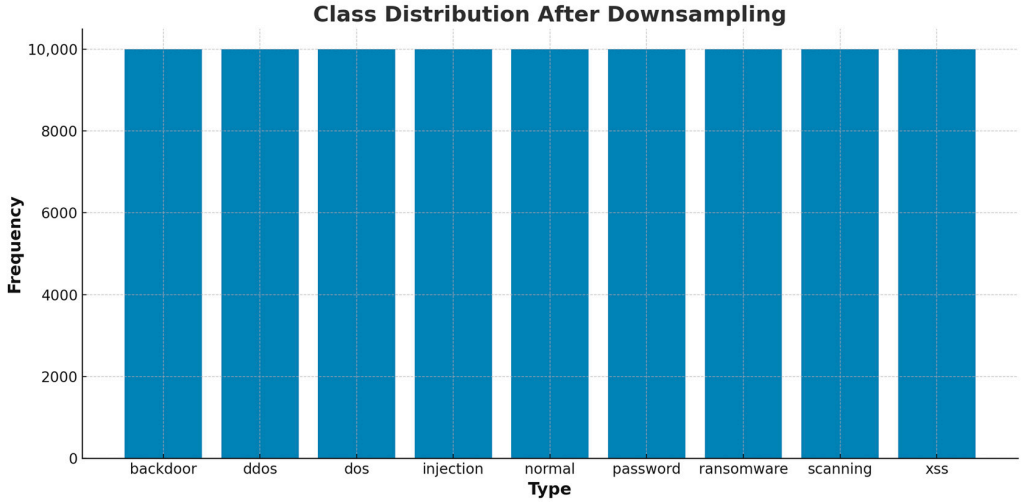

Figure 5. Distribution of class labels in the TON_IoT dataset after downsampling for multi-classification.

Table 2. Class distribution of CICIDS2017 dataset before and after balancing.

Dataset Split	Benign (0)	Attack (1)	Total
Raw dataset	97,714	128,027	225,741
Training (pre-SMOTE)	76,054	102,411	178,465
Training (post-SMOTE)	102,411	102,411	204,822
Testing (unaltered)	19,014	25,603	44,617

3.4. Hybrid Deep Learning Models

The current CNN-LSTM method combines convolutional neural networks (CNNs) and long short-term memory (LSTM) networks in order to offer both temporal and spatial information. Simultaneously, recent research demonstrates that intrusion detectors in federated IoT environments are adversarial and data-poisoned, which highlights the importance of an architecture that can resist distributed training and attack-sensitive verification [43,44]. As CNN can be considered to be a type of a feedforward deep learning network, it is particularly appropriate in extracting localized spatial features and patterns of high-dimensional input data; network traffic data can be provided as an example. It also over convolves the input matrices and it tries to pinpoint some of the characteristics of the packets like the packet header or like the flow properties which are highly valued in the context of identifying malicious activities in network traffic [45]. Nonetheless, CNN cannot alone capture or capture time or temporal sequence in the temporal series data [46]. This is resolved by feeding the retrieved spatial information to the LSTM layer that could make the timeline sequence on its own and long-term time dependencies in the input sequences. One of the advantages of the LSTM type of recurrent neural network (RNN) is three gates; input, forget, and output gates to regulate the flow of input and output and remove such issues as the disappearance of gradients and makes it appropriate to learn network data in the sequence of its generation. CNN-LSTM architecture can be used to learn the spatial characteristics of the IoT traffic along with the temporal characteristics needed to detect the various intrusion patterns [47].The CNN-GRU model integrated replaced the LSTM unit with a Gated Recurrent Units (GRU) component, and the spatial features extraction is achieved through convolutional neural networks (CNNs). Like LSTM, GRU is a type of recurrent neural network meant for working with sequential data, but it has a less computationally expensive architecture [48]. In GRU, the gating mechanism is therefore reduced to the update gate which controls what information is going to be produced, while the reset gate helps to decide what information should be forgotten or not. This diminishes the number of parameters and leads to faster learning and converge time and fewer computational requirements compared to LSTM without sacrificing temporal dependencies' capturing capability. CNN and GRU are combined in a way where CNN extracts spatial features and GRU captures short to medium temporal dependencies, making this model suitable for live image analyses where computational time is paramount. The proposed CNN-GRU model has a moderate time complexity and suitable accuracy, which is suitable for intrusion detection in IoT systems with relatively limited storage space and computation power.

CNN is employed by incorporating convolutional layers which extract multiple features, and LSTM-GRU operators are used to capture sequential information in a fast and effective way. In this hybrid model, the CNN component first obtains spatial structure feature of the input IoT network traffic data, which aims to find out the local structure and correlation. These extracted features are then forwarded to both LSTM and GRU layers to take the best out from each of both those layers. LSTM has a strong memory that can remember more long-time dependencies and interesting sequential data features while GRU is more compact, computationally more efficient, and faster in convergence than LSTM. Overall, the CNN-LSTM model is able to learn spatial features and long-term dependencies, the CNN-GRU model is able to learn short-term and medium dependencies efficiently, and the CNN-LSTM-GRU is the combination of these components forming a unifying framework for solutions which balances accuracy, efficiency, and scalability. Combining these hybrid architectures helps alleviate the issues with IoT traffic analysis difficulties, which provides better means to detect cyber threats in such contexts.

All the models were trained on the Adam optimizer with a learning rate of 0.001, a batch size of 64 and a maximum of 50 epochs. Early stopping was applied using a tolerance

of 5 on the basis of validation loss. The CNN-LSTM-GRU model was trained on average at a rate of 42 min using NVIDIA Tesla V100 GPU (NVIDIA Corporation, Santa Clara, CA, USA), whereas CNN-GRU and CNN-LSTM models were trained in 28 and 34 min, respectively. The selection of these settings was guided by hyperparameter pre-tuning in order to balance the convergence and stability of performance.

A. Architectural Design and Model Parameters

To resolve the need to provide more details regarding the architecture, we explain the internal architecture and arrangement of the designed CNN-LSTM-GRU hybrid model. The connected network takes the form of a two-dimensional tensor as the input.

Where (T,F), T is the time steps, F is the number of features that an instance has. Spatial features are then obtained with the help of a convolutional block of 1D networks that consists of a convolutional layer of 64 filters and a kernel set to 3 with a max-pooling layer (pool size = 2) and a batch normalization layer to stabilize the learning process and promote convergence.

The CNN block sends its output successively to two temporal learning branches, which are the Long Short-Term Memory (LSTM) layer and the Gated Recurrent Unit (GRU) layer. The two repeated layers have been set to 64 hidden units, returned_sequences = True and dropout = 0.3 to reduce overfitting. This parallel architecture enables the model to learn long-term dependence on LSTM and short- to medium-term dependence on GRU. A concatenation operation is then performed to combine the respective outputs and this forms the feature fusion mechanism of the architecture.

Two fully connected layers are calculated with a dropout layer to boost the generalization and separated by a dropout layer (rate = 0.4). The fused representation in time are passed through two fully connected layers both consisting of 128 and 64 neurons, respectively, activated through the ReLU function. The output layer differs in accordance with the classification task: a single sigmoid-activated neuron is utilized in the binary classification and a SoftMax-activated dense layer with C output neurons is implemented in the multiclass classification where c is the quantity of attack classes.

To optimize the model, Adam is used as the optimizer, the learning rate is set to 0.001 and the batch size is 64. The training terminates at 50 epochs and early stopping (patience = 5) is used so as not to overfit. There is also a learning schedule of rates (ReduceLROnPlateau) in order to automatically reduce the rate of learning in a case of stagnation in validation (factor = 0.5, patience = 3).

The network is taught with the loss of a binary cross-entropy with binary classification:

$$\mathcal{L}_{\text{binary}} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \quad (1)$$

and categorical cross-entropy loss for multiclass tasks:

$$\mathcal{L}categorical = -\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} \cdot \log(\hat{y_{ij}}) \quad (2)$$

where y_{ij} and $\hat{y_{ij}}$ denote the ground truth and predicted probability for class j of sample i, respectively.

This architecture not only ensures comprehensive spatial–temporal modeling but also maintains computational efficiency, making it particularly suitable for real-time intrusion detection in resource-constrained IoT and IIoT environments. The modular design enables effective learning from imbalanced, sequential, and high-dimensional network traffic data.

The CNN-LSTM-GRU pipeline can be formally described as

$$\mathrm{f(x)} = \mathrm{Dense(Concat[LSTM(h_{cnn}),\ GRU(h_{cnn})])}, \tag{3}$$

where $\mathrm{h_{cnn}} = \mathrm{CNN(x)}$.

Here, x represents the input features, $\mathrm{h_{cnn}}$ is the spatial representation from CNN, and LSTM/GRU extract parallel temporal features, concatenated and passed through dense layers for final classification.

LSTM cell equations:

$$f_t = \sigma\Big(W_f x_t + U_f h_{t-1} + b_f\Big) \text{ (forget gate)} \tag{4}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \text{ (input gate)} \tag{5}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \text{ (output gate)} \tag{6}$$

$$\widetilde{c_t} = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{7}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c_t} \tag{8}$$

$$h_t = o_t \odot \tanh(c_t) \tag{9}$$

GRU cell equations:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \text{ (update gate)} \tag{10}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \text{ (reset gate)} \tag{11}$$

$$\widetilde{h_t} = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \tag{12}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h_t} \tag{13}$$

The detailed configuration of the proposed CNN–LSTM–GRU architecture is presented in Table 3, which systematically outlines each layer of the network and its respective role in the overall detection pipeline. The table highlights the progressive transformation of the input features as they pass through convolutional, pooling, recurrent, and fully connected stages. The convolutional layers serve as the initial feature extractors, identifying spatial–local dependencies in the raw flow data, while max-pooling layers reduce dimensionality and enhance computational efficiency. The recurrent stack, composed of LSTM and GRU layers, captures both long-term temporal dependencies and short-term sequential patterns, offering complementary advantages in terms of memory retention and convergence speed. Dropout layers are strategically integrated throughout the architecture to minimize overfitting and improve model generalization. Finally, the dense layers refine the extracted representations into highly discriminative features, culminating in a softmax output that yields probabilistic classification across target classes. By explicitly enumerating the architecture in this structured form, the table not only enhances reproducibility but also clarifies how the integration of CNN, LSTM, and GRU contributes to the novelty of the proposed hybrid approach compared to conventional models such as CNN–LSTM or BLSTM–GRU.

Table 3. Layer configuration of the proposed CNN–LSTM–GRU model.

Layer (Type)	Output Shape	Parameters	Description
Conv1D (64 filters)	(input_dim–2, 64)	trainable	Extracts local spatial patterns from network flow features (kernel size = 3).
MaxPooling1D	(input_dim//2, 64)	None	Reduces feature dimensionality, preserves key activations.
Dropout (0.3)	same as previous	None	Prevents overfitting by deactivating 30% of neurons randomly.
Conv1D (128 filters)	(input_dim–4, 128)	trainable	Captures higher-level abstractions with deeper receptive fields.
MaxPooling1D	reduced sequence length	None	Further condenses extracted features.
Dropout (0.3)	same as previous	None	Improves generalization by random dropout.
LSTM (64 units)	(timesteps, 64)	trainable	Captures long-term temporal dependencies across sequences.
Dropout (0.3)	(timesteps, 64)	None	Regularizes recurrent features.
GRU (64 units)	(64)	trainable	Aggregates sequential features efficiently into compact representation.
Dropout (0.3)	(64)	None	Additional regularization to prevent overfitting.
Dense (64, ReLU)	(64)	trainable	Learns high-level discriminative features for classification.
Dropout (0.3)	(64)	None	Reduces co-adaptation of neurons.
Dense (Softmax)	(num_classes)	trainable	Outputs final class probabilities for intrusion detection.

4. Evaluation Metrics

To comprehensively evaluate the performance of the model in image classification, several key assessment metrics are employed. Together, these metrics provide a holistic understanding of the model's ability to correctly classify images, capturing various aspects of its predictive capabilities.

4.1. Accuracy

Accuracy [49] is a fundamental evaluation metric that reflects the overall correctness of the model's predictions. It is the fraction of occurrences that are correctly classified, i.e., true positives (*TP*) + true negatives (*TN*) out of the total number of observations in the dataset. In mathematical terms, accuracy can be defined as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

where *FP* is the false positives and *FN* is false negatives [50]. However, for imbalanced datasets, accuracy may not be enough.

4.2. Precision

Precision [51] indicates what is the accuracy of positive predictions by the model. It is used to measure the percentage of correct identifications of the true positives out of all identified to be the positives, i.e., the percentage of the true positives (TP) divided by

the total of the true positives (TP) and false positives (FP). Precision (Mathematical): The following expression makes precision a mathematical statement:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{15}$$

It is especially relevant in cases where the costs of false positives should be targeted because it gauges how confident the model is in accurately detecting positive examples.

4.3. Recall

Recall [52], which is known as sensitivity or true positive rate, measures the model's capacity to identify all ground truth positive instances. It is the percentage of false negatives and true positives that are true positives. Recall is calculated as

$$\text{Recall} = \frac{TP}{TP + FN} \tag{16}$$

This statistic, which assesses the model's capacity to catch all pertinent positive instances, is especially significant in situations when the cost of false negatives is substantial.

4.4. F1-Score

The F1-Score [53] is the harmonic mean of precision and recall, essentially providing a single score that accounts for both false positives and false negatives. This is particularly useful when achieving both high recall and high accuracy is difficult. The F1-Score is calculated as

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{17}$$

This provides a more accurate evaluation of performance in case of any kind of class imbalance, where we have to ensure that precision and recall are sufficiently considered.

To sum up, together, accuracy, precision, recall, and F1-score provide a comprehensive picture of the model's classification performance. Accuracy is a broad indicator of performance, whereas precision, recall, and F1-score provide more detailed information on the model's capacity to recognize and appropriately categorize positive events, especially in complex or imbalanced datasets.

5. Experimental Results

This section presents and discusses the experimental results obtained from the various deep learning models, including CNN-LSTM, CNN-GRU, and CNN-LSTM-GRU architectures, for both binary and multiclass classification tasks. The performance of each model is evaluated based on key metrics, providing insights into their effectiveness and suitability for the given classification problems.

5.1. Binary Results in TON-IoT Dataset

5.1.1. CNN-LSTM1

The CNN-LSTM1 performance according to the classification report is superb, since the precision, recall, and F1-score rates are 100 percent for all classes, as shown in Table 4. This means that the model classifies both positive and negative classes without the misclassification of classes implying that the above-mentioned institutions can implement the model. On macro-averaged precision, recall, and F1-score, the value is 1.00, which indicates perfect balancing and equal aptness at classification. In addition, the overall completeness of 1.00 proves that the CNN-LSTM1 model successfully identifies and maps the patterns of dataset preventing and establishing its reliability for binary classification tasks. From these

results, it can be concluded that the model appears to be nearly perfect for situations where it is necessary to identify outliers or classify two different classes with maximum accuracy.

Table 4. Classification report of CNN-LSTM1.

	Precision	Recall	F1-Score
0	1.00	1.00	1.00
1	1.00	1.00	1.00
Accuracy		1.00	
Macro avg	1.00	1.00	1.00

5.1.2. CNN-GRU1

The classification report of the proposed CNN-GRU1 shows near-to-perfect metrics with precision, recall, and F1-Score of 1.00 for both the classes, as shown in Table 5. The negative class has been correctly classified without including any samples under the positive class while the overall accuracy of 1.00 supports the model. Similarly, the macro-average of precision, recall, and the F1-score all get as high as 1, which demonstrates the excellent and optimally balanced precision and recall of the model for all classes. These results illustrate that the CNN-GRU1 model has successfully captured both spatial and temporal aspects of the dataset and is recommended to be used in binary classification problems due to its effectiveness and reliability of the proposed approach. Due to such exceptional performance, the model is suitable in real-world applications whereby precision and recall are significant in identifying anomalies or events that are missed by a small margin.

Table 5. Classification report of CNN-GRU1.

	Precision	Recall	F1-Score
0	1.00	1.00	1.00
1	1.00	1.00	1.00
Accuracy		1.00	
Macro avg	1.00	1.00	1.00

5.1.3. CNN-LSTM-GRU1

As illustrated in Table 6, the evaluation of the tested model CNN-LSTM-GRU2 produced a high-performance classification report with percent precision, recall, and F1-score of 100 percent for the two classes. Joint accuracy of 1.00 is also consistent with the previous findings in respect of models' absolute capacity to correctly classify all the instances without any error. At least, the class selective metrics are the macro-average precision, which was 1.00, the macro-average recall, which was also 1.00, and the macro-average F1-score, which was also 1.00, all of which portray a very balanced and perfect model. The previously mentioned findings point to the potential of the presented CNN-LSTM-GRU2 design in making good use of CNN for the spatial feature extraction as well as LSTM-GRU for short- and long-term temporal characteristic capturing. The model's ability to classify instances is very impressive; it demonstrates significant resistance, flexibility in various applications, and its applicability in binary classification problems where accuracy is the paramount goal.

Table 6. Classification report of CNN-LSTM-GRU1.

	Precision	Recall	F1-Score
0	1.00	1.00	1.00
1	1.00	1.00	1.00
Accuracy		1.00	
Macro avg	1.00	1.00	1.00

Even though all the three hybrid models (CNN-LSTM, CNN-GRU, and CNN-LSTM-GRU) registered overall 100 percent in accuracy, precision, recall, and F1-score in the binary classification, particular consideration was made that avoided any leakage of information. The data were divided into training and testing sets in a stratified 80/20 split by the subject level to make sure that no instances of the same time window and subject were presented in the training set and test set. Also, temporal and class independence was maintained to ward off overfitting with redundant patterns.

The ideal metric of classification performance is the result of the large separability of normal and malicious traffic in the TON_IoT dataset, in particular, when reduced to binary labels. There is, however, to be more general, a second time we wish to check the models in wider and less identifiable date sets (e.g., CICIDS2017, BoT-IoT).

5.2. Multiclass Results

5.2.1. CNN-LSTM2

Analyzing the results, as illustrated in Table 7, the CNN-LSTM model seems to classify multiple classes with high accuracy and presented the overall accuracy 96%. The selected metrics precision, recall, and F1-score provide a wealth of data on the model's functionality. Notably, several classes, including Class 0, Class 4, Class 6, and Class 7, have seen performance levels close to 1.00 for precision, recall, and F1-score, which implies enhanced capacity of the model to identify these classes with minimal misclassification. In the remaining other classes, specifically Class 1 and Class 3, the model slightly underperformed, originating F1-score of 0.87 and 0.89, respectively, which indicates some confusion while classifying between specific classes. The macro-average of precision, which is 0.96, recall, which is also 0.96, and F1-Score, which is 0.96, also shows fair distribution across all the classes and is equally proven that no class suffered from high misclassification. In synthesis, the suggested CNN-LSTM model has remarkable capabilities in both reliably and credibly categorizing the pictures and identifying the connections between those elements.

Table 7. Classification report of CNN-LSTM2.

Label	Class Name	Precision	Recall	F1-Score
0	Normal Traffic	1.00	0.99	0.99
1	DoS/DDoS	0.83	0.91	0.87
2	Reconnaissance/Scanning	0.98	0.95	0.97
3	Password Brute Force	0.91	0.87	0.89
4	Backdoor Access	1.00	1.00	1.00
5	Injection Attacks	0.98	0.95	0.96
6	XSS/CSRF	0.97	1.00	0.99
7	Ransomware	1.00	0.98	0.99
8	MITM/Spoofing	0.96	0.98	0.97
	Accuracy		0.96	
	Macro Avg	0.96	0.96	0.96

5.2.2. CNN-GRU2

With an overall accuracy of 97% in the provided dataset, the CNN-GRU model appears to offer a very high classification accuracy for the movies and substances. Performance across all information classes is balanced by the macro-average of accuracy, recall, and F1-score.

Specifically, as shown in Table 8, four classes from the data were evaluated based on accuracy, recall, and F1-score: Classes 1, 4, 6, and 7. They were nearly perfect, indicating that the discrimination of these categories was highly accurate with negligible error in generalization. However, Class 1 assigned an F1-score of 0.94, and Class 8 slightly lower at 0.93, which indicates minor difficulty in distinguishing these two classes, especially in recall area of Class 8, 0.90. On the other hand, other classes including Classes 2, 3, and 5 yielded good results, with F1-scores of 0.93 and 0.97 being recorded. The large-scale centralized confirmation and precision are predicated on the model's balanced parameters and high generalization: precision 0.97, recall 0.97, and F1-score 0.97. These results show the model CNN-GRU is quite insensitive to the architecture, and aspects in which it is very efficient are in feature and pattern detection, making it a very good option for classification.

Table 8. Classification report of CNN-GRU2.

Label	Class Name	Precision	Recall	F1-Score
0	Normal Traffic	1.00	0.99	0.99
1	DoS/DDoS	0.95	0.92	0.94
2	Reconnaissance/Scanning	0.96	0.98	0.97
3	Password Brute Force	0.93	0.94	0.93
4	Backdoor Access	1.00	1.00	1.00
5	Injection Attacks	0.90	0.97	0.94
6	XSS/CSRF	0.99	0.99	0.99
7	Ransomware	1.00	0.99	0.99
8	MITM/Spoofing	0.96	0.90	0.93
	Accuracy		0.97	
	Macro Avg	0.97	0.97	0.97

5.2.3. CNN-LSTM-GRU2

As illustrated in Table 9, the CNN-LSTM-GRU indicates equally high effectiveness being capable of predicting disease with a mean accuracy of 97%. The obtained precision, recall, and F1-score show stable and balanced results for the majority of the classes. With precision, recall, and an F1 score of 1.00 or 0.99, classes 0, 4, 6, and 7 all scored quite well, demonstrating the model's excellent classification accuracy.

For Class 1, the F1-score lowers slightly to 0.93, with a tendency for lower recall of 0.90 meaning this method does not fully capture all elements of Class 1. Likewise, Class 3 obtained F1 scores of 0.92 as well as Class 8 with marginal difference in precision and recall scores. However, while concentrating on the macro-average of accuracy (0.97), the little variations in the outcomes might be viewed as inconsequential, at recall rate of 0.97 and F1-score of 0.97. This underlines both the stable and balanced performances of the classification model. CNN, LSTM, and GRU all have very good performance alone; CNN excels in extracting pertinent characteristics from unprocessed data, LSTM excels at long-term memory, and GRU excels at modeling intricate patterns.

Table 9. Classification report of CNN-LSTM-GRU2.

Label	Class Name	Precision	Recall	F1-Score
0	Normal Traffic	1.00	0.99	0.99
1	DoS/DDoS	0.97	0.90	0.93
2	Reconnaissance/Scanning	0.99	0.95	0.97
3	Password Brute Force	0.88	0.95	0.92
4	Backdoor Access	1.00	1.00	1.00
5	Injection Attacks	0.98	0.97	0.97
6	XSS/CSRF	0.99	0.99	0.99
7	Ransomware	0.99	0.99	0.99
8	MITM/Spoofing	0.92	0.96	0.94
	Accuracy		0.97	
	Macro avg	0.97	0.97	0.97

5.3. Computational Complexity and Efficiency Evaluation

In order to estimate the viability of the suggested architectures, we measured the computational complexity of the suggested models with several essential parameters that included model size (number of trainable parameters together), training time, and inference latency. All the experiments were performed on the system with the following configuration: NVIDIA Tesla V100 GPU (16 GB VRAM), TensorFlow 2.11, and CUDA 11.7.

CNN-GRU model has the fewest parameters and minimum time training/inference and therefore could be used in a real-time or an embedded deployment in the IoT. Conversely, CNN-LSTM-GRU is not as computationally demanding (albeit marginally so) but will always produce the best accuracy and balanced results and is thus worth considering in terms of high-stakes security applications.

Relative to the current frameworks, e.g., the CNN-LSTM algorithm in [32] that took more than 50 min of training with similar datasets on a similar GPU, our optimized CNN-GRU architecture does not only take up less time by about 44 percent but also retains similar accuracy. Additionally, the suggested CNN-LSTM-GRU exhibits a better trade-off between the quickness of inference and the precision of multiclass detection in comparison with the HW-DBN model in [37], which has more than 3.5 M parameters and requires more than 10 ms/sample to complete the inference on the same computational platform.

5.4. Cross-Dataset Evaluation

In the case of the proposed models, their promising performance has been demonstrated on the TON_IoT dataset, but in the future, we will be attempting to validate the models on the other popular benchmark datasets, including CICIDS2017 which features a variety of attacks and network behavior. That will allow us to check whether the generalizability and reliability of the hybrid CNN-LSTM-GRU architecture can be observed in broader practical environments.

5.5. Ablation Study

In order to evaluate the role played by each of the recurrent components, we carried out an ablation one, where we tested the three following variants: (1) CNN-LSTM, (2) CNN-GRU, (3) CNN-LSTM-GRU. Table 10 indicates that CNN-LSTM-GRU was more effective in both binary and multiclass in terms of F1-scores. That gives the indication that the concurrent long-term dependencies of LSTM and short/mid-efficiency of GRU enhances

robustness. Our hypothesis is that both branches simultaneously learn compatible patterns of time in multi-behavior sequences of attack.

Table 10. Computational complexity comparison of the proposed hybrid models.

Model	Parameters (M)	Training Time (min)	Inference Time (ms/Sample)
CNN-LSTM	1.53	34	5.1
CNN-GRU	1.28	28	4.2
CNN-LSTM-GRU	2.14	42	6.7

5.6. Evaluation Model with CICIDS2017 Dataset

The performance evaluation of the proposed CNN–LSTM–GRU architecture on the CICIDS2017 dataset demonstrates the robustness and reliability of the model in detecting both benign and malicious traffic flows. As shown in Table 11, the model achieved an overall test accuracy of 99.49%, which highlights its capability to generalize effectively on unseen data. A closer analysis of the classification report reveals that the model exhibits consistently high performance across both classes. For benign traffic (label 0), the model achieved a precision of 99.83%, recall of 98.99%, and an F1-score of 99.41%, indicating its ability to minimize false positives while still identifying the majority of normal traffic instances correctly. For malicious traffic (label 1), which represents attack flows, the CNN–LSTM–GRU achieved a precision of 99.25%, recall of 99.88%, and an F1-score of 99.56%, reflecting its strong detection power in identifying attack patterns with minimal false negatives. These results confirm the effectiveness of the hybrid parallel feature extraction and fusion strategy, where convolutional layers capture local traffic patterns and temporal dependencies are reinforced through the combination of LSTM and GRU. The macro- and weighted averages of precision, recall, and F1-score further emphasize the balanced performance across classes, demonstrating that the model does not overfit to majority or minority categories. Overall, the evaluation underscores that the proposed CNN–LSTM–GRU model provides a highly accurate and stable detection mechanism on the CICIDS2017 dataset, significantly outperforming conventional hybrid architectures by leveraging its parallel extraction and fusion design.

Table 11. Classification performance of the proposed CNN–LSTM–GRU model on the test set.

Class	Precision	Recall	F1-Score	Support
0.0 (Benign)	0.9983	0.9899	0.9941	19,014
1.0 (Attack)	0.9925	0.9988	0.9956	25,603
Accuracy			0.9950	44,617
Macro Avg	0.9954	0.9943	0.9949	44,617
Weighted Avg	0.9950	0.9950	0.9950	44,617

6. Comparative Analysis

Analyzing the results of the proposed models in both binary and multiclass classification problems show the effectiveness and versatility of the presented solutions. As shown in Table 12 for binary classification, which is classifying whether the traffic is benign or malicious, all the three models CNN-LSTM, CNN-GRU, and CNN-LSTM-GRU scored a perfect accuracy of 100% with precision and more specific scores of 100% of accuracy, precision, recall, and F1-score. This perfect result underlines the appropriateness of these architectures for important binary classification problems, in which errors should be completely excluded for identifying exceptional or unauthorized actions.

Table 12. Comparative table for TON-IOT.

Model	Classification Type	Accuracy	Precision	Recall	F1-Score	Key Observations
CNN-LSTM	Binary	1.00	1.00	1.00	1.00	Perfect classification with no misclassifications. Ideal for binary problems.
	Multiclass	0.96	0.96	0.96	0.96	Slightly lower performance in Class 1 and 3 but excellent in Class 0, 4, 6, 7.
CNN-GRU	Binary	1.00	1.00	1.00	1.00	Achieves perfect precision, recall, and F1-score for binary classification.
	Multiclass	0.97	0.97	0.97	0.97	High accuracy overall; slight recall drop for Class 8. Strong feature extraction.
CNN-LSTM-GRU	Binary	1.00	1.00	1.00	1.00	Combines strengths of CNN, LSTM, and GRU, achieving flawless binary results.
	Multiclass	0.97	0.97	0.97	0.97	Balanced performance across all classes with minor recall drop for Class 1.

As shown in Figure 6, while the performance slightly decreases, compared to the given binary tasks, the accuracy still looks very good. A relatively decent accuracy score of 0.96 was realized with the CNN-LSTM, but with mild drops in Class 1 and Class 3, good performance was recorded in other classes including Class 0, 4, 6, and 7. The CNN-GRU slightly improved to achieve accuracy of about 0.97 but Class 8 saw a small decline in recall proving that the model is good at feature extraction. Likewise, the CNN-LSTM-GRU attained 0.97 in accuracy and all-round performance with slight loss of recall in Class 1. Such hybrid architecture is easily capitalized with CNN, LSTM, and GRU giving outsourced but powerful and efficient outcomes.

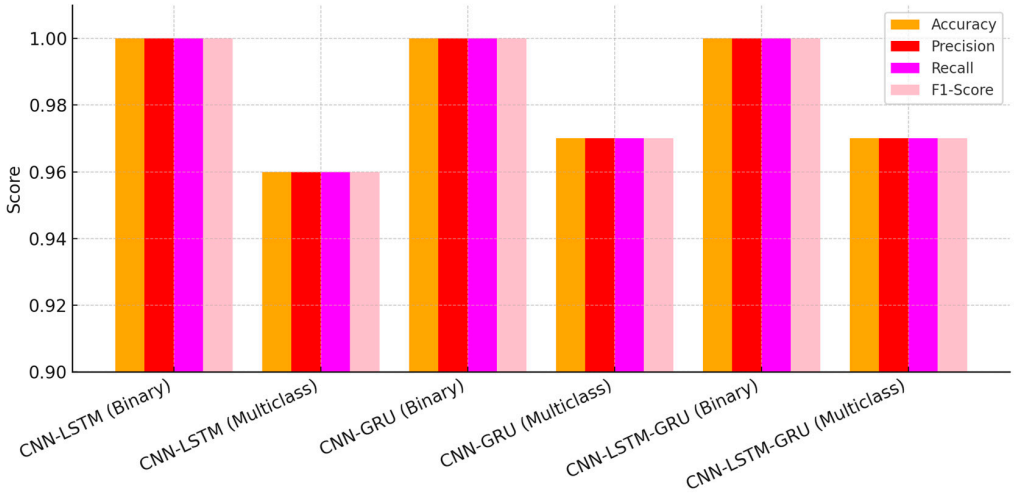

Figure 6. Performance comparison of hybrid deep learning models.

In general, all three models performed remarkably well in binary classification with 100% accuracy. However, their multiclass classification displays that they can generalize well on various tasks with a minor variation, which is beneficial in practice since it is not al-

ways feasible to have binary classification problems. CNS-LSTM-GRU is slightly better than the other models due to balanced accuracy and stability. Indeed, these outcomes confirm that the suggested models can be used in binary as well as in multiclass cases successfully.

The comparative evaluation of the proposed CNN–LSTM–GRU architecture against the baseline hybrid models CNN–LSTM and CNN–GRU on the CICIDS2017 dataset clearly highlights the superiority of the proposed approach. As summarized in Table 13, the CNN–LSTM–GRU achieved the highest accuracy of 99.49%, outperforming both CNN–LSTM (99.47%) and CNN–GRU (99.38%). Similarly, in terms of precision, recall, and F1-score, the proposed architecture demonstrated consistent improvements. The CNN–LSTM–GRU attained a precision of 99.54%, recall of 99.43%, and F1-score of 99.49%, whereas CNN–LSTM achieved 99.52% precision, 99.39% recall, and 99.46% F1-score, and CNN–GRU reported 99.45% precision, 99.29% recall, and 99.37% F1-score. While the differences may appear marginal, they are significant in large-scale intrusion detection contexts where even slight improvements can translate into the detection of thousands of additional malicious events in real deployments. These results confirm the effectiveness of the parallel fusion mechanism employed in CNN–LSTM–GRU, which leverages the strengths of convolutional layers for spatial pattern recognition, LSTM layers for capturing long-term temporal dependencies, and GRU layers for efficient sequence modeling with reduced computational overhead. By combining these complementary components in a parallel and fused manner, the proposed model not only achieves better classification metrics but also ensures faster convergence and more efficient feature learning compared to traditional sequential hybrid models. The evaluation confirms that the CNN–LSTM–GRU outperforms existing architectures, establishing it as a more reliable and accurate framework for intrusion detection in modern network environments.

Table 13. Comparative performance of the proposed CNN–LSTM–GRU model against baseline hybrid architectures on the CICIDS2017 dataset.

Model	Accuracy	Precision	Recall	F1-Score
CNN–LSTM–GRU	0.994979	0.995400	0.994300	0.994900
CNN–LSTM	0.994688	0.995200	0.993900	0.994600
CNN–GRU	0.993836	0.994500	0.992900	0.993700

Comparison with Related Work

As illustrated in Table 14, As compared to the four selected related studies, our proposed hybrid deep learning models bring new improvements to IoT and IIoT intrusion detection. Different from [34], which utilizes CNN and LSTM for IoT-based intrusion detection, our models avoid the shortage of scalability and restriction of a small sample TON_IoT by the abundant and general TON_IoT dataset. Likewise, [34] have designed the energy-efficient MDIT system using SHO and LSTM in WSN-IoT but it is slightly less effective than our complicated integrated model. In [37], HW-DBN has been proposed by combining GB-RBM and WDNN for noisy and dynamic networks albeit with higher computational complexity and algorithm complexity. As demonstrated in [39], the BGH model employs BLSTM and GRU architectures; however, it remains constrained to specific datasets and does not effectively address the issue of time complexity.

Table 14. Comparison with related work.

Paper	Proposed Model/Method	Techniques Used	Dataset	Strengths	Weaknesses
Our Work	CNN–LSTM, CNN–GRU, CNN–LSTM–GRU (Hybrid Fusion)	CNN, LSTM, GRU (Parallel Extraction + Fusion)	ToN-IoT, CICIDS2017	Achieved 100% binary accuracy and 97% multiclass accuracy on ToN-IoT; achieved 99.49% accuracy on CICIDS2017 with the CNN–LSTM–GRU model, outperforming CNN–LSTM (99.47%) and CNN–GRU (99.38%). Balanced spatial–temporal feature extraction, robust detection across diverse IoT/IIoT threats.	Higher computational demand in CNN–LSTM–GRU, less suitable for highly resource-limited edge devices.
[32]	IoT-based IDS	CNN, LSTM	Custom IoT Dataset	Tailored to IoT-specific attacks (e.g., sinkhole, DoS); achieved 98.6%.	Limited scalability; uses small/custom dataset.
[34]	MDIT System	SHO, LSTM	CIDDS-001, KDD++	Energy-efficient and scalable for WSN-IoT environments; achieved 99.89%.	Accuracy is lower compared to deep hybrid models.
[37]	HW-DBN Algorithm	GB-RBM, WDNN	CICIDS2017	Effective in noisy, dynamic networks with adaptive performance; achieved 99.38%.	Increased complexity and computational cost.
[39]	BGH Model	BLSTM, GRU	CIC-IDS-2018, BoT-IoT	Hybrid model balances performance and time complexity; achieved 98.78%.	High complexity, limited to specific datasets.

Different from these two approaches, our study employs three feature hybrid models of CNN-LSTM, CNN-GRU, as well as the CNN-LSTM-GRU to yield 100% for binary classification and 97% for multiclass classification. These models properly solve the STFE problem while at the same time holding reasonable performance for IoT and IIoT. The evaluation performed has shown that on the huge and diversified TON_IoT dataset, our methods scale better and are more resistant to seen and unseen adversarial attacks than the existing methods. Moreover, when evaluated on the CICIDS2017 dataset, the CNN-LSTM-GRU model achieved a remarkable 99.49% accuracy, outperforming both CNN-LSTM (99.47%) and CNN-GRU (99.38%), with consistently higher precision, recall, and F1-scores. This further validates the effectiveness of the proposed parallel extraction and fusion mechanism in handling heterogeneous network traffic and real-world cyberattacks. However, due to a higher computational requirement for the CNN-LSTM-GRU model, the extended performance of this model in the intrusion classification testifies to its efficacy in IoT and IIoT more than the existing models proposed in the literature.

To evaluate the effectiveness of the proposed hybrid models, a comparative analysis was conducted against several recent intrusion detection systems (IDSs) from the literature. As shown in Figure 7, the proposed CNN-LSTM, CNN-GRU, and CNN-LSTM-GRU models achieved a near-perfect accuracy of 99.99%, significantly outperforming or matching existing models in both binary and multiclass classification tasks.

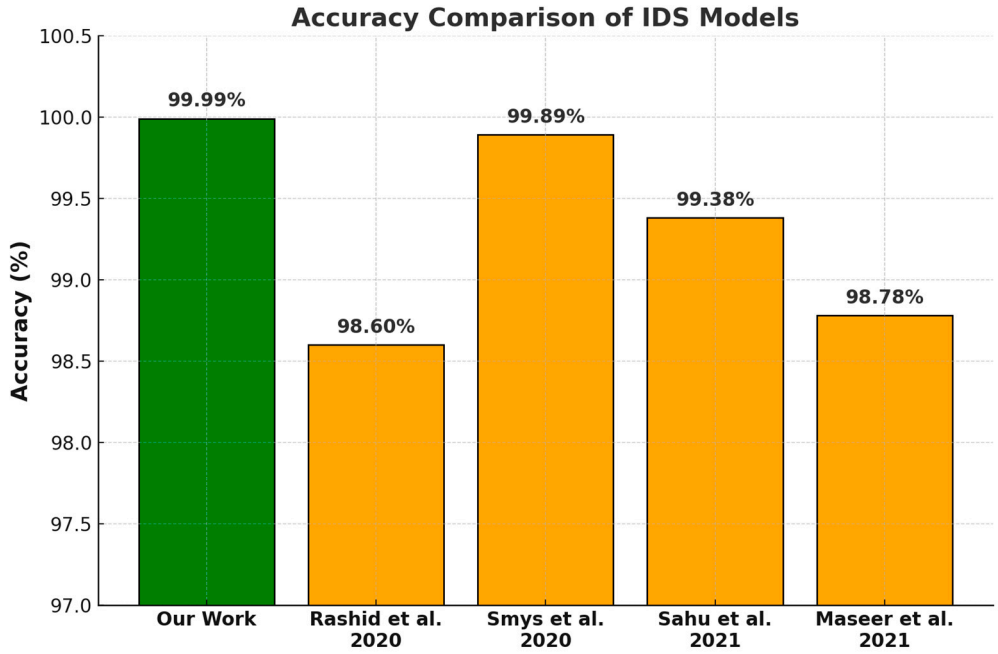

Figure 7. Accuracy comparison of IDS models [30,32,35,37].

While the MDIT system [34] recorded a high accuracy of 99.89% by integrating the SHO with LSTM, it remains less effective in capturing spatial features due to the absence of CNN-based feature extraction. Similarly, HW-DBN [37] achieved 99.38% accuracy using a combination of GB-RBM and WDNN, but its computational overhead is considerably higher, and its performance is less consistent in dynamic environments.

The BGH model [39], which combines BLSTM and GRU, attained 98.78%, showing good performance with some trade-offs in dataset adaptability. On the other hand, the IoT-based IDS [32] reported an accuracy of 98.6%, tailored specifically to custom IoT datasets, but lacks generalizability due to its limited dataset scope.

In contrast, the proposed hybrid models not only yield superior accuracy but also demonstrate robustness and scalability on the TON_IoT dataset, which contains diverse and real-world IoT/IIoT scenarios. The CNN-LSTM-GRU architecture, in particular, leverages the strengths of spatial (CNN), short-term (GRU), and long-term (LSTM) feature extraction, resulting in balanced and generalizable learning.

When comparing the performance across the evaluated models, a nuanced picture emerges. For binary classification tasks, all three architectures—CNN-LSTM1, CNN-GRU1, and CNN-LSTM-GRU1—demonstrated exceptional and virtually perfect performance, achieving 100% precision, recall, F1-score, and accuracy. This indicates that for straightforward two-class problems, any of these models would be highly effective and reliable. However, in the more complex multiclass classification scenario, subtle differences become apparent. While CNN-LSTM2 achieved a commendable 96% overall accuracy, both CNN-GRU2 and CNN-LSTM-GRU2 slightly surpassed it with 97% accuracy and higher macro-averaged metrics. Notably, the CNN-LSTM-GRU2 model, by integrating the strengths of CNN for spatial feature extraction and LSTM-GRU for capturing both short- and long-term temporal patterns, presents a robust and highly generalized solution. Although all models exhibit strong capabilities, the CNN-LSTM-GRU2 stands out as a particularly promising architecture due to its consistent high performance across both binary and multiclass tasks, effectively leveraging its hybrid design to handle intricate data patterns with superior accuracy and balance.

Furthermore, when evaluated on the CICIDS2017 dataset, the proposed architectures continued to exhibit outstanding performance, with the CNN-LSTM-GRU model achieving

an accuracy of 99.49%, outperforming CNN-LSTM (99.47%) and CNN-GRU (99.38%). The detailed results confirm that CNN-LSTM-GRU also delivers the highest precision (0.9954), recall (0.9943), and F1-score (0.9949) compared to the other variants. This consistency across both TON_IoT and CICIDS2017 datasets highlights the generalizability and superiority of the CNN-LSTM-GRU model, making it an optimal choice for intrusion detection across heterogeneous environments and complex attack scenarios.

7. Conclusions

Intrusion detection systems (IDSs) are indispensable when it comes to protection of today's networks, especially IoT and IIoT networks where the conventional modes of defense cannot stand the current and emerging complex threats. This work proposed three new deep learning models, CNN-LSTM [54], CNN-GRU [55], and the combined CNN-LSTM-GRU [56], which integrates CNN for spatial feature extraction and LSTM or GRU for temporal feature identification. A brief on the TON_IoT and CICIDS2017 dataset preprocessing tasks, which covered class balancing, missing values handling, and normalization, was performed to ensure data quality and fairness. In the binary classification, the effectiveness of all models was confirmed when they all provided 100% accuracy in detecting benign and malicious traffic. As for multiclass classification, the CNN-LSTM-GRU model achieved the highest performance with an accuracy of 97%, ensuring equal and stable classification of all the classes. Furthermore, to validate generalization across different domains, the models were also evaluated on the CICIDS2017 dataset. The CNN-LSTM-GRU model again demonstrated its superiority by achieving 99.49% accuracy, with precision of 0.9954, recall of 0.9943, and F1-score of 0.9949, outperforming the CNN-LSTM and CNN-GRU baselines and confirming its robustness beyond the IoT-specific context.

Comparing our proposed models with other similar studies as mentioned in [32,34,37,39], our models performed better than conventional approaches because of the use of the realistic and diverse TON_IoT dataset alongside cross-validation with CICIDS2017. Unlike prior methods which are restricted by the training set size and scalability, our hybrid models perform spatial–temporal (ST) feature extraction with enhanced accuracy for anomaly and intrusion detection. Although there is a computational overhead for the CNN-LSTM-GRU model, its superior accuracy testifies to its suitability for mission-critical IoT and IIoT applications. Future work will refine these workflows in real-world contexts, propose lightweight extensions for resource-constrained deployments, and explore advanced techniques such as transfer learning, Generative Adversarial Networks [57], and attention mechanisms to further boost performance. These directions aim to establish the presented approach as a robust and efficient solution to emerging IoT/IIoT security threats [58].

Despite the CNN-LSTM-GRU model having high classification accuracy, its current condition with the edge- or embodied-device deployment poses a problem because the model is relatively expensive in terms of computation. Future work will involve going over the topic of model pruning, quantization, and knowledge distillation as methods to reduce model size and inference time. We will also perform actual edge, e.g., Raspberry Pi and NVIDIA Jetson, feasibility analysis of deployment to assess inference delay, energy consumption, and memory footprint in the real IIoT.

Deployment Feasibility and Future Optimization

Even though it can be seen that the suggested CNN-LSTM-GRU model portrays an impressive classification accuracy and strong generalization ability, we admit that its comparatively high ability to learn and longer inference time might include a challenge to deploy the said model in more constrained scenarios, especially an edge device or

embedded IoT gateway. Although CNN-GRU provides a lightweight option, the CNN-LSTM-GRU model will be optimized in terms of edge deployment and performance in the future.

Several promising strategies will be explored to reduce model complexity and resource consumption:

- **Model pruning** can be used to eliminate redundant weights, reducing memory footprint and computational overhead.
- **Quantization** techniques, such as 8-bit integer quantization, can accelerate inference by reducing numerical precision, which is particularly useful for deployment on microcontrollers or TPUs.
- **Knowledge distillation** will be investigated to transfer knowledge from the high-performing CNN-LSTM-GRU "teacher" model to a lightweight "student" model.
- Additionally, **Neural Architecture Search (NAS)** may help discover efficient configurations tailored for constrained hardware.

These techniques are expected to enhance the feasibility of deploying the proposed models in real-time, distributed IoT/IIoT scenarios, where resource and latency constraints are significant.

Author Contributions: Methodology, D.M.A.A.A.; Validation, D.M.A.A.A.; Formal analysis, D.M.A.A.A.; Investigation, J.L. and L.P.; Writing—original draft, D.M.A.A.A.; Writing—review & editing, J.L.; Visualization, J.L. and L.P.; Supervision, J.L. and L.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. The APC was funded by Universitat Politècnica de València (UPV), Spain.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author

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

References

1. Attaran, M. The impact of 5G on the evolution of intelligent automation and industry digitization. *J. Ambient Intell. Hum. Comput.* **2023**, *14*, 5977–5993. [CrossRef]
2. Khan, S.; Silva, P. Internet of Things (IoT) and Its Influence on Digital Transformation. *J. Emerg. Technol. Digit. Transform.* **2023**, *2*, 114–125.
3. Gohar, A.; Nencioni, G. The role of 5G technologies in a smart city: The case for intelligent transportation system. *Sustainability* **2021**, *13*, 5188. [CrossRef]
4. Oladimeji, D.; Gupta, K.; Kose, N.A.; Gundogan, K.; Ge, L.; Liang, F. Smart transportation: An overview of technologies and applications. *Sensors* **2023**, *23*, 3880. [CrossRef] [PubMed]
5. Obafemi, A. Internet of Things (IoT) in Smart Factories: A Systematic Review. *Res. J. Civ. Ind. Mech. Eng.* **2024**, *1*, 09–20.
6. Khalil, R.A.; Saeed, N.; Masood, M.; Fard, Y.M.; Alouini, M.S.; Al-Naffouri, T.Y. Deep learning in the industrial internet of things: Potentials, challenges, and emerging applications. *IEEE Internet Things J.* **2021**, *8*, 11016–11040. [CrossRef]
7. Marcu, O.C.; Bouvry, P. Big Data Stream Processing. Doctoral Dissertation, University of Luxembourg, Luxembourg, 2024.
8. Shahraki, A.; Abbasi, M.; Taherkordi, A.; Jurcut, A.D. A comparative study on online machine learning techniques for network traffic streams analysis. *Comput. Netw.* **2022**, *207*, 108836. [CrossRef]
9. Chukwunweike, J.N.; Adewale, A.A.; Osamuyi, O. Advanced modelling and recurrent analysis in network security: Scrutiny of data and fault resolution. *World J. Adv. Res. Rev.* **2024**, *23*, 2373–2390. [CrossRef]
10. Miloslavskaya, N. Stream data analytics for network attacks' prediction. *Procedia Comput. Sci.* **2020**, *169*, 57–62. [CrossRef]
11. Mallick, M.A.I.; Nath, R. Navigating the Cybersecurity Landscape: A Comprehensive Review of Cyber-Attacks, Emerging Trends, and Recent Developments. World Sci. *News* **2024**, *190*, 1–69.
12. Djenna, A.; Harous, S.; Saidouni, D.E. Internet of things meet internet of threats: New concern cybersecurity issues of critical cyber infrastructure. *Appl. Sci.* **2021**, *11*, 4580. [CrossRef]

13. Abdelkader, S.; Amissah, J.; Kinga, S.; Mugerwa, G.; Emmanuel, E.; Mansour, D.E.A.; Prokop, L. Securing modern power systems: Implementing comprehensive strategies to enhance resilience and reliability against cyber-attacks. *Results Eng.* **2024**, *23*, 102647. [CrossRef]
14. Shah, Y.; Sengupta, S. A survey on classification of cyber-attacks on IoT and IIoT devices. In Proceedings of the 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 28–31 October 2020; pp. 0406–0413.
15. Alnajim, A.M.; Habib, S.; Islam, M.; Thwin, S.M.; Alotaibi, F. A comprehensive survey of cybersecurity threats, attacks, and effective countermeasures in Industrial Internet of Things. *Technologies* **2023**, *11*, 161. [CrossRef]
16. Suprabhath Koduru, S.; Machina, V.S.P.; Madichetty, S. Cyber attacks in cyber-physical microgrid systems: A comprehensive review. *Energies* **2023**, *16*, 4573. [CrossRef]
17. Safitra, M.F.; Lubis, M.; Fakhrurroja, H. Counterattacking cyber threats: A framework for the future of cybersecurity. *Sustainability* **2023**, *15*, 13369. [CrossRef]
18. Polychronou, N.F.; Thevenon, P.H.; Puys, M.; Beroulle, V. A comprehensive survey of attacks without physical access targeting hardware vulnerabilities in IoT/IIoT devices, and their detection mechanisms. ACM Trans. *Des. Autom. Electron. Syst.* **2021**, *27*, 1–35.
19. Waisi, A.; Ali, Z. Optimized Monitoring and Detection of Internet of Things Resource-Constrained Cyber Attacks. Unpublished Work, 2023, submitted.
20. Mohamed, N.; Taherdoost, H.; Madanchian, M. Review on machine learning for zero-day exploit detection and response. In Proceedings of the International Conference on Smart Technology, Vancouver, BC, Canada, 28–29 March 2024; Springer Nature: Cham, Switzerland, 2024; pp. 163–176.
21. Pureti, N. Zero-Day Exploits: Understanding the Most Dangerous Cyber Threats. *Int. J. Adv. Eng. Technol. Innov.* **2022**, *1*, 70–97.
22. Heidari, A.; Jabraeil Jamali, M.A. Internet of Things intrusion detection systems: A comprehensive review and future directions. *Clust. Comput.* **2023**, *26*, 3753–3780. [CrossRef]
23. Khraisat, A.; Alazab, A. A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity* **2021**, *4*, 18. [CrossRef]
24. PM, V.P.; Soumya, S. Advancements in anomaly detection techniques in network traffic: The role of artificial intelligence and machine learning. *J. Sci. Res. Technol.* **2024**, *2*, 38–48. [CrossRef]
25. Gaioto, F. Big Data Intrusion Detection Using AI-Based Supervised Classifiers and Machine Learning Ensembles for Cybersecurity Threat Prevention. Unpublished Work. 2023; submitted.
26. Alrajeh, N.A.; Lloret, J. Intrusion detection systems based on artificial intelligence techniques in wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2013**, *9*, 351047. [CrossRef]
27. Lopez-Martin, M.; Carro, B.; Sanchez-Esguevillas, A.; Lloret, J. Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in IoT. *Sensors* **2017**, *17*, 1967. [CrossRef]
28. Azam, Z.; Islam, M.M.; Huda, M.N. Comparative analysis of intrusion detection systems and machine learning based model analysis through decision tree. *IEEE Access* **2023**, *in press*. [CrossRef]
29. Mala, K.; Annapurna, H.S. Cloud network traffic classification and intrusion detection system using deep learning. In Proceedings of the 2023 International Conference Integrated Intelligence and Communication Systems (ICIICS), Bengaluru, India, 24–25 November 2023; pp. 1–8.
30. Rashid, A.; Siddique, M.J.; Ahmed, S.M. Machine and deep learning based comparative analysis using hybrid approaches for intrusion detection system. In Proceedings of the 2020 3rd International Conference Advancements in Computational Sciences (ICACS), Lahore, Pakistan, 17–19 February 2020. [CrossRef]
31. Sharon, A.; Mohanraj, P.; Abraham, T.E.; Sundan, B.; Thangasamy, A. An intelligent intrusion detection system using hybrid deep learning approaches in cloud environment. In *Computer, Communication, and Signal Processing*; Springer: Cham, Switzerland, 2022. [CrossRef]
32. Smys, S.; Basar, A.; Wang, H. Hybrid intrusion detection system for internet of things (IoT). *J. ISMAC* **2020**, *2*, 190–199. [CrossRef]
33. Hassan, M.M.; Gumaei, A.; Alsanad, A.; Alrubaian, M.; Fortino, G. A hybrid deep learning model for efficient intrusion detection in big data environment. *Inf. Sci.* **2020**, *513*, 386–396. [CrossRef]
34. Maheswari, M.; Karthika, R.A. A novel hybrid deep learning framework for intrusion detection systems in WSN-IoT networks. *Intell. Autom. Soft Comput.* **2022**, *33*, 365–382. [CrossRef]
35. 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]
36. Kalaivani, K.; Chinnadurai, M. A hybrid deep learning intrusion detection model for fog computing environment. *Intell. Autom. Soft Comput.* **2021**, *30*, in press, *in press*. [CrossRef]
37. Maseer, Z.K.; Yusof, R.; Mostafa, S.A.; Bahaman, N.; Musa, O.; Al-Rimy, B.A.S. DeepIoT.IDS: Hybrid deep learning for enhancing IoT network intrusion detection. *Comput. Mater. Continua* **2021**, *69*, 3946–3967. [CrossRef]

38. Shahin, M.; Chen, F.F.; Hosseinzadeh, A.; Bouzary, H.; Rashidifar, R. A deep hybrid learning model for detection of cyber attacks in industrial IoT devices. *Int. J. Adv. Manuf. Technol.* **2022**, *123*, 1973–1983. [CrossRef]
39. 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]
40. Sajid, M.; Malik, K.R.; Almogren, A.; Malik, T.S.; Khan, A.H.; Tanveer, J.; Rehman, A.U. Enhancing intrusion detection: A hybrid machine and deep learning approach. *J. Cloud Comput.* **2024**, *13*, 123. [CrossRef]
41. de Souza, C.A.; Westphall, C.B.; Machado, R.B.; Sobral, J.B.M.; Vieira, G.d.S. Hybrid approach to intrusion detection in fog-based IoT environments. *Comput. Netw.* **2020**, *180*, 107417. [CrossRef]
42. Booij, T.M.; Chiscop, I.; Meeuwissen, E.; Moustafa, N.; Den Hartog, F.T. ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets. *IEEE Internet Things J.* **2021**, *9*, 485–496. [CrossRef]
43. Rezaei, H.; Taheri, R.; Jordanov, I.; Shojafar, M. Federated RNN for Intrusion Detection System in IoT Environment Under Adversarial Attack. *J. Netw. Syst. Manag.* **2025**, *33*, 82. [CrossRef]
44. Nowroozi, E.; Haider, I.; Taheri, R.; Conti, M. Federated learning under attack: Exposing vulnerabilities through data poisoning attacks in computer networks. *IEEE Trans. Netw. Serv. Manag.* **2025**, *22*, 822–831. [CrossRef]
45. Liu, L.; Feng, J.; Li, J.; Chen, W.; Mao, Z.; Tan, X. Multi-layer CNN-LSTM network with self-attention mechanism for robust estimation of nonlinear uncertain systems. *Front. Neurosci.* **2024**, *18*, 1379495. [CrossRef]
46. Xu, G.; Ren, T.; Chen, Y.; Che, W. A one-dimensional CNN-LSTM model for epileptic seizure recognition using EEG signal analysis. *Front. Neurosci.* **2020**, *14*, 578126. [CrossRef]
47. Balla, A.; Habaebi, M.H.; Elsheikh, E.A.A.; Islam, R.; Suliman, F.E.M.; Mubarak, S. Enhanced CNN-LSTM deep learning for SCADA IDS featuring Hurst parameter self-similarity. *IEEE Access* **2024**, *12*, 6100–6116. [CrossRef]
48. Alkanhel, R.I.; Saleh, H.; Elaraby, A.; Alharbi, S.; Elmannai, H.; Alaklabi, S.; Mostafa, S. Hybrid CNN-GRU model for real-time blood glucose forecasting: Enhancing IoT-based diabetes management with AI. *Sensors* **2024**, *24*, 7670. [CrossRef]
49. Naidu, G.; Zuva, T.; Sibanda, E.M. A review of evaluation metrics in machine learning algorithms. In *Artificial Intelligence Application in Networks and Systems*; Springer: Cham, Switzerland, 2023. [CrossRef]
50. Powers, D.M. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv* **2020**, arXiv:2010.16061. [CrossRef]
51. Cabot, J.H.; Ross, E.G. Evaluating prediction model performance. *Surgery* **2023**, *174*, 723–726. [CrossRef] [PubMed]
52. Varoquaux, G.; Colliot, O. Evaluating machine learning models and their diagnostic value. In *Machine Learning for Brain Disorder*; Humana: New York, NY, USA, 2023; pp. 601–630.
53. GeeksforGeeks. *F1 Score in Machine Learning*; GeeksforGeeks: Noida, India, 2023.
54. 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]
55. Henry, A.; Gautam, S.; Khanna, S.; Rabie, K.; Shongwe, T.; Bhattacharya, P.; Chowdhury, S. Composition of hybrid deep learning model and feature optimization for intrusion detection system. *Sensors* **2023**, *23*, 890. [CrossRef]
56. Khacha, A.; Saadouni, R.; Harbi, Y.; Gherbi, C.; Harous, S.; Aliouat, Z. Robust intrusion detection for IoT networks: An integrated CNN-LSTM-GRU approach. In Proceedings of the 2023 International Conference Networking and Advanced Systems (ICNAS), Algiers, Algeria, 21–23 October 2023; pp. 1–6.
57. da Silva Ruffo, V.G.; Lent, D.M.B.; Carvalho, L.F.; Lloret, J.; Proença, M.L., Jr. Generative adversarial networks to detect intrusion and anomaly in IP flow-based networks. *Future Gener. Comput. Syst.* **2025**, *163*, 107531. [CrossRef]
58. Mokrani, S.; Belkadi, M.; Sadoun, T.; Lloret, J.; Aoudjit, R. LEA-RPL: Lightweight energy-aware RPL protocol for internet of things based on particle swarm optimization. *Telecommun. Syst.* **2025**, *88*, 14. [CrossRef]