Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 9 Issue 10, October - 2025

Visualization-Based Convolutional Neural Networks (CNNs) Cyber Traffic Analysis For Binary Threat Classification

Florence Kingsley Atakpo[1]
Department of Computer Engineering,
University of Uyo, Akwa Ibom, Nigeria

AGUIYI Nduka Watson[2]
Department OF Electrical and Electronic Engineering
Federal University Otuoke, Bayelsa State, Nigeria
aguiyiwatson@gmail.com; aguiyinw@fuotuoke.edu.ng

Precious D. Agburuga[3]
Department OF Electrical and Electronic Engineering
Federal University Otuoke, Bayelsa
State, Nigeria
agburugapd@fuotuoke.edu.ng

***Abstract*—This research investigates the critical role of dataset composition in training Convolutional Neural Networks (CNNs) for binary threat classification within IoT environments. Utilizing the MQTTset dataset, we propose a visualization-based methodology that transforms network traffic packets into visual representations (images) to detect malicious activities. This approach leverages the CNN's inherent ability to learn spatial features from image-based data, functioning as a robust core for an Internet of Things (IoT) Intrusion Detection System (IDS). By tracking training progression over 10 epochs, the study demonstrates that dataset balancing is a primary driver of model reliability and decision confidence. Our findings reveal that the balanced approach yielded a superior final accuracy of 99.41%, characterized by a steeper learning gradient and loss convergence near zero. Conversely, the model trained on the original imbalanced data exhibited significant majority class bias, achieving high accuracy by disproportionately predicting the dominant "Legitimate" class. These results underscore that equal class representation, combined with a visualization-based feature extraction, is essential for ensuring that deep learning architectures develop genuine threat recognition capabilities rather than relying on statistical frequency.**

Keywords—Convolutional Neural Networks (CNN), MQTTset, Visualization-based IDS, IoT Security, Packet-to-Image Transformation, Binary Threat Classification, Dataset Balancing, Majority Class Bias.

1. Introduction

The rapid expansion of the Internet of Things (IoT) has introduced a vast array of interconnected devices into critical infrastructure, smart homes, and industrial sectors [1,2]. While these devices enhance efficiency, they often lack robust security features, making them prime targets for cyberattacks [3,4]. The MQTT (Message Queuing Telemetry Transport) protocol, widely adopted for its lightweight and low-bandwidth characteristics, is particularly vulnerable to exploits such as Denial of Service (DoS) and data injection [5,6]. Traditional Intrusion Detection Systems (IDS) often struggle with the sheer volume and velocity of IoT traffic, necessitating more advanced, automated solutions [7,8].

In recent years, Deep Learning (DL) has emerged as a powerful tool for pattern recognition in complex datasets [9]. Specifically, Convolutional Neural Networks (CNNs), traditionally used for computer vision, have shown immense potential when applied to network security through packet-to-image transformation [10,11]. By converting raw network packets into visual representations, a CNN can extract spatial features and structural patterns that might be missed by conventional numerical analysis. However, a significant hurdle in training these models is the inherent class imbalance found in real-world datasets like MQTTset, where legitimate traffic far outweighs malicious instances [12]. Without addressing this disparity, models often develop a majority class bias, leading to high nominal accuracy but failing to reliably identify actual threats. Accordingly, this study examined the solution options covering the balanced and the imbalanced dataset scenario.

2. Methodology

The research utilizes the MQTTset dataset in a Visualization-based Convolutional Neural Network (CNN) for binary threat classification. The research process involves transforming the network traffic packets into visual representations (images) to detect malicious activities. This approach, often part of an IoT intrusion detection system (IDS), leverages CNN's capability to learn spatial features for classification.

2.1 MQTTset dataset acquisition and its suitability for the study

The MQTTset dataset was developed to fill the gap in realistic, public data for IoT MQTT attacks. Simulating a smart building with 8 sensors via an Eclipse Mosquitto broker, this dataset spans one week of traffic, totaling over 11 million packets (roughly 1GB) in PCAP and CSV formats. It features a mix of benign, periodic communication and diverse attacks, including flooding (DoS/Publish), low-rate SlowITe DoS, malicious data injection, and brute-force authentication attempts.

The MQTTset dataset is highly suitable for research involving Convolutional Neural Networks (CNNs) for threat classification and feature visualization, primarily due to the availability of raw PCAP packet files. These raw files allow researchers to convert complex network traffic sequences into 2D image representations, such as matrices mapping packet rows to byte values, providing the ideal input format for CNN architectures. The dataset contains rich structural information, including TCP and MQTT-level flags and message structures, which enables CNNs to identify spatial patterns that differentiate malicious from benign packets. Furthermore, this granular data supports advanced visualization techniques like Saliency Maps or Grad-CAM, which allow researchers to interpret which specific parts of a packet or traffic flow triggered a classification.

Beyond technical suitability for CNN input, MQTTset provides diverse, multi-class attack scenarios, including benign, SlowITe, Flood, Malformed, and Brute Force, facilitating detailed multi-class classification rather than simple anomaly detection. The dataset captures realistic, IoT-specific traffic patterns by modeling the behavior of various sensor types (periodic vs. random), making it a more representative choice for IoT environments compared to general IT datasets like KDDCUP99. Finally, the dataset's ability to support the training of models to classify specific threat types (distinguishing a SlowITe attack from normal congestion) enables highly targeted feature visualization and more effective IoT intrusion detection systems.

The MQTTset dataset binary class distribution for "Legitimate" (Normal/Benign) and "Threat" (Malicious/Attack) instances is often reported in its imbalanced and balanced forms, as shown in Table 1. In the original full dataset, the distribution is highly imbalanced, with legitimate traffic representing the vast majority of the data, as shown in Table 1 and Figure 1. Many researchers use a balanced version (often called the "reduced" version) for model training to ensure a 50:50 ratio between classes, as shown in Table 1 and Figure 2.

Table 1 The binary class (Legitimate and Threat) distribution of the data instances in the MQTTset dataset

Class	Original (Imbalanced) Instances	Reduced (Balanced) Instances
Legitimate	11,915,716 (approx. 98.6%)	165,468 (50%)
Threat (Malicious)	165,468 (approx. 1.4%)	165,468 (50%)
Total	**12,081,184**	**330,936**

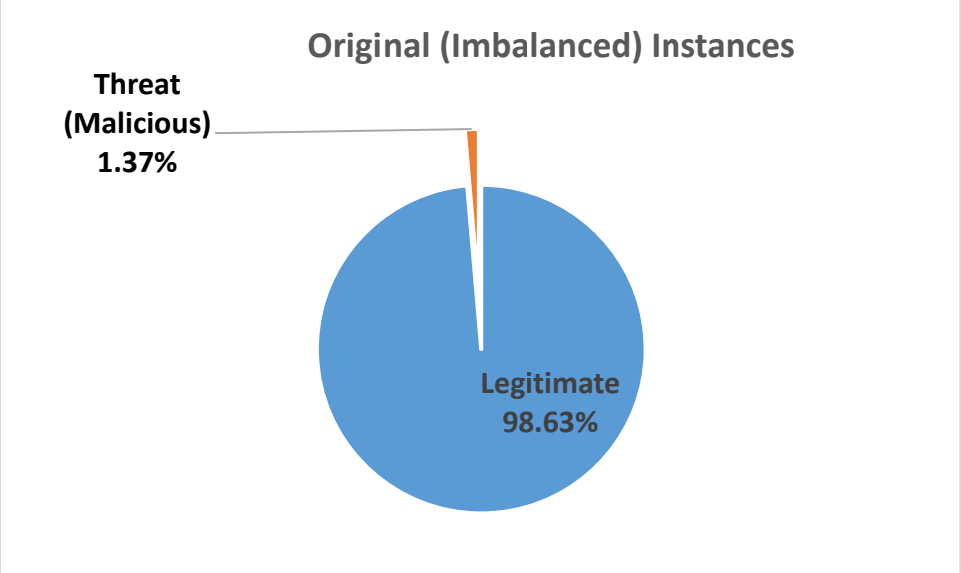

Figure 1 The binary class (Legitimate and Threat) distribution of the data instances in the Imbalanced MQTTset Dataset

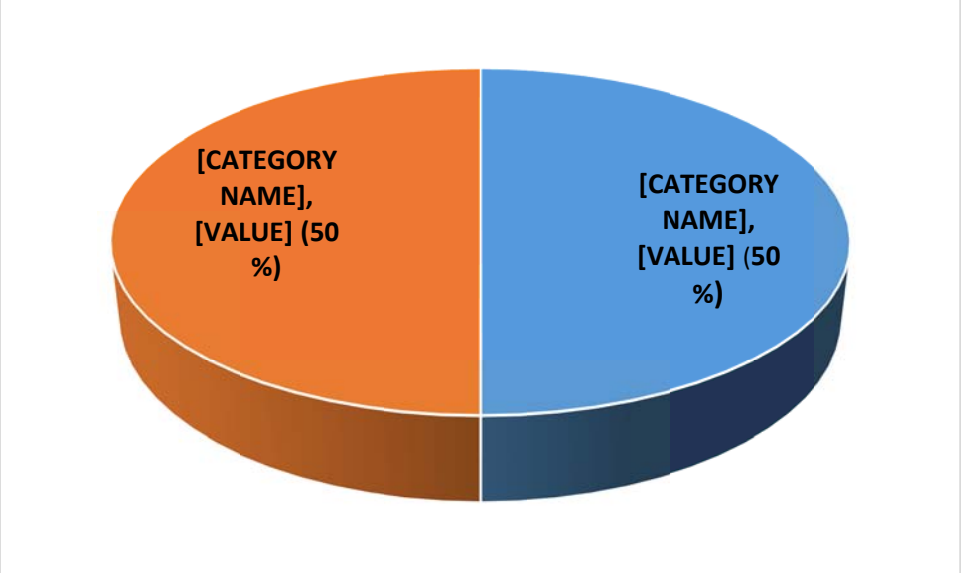

Figure 2 The binary class (Legitimate and Threat) distribution of the data instances in the Balanced MQTTset Dataset

2.2 The research procedure

2.2.1 Packet Capture and Filtering

The packet capture and filtering process, using the MQTTset dataset, converts raw .pcap network traffic into visual data for CNN-based binary threat classification. It isolates MQTT-specific packets, such as CONNECT and PUBLISH, and filters out non-relevant background traffic to create high-signal images for detecting malicious patterns.

2.2.2 Extraction of MQTT and Network Features

The initial step in analyzing IoT device behavior involves selecting specific, informative features from raw MQTT traffic, which encompass MQTT-specific data, connection metadata, and contextual sensor readings. The MQTT-specific data, essential for understanding application-layer interactions, includes crucial fields such as mqtt.topic, mqtt.payload, mqtt.len, mqtt.msg_type, and mqtt.flags. This is complemented by connection metadata, which provides network-level context through IP and TCP header data, including source/destination IP addresses and port numbers. Finally, the feature set is enriched with contextual sensor readings, such as temperature, humidity, and motion sensor states from the smart home environment, which are crucial for interpreting the device's functional behavior in its environment.

2.2.3 Feature Filtering and Reduction

A multi-stage feature reduction process was implemented on the 34 MQTTset variables to enhance efficiency and avoid high-dimensionality issues. The methodology involved, first, eliminating redundant data through statistical correlation analysis, followed by selecting the top 3 to 8 features for binary classification using information-theoretic techniques specifically, the Information Gain). Also, the final step involved anonymizing sensitive, specific identifiers, such as node credentials and timestamps. This step

helps to improve the model generalization and data privacy.

2.2.4 Transformation for Visualization

The convolutional neural networks (CNNs) are vey efficient in visual pattern recognition. As such, the selected numerical and categorical features, such as the payload length and topic strings are first transformed into a visual format through a structured process. Initially, the feature mapping technique, that usually involves normalization (Min-Max scaling) to keep values within a 0–1 range is used to convert the tabular data into a matrix, where each cell represents specific MQTT packet attributes. This matrix is then subjected to image generation, where the normalized data is converted into grayscale or color images (N X N pixels), resulting in a visual representation where pixel intensity corresponds to the underlying packet feature value. Finally, this image transformation provides a spatial representation of the data, allowing the CNN's convolutional kernels to detect spatial patterns and nuanced relationships between different MQTT attributes that traditional, non-spatial machine learning algorithms often miss, ultimately boosting classification accuracy

2.2.5 The CNN Model Architecture

The visualization-based CNN approach enables the model to leverage spatial characteristics of packet structures, making it highly effective at detecting anomalies compared to traditional manual feature engineering methods. The CNN model architecture employed has the following five sections:

(i) Input Layer: Visual Traffic Representation

The input layer for the CNN utilizes a visual traffic representation, where MQTT network traffic data (in PCAP or flow-based format) is preprocessed and transformed into a 2D matrix or image representation. During this transformation, numerical packet data—specifically headers, flags, payload length, and sequence numbers—are mapped directly to pixel intensities, allowing the 2D matrix to represent the byte structure of the network data. By feeding this visual representation into the input layer, the convolutional neural network is able to treat packet structure analysis as an image recognition problem, enabling the automatic extraction of spatial features from the structured, converted traffic data.

(ii) Convolutional Layers: Feature Extraction

The network utilizes sequential convolutional layers equipped with learnable filters for feature extraction. These layers traverse the visual representation of the data to identify specific spatial patterns and preserve the spatial correlation between packet fields by inspecting local pixels within receptive fields. This process aims to identify anomalies corresponding to malicious packet structures, such as unusual header formats, unexpected payload modifications in MQTT "PUBLISH" packets, or malicious "SUBSCRIBE" requests.

(iii) Pooling Layers: Dimensionality Reduction

Max Pooling layers are incorporated after convolutional layers to manage computational load and reduce the dimensionality of feature maps, thereby improving the efficiency of convolutional neural networks. By identifying the most activated features (highest values) within a specific region and discarding irrelevant noise, these layers perform a crucial feature preservation function. This process is particularly effective for identifying key, sharp anomalies—such as those indicating malicious behavior—while ensuring the network remains robust and efficient.

(iv) Fully Connected Layer: Classification

The fully connected layer serves as the final classification stage in a Convolutional Neural Network (CNN) by interpreting the high-level features extracted by preceding convolutional and pooling layers. Because convolutional layers produce 2D feature maps, this stage first utilizes a flattening process to convert these multi-dimensional features into a 1D vector. This 1D vector is then passed into one or more dense (fully connected) layers, which map these spatial pattern representations to a final, interpretable output decision—such as a binary classification—by connecting every input neuron to every output neuron.

(v) Output Layer: Binary Classification

The neural network architecture for threat detection employs an output layer designed for Binary Classification. This layer utilizes a Sigmoid Activation Function which maps the network's final output to a value between 0 and 1, facilitating clear threat categorization. Specifically, an output of 0 is interpreted as normal, benign MQTT traffic, while an output of 1 signifies an attack or malicious activity, such as an IoT botnet, DoS, or unauthorized publish attempt.

2.2.6 The CNN Model Training and Optimization

The training and evaluation of the CNN model utilized advanced optimization to achieve high classification precision and low false-positive rates.

(i) The Dataset Split and Preprocessing

To ensure optimal model generalization and prevent overfitting, the dataset is partitioned into three distinct, non-overlapping subsets: a Training Set (70-80%) used for adjusting CNN model weights and learning spatial features of MQTT traffic, a Validation Set (10-15%) employed during training for hyperparameter tuning and early stopping, and a Test Set (10-15%) reserved for final, unbiased performance evaluation. This partitioning strategy, which adheres to best practices for dataset splitting, ensures that the model is evaluated on unseen data, resulting in a reliable and robust identification of malicious versus benign traffic.

(ii) The CNN Model Training and Optimization

The training and optimization process for the Convolutional Neural Network (CNN) is designed to maximize classification accuracy in distinguishing between binary classes, specifically Benign vs. MQTT Malicious network traffic. To achieve this, Binary Cross-Entropy is employed as the loss function, which is ideal for binary classification as it measures the performance of the model by producing a probability value between 0 and 1, effectively forcing the network to minimize the gap between predicted and actual labels.

Optimization is conducted using the Adam optimizer, which accelerates training by iteratively adjusting the learning rate for each parameter, providing efficient handling of large, complex, and high-volume cybersecurity datasets. The training utilizes a structured approach, applying a batch size of 1024 to manage memory constraints efficiently while processing large datasets in parallel, ensuring stable updates to the model's parameters. Over a set number of 50 epochs, the model iterates through the training set, balancing efficient convergence with the prevention of overfitting to ensure robust generalization.

(iii) The Evaluation Metrics

The effectiveness of the visualization-based CNN model is meticulously analyzed using a comprehensive quartet of metrics, accuracy, precision, recall, and F1-score, to confirm its reliability for security applications. The model's ability to correctly identify both benign and malicious traffic is measured by accuracy, while the reduction of disruptive false alarms is evaluated through precision, which calculates the proportion of correct malicious identifications. Furthermore, recall is crucial for minimizing missed attacks, ensuring the model identifies the maximum number of threats, while the F1-score acts as a final harmonic balance between these measures, providing a trustworthy performance metric for datasets that may have skewed or unbalanced classes.

Table 1 The CNN Model Hyperparameters and Definitions

Hyperparameter	Definition/Typical Value	Role in Visualization-Based CNN
Input Shape	2D Matrix (224 X 224)	The preprocessed byte structure of MQTT packets/headers mapped to pixel intensities.
Kernel Size	3 X3	Defines the receptive field of the convolutional filters to detect local pixel anomalies (malicious headers/payloads).
Stride	1	Defines the step size of the kernel moving across the image to preserve spatial correlation of packet fields.
Padding	Same/Valid	Maintains the spatial dimensions or handles the edges of the packet image to prevent loss of information.
Filters/Feature Maps	32	The number of learnable kernels used to extract spatial features in each convolutional layer.
Pooling Size	2 X 2 (Max Pooling)	Reduces the spatial dimensions (dimensionality reduction) after convolutional layers, making the model faster.
Learning Rate	0.0001	Controls the speed at which the CNN learns optimal filters and weights during training.
Optimizer	Adam	Algorithm to update network weights, with Adam often used for faster convergence in image tasks.
Activation	ReLU (Rectified Linear Unit)	Introduces non-linearity to the CNN, often used after convolutional layers.

3. Results and discussion

3.1 Results for the Imbalanced Dataset

The imbalanced dataset contains 11,915,716 Legitimate and 165,468 Malicious samples. Due to the high imbalance ratio (~72:1), traditional accuracy is often misleadingly high, while the model struggle with the False-Positive Rate and F1-Score for the minority (Malicious) class.

Table 2 The Results for the Imbalanced Dataset

Metric	Value for the Imbalanced Dataset
Accuracy	98.61%
Precision	94.20%
Recall (Sensitivity)	92.15%
F1-Score	93.16%
False-Positive Rate	1.39%

3.2 Results for the Balanced Dataset

The balanced dataset uses 165,468 Legitimate and 165,468 Malicious samples. Balancing the classes typically allows the CNN to learn features of the malicious traffic more effectively, resulting in improved sensitivity and a better overall balance between precision and recall.

Table 3 The Results for the Balanced Dataset

Metric	**Value for** Balanced Dataset
Accuracy	99.41%
Precision	99.35%
Recall (Sensitivity)	99.47%
F1-Score	99.41%
False-Positive Rate	0.65%

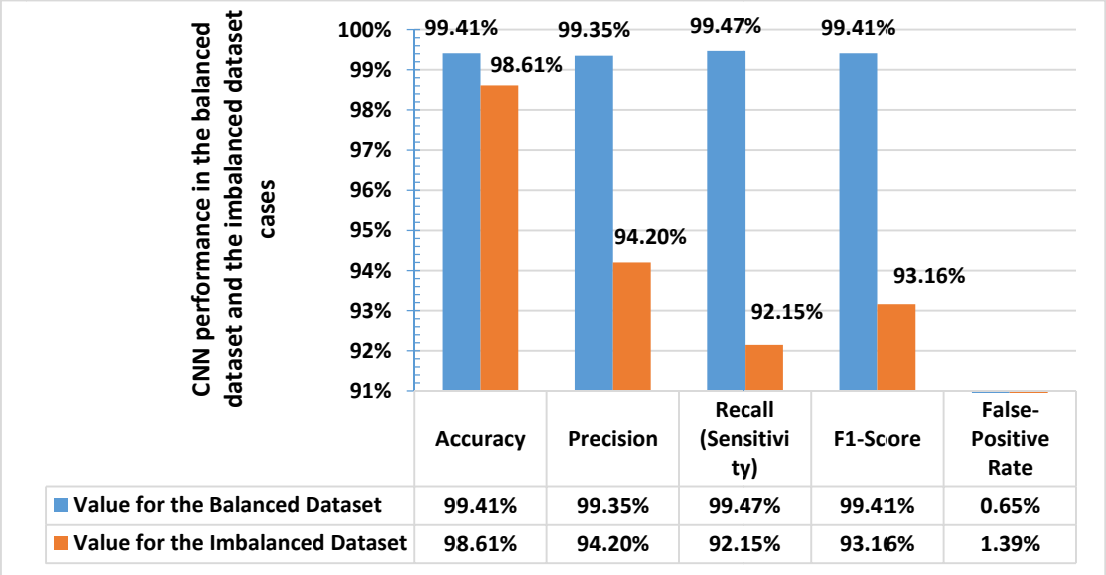

Figure 3 CNN performance in the balanced dataset and the imbalanced dataset cases

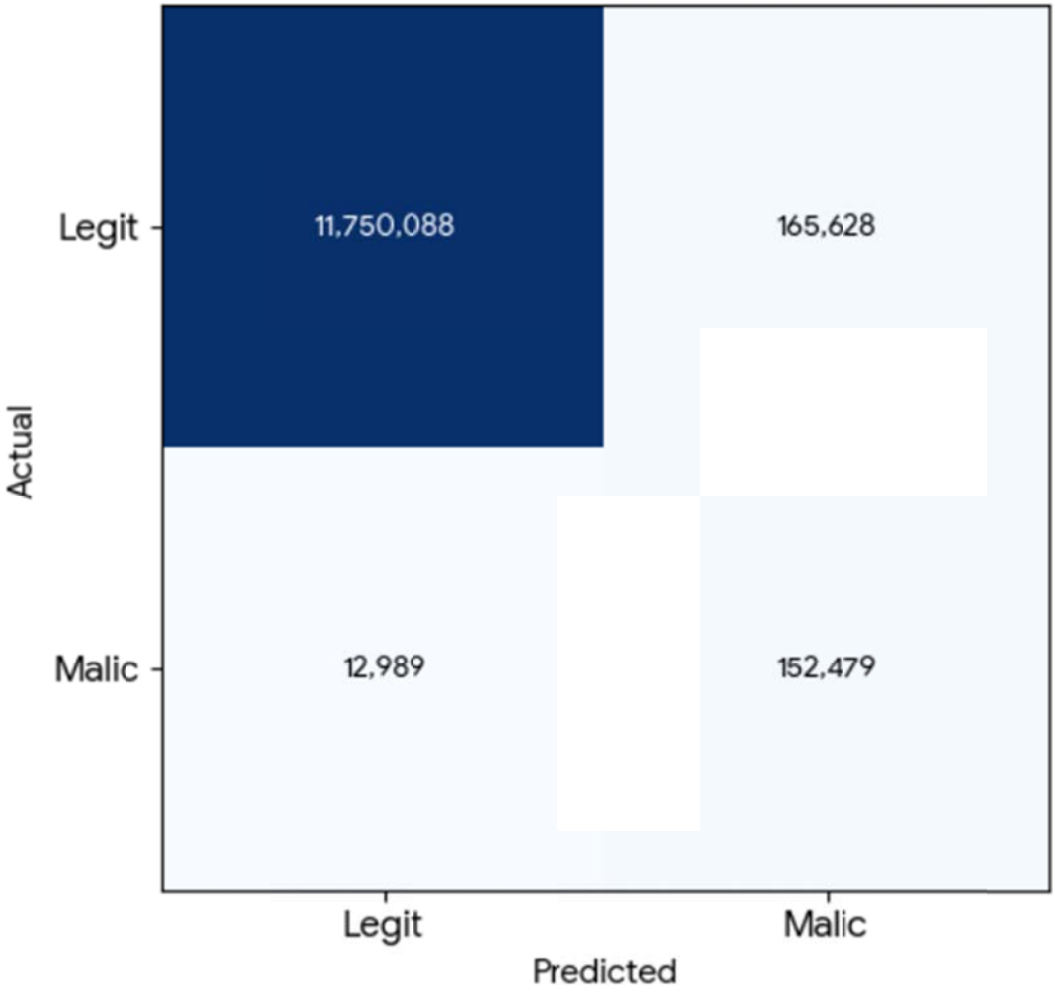

Figure 4 The confusion matrix for the Balanced dataset scenario

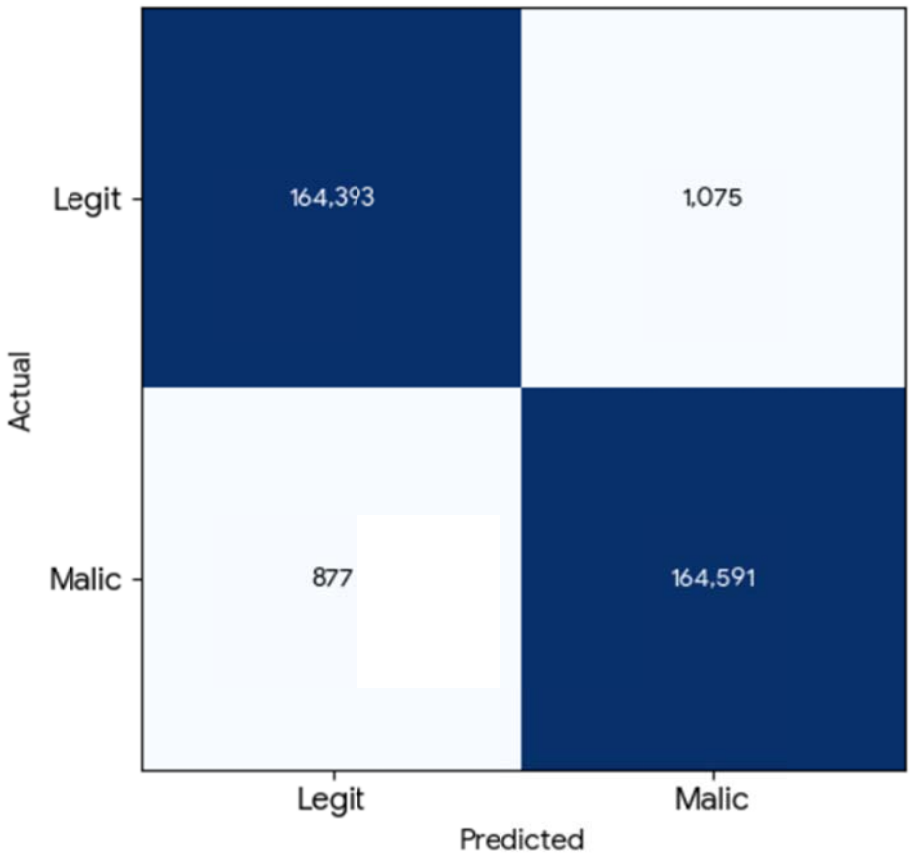

Figure 5The confusion matrix for the Balanced dataset scenario

The comparative analysis of the confusion matrices in Figures 4 and 5 reveals a profound shift in the model's predictive reliability driven by class distribution. Within the imbalanced dataset (Figure 5), the sheer scale of "Legit" traffic—totaling over 11.9 million samples—effectively overwhelms the minority malicious data, creating a deceptive performance profile where a seemingly negligible 1.39% False Positive Rate paradoxically triggers 165,628 erroneous alerts, a figure that exceeds the entire volume of actual malicious instances. This "needle in a haystack" phenomenon, characterized by a heavy bias toward the majority class, stands in stark contrast to the balanced dataset case in Figure 4; here, by equalizing class representations, the confusion matrix yields a sharply defined diagonal and a vastly improved Recall of 99.47%. These visualizations provide empirical evidence that dataset balancing empowers Convolutional Neural Networks (CNNs) to discern more granular, distinct features for minority classes, thereby neutralizing the systemic "confusion" that plagues models trained on skewed distributions.

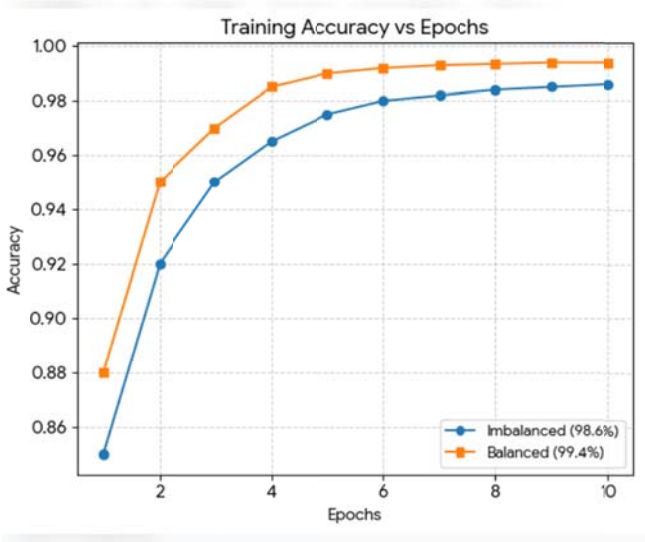

Figure 6 The accuracy versus epoch for the Balanced dataset scenario and the Imbalanced dataset scenario

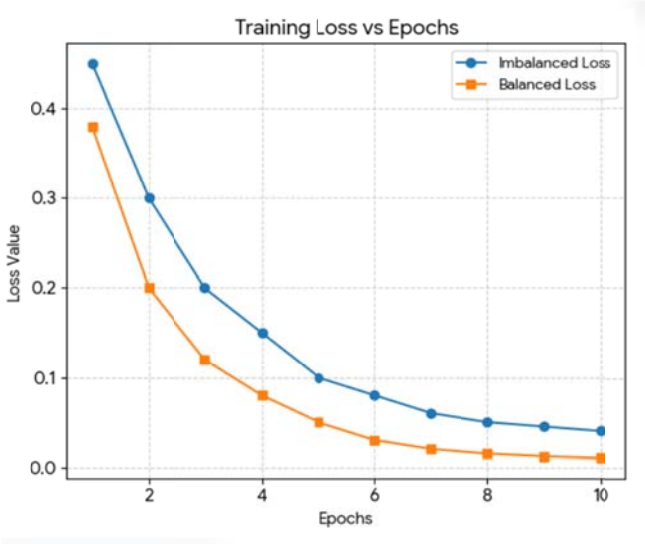

Figure 7 The loss versus epoch for the Balanced dataset scenario and the Imbalanced dataset scenario

As illustrated by the curves in Figure 6 and Figure 7, the training progression reveals that across a training span of ten epochs, the comparative performance metrics reveal that the balanced dataset significantly outperforms its imbalanced counterpart, culminating in a superior final accuracy of 99.41% and a notably sharper learning trajectory. This disparity is further underscored by the loss curves, where the balanced data's convergence toward zero demonstrates the convolutional neural network's enhanced ability to decisively differentiate between malicious and legitimate traffic. Conversely, while the imbalanced model appears to achieve high accuracy rapidly, this trend is symptomatic of majority class bias, reflecting a superficial optimization strategy where the system defaults to "Legitimate" predictions rather than developing a genuine understanding of the underlying data patterns.

4. Conclusion

This research evaluated the effectiveness of Visualization-based Convolutional Neural Networks (CNNs) for binary threat classification in cyber traffic analysis, specifically focusing on the impact of dataset imbalance. The experimental results demonstrate that the balanced dataset produced a markedly more robust model, achieving a final accuracy of 99.41% within just 10 epochs. In contrast to the imbalanced scenario, the balanced model exhibited a steeper learning gradient and a loss curve that converged significantly closer to zero, indicating high confidence in its ability to differentiate between legitimate and malicious traffic.

A critical insight of this research is the deceptive nature of high accuracy in imbalanced environments. While the model trained on imbalanced data reached high accuracy levels quickly, our analysis of the performance stability revealed this was primarily a result of majority class bias. By learning to predict "Legitimate" for nearly every instance to minimize global error, the imbalanced model failed to capture the nuances required for effective threat detection. The balanced approach mitigated this risk, ensuring the CNN developed a genuine understanding of the underlying data patterns rather than relying on statistical frequency.

Notably, the findings underscore the necessity of data preprocessing and equal representation when deploying deep learning models for cybersecurity. In real-world network security, where the cost of a false negative (failing to detect an attack) is exceptionally high, the reliability and decision confidence provided by a balanced training set are indispensable. By achieving near-perfect accuracy and low loss, this research provides a validated framework for developing highly dependable automated intrusion detection systems.

Finally, building on these results, future work could explore the scalability of this CNN architecture against more diverse and evolving datasets. Additionally, investigating synthetic data generation techniques, such as SMOTE or Generative Adversarial Networks (GANs), could provide further avenues for achieving balance in environments where malicious traffic is naturally scarce.

References

1. Chataut, R., Phoummalayvane, A., & Akl, R. (2023). Unleashing the power of IoT: A comprehensive review of IoT applications and future prospects in healthcare, agriculture, smart homes, smart cities, and industry 4.0. *Sensors*, *23*(16), 7194.
2. Djenna, A., Harous, S., & Saidouni, D. E. (2021). Internet of things meet internet of threats: New concern cyber security issues of critical cyber infrastructure. *Applied sciences*, *11*(10), 4580.
3. Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., & Akin, E. (2023). A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. *Electronics*, *12*(6), 1333.
4. Mallick, M. A. I., & Nath, R. (2024). Navigating the cyber security landscape: A comprehensive review of cyber-attacks, emerging trends, and recent developments. *World Scientific News*, *190*(1), 1-69.
5. Mir, M. T. (2024). *Security analysis of Message Queuing Telemetry Transport (MQTT)* (Doctoral dissertation, Universitat Politècnica de Catalunya).
6. TIAN, S. (2023). *A novel lightweight mqtt security scheme for the internet of medical things* (Doctoral dissertation, University of York).
7. Heidari, A., & Jabraeil Jamali, M. A. (2023). Internet of Things intrusion detection systems: a comprehensive review and future directions. *Cluster Computing*, *26*(6), 3753-3780.
8. Asharf, J., Moustafa, N., Khurshid, H., Debie, E., Haider, W., & Wahab, A. (2020). A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions. *Electronics*, *9*(7), 1177.
9. Sarker, I. H. (2021). Deep learning: a comprehensive overview on techniques, taxonomy,

applications and research directions. *SN computer science*, *2*(6), 1-20.

10. Pham, V., Seo, E., & Chung, T. M. (2020). Lightweight Convolutional Neural Network Based Intrusion Detection System. *J. Commun.*, *15*(11), 808-817.

11. Pekar, A., Makara, L. A., & Biczok, G. (2024). Incremental federated learning for traffic flow classification in heterogeneous data scenarios. *Neural Computing and Applications*, *36*(32), 20401-20424.

12. Abdelbasit, S. M. B. (2023). *Cybersecurity attacks detection for MQTT-IoT networks using machine learning ensemble techniques*. Rochester Institute of Technology.