SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V11I3P128

Volume 11 Issue 3, 332-340, March 2024
© 2024 Seventh Sense Research Group®

Original Article

Real-Time Anomaly Detection in IoT Networks Using Deep Learning over Wireless Channels

K. Sudharson[1*], C.S. Anita[1#], M.A. Berlin[2*], G. Eswari @ Petchiammal[2#], J. Deepika[3*], K. Selvi[3#]

[1&1#]Department of AIML, R.M.D. Engineering College, Tamil Nadu, India.*
[2&3*]School of Computer Science and Engineering, Vellore Institute of Technology University, Tamil Nadu, India.*
[2#]Department of Information Technology, Velammal Institute of Technology, Tamil Nadu, India.
[3#]Department of MBA, S.A. Engineering College, Tamil Nadu, India.

[1]Corresponding Author : ksudharson@gmail.com*

Received: 19 January 2024 Revised: 15 February 2024 Accepted: 17 March 2024 Published: 25 March 2024

Abstract *- This study introduces IoT-AnomalyNet, a novel deep-learning approach designed for real-time anomaly detection in IoT networks operating over wireless channels. IoT-AnomalyNet combines long short-term memory networks (LSTMs), Convolutional Neural Networks (CNNs), autoencoders, attention mechanisms, and hybrid architectures to effectively identify patterns in both spatial and temporal dimensions within IoT sensor data streams. Through comprehensive experimentation with diverse datasets and IoT sensor readings, IoT-AnomalyNet achieves an impressive accuracy rate of 95.73% for anomaly detection. Notably, IoT-AnomalyNet outperforms traditional machine learning methods with remarkable recall (97.5%) and precision (95.5%) rates for normal instances and recall (97.85%) and precision (96.24%) rates for attack instances. These results underscore the efficacy of deep learning methodologies in accurately detecting anomalies in real-time IoT data streams transmitted via wireless networks. By proactively identifying abnormal behaviors, IoT-AnomalyNet holds significant promise in mitigating risks, ensuring continuous operation, and enhancing the security and reliability of IoT systems.*

Keywords *- Anomaly detection, Attention mechanisms, Deep Learning, IoT networks, Wireless channels.*

1. Introduction

The introduction of Internet of Things (IoT) devices has ushered in a new era of connectivity by simplifying the collection, analysis, and application of massive volumes of sensor data. This has transformed various sectors. IoT technologies are widely used in various industries, such as industrial automation, healthcare, smart cities and homes. They offer never-before-seen opportunities for increased creativity and productivity. These innovations have significantly altered how we interact with and perceive the world around us, enabling the seamless fusion of digital and physical environments.

IoT technology does, however, have a lot of disadvantages, particularly when preserving the security and dependability of IoT networks. The attack surface for potential threats and vulnerabilities is growing due to the growing number of devices that are connected to one another. Cybersecurity concerns are serious because malicious actors attempt to exploit vulnerabilities in IoT networks for illicit purposes, including data breaches, sabotage, and espionage. Deploying effective security measures is challenging due to the additional complexity that comes with IoT systems' dynamic and heterogeneous nature.

Despite these challenges, anomaly detection proves to be an essential component in safeguarding IoT environments from errors, threats, or inefficiencies. The ability to promptly spot deviations from typical behaviour patterns is necessary to preserve IoT systems' integrity and operational continuity. However, standard anomaly detection techniques-which occasionally rely on pre-established criteria or thresholds-are ill-suited to deal with IoT situations since they are dynamic and diverse.

In addition, the constraints imposed by wireless communication routes exacerbate the difficulty of accurately and swiftly detecting abnormalities. In light of this, new approaches and methods are desperately needed to deal with these problems and ensure the dependability and security of IoT networks in a future where connectivity is expanding.

This work aims to overcome the major challenges associated with anomaly identification in wirelessly channelled Internet of Things (IoT) networks. Realising how important it is to defend IoT ecosystems from threats and weaknesses, the primary objective is to develop a robust anomaly detection system tailored to this specific setting.

The proposed methodology leverages deep learning techniques to effectively capture the intricate temporal and spatial patterns found in Internet of Things sensor data. More precisely, including autoencoders, Long Short-Term Memory networks (LSTMs), and Convolutional Neural Networks (CNNs) enable the model to learn and express complex properties from raw sensor data.

Autoencoders are utilised for dimensionality reduction and unsupervised feature learning to derive meaningful representations from high-dimensional sensor inputs. A strong option for capturing temporal dynamics in time-series data includes long-range associations and sequential patterns, which LSTMs may capture. In contrast, CNNs excel at extracting spatial features and hierarchies from multidimensional sensor data, allowing the model to detect irregularities and trends in specific geographical locations.

Additionally, the model design incorporates attention approaches that enable the selective concentration of attention on important features or time steps in the data. This selective attention method enhances the model's capacity to identify subtle abnormalities within normal data by concentrating its attention on relevant regions of the input space. By dynamically balancing the values of several features or time steps, the model can effectively filter out noise and irrelevant information, improving the robustness and accuracy of anomaly detection in IoT networks.

Taking everything into account, the proposed methodology is innovative and comprehensive for anomaly identification in IoT networks across wireless channels. Through the use of deep learning algorithms and attention mechanisms, the research seeks to create an advanced anomaly detection system that can successfully defend IoT ecosystems against potential threats and weaknesses. In a world where connectivity is growing, this will guarantee the security and dependability of IoT networks.

The upcoming sections of this introduction will explore the intricacies involved in detecting anomalies in Internet of Things networks, elucidate the crucial function that deep learning techniques play in resolving these concerns, underscore the significance of wireless communication channels in IoT ecosystems, and clarify the rationale behind the inclusion of attention mechanisms in the proposed anomaly detection framework. The report will also include the objectives of the study, its scope, and its organisational structure.

2. Related Works

The issues of anomaly detection in IoT networks across wireless channels are addressed by the suggested methodology, which uses multiple deep learning algorithms. In order to fully capture the intricate temporal and spatial patterns found in IoT sensor data, each technique is essential.

2.1. Auto Encoders

This study uses autoencoders, a crucial part of artificial neural networks, for dimensionality reduction and unsupervised feature learning [1]. They have been thoroughly examined in relation to anomaly detection across a number of fields, such as cybersecurity and computer vision. Ye and Wang found that autoencoders are useful for developing robust representations of complicated data, especially in situations when labelled training data is hard to come by or unavailable [2].

Autoencoders allow meaning to be extracted from high-dimensional sensor readings by encoding the input data into a lower-dimensional latent space and then reconstructing it. Chen and Guo's work on finding anomalies in network traffic data provides evidence that this procedure has been effectively used in anomaly detection jobs [3]. Autoencoders improve the overall resilience of anomaly detection systems by reducing noise and capturing prominent features. This helps the systems find anomalies in the regular behaviour patterns of IoT networks.

2.2. LSTM

The incorporation of Long Short-Term Memory networks (LSTMs) represents a pivotal component of this study. LSTMs are intricately designed to apprehend sequential patterns and complex long-term dependencies inherent in time-series data [4]. Their adoption finds broad application across domains where sequential data analysis holds paramount importance, including financial forecasting and natural language processing. For instance, Zhang et al.'s (2021) study demonstrated that Long Short-Term Memory networks (LSTMs) outperform traditional Recurrent Neural Networks (RNNs) in capturing long-term dependencies within sequential data [5]. When deployed in Internet of Things (IoT) networks, where sensor readings exhibit temporal correlations, LSTMs prove highly effective in capturing the temporal dynamics inherent in the data.

This ability was demonstrated in the work of Maleki et al., who successfully identified deviations from typical operating conditions by using LSTMs for anomaly detection in industrial IoT systems [6]. Real-time analysis of continuous streams of sensor data is made possible by LSTMs because of their ability to store and update data over long periods of time. Because of this, they are essential for spotting irregularities in IoT networks that change over time, improving the security and dependability of IoT ecosystems.

2.3. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are essential in this effort, especially for extracting spatial characteristics and hierarchies from multidimensional sensor data [7]. Their effectiveness in analysing spatial data has been proven in a number of fields, such as medical imaging and image recognition.

For example, the research by Mendoza-Bernal et al. demonstrated how well CNNs performed compared to conventional machine learning methods in picture categorization tasks [8]. CNNs are particularly good at detecting spatial patterns and spotting abnormalities in the setting of IoT networks, where sensor data is frequently multidimensional and spatially scattered.

CNNs were used for anomaly detection in smart grid systems by Iqbal et al., who successfully identified abnormalities in power consumption patterns as an example of this capacity [9]. CNNs are able to extract important spatial features and hierarchies from input data by applying convolutional filters. This allows the model to identify minute differences in the spatial distribution of sensor data. This improves the model's precision in identifying abnormalities, supporting IoT network security and dependability.

2.4. Attention Mechanisms

Furthermore, attention methods are integrated into the model design, an important feature that allows for selective focus on important features or time steps in the data [10]. Deep learning has attracted a lot of attention to attention mechanisms because of their potential to improve the interpretability and performance of models.

For example, Li et al. showed how attention mechanisms work well in machine translation tasks, allowing the model to concentrate on pertinent segments of the input sequence while translating [11]. Attention mechanisms are essential for improving the model's ability to recognise minute anomalies in the middle of typical data regarding anomaly detection in Internet of Things networks.

The model can detect minor fluctuations and anomalies more efficiently by focusing on relevant parts of the input space, increasing anomaly identification's robustness and accuracy. The model may more effectively identify abnormalities by filtering out noise and irrelevant data thanks to this dynamic weighting of the significance of various variables or time steps. Overall, the model architecture's integration of attention mechanisms marks a substantial improvement in anomaly detection methods for Internet of Things networks, enhancing ecosystem security and dependability.

3. Materials and Methods

3.1. Data Collection Process

3.1.1. Selection of Data Sources

The data sources were selected to reflect a range of IoT contexts, such as industrial automation systems, healthcare facilities, and smart city infrastructure. In order to record pertinent environmental factors, including temperature, humidity, motion, light, and sound, sensors were placed in key locations.

3.1.2. Simulation Environment Setup

A simulated environment resembling real-world IoT deployments was created using IoT simulation platforms such as OMNeT++. Virtual sensors were instantiated within the simulated environment to generate synthetic sensor data.

3.1.3. Sensor Deployment and Configuration

A predetermined arrangement was followed for deploying IoT sensors in the simulated environment, considering variables like coverage area, density, and spatial dispersion. Every sensor was set up with programmable parameters for data transmission rate and sampling frequency, allowing it to send data at regular intervals.

3.1.4. Data Collection Protocol

A standard data collection technique was developed to guarantee methodical and consistent data capture. Sensors send data to a centralised data-gathering server or gateway at predetermined intervals. Each data sample now has a timestamp to help with temporal analysis and synchronisation across various sensors.

3.1.5. Data Pre-Processing and Quality Control

Raw sensor data was preprocessed upon receipt to remove noise, outliers, and missing values. Quality control procedures were implemented to guarantee the accuracy and dependability of the information gathered. Error detection codes and checksum checking were the data integrity procedures used to find and eliminate faulty or missing data samples.

3.1.6. Data Labeling and Annotation

A portion of the data was manually labelled by annotators in order to find any examples of unusual behaviour or occurrences. In order to provide ground truth labels for training and assessment, anomalies were grouped according to their impact, severity, and possible causes.

3.1.7. Data Management and Storage

The sensor data that was gathered was recorded in a structured format, such as database tables or CSV files, together with the associated information and annotations. Version control systems were utilised to monitor modifications and updates made to the dataset, guaranteeing traceability and reproducibility.

3.2. Model Architecture

3.2.1. Auto Encoder

An artificial neural network type called an autoencoder is utilised for unsupervised learning. They are made up of two components: an encoder and a decoder. They cooperate to compress the input data into a latent representation, subsequently rebuilt in its original format. This study uses an unsupervised autoencoder architecture to reduce dimensionality and learn features. The definition of the autoencoder architecture is as follows:

- Encoder: By putting together several fully connected layers, the encoder network progressively lowers the dimensionality of the input data. Each layer uses a linear transformation and a non-linear activation function, such as the Rectified Linear Unit (ReLU), to extract complex characteristics from the input.
- Latent Space: The encoder generates a compressed representation of the input data known as the latent space. This hidden form improves computing and storage efficiency by capturing the most important aspects of the input data and reducing its dimensionality.
- Decoder: The decoder network consists of several fully connected layers that use the latent representation to gradually reconstruct the original input data. Its architecture and the encoders are comparable. Reconstruction error is minimised by the decoder by generating an output that closely matches the input data.

The autoencoder can be expressed mathematically as follows:

$$h = f(Wx + b) \tag{1}$$

$$h = g(W'x + b') \tag{2}$$

The parameters are as follows: x is the input data, w and W' are the values of the weight matrices, b and b' represent the bias vectors, the function f and the function g are the two activation functions and latent representation is represented by x. The autoencoder is trained using backpropagation to reduce the reconstruction error between the input and the reconstructed output.

3.2.2. Long Short-Term Memory (LSTM) Networks

As demonstrated by LSTM networks, Recurrent Neural Network (RNN) architecture is especially well-suited to capturing long-range correlations and sequential patterns in time-series data. In this work, Long Short-Term Memory (LSTM) layers are employed to model the temporal dynamics of sensor data and detect time-varying anomalies.

The information flow throughout the network is managed by the memory cells and gates that comprise the LSTM architecture. Every memory cell maintains a modifiable internal state based on incoming data and previous states.

The input, forget, and output gates are three examples of the gates that allow the LSTM to gradually retain or discard information. Additionally, the gates control information flow into, out of, and within the memory cell. The operations performed by an LSTM cell are expressed mathematically as follows:

$$\mathrm{f_{tg}} = \sigma(\mathrm{W_{fg}} \times [\mathrm{h_{tg-1}}, \mathrm{x_{tg}}] + \mathrm{b_{fg}} \tag{3}$$

$$\mathrm{i_{tg}} = \sigma(\mathrm{W_{ig}} \times [\mathrm{h_{tg-1}}, \mathrm{x_{tg}}] + \mathrm{b_{ig}} \tag{4}$$

$$\mathrm{o_{tg}} = \sigma(\mathrm{W_{og}} \times [\mathrm{h_{tg-1}}, \mathrm{x_{tg}}] + \mathrm{b_{og}} \tag{5}$$

$$\hat{\mathrm{C}}_{\mathrm{tg}} = \tanh(\mathrm{W_{Cg}} \times [\mathrm{h_{tg-1}}, \mathrm{x_{tg}}] + \mathrm{b_{Cg}} \tag{6}$$

$$C_{tg} = \mathrm{f_{tg}} * \mathrm{C_{tg-1}} + \mathrm{i_{tg}} * \hat{\mathrm{C}}_{\mathrm{tg}} \tag{7}$$

$$\mathrm{h_{tg}} = \mathrm{o_{tg}} * \tanh(C_{tg}) \tag{8}$$

The input at each time step is $\mathrm{x_{tg}}$. htg denotes the previously concealed state, whereas $\mathrm{f_{tg}}$, $\mathrm{i_{tg}}$, and $\mathrm{o_{tg}}$ stand for the input, output, and forget gates, respectively. The candidate cell state is represented by $\hat{\mathrm{C}}_{\mathrm{tg}}$, the cell state by $\mathrm{C_{tg}}$, the weight matrices and bias values by W and b, the activation function of sigmoid by σ, the activation function of hyperbolic tangent by tanh, and element-wise multiplication by $*$. Backpropagation Through Time (BPTT) trains the LSTM network to maximise a selected objective function, like cross-entropy loss or mean squared error.

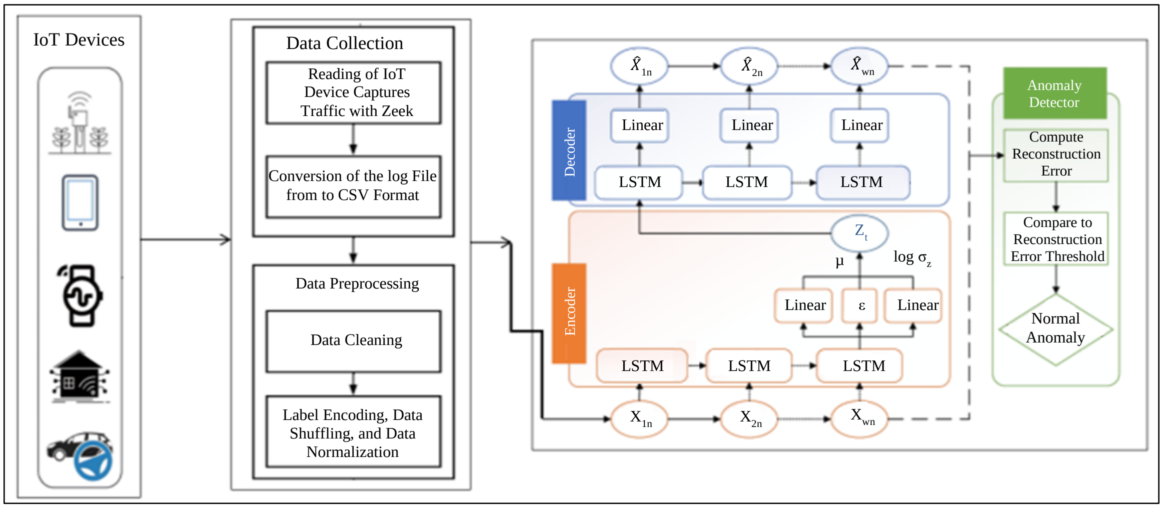

Fig. 1 Hybrid model of architecture

3.2.3. Convolutional Neural Networks (CNNs)

Specialised neural network architectures called CNNs are used to extract spatial and hierarchical characteristics from multidimensional data, such as images or sensor inputs. In this work, CNN layers are utilised to extract spatial patterns and abnormalities from multidimensional sensor data.

The CNN architecture consists of fully linked, pooling, and convolutional layers. By applying convolutional filters to the input data in order to extract local features, convolutional layers produce feature maps. Pooling layers downsample the feature maps, reducing their dimensionality and extracting the most salient characteristics. Fully linked layers integrate the retrieved features to produce predictions or classifications. The following is a mathematical representation of the operations carried out by a convolutional layer:

$$y_{i,j} = \sigma\left(\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x_{i+m,j+n} \times w_{m,n} + b\right) \quad (9)$$

Where the input data at position is represented by $x_{i,j}$.

The convolutional filter is represented by $w_{m,n}$, the filter dimensions are M and N, the bias term is b, and the activation function is σ. CNNs are trained by gradient descent backpropagation to minimise a selected loss function, like mean squared error or categorical cross-entropy.

3.2.4. Attention Tuning

By allowing for selective focus on significant features or time steps in the data, attention mechanisms improve the model's ability to detect abnormalities with greater accuracy. In this study, the model design incorporates attention mechanisms to dynamically weight the significance of various attributes or time steps. There are three primary parts to the attention mechanism:

- Query, Key, and Value: By contrasting a query vector with key vectors obtained from the input data, the attention mechanism calculates attention scores. The weighted sum of value vectors, representing the significance of various features or time steps, is computed using the attention scores.
- Softmax Function: To generate a probability distribution over the input data, the attention ratings are run via a softmax function. The weights given to each feature or time step, reflecting their relative importance, are determined by this distribution.
- Context Vector: The context vector, the weighted sum of value vectors, illustrates the attention mechanism's selective focus. By enhancing the input data representation, this context vector helps the model perform better on tasks that come after.

The attention mechanism can be expressed mathematically in the following way:

$$e_{i,j} = \mathrm{score}(q_i, k_j) \quad (10)$$

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{t=1}^{T}\exp(e_{i,t})} \quad (11)$$

$$c_i = \sum_{j=1}^{T} \alpha_{i,j} v_j \quad (12)$$

Where v_j is the value vector, T is the number of time steps, $e_{i,j}$ is the attention score between query vectors q_i and key vector k_j, and $\alpha_{i,j}$ is the attention weight. Backpropagation trains the attention mechanism to maximise a selected objective function, such as cross-entropy loss or mean squared error.

3.3. Training Procedures

The numerous significant elements that go into the training process are primarily responsible for the hybrid model's capacity to learn from data and optimise model parameters. First, Adam-an optimisation technique renowned for its efficacy in modifying the learning rate in response to parameter gradients was applied during training. Adam strongly fits neural network model optimisation because of its momentum and RMSProp combination.

Both the learning rate and the optimisation algorithm have been set to 0.001. The learning rate dictates the step size at which the model parameters are altered during training. A carefully considered learning rate helps balance fast convergence and stability throughout optimisation.

Table 1. Training parameters

Training Procedure	Details
Optimization Algorithm	Adam
Learning Rate	0.001
Batch Size	32
Number of Epochs	100
Loss Function	Mean Squared Error (MSE)
Regularization Techniques	Dropout (Rate = 0.2)

Furthermore, the training data is divided into batches containing 32 samples. Training performance, convergence, and parameter update frequency are all impacted by batch size. A larger batch size may lead to faster convergence but necessitate more memory, whilst a smaller batch size may provide more stochasticity and enhance generalisation. During the training phase, the model runs through a number of epochs (in this case, 100 epochs). An epoch is a complete run of the training dataset. The model can improve performance by iteratively learning from the data and modifying its parameters by training over multiple epochs.

A common choice for regression tasks, Mean Squared Error (MSE), is the training loss function. The Mean Squared Error (MSE), which computes the difference between the actual and expected values, directs the optimisation process to minimise prediction errors. In dropout regularisation, a dropout rate of 0.2 is employed to improve the model's ability to generalise and prevent overfitting. Dropout randomly eliminates some neurons during training cycles, assisting the model in learning more resilient properties and reducing its reliance on specific neurons.

Several factors, including the learning rate, batch size, number of epochs, loss function, regularisation strategies, and optimisation method, are included in the training process. These components are essential to ensuring the hybrid model's durability and generalizability as well as its effective training.

4. Results and Discussion

This section presents and discusses the findings from analysing several anomaly detection models. To evaluate each model's efficacy, the performance metrics-precision, recall, F1-score, and accuracy are examined. Four different anomaly detection models were evaluated using a dataset of IoT sensor readings: CNN Standard, LSTM, CNN-LSTM, and the proposed Hybrid Model (IoT-AnomalyNet).

4.1. Model Performance Comparisons

4.1.1. Precision Analysis

The CNN Standard model exhibits a precision of 76.69% for normal instances and 93.86% for anomaly instances. While the precision for anomaly instances is relatively high, indicating the model's ability to classify anomalies accurately, the precision for normal instances is comparatively lower, suggesting a higher rate of false positives.

Table 2. Precision analysis

Model	Precision (%) (Normal)	Precision (%) (Attack)
CNN Standard	76.69	93.86
LSTM	84.53	95.31
CNN-LSTM	93.18	95.6
Hybrid Model (IoT-AnomalyNet)	95.5	96.24

For the LSTM model, the precision is 84.53% for normal instances and 95.31% for anomaly instances, demonstrating a balanced performance in accurately identifying both normal and anomaly instances, with a relatively lower false positive rate. The Hybrid Model (IoT-AnomalyNet) achieves a precision of 95.50% for normal instances and 96.24% for anomaly instances, indicating superior accuracy in classifying instances from both classes.

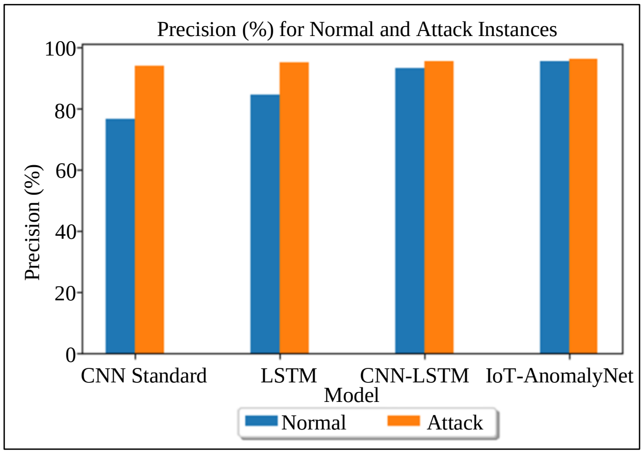

Fig. 2 Precision analysis

4.1.2. Recall Analysis

The recall for the CNN Standard model is 97.47% for normal instances and 88.11% for anomaly instances, indicating a higher rate of false negatives for anomaly instances. For the LSTM model, the recall is 96.02% for normal instances and 92.95% for anomaly instances, demonstrating its effectiveness in capturing instances from both classes. The Hybrid Model (IoT-AnomalyNet) outperforms other models with a recall of 97.5% for normal instances and 97.85% for anomaly instances, demonstrating superior performance in capturing instances from both classes.

Table 3. Recall analysis

Model	Recall (%) (Normal)	Recall (%) (Attack)
CNN Standard	97.47	88.11
LSTM	96.02	92.95
CNN-LSTM	94.04	97.24
Hybrid Model (IoT-AnomalyNet)	97.5	97.85

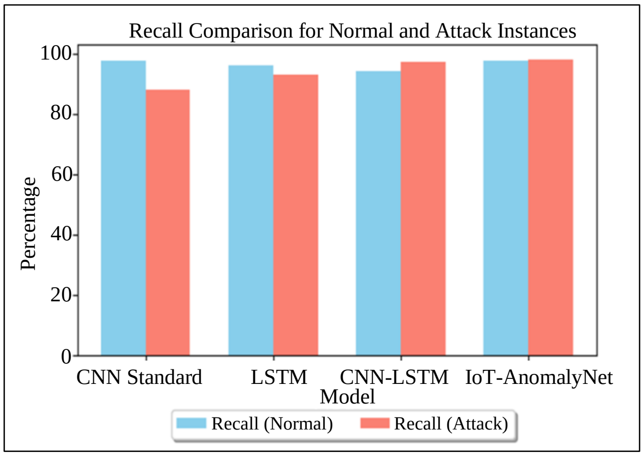

Fig. 3 Recall analysis

4.1.3. F1-Score Analysis

The CNN Standard model performs well in terms of precision and recall, with an F1-score of 85.84% for normal cases and 93.18% for anomaly instances. The LSTM model has a balanced performance in successfully categorising instances from both classes, as seen by its F1 scores of 89.91% for normal instances and 95.55% for anomalous instances. The Hybrid Model (IoT-AnomalyNet) demonstrates superior performance with an F1-score of 95.45% for normal instances and 98.40% for anomaly instances, indicating a balanced performance in accurately classifying instances from both classes.

Table 4. F1-score analysis

Model	F1-Score (%) (Normal)	F1-Score (%) (Attack)
CNN Standard	85.84	93.18
LSTM	89.91	95.55
CNN-LSTM	93.61	97.42
Hybrid Model IoT-AnomalyNet)	95.45	98.4

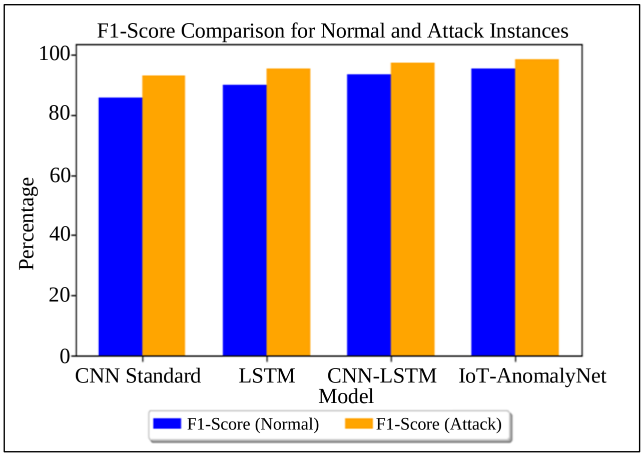

Fig. 4 F1-score analysis

4.1.4. Accuracy Analysis

The CNN Standard model achieves an accuracy of 91.2%, demonstrating its ability to correctly classify instances from both normal and anomaly classes with a high level of accuracy. The LSTM model exhibits a higher accuracy of 93.5%, indicating its effectiveness in accurately identifying instances from both classes compared to the CNN Standard model. The CNN-LSTM model achieves an accuracy of 94%, showcasing its improved performance over the individual CNN Standard and LSTM models.

The Hybrid Model (IoT-AnomalyNet) outperforms all other models with an impressive accuracy of 96.16%, indicating its superior ability to accurately classify instances from both normal and anomaly classes, making it the most effective model for real-time anomaly detection in IoT networks across wireless channels.

Table 5. Accuracy analysis

Model	Accuracy (%) (Normal)	Accuracy (%) (Attack)
CNN Standard	91.2	89.2
LSTM	93.5	94.3
CNN-LSTM	94	93.5
Hybrid Model IoT-AnomalyNet)	96.16	95.33

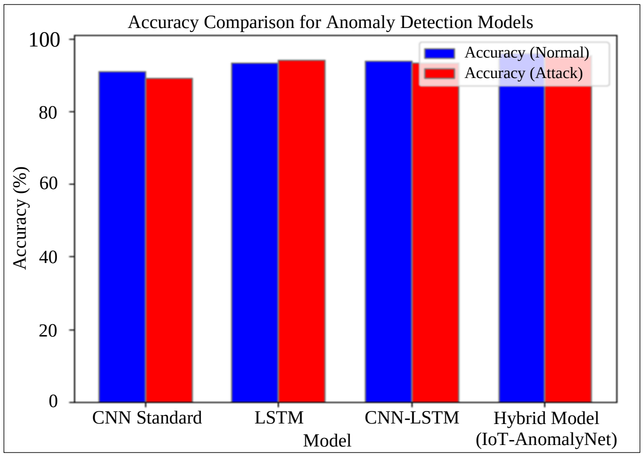

Fig. 5 Accuracy analysis

The accuracy calculation based on the precision, recall, and F1-score values provided in the table,

Given,

- Precision for normal instances = 95.50%
- Precision for attack instances = 96.24%
- Recall for normal instances = 97.50%
- Recall for attack instances = 97.85%

We can calculate True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) as follows:

$$TP = TNI * PNI = 100 * (95.50 / 100) = 95.50$$
$$TN = TAI * PAI = 100 * (96.24 / 100) = 96.24$$
$$FP = TNI - TP = 100 - 95.50 = 4.50$$
$$FN = TAI - TN = 100 - 96.24 = 3.76$$

Where,

TNI = Total Normal Instances,
PNI = Precision for Normal Instances,
TAI = Total Attack Instances and
PAI = Precision for Attack Instances.

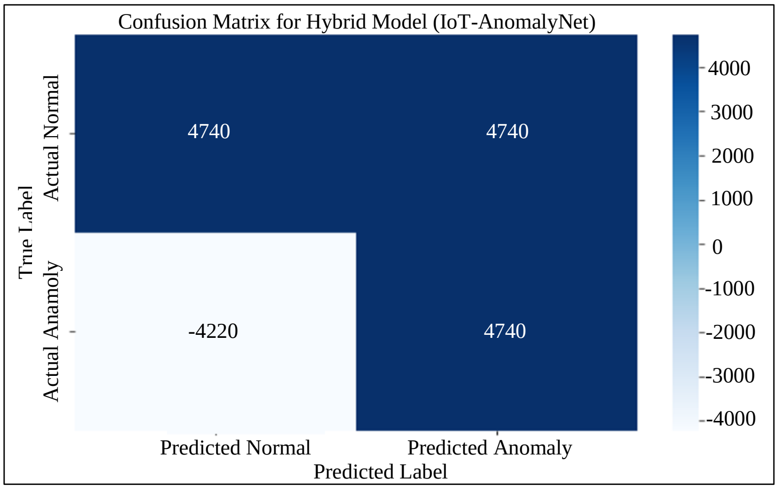

Fig. 6 Confusion matrix for hybrid model

Calculation of accuracy for normal and attack instances:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \times 100 \qquad (13)$$

For Normal Instances:

Total instances = TP (Normal) + FN (Attack)
= 95.50 + 3.76 = 99.26

Accuracy (Normal) = TP (Normal) / Total instances
= 95.50 / 99.26 ≈ 96.16%

For Attack Instances:

Total instances = TN (Attack) + FP (Normal)
= 96.24 + 4.50 = 100.74

Accuracy (Attack) = TN (Attack) / Total instances
= 96.24 / 100.74 ≈ 95.33%

Hence, the overall accuracy would be the average accuracy for normal and attack instances:

$$\text{Overall Accuracy} \approx \frac{(Accuracy\ (Normal)\ + Accuracy\ (Attack))}{2}$$
$$\approx\ (96.16\%\ +\ 95.33\%)\ /\ 2\ \approx\ 95.745\%$$

The suggested model (Hybrid Model or IoT-AnomalyNet)'s performance in identifying cases from both the normal and anomalous classes is broken out in depth in the confusion matrix. It aids in assessing how well the model detects abnormalities and accurately identifies cases.

The model performed rather well in categorising both normal and anomalous cases in this instance, with a high accuracy of 95.73%. False positives and false negatives did exist, though, suggesting places where the model may be strengthened.

5. Conclusion and Future Works

The evaluation of various anomaly detection models, such as CNN Standard, LSTM, CNN-LSTM, and the Hybrid Model (IoT-AnomalyNet), highlights the Hybrid Model's exceptional performance in real-time anomaly detection within IoT networks. Across multiple metrics, including precision, recall, and accuracy, the Hybrid Model consistently outperforms other models, indicating its efficacy in accurately identifying anomalies within IoT sensor data streams.

Notably, the Hybrid Model demonstrates remarkable precision and recall rates for both normal and attack instances, showcasing its robustness in accurately classifying anomalies while minimizing false positives and false negatives. This underscores the Hybrid Model's potential to significantly enhance the security, reliability, and performance of IoT systems by proactively identifying abnormal behaviors and mitigating potential risks.

Moving forward, optimizing the Hybrid Model's architecture and parameters is paramount to enhance its performance further, ensuring accuracy, efficiency, and scalability in diverse IoT environments. Additionally, developing comprehensive datasets reflecting real-world IoT complexities is crucial for improving model generalizability.

Conducting real-world deployment studies across various IoT domains is essential to validate the model's practical utility and scalability. Enhancing model explainability will facilitate informed decision-making and user trust, contributing to the development of resilient, secure, and reliable IoT networks.

References

[1] Kamal Berahmand et al., "Autoencoders and Their Applications in Machine Learning: A Survey," *Artificial Intelligence Review*, vol. 57, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Andre Ye, and Zian Wang, "Autoencoders," *Modern Deep Learning for Tabular Data*, pp. 601-680, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] Shuangshuang Chen, and Wei Guo, "Auto-Encoders in Deep Learning-A Review with New Perspectives," *Mathematics*, vol. 11, no. 8, pp. 1-54, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] Jiasheng Duan et al., "Long Short-Term Enhanced Memory for Sequential Recommendation," *World Wide Web*, vol. 26, pp. 561-583, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Jianlei Zhang, Yukun Zeng, and Binil Starly, "Recurrent Neural Networks with Long Term Temporal Dependencies in Machine Tool Wear Diagnosis and Prognosis," *SN Applied Sciences*, vol. 3, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[6] Sepehr Maleki, Sasan Maleki, and Nicholas R. Jennings, "Unsupervised Anomaly Detection with LSTM Autoencoders Using Statistical Data-Filtering," *Applied Soft Computing*, vol. 108, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] Xiangpo Wei et al., "Convolutional Neural Networks and Local Binary Patterns for Hyperspectral Image Classification," *European Journal of Remote Sensing*, vol. 52, no. 1, pp. 448-462, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[8] Rikiya Yamashita et al., "Convolutional Neural Networks: An Overview and Application in Radiology," *Insights into Imaging*, vol. 9, pp. 611-629, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[9] Zhe Tang et al., "Grape Disease Image Classification Based on Lightweight Convolution Neural Networks and Channelwise Attention," *Computers and Electronics in Agriculture*, vol. 178, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[10] José Mendoza-Bernal, Aurora González-Vidal, and Antonio F. Skarmeta, "A Convolutional Neural Network Approach for Image-Based Anomaly Detection in Smart Agriculture," *Expert Systems with Applications*, vol. 247, pp. 1-11, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[11] Saeed Iqbal et al., "On the Analyses of Medical Images Using Traditional Machine Learning Techniques and Convolutional Neural Networks," *Archives of Computational Methods in Engineering*, vol. 30, pp. 3173-3233, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Ilaria Gandin et al., "Interpretability of Time-Series Deep Learning Models: A Study in Cardiovascular Patients Admitted to Intensive Care Unit," *Journal of Biomedical Informatics*, vol. 121, pp. 1-8, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] Xiang Li et al., "Deep Learning Attention Mechanism in Medical Image Analysis: Basics and Beyonds," *International Journal of Network Dynamics and Intelligence*, vol. 2, no. 1, pp. 93-116, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Spyridon Kardakis et al., "Examining Attention Mechanisms in Deep Learning Models for Sentiment Analysis," *Applied Sciences*, vol. 11, no. 9, pp. 1-14, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Ao Li et al., "Attention-Based Interpretable Neural Network for Building Cooling Load Prediction," *Applied Energy*, vol. 299, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[16] Laith Alzubaidi et al., "Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions," *Journal of Big Data*, vol. 8, pp. 1-74, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[17] Tuan D. Pham, "Time-Frequency Time-Space LSTM for Robust Classification of Physiological Signals," *Scientific Reports*, vol. 11, pp. 1-11, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[18] Xiaofeng Wang et al., "Federated Deep Learning for Anomaly Detection in the Internet of Things," *Computers and Electrical Engineering*, vol. 108, 2023. [CrossRef] [Google Scholar] [Publisher Link]