CNN-LSTM Binary Intrusion Classification For Iot Network Applicable To Environmental Monitoring Systems Based MQTT-IOT-IDS2020 Dataset

Uchua Christiana Member[1]
Advanced Space Technology Applications Laboratory Uyo,
National Space Research and Development Agency,
Federal Capital Territory, Abuja, Nigeria

Edemeka, Victor Usiere[2]
Department of Electrical/ Electronic Engineering
Akwa Ibom State Polytechnic, Ikot Osurua
Akwa Ibom State

Antiga Bassey Ekpenyong[3]
Advanced Space Technology Applications Laboratory Uyo,
National Space Research and Development Agency,
Federal Capital Territory, Abuja, Nigeria

Abstract*—The rapid proliferation of Internet of Things (IoT) devices in environmental monitoring systems, often utilizing the lightweight Message Queuing Telemetry Transport (MQTT) protocol, has increased vulnerability to cyber-attacks. This research presents a hybrid deep learning approach for binary IoT network intrusion classification, specifically designed to identify "Normal" or "Attack" traffic. Leveraging the MQTT-IoT-IDS2020 dataset, the methodology incorporates four phases: data preprocessing, feature selection, development of a hybrid CNN-LSTM (Convolutional Neural Network - Long Short-Term Memory) model, and performance evaluation. To address significant dataset imbalance, 1,859,113 attack instances versus 351,684 normal instances, a comparative analysis was conducted between the raw dataset and a balanced dataset created using SMOTE-RUS (Synthetic Minority Over-sampling Technique - Random Under-Sampling), resulting in a balanced 1,105,398 samples per class. Experimental results show that while the imbalanced model achieved high accuracy (0.95), it was heavily biased toward the "Attack" class (85% normal recall). Conversely, the SMOTE-RUS balanced model significantly improved detection reliability for normal traffic, demonstrating the hybrid CNN-LSTM's efficacy in robust, real-world environmental monitoring scenarios.

Keywords—IoT Security, Intrusion Detection System (IDS), MQTT Protocol, CNN-LSTM, Deep Learning, SMOTE-RUS, MQTT-IoT-IDS2020, Environmental Monitoring, Binary Classification.

1. Introduction

The rapid proliferation of the Internet of Things (IoT) has led to the widespread adoption of smart, interconnected devices, particularly within critical infrastructure like environmental monitoring systems [1,2]. These systems heavily rely on the Message Queuing Telemetry Transport (MQTT) protocol, which is favored for its lightweight publish/subscribe model, making it ideal for resource-constrained environments [3,4]. However, this proliferation has exponentially increased the attack surface for cybercriminals. Environmental monitoring systems are susceptible to various threats, including unauthorized access, denial-of-service (DoS) attacks, and MQTT-specific attacks that can manipulate data, leading to improper control measures [5,6].

Traditional security solutions, such as signature-based intrusion detection systems (IDS), are often inadequate for IoT environments because they fail to detect new, sophisticated, or polymorphic attacks [7,8]. Consequently, machine learning and deep learning approaches have emerged as vital, offering the ability to analyze complex network traffic patterns and detect anomalies in real time [9,10].

A significant challenge in developing an effective IoT-IDS is the high dimensionality and temporal dependency of network traffic data, which requires robust feature extraction. Hybrid deep learning models, particularly those combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, have shown promise in addressing these challenges by capturing spatial features (local patterns) and temporal dependencies (sequential traffic flow).

Furthermore, real-world IoT datasets, such as the MQTT-IoT-IDS2020 dataset, are inherently imbalanced, often containing far more "Attack" instances than "Normal" traffic. Training models on such skewed data leads to biased classifiers that excel in detecting the majority class but fail to accurately identify minority instances, resulting in high false positive rates—a critical issue in environmental monitoring where false alarms are costly.

This research addresses these limitations by proposing a hybrid CNN-LSTM binary intrusion classification model specifically tailored for MQTT-based IoT environmental monitoring. To mitigate the bias caused by imbalanced data, this study explores a hybrid approach of Synthetic Minority Over-sampling Technique and Random Under Sampling (SMOTE-RUS), comparing performance with imbalanced data to develop a more reliable, precise, and resource-efficient intrusion detection system. The ultimate goal of this research is to provide a reliable, balanced classification framework that minimizes false alarms and ensures the robust protection of sensitive environmental monitoring infrastructures against evolving cyber threats.

2. Methodology

This methodology describes a hybrid deep learning approach using a Convolutional Neural Network and Long Short-Term Memory (CNN-LSTM) network to classify IoT network traffic as Normal or Attack, specifically tailored for resource-constrained environmental monitoring systems using the MQTT protocol. The research adopts an experimental, quantitative research design. The workflow consists of four main phases: (i) Data Acquisition and preprocessing (ii) Feature Selection (iii) Hybrid Model Development (CNN-LSTM), and (iv) Model Evaluation.

2.1 The MQTT (Message Queuing Telemetry Transport) network architecture and its suitability for environmental monitoring systems

The MQTT (Message Queuing Telemetry Transport) network architecture is a lightweight, publish/subscribe-based messaging protocol designed for machine-to-machine (M2M) communication, particularly in resource-constrained IoT environments. It operates on top of the TCP/IP stack to ensure reliable, ordered, and bidirectional communication. The architecture of MQTT is shown in Figure 1 while the MQTT network topology during regular functioning is shown in Figure 2. The summary of some MQTT-IoT environmental monitoring applications is presented in Table 1.

During regular functioning, the MQTT network topology is a client-server, star-shaped publish/subscribe architecture. It is designed to decouple devices in space and time. The typical MQTT network topology includes the Centralized Broker, Decoupled Clients, Publish/Subscribe Model, One-to-Many Communication and TCP/IP connection and communication protocol.

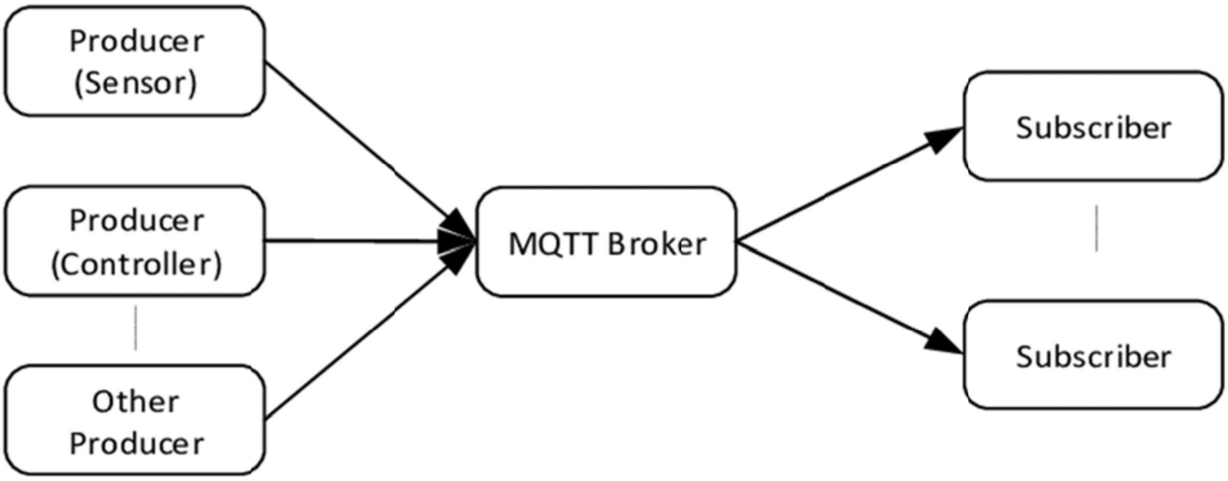

Figure 1 The architecture of the MQTT **[14]**

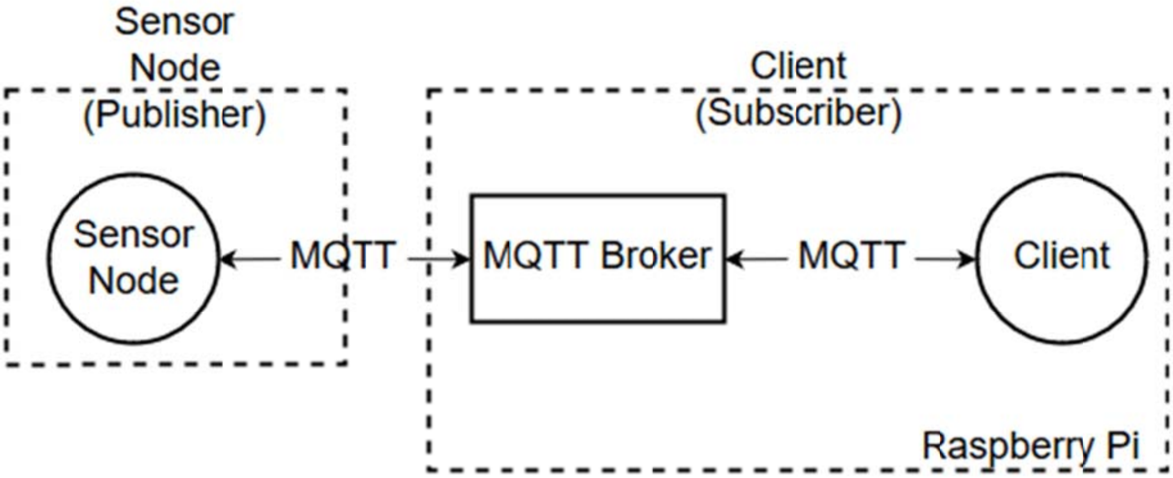

Figure 2 The MQTT network topology during regular functioning **[15]**

Table 1 The MQTT-IoT Environmental Monitoring Applications

Environmental Sector	**Monitored Parameters**	**Typical IoT Sensors/Devices**
Air Quality	Particulate Matter (PM2.5, PM10), CO_2, SO_2, O_3, VOCs	MQ-135, MQ-7, Plantower PMS7003
Weather & Climate	Temperature, Humidity, Atmospheric Pressure, Wind Speed/Direction, Precipitation	DHT11/DHT22, BMP280, Anemometers, Rain Gauges
Water Quality	pH Level, Dissolved Oxygen (DO), Total Dissolved Solids (TDS), Turbidity, Temperature	SEN0189 (Turbidity), pH probes, Conductivity sensors
Smart Agriculture	Soil Moisture, Soil Nutrients, Soil pH, Light Intensity	Capacitive/Resistive Soil Moisture Sensors, LDRs

Urban Environment	Urban Heat Islands, Noise Levels, Waste Level	Temperature sensors, Microphones, Ultrasonic sensors
Disaster Mitigation	Forest Fire Detection, Flood Levels	Flame sensors, Ultrasonic water level sensors

Notably, the centralized MQTT broker (server) is the core of the topology is a central MQTT broker (server). Then all the devices (sensors, actuators, apps) act as MQTT clients and only connect to the broker, never directly to each other. The clients send data ("publish") to the broker on specific topics, and "subscribe" to topics to receive data from other clients. Also, a single publisher can send data to the broker, which then efficiently routes that information to multiple interested subscribers. Finally, each client maintains a persistent TCP/IP connection to the broker, facilitating low-latency communication. Particularly**,** this MQTT network topology or structure ensures that if a publisher is offline, the system continues to work, and the broker manages the message distribution once the publisher reconnects

Most importantly, the MQTT network architecture is highly suitable for deploying numerous, low-cost sensors that must transmit small, intermittent bursts of data reliably to a central dashboard or database. As such, the MQTT (Message Queuing Telemetry Transport) is highly suitable for environmental monitoring systems due to its lightweight design, reliability on unstable networks, and efficient use of resources. These systems often involve numerous low-power sensors scattered over wide areas, making MQTT's publish-subscribe model ideal for collecting data like temperature, humidity, and air quality.

2.2 Dataset Description (MQTT-IoT-IDS2020) Dataset and its Suitability for Environmental Monitoring Systems

The study employed the MQTT-IoT-IDS2020 dataset. The MQTT-IoT-IDS2020 dataset simulates a realistic MQTT-based IoT network, making it ideal for environmental monitoring applications (such as sensors transmitting temperature/humidity). It contains normal traffic and four specific attack scenarios: Aggressive Scan, UDP Scan, Sparta SSH Brute-force, and MQTT Brute-force. The dataset provides CSV files with processed bidirectional and unidirectional flow features.

The MQTT-IoT-IDS2020 dataset is highly suitable for intrusion detection in environmental monitoring systems due to its specific, realistic simulation of MQTT-based IoT networks, which are commonly used for environmental monitoring (e.g., temperature, humidity, and CO2 sensors).

Notably, the MQTT-IoT-IDS2020 dataset is the first to include dedicated MQTT scenarios, simulating a broker, multiple sensors, and a camera, which mirrors the publish-subscribe architecture of typical monitoring networks. Moreover, the dataset contains attacks directly relevant to IoT environments, including MQTT brute-force, Sparta SSH brute-force, and generic network scans (UDP/Aggressive), allowing for the training of classifiers that can identify compromised monitoring sensors. Again, it provides raw .pcap files and three levels of processed features (packet-based, unidirectional, and bidirectional flow), enabling robust, lightweight anomaly detection suitable for low-power, constrained monitoring devices. Furthermore, the dataset provides clear differentiation between normal sensor reading transmissions and malicious attacks, which reduces false alarms in stable environments. The dataset is particularly useful for identifying unauthorized access attempts and data manipulation (such as, false sensor data injection) that could jeopardize the accuracy of environmental monitoring.

2.3 The Dataset Preprocessing and Data Splitting Procedure

To prepare the raw data for the deep learning model, the following steps were implemented:

Step 1. Data Cleaning: Handling missing values, removing irrelevant columns, and cleaning empty fields.

Step 2. Label Encoding: Converting non-numerical categorical features (e.g., IP addresses, protocol types) into numerical representations.

Step 3. Binary Class Labeling: Reclassifying the data into two classes: 'Normal' (0) and 'Attack' (1). The Binary Classification Sample Counts for the MQTT-IoT-IDS2020 Dataset is presented in Table 2.

Step 4. Data Balancing: To address the inherent class imbalance in the MQTT-IoT-IDS2020 dataset, a hybrid sampling approach combining SMOTE (Synthetic Minority Over-sampling Technique) and RUS (Random Under-Sampling) is employed. This strategy ensures the model learns equally from both benign traffic and malicious attacks. The goal of this hybrid method is to reach a 1:1 ratio between classes while mitigating the drawbacks of using either method in isolation.

Step 5. Normalization/Scaling: Applying Min-Max Scaling to normalize numerical features to a range of 0 to 1, ensuring uniform feature contribution.

Step 6. Data Splitting: Dividing the dataset into Training (80%) and Testing (20%) sets to ensure model generalization.

Table 2 The Binary Classification Sample Counts for the MQTT-IoT-IDS2020 Dataset

Class	Binary Label	Description	Sample Count (Packet-Flow)	Sample Count (Uni-Flow)	Sample Count (Bi-Flow)
Normal	0	Standard MQTT sensor operation	351,684	32,042	4,260
Attack	1	Combined malicious scenarios*	1,859,113	41,208	5,622
Total	-	-	2,210,797	73,250	9,882

2.4 The CNN-LSTM Hybrid Model Architecture

In order to address the limitations of traditional, non-temporal machine learning models in capturing complex, time-dependent network patterns, in this research, a hybrid CNN-LSTM architecture is adopted. This model combines the spatial feature extraction capabilities of 1D-Convolutional Neural Networks

(1D-CNN) with the sequential dependency modeling of Long Short-Term Memory (LSTM) networks, creating a powerful tool for IoT intrusion detection. The convolutional neural network (CNN) network structure is shown in Figure 3 while the basic architecture of a typical LSTM model is shown in Figure 4. Again, the CNN-LSTM Hybrid Model Architecture is shown in Figure 5.

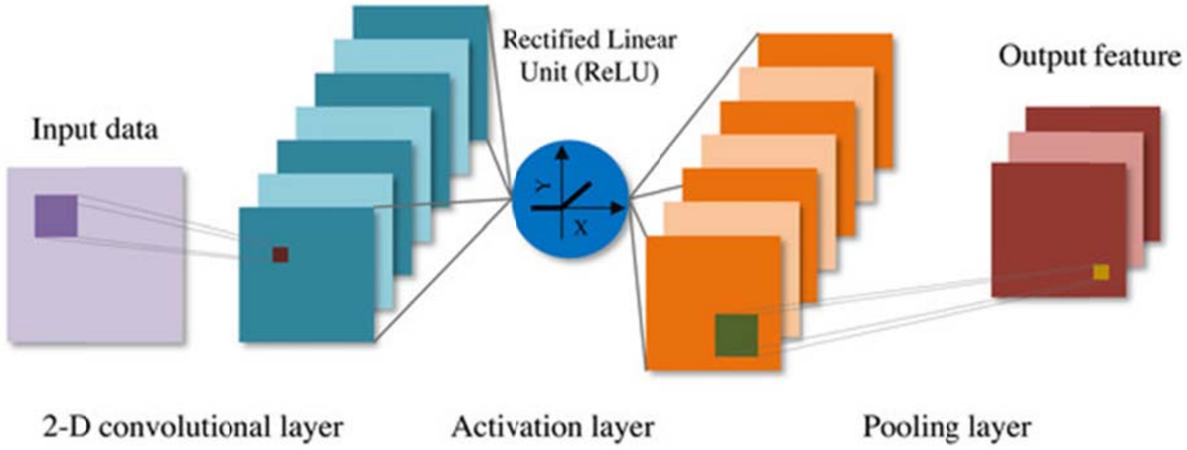

Figure 3 The convolutional neural network network structure of convolutional neural network. [16]

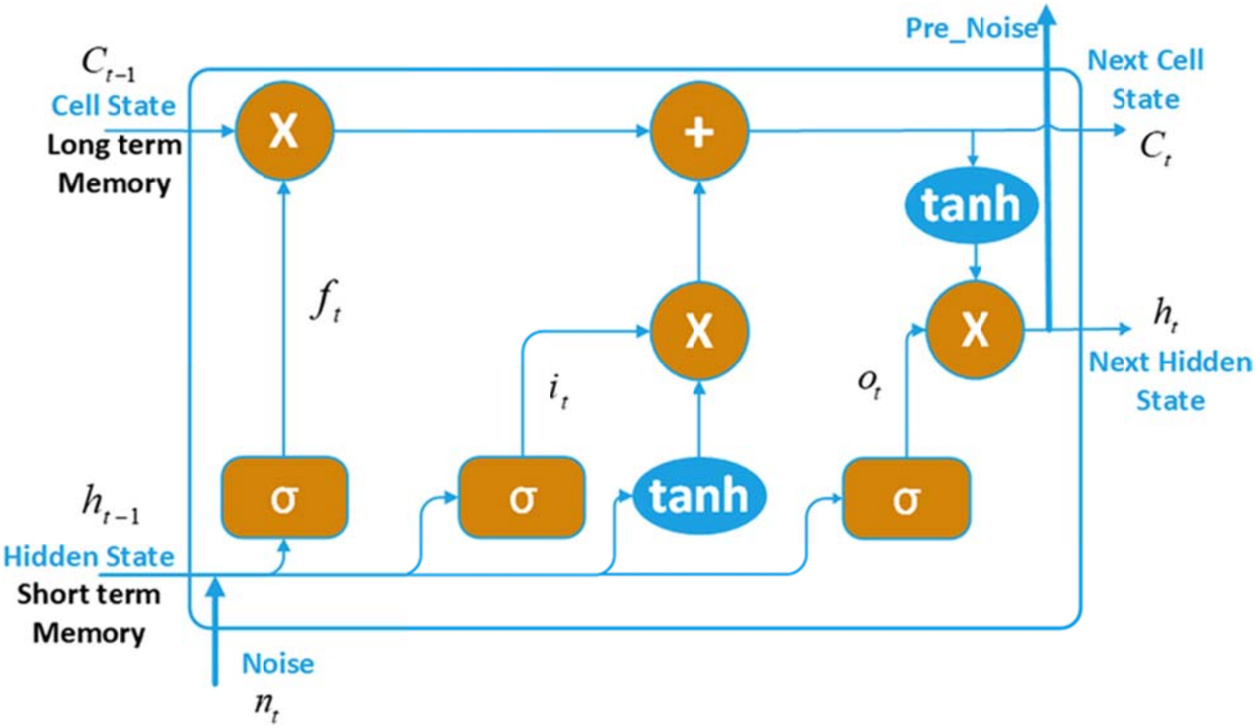

Figure 4 The basic architecture of a typical LSTM model [17]

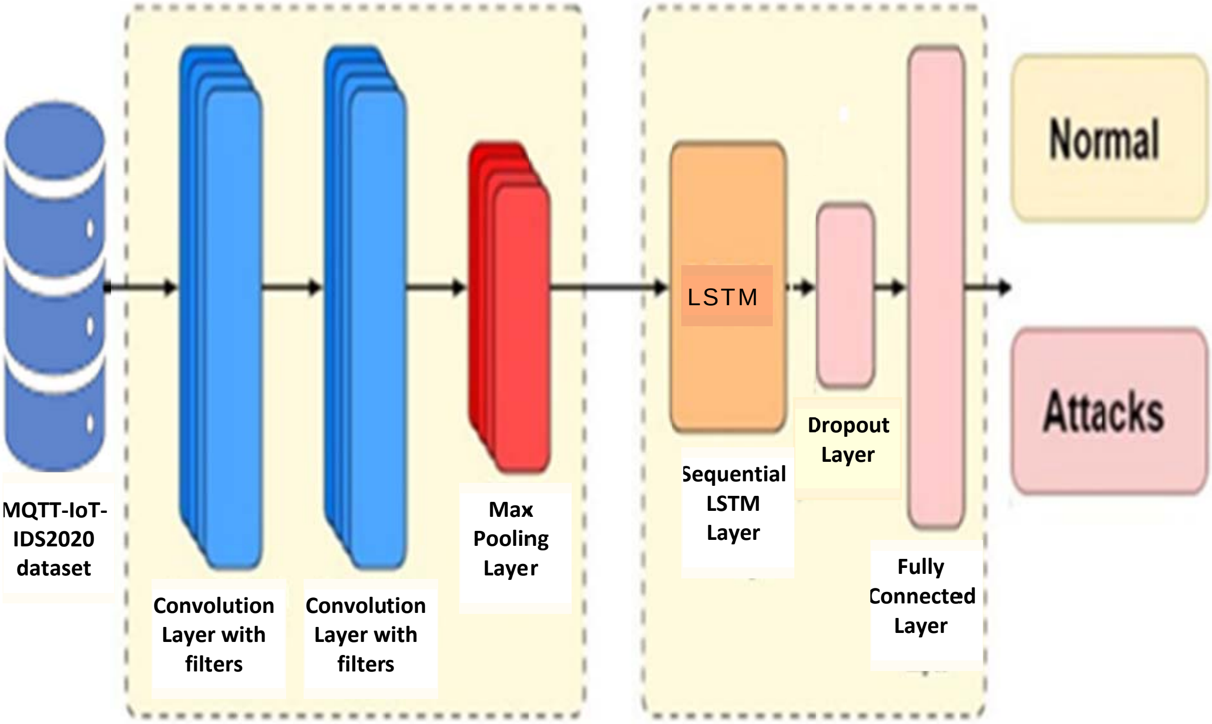

Figure 5 The **CNN-LSTM Hybrid Model Architecture [17]**

The key components of the CNN-LSTM Hybrid Model Architecture are as follows:

i. Input Layer: The input layer accepts preprocessed IoT packet flow features, structured as a 3D tensor (Batch Size, Time Steps, Features), representing the sequential nature of network traffic over time.

ii. 1D-CNN Layers (Spatial Feature Extraction): The initial stages consist of one or more Conv1D layers with rectified linear unit (ReLU) activation functions. These layers identify local, low-level spatial patterns, such as sudden changes in packet length, flag patterns, or specific protocol behavior within a time window.

iii. Max Pooling Layers (Dimensionality Reduction): Following the convolution operations, 1D Max Pooling layers are applied to down-sample the feature maps. This reduces computational complexity, mitigates overfitting by focusing on dominant features, and provides spatial invariance to the detected patterns.

iv. LSTM Layers (Temporal Dependency Modeling): The output of the CNN module is fed into LSTM layers. The LSTM units are specifically designed to analyze long-term, sequential dependencies of the IoT data over time. This is crucial for distinguishing between normal and anomalous behavior that spans multiple packet flows, particularly in detecting subtle, slow-acting environmental sensor anomalies.

v. Dropout Layer (Regularization): A Dropout layer is implemented after the LSTM layer(s) with a rate between 0.2 and 0.5. By randomly deactivating a subset of neurons during training, this layer prevents the model from memorizing training data and improves generalization to unseen attack scenarios.

vi. Dense Layers (Fully Connected): One or more dense (fully connected) layers interpret the combined spatial-temporal features extracted by the hybrid convolutional and recurrent layers.

vii. Output Layer (Binary Classification): The final layer is a dense layer with a single node and a Sigmoid activation function. It produces a probability value between 0 and 1, classifying the input traffic as either Benign (0) or Malicious (1).

2.5 The Model Training and Implementation

The CNN-LSTM Hybrid Model model is trained using the Adam optimizer, chosen for its adaptive learning rate capabilities, which facilitate faster convergence on high-dimensional IoT datasets. Also in the binary classification Binary Cross-Entropy is utilized. The Summary of the CNN-LSTM Hybrid Model parameters are presented in Table 3. The model is developed in Python, leveraging deep learning frameworks such as TensorFlow/Keras or PyTorch for GPU-accelerated training.

Table 3 The Summary of the CNN-LSTM Hybrid Model parameters

Layer Type	**Purpose**	**Configuration**
Input	Accepts Preprocessed Data	(TimeSteps, Features)
Conv1D	Local Spatial Feature Extraction	1-2 Layers (e.g., 64-128 Filters)
MaxPooling1D	Dimensionality Reduction	Pool Size: 2
LSTM	Sequential Dependency Modeling	1-2 Layers (e.g., 64-100 Units)
Dropout	Regularization/Avoid Overfitting	Rate: 0.2 - 0.5
Dense	High-level Interpretation	Fully Connected, ReLU
Output	Binary Classification	Sigmoid (Benign/Malicious)
Optimizer	Model Optimization	Adam
Loss	Error Calculation	Binary Cross-Entropy

2.6 The CNN-LSTM Hybrid Model Evaluation Metrics

The performance of the **CNN-LSTM Hybrid Model** is evaluated using a confusion matrix to determine the four metrics listed in Table 4.

Table 4 The CNN-LSTM Hybrid Model Evaluation Metrics

Metric	**Formula**
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	$\frac{TP}{TP+FP}$
Recall (Detection Rate)	$\frac{TP}{TP+FN}$
F1-Score	$2\times\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$

3. Results and Discussion

The CNN-LSTM Binary intrusion classification is implemented in two different case, first, with the imbalanced dataset and secondly with the SMOTE-RUS balanced dataset. The data samples for the imbalance dataset and the SMOTE-RUS balanced dataset are presented in Table 5. The **performance of the MQTT-IoT-IDS2020 Dataset without data balancing**

Table 5 The data samples for the imbalance dataset and the SMOTE-RUS balanced dataset

Class	Without Data Balancing	With SMOTE-RUS Data Balancing
Normal	**351,684**	**351,684**
Attack	**1,859,113**	**1,859,113**
Macro Avg	**2,210,797**	**2,210,797**

3.1 The Imbalanced Dataset Results

The results of the model performance for the imbalanced dataset is presented in Table 6 and Figure 6. The dataset is heavily imbalanced, with far more "Attack" instances (about 1,859,113) than "Normal" (351,684). The model shows high overall accuracy (0.95), but it is heavily biased toward the majority class ("Attack"). While the "Attack" detection is excellent (Recall 0.99), the "Normal" class has a low recall (0.85), meaning 15% of normal traffic is being falsely flagged as attacks (High False Positives). Essentially, the model works, but it is skewed, making it less reliable for distinguishing normal traffic in a real-world scenario where false alarms are costly.

Table 6 Performance of the binary classification without data balancing

Class	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Support (Samples)**
Normal	**0.98**	**0.85**	**0.91**		**1,105,398***
Attack	**0.94**	**0.99**	**0.96**		**1,105,398***
Macro Avg	**0.96**	**0.92**	**0.94**	**0.95**	**2,210,796**

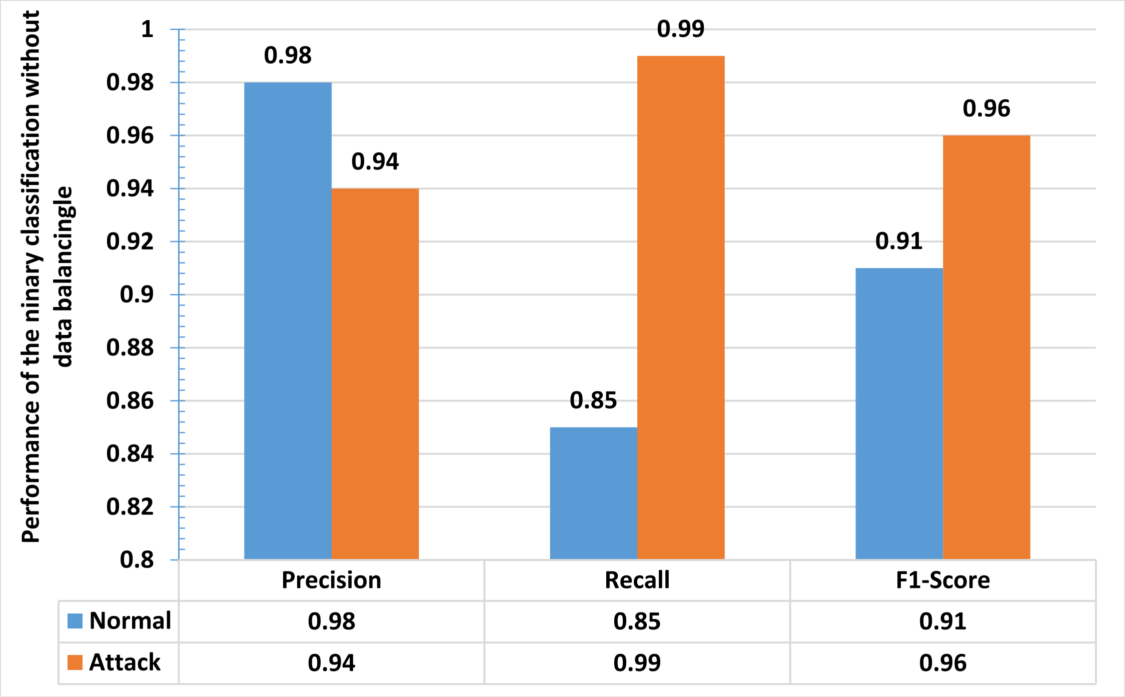

Figure 6 The performance of the binary classification without data balancing

3.2 Balanced Dataset (SMOTE-RUS) Results

The SMOTE-RUS was applied, resulting in an equal distribution of "Normal" and "Attack" samples (1,105,398 each). The results of the model performance after the data baling suing the SMOTE-RUS method are presented in Table 7 and Figure 7. The Macro Avg F1-Score improved from 0.94 to 0.97, and Macro Recall increased from 0.92 to 0.97, indicating a much better balance in detection capabilities. Again, the "Normal" class recall drastically improved from 0.85 to 0.97, meaning fewer false alarms. The "Attack" precision and F1-score remain high, maintaining strong detection of malicious activity. Essentially, the balancing technique successfully eliminated the bias toward the majority class, leading to a robust model that performs well across both classes

In all, using an imbalanced dataset results in high accuracy but poor performance on the minority class (Normal). Applying SMOTE-RUS significantly improves the reliability of the intrusion detection system by balancing the detection rates for both Normal and Attack traffic.

Table 7 Performance of the binary classification model after SMOTE-RUS data balancing

Class	Precision	Recall	F1-Score	Accuracy	Support (Samples)
Normal	0.96	0.97	0.96		1,105,398
Attack	0.97	0.96	0.97		1,105,398
Macro Avg	0.97	0.97	0.97	0.97	2,210,796

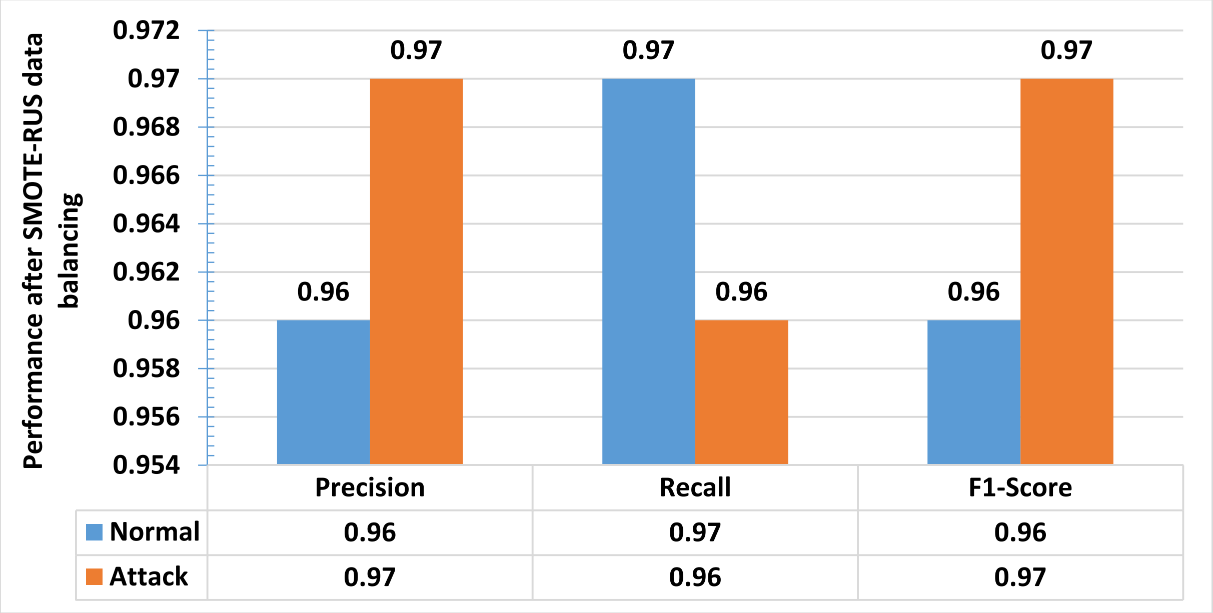

Figure 7 The performance of the binary classification after SMOTE-RUS data balancing

4. Conclusion

This research successfully developed and evaluated a CNN-LSTM hybrid deep learning model for binary intrusion classification within MQTT-based IoT environmental monitoring systems. By leveraging the spatial feature extraction capabilities of CNNs alongside the temporal sequence modeling of LSTMs, the proposed framework provides a robust solution for identifying network threats in resource-constrained environments.

The experimental results highlighted a critical finding regarding dataset distribution. Specifically, the initial implementation on the original imbalance MQTT-IoT-IDS2020 dataset yielded a high accuracy of 0.95, yet suffered from a significant bias toward the majority "Attack" class, resulting in a 15% false positive rate for normal traffic. However, integrating the SMOTE-RUS balancing technique proved essential for real-world applicability. This approach eliminated the majority class bias, significantly improving the recall for "Normal" traffic and ensuring a more reliable classification between benign and malicious activities.

In all, the SMOTE-RUS-enhanced CNN-LSTM model offers a balanced and highly accurate intrusion detection mechanism. Future work will focus on optimizing the model for real-time edge deployment and expanding its capability to handle multi-class attack categorization to further secure IoT ecosystems.

References

1. Goudarzi, A., Ghayoor, F., Waseem, M., Fahad, S., & Traore, I. (2022). A survey on IoT-enabled smart grids: emerging, applications, challenges, and outlook. *Energies*, *15*(19), 6984.
2. Adewuyi, A., Oladele, A. A., Enyiorji, P. U., Ajayi, O. O., Tsambatare, T. E., Oloke, K., & Abijo, I. (2024). The convergence of cybersecurity, Internet of Things (IoT), and data analytics: Safeguarding smart ecosystems. *convergence*, *20*, 21. Adewuyi, A., Oladele, A. A., Enyiorji, P. U., Ajayi, O. O., Tsambatare, T. E., Oloke, K., & Abijo, I. (2024). The convergence of cybersecurity, Internet of Things (IoT), and data analytics: Safeguarding smart ecosystems. *convergence*, *20*, 21.
3. Kaganurmath, S., & Cholli, N. G. (2024, December). Secure Communication in Resource-Constrained IoT Environments using MQTT Protocol: A Review. In *2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS)* (pp. 615-622). IEEE.
4. Aleesha, M., & Laseena, C. A. (2022). Mqtt protocol for resource constrained iot applications: A review. In *proceedings of the International Conference on Systems, Energy and Environment* (p. 2022).
5. Laghari, S. U. A., Li, W., Manickam, S., Nanda, P., Al-Ani, A. K., & Karuppayah, S. (2024). Securing MQTT ecosystem: Exploring vulnerabilities, mitigations, and future trajectories. *IEEE Access*, *12*, 139273-139289.
6. Sharma, A., & Bhushan, K. (2024). A hybrid approach based on PUF and ML to protect MQTT based IoT system from DDoS attacks. *Cluster Computing*, *27*(10), 13809-13834.
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. Agoramoorthy, M., Ali, A., Sujatha, D., TF, M. R., & Ramesh, G. (2023, December). An analysis of signature-based components in hybrid intrusion detection systems. In *2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS)* (pp. 1-5). IEEE.
9. PM, V. P., & Soumya, S. (2024). Advancements in anomaly detection techniques in network traffic: The role of artificial intelligence and machine learning. *Journal of Scientific Research and Technology*, 38-48.
10. Roy, S. (2024). A comprehensive Survey on Network Traffic Anomaly Detection Using Deep Learning. *Preprints*.
11. Li, J., Othman, M. S., Chen, H., & Yusuf, L. M. (2024). Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning. *Journal of Big Data*, *11*(1), 36.
12. Demiss, B. A., & Elsaigh, W. A. (2024). Application of novel hybrid deep learning architectures combining convolutional neural networks (CNN) and recurrent neural networks (RNN): construction duration estimates prediction considering preconstruction uncertainties. *Engineering Research Express*, *6*(3), 032102.
13. Abdelbasit, S. M. B. (2023). *Cybersecurity attacks detection for MQTT-IoT networks using machine learning ensemble techniques*. Rochester Institute of Technology.
14. Wen, Y. A. N. G., & Haider, S. N. (2016, December). Industrial big data platform based on open source software. In *International Conference on Computer Networks and Communication Technology (CNCT 2016)* (pp. 649-658). Atlantis Press.
15. Correia, S. D., & Pinheiro, E. (2018). Software model for a low-cost, IoT oriented energy monitoring platform. International Journal of Computer Science and Engineering (IJCSE).
16. Li, H., Wang, Z., & Li, Z. (2022). An enhanced CNN-LSTM remaining useful life prediction model for aircraft engine with attention mechanism. *PeerJ Computer Science*, *8*, e1084.
17. Zeng, Q., Liang, Y., Chen, G., Duan, H., & Li, C. (2021). Noise prediction of chemical industry park based on multi-station Prophet and multivariate LSTM fitting model. *EURASIP Journal on Advances in Signal Processing*, *2021*(1), 106.