Design and evaluation of artificial intelligence models for abnormal data detection and prediction

[1]Hae-Jong Joo, [2*]Ho-Bin Song

Abstract

In today's system operation, it is difficult to detect failures and take immediate action in the case of a shortage of manpower compared to the number of equipment or failures in vulnerable time zones, which can lead to delays in failure recovery. In addition, various algorithms exist to detect abnormal symptom data, and it is important to select an appropriate algorithm for each problem. In this paper, an ensemble-based isolation forest model was used to efficiently detect multivariate point anomalies that deviated from the mean distribution in the data set generated to predict system failure and minimize service interruption. And since significant changes in memory space usage are observed together with changes in CPU usage, the problem is solved by using LSTM-Auto Encoder for a collective anomaly in which another feature exhibits an abnormal pattern according to a change in one by comparing two or more features. did In addition, evaluation indicators are set for the performance evaluation of the model presented in this study, and then AI model evaluation is performed.

Keywords: *Anomaly Data, Anomaly Detection, Anomaly Prediction Model, Artificial Intelligence Prediction Model, LSTM-AutoEncoder*

I. Introduction

The need for a new mobility environment has rapidly emerged due to reorganization of the competitive structure centered on advanced countries, preoccupation of technology, standardization, and environmental protection[1][2]. As Germany, the origin of the automobile industry and the country with the largest technology, decided to completely ban internal combustion engine vehicles from 2030 (October 2016), electrified, zero-emission vehicles are expected to continue to spread in the future[3][4].

Currently, domestic electric vehicles are showing a steady increase, with 347,395 units as of September 2022, a 33% increase from the previous year. The most important factor among many factors for using an electric vehicle is an electric vehicle charging station. As of December 2021, 35,379 electric vehicle charging stations have been installed in Korea to provide service[5][6][7][8].

Since the operation of an electric vehicle charging station is directly related to the use of electric vehicles, failure prevention and recovery of the electric vehicle charging station system are very important.

However, it is difficult to detect failures and take immediate action in the case of shortage of man-month compared to the number of operating equipment or failures in vulnerable time zones, which can lead to delays in failure recovery.

Develop an AI model that can detect anomalies from the server in order to prevent failure recovery delays and provide an efficient electric vehicle charging station operating environment. When using the AI model, it is possible to drastically reduce the recovery time that was previously delayed from recognizing a failure to returning to the site for action. Furthermore, it is possible to reduce the number of on-duty personnel who stay on duty for recognizing and reporting disabilities during vulnerable hours and to secure labor costs. Market research firm Market & Market has predicted that the global market for cloud-based system operation management solutions will grow from about 125 trillion won in 2020 to

[1] *Professor, University of Kangnam, Dept. of KNU Cham-Injae College (hjjoo@kangnam.ac.kr)*

[2*] *Corresponding Author Professor, University of Mokwon, Dept. of Electrical & Electric Engineering (songhb@mokwon.ac.kr)*

Received: Nov. 21, 2023, Revised: Dec. 11, 2023, Accepted: Dec. 13, 2023

about 253 trillion won in 2027. The developed AI model will be able to contribute to cloud-based system operation management solutions through continuous upgrades.

In this paper, we intend to develop an AI model that can detect anomalies to prevent electric vehicle charging station system failure and minimize service interruption. Conduct a load test and create a data set for AI model training by building your own system that can aggregate data. In addition, through data analysis, 'Anomaly' that can be applied in practice is defined. Also, an ensemble-based isolation forest model was used to efficiently detect multivariate point anomalies that deviated from the mean distribution in the data set generated to predict system failure and minimize service interruption. And since significant changes in memory space usage are observed together with changes in CPU usage, the problem is solved by using LSTM-AutoEncoder for a collective anomaly in which another feature exhibits an abnormal pattern according to a change in one by comparing two or more features. did In addition, evaluation indicators are set for the performance evaluation of the model presented in this study, and then AI model evaluation is performed.

II. Related Works

2-1. Classification of anomalous data

(1) Point Anomalies

Point anomalies are data that behave abnormally at a specific point in time when compared to other values in the time series (global outliers) or adjacent points (local outliers). Point outliers can be univariate or multivariate, depending on whether each affects one or more time-dependent variables. The main feature of this type of anomaly is that the time series returns to its previous steady state in a very short time with only a few observations[9].

(2) Context Anomalies

A data point is considered out of context if it is abnormal in a particular context but otherwise normal. Context often exists in the form of additional variable temporal or spatial properties. A Point or collective anomaly can be an anomaly of a Context if it has some Context properties. The most common example of this kind appears in time series data when points fall within the normal range but do not follow the expected time pattern[10].

(3) Collective Anomalies

Individual data points can only occur in related data. Data points in a time series are related to each other. Therefore, collective anomalies can occur within a time series. Collective anomalies are accumulations of unusual data points. The measurements themselves are inconspicuous and within the range of normal values. However, the temporal proximity of individual occurrences to each other or the type of occurrence relative to the rest of the measurement series may indicate an anomaly[11]. One example of the phenomenon of collective anomalies is an abnormal increase in the variance of a certain amount of data points. Time series sequences that deviate from the general pattern belong to collective anomalies.

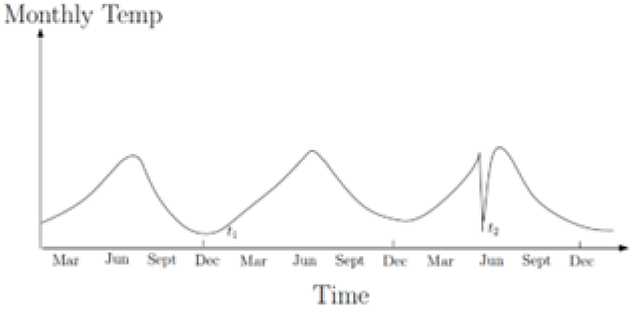


Figure 1. Contextual Anomaly() in monthly temperature

2-2. Anomaly detection in time series data

In this setup of detecting anomalies in time series, an anomaly is an individual instance of a time series that is not unusual in a particular context. This is a widely studied problem in the statistics community. Figure 1 shows an example of a temperature time series showing monthly temperatures in a region over the past several years. A temperature of 35 F may be normal during winter(time) at that location, but the same value may be abnormal during summer(time)[5].

Another set of anomaly detection problems attempts to find subsequences that are anomalous with respect to a given long sequence (time series). Figure 2 is an example of a time series containing unusual subsequences (highlighted regions). The low value itself is not an exception, as it occurs in many different places, but the same low value exists for an unusually long time. This problem corresponds to an unsupervised learning environment due to the lack of labeled data for training, but we assume that most of the long sequences (time series) are normal. If the anomaly subsequence is of unit length, this problem is equivalent to finding context anomalies in a time series with problem setting 1. Abnormal subsequences are also called mismatches.

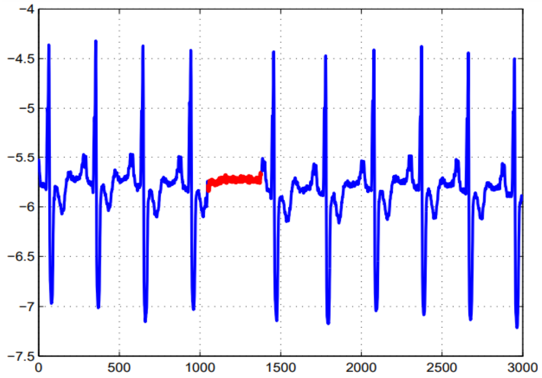


Figure 2. Anomalous of Time Series Data

III. Artificial Intelligence System Model for Anomaly Data Detection

3-1. Definition of system anomalies

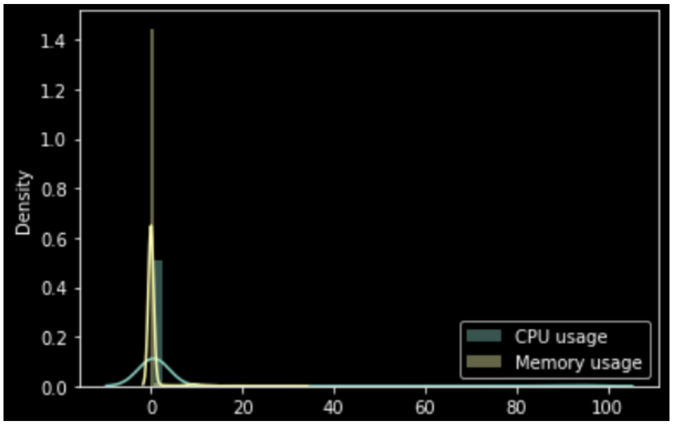


Figure 3. Distribution of performance data sets generated through load testing

As a result of a survey conducted by field server operators, it was confirmed that "abnormality due to congestion of users" occurs most frequently and is difficult to recognize in advance, and Load Runner and a basic Linux stress tool are used to reproduce anomalies caused by congestion of service users. Figure 3 shows the distribution of CPU and Memory usage data in the data set created through the load test: Data is concentrated around '0', and some data away from the dense data is observed.

In addition, through data analysis, excluding some out-liers for a short period of time, an abnormal range in which memory usage also increased when CPU usage increased was confirmed. In this paper, a collective anomaly in which two features simultaneously deviate from normal distribution is defined as an anomaly of time series data, which is not a time point that is significantly different from the previous normal operation or a simple out-lier.

3-2. Acquisition of anomaly data set and establishment of artificial intelligence model verification test bed

One of the major obstacles to anomaly detection is the lack of real data sets containing real anomalies. Open time series data sets, such as NAB (Numenta Anomaly Benchmark) data sets, have limitations in deriving suitable results because they consist of different data from the project, such as sensor data or soil data during the process. Also, it is impossible to create a data set through a service within a set budget.

Therefore, we build a test bed for research and directly create the necessary data set. Through the test bed, it is possible to set the environment and conditions for data set generation in accordance with research, and it can also be used for future research.

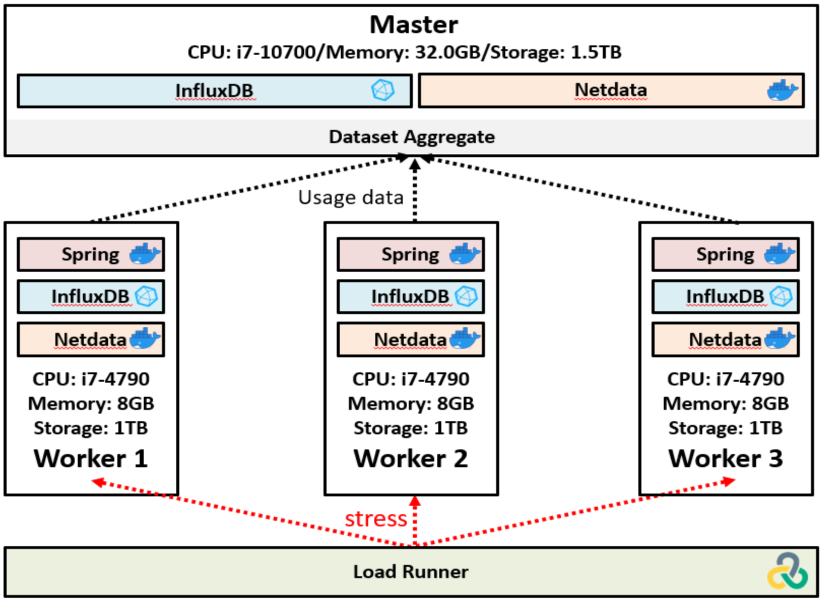


Figure 4. Configuration of the built test bed

In order to create an anomaly data set, a separate data generation system using various solutions is built, and load (stress) is injected into the Spring Web server installed in the worker node with Load Runner on the worker PC, and the real-time CPU of the worker node, Memory usage is stored in influxdb as Netdata. In the built system, worker node real-time data is aggregated into influxdb in the linked master node, and all worker node data can be monitored from the master node.

3-3. AI model for anomaly detection

(1) Isolation Forest

The proposed method, called Isolation Forest or iForest, builds an ensemble of iTrees for a given data set. Above are instances with short average path lengths in iTrees(Figure 5). There are only two variables in this method: the number of trees to build and the subsampling size. iForest's detection performance converges quickly with a very small number of trees and requires only a small subsampling size to achieve high detection performance with high efficiency. Besides the main difference between isolation and profiling, iForest is distinguished from existing model-based, distance-based and density-based methods in the following ways.

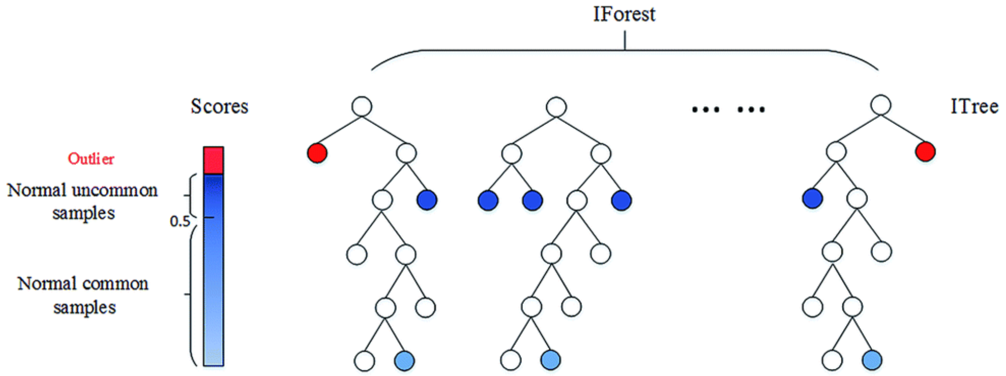


Figure 5. Concept Diagram of Isolation Forest

a) The isolated nature of iTrees allows building partial models and exploiting subsampling to an extent not feasible with conventional methods. You don't need to build much of the iTree that separates the vertices because it's not needed for anomaly detection. Smaller sample sizes produce better iTrees because of reduced swapping and masking effects.
b) iForest does not use distance or density measurements to detect anomalies. This eliminates the major computational cost of distance calculation in all distance-based and density-based methods.
c) iForest has linear time complexity with low constant and low memory requirements. The most performant existing methods only achieve approximate linear time complexity with high memory usage.
d) iForest has the ability to scale to handle high-dimensional problems with very large data sizes and many unrelated properties.

(2) LSTM-AutoEncoder

Various algorithms exist to detect anomalies, and it is important to select an appropriate algorithm for each problem. We use an ensemble-based isolation forest model to efficiently detect multivariate point anomalies that deviate from the mean distribution in the generated data set. In addition, since a significant change in memory usage is observed together with a change in CPU usage, LSTM-AutoEncoder is used for a collective anomaly in which another feature exhibits an abnormal pattern according to a change in one by comparing two or more features.

AutoEncoder is an unsupervised neural network that aims to learn the best encoding-decoding scheme from data(Figure 6). It generally consists of an input layer, an output layer, an encoder neural network, a decoder neural network, and a latent space. When data is fed into the network, the encoder compresses the data into the latent space while the decoder decompresses the encoded representation into the output layer. The encoded-decoded output is compared with the initial data and the error is propagated back through the architecture to update the weights of the network. In particular, to obtain a given encoded representation, the decoder reconstructs this representation and provides an output. AutoEncoders are trained by minimizing reconstruction errors.

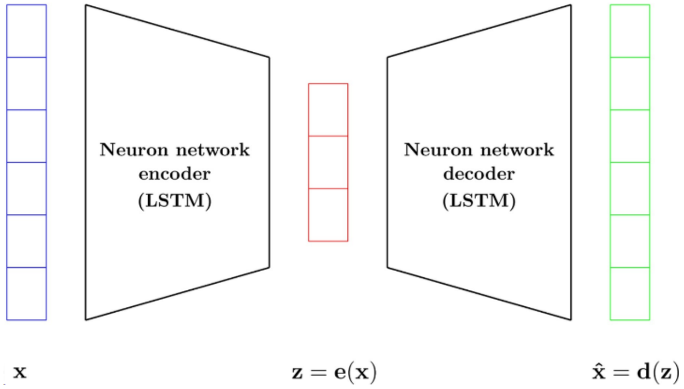

Neuron network encoder (LSTM)
Neuron network decoder (LSTM)
$\mathbf{x}$
$\mathbf{z} = \mathbf{e}(\mathbf{x})$
$\hat{\mathbf{x}} = \mathbf{d}(\mathbf{z})$

Figure 6. Concept Diagram of LSTM-AutoEncoder

The main purpose of AutoEncoder is not to simply copy input to output. By constraining the latent space to have dimensions smaller than the input, the AutoEncoder should learn the most salient features of the training data. In other words, an important feature of the AutoEncoder design is to reduce the data dimensionality while retaining key information in the data structure.

The LSTM AutoEncoder proposed in this paper refers to an AutoEncoder in which both encoder and decoder are LSTM networks. The ability of LSTMs to learn patterns in data over long sequences makes them suitable for time series prediction or anomaly detection. That is, the use of LSTM cells is to capture time dependencies in multivariate data. Encoder-decoder models trained using only normal sequences can be used to detect anomalies in multivariate time series. The encoder-decoder only saw normal instances during training and learned how to reconstruct them. If you input an anomalous sequence, it may not reconstruct well, resulting in a higher error. This has practical implications as unusual data may not always be available or it is impossible to cover all types of such data.

The LSTM-AutoEncoder Folw Chart is shown in Figure 7, where timestops=30, num_features=2, return_sequences=True (each cell emits a signal per time step), return_sequences=False (only the last type stem cell emits a signal) The result is:

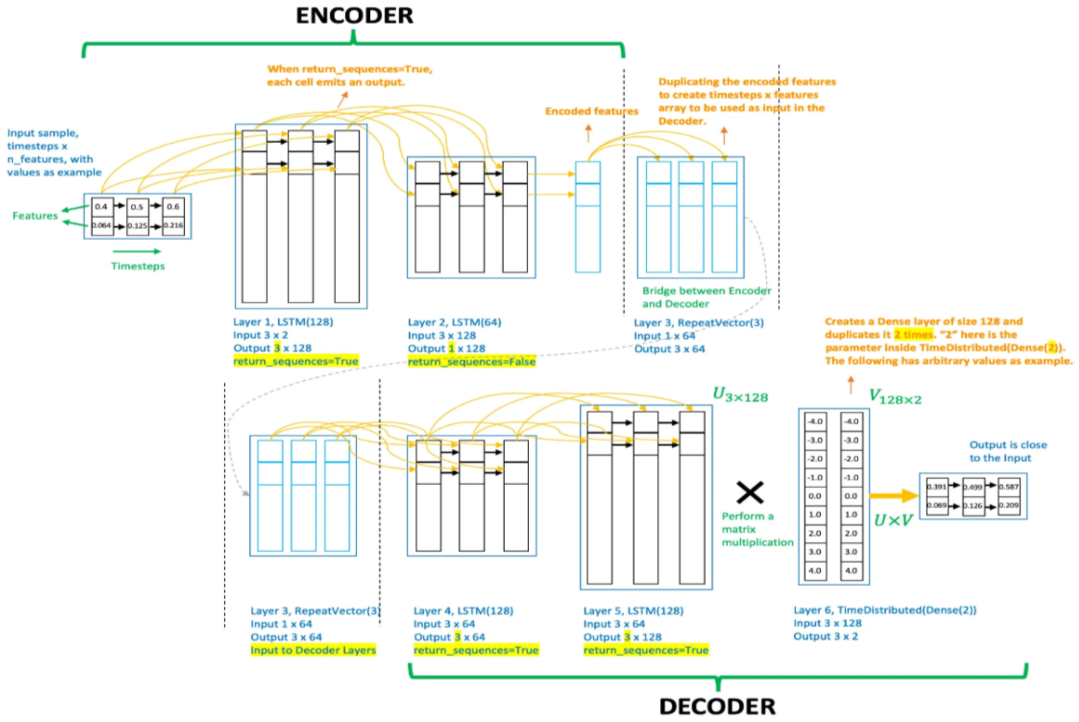


Figure 7. LSTM-AutoEncoder Flow Chart

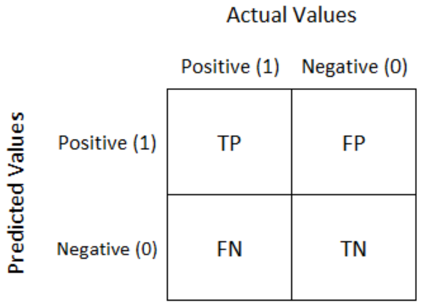
① LSTM 128 reads the input data and outputs 128 features with 30 time steps for each since return_sequences=True.
② LSTM(64) receives 30x128 input from layer 1 and reduces the size to 64.
③ Since return_sequences=False, a vector of size 1x64 is output.
④ This output is the encoded feature vector of the input data.
⑤ RepeatVector(30) replicates the feature vector 30 times.
⑥ The RepeatVector layer acts as a bridge between the encoder and decoder modules.
⑦ The decoder layer is the reverse order of the encoder.
⑧ The TimeDistributed layer creates a vector with the same length as the number of features output in the previous layer.
⑨ Therefore, the TimeDistributed layer creates 128 vectors and replicates them twice (=num_features).
⑩ The output of layer 5 is a 30(timesteps)×128 array

⑪ The output of layer 6 is a 128×2 array
⑫ Matrix multiplication between layers 5 and 6 produces a 3×2 output.
⑬ The goal is to make the 3×2 output as close as possible to the input.
⑭ In the above model, the input and output dimensions are matched.

IV. Evaluation of the proposed system AI model

4-1. Evaluation Indicators

The definition of the confusion matrix for discriminating anomalies proposed in this paper is shown in Figure 8.



Predicted Values	Actual Values: Positive (1)	Actual Values: Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

Figure 8. Confusion Matrix

Here,
① true positive: prediction is true when it is actually true
② true negative: prediction is also false when it is actually false
③ false positive: prediction is true when it is actually false
④ false positive: prediction is false when it is actually true

At this time, through the confusion matrix, precision, accuracy, recall, and f1 score can be derived as follows.

● Precision($\frac{TP}{TP+FP}$) : A measure that describes how true predictions turn out to be.

● Accuracy($\frac{TP+TN}{Total}$) : A measure that describes how accurate the predictions were in the entire data set.
● Recall($\frac{TP}{TP+FN}$) : A measure that describes how true the predictions were for all data points that were actually true.
● F1 score($\frac{2*prcision*recall}{prcision+recall}$) : harmonic mean of precision and recall

4-2. Model Evaluation

Through the AI model system proposed in this paper, data preprocessing in isolation forest and LSTM-AutoEncoder, Train Dataset task that includes only normal data in the entire data set, and Test Dataset task that includes both normal and abnormal data in the entire data set 9 gave the same result.

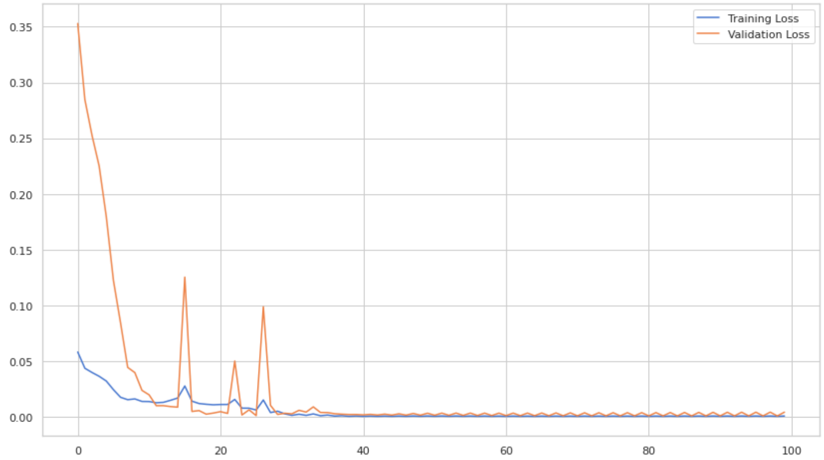


Figure 9. The Result of Train, Validation Loss

The prediction result for the original data in the proposed AI model is shown in Figure 10, and the loss-based anomaly prediction result is shown in Figure 11.

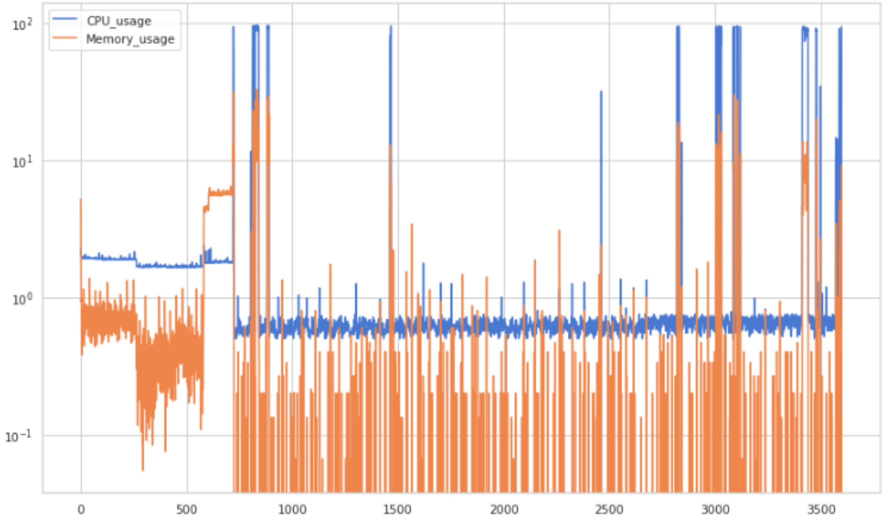


Figure 10. Original data prediction result

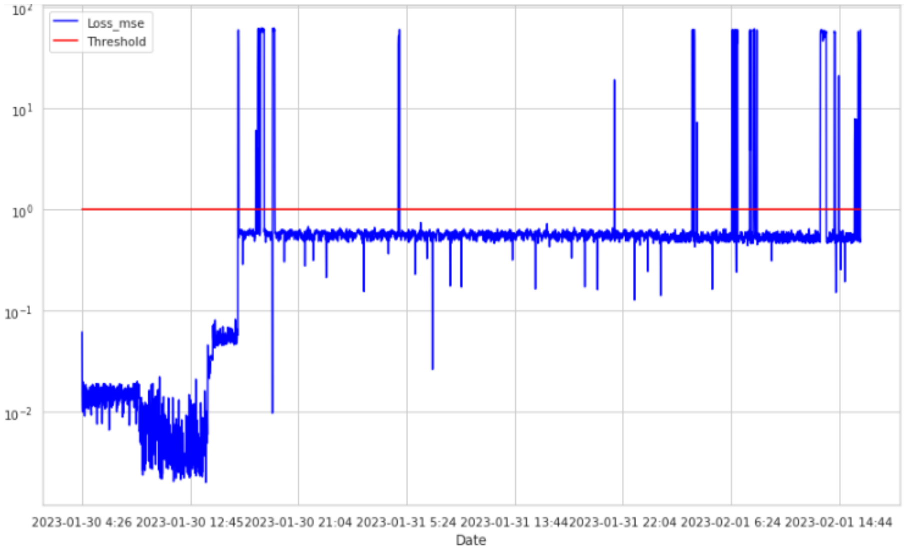


Figure 11. Loss-based anomaly prediction result

Comparing the result graphs in Figure 10 and Figure 11, it is possible to confirm the detection of abnormal data that exceeds the threshold.

V. Conclusion

In this paper, an ensemble-based isolation forest model was used to efficiently detect multivariate point anomalies that deviated from the mean distribution in the data set generated to predict system failure and minimize service interruption. And since significant changes in memory space usage are observed together with changes in CPU usage, the problem is solved by using LSTM-AutoEncoder for a collective anomaly in which another feature exhibits an abnormal pattern according to a change in one by comparing two or more features. did In addition, evaluation indicators are set for the performance evaluation of the model presented in this study, and then AI model evaluation is performed.

The expected effect of this study is that by applying the proposed AI model, anomalies can be detected not only in electric vehicle charging station systems, but also in about 5,000 domestic private companies/public institutions equipped with large-scale networks and servers. In particular, in the case of public institutions, the marketability is sufficient because they comply with the standard of introducing domestic software with priority.

As for the utilization method of this study, if potential anomalies due to congestion of users of the server are discovered, failures can be prevented in advance by taking precautionary measures against expected failures and problem resources before actual service failures occur. In addition, it reduces the operational/analytical man-hours required to set sophisticated thresholds for each time slot when resources and usage increase, and supports efficient operation.

VI. References

[1] M. S. Islam, W. Pourmajidi, L. Zhang, J. Steinbacher, T. Erwin and A. Miranskyy, "Anomaly Detection in a Large-Scale Cloud Platform", 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), Madrid, ES, 2021

[2] Tanja Hagemann, Katerina Katsarou, "Reconstruction-based anomaly detection for the cloud", 2020 4th International Conference on Cloud and Big Data Computing, 2020

[3] S. N. Shirazi, S. Simpson, A. Gouglidis, A. Mauthe and D. Hutchison, "Anomaly Detection in the Cloud Using Data Density", 2016 IEEE 9th International Conference on Cloud Computing (CLOUD),

San Francisco, CA, USA, 2016

[4] D. Smith, Q. Guan and S. Fu, "An Anomaly Detection Framework for Autonomic Management of Compute Cloud Systems”, 2010 IEEE 34th Annual Computer Software and Applications Conference Workshops, Seoul, Korea (South), 2010

[5] X. Jin and X. Qiu, "An Adaptive Anomaly Detection Method for Cloud Computing System”, 2022 IEEE 5th International Conference on Electronics Technology (ICET), Chengdu, China, 2022

[6] M. A. Bashar and R. Nayak, "TAnoGAN: Time Series Anomaly Detection with Generative Adversarial Networks”, 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, ACT, Australia, 2020

[7] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante and K. Veeramachaneni, "TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks”, 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 2020

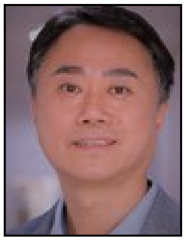
[8] A. Lavin and S. Ahmad, "Evaluating Real-Time Anomaly Detection Algorithms -- The Numenta Anomaly Benchmark”, 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 2015

[9] Kim, S., Choi, K., Choi, H.-S., Lee, B., & Yoon, S., “Towards a Rigorous Evaluation of Time-Series Anomaly Detection”, AI Conference on Artificial Intelligence, 2022

[10] Deepthi Cheboli, Anomaly Detection of Time Series, IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master Of Science, 2010

[11] Subutai Ahmad, Alexander Lavin, Scott Purdy, Zuha Agha, “Unsupervised real-time anomaly detection for streaming data”, Neurocomputing, Volume 262, 2017

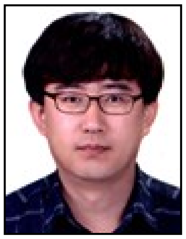
Authors



Hae-Jong Joo

2008 : Ph.D of Computer Education, Cumberland University
2010 : Ph.D of Computer Engineering & Science, Myongji University

Research Interests : Data Science, Intelligence SW, Data Mining, Metaverse Platform, Video Big-Data QC



Ho-Bin Song

2006 : Ph.D Of Electrical Engineering, Myongji University

Research Interests : Big-data, AI, Power Electronics, Electric Vehicle