UTM UNIVERSITI TEKNOLOGI MALAYSIA
INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

Intrusion Detection System using Convolutional Neural Network for Industrial Internet of Things Security

Poh Yee Heng[1] & Yusliza Yusoff[2*]
Faculty of Computing,
Universiti Teknologi Malaysia,
81310, UTM Johor Bahru, Johor, Malaysia
Email: janice.hengpy@gmail.com[1]; yusliza@utm.my[2]

Submitted: 11/3/2025. Revised edition: 5/5/2025. Accepted: 6/5/2025. Published online: 27/5/2025
DOI: https://doi.org/10.11113/ijic.v15n1.544

Abstract—**The rise of Industry 4.0 has led to the widespread adoption of Industrial Internet of Things (IIoT) devices, enhancing manufacturing efficiency while introducing significant cybersecurity risks. IIoT environments are highly susceptible to cyber threats such as Denial-of-Service (DoS), SQL injection, and ransomware, which can lead to production downtime and data breaches. Traditional intrusion detection systems (IDS) often fail to detect evolving threats, resulting in high false negative rates. This research proposes an advanced IDS integrating Convolutional Neural Networks (CNN) with Long Short-Term Memory (LSTM) to enhance IIoT security. By leveraging both spatial and temporal feature extraction, the proposed model effectively identifies network anomalies in real-time industrial environments. This study contributes to IIoT cybersecurity by developing an IDS capable of improving threat detection through the integration of CNN and LSTM architectures. The approach enhances pattern recognition and sequential dependency modeling, making it more adaptive to dynamic cyber threats. The model is trained and evaluated on a large-scale IIoT dataset, achieving a binary classification accuracy of 71%, outperforming several state-of-the-art models. The CNN-LSTM IDS demonstrates a strong ability to recognize normal traffic, with a recall of 99%, significantly reducing false alarms. In multi-class classification, the model successfully identifies certain high-volume attack types, such as DDoS. These findings underscore both the strengths and limitations of deep learning-based intrusion detection in IIoT environments. While the proposed model offers significant improvements, further research is needed to address the detection** ***of low-frequency attacks and optimize classification performance.***

Keywords—**Convolutional Neural Network, Long-Term-Short-Memory Network, Intrusion Detection System, Industrial Internet of Things**

I. INTRODUCTION

In the era of Industry 4.0, component manufacturing is gaining popularity among industries due to its numerous benefits. The integration of physical processes with digital connectivity, driven by the rapid expansion of Internet of Things (IoT) devices, this technological advancement has significantly influenced society. Particularly in the manufacturing sector, IoT devices play a crucial role in collecting data from automation machines. The worldwide adoption of IoT devices is projected to reach around 75.44 billion by the year 2025 [1]. The Industrial Internet of Things (IIoT) is specifically designed for the manufacturing industry, enabling the interconnection and intelligence of industrial systems with the help of sensors and actuators [2]. However, as the manufacturing industry rapidly adopts IIoT technology, the increased connectivity has also elevated the risk of cyber-attacks. These malicious activities, known as intrusions, involve monitoring computer systems or networks and analyzing events for signs of security problems [3].

The IIoT devices are increasingly vulnerable to intrusions and malware due to their interconnected nature and often inadequate security measures, presenting significant challenges in effectively identifying and analyzing these threats in real-time to prevent potential disruptions in industrial operations. Traditional IDS may not be sufficient to address the sophisticated and evolving nature of cyber threats targeting IIoT devices, because they can lead to a higher false negative since they cannot identify unknown attacks. Thus, implementing advanced machine learning techniques, for instance CNN [4], offers a promising approach for enhancing the detection capabilities of IDS. To ensure the reliability and effectiveness of a deep learning-based IDS, validating the intrusion classification model through stringent performance metrics is crucial to ensuring its reliability and effectiveness. The challenge lies in accurately assessing the model's performance and ensuring it meets the required standards for performance metrics.

The aim of this research is to apply CNN implemented by Long Short-Term Memory (LSTM) to develop a complete Intrusion Detection System for IIoT security. To achieve the aim, the objectives of this research are: (1) to identify and analyze the intrusions or malware in IIoT, (2) to develop and IDS using a deep learning approach by combining CNN with LSTM, and (3) to validate the intrusion classification model through the performance and verify the intrusion classification model using metrics such as accuracy, F1-score, precision, and recall.

The outline for the research article starts with section one with brief introduction of related works in the intrusions in Industrial Internet of Things, section two with a the related works about existing deep learning techniques in the IDS in IIoTs, section three with introduction of the proposed CNN+LSTM network algorithms, section four with experimental details, and their corresponding results which are given comprehensive discussion on their effectiveness over the proposed CNN+LSTM and other existing deep learning models, finally the section five details the conclusion related to the proposed experiment and methodology.

II. RELATED WORKS

The implementation of CNNs for IDS in the IIoT has garnered significant attention in recent years. CNNs, known for their ability to automatically extract features and handle high-dimensional data, offer a promising solution for the complex and evolving cybersecurity threats facing IIoT environments.

This section reviews related works that explore the application of CNNs in IDS for IIoT, highlighting their methodologies, performance, and the unique challenges addressed, with the summary as shown in Table I. Through this review, the effectiveness and advancements of CNN-based IDS in safeguarding IIoT systems are elucidated, providing a comprehensive understanding of current research trends and future directions.

Despite significant advancements in CNN-based intrusion detection models for IIoT environments, existing approaches exhibit certain limitations that necessitate further improvement. Several studies have attempted to enhance feature extraction and classification performance through various deep learning architectures. For instance, CNN with Vision Transformers (ViT) [5] to leverage spatial attention mechanisms, achieving a binary classification accuracy of 96.3% and a multiclass accuracy of 96.4%. However, this model heavily relied on the flow-to-image conversion technique, which may introduce computational overhead and data transformation biases. Similarly, the Res-CNN-SRU model [6] incorporating residual connections and Simple Recurrent Units (SRU) to mitigate the vanishing gradient problem, achieving 98.79% accuracy. Nevertheless, the SRU's efficiency in capturing long-term dependencies remains suboptimal compared to LSTM-based approaches.

Another notable study about MBConv-ViT [7], a hybrid model that combined deep separable convolutions and multi-head attention for global and local feature extraction, achieving an exceptional accuracy of 99.99%. While this model demonstrated superior classification performance, it struggled with imbalanced datasets, potentially leading to biased detection results favoring majority classes. Additionally, the integration of CNN, LSTM, and attention mechanisms for intrusion detection [8], but their approach yielded an F1-score of 85%, indicating potential challenges in capturing complex attack patterns effectively. Moreover, the CNN-LSTM-GRU [9] for detecting cyber threats in electric vehicle charging stations, achieving 100% binary classification accuracy. While this model effectively captured sequential dependencies, the inclusion of both LSTM and GRU increased computational complexity, which may hinder real-time application feasibility.

Given these gaps, the proposed CNN-LSTM model is chosen as an optimal solution for IIoT intrusion detection due to its ability to effectively capture both spatial and temporal features. CNN is well-suited for learning hierarchical spatial representations from network traffic data, while LSTM excels in modeling long-term dependencies, which is crucial for detecting sequential attack patterns over time. Unlike ViT-based models that require extensive pre-processing, CNN directly processes raw network traffic, reducing computational overhead. Moreover, compared to SRU and GRU, LSTM offers superior retention of past information through its gating mechanism, making it more effective for analyzing time-dependent network anomalies.

By integrating CNN and LSTM, the proposed model overcomes the limitations of prior approaches by efficiently extracting spatial correlations while preserving temporal dependencies in network traffic data. This hybrid architecture enhances the detection of sophisticated cyber threats, improves classification accuracy across diverse attack types, and ensures robust performance even in the presence of imbalanced datasets. Consequently, CNN-LSTM emerges as the most suitable deep learning framework for enhancing IIoT security, addressing the deficiencies of existing methods while maintaining computational efficiency and adaptability to evolving attack patterns.

TABLE I. SUMMARY OF THE PERFORMANCE OF CNN APPROACHES FOR INTRUSION DETECTION SYSTEM IN IIOT

No	Model	Dataset	Result	Advantages	Disadvantages
1	CNN with ViT classifier [5]	CIC IDS2017	• Binary classification: Accuracy 96.3% • Multiclass classification: Testing Accuracy 96.4%	• 8.09% higher accuracy compared to the other algorithms.	• Reduced performance for categories with small data sizes due to lower information gain for distinguishing features.
2	Res-CNN-SRU [6]	Gas pipeline industry dataset proposed by Mississippi State University	• High accuracy of 98.79% • Precision of 95.34% • Recall of 95.04% • F1-score of 95.19%	• Improved recognition rate. • Reduced false alarm rate. • Shortened the training time.	• Not ideal to detect unknown attacks.
3	MobileNet CNN with ViT [7]	TON-IIoT	• Overall accuracy of 99.99%	• Better performance in feature extraction and correlation.	• Biased model performance due to the imbalanced dataset.
4	Combination of CNN, LSTM, and attentions [8]	UNSW-NB15	• Accuracy of 87% • F1-score of 85% • Precision of 90% • Recall of 81%	• Allowed the IDS to operate proactively, anticipating and counteracting emerging attack techniques. • Enhancing the system's ability to detect unknown attacks.	• Computational complicated. • Lower accuracy.
5	Combination of CNN, LSTM, and GRU [9]	Electric Vehicle Charging Station network data	• 100% accuracy in binary classification • 97.44% accuracy in six-class classification • 96.90% accuracy in fifteen-class classification	• High accuracy in detecting cyber threats. • Allowed to detect complex intrusion patterns.	• Still have limitation on new unknown threats.
6	DCNN [10]	IoTID20	• 99.84% accuracy in binary classification • 98.12% accuracy in multiclass classification • 77.55% accuracy in multiclass subclassification	• DCNN model with the Nadam optimizer significantly enhanced the performance of malicious attack detection.	• Finding the optimal batch sizes for different classification tasks, which required additional experimentation and tuning. • Different optimizers (Adam, Nadam, and AdaMax) exhibited varying performances, necessitating extensive testing to determine the best one.
7	Dual CNN [11]	BoT IoT 2020	• 98.04% accuracy • 98.09% precision • 99.85% recall • 98.96% F1-score.	• Effectively detected intrusions by learning hierarchical representations and uncovering patterns and correlations often missed by traditional techniques	• Faced challenges in developing mathematical models and logical frameworks to verify its correctness. • Limited in handling unknown and complex attacks.
8	Three-tier CNN [12]	Real time SDN-IoT dataset	• Detection rate improvement at 25%: 38.72%, at 50%: 58.00% • Decreased failure rate and delay with increasing number of switches • Increased throughput up to 50% • Accuracy in detection of attacks: 99%	• Improved security for IoT devices. • Detection and prevention of intrusions. • Feasibility of multi-tier and deep learning methods.	• Overloading problem of switches on a large-scale environment. • Risk of switch overloading.

III. RESEARCH METHODOLOGY

A CNN combined with LSTM is proposed for an IIoT IDS to overcome the shortcomings of traditional IIoT IDS, particularly its high false negative rate.

The research framework depicted in Fig. 1 functions as a strategic guide, ensuring systematic and efficient execution of the proposed IDS using CNN with LSTM. The project is delineated into four primary phases: Data Pre-processing, Reshaping Tabular Data into Pseudo-image, Development of Classification Model, and Performance Measurement. Each module is tailored to address critical facets of the CNN-based IDS development and evaluation process.

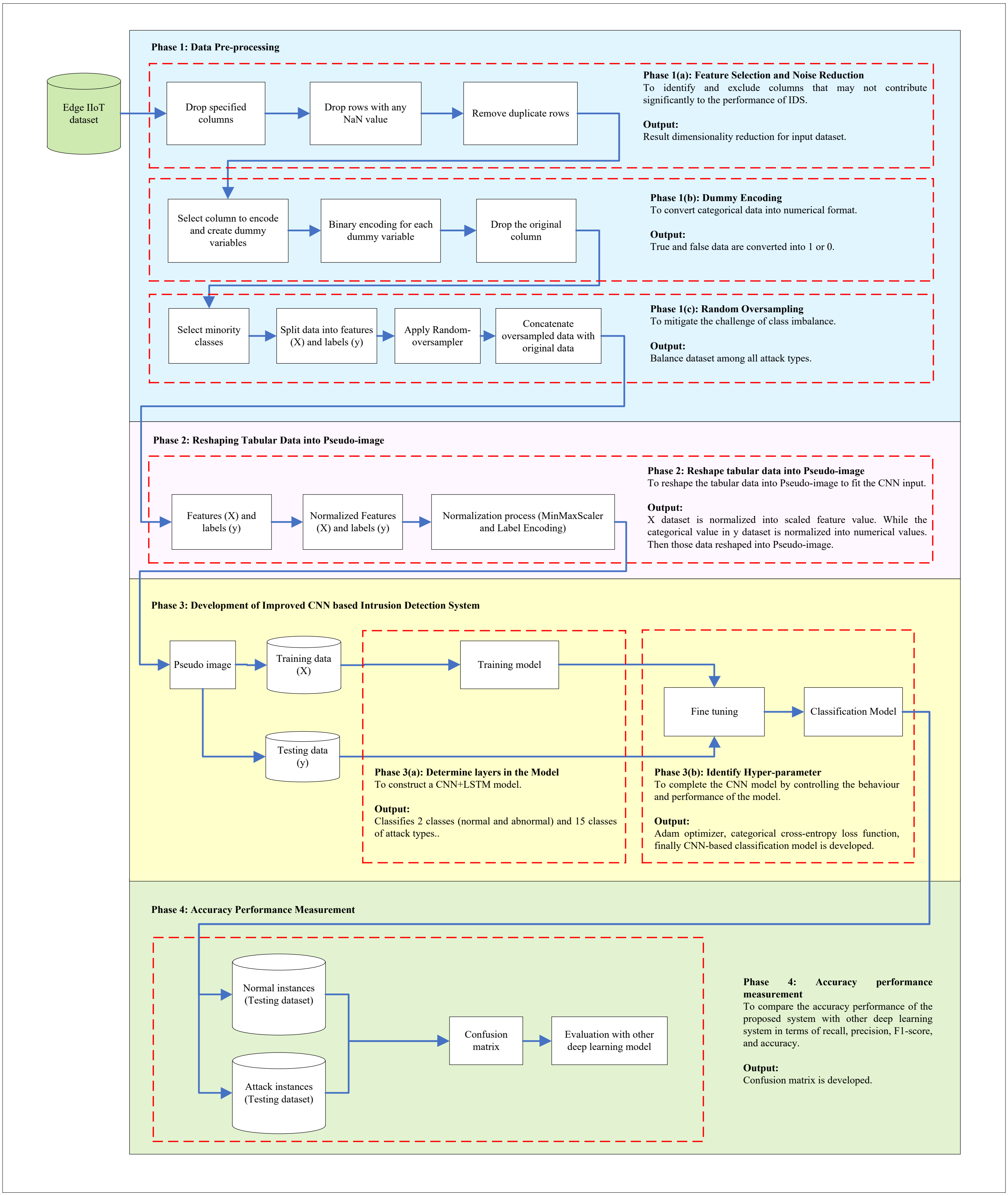

Phase 1: Data Pre-processing
Edge IIoT dataset
Drop specified columns
Drop rows with any NaN value
Remove duplicate rows
Phase 1(a): Feature Selection and Noise Reduction
To identify and exclude columns that may not contribute significantly to the performance of IDS.
Output:
Result dimensionality reduction for input dataset.
Select column to encode and create dummy variables
Binary encoding for each dummy variable
Drop the original column
Phase 1(b): Dummy Encoding
To convert categorical data into numerical format.
Output:
True and false data are converted into 1 or 0.
Select minority classes
Split data into features (X) and labels (y)
Apply Random-oversampler
Concatenate oversampled data with original data
Phase 1(c): Random Oversampling
To mitigate the challenge of class imbalance.
Output:
Balance dataset among all attack types.
Phase 2: Reshaping Tabular Data into Pseudo-image
Features (X) and labels (y)
Normalized Features (X) and labels (y)
Normalization process (MinMaxScaler and Label Encoding)
Phase 2: Reshape tabular data into Pseudo-image
To reshape the tabular data into Pseudo-image to fit the CNN input.
Output:
X dataset is normalized into scaled feature value. While the categorical value in y dataset is normalized into numerical values. Then those data reshaped into Pseudo-image.
Phase 3: Development of Improved CNN based Intrusion Detection System
Pseudo image
Training data (X)
Testing data (y)
Training model
Fine tuning
Classification Model
Phase 3(a): Determine layers in the Model
To construct a CNN+LSTM model.
Output:
Classifies 2 classes (normal and abnormal) and 15 classes of attack types..
Phase 3(b): Identify Hyper-parameter
To complete the CNN model by controlling the behaviour and performance of the model.
Output:
Adam optimizer, categorical cross-entropy loss function, finally CNN-based classification model is developed.
Phase 4: Accuracy Performance Measurement
Normal instances (Testing dataset)
Attack instances (Testing dataset)
Confusion matrix
Evaluation with other deep learning model
Phase 4: Accuracy performance measurement
To compare the accuracy performance of the proposed system with other deep learning system in terms of recall, precision, F1-score, and accuracy.
Output:
Confusion matrix is developed.

Fig. 1. Research Framework of the Proposed IDS using CNNs with LSTM Network

A. Data Pre-processing

To ensure the proposed solution, CNN with LSTM is effective in IIoT environments, it is planned to be evaluated on the famous public benchmark network security datasets, which is Edge-IIoT dataset [13]. To build a trained model, a dataset with labels is required. For this research, a total of 2219201 data points were collected in a CSV file from a real-world IIoT environment. It includes normal traffic data from various IoT sensors and attack traffic covering 14 different attack types (multiclass). Those 2219201 data are also categorized in 2 labels, namely 0 and 1, where 0 indicates Normal while 1 indicates Attack (binary-class).

Firstly, the process of feature selection and noise reduction involves a meticulous examination of the dataset to identify and exclude columns that may not contribute significantly to the performance of the IDS, such as time, type of host, type of tcp, query number, DNS types, and decoded MQTT. By carefully selecting and dropping 18 columns as shown in Table II, this can build a more effective and efficient IDS. This approach enhances the model's ability to identify attack patterns while maintaining privacy and reducing unnecessary complexity.

Dummy encoding, also known as one-hot encoding, is a technique used to convert categorical data into a numerical format that can be utilized by CNN-LSTM model. Features such as 'http.request.method', 'http.referer', 'http.request.version', 'dns.qry.name.len', 'mqtt.conack.flags', 'mqtt.protoname', and 'mqtt.topic' are inherently categorical, so these columns applied dummy encoding to ensure the accurate conversion of categorical features into a numerical format, making them compatible with the proposed model.

TABLE II. REASON OF EXCLUDING THE SELECTED FEATURE

No	Feature	Protocol / Type	Reason for Exclusion
1	'frame.time'	General	Timestamp not essential for pattern recognition and may introduce noise.
2	'ip.src_host'	IP	Hostnames are redundant with IPs and may expose sensitive info.
3	'ip.dst_host'	IP	Same as above; not useful for generalized model learning.
4	'arp.dst.proto_ipv4'	ARP	Protocol-specific address detail, not meaningful in attack pattern detection.
5	'arp.src.proto_ipv4'	ARP	Similar to arp.dst.proto_ipv4, minimal contribution to classification performance.
6	'dns.qry.type'	DNS	Query type alone does not significantly indicate malicious behavior.
7	'icmp.transmit_timestamp'	ICMP	Timing field, high variance and not consistently present across attacks.
8	'icmp.unused'	ICMP	Reserved/unused field, often null or irrelevant.
9	'http.request.uri.query'	HTTP	Contains sensitive user data; also too variable.
10	'http.request.full_uri'	HTTP	Full URI can leak sensitive info; varies too much for pattern learning.
11	'http.tls_port'	HTTP/TLS	Duplicate of port info; does not improve model prediction.
12	'tcp.dstport'	TCP	Dropped to reduce redundancy; similar roles played by other TCP features.
13	'tcp.options'	TCP	High dimensional and complex to encode effectively.
14	'tcp.payload'	TCP	Payload content is noisy and often encrypted or variable.
15	'tcp.srcport'	TCP	As with tcp.dstport, often less predictive in isolation.
16	'udp.port'	UDP	Removed to reduce dependency on specific service ports.
17	'mqtt.msg_decoded_as'	MQTT	Decoded interpretation is complex and not directly useful.
18	'mqtt.msg'	MQTT	Raw MQTT message payload, which may contain encrypted or irrelevant content.

Class imbalance arises when some classes are underrepresented compared to others, causing models to be biased and perform poorly on minority classes. In this dataset, there is a significant imbalance among the different classes, with 'Normal' having 1615643 instances while others like 'MITM' and 'Fingerprinting' have as few as 1214 and 1001 instances, respectively. This severe imbalance can lead to biased machine learning models that are ineffective at predicting minority classes. The method employed is Random Oversampling, it is used to balance the dataset, ensuring that the model can learn from all classes more effectively and make accurate predictions across the board.

B. Reshaping Tabular Data into Pseudo-Image

In order to effectively utilize CNNs for IDS in IIoT environments, the raw tabular data must be transformed into a format that preserves spatial relationships. This section details the preprocessing pipeline, which includes feature normalization, label encoding, and restructuring of the dataset into a pseudo-image representation suitable for CNN input.

Firstly, the data is partitioned into X and y datasets. The dataset consists of n samples and d features, where each sample represents an observation from network traffic, and each feature corresponds to a specific network attribute. Then, the dataset is partitioned into 80% for training and 20% for testing to ensure an optimal balance between model learning and evaluation. This division is widely adopted in the proposed CNN+LSTM model, as a larger training set enhances generalization, while a sufficiently large testing set provides a reliable assessment of model performance on unseen data. Allocating too little data to training may lead to underfitting, whereas an insufficient test set could yield unreliable performance metrics. Additionally, stratified sampling is applied to maintain class distribution, ensuring that both training and testing subsets represent the original dataset proportions. The stratification is crucial to ensure that the distribution of each attack class remains the same in both training and testing sets, preventing potential bias due to variations in class proportions. It does not handle class imbalance directly but rather maintains a consistent class ratio between the two subsets.

Next, the y dataset is normalized by label encoding. Label encoding converts categorical target variables into numerical form. This is essential for CNN algorithms that cannot work directly with categorical data, so the y dataset can be compatible with the training model. After encoding, the labels are one-hot encoded using the categorical encoding. This transformation converts each categorical label into a binary vector of length (number of attack classes), enabling the network to learn from categorical outputs. In addition, X dataset is normalized as well by using Min-Max scaling method, because Neural networks are sensitive to input feature scaling, and unnormalized features can lead to poor convergence and suboptimal performance.

Finally, the processed data can be reshaped into Pseudo-images. Since CNNs require 2D spatial input, the 1D feature vector of each sample is reshaped into a pseudo-image format. This transformation preserves the original feature order while introducing a single-channel structure akin to grayscale images. The resulting shape (n, d) ensures that convolutional layers can effectively learn feature correlations.

C. Development of Classification Model

This CNN-LSTM model as shown in Fig. 2 is designed for intrusion detection by analyzing images generated from network flow data, leveraging both spatial and temporal patterns for improved classification of attack types.

The first stage of the model involves spatial feature extraction using two consecutive one-dimensional convolutional layers (Conv1D). These layers are responsible for capturing local patterns in the network traffic data. The ReLU activation function is applied to introduce non-linearity, followed by a global average pooling layer that reduces the feature dimensionality while retaining critical spatial representations. The input dataset, initially structured as tabular data with shape $(n,92,1)$, is expanded to (n,92,3) by replicating the single feature channel three times. This transformation enables the network to process the input in a manner more similar to RGB image channels, allowing CNN layers to better exploit feature relationships. The CNN component employs Conv1D to extract local spatial dependencies within the input data. Two convolutional layers are used with 32 and 64 filters, respectively, each with a kernel size of 3 and ReLU activation function. The padding is set to "same" to maintain the original input size.

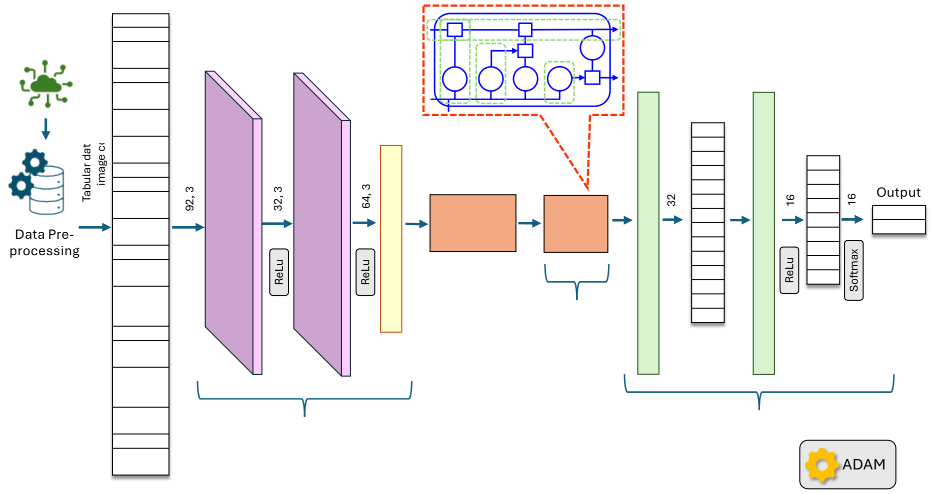


Fig. 2. Model Architecture of Purposed CNN-LSTM Model

Following the convolutional layers, Global Average Pooling (GAP) is applied to reduce the feature dimensionality while preserving spatial information. GAP helps to reduce the risk of overfitting compared to fully connected layers and allows for more efficient model training.

Next, the extracted features are reshaped and passed through an LSTM layer for temporal feature extraction. The LSTM component captures sequential dependencies in network traffic patterns, making the model effective in detecting time-dependent attack behaviors. The LSTM output is then forwarded to a fully connected (dense) classification module.

To ensure IDS models generalize effectively to unseen network traffic, dropout layers with a 50% probability are added after both the LSTM and fully connected layers. This helps prevent overfitting by randomly deactivating neurons during training, encouraging the model to learn robust features instead of simply memorizing the training data. This is crucial in IIoT environments, where network conditions may vary due to dynamic workloads and varying attack strategies.

The classification module consists of multiple dense layers interleaved with dropout layers to prevent overfitting. The final dense layer applies the ReLU activation function, followed by a softmax activation to classify network traffic into either two broad categories (normal vs. attack) or 15 specific attack types. The Adam optimizer is employed to optimize the model's performance by adjusting the learning rate dynamically.

This hybrid CNN-LSTM architecture effectively combines spatial and temporal feature extraction, making it highly suitable for real-time intrusion detection in dynamic IIoT environments.

D. Measurement Metrics

In this research, the performance evaluation of the proposed model is analyzed by its precision, recall, accuracy, and F1-score. For those unfamiliar with these metrics, a higher precision score indicates the model's ability to accurately identify the correct targets. The confusion matrix summarizes four key components: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). In this context, True Positives (TP) refer to instances where the model correctly identifies the presence of a defect, as confirmed by the ground truth. True Negatives (TN) represent cases where the model accurately detects the absence of defects. False Positives (FP) occur when the model incorrectly predicts a defect that isn't present, while False Negatives (FN) represent instances where the model fails to identify an existing defect, as indicated by the ground truth.

The accuracy measures out of those all images, how many of it is actually attack and actually normal, which indicates the ratio of TP and TN out of all predictions (TP+TN+FP+FN) as shown in Eq. 1. The higher accuracy, the better the CNN model in detecting the attacks in IIoT.

The precision measures out of those predicted attacks, how many of it is actually attack, which indicates the ratio of TP out of all positive predictions (TP+FP) as shown in Eq. 2. The precision value may vary based on the model's confidence threshold.

Recall measures the ability of CNN network to correctly detect the image as attack, which indicates the ratio of TP out of all predictions (TP+FN) as shown in Eq. 3. The higher recall, the better the CNN model in detecting the normal traffic in IIoT.

F1 score finds the most optimal confidence score threshold where precision and recall give the highest F1 score, so it is suitable to evaluate the model performance. The F1 score calculates the balance between precision and recall as shown in Eq. 4. If the value of F1 score is high, precision and recall are high, and vice versa.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

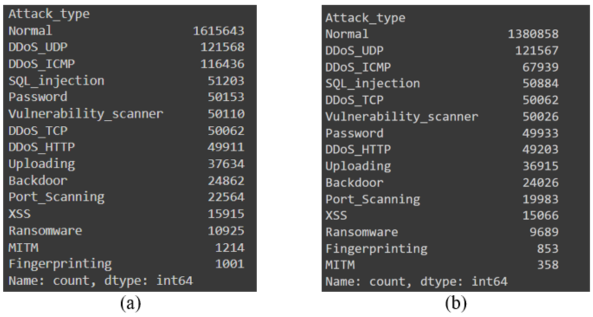
$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (4)$$

IV. RESULT AND DISCUSSION

This chapter presented a comprehensive evaluation of the proposed CNN + LSTM model for intrusion detection in Edge-IIoT networks, focusing on both binary and multi-class classification tasks. The performance analysis was conducted using key evaluation metrics, including accuracy, precision, recall, and F1-score, to assess the model's capability in distinguishing normal and attack traffic, as well as various cyber threats.

A. Result for Data Pre-Processing

By carefully selecting and dropping 18 columns like 'frame.time', 'ip.src_host', 'ip.dst_host', 'arp.dst.proto_ipv4', 'arp.src.proto_ipv4', 'dns.qry.type', 'icmp.transmit_timestamp', 'icmp.unused', 'http.request.uri.query', 'http.request.full_uri', 'http.tls_port', 'tcp.dstport', 'tcp.options', 'tcp.payload', 'tcp.srcport', 'udp.port', and 'mqtt.msg_decoded_as.' and 'mqtt.msg', this approach enhances the model's ability to identify attack patterns while maintaining privacy and reducing unnecessary complexity. The updated counts of each attack as shown in Fig. 3.



```
Attack_type
Normal                     1615643
DDoS_UDP                    121568
DDoS_ICMP                   116436
SQL_injection                51203
Password                     50153
Vulnerability_scanner        50110
DDoS_TCP                     50062
DDoS_HTTP                    49911
Uploading                    37634
Backdoor                     24862
Port_Scanning                22564
XSS                          15915
Ransomware                   10925
MITM                          1214
Fingerprinting                1001
Name: count, dtype: int64
```

(a)

```
Attack_type
Normal                     1380858
DDoS_UDP                    121567
DDoS_ICMP                    67939
SQL_injection                50884
DDoS_TCP                     50062
Vulnerability_scanner        50026
Password                     49933
DDoS_HTTP                    49203
Uploading                    36915
Backdoor                     24026
Port_Scanning                19983
XSS                          15066
Ransomware                    9689
Fingerprinting                 853
MITM                           358
Name: count, dtype: int64
```

(b)

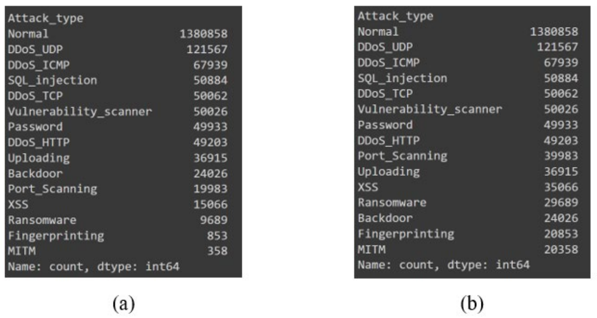
Fig. 3. Counts for each Attack Type; (a) Before Feature Selection and Noise Reduction; (b) After Feature Selection and Noise Reduction

Dummy encoding allows for the retention of all unique categories within a feature without imposing any ordinal relationship between them. For instance, 'http.request.method' might include categories like "GET", "POST", "PUT", etc. Dummy encoding ensures that each method is treated distinctly, preserving the integrity and specificity of the data. The result of counts of each attack before and after dummy encoding remain unchanged.

Then, Random oversampling is used for the dataset. There is a significant imbalance among the different classes, with 'Normal' having 1615643 instances while others like 'MITM' and 'Fingerprinting' have as few as 1214 and 1001 instances, respectively. This severe imbalance can lead to biased machine learning models that are ineffective at predicting minority classes. The purpose of the technique used here is to address this issue by balancing the dataset, ensuring that the model can learn from all classes more effectively and make accurate predictions across the board.

As result shown in Fig. 4, Random Oversampling works by randomly duplicating instances from the minority classes until the desired balance is achieved. In this code, the specific minority classes are specified, which are 'Port_Scanning', 'XSS', 'Ransomware', 'Fingerprinting', 'MITM' and specified the number of desired samples for each class (20000 in this case).

The process involves filtering the dataset to isolate these minority classes, applying the 'RandomOverSampler' to generate additional instances, and finally concatenating the oversampled data back with the original dataset.



```
Attack_type
Normal                    1380858
DDoS_UDP                   121567
DDoS_ICMP                   67939
SQL_injection               50884
DDoS_TCP                    50062
Vulnerability_scanner       50026
Password                    49933
DDoS_HTTP                   49203
Uploading                   36915
Backdoor                    24026
Port_Scanning               19983
XSS                         15066
Ransomware                   9689
Fingerprinting                853
MITM                          358
Name: count, dtype: int64
```

(a)

```
Attack_type
Normal                    1380858
DDoS_UDP                   121567
DDoS_ICMP                   67939
SQL_injection               50884
DDoS_TCP                    50062
Vulnerability_scanner       50026
Password                    49933
DDoS_HTTP                   49203
Port_Scanning               39983
Uploading                   36915
XSS                         35066
Ransomware                  29689
Backdoor                    24026
Fingerprinting              20853
MITM                        20358
Name: count, dtype: int64
```

(b)

Fig. 4. Counts for each Attack Type; (a) Before Random Oversampling (b) After Random Oversampling

B. Result of Reshaping Tabular Data into Pseudo-Image

The data normalization process seeks to standardize the range of independent variables or features in the dataset, ensuring that each feature contributes equally to the training of the machine learning model. This is especially crucial for models that use gradient-based optimization, like CNNs, as it prevents features with larger ranges from dominating the objective function and skewing the results.

Data Shapes and Unique Values of X dataset after MinMaxScaler:

- X_train shape = (1541843, 92, 1): The training dataset consists of 1541843 samples, each with 92 features. The additional dimension (1) indicates that the features are treated as single-channel inputs, similar to how grayscale images are processed in image recognition tasks.
- X_test shape = (385461, 92, 1): The test dataset consists of 385461 samples, each with the same 92 features.

The shapes of 'X_train' and 'X_test' confirm that the data has been properly structured for input into the CNN, with each feature scaled to a range that ensures uniform contribution during the model training process.

Unique Values in Labels of Y dataset from label encoding:

- Y_train: The unique values in the training labels range from 0 to 14, indicating that there are 15 distinct classes in the dataset. This is consistent with the classification task at hand, where the goal is to categorize each sample into one of these 15 classes.
- Y_test: Similarly, the unique values in the test labels range from 0 to 14, confirming that the test set also includes samples from all 15 classes.

The data normalization process ensures that all features in the dataset contribute equally to model training by scaling them to a consistent range. This prevents features with larger scales from dominating the learning process and promotes more stable and reliable convergence during training. The shapes of the training and test datasets confirm that the data is properly structured for input into the CNN. Additionally, label encoding converts categorical target variables into a numerical format that is compatible with machine learning algorithms, allowing the model to effectively learn and classify the samples into their respective classes.

C. Experimental Result

1) Binary Classification

The confusion matrix as shown in Table III demonstrates that the model performs well in identifying normal network traffic, with 272929 true negatives, indicating a strong ability to correctly classify benign activity in the IIoT environment. This high accuracy in recognizing normal traffic is essential in minimizing false alarms, ensuring that legitimate network activity is not mistakenly flagged as malicious.

TABLE III. CONFUSION MATRIX OF BINARY CLASSIFICATION FOR NORMAL AND ATTACK LABEL

		Predicted Label	
True Label	**Normal (0)**	272929	3469
	Attack (1)	107766	1297

Additionally, the model successfully detects 1297 attack instances, showcasing its capability to recognize cyber threats within the Edge-IIoT dataset. This highlights the effectiveness of the convolutional and sequential learning approach in extracting meaningful patterns from network traffic, enabling the detection of potentially harmful activities. The combination of CNN and LSTM likely contributes to this strength by capturing both spatial and temporal dependencies in the data.

Furthermore, the relatively low number of false positives (3469) suggests that the model maintains a balance in its decision-making process, reducing unnecessary security alerts. Although a significant number of attack instances (107766) were misclassified as normal (false negatives), the model still provides a solid foundation for intrusion detection. Its strengths in identifying normal traffic and detecting attacks in certain cases make it a valuable component for IIoT security. This is crucial for practical deployment in IIoT security, where excessive false alarms can lead to inefficiencies and unnecessary interventions. With these strengths, the model provides a solid foundation for further enhancement, potentially making it a valuable component in real-world intrusion detection systems.

TABLE IV. EDGE-IIOT BINARY CLASSIFICATION RESULT

Class	Model	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)
Normal	CNN + LSTM (Proposed Solution)	71	99	72	83
	AlexNet CNN + ViT [5]	50	77	32	25
	MobileNet CNN + ViT [7]	54	74	65	78
	Res + CNN + SRU [6]	65	55	34	79
Attack	CNN + LSTM (Proposed Solution)	71	1	27	2
	AlexNet CNN + ViT [5]	42	14	18	20
	MobileNet CNN + ViT [7]	33	23	6	17
	Res + CNN + SRU [6]	35	15	15	11

The comparative analysis in Table IV highlights the performance of the proposed CNN + LSTM model against other state-of-the-art models for binary classification. The CNN + LSTM model outperforms alternative approaches in overall accuracy (71%) and demonstrates significantly higher recall (99% for normal traffic). This indicates that the model can successfully identify most normal instances, reducing false alarms in industrial IoT environments. However, the recall for attack traffic is considerably lower (1%), suggesting that a large proportion of attacks go undetected. Despite this, the precision for normal traffic (72%) remains higher than the competing models, reinforcing its reliability in recognizing benign traffic.

Compared to other models, AlexNet CNN + ViT [5] and MobileNet CNN + ViT [7] exhibit lower accuracy (50% and 54%, respectively) and struggle with precision and F1-score, particularly in detecting normal traffic. The Res + CNN + SRU [6] model shows a moderate performance with 65% accuracy, achieving a higher F1-score (79%) for attack traffic but at the cost of significantly lower recall (55% for normal and 15% for attack traffic).

Overall, the CNN + LSTM model demonstrates superior performance in distinguishing normal traffic and maintaining a balanced accuracy score. While it faces challenges in detecting attack traffic, its high recall for normal traffic and competitive F1-score suggest that it provides a strong foundation for industrial IoT intrusion detection. Further enhancements, such as feature engineering and hybrid learning approaches, could help improve the model's ability to capture diverse attack patterns while maintaining its strengths in recognizing normal traffic.

2) *Multiclass Classification*

The normalized confusion matrix for multi-class classification of the Edge-IIoT dataset presented in Fig. 5 provides insights into the model's ability to differentiate between various types of network attacks. The matrix presents classification performance across multiple attack categories, including DDoS (HTTP, ICMP, TCP, UDP), SQL injection, ransomware, port scanning, MITM (Man-in-the-Middle), fingerprinting, and more, along with normal traffic. The diagonal elements represent the correctly classified instances for each class, while off-diagonal values indicate misclassifications, showing how different attack types may be confused with one another.

The confusion matrix indicates that the model performs well in detecting certain attack types, particularly those with a stronger presence in the dataset. The darker diagonal values suggest that a substantial proportion of samples for each class were classified correctly, demonstrating the model's ability to recognize distinguishing features of various cyber threats. For instance, MITM, SQL injection, and certain DDoS attacks appear to be well-classified, as indicated by their relatively higher values along the diagonal. This suggests that the model has effectively learned key characteristics of these attacks from the training data.

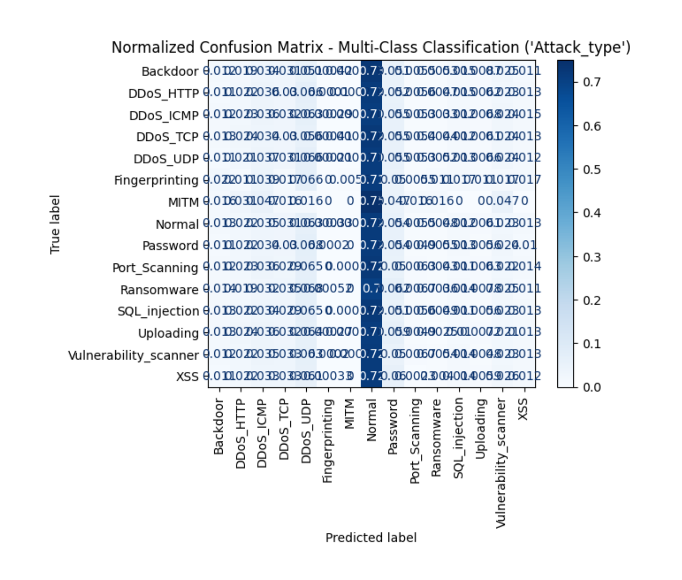


Fig. 5. Confusion Matrix of Multi-class Classification for each Attack

However, some degree of misclassification is observed between attack types with similar behaviors, as seen in the off-diagonal values. For example, different DDoS variants (HTTP, ICMP, TCP, UDP) exhibit some level of confusion, likely due to their overlapping traffic patterns. Similarly, certain scanning attacks (port scanning, fingerprinting, vulnerability scanning) show misclassification tendencies, which may be attributed to their shared methodology of probing network vulnerabilities. Despite these challenges, the confusion matrix reflects the model's capacity to identify and differentiate various cyber threats, making it a valuable tool for intrusion detection in IIoT environments. Further optimization, such as refining feature selection and incorporating attention mechanisms, could further enhance classification accuracy.

TABLE V. EDGE-IIOT MULTI-CLASS CLASSIFICATION RESULT

Class	Precision (%)	Recall (%)	F1-score (%)	Total Accuracy (%)
Backdoor	1.21	1.23	1.22	52.33
DDoS_HTTP	2.55	2.20	2.36	
DDoS_ICMP	3.63	3.61	3.62	
DDoS_TCP	2.48	3.04	2.73	
DDoS_UDP	6.55	6.61	6.58	
Fingerprinting	0	0	0	
MITM	0	0	0	
Normal	71.71	71.65	71.68	
Password	2.58	5.38	3.48	
Port_scanning	1.18	0.63	0.82	
Ransomware	0.38	0.36	0.37	
SQL_injection	2.36	1.11	1.51	
Uploading	2.26	0.72	1.09	
Vulnerability_scanner	2.52	2.26	2.38	
XSS	0.73	1.19	0.91	

Based on Table V, the evaluation of the multi-class classification performance for the intrusion detection system on the Edge-IIoT dataset provides a comprehensive insight into the model's capability to distinguish between different types of cyber threats. The overall accuracy of 52.33% indicates that the model demonstrates moderate effectiveness in classifying network traffic into distinct attack categories. Among the evaluated classes, normal traffic is classified with significantly higher precision (71.71%), recall (71.65%), and F1-score (71.68%), suggesting that the model is well-optimized for distinguishing between benign and malicious activity. This high recognition rate for normal instances helps in reducing false alarms and improving the reliability of the intrusion detection system in real-world industrial IoT environments.

However, the classification performance varies significantly across different attack types. Certain attack categories, such as DDoS-related attacks (DDoS_HTTP, DDoS_ICMP, DDoS_TCP, and DDoS_UDP), demonstrate relatively higher recognition rates, with DDoS_UDP achieving the highest F1-score (6.58%) among all attack types. This suggests that the model is better at detecting high-volume, network-flooding attacks compared to more sophisticated, stealthy threats. Conversely, classes such as MITM (Man-in-the-Middle) and Fingerprinting attacks show zero recall, precision, and F1-score, indicating that the model struggles to identify these attack patterns. This could be attributed to an imbalance in the dataset, where certain attack types might have significantly fewer instances, leading to poor generalization.

Other cyber threats, such as password attacks, SQL injection, and vulnerability scanning, exhibit low detection rates, with F1-scores ranging between 1.09% and 3.48%. This suggests that while the model can identify certain attack types to some extent, further optimization is required to enhance its ability to recognize sophisticated exploits that rely on application-layer vulnerabilities rather than sheer network traffic anomalies. The relatively low performance across multiple attack categories indicates a need for feature refinement, improved model architectures, or the integration of additional contextual data to enhance classification accuracy across all categories.

Overall, while the model shows strong capabilities in classifying normal traffic and detecting high-volume attacks, it faces challenges in recognizing more sophisticated threats, particularly low-frequency and evasive attack types.

D. Comparative Analysis of Edge-IIoT Classification Result

The proposed CNN + LSTM model was evaluated using the Edge-IIoT dataset and compared against three existing deep learning-based intrusion detection models: AlexNet CNN + ViT [5], MobileNet CNN + ViT [7], and Res + CNN + SRU [6] to further validate the multiclass classification performance. The results of this comparison are summarized in Table VI, providing insights into the model's relative performance against established approaches.

TABLE VI. COMPARISON OF EDGE-IIOT CLASSIFICATION RESULT

Model	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
CNN + LSTM (Proposed Solution)	52.36	52.33	52.31	52.33
AlexNet CNN + ViT [5]	48.92	48.75	48.80	48.77
MobileNet CNN + ViT [7]	49.15	49.02	49.08	49.04
Res + CNN + SRU [6]	50.47	50.32	50.29	50.30

The CNN + LSTM model achieves a precision of 52.36%, recall of 52.33%, F1-score of 52.31%, and accuracy of 52.33%, surpassing the competing models in every aspect. Among the baseline models, the Res + CNN + SRU approach exhibits the highest performance, achieving an accuracy of 50.30%, which is still lower than the proposed method. This indicates that the hybridization of convolutional and sequential learning techniques, as implemented in the CNN + LSTM model, enhances intrusion detection capabilities by capturing both spatial and temporal features in network traffic data.

The AlexNet CNN + ViT and MobileNet CNN + ViT models demonstrate comparatively lower performance, with accuracy values of 48.77% and 49.04%, respectively. The decline in performance for these models may be attributed to their reliance on vision transformer (ViT) components, which, despite their effectiveness in image-related tasks, may not be as well-suited for sequential intrusion detection tasks in IIoT networks. Additionally, the computational complexity of ViTs may introduce inefficiencies in processing high-dimensional network traffic data, leading to suboptimal classification results.

The radar chart illustrated in Fig. 6 presents a comparative analysis of different deep learning models for Edge-IIoT classification based on four performance metrics: precision, recall, F1-score, and accuracy. The proposed CNN + LSTM model achieves the highest performance across all metrics, with values around 52.5%, demonstrating its superior ability to capture spatial and temporal dependencies for effective intrusion detection. The ResNet + CNN + SRU model [6] follows, with slightly lower performance but still outperforming the other benchmark models. MobileNet CNN + ViT [7] and AlexNet CNN + ViT [5] exhibit lower performance, with scores below 50%, indicating limitations in their capability to extract meaningful patterns from IIoT network traffic. The results highlight the effectiveness of integrating CNN and LSTM for IIoT cybersecurity, reinforcing its robustness in identifying cyber threats with improved classification accuracy compared to alternative architectures.

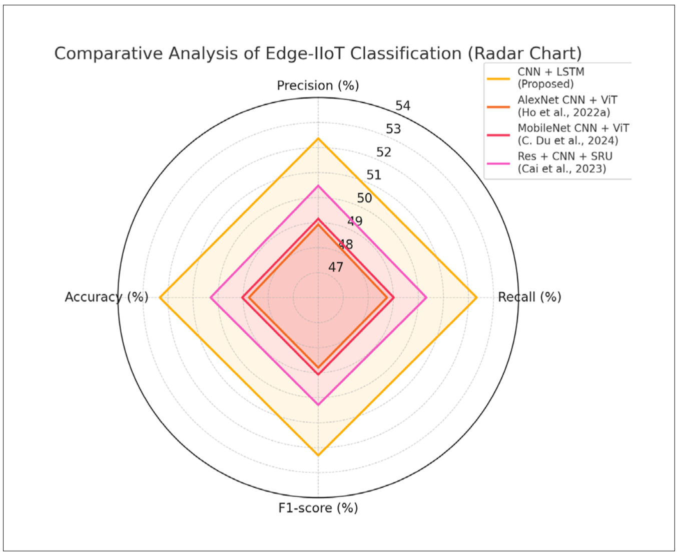


Fig. 6. Comparison of Performance of Deep Learning Model in Edge-IIoT

In terms of recall, the proposed CNN + LSTM model also outperforms the other models (52.33%), demonstrating its capability to detect a higher proportion of actual intrusions. A high recall score means that the model effectively captures both common and rare attack patterns, reducing the risk of undetected cyber threats. The ResNet + CNN + SRU model follows with a slightly lower recall, indicating a moderate ability to detect threats. However, the MobileNet CNN + ViT and AlexNet CNN + ViT models show lower recall values, suggesting that they have a higher tendency to miss certain intrusions, potentially leaving IIoT networks vulnerable to undetected cyberattacks.

The F1-score, which balances precision and recall, is also highest for the CNN + LSTM model (52.31%). This confirms that the proposed model provides a well-balanced performance in reducing both false positives and false negatives, ensuring more accurate intrusion detection. The ResNet + CNN + SRU model comes in second, demonstrating a reasonable trade-off between precision and recall, though it still underperforms compared to CNN + LSTM. Meanwhile, the MobileNet CNN + ViT and AlexNet CNN + ViT models have the lowest F1-scores, highlighting their limitations in achieving a stable balance between detecting threats and minimizing false alarms.

Finally, the proposed CNN + LSTM model achieves the highest accuracy (52.33%), reaffirming its overall effectiveness in correctly classifying IIoT intrusions. A high accuracy score suggests that the model generalizes well across different attack types, making it a more robust solution for real-world applications. The ResNet + CNN + SRU model ranks second, showing competitive performance but still slightly lagging behind CNN + LSTM. The MobileNet CNN + ViT and AlexNet CNN + ViT models, however, display lower accuracy, indicating that they struggle to consistently classify different intrusion types. This could be due to limitations in feature extraction or an inability to capture sequential dependencies effectively.

Overall, the results highlight the advantages of integrating convolutional and recurrent neural networks for IIoT security applications. The superior performance of the CNN + LSTM model suggests that leveraging both spatial feature extraction and sequential dependency modeling enhances intrusion detection accuracy. These findings reinforce the effectiveness of hybrid deep learning architectures in addressing security challenges in IIoT environments and provide a foundation for further advancements in intelligent threat detection systems.

V. CONCLUSION

This research offers several key contributions to the field of intrusion detection in Edge-IIoT environments by developing a hybrid deep learning model that strengthens network security. The study presents a CNN and LSTM-based architecture that effectively captures both spatial and temporal dependencies in network traffic data, tackling the challenges posed by dynamic and evolving cyber threats. By integrating convolutional layers for feature extraction with sequential learning through LSTMs, the model improves the detection of malicious activities while maintaining high recognition of normal network behavior.

This research aimed to enhance the security of IIoT environments by developing an advanced IDS capable of identifying and analyzing cyber threats. Through comprehensive analysis of IIoT network traffic, various types of intrusions and malware were examined, including DoS, SQL injection, ransomware, and other attack vectors. The study highlighted the vulnerabilities present in IIoT devices and the challenges faced by traditional IDS in detecting evolving cyber threats. The findings emphasize the need for robust detection mechanisms that can adapt to dynamic attack patterns while maintaining high accuracy and minimal false alarms.

To address these challenges, this research implemented a deep learning-based IDS by integrating CNN and LSTM. The CNN component enabled effective spatial feature extraction from network traffic data, while LSTM captured temporal dependencies, improving the model's ability to detect anomalies in IIoT networks. The proposed CNN-LSTM model was trained and tested on a large-scale IIoT dataset, demonstrating its capability in differentiating normal and malicious activities.

Compared to traditional IDS and other state-of-the-art deep learning models, the proposed approach exhibited superior performance in identifying network anomalies, highlighting its effectiveness in IIoT security applications.

The evaluation of the intrusion classification model was conducted using performance metrics such as accuracy, F1-score, precision, and recall. The model achieved a binary classification accuracy of 71%, successfully detecting normal traffic with a recall of 99%, thereby reducing false positives. However, the detection of attack traffic, particularly low-frequency attacks, remains an area for further improvement. In multi-class classification, the model demonstrated its capability to recognize high-volume attacks, such as Distributed Denial-of-Service (DDoS), but struggled with more sophisticated attack types like Man-in-the-Middle (MITM) and fingerprinting attacks. These results indicate that while the proposed model enhances IIoT security, additional optimization is required to improve its ability to classify diverse cyber threats more effectively.

In conclusion, this research provides a significant contribution to IIoT cybersecurity by developing a deep learning-based IDS that enhances intrusion detection capabilities. The findings underscore the importance of leveraging both spatial and temporal analysis for improved threat detection. Future enhancements, including feature selection refinements, adversarial training, and attention mechanisms, could further improve classification performance. The proposed model serves as a foundation for advancing IDS solutions in IIoT environments, ensuring better protection against emerging cyber threats in industrial systems.

VI. FUTURE RESEARCH DIRECTIONS

Future research should focus on enhancing the effectiveness of intrusion detection systems in Industrial Internet of Things (IIoT) environments by addressing the limitations identified in this study. One potential direction is the integration of attention mechanisms within the CNN-LSTM framework to improve the model's ability to focus on critical network traffic patterns. Attention-based models have demonstrated significant success in various domains, particularly in natural language processing and time-series analysis, and could enhance the detection of sophisticated attack patterns that might otherwise be overlooked. Additionally, refining feature engineering techniques and incorporating domain-specific knowledge into feature selection can further optimize the model’s performance.

Another promising avenue is the exploration of hybrid deep learning architectures that combine multiple neural network structures to improve classification accuracy. For instance, transformer-based models could be integrated with CNN-LSTM to capture both local and long-range dependencies in network traffic. This could address the challenge of detecting low-frequency and highly evasive cyber threats such as MITM and fingerprinting attacks. Furthermore, leveraging federated learning techniques would enable intrusion detection models to be trained across multiple IIoT environments without compromising data privacy, making the system more adaptable to real-world deployment in diverse industrial settings.

Lastly, future research should focus on improving the generalizability of intrusion detection models by expanding datasets and incorporating real-world attack scenarios. Current models often suffer from performance degradation when applied to different IIoT environments due to variations in network behavior and attack characteristics. Developing a robust anomaly detection framework that can adapt to evolving cyber threats through continual learning mechanisms would significantly enhance the reliability of intrusion detection systems. Additionally, benchmarking against larger and more diverse datasets, as well as conducting real-time implementation and validation in industrial settings, will provide deeper insights into the practical applicability and effectiveness of deep learning-based security solutions in IIoT networks.

ACKNOWLEDGMENT

I would like to express our sincere gratitude to our supervisor, Dr. Yusliza binti Yusoff, for his valuable guidance and support throughout the research process. Her expertise and insights were invaluable in shaping the research and helping me to overcome challenges. Without her continued support and interest, this research would not have been the same as presented here.

CONFLICTS OF INTEREST

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

REFERENCES

[1] T. Alam. (2018). A Reliable Communication Framework and Its Use in Internet of Things (IoT). *International Journal of Scientific Research in Computer Science, Engineering and Information Technology, IJSRCSEIT*, *5*(10), 450–456. https://www.researchgate.net/publication/325645304.

[2] Y. Shah and S. Sengupta. (2020). A Survey on Classification of Cyber-attacks on IoT and IIoT Devices. *2020 11th IEEE Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2020*, Institute of Electrical and Electronics Engineers Inc., 0406–0413. Doi: 10.1109/UEMCON51285.2020.9298138.

[3] Rebecca Gurley Bace. (2000). *Intrusion Detection*, Technology Series. Macmillan Technical Publishing USA. https://books.google.com.sg/books?id=VLgVMlV476IC&printsec=frontcover#v=onepage&q&f=false

[4] A. Krish, L. M. Ashik, A. Mathewkutty, D. Jacob, and M. Hari. (2020). Intrusion Detection and Prevention System Using Deep Learning. *Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020).*

[5] C. M. K. Ho, K. C. Yow, Z. Zhu, and S. Aravamuthan. (2022). Network Intrusion Detection via Flow-to-image Conversion and Vision Transformer Classification. *IEEE Access*. Doi: 10.1109/ACCESS.2022.3200034.

[6] Z. Cai, Y. Si, J. Zhang, L. Zhu, P. Li, and Y. Feng. (2023). Industrial Internet Intrusion Detection based on Res-CNN-SRU. *Electronics (Switzerland)*, *12*(15). Doi: 10.3390/electronics12153267.

[7] C. Du, Y. Guo, and Y. Zhang. (2024). A Deep Learning-Based Intrusion Detection Model Integrating Convolutional Neural Network and Vision Transformer for Network Traffic

Attack in the Internet of Things. *Electronics (Basel)*, *13*(14), 2685. Doi: 10.3390/electronics13142685.

[8] K. Psychogyios, A. Papadakis, S. Bourou, N. Nikolaou, A. Maniatis, and T. Zahariadis. (2024). Deep Learning for Intrusion Detection Systems (IDSs) in Time Series Data. *Future Internet*, *16*(3). Doi: 10.3390/fi16030073.

[9] D. Kilichev, D. Turimov, and W. Kim. (2024). Next–Generation Intrusion Detection for IoT EVCS: Integrating CNN, LSTM, and GRU Models. *Mathematics*, 12(4). Doi: 10.3390/math12040571.

[10] S. Ullah *et al.* (2022), A New Intrusion Detection System for the Internet of Things via Deep Convolutional Neural Network and Feature Engineering. *Sensors*, *22*(10). Doi: 10.3390/s22103607.

[11] B. A. Alabsi, M. Anbar, and S. D. A. Rihan. (2023). CNN-CNN: Dual Convolutional Neural Network Approach for Feature Selection and Attack Detection on Internet of Things Networks. *Sensors*, 23(14). Doi: 10.3390/s23146507.

[12] A. Ali, A. Ali, and M. M. Yousaf. (2020). Novel Three-Tier Intrusion Detection and Prevention System in Software Defined Network. *IEEE Access*, *8*, 109662–109676. Doi: 10.1109/ACCESS.2020.3002333.

[13] M. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke. (2022). Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning. Doi: https://doi.org/10.36227/techrxiv.18857336.v1.