informatics

MDPI

Article

Performance Evaluation of Deep Learning Models for Classifying Cybersecurity Attacks in IoT Networks

Fray L. Becerra-Suarez *, Victor A. Tuesta-Monteza, Heber I. Mejia-Cabrera and Juan Arcila-Diaz *

Grupo de Investigación en Inteligencia Artificial y Ciberseguridad, Universidad Señor de Sipán, Chiclayo 14000, Peru

* Correspondence: bsuarezf@uss.edu.pe (F.L.B.-S.); diarcilaju@uss.edu.pe (J.A.-D.)

Abstract: The Internet of Things (IoT) presents great potential in various fields such as home automation, healthcare, and industry, among others, but its infrastructure, the use of open source code, and lack of software updates make it vulnerable to cyberattacks that can compromise access to data and services, thus making it an attractive target for hackers. The complexity of cyberattacks has increased, posing a greater threat to public and private organizations. This study evaluated the performance of deep learning models for classifying cybersecurity attacks in IoT networks, using the CICIoT2023 dataset. Three architectures based on DNN, LSTM, and CNN were compared, highlighting their differences in layers and activation functions. The results show that the CNN architecture outperformed the others in accuracy and computational efficiency, with an accuracy rate of 99.10% for multiclass classification and 99.40% for binary classification. The importance of data standardization and proper hyperparameter selection is emphasized. These results demonstrate that the CNN-based model emerges as a promising option for detecting cyber threats in IoT environments, supporting the relevance of deep learning in IoT network security.

Keywords: Internet of Things (IoT); cybersecurity; deep learning; CICIoT2023; DNN; CNN; LSTM

Citation: Becerra-Suarez, F.L.; Tuesta-Monteza, V.A.; Mejia-Cabrera, H.I.; Arcila-Diaz, J. Performance Evaluation of Deep Learning Models for Classifying Cybersecurity Attacks in IoT Networks. *Informatics* **2024**, *11*, 32. https://doi.org/10.3390/informatics11020032

Academic Editors: Augusto Neto and Roger Immich

Received: 29 March 2024
Revised: 3 May 2024
Accepted: 8 May 2024
Published: 17 May 2024

1. Introduction

The term "Internet of Things" (IoT), also known as "Internet of Everything" or sometimes referred to as "industrial internet" [1], refers to a communication infrastructure in which multiple tangible devices are enabled to establish connections and communication over the global network known as the internet. These devices, commonly referred to as "smart objects", incorporate all kinds of sensors, software, and electronic components that allow them to capture information, store it, analyze it, process it, and share it with other devices and systems, making the surrounding environment smarter [2–4]. The variety of these smart objects can range from household devices to complex industrial machinery and transportation systems. It is estimated that by the year 2030, the number of devices connected to the internet could exceed 29 trillion [5].

The potential applications of IoT encompass a wide and diverse range of fields, and its impact is manifested in various industries, such as tourism [6], home healthcare [7], agriculture [8,9], and finance [10], among others. As the number of devices connected to the global network continues to grow, it has raised new concerns regarding security, stemming from the vulnerabilities presented by IoT devices, such as authentication, access control, device security, and heterogeneity, among others [2,11–13]. These vulnerabilities require defense strategies against potential attacks.

The proliferation and expansion of physical devices connected to the network make them attractive targets to be hijacked into botnets and used in attacks such as phishing and distributed denial of service (DDoS) [14]. Malware attacks like Mirai, Hajime, and Bashlite, among others, also pose a significant challenge to IoT security [15,16], along with web-based attacks [17]. To achieve the necessary level of protection, solutions based on

traditional approaches such as traffic protection systems, firewalls, and managed security services have been implemented. However, these measures are not sufficient to counter these attacks due to their complexity, and they are rule-based, which limits protection against network-circulating attacks [18].

Given the growing threat of cyberattacks on IoT networks, it is necessary to evaluate the performance of deep learning models for efficiently and accurately classifying these attacks. The effectiveness of these models in detecting and mitigating threats can significantly contribute to improving security in IoT environments, protecting both devices and the sensitive data transmitted through them.

To address these challenges, researchers and professionals have turned to deep learning models, which have demonstrated suitable temporal performance and high detection rates with great accuracy for classifying and mitigating threats in IoT networks. In reference [18], three deep learning models based on deep neural networks (DNN), convolutional neural networks (CNN), and recurrent neural networks (RNN) were implemented to detect cyberattacks from the CICIoT2023 dataset, where the RNN model achieved a higher accuracy of 96.56% for multiclass classification. In the work by Akgun et al. [19], they developed a hybrid model based on DNN, CNN, and long short-term memory (LSTM) to classify DDoS attacks from the CIC-DDoS2019 dataset, achieving an accuracy of 99.30% for multiclass classification. In the study presented by Wang et al. [20], they proposed a lightweight method called DL-BiLSMT, which combines bidirectional long short-term memory networks (BiLSTMs) and DNN. This method was evaluated using a subset of data from CICIDS2017, N-BaIoT, and CICIoT2023, achieving accuracy rates of 93.13%, 99.98%, and 99.67%, respectively. In reference [14], gated recurrent units (GRU), an advanced variant of LSTM, were implemented to detect multi-vectorial DDoS attacks, achieving accuracies of 99.82% and 99.85% on the CICDDoS2019 and CICIoT2023 datasets, respectively. Despite the favorable results, the training and testing time exceeds 60 min, which should also be considered.

In research [21], a DNN was implemented that achieved an accuracy of 99.14% on the CICIoT2023 dataset, although its precision was only 67.6%. In [22], a new model for feature selection from the CICIoT2023 dataset based on extra tree classifier was proposed, which was implemented with an LSTM, achieving a multiclass classification accuracy rate of 92%. In reference [23], a federated learning approach based on deep learning was employed to predict attacks using the CICIoT2023 dataset, reaching an experimental accuracy of 99%. Other classical machine learning algorithms have also been implemented. In the study by Le et al. [24], a blending model was developed as a combination of three classifiers: gradient boosting, random forest, and decision tree. The performance of this method showed an accuracy rate of 99.51% and 100% on the CICIoT2023 and IoTID20 datasets, respectively. In the study by [25], a comparative analysis of different machine learning approaches was conducted and evaluated with the ToN-IoT and BoT-IoT databases, demonstrating that the neural network-based model achieved the best result of 99.9% accuracy. Although the results obtained for the BoT-IoT dataset are not detailed, it appears to be a good option for predicting threats in IoT networks.

These models have shown high detection rates and accuracy in identifying threats, making them ideal candidates for protecting IoT devices against malicious attacks. The main objective of this study is to evaluate the performance of different deep learning models including DNN, LSTM, and CNN for classifying cybersecurity attacks in IoT networks related to DDoS, denial of service (DoS), recon, web, brute force, spoofing, and Mirai. The CICIoT2023 dataset was used, which underwent data preprocessing and feature selection, resulting in a new simplified dataset. The performance of these models was evaluated in terms of accuracy, precision, recall, and F1 score in threat identification, with the purpose of providing an effective tool for protecting IoT devices against malicious attacks.

2. Materials

Dataset

To conduct the present research, a dataset called CICIoT2023 [26] was employed, which is novel, extensive, and very recent. This dataset was designed to evaluate large-scale attacks in the Internet of Things (IoT) ecosystem. The CICIoT2023 was created by the Canadian Institute for Cybersecurity and provides a realistic representation of attacks in an IoT topology composed of 105 devices. This dataset includes 33 types of attacks, classified into categories such as DDoS, DoS, recon, web, brute force, spoofing, and Mirai. In total, the CICIoT2023 contains an impressive record of 46,686,579 events and presents 47 distinctive features. Table 1 provides a detailed description of the CICIoT2023 dataset. This table includes the type of attack, the request target, the total number of records, the percentage of records used for training and validation of the proposed models, and the percentage distribution of classes. The total number of records refers to the quantity of feature tuples extracted from the original pcap files, which are summarized within a fixed-size packet window. These features are derived from a sequence of packets carrying information between two hosts [26].

Table 1. Description of the CICIoT2023 Dataset.

Type	Target	Total Number of Records	Percentage of Records Used (1%)	Class Distribution
Benign	Benign	1,098,195	10,982	2.35%
DDoS	Attack	33,984,560	339,846	72.79%
DoS	Attack	8,090,738	80,907	17.33%
Mirai	Attack	2,634,124	26,341	5.64%
Recon	Attack	354,565	3546	0.76%
Spoofing	Attack	486,504	4865	1.04%
Web	Attack	24,829	248	0.05%
Bruteforce	Attack	13,064	131	0.03%
Total		46,686,579	466,866	

The different deep learning models were implemented in Google Colab, an online platform that provides free access to computing resources such as GPUs and TPUs. This tool proved to be essential for the development of the research, as it allowed for the efficient and scalable execution of deep learning models, significantly reducing training times and facilitating experimentation with different architectures and parameters.

3. Methods

3.1. Preprocessing, Feature Selection, and Data Standardization

The data preprocessing stage constitutes the most crucial and important phase in supervised learning, during which a series of transformations and adjustments are applied to the data with the purpose of improving both the data quality and the results of any machine learning model [27]. The database contains 46 features and 46,686,579 records that host different attacks on IoT devices.

With the aim of simplifying the database, improving processing time, and reducing the required memory space, the following data cleaning and selection actions were performed. Outliers such as null values, duplicates, empty values, positive infinity, and negative infinity were removed, resulting in the elimination of a total of 34 records. However, due to the large number of remaining records, it was decided to use only 1% of the total records, equivalent to 466,866 records randomly selected in each attack category, as described in Table 1. This random sampling strategy allowed for reducing computational load and maintaining data representativeness in subsequent analysis. Additionally, six features were identified that contained only zero values in all records, which were removed. These features are "ece_flag_number", "cwr_flag_number", "Telnet", "SMTP", "IRC", and "DHCP". As a result, a new dataset with 40 features and another one with the "label" value used as a tag containing the name of each attack category was obtained. The details of these features are presented in Table 2 considering their maximum and minimum values for each feature.

Table 2. Final features of the new simplified dataset.

#	Characteristics	Minimum and Maximum Values	#	Characteristics	Minimum and Maximum Values
1	flow_duration	[0;68,378.35]	21	SSH	[0;1]
2	Header_Length	[0;9,861,631.0]	22	TCP	[0;1]
3	Protocol Type	[0;47]	23	UDP	[0;1]
4	Duration	[0;255]	24	ARP	[0;1]
5	Rate	[0;7,340,032.0]	25	ICMP	[0;1]
6	Srate	[0;7,340,032.0]	26	IPv	[0;1]
7	Drate	[0;1.232]	27	LLC	[0;1]
8	fin_flag_number	[0;1]	28	Tot sum	[42;8,5296.6]
9	syn_flag_number	[0;1]	29	Min	[42;3380.3]
10	rst_flag_number	[0;1]	30	Max	[42;27,052]
11	psh_flag_number	[0;1]	31	AVG	[42;7618.42]
12	ack_flag_number	[0;1]	32	Std	[0; 6961.53]
13	ack_count	[0;4.6]	33	Tot size	[42; 4483.9]
14	syn_count	[0;7.9]	34	IAT	[0;167,639,419.98]
15	fin_count	[0;27.2]	35	Number	[1;13.5]
16	urg_count	[0;3466.6]	36	Magnitude	[9.17;121.46]
17	rst_count	[0;8838.5]	37	Radius	[0;9865.62]
18	HTTP	[0;1]	38	Covariance	[0;48,937,857.68]
19	HTTPS	[0;1]	39	Variance	[0;1]
20	DNS	[0;1]	40	Weight	[1;244.6]

Given the different scales of each descriptor, the dataset was standardized using the StandardScaler() method to transform the data so that the mean of the resulting distribution is zero and the standard deviation is one. This transformation is achieved by subtracting the mean value of each observation and dividing by the standard deviation, as shown in Equation (1).

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

where z is the transformed value of the feature, x is the original value of each descriptor, μ is the mean, and σ is the standard deviation of the feature in the dataset.

3.2. Proposed Models of Deep Learning Architecture

Deep learning has emerged as an efficient and high-performance machine learning method for solving very complex classification and prediction problems.

Different performance comparisons of two types of deep learning architectures were conducted using the new dataset obtained from CICIoT2023. Based on these two models, a new model was proposed. Figure 1 illustrates the principle of operation of the system proposed in this research. The new dataset is divided into training, testing, and tuning through random selection. In each of the models, the selection of suitable hyperparameters includes network size, types of layers, activation functions, and optimizers. Therefore, we experimented with different types of layers for each proposed model, such as dense layer, LSTM layer, and convolutional layers of the proposed model.

The first architecture employed is based on a DNN, as shown in Figure 2a, consisting of five densely connected layers. This architecture begins with an input layer that has a number equal to the descriptors in the training data. It is followed by several dense layers, each with a "ReLU" activation function, introducing nonlinearities into the network. The dense layers have 64, 128, 256, 512, 256, and 128 neurons, respectively, progressively increasing the complexity and learning capacity of the model. A dropout layer with a rate of 0.2 was added to regularize the model and reduce overfitting. Then, a batch normalization layer was incorporated to normalize the activation of each layer and accelerate training. Finally, the output layer has the number of neurons equal to the number of classes in the simplified dataset, with a "softmax" activation function for multiclass classification.

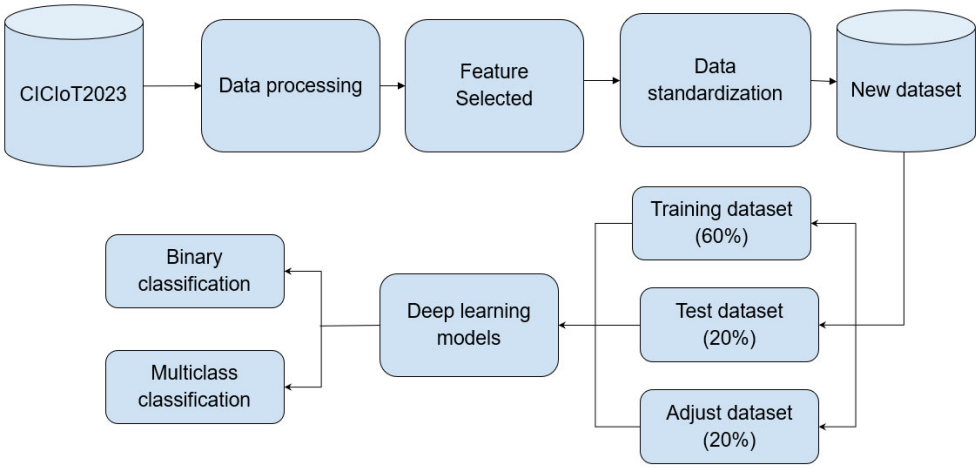

Figure 1. General Scheme of the Proposed Method.

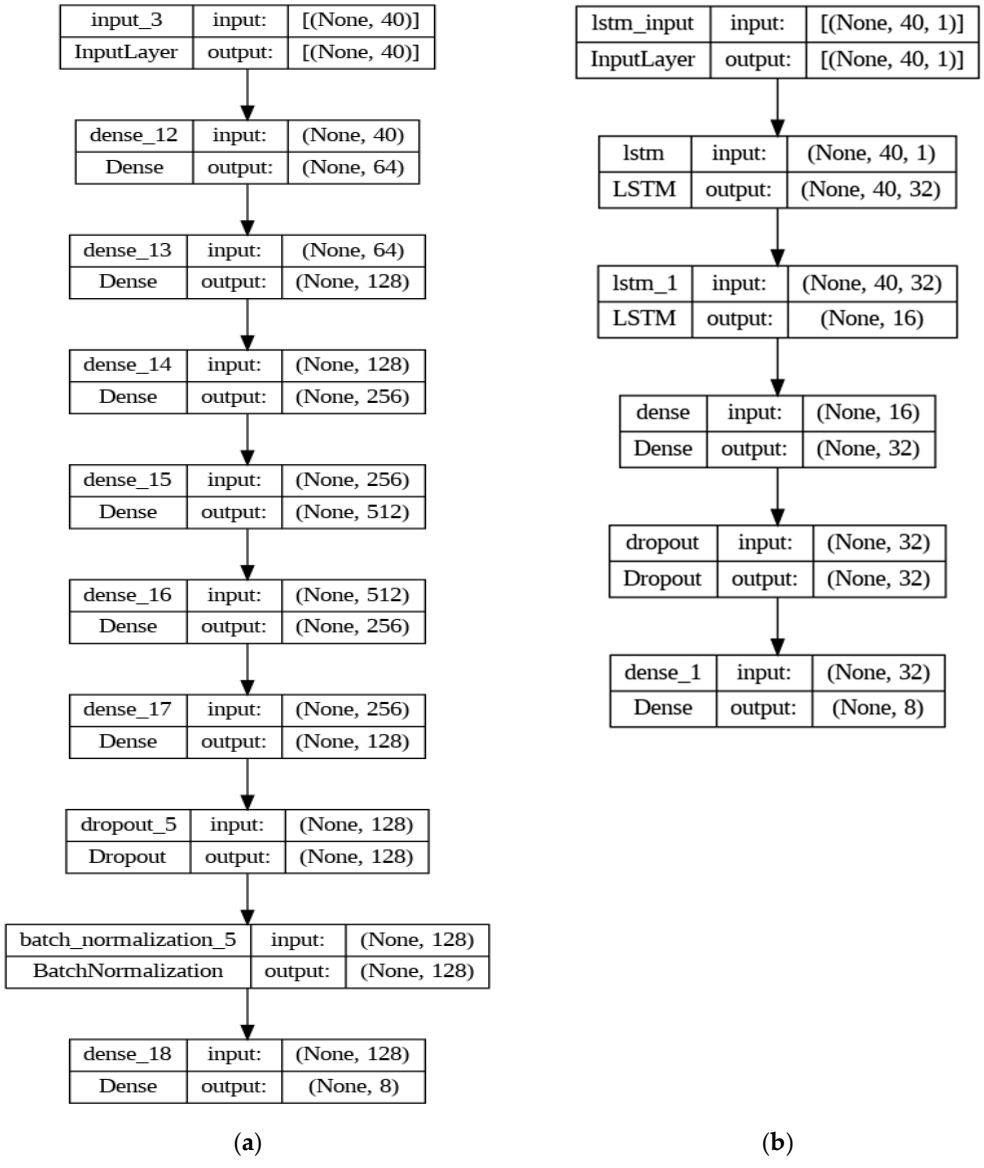

Figure 2. Reference Models of Deep Learning. In (**a**), the architecture based on a DNN is presented, and in (**b**), the architecture based on LSTM is presented.

The second architecture is based on a long short-term memory (LSTM) recurrent neural network model, as shown in Figure 2b. This architecture begins with an LSTM layer with 32 units and "return_sequences=True", meaning it returns complete sequences instead of just the last output. The LSTM layer has an input shape equal to the number of descriptors in the training dataset; next is another LSTM layer with 16 units, followed by a dense layer with 32 units and the ReLU activation function. A dropout layer with a rate of 0.2 is included. Finally, the output layer has a number of units equal to the number of classes in the simplified dataset.

The third architecture, as detailed in Figure 3, is based on a CNN consisting of several layers that process one-dimensional data. The input layer, like in the previous models, is equal to the number of descriptors in the training dataset. The first section of the network consists of three parallel convolutional branches, each followed by a ReLU activation function. Each branch has a convolutional filter with different kernel sizes (3, 5, and 11) and the same number of filters (64), allowing the network to capture patterns at different scales. The outputs of these branches are then concatenated along the last axis (axis = −2), meaning they are combined along the features. After concatenation, an additional convolutional layer with 72 filters and a kernel size of 7 was added, followed by a MaxPooling layer to reduce dimensionality. Next, the output was flattened to feed into dense layers, which are used to learn more abstract representations of the data. A dense layer with 256 neurons with a ReLU activation function was added, followed by a dropout layer with a rate of 0.2 to regularize the network and prevent overfitting. Finally, the output layer has a number of units equal to the number of classes in the dataset and uses the softmax activation function for multiclass classification.

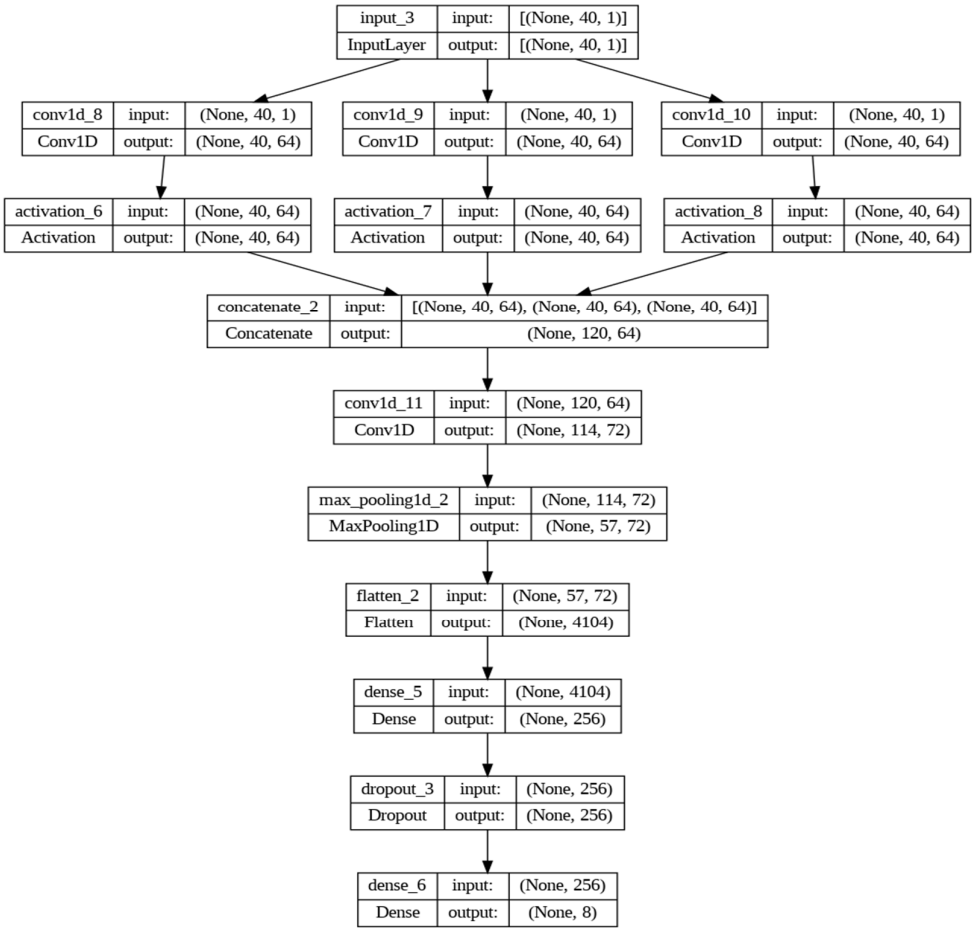

Figure 3. Architecture Model based on CNN.

3.3. Performance Evaluations

The detection of an attack can be classified as true positive (TP) or true negative (TN) when accurate detections are made on the attacks. Conversely, it is classified as false positive (FP) or false negative (FN) when detections are incorrect.

To evaluate the performance of the proposed models, metrics based on the confusion matrix were used to assess the models' ability to classify the different classes in the simplified dataset.

The metrics evaluated in this study were as follows:

Precision (P) is a metric used to measure the quality of predictions, minimizing false positives and maximizing the number of correctly classified true positives. It can be calculated using Formula (2).

$$P = \frac{TP}{TP + FP} \tag{2}$$

Recall (R): Assesses the classification accuracy of all elements within a given class.

$$\boldsymbol{R = \frac{TP}{TP + FN}} \tag{3}$$

F1 Score (F): This indicator provides a balance between precision and recall, allowing for a better comparison of combined performance.

$$F = 2 * \frac{(P * R)}{(P + R)} \tag{4}$$

Accuracy (Acc): Evaluating the prediction that the algorithm makes correctly and returning an accurate classification.

$$\boldsymbol{Acc = \frac{TP + TN}{TP + TN + FP + FN}} \tag{5}$$

4. Results

The configuration of the proposed deep learning models was carried out as follows. The training of each model was conducted for 50 epochs, using the Adam optimizer with a learning rate of 0.001, which is a popular choice in deep learning due to its ability to adapt dynamically to the learning rate during training, which can improve model convergence. Additionally, the "sparse_categorical_crossentropy" loss function was employed, suitable for classification problems with multiple classes. During this phase, callbacks were included to enhance the model's generalization ability and prevent overfitting.

The callbacks used in the model training include ModelCheckpoint, which saves the model with the best score for "val_accuracy" on the test set; ReduceLROnPlateau, which monitors "val_loss" and reduces the learning rate if the loss stops improving, with a reduction factor of 0.1, patience of 3, and a minimum learning rate limit (min_lr) of 1×10^{-7}. Additionally, CSVLogger was used to log the training progress to a CSV file, TensorBoard for visualizing training and validation metrics, and EarlyStopping to stop training if "val_loss" did not improve after six epochs. These callbacks were used to improve the performance and generalization of each model during the training and testing phase.

In Figures 4–6, the training and validation results for the accuracy and loss metrics of each proposed model are presented. It can be observed that, in all three models, accuracy increases in both datasets, indicating that the models learned to classify the data correctly. Additionally, the loss decreases steadily in each epoch, indicating that the models learned to minimize the error between their predictions and the actual labels.

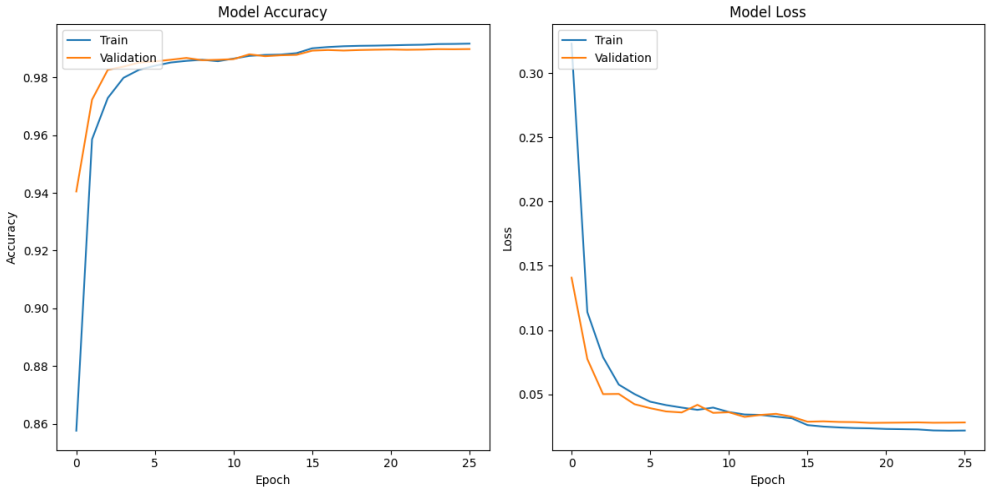

Figure 4. Training and validation results for the DNN-based architecture.

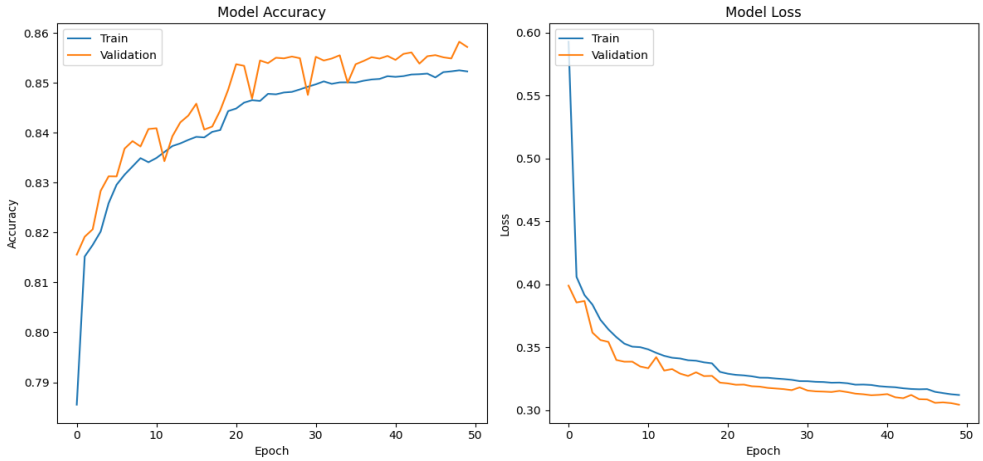

Figure 5. Training and validation results for the LSTM-based architecture.

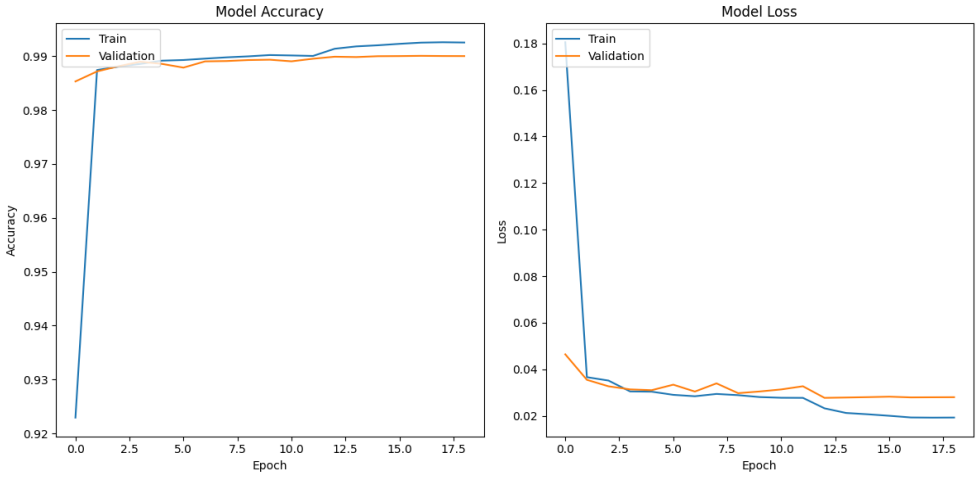

Figure 6. Training and validation results for the CNN-based architecture.

The CNN-based model shows the best results, with a val_loss and val_accuracy of 0.0279 and 0.9901 after 17 epochs, respectively, followed by the DNN-based model, which achieved a val_loss and val_accuracy of 0.02771 and 0.9896 after 25 epochs, respectively. In contrast, the LSTM model required more training and validation epochs but did not surpass 0.9 in the val_accuracy metric.

To evaluate the effectiveness of the proposed models in classifying threats in an IoT network, the results of the confusion matrix are shown in Figures 7–9. In this matrix, the rows represent the true labels, and the columns represent the predictions of the deep learning model. The darker the color of a square on the diagonal of the matrix, the higher the number of samples correctly predicted for that category.

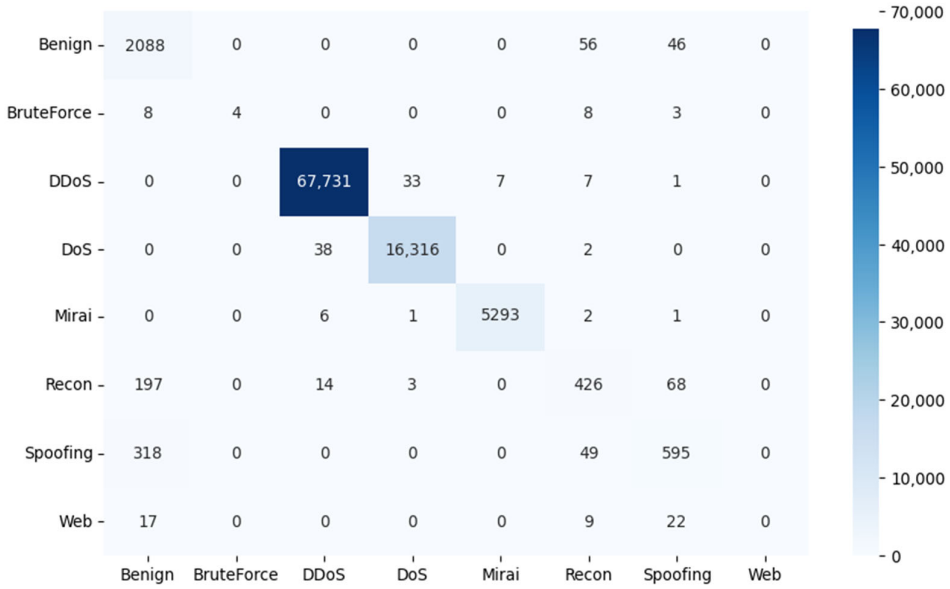

Figure 7. Confusion matrix results for the DNN architecture.

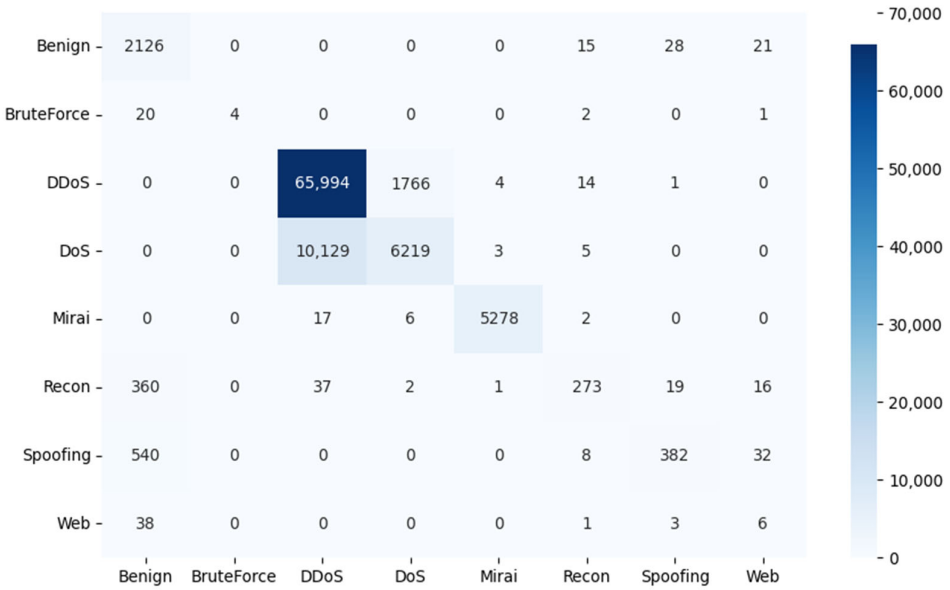

Figure 8. Confusion matrix results for the LSTM architecture.

In Table 3, the evaluation of the proposed models using confusion matrix metrics is presented, highlighting the superior performance of the CNN model compared to the DNN and LSTM models. The CNN model achieved the highest accuracy with a value of 99.10%, indicating its ability to correctly classify threats in IoT environments. Additionally, its precision (99.08%) and recall (99.10%) scores further support its effectiveness in both identifying true positives and minimizing false positives and false negatives, respectively.

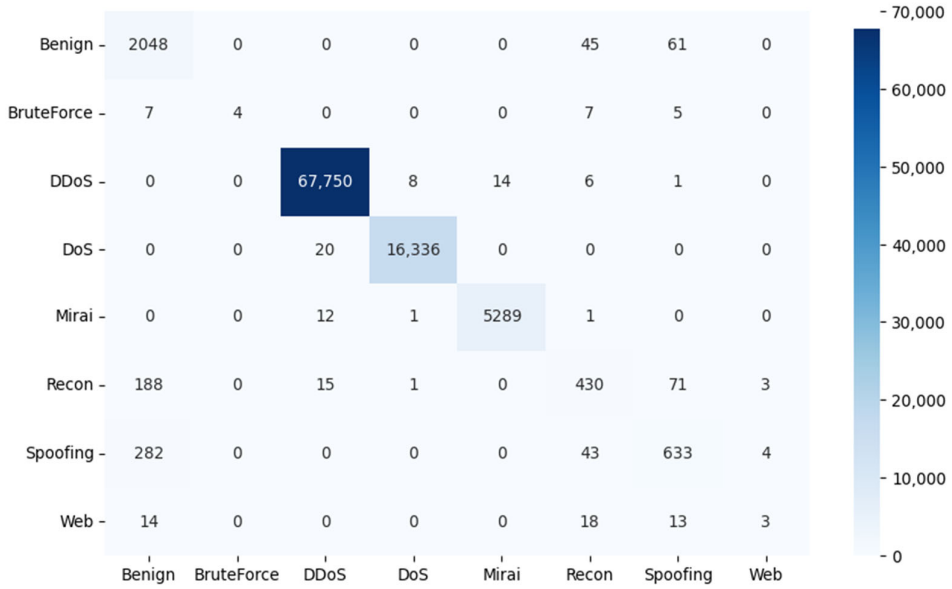

Figure 9. Confusion matrix results for the CNN architecture.

Table 3. Performance of deep learning architectures for multiclass classification.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Model Size	Training Time (Seconds)	Inference Time (Seconds)
DNN	99.02	98.97	99.02	98.95	1.30 MB	2372	11
LSTM	85.98	85.37	85.98	84.03	32.41 KB	3701	25
CNN	99.10	99.08	99.10	99.05	4.15 MB	767	6

Although precision and accuracy are important metrics for evaluating a model's performance, the F1 score provides a more comprehensive assessment, particularly when it comes to classifying multiple classes of attacks in IoT networks. By combining precision and recall into a single metric, the F1 score helps to demonstrate the predictive capability and effectiveness of the models in this specific context. Given the presence of class imbalance in the dataset, as evidenced in Table 1, the F1 score becomes a crucial metric, as this imbalance could significantly impact the interpretation of precision and accuracy. In the case of the CNN-based model, it achieved the best result of 99.05% for the F1 metric, validating its superior performance despite the class imbalance.

Another important aspect to consider is the inference time used by each model. Firstly, it was observed that the size of the CNN model is considerably larger compared to the other two models, at 4.15 megabytes, while the LSTM model has a much smaller size, at only 32.41 kilobytes. However, despite its size, the CNN model requires less time for both training and inference compared to the DNN and LSTM models. The CNN model showed the shortest training time at 767 s and the fastest inference time at just 6 s. This suggests that, despite its greater complexity and size, the CNN model manages to maintain notable efficiency in terms of processing time compared to the other models.

Regarding binary classification, the last layer of each proposed architecture in this study consists of a dense layer with a single output that uses the "sigmoid" function for comparing the two classes. In this case, all threats from the preprocessed dataset were grouped into the "Attack" class, while the other class remained named "Benign". According to the results in Table 4, the CNN-based model achieved the highest accuracy score, at 99.40%, closely followed by the other models. Additionally, this model also yielded the shortest training time compared to the others.

Table 4. Performance of deep learning architectures for binary classification.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Model Size	Training Time (Seconds)	Inference Time (Seconds)
DNN	99.38	99.42	99.38	99.40	1.30 MB	1379	6
LSTM	99.36	99.39	99.36	99.37	31.63 KB	1036	10
CNN	99.40	99.43	99.40	99.41	4.14 MB	618	7

Table 5 presents a comparative analysis of various models used in threat detection in IoT networks, including the present study, which utilized the CICIoT2023 dataset. Each entry describes the bibliographic reference, model architecture, number of descriptors, evaluation metrics, and temporal metrics regarding training and inference times. Compared to the previously mentioned studies, the proposed CNN model in this work exhibited equal or superior accuracy compared to others. Compared to the previously mentioned studies, the proposed CNN model in this work exhibited an equal or superior F1 score compared to others, except for in the study [24], which outperformed our results by 0.02% in the F1 metric. However, it is worth noting that the study [24] was based solely on six features, namely "IAT", "Magnitude", "Total size", "Minimum", "Flow duration", and "Total sum", implying lower data complexity, which could explain their superior results in F1. It is important to highlight the need to consider more features that were not included, especially given the threshold restriction they used after applying the MDI method.

Table 5. Comparison of results obtained with other studies.

Reference	Approach	Number of Features	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Training Time (Seconds)	Inference Time (Seconds)
[18]	DNN	47	88.64	91.2	88.64	88.51	-	-
	CNN		96.37	96.15	96.37	95.51	-	-
	RNN		96.52	96.25	96.52	95.73	-	-
[20]	CNN	-	92.21	91.49	92.22	91.26	1515.4	7.2
	RNN		92.73	91.24	92.73	91.50	717.8	8.5
	LSTM		92.75	91.32	92.75	91.52	764.8	6.6
	BiLSTM		93.05	91.33	93.05	91.73	792.6	8.0
	DL-BiLstm		93.13	91.80	93.13	91.94	708.4	6.4
[21]	DNN	47	99.11	67.94	90.66	97.72	-	-
[24]	Blending Model (DT, RF, GB)	6	99.51	98.51	99.63	99.07	448.10	3.89
[14]	GRU	-	99.85	-	-	-	3831	-
Our study	CNN	40	99.10	99.08	99.10	99.05	767	6

Additionally, the training and inference times are significantly better, suggesting the effectiveness of the proposed model. It is relevant to highlight that the CNN architecture used in this study achieves comparable or superior results with reduced computational burden, positioning it as a promising approach for cybersecurity threat detection in a realistic IoT environment.

5. Conclusions

This article evaluated the performance of different deep learning architectures (DNN, LSTM, and CNN) for classifying attacks in the IoT ecosystem. A recent and realistic dataset, CICIoT2023, containing eight attack categories and a class of benign records, was used. An analysis and removal of outliers and irrelevant features for the models were performed, resulting in a new dataset with 40 features. Due to the large number of records in CICIoT2023, 1% of each class was randomly selected and divided into 60% for training, 20% for testing, and 20% for validation. The results show that the proposed CNN-based model achieved an accuracy of 99.10% in multiclass classification and 99.40% in binary classification, outperforming other models in the study and those from another research. Additionally, the inference time of the proposed model is reasonable compared to

reference models, demonstrating its effectiveness in detecting various types of attacks in IoT networks.

The results obtained with the LSTM-based model were unfavorable. Although this architecture is highly effective for modeling long temporal sequences and capturing long-term dependencies in data, it does not suit the context of attack detection in IoT networks. This is because the structure of the sequences of the 40 descriptors lacks significant long-term dependencies for classification; instead, they are characterized by being local attributes independent of time. Additionally, the LSTM network tends to overfit with data that exhibit less temporal sequentiality, as is the case with the preprocessed database used in this study. Despite employing 50 training epochs, resulting in a total training time of 3701 s, the results were not favorable. This suggests a limited capacity for the LSTM network to progressively adapt to the dataset.

The DNN model, whose results were slightly inferior to those of the convolutional neural network, attributes its performance to a simpler architecture, limiting its ability to effectively capture the characteristics of complex data such as attack patterns in IoT environments, thereby affecting its ability to perform accurate classification. On the other hand, the CNN-based architecture proved to be more efficient in extracting relevant features from attack data in IoT networks. It was observed that the CNN has an effective capacity to adapt to complex data, thanks to the three convolutional layers included in its architecture, each with a kernel size of 3, 5, and 11, and 64 filters per layer, allowing it to capture patterns at different scales of the input data. Subsequently, the outputs of these layers were concatenated, and an additional convolutional layer with 72 filters and a kernel size of 7 was added, contributing to its high performance in the classification of attacks in IoT environments.

Based on the results obtained, it is confirmed that the CNN architecture is the most suitable for classifying attacks in IoT networks, surpassing DNN and LSTM architectures in the various evaluated metrics and inference time. In order to expand knowledge in this field, it is suggested to explore other deep learning architectures with potential for classifying attacks in IoT networks, evaluate the performance of proposed models on diverse IoT attack datasets, and develop real-time intrusion detection systems for IoT networks. It is expected that the results of this study will contribute significantly to the advancement in the field of IoT network security, paving the way for the development of more effective and efficient intrusion detection systems.

Author Contributions: Conceptualization, investigation, methodology, validation, supervision, and formal analysis, F.L.B.-S.; writing, review, editing, and visualization, F.L.B.-S. and J.A.-D.; software, data curation, project administration, and funding acquisition, F.L.B.-S., J.A.-D., V.A.T.-M. and H.I.M.-C. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by Universidad Señor de Sipán (Perú).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data may be provided free of charge to interested readers by requesting the correspondence author's email.

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

References

1. Gerodimos, A.; Maglaras, L.; Ferrag, M.A.; Ayres, N.; Kantzavelou, I. IoT: Communication protocols and security threats. *Internet Things Cyber-Phys. Syst.* **2023**, *3*, 1–13. [CrossRef]
2. Mishra, N.; Pandya, S. Internet of Things Applications, Security Challenges, Attacks, Intrusion Detection, and Future Visions: A Systematic Review. *IEEE Access* **2021**, *9*, 59353–59377. [CrossRef]
3. Alahmadi, A.A.; Aljabri, M.; Alhaidari, F.; Alharthi, D.J.; Rayani, G.E.; Marghalani, L.A.; Alotaibi, O.B.; Bajandouh, S.A. DDoS Attack Detection in IoT-Based Networks Using Machine Learning Models: A Survey and Research Directions. *Electronics* **2023**, *12*, 3103. [CrossRef]

4. Haque, S.; El-Moussa, F.; Komninos, N.; Muttukrishnan, R. A Systematic Review of Data-Driven Attack Detection Trends in IoT. *Sensors* **2023**, *23*, 7191. [CrossRef]
5. IoT Connected Devices Worldwide 2019–2030. Statista. Available online: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (accessed on 31 October 2023).
6. Popova, P.; Marinova, K.; Popov, V. Internet of Things and Big Data Analytics for Risk Management in Digital Tourism Ecosystems. *Risks* **2023**, *11*, 180. [CrossRef]
7. Fehringer, H.; Stary, C. User-Informed Adaptation in IoT Home Healthcare: Grounding Development in Empirical Evidence. *J. Theor. Appl. Electron. Commer. Res.* **2023**, *18*, 1901–1925. [CrossRef]
8. Alahmad, T.; Neményi, M.; Nyéki, A. Applying IoT Sensors and Big Data to Improve Precision Crop Production: A Review. *Agronomy* **2023**, *13*, 2603. [CrossRef]
9. Ndjuluwa, L.N.P.; Adebisi, J.A.; Dayoub, M. Internet of Things for Crop Farming: A Review of Technologies and Applications. *Commodities* **2023**, *2*, 367–381. [CrossRef]
10. Allioui, H.; Mourdi, Y. Exploring the Full Potentials of IoT for Better Financial Growth and Stability: A Comprehensive Survey. *Sensors* **2023**, *23*, 8015. [CrossRef] [PubMed]
11. Aldhaheri, A.; Alwahedi, F.; Ferrag, M.A.; Battah, A. Deep learning for cyber threat detection in IoT networks: A review. *Internet Things Cyber-Phys. Syst.* **2024**, *4*, 110–128. [CrossRef]
12. Chaudhary, S.; Mishra, P.K. DDoS attacks in Industrial IoT: A survey. *Comput. Netw.* **2023**, *236*, 110015. [CrossRef]
13. Kumari, P.; Jain, A.K. A comprehensive study of DDoS attacks over IoT network and their countermeasures. *Comput. Secur.* **2023**, *127*, 103096. [CrossRef]
14. Aguru, A.D.; Erukala, S.B. A lightweight multi-vector DDoS detection framework for IoT-enabled mobile health informatics systems using deep learning. *Inf. Sci.* **2024**, *662*, 120209. [CrossRef]
15. Lei, T.; Xue, J.; Wang, Y.; Baker, T.; Niu, Z. An empirical study of problems and evaluation of IoT malware classification label sources. *J. King Saud Univ.—Comput. Inf. Sci.* **2024**, *36*, 101898. [CrossRef]
16. Affinito, A.; Zinno, S.; Stanco, G.; Botta, A.; Ventre, G. The evolution of Mirai botnet scans over a six-year period. *J. Inf. Secur. Appl.* **2023**, *79*, 103629. [CrossRef]
17. Kaur, B.; Dadkhah, S.; Shoeleh, F.; Neto, E.C.; Xiong, P.; Iqbal, S.; Lamontagne, P.; Ray, S.; Ghorbani, A.A. Internet of Things (IoT) security dataset evolution: Challenges and future directions. *Internet Things* **2023**, *22*, 100780. [CrossRef]
18. Abbas, S.; Bouazzi, I.; Ojo, S.; Al Hejaili, A.; Sampedro, G.A.; Almadhor, A.; Gregus, M. Evaluating deep learning variants for cyber-attacks detection and multi-class classification in IoT networks. *PeerJ Comput. Sci.* **2024**, *10*, e1793. [CrossRef] [PubMed]
19. Akgun, D.; Hizal, S.; Cavusoglu, U. A new DDoS attacks intrusion detection model based on deep learning for cybersecurity. *Comput. Secur.* **2022**, *118*, 102748. [CrossRef]
20. Wang, Z.; Chen, H.; Yang, S.; Luo, X.; Li, D.; Wang, J. A lightweight intrusion detection method for IoT based on deep learning and dynamic quantization. *PeerJ Comput. Sci.* **2023**, *9*, e1569. [CrossRef]
21. Neto, E.C.P.; Dadkhah, S.; Ferreira, R.; Zohourian, A.; Lu, R.; Ghorbani, A.A. CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment. *Sensors* **2023**, *23*, 5941. [CrossRef]
22. Khanday, S.A.; Fatima, H.; Rakesh, N. A Novel Data Preprocessing Model for Lightweight Sensory IoT Intrusion Detection. *Int. J. Math. Eng. Manag. Sci.* **2024**, *9*, 188–204. [CrossRef]
23. Abbas, S.; Al Hejaili, A.; Sampedro, G.A.; Abisado, M.; Almadhor, A.; Shahzad, T.; Ouahada, K. A Novel Federated Edge Learning Approach for Detecting Cyberattacks in IoT Infrastructures. *IEEE Access* **2023**, *11*, 112189–112198. [CrossRef]
24. Le, T.-T.-H.; Wardhani, R.W.; Putranto, D.S.C.; Jo, U.; Kim, H. Toward Enhanced Attack Detection and Explanation in Intrusion Detection System-Based IoT Environment Data. *IEEE Access* **2023**, *11*, 131661–131676. [CrossRef]
25. Inuwa, M.M.; Das, R. A comparative analysis of various machine learning methods for anomaly detection in cyber attacks on IoT networks. *Internet Things* **2024**, *26*, 101162. [CrossRef]
26. CIC IoT Dataset 2023. Available online: https://www.unb.ca/cic/datasets/iotdataset-2023.html (accessed on 31 October 2023).
27. Frye, M.; Mohren, J.; Schmitt, R.H. Benchmarking of Data Preprocessing Methods for Machine Learning-Applications in Production. *Procedia CIRP* **2021**, *104*, 50–55. [CrossRef]