Thesis for the Degree of Master

A Hybrid Approach of ConvLSTM-DT and GPT-4 for Real-time Anomaly Detection Decision Support in Edge-Cloud Environments

엣지-클라우드 환경에서 실시간 이상 탐지 결정 지원을 위한 ConvLSTM-DT 와 GPT-4 의 하이브리드 접근 방식

Department of Electronics Engineering

Graduate School

Kookmin University

Radityo Fajar Pamungkas

2023

A Hybrid Approach of ConvLSTM-DT and GPT-4 for Real-time Anomaly Detection Decision Support in Edge-Cloud Environments

엣지-클라우드 환경에서 실시간 이상 탐지 결정 지원을 위한 ConvLSTM-DT 와 GPT-4 의 하이브리드 접근 방식

In charge of major work Professor Yeong Min Jang

A thesis submitted to the Department of Electronics Engineering and the Graduate School of Kookmin University in partial fulfillment of the requirements for the degree of Master

November 2023

Department of Electronics Engineering

The Graduate School

Kookmin University

Radityo Fajar Pamungkas

2023

Acknowledgement

First and foremost, I express my heartfelt gratitude to God for granting me the invaluable opportunity to pursue a master's degree at Kookmin University. I extend my sincerest thanks for the blessings that accompanied me throughout my studies in Korea. I want to express my sincere thanks to my supervisor, Prof. Yeong Min Jang, for his guidance, support, insightful ideas, constructive comments, and engaging discussions. His mentorship has been instrumental, especially in the successful completion of this thesis and my master's study. I also want to express my appreciation to Kookmin University for supporting my master's study with excellent researcher scholarships and providing world-class research facilities.

I would also like to express my gratitude to the members of the Wireless Communications and Artificial Intelligence (WiComAI) lab – Krishna, Ones, Faridh, Khairi, Rangga, Umam, Herfandi, Wang, Tuan Anh, Shin Eun Bi, and others. Their insightful comments and discussions have significantly enriched my academic journey.

Furthermore, I extend my thanks to the Youngnak IWE church community and Teras Cerdas community members for their unwavering prayers and continuous support. Lastly, I dedicate this thesis to my family, whose boundless love and encouragement have been a constant source of strength, guiding me through every step of this academic endeavor.

Radityo Fajar Pamungkas

Kookmin University, Seoul, Korea

December 2023

Table of Contents

List of Figures

List of Tables

Acronyms

Abbreviation	Full form
ADAM	Adaptive Momentum Estimation
ConvLSTM	Convolutional Long Short-Term Memory
DT	Dynamic Thresholding
IoT	Internet of Things
LSTM	Long Short-Term Memory
MSE	Mean Square Error
PLC	Programmable Logic Controller
PM	Particulate Matter
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
LLM	Large Language Model
GPT	Generative Pre-Training
PM	Particulate Matter
iForest	Isolation Forest
OC-SVM	One Class Support Vector Machine
HBOS	Histogram-based Outlier Score
LOF	Local Outlier Factor

Notation

r	Pearson's correlation coefficient
$\bar{x}$	Mean of input value
$\bar{y}$	Mean of target value
x_i	i-th input value
y_i	i-th target value
μ	Mean value
σ	Standard distribution
Z	Normalized value using Z-score method
$y^{l(i,j)}$	Convolution output
$*$	Dot product
i_t	LSTM input gate
f_t	LSTM forget gate
C_t	LSTM cell output
o_t	LSTM output gate
H_t	LSTM hidden state
$\circ$	Hadamard product

Abstract

A Hybrid Approach of ConvLSTM-DT and GPT-4 for Real-time Anomaly Detection Decision Support in Edge-Cloud Environments

By Radityo Fajar Pamungkas

Department of Electronics Engineering
Graduate School, Kookmin University,
Seoul, Korea

Anomaly detection is crucial in various domains for early identification of abnormal behavior. This research introduces an innovative approach that combines prediction-based detectors with Large Language Models (LLMs) for anomaly detection, focusing on indoor air quality data from multiple sensors. The hybrid approach integrates Convolutional Long Short-term Memory with non-parametric dynamic thresholding (ConvLSTM-DT) for prediction-based anomaly detection and fine-tuned GPT-4 for generating human-understandable explanations. Each sensor parameter has its specific model for accurate predictions. Furthermore, Dynamic thresholding and continuous learning adapts to the dynamic environment, update model and setting non-parametric confidence intervals for anomaly detection in rapidly changing scenarios. The system deploys anomaly detection on the edge for reduced latency and fast detection, while LLM processing occurs on the cloud for resource optimization. The results demonstrate accurate anomaly detection and well-explained reasoning for real-time decision-making, offering a novel approach for comprehensive anomaly detection solutions in various applications.

Keywords: Anomaly detection, Large Language Models, ConvLSTM-DT, Dynamic thresholding, Human-understandable explanations.

Chapter 1
Introduction

1.1 Background

In recent years, the rapid increase in data-driven applications and the growing complexity of modern systems have highlighted the critical significance of anomaly detection. Anomalies, often indicative of potentially harmful behavior within systems, can have long-term effects and wide-ranging consequences in various domains, such as industrial operations. Traditional anomaly detection methods, while effective to some extent, often heavily rely on expert analysis, making them resource-intensive and susceptible to human bias. This dependency on expert input can result in scalability, reliability, and consistency challenges within anomaly detection systems[1].

Furthermore, traditional anomaly detection methods based on statistical and machine learning approaches exhibit limitations, particularly when handling time-series data from multiple sensors. Anomalies in this context may be multi-dimensional and not easily characterized by conventional patterns. Consequently, traditional approaches can produce a significant number of false alarms, compromising the effectiveness and trustworthiness of anomaly detection systems[2].

In response to these challenges, this study aims to contribute to the field of anomaly detection by proposing a hybrid approach that harnesses the strength of prediction-based detector algorithms and the capabilities of Large Language Models (LLMs). This work centers on monitoring and detecting anomalies in air quality data collected from various sensors, including temperature, humidity, PM (Particulate Matter), and CO2 levels, all situated within a single indoor room environment. The ability to predict and detect

anomalies in this air quality data is of paramount importance, given its profound implications for human health, particularly for those occupying the room.

The proposed hybrid approach combines Convolutional Long Short-Term Memory with non-parametric dynamic thresholding (ConvLSTM-DT) for prediction-based anomaly detection and utilizes fine-tuned GPT-4 for LLM-based explainability. This hybrid approach is essential to address the aforementioned limitations and enhance the interpretability and adaptability of anomaly detection models. It not only facilitates anomaly detection across a variety of sensors but also provides human-understandable explanations and potential solutions for addressing anomaly situations. This makes it a powerful tool for real-world implementation where swift decision-making is imperative.

1.2 Related Works

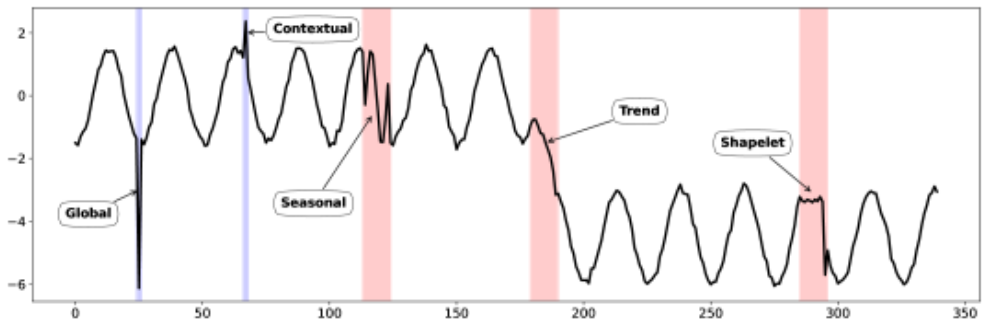

Figure 1. Various types of time series anomalies from public dataset[3].

There are various types of time series anomalies, as illustrated in Figure 1[4]. Global anomalies manifest as spikes in time series, represented by points with extreme values. These anomalies are often known as point anomalies. Contextual anomalies, on the other hand, are subsequences of data points situated within a specific proximity range but exhibit deviations from established context or patterns. Seasonal anomalies are subsequences of data

points that share similar shapes and trends, yet their seasonality deviates from the overall seasonal pattern. This deviation can result from the occurrence of concept drift. Trends anomalies signify events that induce a lasting shift in the data. Finally, shapelet anomalies denote subsequences of data points whose shapelets deviate from the typical shapelet component of the sequence.

Multiple techniques can be employed for anomaly detection on time series, as depicted in Figure 2. Currently, a significant number of researchers are exploring the utilization of various deep learning architectures for anomaly detection. This is due to the inherent power of deep learning methods in modeling data dependencies, particularly in data with complex structures. In deep learning techniques, there are two different approaches that are commonly used for anomaly detection, prediction-based approaches and reconstruction-based approaches.

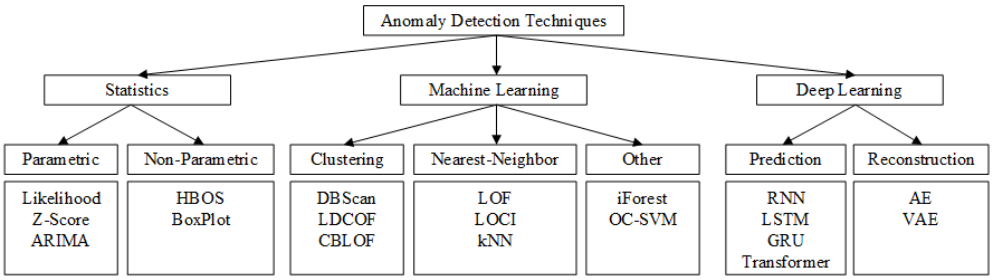

Figure 2. Common anomaly detection techniques for time series data.

Malhotra et al [5] introduced LSTM-AD, a model renowned for its long-term memory capabilities. A noteworthy innovation is the combination of hierarchical recurrent processing layers, a groundbreaking approach for detecting anomalies within univariate time series data. A captivating aspect of LSTM-AD is the first deep learning model that has the ability to perform anomaly detection without requiring labeled training data or unsupervised learning. By stacking recurrent hidden layers, this architecture empowers the

model to learn higher-order temporal trends without any prior knowledge of their temporal length.

Similarly, in the realm of anomaly detection, Chauhan et al [6] proposed DeepLSTM, a model that leverages a stacked LSTM recurrent network. This approach begins by training on normal data and subsequently employs maximum likelihood estimation to predict the error vector based on multivariate Gaussian distributions. Following this, the model is used to forecast a combination of both abnormal and normal validation data, while concurrently recording the probability density function (PDF) values associated with the error. Notably, this method offers the unique benefit of direct application to raw time series data, bypassing the need for preprocessing steps.

On another front, Hundmand et al. [7] embarked on the implementation of a non-parametric, dynamic, and unsupervised thresholding technique to assess errors for anomaly detection. LSTM-NDT combines multiple techniques, including LSTM, to achieve superior predictive performance. A key feature of this approach is the integration of dynamic thresholding and anomaly scoring, both of which can be automatically adjusted to accommodate the diverse, unstable, and noisy characteristics often encountered in dynamic data.

Thill et al [8] introduced the Temporal Convolution Network coupled with an autoencoder framework (TCN-AE), for prediction-based anomaly detection. In contrast to a vanilla autoencoder, TCN-AE replaces the dense layer architecture with a more potent and flexible CNN architecture, making it adaptable to varying input sizes. This architecture incorporates two temporal convolutional networks (TCNs)[9] for both encoding and decoding processes. Concurrently, Dai et al [10] developed the Graph-Augmented Normalizing

Flow (GANF) as an anomaly detection framework, grounded in graph neural networks (GNNs), to harness spatial feature correlations. Normalizing flow functions as a profound generative model in unsupervised learning, enabling the investigation of the inherent data distribution and addressing the issue of limited labels. GANF is cast as a Bayesian model, offering an estimate of the density for each data instance. This estimation aligns with the hypothesis that anomalies are more likely to be found in low-density regions.

In the case of reconstruction-based approaches, models such as Autoencoders (AE), LSTM-AE, and LSTM Variational Autoencoders (LSTM-VAE) have gained recognition. In contrast to prediction-based anomaly detection, which typically relies on identifying anomalies through deviations between actual and predicted values, this approach involves learning a latent low-dimensional representation to reconstruct the original input. Consequently, reconstruction-based approaches are often better suited for capturing contextual and collective anomalies[11].

While there has been extensive research on time series anomaly detection, the author has identified only one study that incorporated anomaly detection through explanation (ADE)[12]. In this context, Gilpin[12] employed reasonableness monitors based on anomaly detection within the context of simulated semi-autonomous vehicles. These simulations were designed to replicate real-world, anomalous driving scenarios. Furthermore, Gilpin utilized argument trees to generate reasoning and explanations for the detected anomalies. Gilpin[12] work represents a pioneering effort that has introduced the emerging field of explanatory anomaly detection, particularly in the realm of system-level explanations.

In the domain of Internet of Things (IoT) and air quality monitoring, the author could not find any study that integrates both anomaly detection and LLMs. Most existing approaches primarily rely on either prediction-based or reconstruction-based anomaly detection methods to identify anomalous behavior, without offering comprehensive explanations or reasoning. This study addresses this gap by introducing a hybrid approach that combines prediction-based anomaly detection with LLMs to provide a robust framework for constructing explanations and reasoning in the context of air quality monitoring within IoT systems.

1.3 Contributions

This work's primary contributions are:

1. A hybrid approach of ConvLSTM-DT and Fine-tuned GPT-4 is introduced and presented in detail. This model leverages the integration of prediction-based anomaly algorithms and well-performed LLMs to generate anomaly reasoning and possible solutions in anomaly situations.
2. An effective method for deploying the model in edge-cloud environments is outlined, ensuring seamless integration and optimal performance in real-world edge computing scenarios.
3. To assess the performance of anomaly detection, a comparative evaluation employs multiple machine learning anomaly detection algorithms, including Isolation Forest (iForest), One-Class Support Vector Machine (OC-SVM), Histogram-Based Outlier Score (HBOS), and Local Outlier Factor (LOF). These algorithms serve as benchmarks to gauge the effectiveness of the proposed hybrid approach.

The detailed content of this work is arranged as follows:

- **Chapter 1** presents the motivation and background of integrating anomaly detection with large language models.
- **Chapter 2** describes the architecture of the IoT system used for data collection and system implementation.
- **Chapter 3** explains the details of the ConvLSTM-DT method as a prediction-based anomaly detection technique.
- **Chapter 4** elaborates on the integration approach of LLM in anomaly detection to provide reasoning and explanations.
- **Chapter 5** presents the results of anomaly detection and compares them with well-known models.
- **Chapter 6** concludes the work results and outlines future research direction.

Chapter 2
System Overview

2.1 IoT Platform

In the era of Industry 4.0 and the emergence of IoT technology, sensors have become a pivotal role within the industry area. The deployment of sensors empowers industries with invaluable insights into the events transpiring within their factory or manufacturing facilities, that in the long run can enhance operational efficiency and product quality. To achieve this, industries often need to install a large number of sensors, numbering in the hundreds or even thousands, to capture a detailed picture of their processes. However, this rapid increasing of the sensors inevitably leads to the generation of huge data, underscoring the imperative need for a robust IoT platform capable of efficiently managing the myriad sensors, preserving the data they produce, and autonomously conducting data analysis[13].

Several well-established communication protocols, such as Zigbee, Wi-Fi, LoRa, and NB-IoT, are frequently employed in IoT systems to facilitate seamless data exchange[14], [15]. Alongside these methods, communication software such as MQTT, RestAPI, OPC-UA, Modbus, and CoAP are instrumental in establishing data transfer mechanisms between IoT devices[16]. Each IoT system plays a critical role in data processing, device management, data flow control and the harmonization of data from diverse devices, thus ensuring operation within the industrial setting.

In this study, an IoT system was established to gather indoor environmental data for the training and implementation of the proposed model. The primary components of the IoT framework, as illustrated in Figure 3, comprise the device layer, which includes sensors and actuators, edge layer, and cloud layer. To manage data collection from the sensors and control the actuator,

Wago PLC PFC200 750-8212 and Schneider Modicon M262 devices are deployed in this layer. These PLCs are connected via Modbus RTU and Modbus TCP connections, and subsequently, they send data to the edge server using OPC-UA connections. Moreover, specific industrial sensors have the capability to transmit data directly to the edge using MQTT.

In the edge layer, it encompasses an edge server responsible for data aggregation, edge computation, and real-time telemetry data analysis. In this study, a NVIDIA Jetson Nano was selected as the edge server due to its specific features and specifications. The data from multiple sensors is stored in a local NoSQL database, MongoDB, every five seconds. The use of MongoDB time series collections enables efficient data storage and offers advantages such as reduced missing data and latency. The edge server's capabilities extend to anomaly prediction for each sensor, and it generates prompts that are subsequently sent to the cloud server for further analysis.

Lastly, the cloud layer plays an important role in producing explanations and reasoning for anomalies based on the prompts received from the edge server. These anomaly explanations can be directly transmitted to maintenance operators for validation. Furthermore, this cloud layer supports data visualization, system reporting, and maintenance functionalities to create reliable operation and maintenance system.

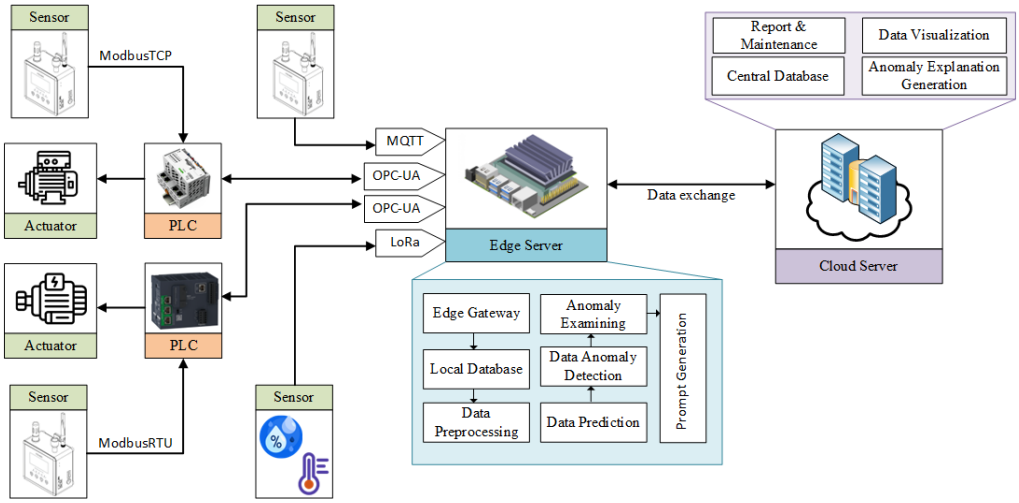

Figure 3. IoT system developed in this study.

2.2 Collected Data

Data acquisition and collection in this research are conducted using the provided IoT platform. The dataset comprises nine parameters: temperature, humidity, luminance, CO2, PM0.3, PM0.5, PM1.0, PM2.5, PM5.0, and PM10.0, acquired from seven sensors. The placement of each sensor is depicted in Figure 4.

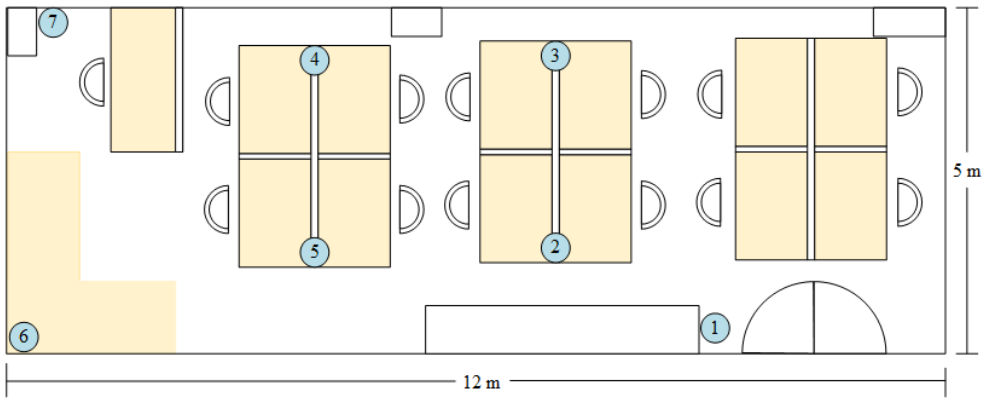

Figure 4. Sensor's placement at indoor laboratory environment.

Depending on the connection type and sensor type specification, these sensors have varying time sampling intervals, ranging from one second to three seconds, The experiments were carried out from October 20th to 25th, 2023, within the indoor environment of one of the laboratories at CCRF Kookmin University. Detailed statistics of all the sensor data are provided in Table 1.

Table 1. Detailed description of the collected data

Variables	Data Count	Mean	Standard Deviation	Min	Max
Temperature 1	172,826	20.85	0.92	19.2	28.3
Temperature 2	172,801	24.15	0.77	22.3	28.1
Temperature 3	172,784	26.10	1.38	19.5	29.7
Temperature 4	547,217	22.25	1.26	20.3	28.0
Temperature 5	547,217	21.63	1.27	19.5	29.6
Temperature 6	547,217	21.72	1.13	20.0	28.0
Temperature 7	547,217	22.29	1.06	20.33	29.53
Humidity 1	172,826	36.75	5.69	27.3	52.5
Humidity 2	172,801	34.79	5.28	26.0	49.4
Humidity 3	172,784	35.43	5.67	25.7	50.5
Humidity 4	547,217	39.58	5.10	30.5	53.2
Humidity 5	547,217	40.13	4.96	30.9	53.6
Humidity 6	547,217	39.17	5.89	27.3	54.1
Humidity 7	547,217	36.73	6.77	24.58	52.66
CO2	547,217	59.14	28.34	24.0	190.0
Lux	547,217	225.28	296.41	649.0	2.0
PM0.3	547,217	31,170.89	13,426.04	10,130	65,535.0
PM0.5	547,217	2,864.85	1,262.28	1,104.0	29,615.0
PM1.0	547,217	643.29	270.46	205.0	4554.0
PM2.5	547,217	151.95	78.39	31.0	1890.0
PM5.0	547,217	13.10	10.16	0.0	213.0
PM10.0	547,217	1.91	3.01	0.0	61.0

Chapter 3

Prediction-based Anomaly Detection

3.1 Data Preprocessing

Data preprocessing stands as a critical initial phase in the development of a machine learning model, as it enhances the machine learning model's ability to understand the data and accelerates the training phase. This essential step encompasses various techniques employed to convert raw data into cleaner and more insightful information. In the context of this study, we implement three specific data preprocessing steps on the collected dataset.

3.1.1. Data Cleaning

The utilization of real-world IoT data from our implemented platform introduces inherent challenges associated with missing data. Such data gaps are primarily caused by various factors, including dissimilar time sampling rates among sensors and communication errors. These challenges complicate the alignment of data in the overall dataset and can introduce substantial errors in predictive modeling. Extensive research has underscored the direct impact of missing data on machine learning model performance.

To address this issue, multiple techniques are available for imputing missing data, including forward and backward imputation. In our study, we opted for the linear interpolation method to fill gaps within the indoor environmental dataset. Linear interpolation involves estimating missing data points by connecting neighboring observations with straight-line segments, and it is a practical choice for an environment where rapid changes within one-second intervals are unlikely. The linear interpolation technique is detailed in Equation 1. In this context, y represents the value used to replace the missing data and x denotes the independent variable.

$$y = y_1 + (x - x_1)\frac{y_2 - y_1}{x_2 - x_1} \quad (1)$$

Moreover, a moving average technique was employed to achieve data smoothing, with its formulation depicted in Equation 2. In this equation, t represents current time and n denotes the size of the rolling window used for averaging the values.

$$y = \frac{y_t + y_{t-1} + \cdots + y_{t-n-1}}{n} \quad (1)$$

3.1.2. Feature selection

In environmental data analysis, effective feature selection is essential for uncovering the most influential variables within complex datasets. Feature selection, a critical step in data preprocessing, enables the identification of key features that have the most significant impact on predicting target feature. Then, choose just the data features that have a high correlation score with the predicting target feature and discard the rest. In this study, the Pearson correlation coefficient[17] is employed to assess the correlation score between dataset features. The mathematical equation for Pearson's correlation coefficient can be represented as follows:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \quad (3)$$

where r represents the correlation coefficient score, x and y are the value in the dataset.

Figure 5 illustrates the correlation score within the dataset's confusion matrix. This matrix serves as the basis for feature selection, with a threshold of | 0.3 | applied. For instance, when considering the CO2 level, only a select few features exhibit correlations above this threshold. These features encompass all temperature and humidity sensors, luminance, PM0.3, PM0.5,

and PM1.0, while the remaining features with correlation values below the threshold are omitted. Consequently, each prediction model designed to forecast specific features varies in input dimensions based on the correlation scores.

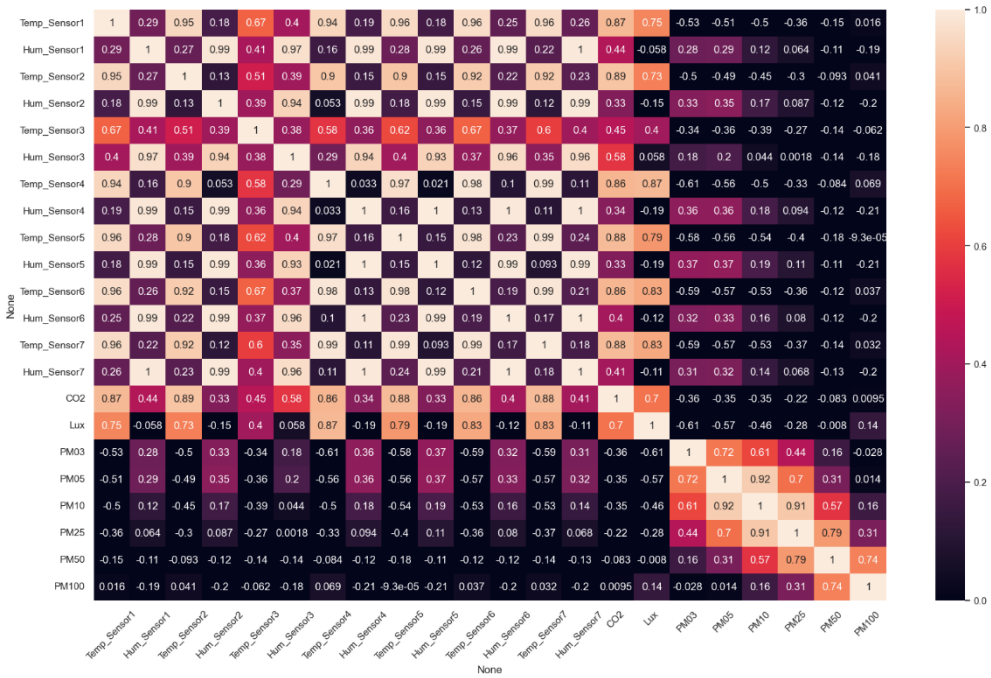

Figure 5. Correlation confusion matrix based on Pearson's correlation.

3.1.3. Normalization

The different distributions of time-series, as detailed in Table 1 for each feature, introduces the potential for increased uncertainty if the data is not appropriately pre-processed prior to being fed into machine learning models. To address this concern, a normalization technique is applied to scale the entire dataset, allowing model to convergence quicker. In this context, the MinMaxScaler normalization method was selected, which scales the dataset to a range between -1 and 1. The scaling calculation for this method is defined as follows:

$$x' = 2\left(\frac{x - min(x)}{max(x) - min(x)}\right) - 1 \quad (4)$$

where x represents the actual value, x' denotes the normalized value, $min(x)$ and $max(x)$ denote the minimum and maximum value within dataset.

Table 2. Detailed data description after implementing data cleaning and MinMax normalization.

Variables	Data Count	Mean	Standard Deviation
Temperature 1	547,217	-0.132	0.482
Temperature 2	547,217	-0.361	0.418
Temperature 3	547,217	0.01	0.436
Temperature 4	547,217	-0.19	0.555
Temperature 5	547,217	-0.171	0.522
Temperature 6	547,217	-0.064	0.515
Temperature 7	547,217	-0.135	0.57
Humidity 1	547,217	-0.042	0.589
Humidity 2	547,217	-0.032	0.586
Humidity 3	547,217	-0.044	0.579
Humidity 4	547,217	0.017	0.586
Humidity 5	547,217	0.004	0.590
Humidity 6	547,217	0.023	0.584
Humidity 7	547,217	0.03	0.576
CO2	547,217	-0.588	0.345
Lux	547,217	-0.305	0.92
PM0.3	547,217	-0.067	0.409
PM0.5	547,217	-0.623	0.263
PM1.0	547,217	-0.746	0.159
PM2.5	547,217	-0.837	0.107
PM5.0	547,217	-0.842	0.118
PM10.0	547,217	-0.929	0.102

The updated dataset values, following data cleaning and normalization, are presented in Table 2. A comparison between the data counts in Table 1 and Table 2 reveals that the values for each data count have been modified as a consequence of applying linear interpolation. Furthermore, the data characteristics have also been altered due to the implementation of Min-Max normalization.

3.2 Long Short-Term Memory (LSTM) Model

LSTM [18] represents a specialized iteration of Recurrent Neural Networks (RNN), designed to address the shortcomings of conventional RNNs in handling long-duration temporal dependencies effectively. In comparison to RNNs, LSTM boasts a distinct architectural structure that facilitates the learning of prolonged dependencies. This is accomplished through the incorporation of memory units within LSTM, enabling the preservation of cell states over extended time intervals. Moreover, LSTM incorporates four gates: the input gate, forget gate, output gate, and cell state gate. During each time step t, LSTM processes an input sequence X and iteratively generates an output sequence Y for t=1,2,3,…,T. In each step, LSTM updates all of its units and subsequently computes the loss for all weights. Figure 6 provides a visual representation of the LSTM architecture, while the LSTM equations are expressed as follows:

$$s(t) = y_{\varphi}(t) \odot s(t-1) + y_i(t) \odot g\big(Z_c(t)\big) \tag{5}$$

$$Z_c(t) = W_{c,x}X(t) + W_{c,j}j(t-1) + b_c \tag{6}$$

$$y_{\varphi}(t) = W_{c,x}x(t) + W_{c,j}j(t-1) + b_c \tag{7}$$

$$y_i(t) = \sigma\big(W_{i,x}x(t) + W_{i,j}j(t-1) + W_{i,s}s(t-1) + b_i\big) \tag{8}$$

$$y_o(t) = \sigma\big(W_{o,x}x(t) + W_{o,j}j(t-1) + W_{o,s}s(t-1) + b_o\big) \tag{9}$$

$$j(t) = y_o(t) \odot h(s(t)) \tag{10}$$

$$y_{out} = f_{out}(W_{(y_{out},j)}j(t) + b_{y_{out}}) \tag{11}$$

In this context, $s(t)$ represents the current cell state, y corresponds to the output of each gate, and $\odot$ is the component-wise multiplication. Additionally, W denotes the weight matrices, b signifies the bias vectors associated with each gate, and therefore Z denotes the sum of weighted inputs. Furthermore, σ, g, h and f_{out} are the output activation functions and y_{out} is the final output of the network.

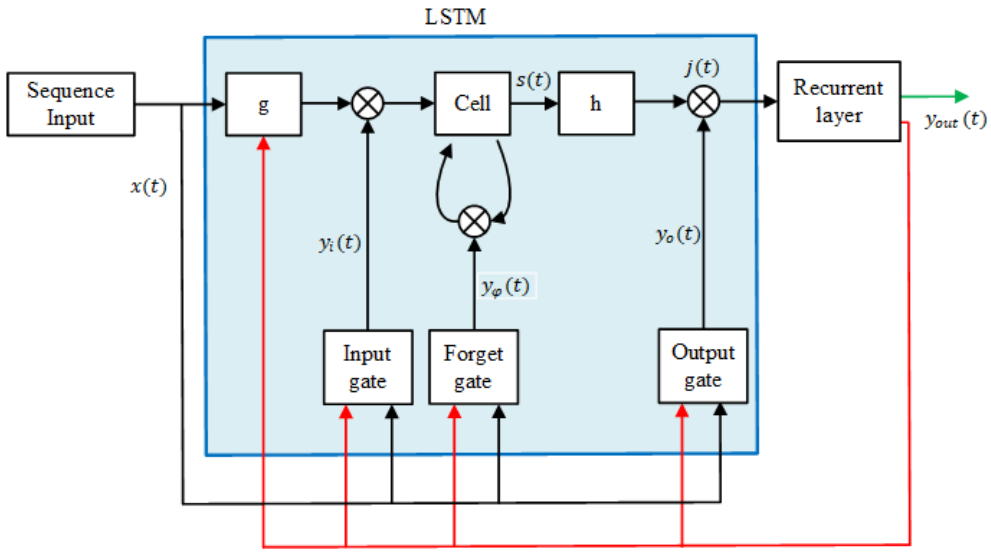

Figure 6. Architecture of LSTM layer.

3.3 Convolutional LSTM (ConvLSTM) Model

The architecture of ConvLSTM is depicted in Figure 7. While typical time-series forecasting models based on recurrent networks primarily focus on temporal characteristics, they often overlook valuable spatial information within the data. To overcome this limitation, some studies incorporate a convolutional layer to extract spatial information and subsequently apply recurrent neural network-based techniques, including LSTM and Gated Recurrent Unit (GRU)[19], along the temporal axis to comprehensively leverage both spatial and temporal information[20]. While the structure of the LSTM used in this context remains consistent with a regular LSTM, the key distinction lies in the input side, where it employs a convolutional layer to capture spatio-temporal information. The formulations for the ConvLSTM[21] are as follows:

$$i(t) = \sigma\left(W_{x,i} * x(t) + W_{h,i} * \mathcal{H}(t-1) + W_{c,i} \circ C(t-1) + b_i\right) \quad (12)$$

$$C(t) = f(t) \circ C(t-1) + i(t) \circ g\left(W_{x,c} * x(t) + W_{h,c} * \mathcal{H}(t-1) + b_c\right) \quad (13)$$

$$f(t) = \sigma\left(W_{x,f} * x(t) + W_{h,f} * \mathcal{H}_{t-1} + W_{c,f} \circ C(t-1) + b_f\right) \quad (14)$$

$$o(t) = \sigma\left(W_{x,o} * x(t) + W_{h,o} * \mathcal{H}(t-1) + W_{c,o} \circ C(t) + b_o\right) \quad (15)$$

$$\mathcal{H}(t) = o(t) \circ tanh(C(t)) \quad (16)$$

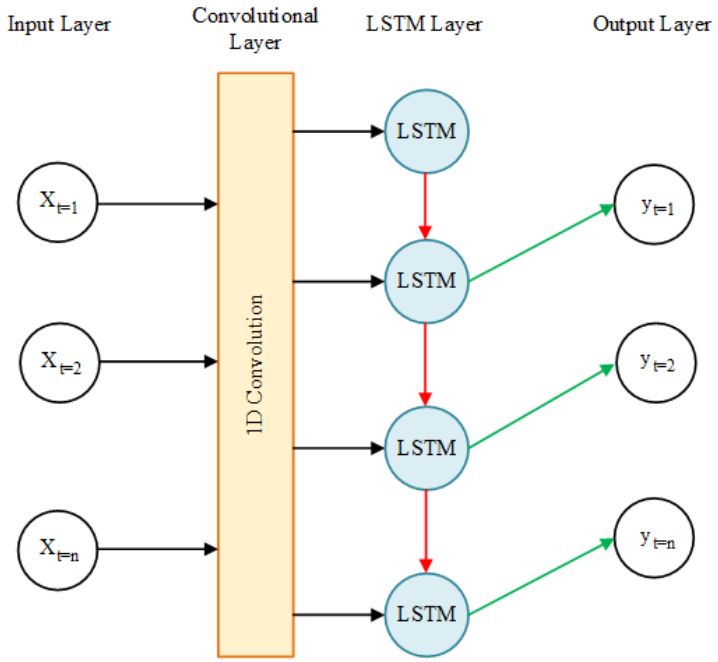

Figure 7. Convolutional LSTM architecture.

Similar to the LSTM, W represents the weight of each layer, b represents the network bias, $i(t)$ stands for the input gate, $f(t)$ for the forget gate, $C(t)$ represent the cell output, and $o(t)$ denotes the output gate. Meanwhile, $\mathcal{H}(t)$ denotes the final state, and σ denotes the applied activation function, which is in this context is either a hyperbolic tangent (tanh) or sigmoid function. Furthermore,'*' denotes the convolution operation and '∘' indicates the Hadamard product.

3.4 Optimizer

A machine learning optimizer is a fundamental component of the training process, crucial for enhancing the performance of machine learning models. Its primary role is to iteratively adjust the model's parameters to minimize a defined loss function, thereby optimizing the model's predictive accuracy.

This iterative process involves evaluating the model's performance on the training data, computing the gradients of the loss function with respect to the model's parameters, and then updating the parameters in a direction that minimizes the loss. By iteratively fine-tuning the model's parameters, the optimizer enables the model to learn from data and improve its ability to make accurate predictions on unseen data. The choice of optimizer and its hyperparameters can significantly impact the training process and the model's final performance. The optimizer employs various optimization algorithms, such as stochastic gradient descent (SGD)[22], adaptive moment estimation (ADAM)[23], or RMSprop, to find the optimal parameter values.

In this study, ADAM is employed as the optimizer. ADAM was designed to replace the classical SGD optimizer. ADAM works by combining the advantages of two optimization techniques, namely RMSprop and Momentum. It maintains a dynamic learning rate for each parameter, adapting it based on the past gradients and their magnitudes. By doing so, ADAM effectively handles sparse gradients and noisy data, making it particularly suitable for deep learning tasks with complex and large datasets. Additionally, ADAM incorporates bias correction mechanisms to counteract potential biases introduced during the early stages of training, ensuring more stable and reliable optimization. This optimizer's adaptive learning rate and the ability to efficiently navigate complex loss landscapes make it a popular choice in machine learning applications, offering improved convergence and better generalization performance.

ADAM optimizer can adapt the learning rates for each parameter based on the past gradients and squared gradients, allowing for efficient optimization and convergence during training. The formulations of ADAM are follows:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \quad (17)$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \quad (18)$$

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t} \quad (19)$$

$$\widehat{v_t} = \frac{v_t}{1-\beta_2^t} \quad (20)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\widehat{v_t}} + \varepsilon} \widehat{m_t} \quad (21)$$

where it accumulates exponentially decaying averages of past gradients $m_t$and squared gradients v_t, with β_1 and β_2 as hyperparameters controlling the decay rates. These moving averages help smooth out gradient variations and adjust step sizes for parameter updates. The algorithm calculates bias-corrected estimates ($\widehat{m_t}$ and $\widehat{v_t}$) and updates the parameters θ_{t+1} in each iteration t. The learning rate η controls the step size, while ε is a small constant for numerical stability.

3.5 Non-Parametric Dynamic Thresholding

To effectively monitor thousands of telemetry channels influenced by changing environmental conditions and command sequences, there's a critical need for a fast and unsupervised approach to detect anomalies in predicted values. While common methods like distance-based techniques are available, they often come with high computational costs, involving comparisons between each data point and a set of k neighbors. Another frequently employed method relies on the assumption of Gaussian distributions for past smoothed errors, which enables swift comparisons with concise representations of previous errors [7], [24], [25]. However, this approach can face challenges when parametric assumptions are violated. To address this issue, non-parametric dynamic thresholds play a crucial role. These non-parametric dynamic thresholds can be divided into three key processes: error and

smoothing, threshold calculation, and anomaly scoring. The equations related to this approach are as follows:

$$e_s = \left[e^{(t-h)}, e^{(t-h-1)}, \ldots, e^{(t)}\right] \tag{22}$$

$$\epsilon = argmax(\epsilon) = \frac{\frac{\Delta\mu(e_s)}{\mu(e_s)} + \frac{\Delta\sigma(e_s)}{\sigma(e_s)}}{|e_a| + |E_{seq}|^2} \tag{23}$$

$$\Delta\mu(e_s) = \mu(e_s) - \mu(\{e_s \in e_s | e_s < \epsilon\}) \tag{24}$$

$$\Delta\sigma(e_s) = \sigma(e_s) - \sigma(\{e_s \in e_s | e_s < \epsilon\}) \tag{25}$$

$$e_a = \{e_s \in e_s | e_s > \epsilon\} \tag{26}$$

$$E_{seq} = \{e_a | e_a \in e_a\} \tag{27}$$

$$s^{(t)} = \frac{max\left(e_{seq}^t\right) - argmax(\epsilon)}{\mu(e_s) + \sigma(e_s)} \tag{28}$$

Where e denotes the set of errors, e_s denotes the smoothed errors, ϵ is the chosen threshold, E_{seq} represents error sequence, and $s^{(t)}$ is anomaly score. In simpler terms, the objective is to look for a threshold that, when applied to the data, would lead to the most significant reduction in both the average error and the spread of errors.

3.6 Continous Training

Continuous training is a pivotal component of maintaining the reliability of anomaly detection models, especially in dynamic environments where concept drift is prevalent. Concept drift refers to the evolving nature of data distributions over time, making models susceptible to performance degradation if they are not retrained to adapt. In this context, our approach involves a proactive strategy where the training of the anomaly detection model is triggered when the anomaly rate surpasses a predefined limit. This threshold-based approach is designed to ensure that the model remains in sync

with the evolving data patterns and maintains its accuracy in real-world applications.

By setting an anomaly rate threshold, the system can monitor the model's performance against current data. When the anomaly rate exceeds this limit, it serves as an early warning signal that the model may be facing concept drift. Consequently, the training process is initiated automatically, enabling the model to relearn from recent data and recalibrate its anomaly detection capabilities. Continuous training in response to concept drift not only safeguards the model against performance degradation but also ensures that it can effectively adapt to dynamic conditions.

This approach addresses the challenges of real-time decision-making in rapidly changing environments, where the ability to detect anomalies accurately is critical. Continuous training, triggered by anomaly rate thresholds, serves as an indispensable tool to maintain the model's reliability and performance over time, making it an essential component for anomaly detection in dynamic conditions.

Chapter 4
Large Language Model Integration

4.1 Transformer Model

The Transformer has demonstrated its exceptional capabilities across a diverse range of tasks, including machine translation[26], document generation[27], and syntactic parsing[28]. The foundation of the Transformer architecture lies in the innovative concept of Self-Attention [26], which is effectively implemented through multi-head attention mechanisms. This design offers us a highly structured memory system that excels in handling long-term dependencies within text, far surpassing the capabilities of traditional recurrent networks. This results in robust transfer performance across a wide array of tasks.

One of the key strengths of the Transformer model is its capacity for parallel data processing, significantly reducing the time required to deploy the model. Moreover, it is important to note that well-established Generative Pre-Training (GPT)[29] models, such as GPT-3 and GPT-4, are built upon the Transformer framework. These models leverage the Transformer's architecture to achieve state-of-the-art performance in tasks that demand natural language understanding and generation. This ability to seamlessly transition from numbers to human-understandable explanations makes Transformer-based models a powerful choice for anomaly detection tasks to provides valuable insights into why they are anomalous. In this study, GPT-4 from OpenAI is utilized as the LLM model to generate human-understandable explanations for anomaly data and provide potential solutions in anomalous situations.

4.2 Fine-Tuning GPT-4 Model.

While GPT-4 is a powerful and large-scale model, its base version is designed to be general-purpose, which may not fully meet the specific requirements of task-specific anomaly detection, especially when applied to

domains like time-series anomaly detection in machine and environmental data. The reasoning generated by the base LLM may be overly general for the intricacies of anomaly detection in these domains.

To address this, fine-tuning GPT-4 with a custom dataset provides several key advantages[30]. It allows for the tailoring of the model's capabilities to the specific task or domain, leading to higher accuracy and more comprehensive reasoning. Additionally, fine-tuned models can be rapidly deployed for practical applications, making them particularly suitable for tasks where real-time or time-sensitive responses, such as anomaly detection, are essential.

As of Fall 2023, OpenAI has introduced features for fine-tuning GPT 3.5 Turbo and GPT-4. This development opens up the possibilities for users to customize these models to suit their specific use cases. In this study, a custom dataset has been created, comprising simulations of anomaly situations and their corresponding reasoning. This custom dataset will be preprocessed to align with the format of the base GPT-4 model, and the GPT-4-based model, initialized with pre-trained weights, will undergo fine-tuning in the Microsoft Azure AI cloud environment. This fine-tuning process ensures that the model is well-equipped to provide accurate and domain-specific anomaly detection and reasoning.

4.3 Large Language Model Integration

The process of integrating LLM with anomaly detection is illustrated in Figure 8. To achieve this integration, the OpenAI API was utilized. The ConvLSTM-DT algorithm first examines data from each sensor. Whenever an anomaly is detected, the program is triggered to generate a simple anomaly explanation. This explanation typically includes essential details such as the

anomaly score, sensor location, the average anomaly score over the last 5 minutes, and the collective anomaly score within the same room.

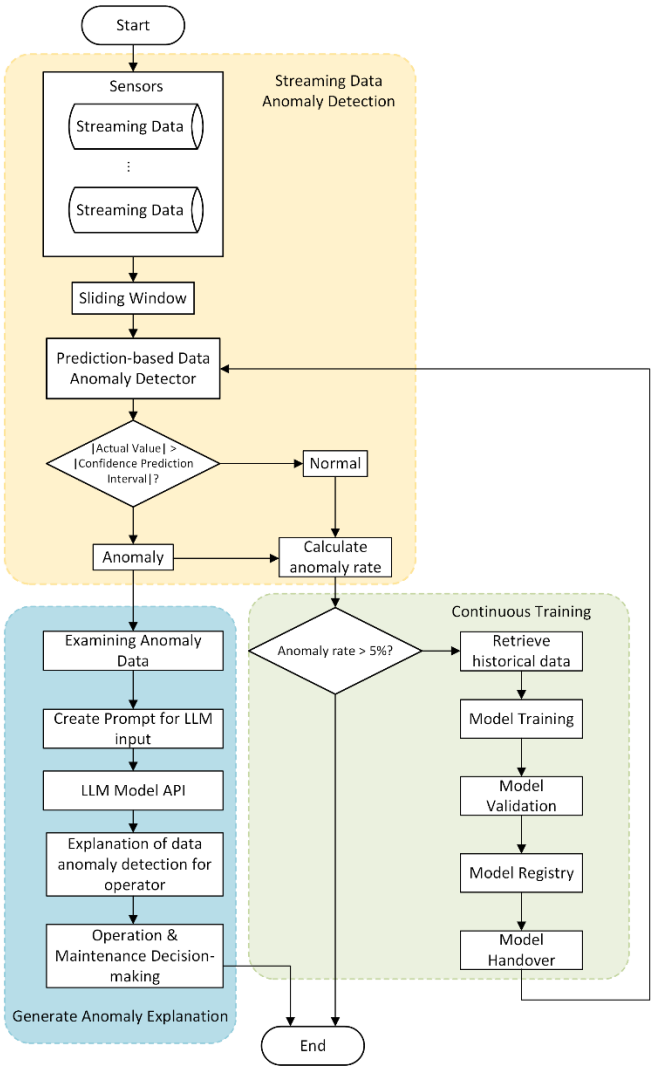

Figure 8. Flowchart of integration between prediction-based anomaly detection and large language model.

These detailed anomaly data are then structured into a prompt that follows the OpenAI API format. Subsequently, this prompt is transmitted to the cloud. In the cloud, the fine-tuned GPT-4 model comes into action. It receives the prompt and processes it to generate human-understandable explanations and

possible solutions for the detected anomaly situation. This feedback is invaluable for operators as it provides clear insights into the anomaly and recommendations for addressing it. This approach ensures efficient management of anomaly situations.

This approach is designed to optimize system performance. Anomaly detection is executed at the edge, which reduces latency and ensures fast detection. On the other hand, the LLM, which demands significant computational resources, is executed in the cloud. This division of labor ensures a balance between real-time detection and resource-intensive reasoning, enhancing the overall efficiency and effectiveness of the system.

Chapter 5

Results and Discussion

5.1 Training Environment

The computational processes in this study were executed on a computer equipped with an AMD Ryzen 5 3400G processor, 24GB of RAM operating at 2667MHz, an NVIDIA RTX 3080Ti GPU with 12GB of memory, and the Windows 10 operating system. The primary tools used for conducting experiments and analysis within the deep learning framework were Python 3.10 and TensorFlow 2.9. The deep learning model underwent training using a batch size of 128, and the initial learning rate was set at $5x10^{-5}$. Learning rate decay was implemented if the validation loss did not decrease within 5 epochs. The mean squared error (MSE) served as the loss function, and a maximum of 200 epochs were allowed for training. Early stopping was employed as a precautionary measure against overfitting. Table 3 summarizes the hyperparameter tuning settings utilized during the training of the ConvLSTM prediction model.

Table 3. Hyperparameter tuning settings.

Hyperparameter	Setting Value
Batch Size	128
Learning rate	0.00005
Optimizer	ADAM
Learning rate schedule	Monitoring: Validation loss (MSE), Patience: 5
Number of epochs	100
Early Stopping	Monitoring: Validation loss (MSE), Patience: 20
Loss Function	Mean Squared Error Loss
Total Param	663,944

5.2 Evaluation Metrics

5.2.1 Mean Squared Error (MSE)

The MSE is a measure that calculates the average of the squared differences between predicted values and observed data. It provides a quantification of how much the model's predictions vary from the actual data points. The MSE method squares these differences to account for both positive and negative errors, giving more weight to larger errors. A lower MSE signifies a closer alignment between the model's predictions and the actual data, indicating a more accurate and precise model. The equation for MSE is as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \quad (29)$$

Where n indicates the total number of values in the validation or test set, Y_i is the actual value from the sensors, and $\hat{Y}_i$ is the prediction values.

5.2.2 F1 Score

The F1 score serves as a metric that integrates both precision and recall, offering a balanced assessment of a model's overall performance. This performance metric takes into account both the true positive (TP) and false positive (FP) rates, as well as the false negative (FN) rate. The F1 score is computed using the formula:

$$Precision = \frac{TP}{TP + FP} \quad (30)$$

$$Recall = \frac{TP}{TP + FN} \quad (31)$$

$$F1\ Score = 2\ x\ \frac{recall\ x\ precision}{recall + precision} \quad (32)$$

5.2.3 Area Under the Score (AUC) Score

Area Under the Receiver Operating Characteristic (ROC) Curve or usually known as AUC is a commonly used metric to assess the performance of a model. This measurement metric quantifies the overall performance of the model across different threshold settings. A model with a higher AUC-ROC score indicates that it has a better ability to distinguish between normal and anomalous instances. The formula is as follows:

$$False\ positive\ rate\ (F_{PR}) = \frac{FP}{FP + TN} \tag{33}$$

$$AUC = \int Recall\ d(F_{PR}) \tag{34}$$

5.3 Prediction Model Performance

In the implementation, it is important to highlight that each sensor parameter has its own specific model designed to predict the next one hour worth of data based on the input of one hour of data. This means that the training and validation loss for each parameter and sensor can exhibit variations based on their unique characteristics and the complexity of the prediction task.

Figure 9 provides a visual representation of the model training progress for temperature and humidity forecasting. The depicted graphs illustrate the behavior of the model's training and validation loss as the training process progresses. In these graphs, the training loss and validation loss consistently decrease and stabilize at around 60 to 70 epochs. This behavior suggests that the models are effectively learning from the data without showing signs of significant underfitting or overfitting. These loss trends indicates that the

models are fitting the data well, which is a positive sign of their predictive capabilities.

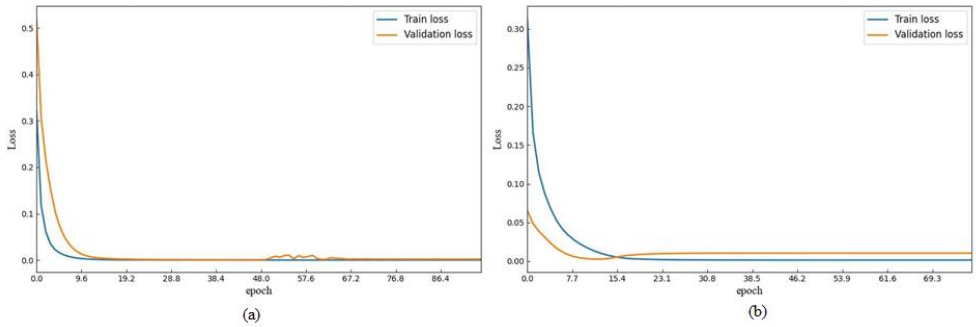

Figure 9. Evolution of training and validation loss. (a) Temperature training progress. (b) Humidity training progress.

The histogram presented in Figure 10 illustrates the prediction error values concerning the ground truth data, and it also provides a count of data points associated with each error value. This analysis is conducted for a total of 3,600 data points, equivalent to one hour's worth of data. The results depicted in Figure 10 clearly indicate that the prediction model exhibits exceptional performance.

The key observation from the histogram is that the model consistently generates a relatively small range of squared error values. For all the features examined, the majority of errors fall below 0.5, with only a few data points exhibiting errors exceeding 1.2. This observation is significant, as it suggests that the model effectively captures the trends in the data and predicts the various features with a high degree of accuracy. This level of performance positions the model as a strong foundation for an effective anomaly detector, as it demonstrates a robust ability to capture and reproduce the expected patterns in the data.

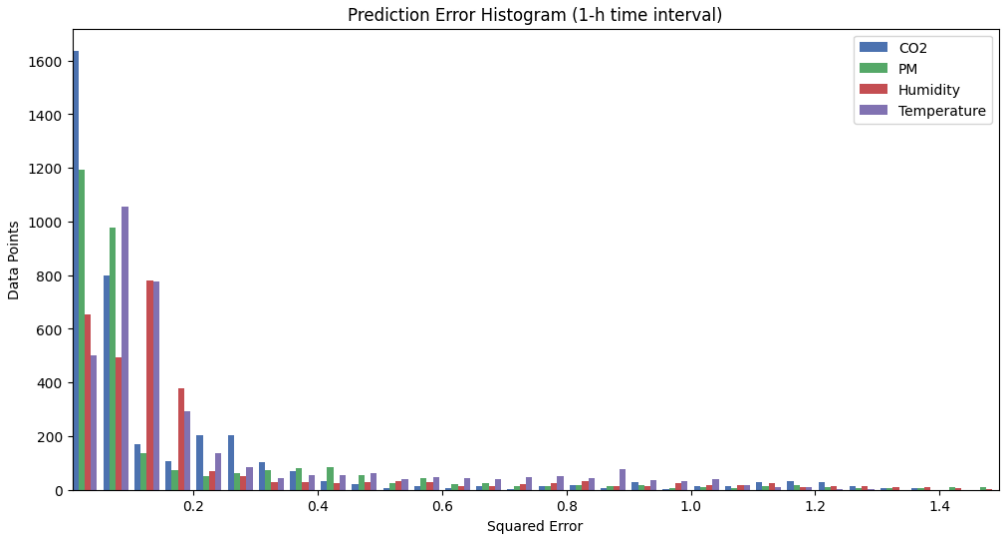

Figure 10. Histogram of the prediction errors in 3,600 data points.

5.4 Anomaly Detection Performance

Following the execution of the prediction process, the subsequent steps involve the calculation of dynamic thresholds and anomaly scoring to determine whether the data should be categorized as normal or abnormal. The non-parametric dynamic threshold creation involves the establishment of a confidence interval, which effectively sets upper and lower limits for the anomaly detection process.

Figure 11 visually illustrates the relationship between humidity and temperature data, presenting both the prediction and actual data. In the plot, the green line represents the actual data, while the blue line depicts the prediction data. From the plot, it can be seen that the ground-truth and prediction data are close to each other and can follow the data trends of the ground-truth data. Notably, the last 3,600 data points in the actual value dataset appear as empty, which is attributed to the continuous influx of incoming data. Additionally, the data exhibits changing patterns at multiple points in the plot,

necessitating the model's quick adaptation to these dynamic environmental shifts. The dynamic threshold plays a crucial role in enabling the model to effectively adapt to and accommodate these evolving data patterns.

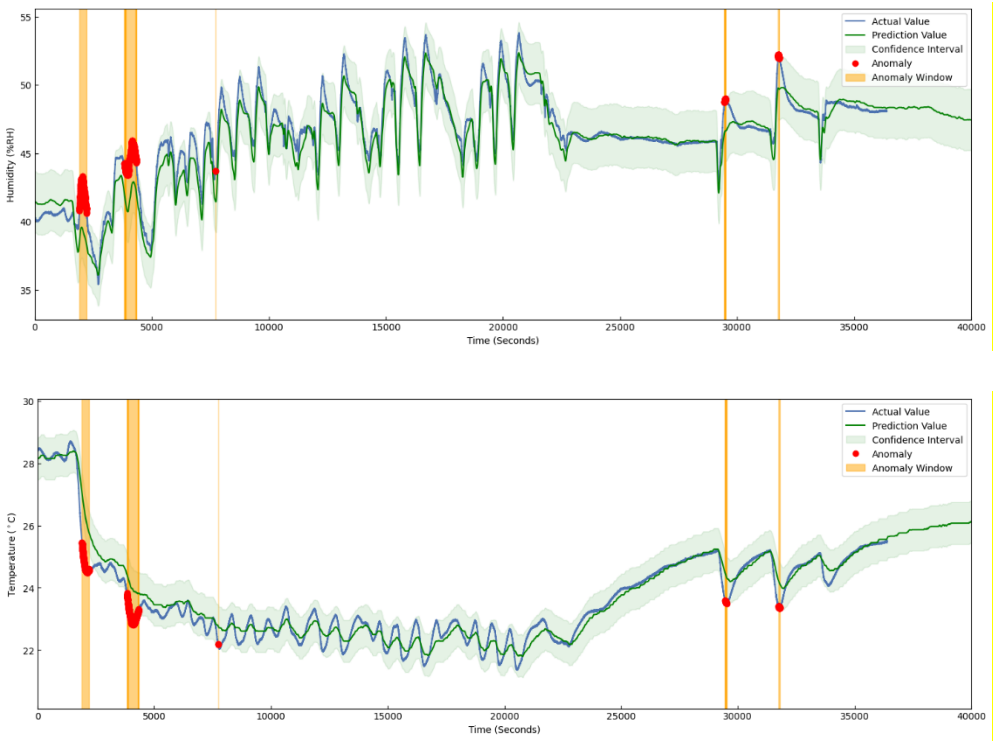

Figure 11. Plot of the prediction-based anomaly detection results.

Features	F1 Score				AUC			
	IF	**SVM**	**LOF**	**Pro**	**IF**	**SVM**	**LOF**	**Pro**
Temperature	0.85	0.79	0.73	**0.92**	0.92	0.89	0.86	**0.96**
Humidity	0.86	0.8	0.74	**0.89**	0.91	0.88	0.87	**0.95**
CO2	0.84	0.78	0.72	**0.88**	0.9	0.86	0.84	**0.93**
PM	0.81	0.84	0.77	**0.92**	0.94	0.92	0.88	**0.97**

Note: IF : Isolation Forest, SVM : One-class Support Vector Machine, LOF : Local Outlier Factor, Pro : Proposed Method.

5.5 Anomaly Explanation Generation Results

In this section, the outcomes of a hybrid approach for generating anomaly explanations are described, shedding light on the quality and reasonableness of the explanations produced. The explanations generated by the fine-tuned GPT-4 model are evaluated based on their coherence and relevance. Our findings indicate that the explanations provided for detected anomalies consistently exhibit a high degree of coherence, making them easily understandable for operators and end-users. This coherency stems from the model's ability to analyze the anomalous data, identify key contributing factors, and articulate them in a clear and logical manner. As a result, the anomaly explanations are not only informative but also readily comprehensible, ensuring that users can quickly grasp the underlying issues.

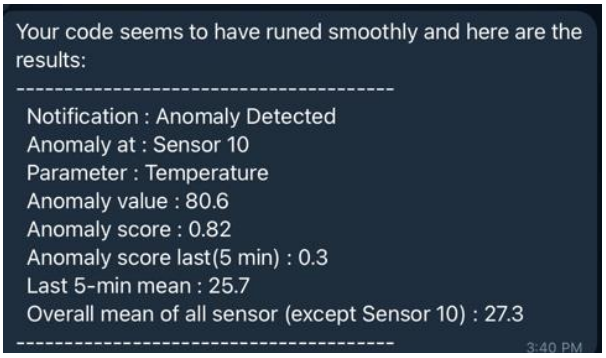

Figure 12. The prompt as input for the fine-tuned GPT-4 model.

Furthermore, the generated explanations are highly relevant to the detected anomalies, offering a clear and concise connection between the observed data patterns and the anomalies themselves. The explanations effectively highlight the specific aspects of the data that led to the anomaly detection, thus providing valuable insights into the root causes of these anomalies. This relevance ensures that the explanations are not only informative but also directly applicable for users seeking to address and rectify

anomalous situations. Overall, the anomaly explanations produced by the fine-tuned GPT-4 model stand as a testament to the effectiveness of our approach in delivering meaningful and actionable insights for anomaly detection scenarios.

The model's functionality and effectiveness are exemplified in Figure 12 and Figure 13, showcasing a clear illustration of its input and output. This figure presents a specific use case where the model processes input data and provides an explanation in response to a detected anomaly. It demonstrates the model's ability to understand the anomalous data, generate relevant

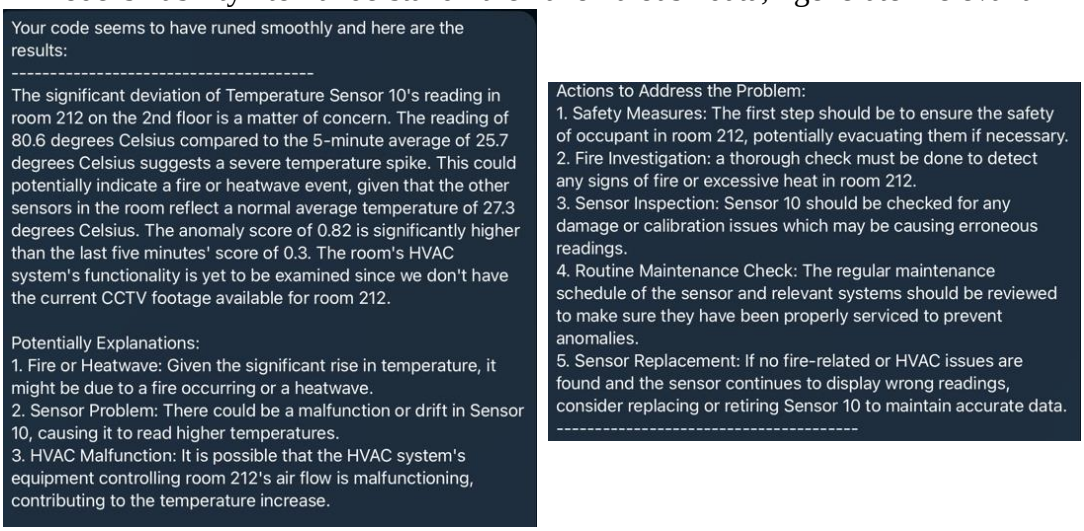

Figure 13. The generated explanation results.

explanations, and deliver them in a human-readable format. The input data typically consists of various sensor readings and relevant information that the model analyzes to identify anomalies. The model then responds with well-structured explanations, offering insights into the detected anomalies, potential reasons behind them, and even suggested courses of action. This clear and intuitive input-output interaction highlights the model's capacity to be a valuable tool in real-world applications where anomaly detection and understanding are of paramount importance.

Chapter 6
Conclusion and Future Research Directions

In conclusion, this research introduces a novel hybrid approach that combines prediction-based anomaly detection using Convolutional Long Short-term Memory with non-parametric dynamic thresholding (ConvLSTM-DT) and the fine-tuned GPT-4 model for Large Language Model (LLM)-based explanation generation. This hybrid approach significantly improves the interpretability and adaptability of anomaly detection models. By integrating the strengths of both prediction-based algorithms and LLMs, the model not only detects anomalies but also provides human-understandable explanations and potential solutions, making it a powerful tool for real-world applications where quick decision-making is crucial.

Furthermore, the implementation of this approach on an edge-cloud architecture enhances its performance by reducing latency and ensuring fast detection. The use of custom fine-tuning with GPT-4 tailored to the specific anomaly detection task further amplifies its accuracy. This study emphasizes the importance of domain-specific fine-tuning in achieving precise and meaningful explanations. As the field of anomaly detection continues to evolve, the presented hybrid approach, along with the use of advanced LLMs, promises to play a pivotal role in improving anomaly detection across various domains, ultimately leading to safer and more efficient operations in complex systems.

For the future work, there is significant potential for enhancing anomaly detection by incorporating commonsense reasoning trees. This addition could further bolster the accuracy and depth of anomaly reasoning and provide a

more comprehensive understanding of complex situations. Moreover, the integration of multimodal sensor fusion, which combines time-series data with visual and auditory inputs, could lead to more holistic anomaly detection solutions capable of capturing anomalies that span various data modalities. Additionally, exploring the integration of reliable control systems to facilitate intelligent decision-making services in real-time scenarios offers an exciting avenue for improving anomaly detection systems, ultimately advancing their capabilities and impact across diverse domains.

References

[1] C. C. Aggarwal, “Outlier Analysis”, 2nd ed. Springer Publishing Company, Incorporated, 2016.

[2] G. Pang, C. Shen, L. Cao, and A. Van Den Hengel, “Deep Learning for Anomaly Detection: A Review,” ACM Comput. Surv., vol. 54, no. 38, p. 1-38, Mar. 2021.

[3] K.-H. Lai, D. Zha, J. Xu, and Y. Zhao, “Revisiting Time Series Outlier Detection: Definitions and Benchmarks,” in NeurIPS Datasets and Benchmarks, 2021.

[4] Z. Z. Darban, G. I. Webb, S. Pan, C. C. Aggarwal, and M. Salehi, “Deep Learning for Time Series Anomaly Detection: A Survey,” ArXiv: 2211.05244, Nov. 2022.

[5] P. Malhotra, L. Vig, G. M. Shroff, and P. Agarwal, “Long Short Term Memory Networks for Anomaly Detection in Time Series,” in The European Symposium on Artificial Neural Networks (ESANN), 2015.

[6] S. Chauhan and L. Vig, “Anomaly detection in ECG time signals via deep long short-term memory networks,” in 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2015, pp. 1–7.

[7] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, “Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding,” in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Jul. 2018, pp. 387–395.

[8] M. Thill, W. Konen, and T. Bäck, “Time Series Encodings with Temporal Convolutional Networks,” in Bioinspired Optimization Methods and Their Applications: 9th International Conference, BIOMA 2020, Nov. 2020, pp. 161–173.

[9] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, “Temporal Convolutional Networks for Action Segmentation and Detection,” ArXiv:1611.05267, Nov. 2016.

[10] E. Dai and J. Chen, “Graph-Augmented Normalizing Flows for Anomaly Detection of Multiple Time Series,” ArXiv:2202.07857, Feb. 2022.

[11] D. L. B.-E. S. A. K. V. Lawrence Wong, “AER: Auto-Encoder with Regression for Time Series Anomaly Detection,” ArXiv:2212.13558, Dec. 2022.

[12] Leilani Hendrina Gilpin, “Anomaly Detection Through Explanations,” MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 2020. Accessed: Nov. 06, 2023. [Online]. Available: https://groups.csail.mit.edu/mac/users/gjs/lgilpin-PhD-EECS-Sept2020.pdf

[13] S. Kumar, P. Tiwari, and M. Zymbler, “Internet of Things is a revolutionary approach for future technology enhancement: a review,” J Big Data, vol. 6, no. 1, p. 111, Dec. 2019.

[14] A. Boni, V. Bianchi, A. Ricci, and I. De Munari, “NB-IoT and Wi-Fi Technologies: An Integrated Approach to Enhance Portability of Smart Sensors,” IEEE Access, vol. 9, pp. 74589–74599, May. 2021.

[15] S. Ugwuanyi, G. Paul, and J. Irvine, “Survey of IoT for Developing Countries: Performance Analysis of LoRaWAN and Cellular NB-IoT Networks,” Electronics (Basel), vol. 10, no. 18, p. 1808, Sept. 2021.

[16] Z. Wang, D. Han, Y. Gong, and Y. Zhao, “Multi-protocol Integration and Intercommunication Technology Based on OPC UA and MQTT,” J Phys Conf Ser, vol. 2173, no. 1, p. 12070, Jan. 2022.

[17] P. Sedgwick, “Pearson’s correlation coefficient,” BMJ, vol. 345, no. jul04 1, pp. e4483–e4483, Jul. 2012.

[18] H. Sak, A. W. Senior, and F. Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling,” in INTERSPEECH, 2014, pp. 338–342.

[19] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” CoRR, vol. abs/1412.3555, 2014.

[20] I. B. K. Y. Utama, R. F. Pamungkas, M. M. Faridh, and Y. M. Jang, “Intelligent IoT Platform for Multiple PV Plant Monitoring,” Sensors (Basel), vol. 23, no. 15, p. 6674, July 2023.

[21] X. SHI, Z. Chen, H. Wang, D.-Y. Yeung, W. Wong, and W. WOO, “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting,” in Advances in Neural Information Processing Systems, Curran Associates, Inc., 2015.

[22] H. E. Robbins, “A Stochastic Approximation Method,” Annals of Mathematical Statistics, vol. 22, pp. 400–407, 1951.

[23] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” CoRR, vol. abs/1412.6980, Dec. 2014.

[24] D. T. Shipmon, J. M. Gurevitch, P. Piselli, and S. T. Edwards, “Time Series Anomaly Detection; Detection of anomalous drops with limited features and sparse examples in noisy highly periodic data,” ArXiv: 1708.03665, Aug. 2017.

[25] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, “Unsupervised real-time anomaly detection for streaming data,” Neurocomputing, vol. 262, pp. 134–147, 2017.

[26] A. Vaswani et al., “Attention Is All You Need,” ArXiv: 1706.03762, Aug. 2023.

[27] P. J. Liu et al., “Generating Wikipedia by Summarizing Long Sequences,” ArXiv: 1801.10198, Jan. 2018.

[28] N. Kitaev and D. Klein, “Constituency Parsing with a Self-Attentive Encoder,” ArXiv: 1805.01052, May 2018.

[29] A. Radford and K. Narasimhan, “Improving Language Understanding by Generative Pre-Training,” 2018.

[30] B. Peng, C. Li, P. He, M. Galley, and J. Gao, “Instruction Tuning with GPT-4,” ArXiv: 2304.03277, Apr. 2023.

Abstract (Korean)

예측 기반 검출기 알고리즘과 대규모 언어 모델을 활용한 설명 가능한 이상 탐지를 위한 혼합 접근 방식

라디토 파자르 파뭉카스
전자공학과
국민대학교 대학원,
서울

이상 감지는 이상 행동을 조기에 식별하기 위해 다양한 영역에서 중요합니다. 이 연구는 다중 센서의 실내 공기 질 데이터에 초점을 맞춰 이상 감지를 위한 예측 기반 검출기와 LLM(Large Language Model)을 결합하는 혁신적인 접근 방식을 소개합니다. 하이브리드 접근 방식은 예측 기반 이상 감지를 위한 비모수 동적 임계값(ConvLSTM-DT)과 인간이 이해할 수 있는 설명을 생성하기 위한 미세 조정 GPT-4 와 컨볼루션 Long-Term Memory 를 통합합니다. 각 센서 매개 변수에는 정확한 예측을 위한 특정 모델이 있습니다. 또한 동적 임계값 및 연속 학습은 급변하는 시나리오에서 이상 감지를 위한 동적 환경, 업데이트 모델 및 비모수 신뢰 구간 설정에 적응합니다. 시스템은 지연 시간을 줄이고 빠른 감지를 위해 에지에 이상 감지를 배포하는 반면 자원 최적화를 위해 LLM 처리가 클라우드에서 발생합니다. 결과는 실시간 의사 결정을 위한 정확한 이상 감지와 잘 설명된 추론을 보여주며, 다양한 응용 분야에서 포괄적인 이상 감지 솔루션에 대한 새로운 접근 방식을 제공합니다.

키워드: 이상 탐지, 대규모 언어 모델을, ConvLSTM-DT, 동적 임계값, 인간이 이해할 수 있는 설명.

Author's Contribution

Journal Papers

- **Radityo Fajar Pamungkas**, Ida Bagus Krishna Yoga Utama, and Yeong Min Jang, 2023. "*A Novel Approach for Efficient Solar Panel Fault Classification Using Coupled UDenseNet*" Sensors 23, no. 10:4918. https://doi.org/ 10.3390/s23104918.
- Ida Bagus Krishna Yoga Utama, **Radityo Fajar Pamungkas,** Muhammad Miftah Faridh, Yeong Min Jang, 2023. "*Intelligent IoT Platform for Multiple PV Plant Monitoring*" Sensors 23, no. 15:6674. https://doi.org/10.3390/s23156674.
- Ones Sanjerico Sitanggang, Van Linh Nguyen, Huy Nguyen, **Radityo Fajar Pamungkas**, Muhammad Miftah Faridh, Yeong Min Jang, 2023. "*Design and Implementation of 2D MIMO OCC System Based on Deep Learning*" Sensors 23, no. 17:7637. https:// 10.3390/s23177637

Conference Papers

- **Radityo Fajar Pamungkas**, Ida Bagus Krishna Yoga Utama, Yeong Min Jang, "*Data Anomaly Detection in IoT System Based on Extended Isolation Forest and Sliding Window*" The 32nd Joint Conference on Communications nad Information (JCCI), 2022.
- Md Morshed Alam, Raihan Bin Mofidul, **Radityo Fajar Pamungkas,** ByungDeok Chung, Yeong Min Jang, "*Abnormal Voltage Regulation Detection in on-Grid PV-ESS System by Support Vector Machine with Principal Component Analysis*" 2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN), Barcelona, Spain, July 2022, pp. 500-503, doi: 10.1109/ICUFN55119.2022.9829685.
- Noh Hyeon Su, **Radityo Fajar Pamungkas,** Yeong Min Jang, "최근 정보통신기술 스마트 홈 에너지 관리 시스템을 위한 AI-IoE 프레임워크에 관한 연구" The 3rd Korea AI Conference, 2022.
- **Radityo Fajar Pamungkas,** Ida Bagus Krishna Yoga Utama, Yeong Min Jang, "*Deep Learning-based Photovoltaic Panels Defect Detection Using Aerial Thermography Imaging*" The 3rd Korea AI Conference, 2022.
- **Radityo Fajar Pamungkas,** Ida Bagus Krishna Yoga Utama, Yeong Min Jang, "*Solar Photovoltaic Modules Fault Classification based on Deep Learning*" The 1st Korea Energy Conference, 2022.

- Muhammad Miftah Faridh, Ida Bagus Krishna Yoga Utama, Duc Hoang Tran, **Radityo Fajar Pamungkas**, Yeong Min Jang, "*Solar Power Generation Forecasting Based on Regional Weather*" The 1st Korea Energy Conference, 2022.
- Ones Sanjerico Sitanggang, **Radityo Fajar Pamungkas,** Ida Bagus Krishna Yoga Utama, Yeong Min Jang, "*Forecasting of Building Electricity Consumption Based on Weather Data*" The 1st Korea Energy Conference, 2022.
- **Radityo Fajar Pamungkas**, Ida Bagus Krishna Yoga Utama, Muhammad Miftah Faridh, Md Morshed Alam, ByungDeok Chung, Yeong Min Jang, "*Forecasting Solar Energy Production using a Hybrid GCN-BiLSTM Model*" 2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Bali, Indonesia, February 2023, pp. 053-056, doi: 10.1109/ICAIIC57133.2023.10067088.
- Ida Bagus Krishna Yoga Utama, Duc Hoang Tran, **Radityo Fajar Pamungkas**, ByungDeok Chung, Yeong Min Jang, "*Predicting Indoor PM2.5 Concentration using LSTM-BNN in Edge Device*" 2023 International Conference on Artificial Intelligence in Information and Communication 10.1109/ICAIIC57133.2023.10067057. (ICAIIC), Bali, Indonesia, February 2023, pp. 211-215, doi: 10.1109/ICAIIC57133.2023.10067057.