Model Creation Method for Anomaly Detection in Refrigeration and Air Conditioning Systems

Toshiaki Hirata
Tokyo Information Design Professional University
hirata@tid.ac.jp

Yusuke Kuriyama
Computron Co., Ltd.
y-kuriyama@computron.co.jp

Takashi Hinohara
Computron Co., Ltd.
hinohara@computron.co.jp

Sumiei Takahashi
DAIKOU GIKEN Co., Ltd.
smi1006@daikou-giken.co.jp

Ken-ichi Yoshida
University of Tsukuba
yoshida.kenichi.ka@u.tsukuba.ac.jp

Toshiro Ogasawara
Saitama Industrial Promotion Public Corporation
ogasawara.toshiro@saitama-j.or.jp

Masaaki Go
Saitama Industrial Promotion Public Corporation
go.masaaki@saitama-j.or.jp

***Abstract*—Predicting anomalies and diagnosing various machines using IoT technology have been widely studied. Refrigeration and air-conditioning systems are among them. Anomaly detection systems for refrigeration and air-conditioning systems often work with many distributed devices, posing a challenge in creating a learning model. Additionally, the data collection period to create learning models needs to be shortened. In this study, we propose a method to automate anomaly detection model creation for a large number of distributed refrigeration and air-conditioning systems and to improve the efficiency of the operation of the diagnosis system. In addition, we propose a fine-tuning method that creates a learning model for the target device with minimal learning data based on a learning model created by another device, thus reducing the learning model creation period.**

***Index Terms*—IoT, Anomaly detection, AI, Data mining**

I. Introduction

Machines installed in buildings and production facilities constitute critical infrastructure in modern society. Anomalies in such machines significantly affect the environment and business continuity. Therefore, anomaly prediction and diagnosis are essential for preventing such problems. The authors utilized Internet of Things(IoT) technology to install a data-collection and diagnosis system for existing air conditioners, focusing on anomaly detection technology in air conditioning systems [1]. Our research discussed strategies for using low-cost sensors in noisy environments at the right locations and ensuring precise tuning of seasonal diagnosis models.

This study focuses on creating an anomaly-detection model for refrigeration and air-conditioning systems. Refrigeration and air-conditioning systems for small- and medium-sized facilities are often distributed and installed in many facilities, making it crucial to improve the efficiency of creating anomaly detection models for such a large number of systems. In this paper, we propose an automatic anomaly detection model creation method for refrigeration and air-conditioning systems to improve the efficiency of the diagnostic system operation. In addition, we propose a fine-tuning method that creates a learning model for the target device with minimal learning data based on a learning model created by another device. This approach significantly reduced the learning model creation period, thereby greatly assisting the operation of diagnostic systems.

This paper discusses the model creation procedure and fine-tuning method. The remainder of this paper is organized as follows. Section 2 discusses related studies in this domain. Section 3 explains the problem, while sections 4 and 5 describe our solution approach. Section 6 presents experimental results, and finally, section 7 summarizes our findings.

II. Related work

Extensive research has been conducted on anomaly detection in machines. Ito et al. [2] proposed a method for detecting UNIX command-sequence anomalies using a combination of an online sequential extreme learning machine [3] and an autoencoder [4]. Harada et al. [5] detected anomalies in an aquarium management system using a local outlier factor [6]. Shi et al. [7] realized a network intrusion detection system using a support vector machine [8] and deep belief network [9] with semi-supervised learning. However, these studies did not address the efficiency of the learning model creation.

Additionally, the automatic creation of learning models in machine learning has been studied. Lee et al. [10] proposed a support tool for creating a deep learning model that realizes network creation through a Graphical User Interface and automatic tuning of hyperparameters. Thar et al. [11] proposed a model generation framework for optimal resource management to predict and cache popular content to reduce operational costs in MVNOs. Okano [12] proposed a machine learning analysis method to express requirements in a decision table and automatically generate a model using a decision tree. These studies focused on network model generation and parameter tuning and not on automating model creation from training data.

In this study, we propose a method to automate the creation of an anomaly detection model for refrigeration and air-conditioning systems. Furthermore, we propose a fine-tuning

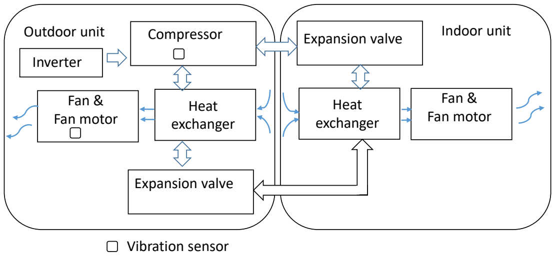

Fig. 1. Refrigerator and IoT sensors

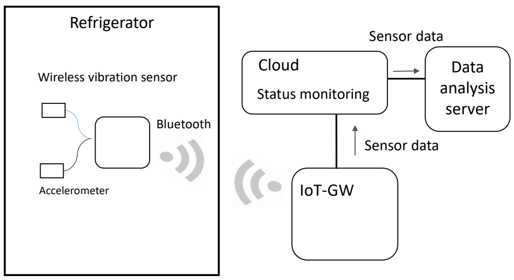

Fig. 2. Data collection subsystem

method designed to create a learning model for the target device within a short period.

III. Refrigerator and IoT sensors

Figure 1 shows the configuration of the refrigerator and IoT sensors. In this study, we focus on refrigerators comprising indoor and outdoor units. A compressor, which is the power source, compresses the refrigerant to increase its temperature and pressure. The expansion valve lowers the temperature and pressure by passing the refrigerant through a narrow gap, thereby automatically adjusting the flow rate and superheat. The fan motor moves the indoor and outdoor air, and the heat exchanger exchanges heat between the refrigerants and indoor and outdoor air. The temperature control mechanism controls the operation of the compressor using an inverter and a temperature controller.

During the experiment, vibration sensors were installed to detect equipment failures at an early stage. Based on preliminary experiments [1], the compressor and outdoor fan motor were selected as the main parts on which the IoT sensors were installed. Vibration data were collected using a 3-axis vibration sensor, each installed in the x-, y-, and z-axis directions.

IV. Anomaly detection system

A. Data collection subsystem

Figure 2 shows the configuration of the data collection subsystem. The data collected from each sensor were aggregated in the IoT-GW(Gateway) and transmitted to a cloud-based data analysis server. In the cloud, sensor data can be monitored in real-time, and data analysis for anomaly detection is performed on the data analysis server.

TABLE I
Data collection conditions

Sensor type	Measured amount(unit)	Communication	Sampling rate
Vibration sensor	Acceleration(m/s^2) (3axis)	Bluetooth	50Hz

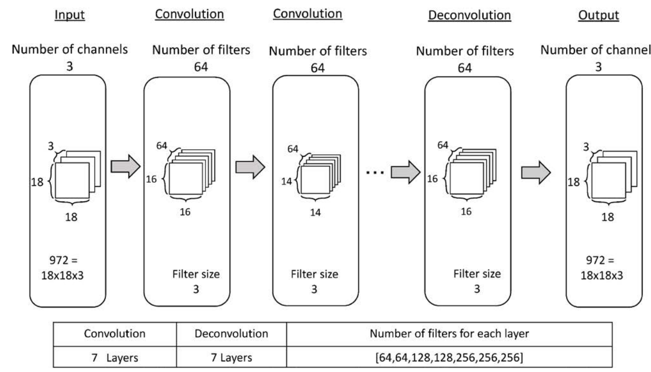

Fig. 3. Configuration of Convolutional Autoencoder

Table I lists the data-collection conditions for each sensor. Based on the sensor specifications, the sampling rate was set to 50 Hz for the vibration sensor. According to the device specifications, Bluetooth was used as the communication method for the vibration sensor.

B. Data analysis procedure

We analyzed data using a convolutional autoencoder (CAE) [13]. An autoencoder [4] is a type of neural network that creates a normal operation status model. The CAE is a type of autoencoder that has convolution and deconvolution layers for data (See Figure 3). As an input for the CAE, we used 324 points power spectra obtained using a fast Fourier transform (FFT) on each of the three axes vibration values. Subsequently, the size and number of filters were determined based on preliminary experiments [1].

When the input data represents normal operation status, the difference between the output and input is small. We can detect the anomaly status by comparing the difference with a specific threshold. The anomaly score E for the predicted data x and the observed data x' is defined as the mean square error as follows:

$$E = \frac{\sum_{n=1}^{N}(\| x_n - x'_n \|)^2}{N}$$

Here the input dimension N of x is 972.

The learning model adopts a method that accounts for seasonality, as proposed in our previous study [1]. A different learning model was used for each season to improve the anomaly detection accuracy, and the model with the lowest anomaly score was selected from the various learning models.

V. Learning model generation method

A. Automatic model creation method

The basic concept of the automatic model creation method is to create a learning model using data from the initial model and subsequently add data when the anomaly score exceeds the threshold during normal operation. Specifically, we added data in the range where predetermined M points consecutively exceeded the threshold. The threshold value for anomaly detection was set based on multiple values of the average value of the anomaly score during normal operation and a value that can be distinguished between normal and abnormal conditions in failure tests. The procedure for creating the learning model is as follows:

Procedure 1

The procedure for creating the first learning model is shown in Figure 4 (top left).

- Step1: The time-series data from the first period is used as the initial training data.
- Step2: First learning model is created using the initial training data.
- Step3: Determine the threshold as previously described in this subsection.

Procedure 2

The procedure for updating the learning model is shown in Figure 4 (top right and bottom).

- Step1: After creating the first learning model(previously learned model), collect data in the range where predetermined M points successively exceed the threshold among the new evaluation data (time-series data).
- Step2: Update the first learning model with data collected in Step1.
- Step3: Predict all past time-series data using the updated first learning model.
- Step4: If the predetermined M points of the prediction results in Step3 do not consecutively exceed the threshold, the updated model is adopted.
- Step5: If the predetermined M points of the prediction results in Step3 consecutively exceed the threshold, a second learning model(newly created model) is created using the data extracted in Step1.
- Step6: Determine a common threshold for the first and second learning model as described earlier in this subsection.

In terms of the amount of learning data, the proposed procedure is expected to require less data than manual learning model creation (manual method). Before designing the procedure explained above, we used a manual method that uses whole one-day data to update the CAE model. Not only does the manual method require large data (i.e., one-day data), but it also requires human assessment of updated models. Since the above procedure requires data that the previous model classified the data period as abnormal, the amount of required data is less than the manual method. It also reduces the man-hour for assessment.

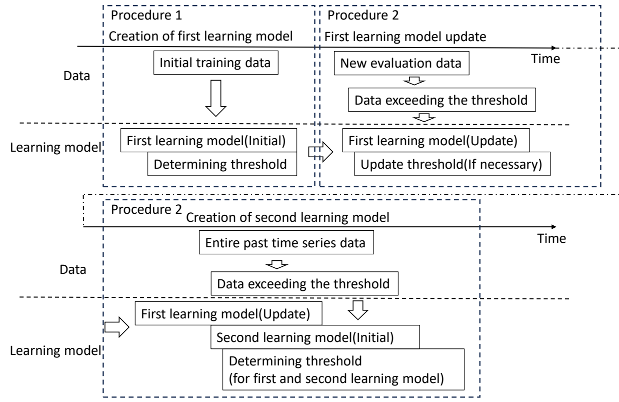

Fig. 4. Automatic model creation procedure

TABLE II
Target network for re-learning on fine-tuning

Layers	Convolution/Deconvolution	Number of filters	Re-learn
1 - 2	Convolution	64	Yes
3	Convolution	128	Yes
4	Convolution	128	No
5 - 7	Convolution	256	No
5 - 7	Deconvolution	256	No
4	Deconvolution	128	No
3	Deconvolution	128	Yes
1 - 2	Deconvolution	64	Yes

B. Fine-tuning method

A learning model was created from the short-term learning data of another device, based on the learning model created for a device that collects long-term data using the fine-tuning method [14]. In this study, we set the target network for re-learning during fine-tuning to the specifications as listed in Table II. Specifically, one to three layers from both the input and output layers were chosen for re-learning.

VI. Experimental results

A. Test items and condition

Table III provides an overview of the field tests. For three months, we collected data for normal operation status and data on the failure test day. For the failure test, the air volume was decreased by 40% after the inhalation port was closed while maintaining a fixed temperature setting of -23 degrees.

B. Automatic model creation method

During the continuous operational test period, we generated a learning model according to the procedure described in section V. The threshold value was five times the average normal score to distinguish between normal and abnormal conditions. This threshold was based on the results of the experiment, including the failure test. In addition, parameter

TABLE III
TEST ITEMS AND CONDITION

#	Test type	Date	Test condition
1	Failure test	October 27, 2022	Air volume reduction due to closure of inhalation port of indoor outdoor unit (Temperature setting:-23 °C)
2	Continuous operation test	From September 16, 2022 to December 26, 2022	24-hour continuous operation excluding failure test days (Temperature setting:-23 °C)

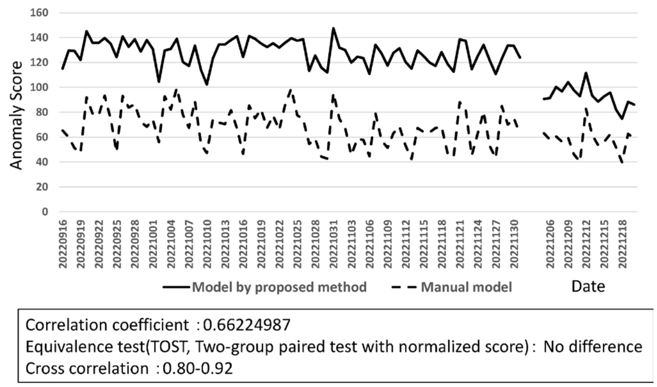

Fig. 5. Transition in CAE anomaly score daily average (compressor)

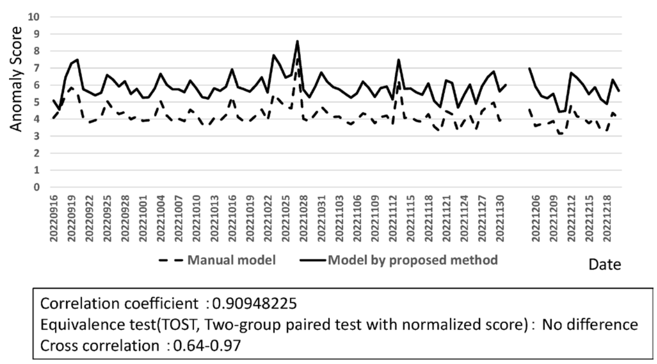

Fig. 6. Transition in CAE anomaly score daily average (outdoor unit fan motor)

M, indicating the number of continuous threshold-exceeding points for assessing abnormalities, was set to 100. For the evaluation, the learning model created by the proposed method and the manually created learning model were compared for the anomaly score of the data during the continuous operation period and the failure test.

Figures 5 and 6 compare the transition in CAE anomaly score daily average of both methods for the compressor and outdoor unit fan motor, respectively. In addition, the correlation coefficient, Two One-Sided Tests(TOST) using two-group paired test with the normalized score, and cross-correlations were calculated to evaluate the equivalence of the results of both methods. The results indicated that both methods were almost equivalent in daily average transition.

Figures 7 and 8 show the transition in the failure test CAE anomaly scores for the compressor and outdoor unit fan motor using both methods, respectively. The correlation

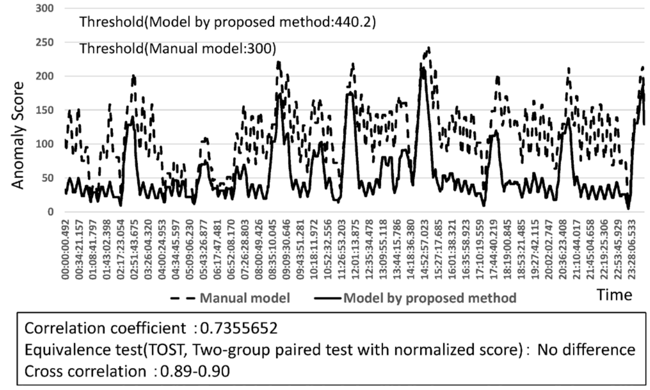

Fig. 7. Compressor data analysis in anomaly test

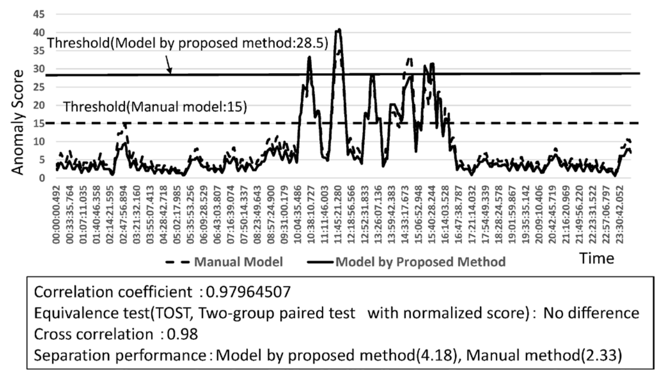

Fig. 8. Outdoor unit fan motor data analysis in anomaly test

coefficients, TOST, and cross-correlations were calculated. While the compressor was unable to detect anomalies, the overall trend was similar in terms of the correlation coefficients and equivalence test results. The proposed method exhibited better separation performance for the fan motor (Figure 8), but the overall transition trend was the same in terms of the correlation coefficient and equivalence test results. The separation performances were determined by the ratio of the maximum value of the abnormal section to that of the normal section during the test period.

Figures 9 and 10 show the transition in the cumulative number of added learning data for the compressor and outdoor unit fan motor, respectively. In both cases, the proposed method required less learning data than the manual method. Although the timing of the model update is slightly different in the proposed method than in the manual method, the learning model was updated with a minimal data, as explained in Section V.

Table IV provides a comparison of the man-hours required by both methods during the test period. In the proposed method, the learning data for the initial model creation must be prepared manually; however, the model update can be executed automatically (accuracy checks after updates must be performed manually). In the manual method, the results of the evaluation of the observational data must be reviewed daily to determine if an update of the model is necessary. How-

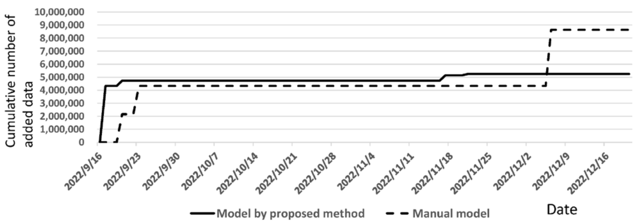

Fig. 9. Transition in the cumulative number of added learning data(compressor)

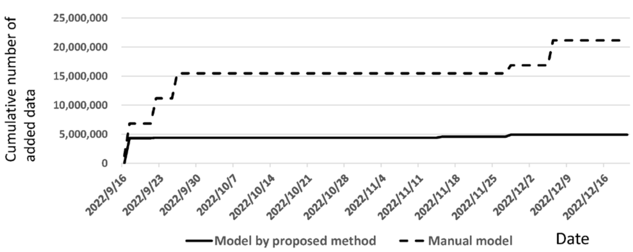

Fig. 10. Transition in the cumulative number of added learning data(outdoor unit fan motor)

TABLE IV
COMPARISON OF MAN-HOURS BETWEEN THE PROPOSED METHOD AND THE MANUAL METHOD

	Manual		Proposed method	
	Number of executions	man-hours (Hour)	Number of executions	man-hours (Hour)
Initial model creation & threshold determination	1	1.0	1.0	0.75
Model update & threshold determination	3	4.0	4	2.0
Daily observation data evaluation	40	5	0-40	0.0-5.0
Total	-	10.0	-	2.75-7.75

ever, daily evaluation result checks are unnecessary because the proposed method automates the model update decision. Consequently, the proposed method can reduce man-hours by 22%–85% compared with the manual method.

The proposed method determined the threshold setting parameter (a multiple of the average value of the normal score) and the number of consecutive threshold-exceeding points M based on past experimental results [1]. Optimal parameters should be established in future studies.

C. Fine tuning method

Based on the air conditioner learning model developed in our previous study [1], we created a fine-tuning model for refrigerators in this study. The network target for re-learning during fine-tuning was set as specified in Table II, and the number of training data points was set as shown in Table V. In this experiment, the effect of fine-tuning was evaluated independently; therefore, the automatic model creation method described in subsection B was not used. However, it was also possible to perform the automatic model creation method and the fine-tuning method in combination.

Figures 11 and 12 show the transition in CAE anomaly score daily average in the compressor and outdoor unit fan motor, respectively. Each graph compares the results evaluated using the fine-tuned model and the model created from scratch (the original model) for the same device. Because the original and fine-tuning models used different training data, the anomaly scores were normalized for comparison.

For the compressor, the original and fine-tuning model were almost equivalent in terms of the overall transition trend of the CAE anomaly score, correlation coefficient, and equivalence test (TOST) results. In contrast, for the fan motor, the equivalence between the original and fine-tuning model was slightly lower in terms of the overall transition trend of the CAE anomaly score, correlation coefficient, and equivalence test results. However, these differences are not significant for practical use.

Additionally, as shown in Table V, the fine-tuning model required only 0.03%–0.01% of the training data compared with the original model.

TABLE V
THE NUMBER OF TRAINING DATA

	Number of training data for base model	Number of training data for fine tuning
Compressor	8,640,015	2,777
Outdoor unit fan motor	21,174,008	2,777

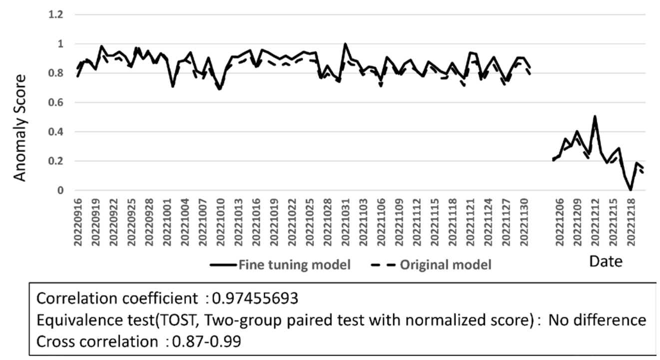

Fig. 11. Transition in CAE anomaly score daily average (Compressor)

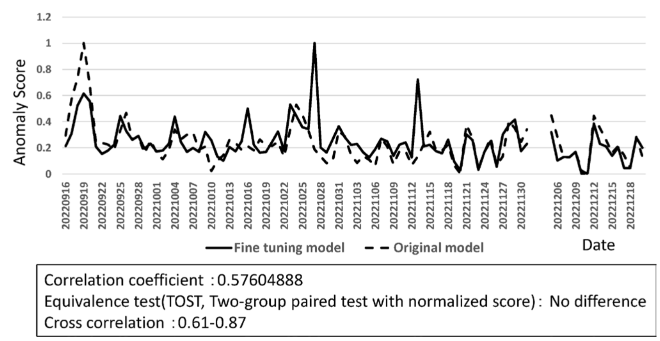

Fig. 12. Transition in CAE anomaly score daily average (Outdoor unit fan motor)

VII. Conclusion

In this study, we proposed a method to automate the creation of an anomaly detection model for refrigerators. Furthermore, we proposed a fine-tuning method to create a learning model for the target device. The characteristics of the proposed method are as follows:

- Enhanced efficiency in the learning model creation for anomaly detection.
- Reduction in the learning model creation period through fine-tuning.

Experimental results have been published in this paper, demonstrating the effectiveness of the proposed method.

References

[1] T.Hirata,K.Yoshida,K.Koido,S.Takahashi,Anomaly detection in air conditioners using IoT technologies,Proceedings of COMPSAC2021, pp.1553-1559, Jul.2021.

[2] R. Ito, M. Tsukada, M. Kondo, H. Matsutani, An Adaptive Abnormal Behavior Detection using Online Sequential Learning, 2019 IEEE CSE and IEEE EUC, pp.436-440.

[3] N.-Y. Liang, G.-B. Huang, P. Saratchandran,and N. Sundararajan, Fast and Accurate Online Sequential Learning Algorithm for Feedfoward Networks, IEEE Transactions on Neural Networks, vol.17, no.6, pp.1411-1423, Nov. 2006.

[4] G.E. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science, Vol.313, No.5786, pp.504-507, 2006.

[5] Y. Harada, Y. Yamagatay, O. Mizuno, E. Choiy,)Log-based Anomaly Detection of CPS Using a Statistical Method, IWESEP2017,March 13 2017.

[6] M.M. Breunig, H.-P. Kriegel, R.T. Ng, and J. Sander, LOF:Identifing Density-Based Local Outliers, Proc. of SIGMOD, pp.1-12, 2000.

[7] N. Shi, X. Yuan, J. Hernandez, K. Roy, A. Esterline, Self-learning Semi-supervised Machine Learning for Network Intrusion detection, 2018 CSCI, pp.59-64.

[8] S. Ravi, Semi-supervised Learinig in Support Vector Machines. Poject Report COS521, 2014.

[9] M. Nadeem, O. Marshall, S. Singh, X. Fang and X. Yuan, Semi-supervised Deep Neural Network for Network Intrusion Setection. CCERP 2016.

[10] K. M. Lee, K. S. Hwang, K. I. Kim, S. H. Lee, K. S. Park, A Deep Learning Model Generation Method for Code Reuse and Automatic Machine Learning, RACS ’18, October 9-12, 2018, Honolulu, HI, USA.

[11] K. Thar, T. Z. Oo, Y. K. Tun, D. H. Kim, K. T. Kim, C. S. Hong, A Deep Learning Model Generation Framework for Virtualized Multi-Access Edge Cache Management, IEEE Access, VOLUME 7, 2019, pp.62734-62749.

[12] M. Okano, A study on automatic software model generation by machine learning Decision tables and a decision tree analysis approach, IPSJ/SIGSE Software Engineering Symposium (SES2019), pp.199-206(in Japanese).

[13] A. Alqahtani, X. Xie, J. Deng, M.W.Jones: A Deep Convolutional Auto-encoder with embedded clustering,2018 25th IEEE International Conference on Image Processing.

[14] K. P. Murphy, Probabilistic Machine Learning: An Introduction, MIT Press, 2022.