SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I8P116

Volume 11 Issue 8, 151-159, August 2024

Original Article

Anomaly Detection in IoT Sensor Data Using Auto Encoder-Based Unsupervised Learning

Kusuma Shalini[1], Anvesh Thatikonda[2]

[1,2]*ECE, Chaitanya, Deemed to be University, Telangana, India.*

[1]*Corresponding Author : kusumashalini9@gmail.com*

Received: 09 June 2024 Revised: 23 July 2024 Accepted: 11 August 2024 Published: 31 August 2024

Abstract *- In recent years, automated systems have emerged, and these automated systems should take the data through their sensors and identify abnormal patterns called anomalies. Anomaly is an abnormal pattern in sequence data, like malfunctions, hazards, etc., in sequence data. By reading this data continuously from time to time, the model learned the different patterns, such as regular and abnormal, and separated the abnormal patterns. Many researchers have worked on this, using data like environment, industry, etc., and standard pattern identification methods to deep learning models like LSTM. This paper presents a novel approach to detecting anomalies in IoT sensor data, including time, temperature, etc., and trains an unsupervised auto-encoder model to predict anomalies at various threshold levels. Moreover, we got the 0.0004720 mean square error, at this level, the data is reconstructed.*

Keywords - *IoT sensor data, Anomaly detection, Unsupervised learning, Autoencoder, Deep learning.*

1. Introduction

In recent years, automated systems (IoT) have been used everywhere to provide security, detect faults, and identify malfunctions. The systems will be used anywhere, such as the environment, networks, IOT systems, etc. So deploying IoT automated systems is increasing day by day, and the role of these systems is also increasing, like automatic fault, malfunction, and abnormal things identification. These systems continuously capture the data irrespective of time and other factors, so a huge amount of data is created per second. It is impossible for human beings to detect and identify abnormal data, also called anomaly.

Anomaly detection is crucial to ensure reliability, security, and other factors. Traditional methods used pattern recognition methods to detect anomalies. With the emergence of Artificial Intelligence (AI), automated detection has increased. With AI, the consistency and reliability of the system are increased.

So, by using AI, we can identify abnormal data. Many researchers worked on this like[20] proposed a normal probability-based approach to detect the anomalies. But over the years, the research moved from probability-based to deep learning-based methods. The researchers like [21] implemented simple clustering methods to segregate the abnormal data, and some have used outlier detection methods to find different data points. Some researchers like [2, 3 and 11] used the deep learning model LSTM and trained a huge amount of data to find anomalies.

The main challenges are a lack of fault data and no supervised or target values to train. So, these data should be clearly observed from the beginning and identify the different patterns, which are not uniquely distributed with the maximum amount of data.

1.1. Contribution

- Proposed a deep auto-encoder approach to detect the anomalies from IoT sensor data.
- Found the anomalies at different threshold levels and detected anomalies at each feature.
- Presents the detailed result analysis and interpretation.

2. Related Work

Anomaly detection is crucial in various domains to identify abnormal patterns indicating system malfunctions, faults, or security breaches. With the advent of deep learning, auto-encoder-based methods have gained popularity for their ability to learn complex data representations and effectively detect anomalies. Most researchers worked on sensor data and used simple, unsupervised learning methods. Like [16 and 20] worked on environmental sensor data. The first proposed a Bayesian method to detect anomalies, like hazards or sensor malfunctions, by continuously updating probability distributions based on incoming data. The other proposed a data-driven modeling approach for anomaly detection in

streaming environmental sensor data. This method will differentiate normal and abnormal patterns in historical data and identify the anomaly.

The proposed a method that first separates the normal and abnormal data by constructing a contextual lattice for anomaly detection in monitoring [19]. They used sequence sensor data, and the proposed method detected anomalies indicative of potential equipment malfunctions or degradation by analyzing historical data and learning standard patterns.

Hayes and Capretz [18] worked on wireless sensor data to detect abnormal patterns by analyzing the context of surrounding data or an environment. They used the Outlier detection method to separate the abnormal data. O'Reilly et al. [21] proposed a clustering method; they clustered data within the sensor range and used wireless sensor networks in a static environment. The data which do not belong to any cluster at the point is treated as an anomaly, and makes all the nodes into clusters. In [17], used time series data, implemented box plots with all data combinations, and then used a concept of identifying outliers to detect anomalies. The proposed method for anomaly detection is based on sensor data in petroleum industry applications. The proposed method detected anomalies indicative of potential equipment malfunctions or process deviations by analyzing various sensor data, including temperature, pressure, and flow rate.

Fan et al. in [1] proposed an auto-encoder-based method for unsupervised anomaly detection in building energy data. The model explored the effectiveness of autoencoders in identifying anomalous patterns within energy consumption data. This method detected anomalies based on the comparison; they compared actual input with constructed input. Chen et al. [5] and [6], [12] worked on network data of IoT systems in that the anomalies are like detecting traffic and control flow. They used auto-encoder models and reconstructed the data.

Provotar et al. [2] proposed an unsupervised deep learning model that is Long-Short-Term Memory (LSTM) with an encoder and decoder structure to detect anomalies. Using two types of time series data, artificial signals and rare sound detection, this model neglects noise samples and can detect temporal dependencies in the given x.

Bae [10] worked on general security data to find security issues. Trained unsupervised learning models like DBSCAN and KDD models to capture the anomalies. It also works for intrusion scoring for intelligent factory environments.

Ahmad et al. [3] implemented an auto encoder-based deep learning method for detecting anomalies in rotating machines. This method learns patterns of rotating machine normal vibrations and abnormal vibrations. To detect abnormal vibrations, a fixed threshold value is used, and reconstructed the data. For this approach, the two machines data and got an F1-score of 99.6%.

Adkisson et al. [4] implemented an auto encoder-based approach to detect anomalies in IoT-based smart farming ecosystems. To detect any attacks on the hardware or any networking attacks on the device. Smart farming with IoT includes soil moisture, temperature, and crop health detection using various sensors. These sensors capture huge amounts of data per second. This deep learning based system will identify different patterns and find the anomalies.

Lee et al. in [7] proposed a convolution network model with an autoencoder to detect anomalies in gas turbines. And they also compared reconstructed data with isolation forest and k-means clustering algorithm. This method will find the abnormalities in the gas turban machine. [8] proposed an artificial neural network method for finding abnormal signal vibrations in rolling bearings. This method uses variable cumulative reconstruction error to detect the anomalies. Because mechanical objects will work at different speeds, this identification of errors is difficult. In their model, they used convolution layers and a weighted normalization approach.

Nazir et al. [9] introduced an auto encoder-based anomaly detection method for SCADA networks. The proposed method detected anomalies indicative of potential cyber threats or network intrusions by analyzing network traffic data.

Muneer et al. [11] introduced a novel hybrid deep autoencoder neural network approach on the gas turban dataset. Before training this high dimensional data, they balanced the classes by using up sampling method. And used different optimizers like gradient, Adam, etc, and fine tuned the parameters. And the model performed well, with an accuracy of 0.99. This anomalies are detected by reducing the high dimensional data to low dimensional data. In [11] detected anomalies in SCADA networks, the anomalies are like cyber attacks, and network flow. For this, they use LSTM autoencoders using statistical data filtering by combining LSTM autoencoders with statistical data filtering techniques. They also compared the results with existing models. Zhang et al. [13] implemented a deep conventional auto-encoding method to detect multi-sensor time-series signals. They used a bidirectional LSTM model to train and reconstruct the original data after detecting anomalies.

Chen [14] worked on robot data, that is, anomaly detection in the robots. For that, they implemented an unsupervised model for anomaly detection in industrial agent sensor data using a sliding-window convolution variation. The model detected anomalies indicative of potential faults or malfunctions in industrial robots by analyzing sequential sensor data. This approach contributes to the safety and productivity of industrial operations. Hu et al.[15]

implemented an LSTM model, it works and identifies the long term dependencies to find the normal and other behavior of input data; they selected the threshold value with the kernel density estimation method. The average more on train and test data is 0.026 and 0.035. The dataset used to train these models is from power generation systems.

3. Materials and Methods

This paper presents an auto-encoder model, as shown in Figure 1. The model can capture abnormal patterns. Moreover, a regression model was used to predict the error. The model consists of an encoder and a decoder part. As per Equation (1), five layers are present in the encoder with 256 input units; each layer is a normalized batch. The normalized method will improve the training process by normalizing the data at each layer. And the dropout rate is used as 0.1, and the next layer has 128 units, 64, 32, and 16 units with Leaky RELU activation function. In this model, 0.1% of parameters are dropped; after each epoch, the remaining parameters will be updated with an Adam optimizer. On the Decoder side, as per Equation (2), it takes the output of the encoder to 32 unit layer, which is passed to 64 unit layer, then 128, and finally, 256 units will give the output and identify the anomalies.

$$h = f(w_e x + b_e) \quad (1)$$

$$\hat{x} = f(w_d h + b_d) \quad (2)$$

*where x^ is the reconstructed output x is the original data

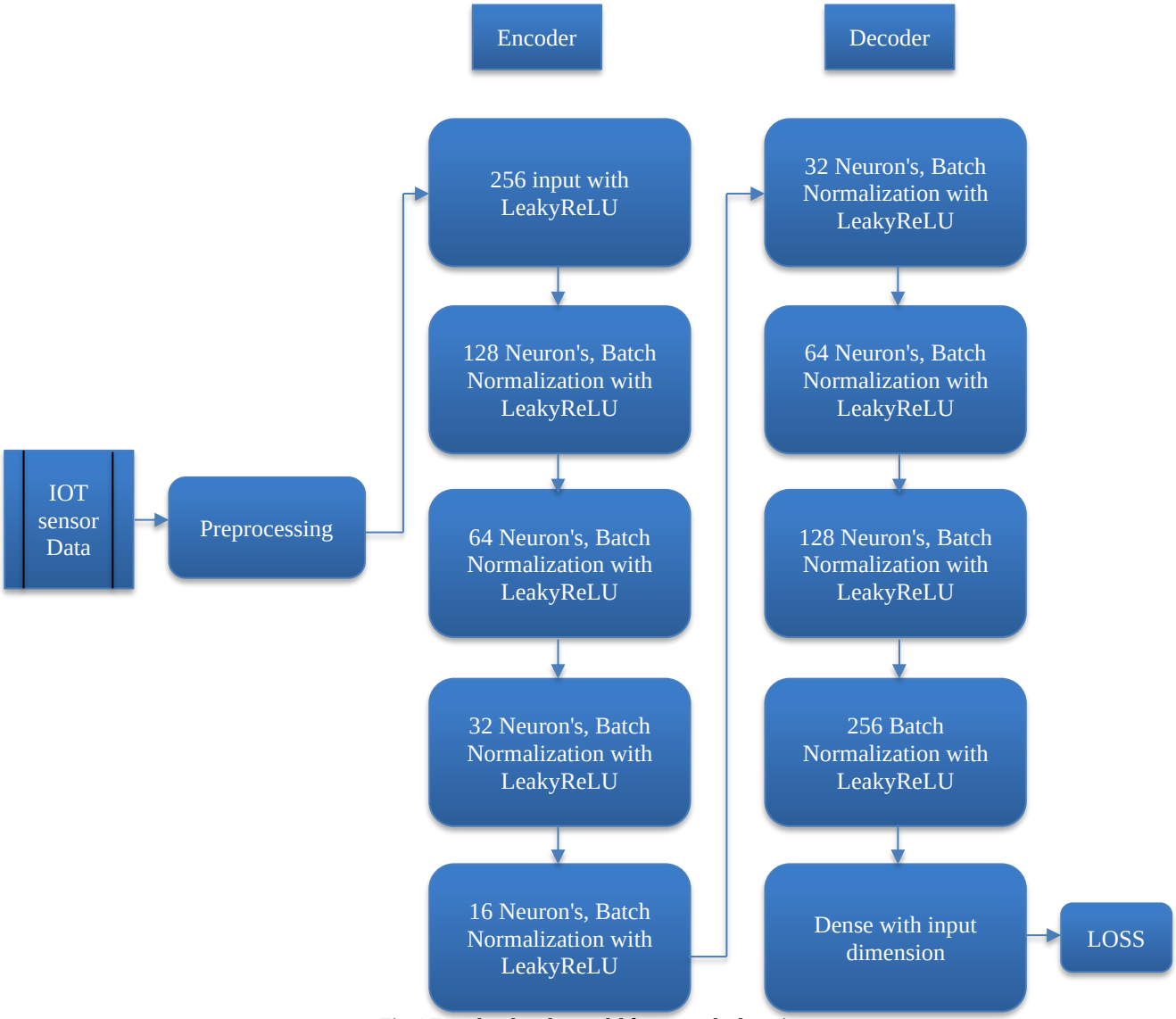

Fig. 1 Encoder-decoder model for anomaly detection

The image depicts an autoencoder neural network designed for processing IoT sensor data. It consists of:

1. Input: IoT sensor data.
2. Preprocessing: Prepares data for the encoder.
3. Encoder:
   - 5 layers: 256 → 128 → 64 → 32 → 16 neurons.
   - Each layer uses batch normalization and Leaky ReLU activation.
4. Decoder:
   - Mirrors the encoder with 4 layers: 32 → 64 → 128 → 256 neurons.
   - Ends with a dense layer to reconstruct the input dimension.
5. Loss Function: Measures the difference between the input and reconstructed output.

The autoencoder is likely used for tasks like anomaly detection or data compression, learning a compact representation of the input data.

3.1. Data Set

We used IoT sensor data, which consists of time, temperature, humidity, air quality, light, and loudness. First time is converted to numerical data, and then all the features are normalized with minmax scaling with Equation (4), so the range of scaled data is 0 to 1. In total, the data consists of 6558 rows, which are split into 75% data for training and 25% for testing. Figure 2 illustrates the correlation between all the features; from this, it is observed that all the features are normally correlated. Figure 3 illustrates the distribution of all features like minimum and maximum range, and it is helpful to find the outliers in the data; based on this, we identified outliers in temperature, humidity, and sound, and we removed all outliers from the data and also removed null values. Out of all the features, only the air quality feature is without outliers, so the remaining features are the anomalies removed with a standard normalizing approach.

$$Min\_Max(X_i) = \frac{x - \min(X)}{Max(x) - Min(x)} \quad (4)$$

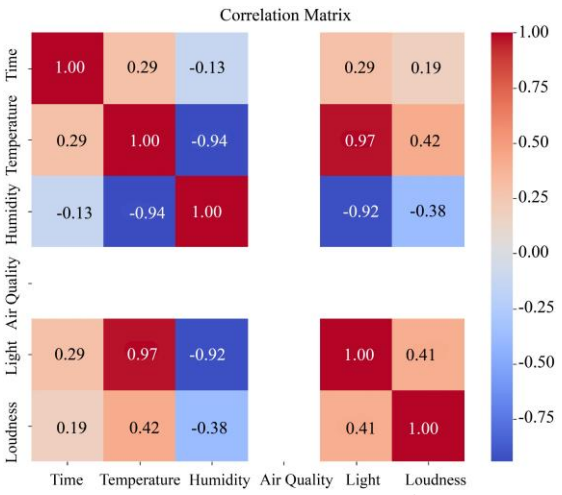

Fig. 2 Correlation between the features

4. Result Analysis

The proposed auto-encoder model is trained for 100 epochs by using a learning rate of 0.0001, and to change the weights, an Adam optimizer is used. It is an unsupervised learning model that finds errors. It uses MSE to calculate the loss of the model at each epoch. From Figure 5, the training and validation errors are consistently reduced to their minimum level. Training and validation loss are overlapped, so it is necessary to fit training the model; the essential features were identified; as shown in Figure 4, it is observed that all the features are above 0.05, humidity being the most crucial feature, with 0.08.

The performance of the model is evaluated with Mean Square Error, Mean Absolute Error, and R2 error concerning Equations (5),(6) and (9).

The threshold values are changed to find the anomalies and MSE, MAE, and R2 errors, as shown in Table 1. We selected the threshold value as 0.9. At this threshold value, we reconstructed the original data with updated weights using Equations (1) and (2); Figure 6 illustrates the detected anomalies of all features at a threshold value of 0.9. This graph illustrates sensor reading over the time 0 to 6000 units, in this all feature behavior and anomaly places. The anomalies are found at 0 to 1000 time scale, but after that, they reduced. Figure 7 illustrates the original and reconstructed data. At a threshold level of 0.9, we found 1312 anomalies out of 6558 data samples, and these samples were reconstructed, as shown in Figure 7.

Table 1. Error of proposed model after reconstructing data at different threshold values

Threshold	MSE	MAE	R2
0.5	0.0005281356581428829	0.014958536869492625	0.788817960701202
0.9	0.0004720127367610996	0.013937838938940 82	0.795417589164942 2
1.5	0.0005222356581829389	0.014959127346492625	0.786432856070120 2

In Figure 8, at threshold level 0.9, more anomalies are identified on temperate features. After that feature is reconstructed, the number of anomalies is reduced. With manifold application, all the high-dimensional data is represented in 2D space in Figure 9. From this, it is observed that the data is consistently distributed. We also constructed 3 to 4 clusters to map all the features and anomalies into the clusters and optimize weights at MSE of 0.00047201273677610996. It also reconstructed and compared the anomaly data with the original data. Our study identified 1312 samples as defective and used 6558 samples for training. The model was trained for 100 epochs, during which we observed a significant reduction in loss. Our results demonstrate the effectiveness of the deep auto-encoder

method in detecting anomalies in large volumes of IoT sensor data while also achieving near-perfect data reconstruction.

5. Conclusion

This paper utilized the deep auto-encoder method to detect the image showing a series of line graphs displaying data over time for five different environmental factors: Temperature, Humidity, Air Quality, Light, and Loudness.

1. Temperature: This graph shows fluctuations between 20 and 70 units (likely degrees, but the unit is not specified). There are significant spikes and drops, indicating varying temperature levels over time.
2. Humidity: This graph indicates humidity levels ranging between 30 and 60 units, with periods of relative stability interspersed with sharp changes.
3. Air Quality: The Air Quality graph is nearly flat, showing minimal variation around 76-78 units. This suggests that the air quality remained constant during the observed period.
4. Light: The Light graph fluctuates between 630 and 670 units, with distinct peaks, indicating varying light intensity over time.
5. Loudness: This graph shows significant variability in loudness levels, fluctuating broadly between 0 and 400 units. There are frequent peaks and valleys, suggesting a dynamic noise environment.

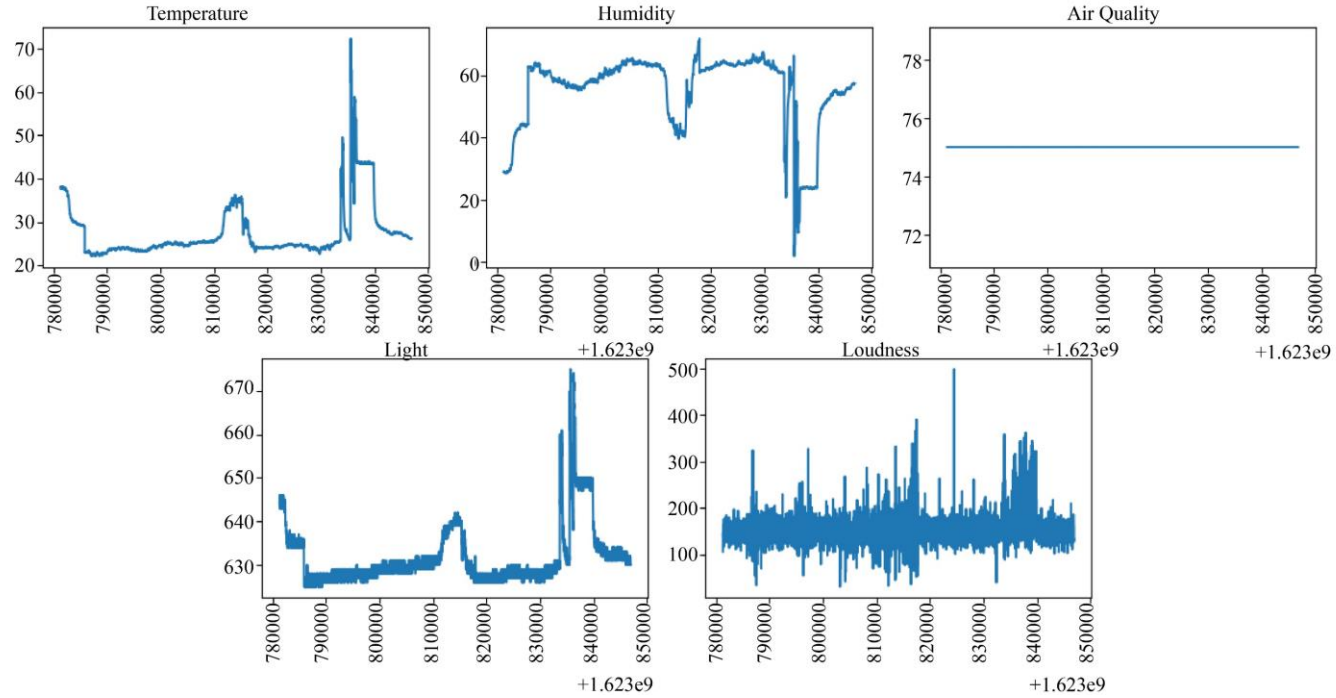

Fig. 3 Density graphs of all the features concerning time

$$MSE/SSE(Sum\ of\ Squares = \frac{1}{number\ of\ samples}\left(actual_{target} - predicted\_target\right)^2 \quad (5)$$

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST} \quad (6)$$

$$Sum\ of\ squares\ total\ (SST) = \sum \left(actual_{target} - predicted_{target}\right)^2 \quad (7)$$

$$Sum\ of\ Square\ regression\ (SSR) = \sum \left(predicted_{target} - avg(predicted_{target}\right)^2 \quad (8)$$

$$MAE = \frac{1}{number\ of\ samples}\left\|X_{actual} - X_{predicted}\right\| \quad (9)$$

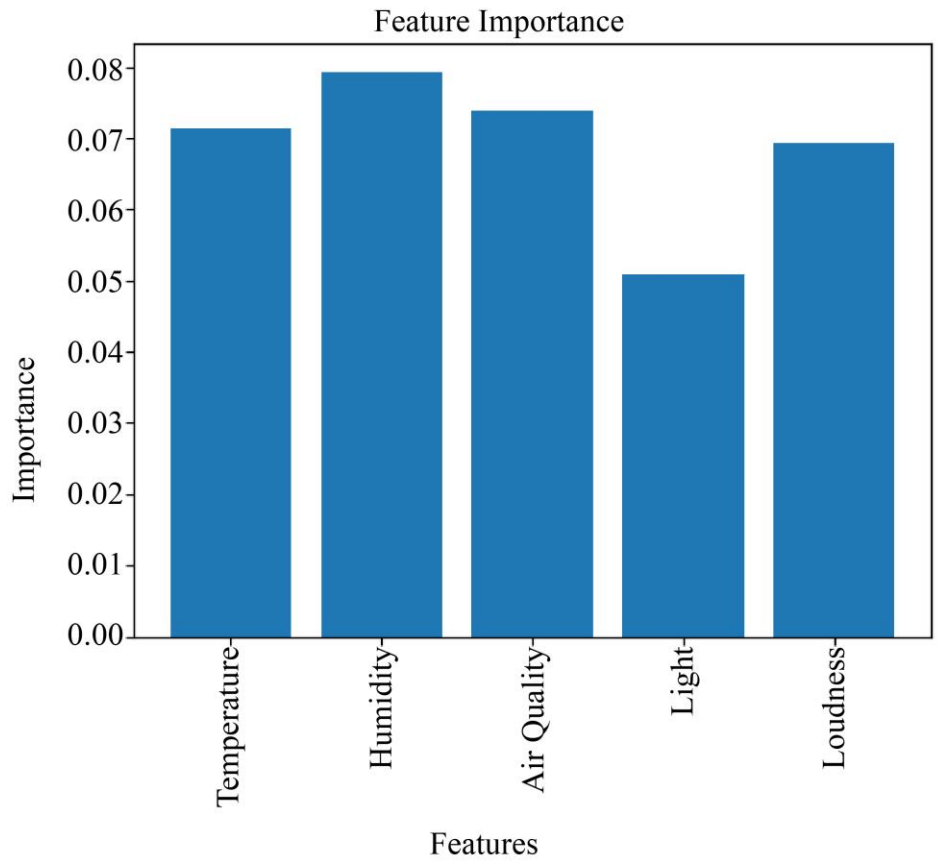

Fig. 4 Features importance graph expect time

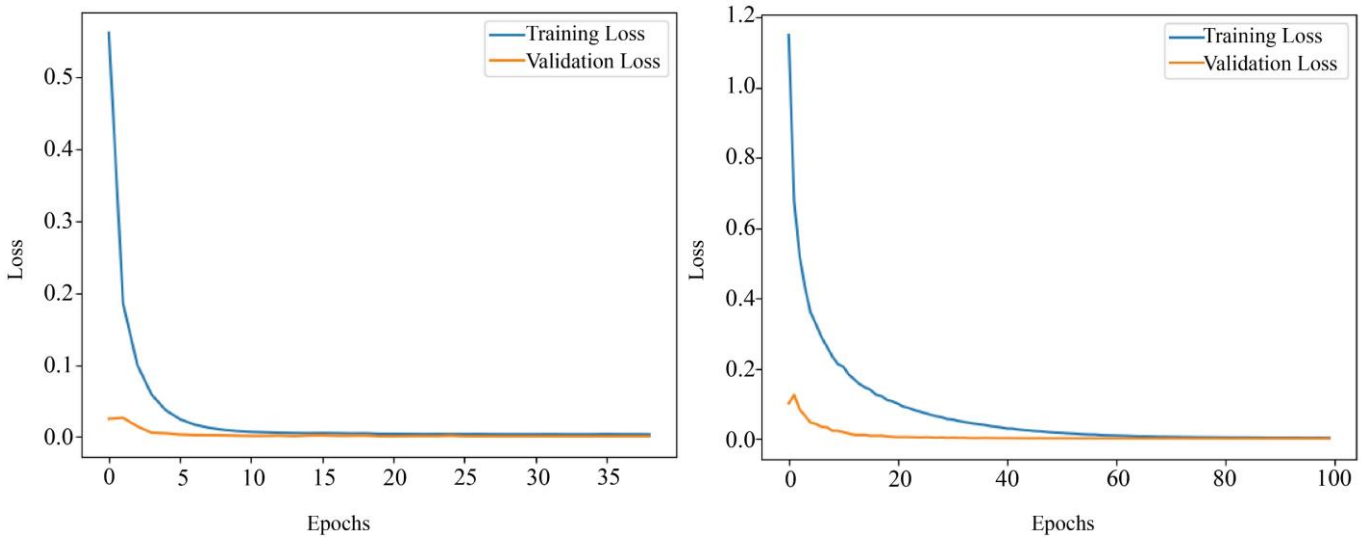

Fig. 5 Training and validation loss with early stop and without early stop

- The behavior in both graphs suggests that the model is learning effectively as both training and validation losses decrease and stabilize.
- The quick convergence of the validation loss in both cases indicates that the model generalizes well to the validation data.
- The left graph represents a quicker convergence compared to the right graph, possibly due to fewer epochs or a smaller, simpler model.

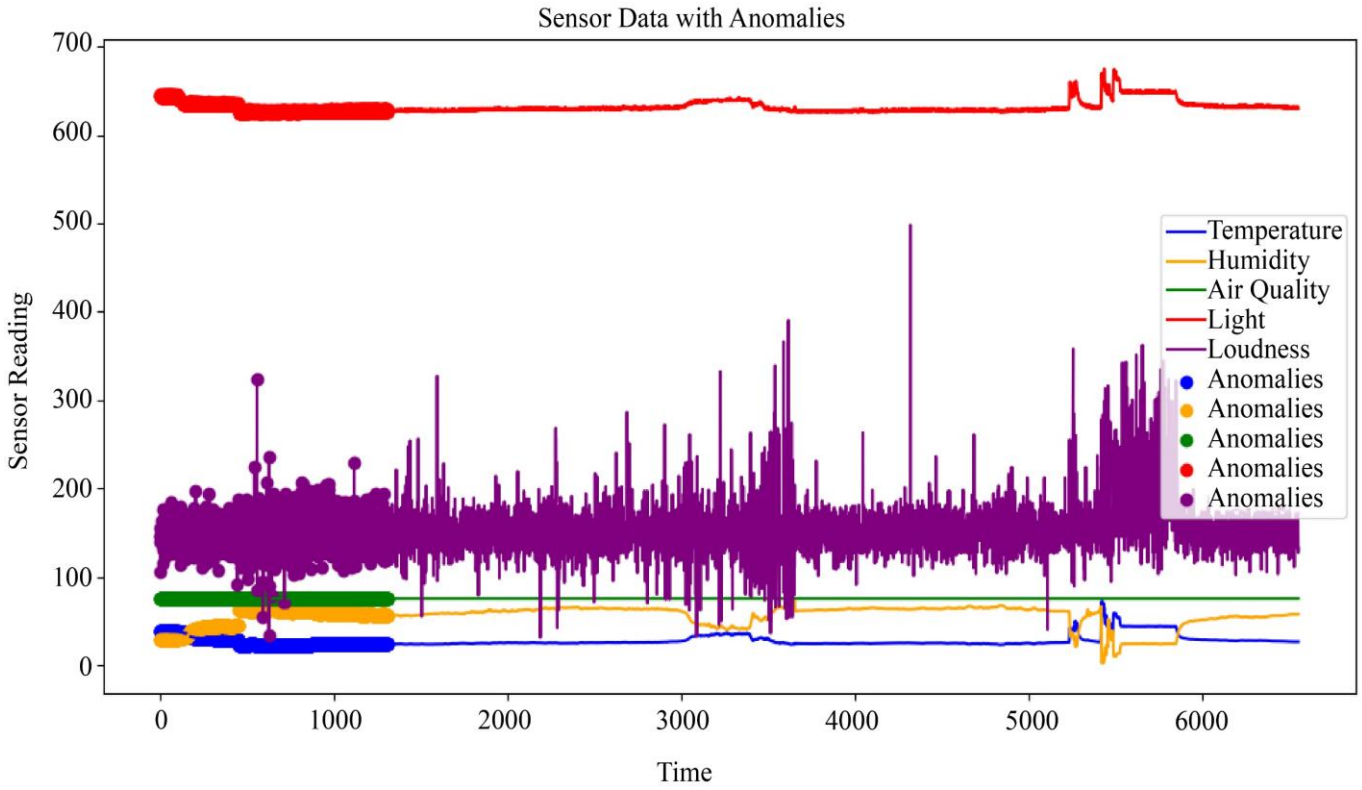

Fig. 6 Detected anomalies in different features

- The graph visualizes multiple sensor readings, showing how each behaves over time.
- Loudness appears to be the most unstable, with frequent and significant anomalies.
- Temperature, Humidity, Air Quality, and Light show more stability, with fewer anomalies.
- This type of visualization helps in monitoring and detecting unusual patterns in sensor data, crucial for applications like anomaly detection in IoT systems.

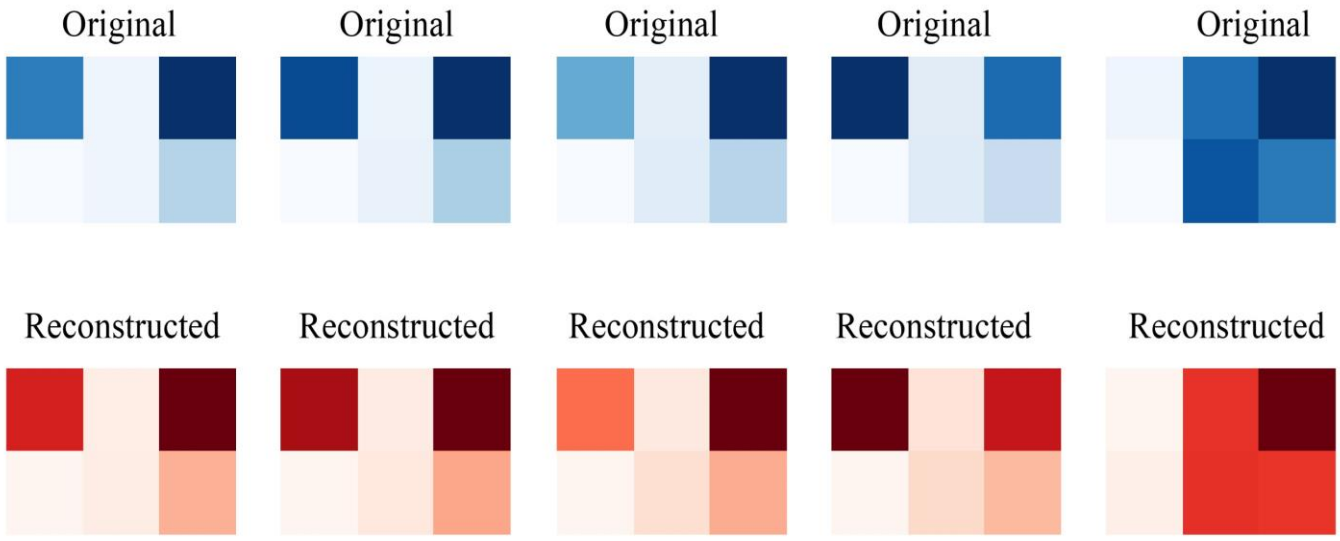

Fig. 7 Original and reconstructed data

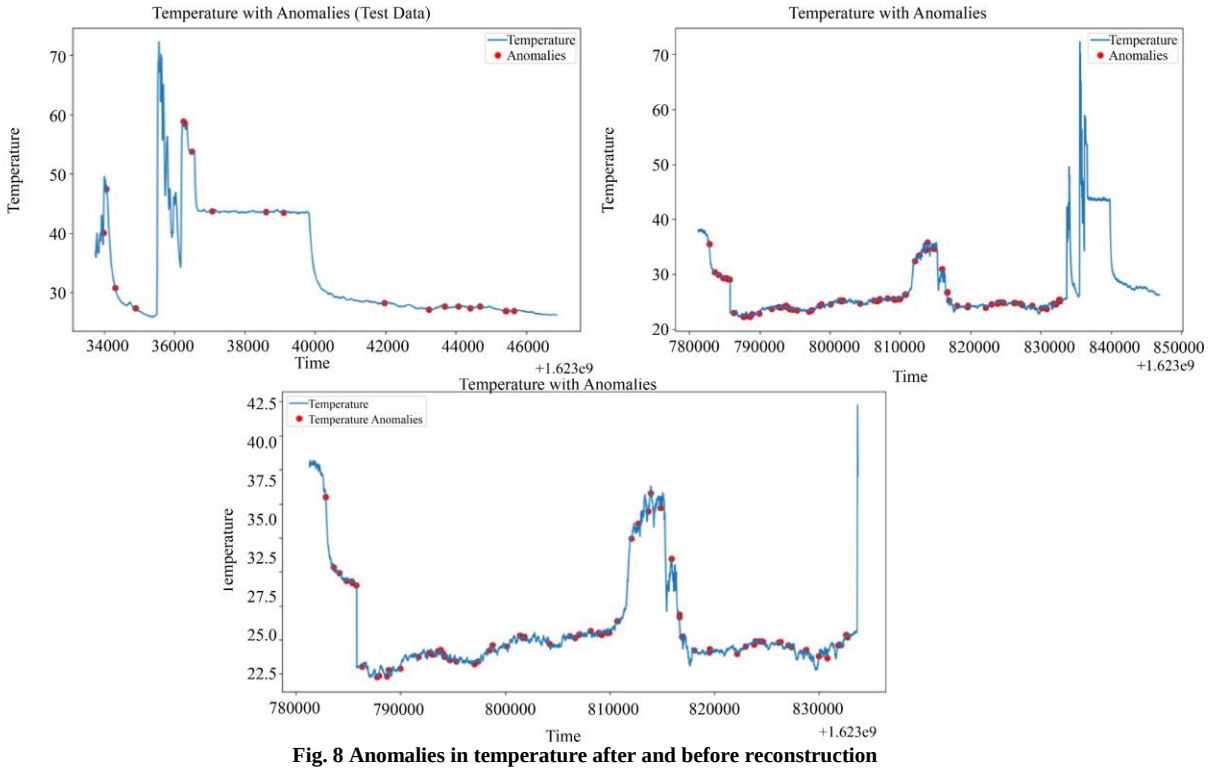

Fig. 8 Anomalies in temperature after and before reconstruction

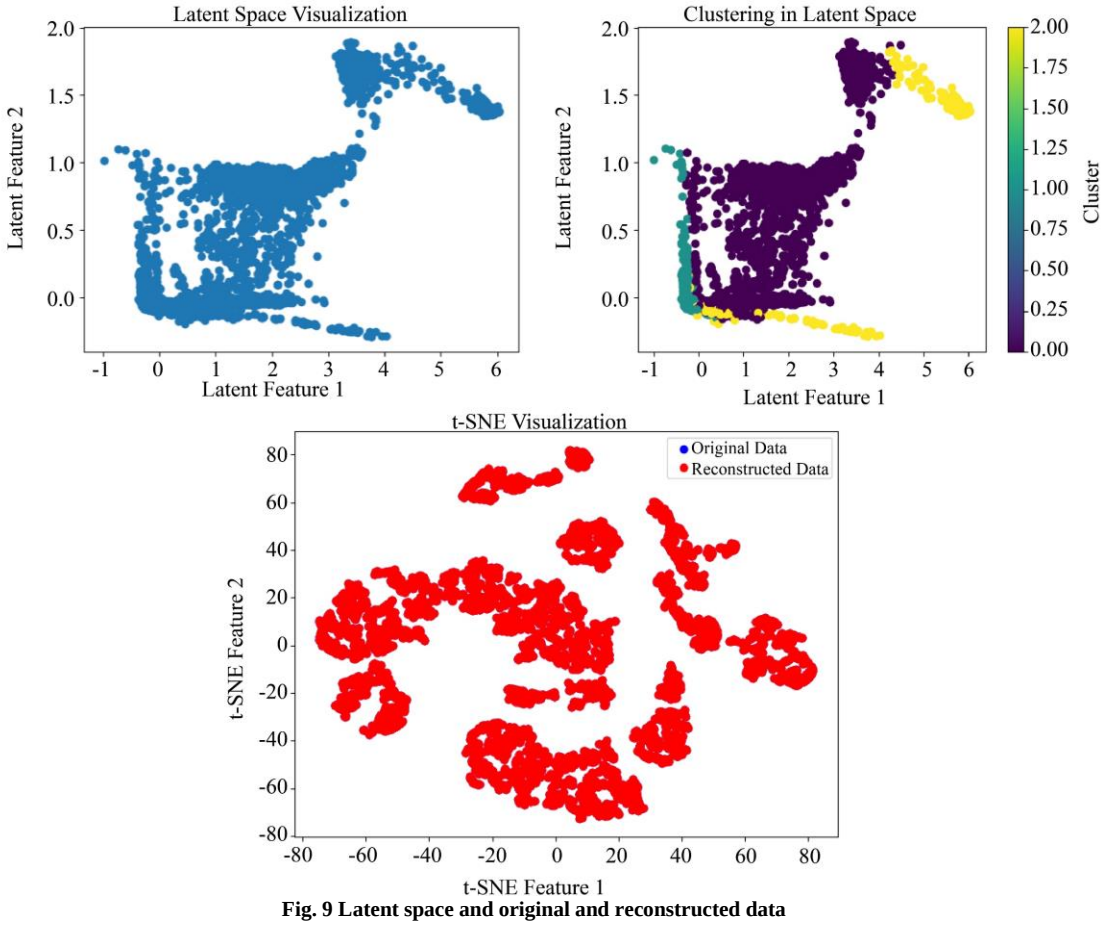

Fig. 9 Latent space and original and reconstructed data

References

[1] Cheng Fan et al., "Analytical Investigation of Autoencoder-Based Methods for Unsupervised Anomaly Detection in Building Energy Data," *Applied Energy*, vol. 211, pp. 1123-1135, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[2] Oleksandr I. Provotar, Yaroslav M. Linder, and Maksym M. Veres, "Unsupervised Anomaly Detection in Time Series Using LSTM-Based Autoencoders," *IEEE International Conference on Advanced Trends in Information Theory*, Kyiv, Ukraine, pp. 513-517, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[3] Sabtain Ahmad et al., "Autoencoder-Based Condition Monitoring and Anomaly Detection Method for Rotating Machines," *2020 IEEE International Conference on Big Data*, Atlanta, GA, USA, pp. 4093-4102, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Mary Adkisson et al., "Autoencoder-based Anomaly Detection in Smart Farming Ecosystem," *IEEE International Conference on Big Data*, Orlando, FL, USA, pp. 3390-3399, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Zhaomin Chen et al., "Autoencoder-Based Network Anomaly Detection," *Wireless Telecommunications Symposium*, Phoenix, AZ, USA, pp. 1-5, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[6] Tie Luo, and Sai G. Nagarajan, "Distributed Anomaly Detection Using Autoencoder Neural Networks in WSN for IoT," *IEEE International Conference on Communications*, Kansas City, MO, USA, pp. 1-6, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[7] Geunbae Lee et al., "Unsupervised Anomaly Detection of the Gas Turbine Operation via Convolutional Auto-Encoder," *IEEE International Conference on Prognostics and Health Management*, Detroit, MI, USA, pp. 1-6, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] Zhiyuan Li et al., "Unsupervised Machine Anomaly Detection Using Autoencoder and Temporal Convolutional Network," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1-13, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Sajid Nazir, Shushma Patel, and Dilip Patel, "Autoencoder Based Anomaly Detection for SCADA Networks," *International Journal of Artificial Intelligence and Machine Learning*, vol. 11, no. 2, pp. 83-99, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Gimin Bae et al., "Autoencoder-Based on Anomaly Detection with Intrusion Scoring for Smart Factory Environments," *Parallel and Distributed Computing, Applications and Technologies*, Jeju Island, South Korea, pp. 414-423, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[11] Amgad Muneer et al., "A Hybrid Deep Learning-Based Unsupervised Anomaly Detection in High Dimensional Data," *Computers, Materials & Continua*, vol. 70, no. 3, pp. 5363-5381, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[12] 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]

[13] Yuxin Zhang et al., "Unsupervised Deep Anomaly Detection for Multi-Sensor Time-Series Signals," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 2, pp. 2118-2132, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Tingting Chen et al., "Unsupervised Anomaly Detection of Industrial Robots Using Sliding-Window Convolutional Variational Autoencoder," *IEEE Access*, vol. 8, pp. 47072-47081, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] Di Hu et al., "Anomaly Detection of Power Plant Equipment using Long Short-Term Memory Based Autoencoder Neural Network," *Sensors*, vol. 20, no. 21, pp. 1-18, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[16] David J. Hill, and Barbara S. Minsker, "Anomaly Detection in Streaming Environmental Sensor Data: A Data-Driven Modeling Approach," *Environmental Modelling & Software*, vol. 25, no. 9, pp. 1014-1022, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[17] Luis Martí et al., "Anomaly Detection Based on Sensor Data in Petroleum Industry Applications," *Sensors*, vol. 15, no. 2, pp. 2774-2797, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[18] Michael A. Hayes, and Miriam A.M. Capretz, "Contextual Anomaly Detection in Big Sensor Data," *IEEE International Congress on Big Data*, Anchorage, AK, USA, pp. 64-71, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[19] Julien Rabatel, Sandra Bringay, and Pascal Poncelet, "Anomaly Detection in Monitoring Sensor Data for Preventive Maintenance," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7003-7015, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[20] David J. Hill, Barbara S. Minsker, and Eyal Amir, "Real-Time Bayesian Anomaly Detection for Environmental Sensor Data," *Proceedings of the Congress-International Association for Hydraulic Research*, vol. 32, no. 2, 2007. [Google Scholar]

[21] Colin O'Reilly et al., "Anomaly Detection in Wireless Sensor Networks in a Non-Stationary Environment," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1413-1432, 2014. [CrossRef] [Google Scholar] [Publisher Link]