



JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGY

<https://e-journal.uum.edu.my/index.php/jict>

How to cite this article:

Benamor, Z., Zianou, A. S., & Mounir, H. (2026). Scatter plot-based tabular-to-image conversion method with a modified VGG16 model for intrusion detection in IoT networks. *Journal of Information and Communication Technology*, 25(4), 1-20. <https://doi.org/10.32890/jict2026.25.4.1>

Scatter Plot–Based Tabular-to-Image Conversion Method with a Modified VGG16 Model for Intrusion Detection in IoT Networks

¹Zahia Benamor, ²Ahmed Seghir Zianou & ³Hemam Mounir

^{1,2&3}ICOSI Laboratory, Khenchela, Algeria

^{1,2&3}Department of Computer Science,

Abbes Laghrour University Khenchela, Algeria

^{*1} benamor.zahia@univ-khenchela.dz

² zianou.ahmed.saghir@univ-khenchela.dz

³ hemam.mounir@univ-khenchela.dz

^{*}Corresponding author

Received: 27/10/2025

Revised: 18/3/2026

Accepted: 12/8/2026

Published: 31/10/2026

ABSTRACT

Security is one of the most severe problems in the Internet of Things (IoT) environment. Early detection of attacks is crucial, as it can significantly enhance network security. Convolutional neural networks (CNNs) represent a significant advancement in image classification; however, the application of CNNs to the classification of tabular data remains less explored for Intrusion Detection Systems (IDSs) compared to their success in image-based tasks. In this paper, we develop and evaluate a novel method for the classification of data using a modified VGG16 model by transforming tabular data into images. This transformation handles each block of samples as an image. A modified VGG16 model is then trained to classify the transformed images. We applied this method to the CICIoT2023 cybersecurity dataset, and the model achieved an accuracy of 98.05% with precision, recall, and F1-score of 98.15%, 98.05%, and 98.04%, respectively. The experimental results show that the proposed method achieves superior detection performance and greater classification robustness compared to other state-of-the-art methods. The significance of this study lies in detecting intrusions in IoT networks effectively and efficiently by utilizing tabular to image transformation and VGG16 model.

Keywords: CICIoT2023, CNN, Internet of Things, intrusion detection systems, tabular-to-image conversion.

INTRODUCTION

The exponential increase in data volume has led to a rise in network intrusions. These incidents increasingly target critical infrastructure, including industrial control systems, IoT devices, and cloud and telecommunications services (S. Wang et al., 2023). Network intrusion refers to undesirable, harmful, and unauthorized access or activity within a network domain. It involves unauthorized access to networked systems to steal information and resources, thereby jeopardizing network security. These intrusions can be passive, operating stealthily, or active, directly affecting network resources and bandwidth (Devendiran & Turukmane, 2024; Latif et al., 2022). Therefore, using of security technique is essential to detect network intrusions and protect information from them. Antivirus software, firewalls and intrusion detection systems, are used to improve security.

The primary function of an IDS is to monitor network traffic for suspicious activity or any changes in system behaviour and alert the system or network administrator if any irregularities are detected. Based on their data source, IDSs are either Network-based (NIDS) or Host-based (HIDS). NIDS analyse network traffic flows, while HIDS examines local system logs (Liu et al., 2020). Additionally, there is another classification of IDS based on their detection method, namely anomaly-based and signature-based (or misuse-based). Anomaly detection establishes a model of normal behaviour to identify abnormalities, enabling the detection of novel attacks but often causing a high false positive rate. In contrast, signature-based detection relies on a database of predefined attack signatures to identify known threats accurately, though it cannot detect unknown or new attacks (Devendiran & Turukmane, 2024). The emergence of Artificial Intelligence (AI) has enabled its incorporation into IDS for detecting complex and sophisticated attacks, and solving some traditional IDS issues, such as insufficient feature extraction and inaccurate model classification.

The integration of AI to improve the ability of NIDS to detect intrusions, particularly through Machine Learning (ML) and Deep Learning (DL), has revolutionised IDS by enabling robust detection of complex cyber threats. While ML algorithms like Support Vector Machine (SVM) and Random Forest (RF) effectively analyse network data, they require manual feature engineering and often exhibit lower detection rates (Akash et al., 2022; Al-Na'amneh et al., 2023). DL techniques have been adopted to overcome these limitations. DL achieves superior performance due to its ability to automatically extract features. This allows DL models to directly process high-dimensional data, removing the need for manual feature engineering, enabling them to excel at identifying complex patterns, and reducing the time required to process data (Rodríguez et al., 2022; Sun & Wang, 2025). These capabilities of DL demonstrate its immense potential in the field of image analysis. Tabular data is fundamental to IDS due to its structured nature and compatibility with ML methods. However, challenges such as data heterogeneity, class imbalance, and the need for interpretability and scalability must be carefully addressed to build robust intrusion detection systems (Jiang et al., 2025).

Since network traffic is a non-image format and lacks spatial structure, there is a significant need to convert tabular traffic data into images to benefit from the powerful learning capabilities of CNNs (S. Wang et al., 2023). In recent years, the applications of DL schemes for IDS have emerged as a new field of study and have become an attractive area for researchers. Several techniques of converting network traffic into images have been proposed in the literature, and they differ from one another, some of them convert network traffic to grayscale (Al-Turaiqi & Altwaijry, 2021), RGB images (Choi et al., 2020), and serialised images (Nazarri et al., 2023). Besides, some researchers adopt existing methods from other fields such as: DeepInsight (Sharma et al., 2019) to transform genetic information, Image Generator for Tabular Dataset (IGTD) algorithm (Zhu et al., 2021) in the field of cancer cell line (CCL)

gene expression data and molecular descriptors of drugs. By transforming tabular data into images, these methods enable the application of powerful CNN models to tabular network traffic, and enhance model performance.

Although a number of studies in this field have been performed to investigate intrusion system vulnerabilities and have proposed techniques to convert network traffic data into images, the detection and classification of attacks using CNN with image data are still at an early stage and deserve more attention. Moreover, detecting different and new attack classes extracted from a real-life IoT scenario is a requirement. Motivated by these limitations, we addressed the challenge of applying DL to tabular data for network intrusion detection by proposing a new technique to convert network traffic data into images. This technique maps features into a two-Dimensional space, and preserves relationships between features. Additionally, we modified VGG16 model and adapted it for classifying generated images targeting intrusion systems in IoT networks, as VGG16 is designed for large natural images and has a high computational cost. The following section introduces related works that exist in this field, establishing the context for the study. The subsequent section describes the proposed methodology, followed by a detailed presentation and discussion of the results. Finally, the overall conclusions and implications of the study are outlined to conclude the paper.

RELATED WORKS

Researchers have proposed various methods for transforming tabular network data into images in the field of intrusion detection systems. They applied this transformation to enhance their models and leverage the capabilities of Convolutional Neural Networks in image processing. Mamdouh et al. (2025) introduced a new method called Tab2Visual to transform tabular datasets into visual representations across ten datasets of different domains, including medical, e-commerce, and engineering. Their main idea is to encode each feature as a bar of varying width to enable CNNs and Vision Transformers (ViTs) to process tabular data as images. They normalized tabular data and represented each instance as a row of multiple bars, with each corresponding to a specific value. They assigned each feature a unique colour to enhance visual differentiation, while bar widths are proportional to feature magnitudes. Their experiments showed the effectiveness of Tab2Visual in handling data with limited features and in offering tailored augmentation strategies. While the study introduces a new conversion technique, it does not specifically address intrusion detection in Internet of Things environments.

Similarly, Alenizy and Berri (2025) introduced the Novel Algorithm for Convoluting Tabular Data (NCTD), which transforms tabular data into images to leverage Convolutional Neural Networks for classification. The NCTD method involved normalizing the data, converting each row into a 1D grayscale pixel array, and then applying a series of mathematical transformations (including duplication and progressive rotation) to expand this array into an $N \times N$ matrix. This matrix was then positioned in all four quadrants (top-left, top-right, bottom-left, and bottom-right) of a larger $2N \times 2N$ image to enhance its representational capacity. Although their experiments demonstrated the superior performance of NCTD on ten datasets in healthcare, education, and socioeconomic analysis using a CNN architecture, its applicability remains unexamined in the context of intrusion detection systems. Furthermore, the method is currently limited to processing grayscale image representations.

Another conversion method was proposed by Gavrylenko and Poltoratskyi (2024) to address the challenge of applying DL to tabular data for NIDS. After the pre-processing step and selecting four classes from ten of the UNSW-NB15 dataset, they transformed each row into a two-Dimensional line

plot, where the x-axis represents the various features and the y-axis represents their corresponding normalized values. These plots were converted into grayscale images, stored in PNG format, and used to train a CNN classification model. The experimental results showed classification quality improvement, with accuracy increased by 3.68% compared to the Image Generator for Tabular Data (IGTD) method, and other metrics such as F1-score, Precision, and Recall improved by up to 3.71%, 3.12%, and 3.93%, respectively. Moreover, the recognition time decreased by 0.17 seconds, while the training time increased by 10.9 seconds. Nevertheless, the study used only four of the ten classes from UNSW-NB15, limiting its applicability to broader attack types. However, this study enhances the quality of detecting intrusions within computer networks.

Nazarri et al. (2023) adopted the IGTD algorithm to detect intrusions in their work, they applied an improvised version of the IGTD algorithm to generate a notably clear image for IDS, using two ranking correlation functions to rank distance dissimilarities between features: the Jaccard index and the Euclidean distance, on the CICIDS2017 and CIRA-CIC-DoHBrw-2020 datasets. Their experimental results showed that the generated images had a notable difference between attack and benign traffic. Furthermore, the highest-performing model they trained achieved an accuracy rate of almost 80%. Nevertheless, because their method is designed for binary classification, it cannot identify the specific category of an attack. Similarly, to detect attacks in smart grids, Khoei et al. (2022) adopted the DeepInsight method to generate $n \times n$ single-channel images. First, they selected 39 of the 42 features from the UNSW-NB15 dataset and then used the DeepInsight method to obtain images. After that, a ResNet-50 model was used for both binary classification and multiclass classification. Despite leveraging Transfer Learning (TL) and adapting a method from another domain, the approach does not consider resource-constrained IoT environments and is evaluated on an outdated dataset.

To facilitate the preservation of sequence information, reduce the dimensionality, and enhance the local features, Yan et al. (2024) adopted the Markov Transition Field (MTF) method to transform one-dimensional network traffic data into two-dimensional images. This approach was applied to the CIC-IDS2018 and N-BaIoT datasets. Following this transformation, the resulting images were filtered using the Unsharp Masking technique to obtain high-quality images with sharpened edge details. Additionally, to improve the accuracy of data classification and detection, they integrated three DL models, MobileNet, VGGNet and ResNet into a soft-voting ensemble model to obtain an effective IoT intrusion detection architecture and to capture the temporal dependency and visualize recurrences in the sequence of arbitration IDs of the CAN packet. While the study specifically targets IoT environments, the evaluation is limited to classification problems with only three or four classes.

Desta et al. (2022) proposed an IDS method for In-Vehicle Networks Intrusion Detection named the Rec-CNN method. This method used a Recurrence Plot (RP), beginning by collecting 128 arbitration IDs of packets arriving on the CAN bus, encoding them into a number between zero and the total number of unique arbitration IDs, and then converting these encoded values into images using RPs with a dimension of 128×128 . They evaluated their method using two datasets; the first was collected from their test vehicle, and the second was published alongside the Inception-ResNet-based method. Although the work demonstrates practical applicability, the classification scenarios involve no more than five attack classes.

Naaj et al. (2024) developed an IDS for smart grids by converting 1D network data from the CSE-CIC-IDS2018 dataset into 2D images using a Gramian Angular Field (GAF). They then applied TL. Their model achieved 92% accuracy in multiclass classification and significantly outperformed a baseline CNN in training speed. The proposed approach was extensively evaluated on the NSL-KDD dataset.

Likewise, the work of Chen et al. (2024) used the t-distributed Stochastic Neighbour Embedding (t-SNE) dimensionality reduction algorithm for conversion. First, t-SNE was applied to the entire dataset to map all features onto a 2D plane based on their cosine similarity, creating a global feature location map, where each feature was assigned fixed coordinates. For every single sample, its feature values were then mapped as pixel intensities onto these predetermined coordinates, generating a unique 64 x 64 x 1 grayscale image. If multiple features were mapped to the same location, pixel value averaging was applied. They conducted experiments on the NSL-KDD and CICIOT2023 datasets. The VGGIncepNet is used to classify the generated images. However, these studies use grayscale images, which may reduce the information richness of the resulting images.

The CICIOT2023 dataset has emerged as a prominent benchmark for evaluating IDS models. Numerous studies have leveraged this dataset to evaluate the performance of various methodologies, from ensemble methods like the AdaBoost algorithm implemented by Fahrezi et al. (2024) to various DL architectures. The latter category includes Recurrent Neural Networks (RNNs) employed by Abbas et al. (2024), Bidirectional LSTMs (DL-BiLSTM) by Wang et al. (2023), standard LSTMs by Hizal et al. (2024) and Becerra-Suarez et al. (2024). Multi-Layer Perceptron (MLP) by Gheni and Al-Yaseen (2024). A key characteristic of these works is that they operate directly on the tabular data, without any conversion to images. Despite this straightforward approach, these models have demonstrated notable performance, achieving accuracy rates ranging from 76% to 97%. A common characteristic observed in the related works on tabular-to-image conversion is the transformation of every sample into a distinct image. This process, however, introduces significant computational overhead, adversely affecting the training speed and inference efficiency of the models. Table 1 provides a summary of the approaches from the related work that were applied to convert tabular data into images for classification tasks.

Table 1

Summary of Image Generation Techniques Used in Related Works

Reference	IoT	IDS	Number of classes	Classifier	Images color	Dataset	Conversion method
(Mamdouh et al., 2025)	×	×	2 6	CNN	RGB	10 datasets from UCI Machine Learning Repository	Bar plot
(Alenizy & Berri, 2025)	×	×	2,3, 6,26	CNN	Grayscale	10 datasets	NCTD
(Gavrylenko & Poltoratskyi, 2024)	×	✓	4	CNN	Grayscale	UNSW-NB15	Line plot
(Nazarra et al., 2023)	×	✓	2	CNN	RGB	CICIDS2017 CIRA-CIC-DoHBrw-2020	IGTD
(Khoei et al., 2022)	×	✓	2 9	ResNet-50	Grayscale	UNSW-NB15	DeepInsight
(Yan et al., 2024)	✓	✓	4 3	MobileNet, VGGNet and ResNet	Grayscale	CICIDS2018 N_BaIoT	MTF
(Desta et al., 2022)	✓	✓	2,4,5	CNN	Black and white	CAR HACKING Collected from a real city bus	Recurrence Plot
(Naaj et al., 2024)	×	✓	7	CNN	Grayscale	CIC-IDS-2018	GAF
(Chen et al., 2024)	✓	✓	8	VGG16 VGGIncepNet	Grayscale	CICIOT2023	t-SNE transformation
Our work	✓	✓	20	Modified VGG16	RGB	CICIOT2023	Scatter plot

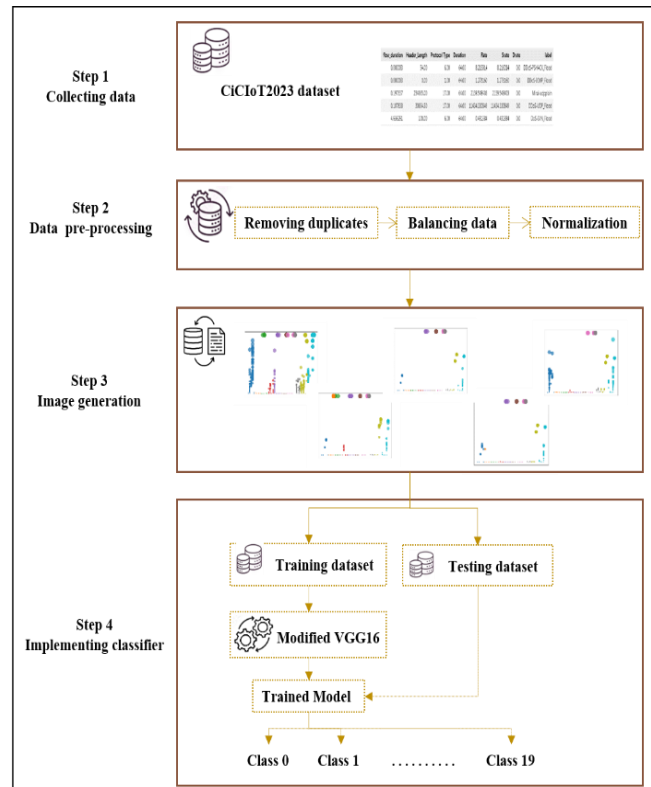
Despite the promising results reported in the literature, several limitations remain in existing techniques for tabular-to-image conversion. A critical analysis of the reviewed studies identifies key research gaps, including the limited consideration of IoT-specific intrusion detection scenarios, the widespread use of grayscale image encoding, and the reliance on outdated or non-IoT datasets. To address these limitations, this work proposes a novel technique that converts tabular network traffic data into color image representations. Furthermore, a modified VGG16 model is employed as the classifier to detect intrusions in IoT environments. In addition, a recent IoT dataset is used to evaluate the performance of the proposed technique.

THE PROPOSED METHOD

This paper proposes a novel method for converting tabular data into images to facilitate intrusion detection in IoT networks using a CNN. Unlike existing methods that convert each sample into a single image, our approach consolidates chunks or blocks of samples into composite images, and proposes a modified VGG16 model to classify the generated images. This batch-processing strategy improves computational efficiency. To evaluate its effectiveness, we conducted two sets of experiments. First, the proposed method was compared against ML models, including RF and eXtreme Gradient Boosting (XGBoost). These models were selected due to their strong performance on the CICIoT2023 dataset, as demonstrated in the experiments conducted by Vicente et al. (2025). Second, a comparative analysis was conducted against existing state-of-the-art techniques, including both methods that operate directly on tabular data and those that employ tabular-to-image conversion. The workflow of the proposed model, which comprises several sequential steps, is illustrated in Figure 1.

Figure 1

Architecture of Proposed IDS Model



As shown in Figure 1, the proposed methodology for intrusion detection in IoT networks consists of the following sequential steps:

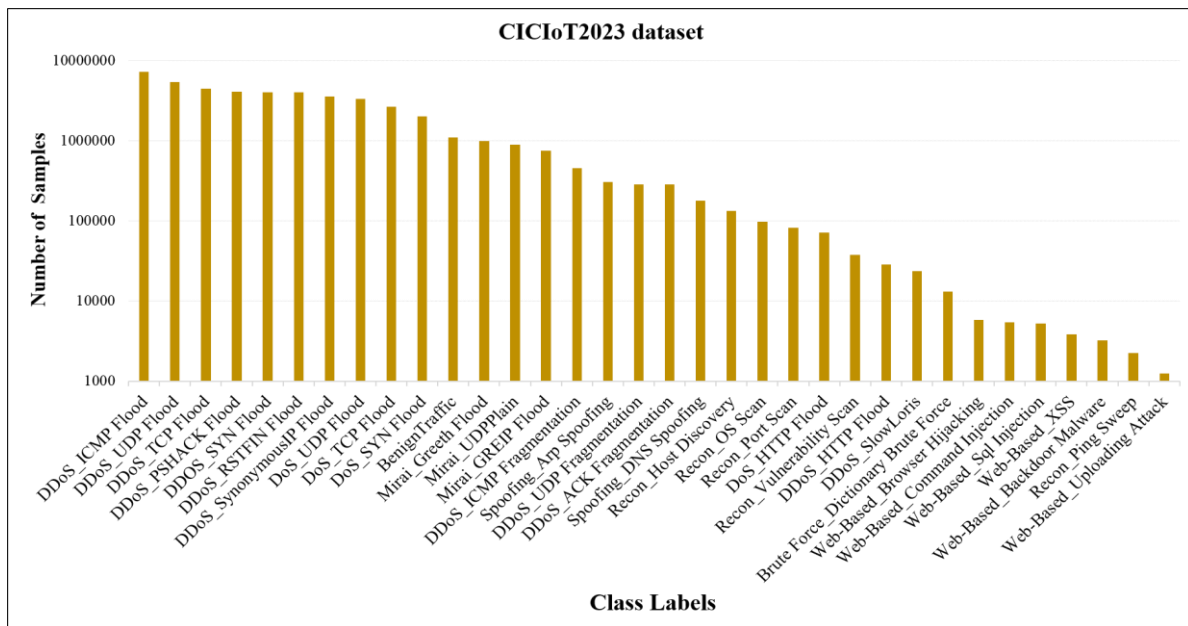
1. Collecting data from IoT devices.
2. Pre-processing the collected data.
3. Transforming the data into images.
4. Implementing the modified VGG16 classifier.

Dataset Collection

This research used a publicly available IoT attack dataset named CICIoT2023, which was proposed by Neto et al. (2023). The CICIoT2023 dataset is distinguished from other datasets, such as CICIDS2017 (Sharafaldin et al., 2018), UNSW-NB15 (Moustafa & Slay, 2015), and NSL-KDD (Tavallaee et al., 2009). It focuses on the IoT environment, unlike other datasets that focus on data traffic in traditional networks, and it uses a variety of devices and protocols. The Canadian Institute for Cybersecurity (CIC) developed this dataset in 2023 using 105 IoT devices. The CICIoT2023 dataset contains 46,686,579 IoT traffic instances, which have 46 features and one target variable that can be categorized into 8 categories, including 33 attack types and one benign type. The distribution of CICIoT 2023 is illustrated in Figure 2. From this, it is observed that the sample distribution across attack classes is highly skewed. The majority class, DDoS_ICMP Flood attack, contains 7,200,504 instances, while the minority class, Uploading attack, contains only 1,252, resulting in a pronounced class imbalance.

Figure 2

Classes Distribution of CICIoT 2023 Dataset



Data Pre-processing

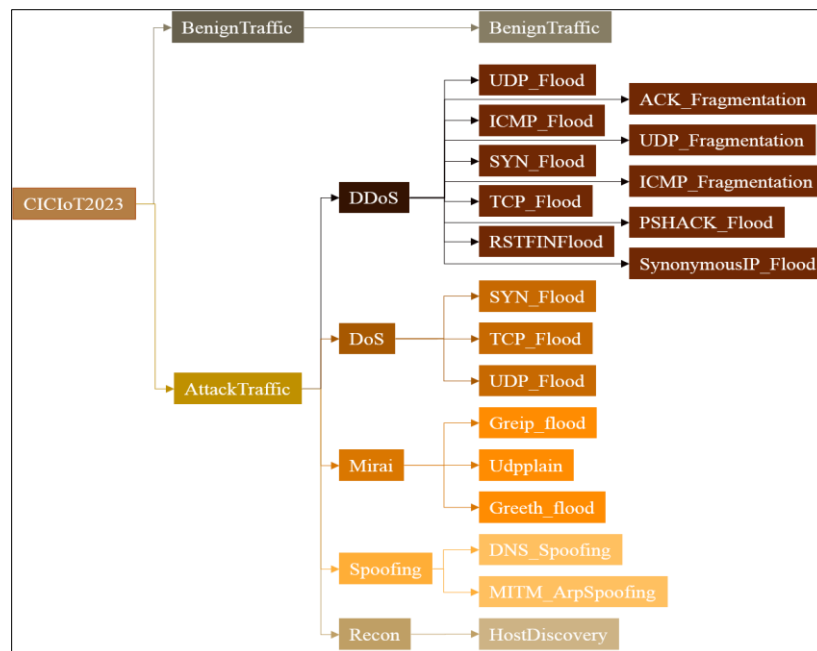
In this study, we applied a pre-processing pipeline for training a DL model on the CICIoT2023 dataset. The dataset initially consists of 169 Comma-Separated-Values (CSV) files and the same number of Packet CAPture (PCAP) files, each containing network traffic data. To ensure the quality of the data

and its adequacy for conversion into images, several crucial steps were taken. A critical data pre-processing step involved removing duplicate samples from each file. This step is essential for DL models, as duplicates can lead to overfitting by causing the model to memorize repeated patterns rather than learning generalizable features (Rakine et al., 2025). As Deep Learning models are highly sensitive to missing data, we verified the absence of null and NaN values. After these processing operations, we obtained a clean dataset.

Imbalanced datasets pose a common challenge in cybersecurity, as highlighted by Al-Qarni et Al-Asmari (2024) in their paper; this may lead to inaccurate model results. The CICIoT2023 dataset also has the limitation of imbalance between the classes in the samples (Salman & Yong, 2025), exhibiting a significant class imbalance as illustrated in Figure 2. Oversampling and under-sampling are common solutions used for imbalance problems. However, oversampling distorts data and slows training. On the other hand, Under-sampling can remove crucial information needed for accurate classification (Man & Sun, 2021). To address this, a sample threshold of 50,000 instances per class was applied, retaining only the classes meeting this criterion. This threshold was selected to ensure sufficient sample representation for each class, while avoiding the inclusion of severely underrepresented classes that could negatively influence model generalization. Consequently, this pre-processing step reduced the number of classes from 34 to 20, including benign traffic; the retained classes are shown in Figure 3. This under-sampling method preserves the original data while guaranteeing that the dataset is suitably representative of each class. The resultant dataset comprises 1,000,000 samples characterized by 47 features, including the target variable.

Figure 3

CICIoT2023 Dataset Classes Selected for Experiment



Variation in feature values can mislead the model and lead to inaccurate results (Alsaedi et al., 2020). The model is biased toward larger feature values. The absence of normalization results in slower convergence and worse model performance. Therefore, data normalization plays a crucial role in preventing larger feature values from outperforming smaller ones by normalizing them within the same

range. We used MinMaxScaler to convert all feature values to a range between 0.0 and 1.0 to prepare the data for deep learning, as shown in Equation 1.

$$X_{\text{new}} = (X - X_{\text{min}}) / (X_{\text{max}} - X_{\text{min}}) \quad (1)$$

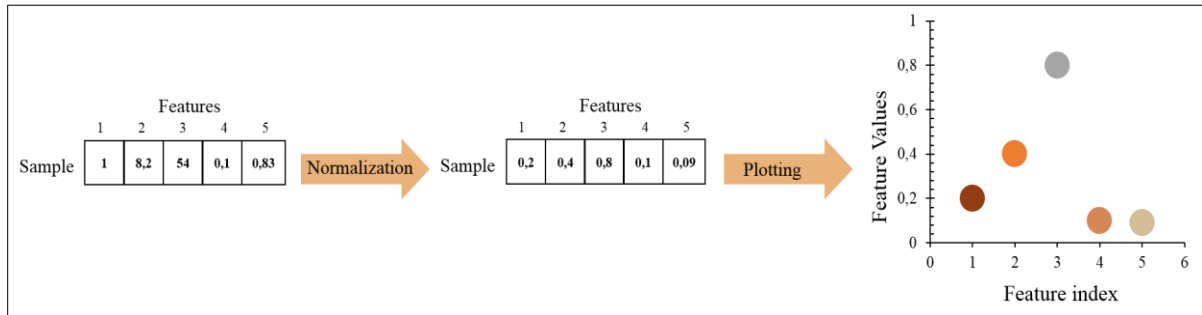
where X is the original value, X_{new} is the normalized value, and X_{max} and X_{min} are the maximum and the minimum values of the feature, respectively. Lastly, as DL models are excellent at automatically extracting relevant features from high-dimensional data, we utilized all 46 features. By retaining the full feature set, we enable the model to capture the complex patterns and relationships within the network traffic data, which is vital for accurate intrusion detection.

Image Generation

To leverage the powerful feature extraction capabilities of DL models designed for image classification and adapt them for intrusion detection in IoT networks, we propose a novel transformation method that generates RGB images from tabular network traffic data. In this step, we convert the normalized data into visual representations suitable for DL models. To transform the tabular data into a format suitable for a CNN model, the data of M samples were first divided into chunks of $N \times N$, in our case N corresponds to 46 features, and the same number, N , of successive samples, with each chunk generating a single image. These chunks were converted into two-dimensional image representations using a scatter plot visualization from the Seaborn library in Python. In these plots, the features become the indices (x-axis values), and their corresponding normalized values become the y-axis values. Figure 4 shows the plot of each sample from a chunk.

Figure 4

Plotting of Each Sample



Each data point is plotted as a marker whose size is determined by its value. Prior to visualization, the feature data were normalized to a $[0.0, 1.0]$ range using a MinMaxScaler. A subsequent scaling transformation, formalized in Equation 2, was applied to each normalized feature value X_{new} (obtained from Equation 1) to generate an appropriate marker size X' for the plot.

$$X' = (X_{\text{new}} + \epsilon) \times \alpha \quad (2)$$

where ϵ is a small constant added to prevent zero-valued features from becoming invisible in the plot (as a marker size of zero is not rendered), and α is a multiplicative scaling factor that projects the values into a range suitable for visible marker sizes. This ensures that even a normalized feature value of 0.0

is represented by a perceptible marker, rather than being omitted. We labeled the resulting RGB scatter plots images with their corresponding class names and saved them in PNG format to preserve lossless image quality. This collection of generated images constitutes the input for the proposed CNN model for classification. Our proposed method is illustrated in Algorithm 1.

Algorithm 1. The Proposed Converting Tabular Data Into Images Algorithm

Input: - 2D array of shape (n_samples, n_features) where n_features = 46

- ϵ and α : constant

Steps:

1. Normalize the data to [0, 1]

2. Apply linear transformation to each value:

for each element d in data:

$$d = (d + \epsilon) \times \alpha$$

3. Determine chunk size:

$$\text{num_chunks} = \text{n_samples} / \text{n_features}$$

4. Group data into chunks:

chunks = empty list

for i from 0 to num_chunks - 1:

start_row = $i \times \text{n_features}$

end_row = start_row + n_features

chunk = submatrix of data from rows [start_row to end_row-1] and all columns

add chunk to chunks

5. For each chunk in chunks:

- Plot the chunk as a 46×46 image

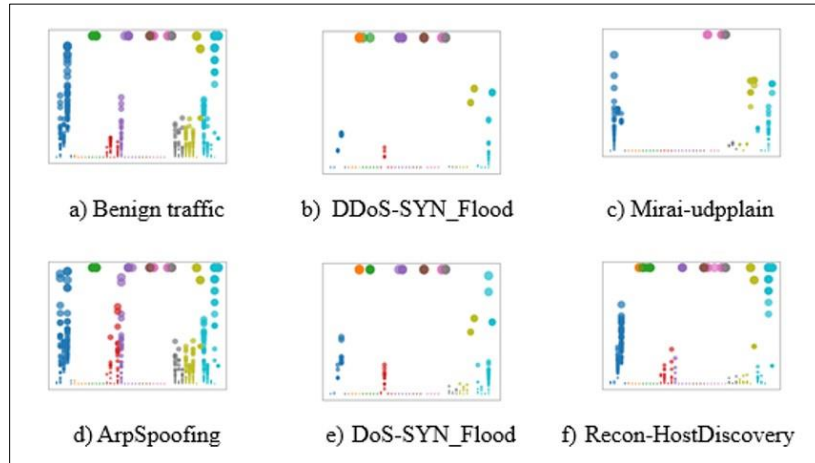
- Save the plot as PNG file

Output:

- PNG images data

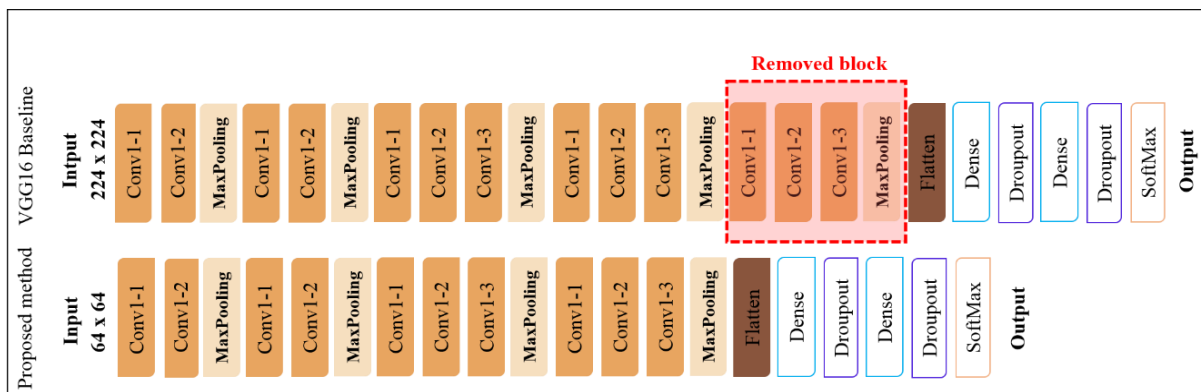
This approach can reduce the number of images, where instead of generating M images from M samples (each image converted into one image), we get M/N images. We standardize the dimensions of these initial 46×46 images to 64×64 pixels, maintaining a 3-channel (RGB) format. The purpose of choosing this image resolution is to balance the preservation of detail with computational costs. Higher resolutions capture more detail but demand more processing power, which conflicts with the resource constraints of the IoT environment. Figure 5 shows examples of generated images from the CIIoT2023 dataset.

For illustration, to generate an image corresponding to benign traffic (Figure 5 (a)), a subset of 46 samples is first selected from the benign traffic data to form a block. Each sample consists of a set of feature values, which are normalized to the range [0.0, 1.0] using the normalization process defined in Equation 1. Subsequently, the normalized data are mapped into a two-dimensional image, where feature indices define the x-axis positions and the corresponding normalized values define the y-axis coordinates. The resulting points from all samples within the block are then plotted using a scatter plot representation. To enhance visual clarity, a scaling transformation is applied to the plotted data using Equation 2. Finally, the generated RGB image is saved in PNG format and labeled as the benign class.

Figure 5*Generated Images from CICIoT2023***Proposed Classifier Model**

In this work, we propose an efficient CNN model for processing images derived from tabular data. Our approach builds upon the original VGG16 architecture introduced by Simonyan and Zisserman (2015) in their seminal work on deep visual recognition. VGG16 is a classic DL feature extraction network that consists of 13 convolutional layers and three fully connected layers. While keeping the core structural components of the baseline VGG16 model, our modified version includes the following key adaptations:

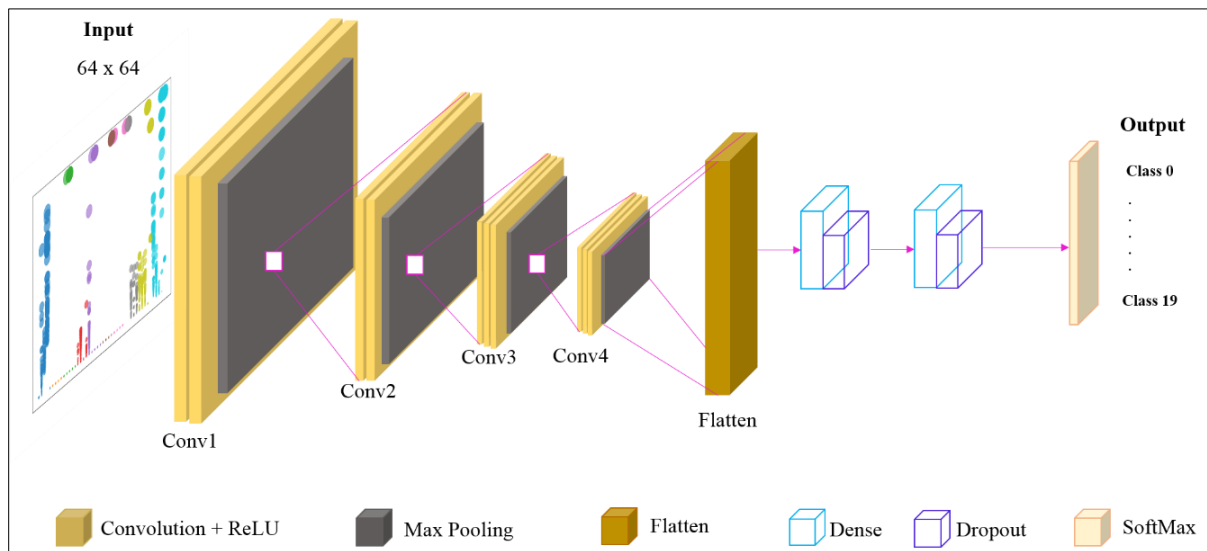
1. Input Transformation: Network traffic data is converted into RGB images of 64 x 64 pixel to serve as input to the model, differing from the standard 224 x 224 input of the original VGG16.
2. Output Layer: The output layer is configured for 20 classes, aligning with the requirements of the intrusion detection dataset.
3. Reduced Depth: The final convolutional block of the original VGG16 is removed, as Figure 6 depicts, to optimize computational efficiency while maintaining performance.

Figure 6*Proposed Modified VGG16 Model*

This study utilizes a modified VGG16 architecture for the multiclass classification of network traffic images into twenty distinct categories, encompassing both attack and normal traffic. Without relying on TL or pre-trained weights, the original architecture was adapted by removing the fifth convolutional block to suit the RGB image dimensions of 64 x 64 pixels and the requisite number of output classes. The final model, depicted in Figure 7, comprises ten convolutional layers, each followed by a Rectified Linear Unit (ReLU) activation function, and three fully connected layers. Convolutional operations employ 3 x 3 filters with a stride of 1 and padding of 1 to extract fine-grained spatial features. Spatial resolution is systematically reduced through max-pooling layers with a 2 x 2 window and a stride of 2, strategically placed after every two to three convolutional layers to preserve critical information while controlling dimensionality.

Figure 7

Architectural Diagram of Proposed Model



The network includes three fully connected layers, a dropout layer (with a hyperparameter of 0.5) for regularization, and a final softmax layer for classification. For training, we implement the Adam optimizer with a learning rate of 0.0001 and use the categorical cross-entropy loss function. Drawing from insights in (Vishwakarma & Kesswani, 2023), where the authors compared different CNN hyperparameters such as epochs and batch size, we adopted their optimal values (50 epochs and a batch size of 128). To mitigate overfitting, accelerate training, and preserve the best model weights, we implemented an EarlyStopping callback with a patience parameter of 5 and `restore_best_weights=True`. Additionally, the learning rate was dynamically adjusted using a ReduceLROnPlateau scheduler. The model was trained on the training dataset and evaluated on 4,360 test images using standard performance metrics.

Performance Evaluation Metrics

The proposed model is trained with the generated images from tabular CICIoT2023 data for a range of iterations. To assess the effectiveness of our model, we have utilised four important metrics namely accuracy, precision, recall, and F1-scores (Briner et al., 2023). These metrics are calculated from a confusion matrix. Table 2 summarizes the definitions of these evaluation metrics. True Positive (TP)

refers to correctly predicted positives, whereas True Negative (TN) denotes correctly predicted negatives. False Positives (FP) are incorrect positive predictions, and False Negatives (FN) are incorrect negative predictions.

Table 2

Performance Metrics

Metric	Definition	Equation
Accuracy	Portion of all normal and attack instances properly classified	$(TP+TN)/(TP+TN+FP+FN) \times 100$
Precision	Proportion of correctly classified positive classifications	$TP/(TP+FP) \times 100$
Recall	Percent of True Positives that a model predicted accurately	$TP/(TP+FN) \times 100$
F1-score	Harmonic mean of recall and precision	$2 \times (\text{Recall} \times \text{Precision}) / (\text{Recall} + \text{Precision}) \times 100$

RESULTS AND DISCUSSION

This section analyses the results based on the aforementioned evaluation metrics. The model's performance is first compared against the ML algorithms from our initial experiments. Subsequently, it is benchmarked against state-of-the-art ML and DL models from the related literature that were applied to the CICIOT2023 dataset. This comparative analysis demonstrates the superiority of the proposed image conversion method combined with the modified VGG16 architecture. We performed experiments on input images obtained by converting the CICIOT2023 tabular data to images for classifying the network traffic as either normal or one of the attack classes using a modified VGG16 model. Training and validation of the model were performed on an Intel(R) Core (TM) i5-4210U CPU @ 1.70GHz 2.40GHz processor and 12GB RAM, and a 64-bit Windows 7 operating system; we used both Visual Studio Code and Google Colab to run the code. Additionally, the data was split into 80% training and 20% testing. Validation used 20% of the training data, drawn from the dataset images. Table 3 shows the number of samples and images used in each set for this work.

Table 3

Summary of Samples and Images used in this Work

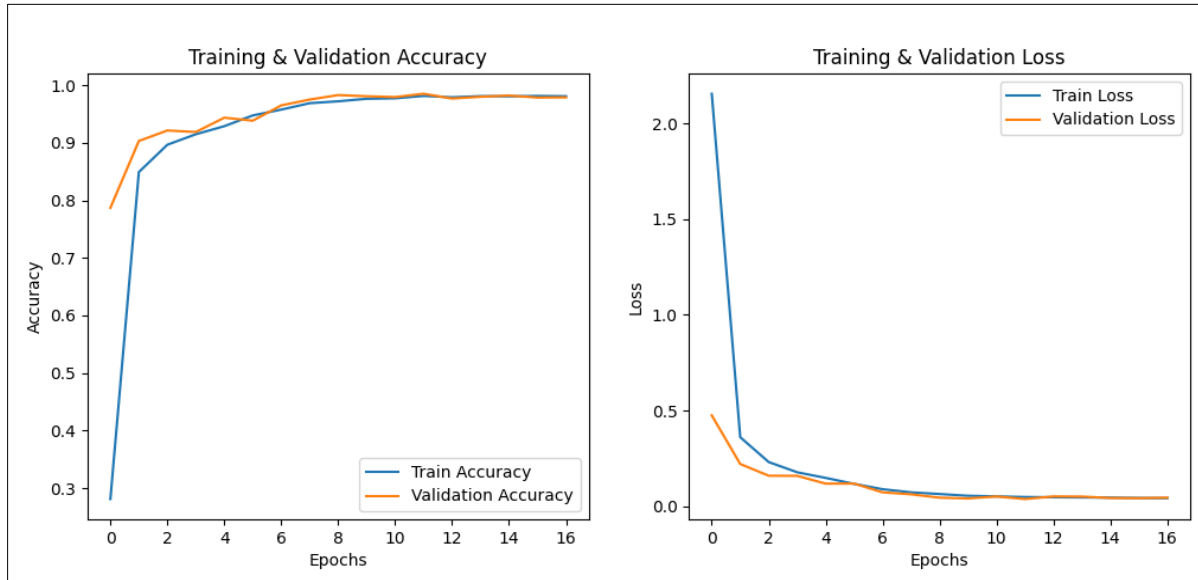
Number of samples	Number of images	Train set	Validation set	Test set
1 000 000	21 740	13 920	3 460	4 360

In our study, accuracy, precision, recall, and F1-score, as given in Table 2, were used to evaluate the proposed model. Figure 8 shows the accuracy and loss curves that were achieved with data images obtained from the generation process. Notably, as the number of epochs increases, the validation and training loss curves gradually decline; however, the validation and training accuracy curves both show an exponential rise. Additionally, the validation loss curve depicts the model's performance on unseen validation data, offering insights into its generalisation capabilities. The lack of significant divergence between the training and validation loss curves indicates effective model training without substantial

overfitting. We observe that the training stopped before reaching 50 epochs due to the use of the Early stopping callback, when the validation loss plateaued (patience=5), with the model reverting to the best weights. The ReduceLROnPlateau scheduler adjusted the learning rate dynamically to optimize convergence. This approach prevented overfitting while ensuring efficient training with stable validation performance.

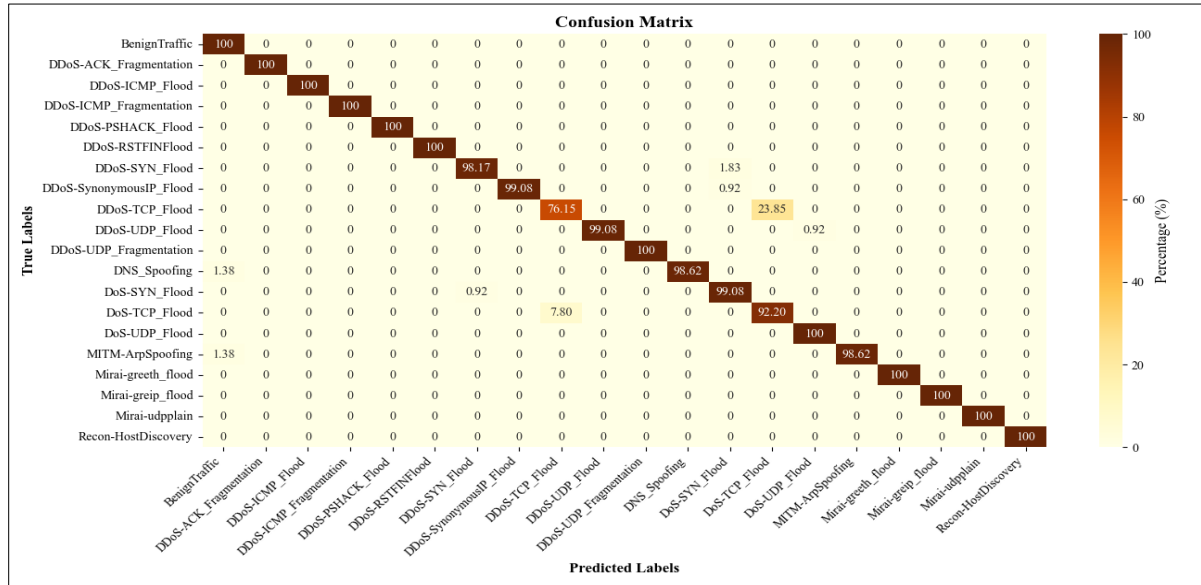
Figure 8

Accuracy and Loss Curves



The proposed model showed strong convergence and excellent performance across all evaluation metrics. The model achieved approximately 98.21% validation accuracy and a 0.044 validation loss within 17 epochs. The model achieved a test accuracy of 98.05%, with a precision of 98.15%, a recall of 98.05%, and an F1-score of 98.04%. During training, as Figure 8 shows, the model demonstrated steady convergence, with training loss decreasing from 2.15 to 0.043, and accuracy improving from 28.10% to almost 98.10%. The confusion matrix (Figure 9) revealed generally strong performance across most classes, with particularly high classification accuracy in categories such as BenignTraffic, DDoS-ICMP-Flood, and Mirai-udpplain, which were predicted with near-perfect precision. This can be attributed to the distinct traffic patterns exhibited by these classes, where their feature distributions differ significantly from other attack types. Besides, the proposed Modified VGG16 model is able to effectively learn the discriminative spatial patterns associated with these classes.

On the other hand, some misclassifications were observed between closely related attack types, particularly between DDoS-TCP_Flood and DoS-TCP_Flood. This behaviour is likely due to the inherent similarity in their underlying network characteristics. Both attack types (DDoS-TCP_Flood and DoS-TCP_Flood) involve high-volume TCP traffic characterised by frequent packet bursts, elevated connection requests, and similar statistical distributions of packet sizes and inter-arrival times. Consequently, when these network flows are converted into image representations for analysis, the resulting images exhibit analogous spatial patterns, textures, and intensity distributions. The visual similarity confuses the model because it can't find clear differences between attack types. As a result, their corresponding image representations exhibit similar spatial patterns, making it more challenging for the model to differentiate between them.

Figure 9*Confusion Matrix of the Proposed Model*

Regarding Table 4, the comparative analysis of our proposed conversion method with the modified VGG16 model and the XGBoost model on tabular data indicates a clear performance difference (Table 3). With the proposed method, we achieve a significantly better classification rate (98.05% accuracy) compared to the XGBoost model (97.98% accuracy). This notable difference of 0.07% underscores the critical importance of our method. Moreover, there is a noteworthy variance of 0.45% between our model and the RF model on tabular data.

Table 4*Compared Model Performances with ML on Tabular Data*

Model	Accuracy %	Precision %	Recall %	F1-score %
XGBoost	97.98	97.88	97.88	97.87
RF	97.60	97.62	97.61	97.60
Proposed CNN	98.05	98.15	98.05	98.04

To evaluate the effectiveness of the proposed methods more comprehensively, we selected six remarkable works on the CICIOT2023 dataset in recent years for comparative experiments. These include the ML method AdaBoost (Fahrezi et al., 2024), and various DL architectures, including Recurrent Neural Networks (RNNs) (Abbas et al., 2024), Bidirectional LSTMs (DL-BiLSTM) (Z. Wang et al., 2023), standard LSTMs (Hizal et al., 2024), and the Multi-Layer Perceptron (MLP) (Gheni & Al-Yaseen, 2024). Although these comparative studies use the same dataset (CICIOT2023), the number of detected categories varies among them, which may affect classification difficulty and performance metrics. However, all methods share the same data features and traffic patterns. Furthermore, the study evaluates the relative effectiveness of different tabular-to-image transformation techniques and DL models. Table 5 provides a comparison of the proposed approach with existing methods for the CICIOT2023 dataset across various classification tasks in both its native tabular form and after image conversion.

Table 5*Performance of the Proposed Model Compared to State-Of-The-Art Studies*

Reference	Data type	Class number	Method	Accuracy %	Precision %	Recall %	F1-score %
(Abbas et al., 2024)	Tabular data	34	RNN	96.52	96.25	96.52	95.73
(Z. Wang et al., 2023)	Tabular data	8	DL-BiLSTM	93.13	91.80	93.13	91.94
(Gheni & Al-Yaseen, 2024)	Tabular data	8	MLP	97.46	97	97	97
(Hizal et al., 2024)	Tabular data	12	LSTM	91.27	91.45	94.30	92.32
(Fahrezi et al., 2024)	Tabular data	8	AdaBoost	76	84	77	77
(Chen et al., 2024)	Image data	8	VGG16	89	91	89	90
(Chen et al., 2024)	Image data	8	VGGIncepNet	92	92	93	92
Proposed	Image data	20	Modified VGG16	98.05	98.15	98.05	98.04

The proposed model consistently outperforms other approaches, achieving 98.05%, 98.15%, 98.05%, and 98.04% in accuracy, precision, recall, and F1-score across 20-class classification tasks. This demonstrates the superior performance and reliability of the proposed image generation method and the modified VGG16 model in IoT attack detection. Compared to DL methods, specifically VGG16 with images generated using the t-SNE method, our modified VGG16 with the proposed images enhances detection accuracy, precision, recall, and F1-score by more than 9.05%, 7.15%, 9.05%, and 8.04%, respectively. Moreover, compared to VGGIncepNet, the accuracy, precision, recall, and F1-score increase by 6.05%, 6.15%, 5.05%, and 6.04%, respectively. On the other hand, Compared to ML methods such as AdaBoost on tabular data, our method achieved a 22.05%, 14.15%, 21.05%, and 21.04% improvement in detection accuracy, precision, recall, and F1-score, respectively, indicating that this traditional method is limited in its feature extraction capability.

CONCLUSION

The rapid spread of Internet of Things devices has been accompanied by an increase in intrusion attacks. Intrusion Detection Systems represent a critical defense mechanism for enhancing cybersecurity. To leverage the powerful feature extraction capabilities of CNNs, which are designed for image data, this study proposes a novel method for converting tabular network traffic data into a colour image format for multiclass classification. The novelty of the proposed approach lies in the use of a scatter-plot-based encoding strategy, which transforms tabular data into structured visual representations while preserving feature distribution patterns and inter-feature relationships in a two-dimensional image, unlike conventional grayscale techniques. Moreover, a modified version of the VGG16 architecture was

introduced to effectively process the generated images while reducing computational complexity, making the model suitable for resource-constrained environments. Experimental evaluation on the CICIoT2023 dataset demonstrates that the proposed framework outperforms several state-of-the-art intrusion detection methods, achieving 98.05% accuracy, 98.15% precision, 98.05% recall, and 98.04% F1-score. These results highlight the effectiveness of combining visual encoding techniques with optimized VGG16 architectures for enhancing intrusion detection performance in an IoT environment.

In the future, further work is needed to improve classifier performance. We propose the following directions for future research: first, to compare our proposed tabular-data-to-image transformation method for IDS with the state-of-the-art tabular data transformation methods, such as DeepInsight and IGTD; second, to investigate the generalizability and performance of our conversion framework when incorporating various convolutional neural network architectures, such as ResNet and VGGNet; finally, to implement and evaluate our model in a real-time Intrusion Detection System environment to assess its practical utility and robustness.

ACKNOWLEDGMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

REFERENCES

- Abbas, S., Bouazzi, I., Ojo, S., Al Hejaili, A., Sampedro, G. A., Almadhor, A., & Gregus, M. (2024). Evaluating deep learning variants for cyber-attacks detection and multi-class classification in IoT networks. *PeerJ Computer Science*, 10, e1793. <https://doi.org/10.7717/peerj-cs.1793>
- Akash, N. S., Rouf, S., Jahan, S., Chowdhury, A., & Uddin, J. (2022). Botnet detection in IoT devices using random forest classifier with independent component analysis. *Journal of Information and Communication Technology*, 21. <https://doi.org/10.32890/jict2022.21.2.3>
- Alenizy, H. A., & Berri, J. (2025). Transforming tabular data into images via enhanced spatial relationships for CNN processing. *Scientific Reports*, 15(1), 17004. <https://doi.org/10.1038/s41598-025-01568-0>
- Al-Na'amneh, Q., Aljaidi, M., Gharaibeh, H., Nasayreh, A., Al Mamlook, R. E., Almatarneh, S., Alzu'bi, D., & Husien, A. S. (2023). Feature selection for robust spoofing detection: A Chi-square-based machine learning approach. *2023 2nd International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI)*, 1–7. <https://doi.org/10.1109/EICEEAI60672.2023.10590243>
- Al-Qarni, E. A., & Al-Asmari, G. A. (2024). Addressing imbalanced data in network intrusion detection: A review and survey. *International Journal of Advanced Computer Science and Applications*, 15(2). <https://doi.org/10.14569/IJACSA.2024.0150215>
- Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., & Anwar, A. (2020). TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *IEEE Access*, 8, 165130–165150. <https://doi.org/10.1109/ACCESS.2020.3022862>
- Al-Turaiki, I., & Altwaijry, N. (2021). A convolutional neural network for improved anomaly-based network intrusion detection. *Big Data*, 9(3), 233–252. <https://doi.org/10.1089/big.2020.0263>

- Becerra-Suarez, F. L., Tuesta-Monteza, V. A., Mejia-Cabrera, H. I., & Arcila-Diaz, J. (2024). Performance evaluation of deep learning models for classifying cybersecurity attacks in IoT networks. *Informatics*, 11(2), 32. <https://doi.org/10.3390/informatics11020032>
- Briner, N., Cullen, D., Halladay, J., Miller, D., Primeau, R., Avila, A., Basnet, R., & Doleck, T. (2023). Tabular-to-image transformations for the classification of anonymous network traffic using deep residual networks. *IEEE Access*, 11, 113100–113113. <https://doi.org/10.1109/ACCESS.2023.3323927>
- Chen, J., Xiao, J., & Xu, J. (2024). VGGIncepNet: Enhancing network intrusion detection and network security through non-image-to-image conversion and deep learning. *Electronics*, 13(18), 3639. <https://doi.org/10.3390/electronics13183639>
- Choi, E., Kim, J. W., Kim, H. J., Kim, J., & Shim, M. (2020). CNN-based network intrusion detection against denial-of-service attacks. *Electronics*. <https://doi.org/10.3390/electronics9060916>
- Desta, A. K., Ohira, S., Arai, I., & Fujikawa, K. (2022). Rec-CNN: In-vehicle networks intrusion detection using convolutional neural networks trained on recurrence plots. *Vehicular Communications*, 35, 100470. <https://doi.org/10.1016/j.vehcom.2022.100470>
- Devendiran, R., & Turukmane, A. V. (2024). Dugat-LSTM: Deep learning based network intrusion detection system using chaotic optimization strategy. *Expert Systems with Applications*, 245, 123027. <https://doi.org/10.1016/j.eswa.2023.123027>
- Fahrezi, S. F., Nugraha, A., Luthfiarta, A., & Primadya, N. D. (2024). Optimizing performance of AdaBoost algorithm through undersampling and hyperparameter tuning on CICIoT 2023 dataset. *Techné: Jurnal Ilmiah Elektroteknika*, 23(2). <https://doi.org/10.31358/techné.v23i2.467>
- Gavrylenko, S., & Poltoratskyi, V. (2024). Network intrusion detection model based on convolutional neural networks and tabular data converted into images. *Системи Управління, Навігації Та Зв'язку. Збірник Наукових Праць*, 4(78), 52–55. <https://doi.org/10.26906/SUNZ.2024.4.052>
- Gheni, H. Q., & Al-Yaseen, W. L. (2024). Two-step data clustering for improved intrusion detection system using CICIoT2023 dataset. *E-Prime - Advances in Electrical Engineering, Electronics and Energy*, 9, 100673. <https://doi.org/10.1016/j.prime.2024.100673>
- Hizal, S., Cavusoglu, U., & Akgun, D. (2024). A novel deep learning-based intrusion detection system for IoT DDoS security. *Internet of Things*, 28, 101336. <https://doi.org/10.1016/j.iot.2024.101336>
- Jiang, J.-P., Liu, S.-Y., Cai, H.-R., Zhou, Q., & Ye, H.-J. (2025). *Representation learning for tabular data: A comprehensive survey* (arXiv:2504.16109). arXiv. <https://doi.org/10.48550/arXiv.2504.16109>
- Khoei, T. T., Hu, W. C., & Kaabouch, N. (2022). Residual convolutional network for detecting attacks on intrusion detection systems in smart grid. *2022 IEEE International Conference on Electro Information Technology (eIT)*, 231–237. <https://doi.org/10.1109/eIT53891.2022.9813983>
- Latif, S., Dola, F. F., Afsar, Md. M., Jahan Esha, I., & Nandi, D. (2022). Investigation of machine learning algorithms for network intrusion detection. *International Journal of Information Engineering and Electronic Business*, 14(2), 1–22. <https://doi.org/10.5815/ijieeb.2022.02.01>
- Liu, J., Xiao, K., Luo, L., Li, Y., & Chen, L. (2020). An intrusion detection system integrating network-level intrusion detection and host-level intrusion detection. *2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS)*, 122–129. <https://doi.org/10.1109/QRS51102.2020.00028>
- Mamdouh, A., El-Melegy, M., Ali, S., & Kikinis, R. (2025). *Tab2Visual: Overcoming limited data in tabular data classification using deep learning with visual representations* (arXiv:2502.07181). arXiv. <https://doi.org/10.48550/arXiv.2502.07181>

- Man, J., & Sun, G. (2021). A residual learning-based network intrusion detection system. *Security and Communication Networks*, 2021, 1–9. <https://doi.org/10.1155/2021/5593435>
- Moustafa, N., & Slay, J. (2015). *UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)*. <https://doi.org/10.1109/MilCIS.2015.7348942>
- Naaj, F. A., Himeur, Y., Mansoor, W., & Atalla, S. (2024). Intrusion detection using time-series imaging and transfer learning in smart grid environments. In M. Ben Ahmed, A. A. Boudhir, R. El Meouche, & İ. R. Karaş (Eds.), *Innovations in smart cities applications Volume 7* (Vol. 906, pp. 585–595). Springer Nature. https://doi.org/10.1007/978-3-031-53824-7_52
- Nazarri, M. N. A. A., Yusof, M. H. M., & Almohammed, A. A. (2023). Generating network intrusion image through IGTD algorithm for CNN classification. *2023 3rd International Conference on Computing and Information Technology (ICCIT)*, 172–177. <https://doi.org/10.1109/ICCIT58132.2023.10273902>
- Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., & Ghorbani, A. A. (2023). CICIOT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. *Sensors*, 23(13), 5941. <https://doi.org/10.3390/s23135941>
- Rakine, I., Oukaira, A., El Guemmat, K., Atouf, I., Ouahabi, S., Talea, M., & Bouragba, T. (2025). Comprehensive review of intrusion detection techniques: ML and DL in different networks. *IEEE Access*, 13, 104345–104367. <https://doi.org/10.1109/ACCESS.2025.3579990>
- Rodríguez, E., Valls, P., Otero, B., Costa, J. J., Verdú, J., Pajuelo, M. A., & Canal, R. (2022). Transfer-learning-based intrusion detection framework in IoT networks. *Sensors*, 22(15), 5621. <https://doi.org/10.3390/s22155621>
- Salman, W. A. H., & Yong, C. H. (2025). Overview of the CICIOT2023 dataset for internet of things intrusion detection systems. *Mesopotamian Journal of Big Data*, 5, 50–60. <https://doi.org/10.58496/MJBD/2025/004>
- Sharafaldin, I., Habibi Lashkari, A., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 108–116. <https://doi.org/10.5220/0006639801080116>
- Sharma, A., Vans, E., Shigemizu, D., Borojevich, K. A., & Tsunoda, T. (2019). DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture. *Scientific Reports*, 9(1), 11399. <https://doi.org/10.1038/s41598-019-47765-6>
- Simonyan, K., & Zisserman, A. (2015). *Very deep convolutional networks for large-scale image recognition* (arXiv:1409.1556; Issue arXiv:1409.1556). arXiv. <https://doi.org/10.48550/arXiv.1409.1556>
- Sun, Y., & Wang, Z. (2025). Intrusion detection in IoT and wireless networks using image-based neural network classification. *Applied Soft Computing*, 177, 113236. <https://doi.org/10.1016/j.asoc.2025.113236>
- Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. (2009). A detailed analysis of the KDD CUP 99 data set. *IEEE Symposium. Computational Intelligence for Security and Defense Applications, CISDA*, 2. <https://doi.org/10.1109/CISDA.2009.5356528>
- Vicente, A. C., Rosa, R. L., & Guimarães, F. G. (2025). Comparative analysis of machine learning algorithms for anomaly detection in IoT networks using CICIOT2023 dataset. *INFOCOMP Journal of Computer Science*, 24(2). <https://infocomp.dcc.ufla.br/index.php/infocomp/article/view/5342>
- Vishwakarma, M., & Kesswani, N. (2023). *A transfer learning based intrusion detection system for internet of things*. <https://doi.org/10.21203/rs.3.rs-2930837/v1>

- Wang, S., Xu, W., & Liu, Y. (2023). Res-TranBiLSTM: An intelligent approach for intrusion detection in the internet of things. *Computer Networks*, 235, 109982. <https://doi.org/10.1016/j.comnet.2023.109982>
- Wang, Z., Chen, H., Yang, S., Luo, X., Li, D., & Wang, J. (2023). A lightweight intrusion detection method for IoT based on deep learning and dynamic quantization. *PeerJ Computer Science*, 9, e1569. <https://doi.org/10.7717/peerj-cs.1569>
- Yan, Y., Yang, Y., Fang, S., Gao, M., & Chen, Y. (2024). MUS model: A deep learning-based architecture for IoT intrusion detection. *Computers, Materials & Continua*, 80(1), 875–896. <https://doi.org/10.32604/cmc.2024.051685>
- Zhu, Y., Brettin, T., Xia, F., Partin, A., Shukla, M., Yoo, H., Evrard, Y. A., Doroshov, J. H., & Stevens, R. L. (2021). Converting tabular data into images for deep learning with convolutional neural networks. *Scientific Reports*, 11(1), 11325. <https://doi.org/10.1038/s41598-021-90923-y>