2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Refrigerated Showcase Fault Detection by an Autoencoder with Coin Betting and Maximum Correntropy Criterion

Masato Igarashi and Yoshikazu Fukuyama
Graduate School of Advanced Mathematical Sciences
Meiji University
Tokyo, Japan
yfukuyama@meiji.ac.jp

Yuichi Shimasaki, Yuto Osada, Kenya Murakami, Tatsuya Iizaka, Adamo Santana, and Tetsuro Matsui
Corporate R&D headquarters
Fuji Electric Co., Ltd.
Tokyo, Japan

***Abstract*—This paper proposes refrigerated showcase fault detection by an autoencoder with coin betting and Maximum Correntropy Criterion (MCC). In actual situations, showcase data may include outliers which are incorrectly stored data. Radio frequency interference or incorrect sensor setting cause the outliers. When the outliers are included in learning data, the conventional autoencoders using least square error (LSE) may be influenced by the outliers. On the other hand, even when the outliers are included in learning data, autoencoders using MCC can reduce influence from the outliers. Moreover, the conventional artificial neural networks utilize various learning algorithms such as stochastic gradient descent with momentum (SGDM), adaptive moment estimation (Adam), adaptive gradient algorithm (AdaGrad). These methods have hyperparameters related to a learning rate. Since the hyperparameters affect learning strongly, it is required to tune the hyperparameters appropriately and the tuning requires engineering costs. On the other hand, coin betting can automatically tune a learning rate appropriately while learning. Therefore, the coin betting is expected to reduce the engineering costs for parameter tuning. Practicability of the proposed method is verified by comparison with an autoencoder with SGD and LSE, an autoencoder with SGDM and MCC, an autoencoder with Adam and MCC, and an autoencoder with AdaGrad and MCC. The results are verified by the Friedman test, a post hoc test using the Wilcoxson signed-rank sum test with the Holm correction, and parameter sensitivity analysis.**

***Keywords—* refrigerated showcase, fault detection, hyperparameter tuning, coin betting, outlier treatment, Maximum Correntropy Criterion**

I. Introduction

In convenience stores and supermarkets, refrigerated showcases are utilized to keep food and drinks fresh and display them. Circulating cooled air keeps the refrigerated showcases cool. In the refrigerated showcases, faults such as refrigerant leakage and frost formation may occur. The faults may cause the refrigerated showcases to be unable to be cool and deterioration of food quality. The food whose quality has deteriorated must be discarded. Consequently, by the faults, sales opportunities may be lost. Therefore, for supporting customer service, detection of the refrigerated showcase faults with high accuracy is necessary.

Fig. 1 shows an example of a supposed showcase fault detection system. The system has three processes. In the first process, showcase data are gathered in a data center. In the second process, fault detection models are constructed offline in the data center using the gathered showcase data. In the third process, showcase conditions are estimated online using the fault detection models in the data center. In this estimation, online showcase data are utilized. If this system is put to practical use, faults occurring in the refrigerated showcases around the world can be detected online.

Satisfaction of five requirements except detecting the refrigerated showcase faults with high accuracy is necessary for detecting the refrigerated showcase faults practically. The requirement one is to be able to construct a fault detection model using only refrigerated showcase data without any specialist knowledge. In the world, huge number of the refrigerated showcases are utilized in various places such as supermarkets and convenience stores. The refrigerated showcases have different characteristics in response to places where they are utilized. Therefore, adjustment of the fault detection method for each refrigerated showcase by specialists is not practical. The requirement two is to be able to construct a fault detection model treating nonlinear correlation data because nonlinear correlation data are included in refrigerated showcase data [1]. The requirement three is to be able to construct a fault detection model even when the refrigerated showcase data include outliers. The outliers are data storing values differing from actual values. The outliers may be included in the refrigerated showcase data by various reasons such as radio frequency interference and incorrect sensor setting. The outlies affect fault detection accuracy. Therefore, for detection of the refrigerated showcase faults with high accuracy, elimination of the outliers in advance is required. However, for elimination of the outliers, huge engineering costs are required. The requirement four is to be able to a construct fault detection model using only normal refrigerated showcase data. The refrigerated showcase faults rarely occur. Therefore, it is difficult to obtain enough fault data of the refrigerated showcases. It is also difficult to utilize fault data of the refrigerated showcases for learning. The requirement five is to be able to tune hyperparameters easily. In machine learning, the hyperparameters affect fault detection accuracy strongly. Therefore, they are required to tune appropriately in advance and the tuning may require huge engineering costs.

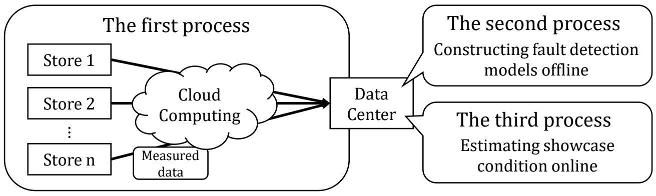


Fig. 1 An example of a supposed showcase fault detection system.

Conventionally, fault detection methods of the refrigerated showcases and air conditioning equipment are categorized into physical model based methods [2,3], classical artificial intelligence based methods [4,5], and machine learning based methods [6,7]. The fault detection methods using the physical model and the classical artificial intelligence require specialist knowledge and adjustment of the fault detection method for the refrigerated showcases at each store. Therefore, the requirement one is not satisfied by these methods. The conventional machine learning based fault detection methods require normally measured data not including the outliers. Therefore, the requirement three is not satisfied by these methods. Consequently, the conventional fault detection methods proposed so far cannot satisfy all of the five requirements.

For the refrigerated showcase fault detection, the authors have proposed autoencoder based methods [8,9], artificial neural network (ANN) with maximum correntorpy criterion (MCC) based methods [10-13], an ANN with the MCC and the adaptive kernel size tuning based method [14], and an autoencoder with the MCC and the adaptive kernel size tuning based method [15] and verified effectiveness of the methods. The autoencoder based methods satisfy the requirement one, two, and four. The ANN with the MCC based methods satisfy the requirement one, two, and three. The ANN with the MCC and the adaptive kernel size tuning based method satisfies the requirement one, two, three, and a part of the requirement five. The autoencoder with the adaptive kernel size tuning and the MCC based method satisfies the requirement one, two, three, four, and a part of the requirement five. However, the method has hyperparameters related to a learning rate. Therefore, the method does not satisfy the requirement five and requires engineering costs of tuning the hyperparameters related to a learning rate. On the other hand, coin betting was proposed in 2017 to reduce the engineering costs to tune the hyperparameters related to a learning rate [16]. By applying the method, reducing the engineering costs of tuning the hyperparameters related to a learning rate can be expected.

As a fault detection method for the refrigerated showcases including the outliers in measured data, this paper proposes refrigerated showcase fault detection by an autoencoder with coin betting, the MCC, and the adaptive kernel size tuning. The method satisfies the requirement one, two, three, and four. Moreover, it does not have hyperparameters related to a learning rate which are required to tune conventionally. Practicability of the proposed method is verified by comparison with an autoencoder with stochastic gradient descent (SGD) and least square error (LSE), an autoencoder with stochastic gradient descent with momentum (SGDM) and the MCC, an autoencoder with adaptive moment estimation (Adam) and the MCC, and an autoencoder with adaptive gradient algorithm (AdaGrad) and the MCC with refrigerated showcase actual data. The results are verified by the Friedman test, a post hoc test using the Wilcoxson signed-rank sum test with the Holm correction, and parameter sensitivity analysis.

II. Coin Betting for Learning an Artificial Neural Network

Conventionally, various methods such as SGDM, Adam, and AdaGrad have been proposed for ANN learning. However, the methods essentially have hyperparameters related to a learning rate. Tuning the hyperparameters appropriately is required and engineering costs of the tuning are also required. On the other hand, F. Orabona, et. al. proposed coin betting to solve the challenge in 2017 [16]. The coin betting does not have the hyperparameters related to the learning rate. In this paper, for reducing the engineering costs to tune the hyperparameters, the coin betting is utilized for updating autoencoder parameters. As an example, autoencoder parameter update formulas in the coin betting between input and hidden layers are expressed with the following equations.

$$g_{ij}(t) = -\frac{\partial LF_d}{W_{ij}(t)} \quad (i = 1, \dots, N_i, j = 1, \dots, N_h) \quad (1)$$

$$L_{ij}(t) = \max\left(L_{ij}(t-1), \left|g_{ij}(t)\right|\right) \quad (i = 1, \dots, N_i, j = 1, \dots, N_h) \quad (2)$$

$$G_{ij}(t) = G_{ij}(t-1) + \left|g_{ij}(t)\right| \quad (i = 1, \dots, N_i, j = 1, \dots, N_h) \quad (3)$$

$$Reward_{ij}(t) = \max\left(Reward_{ij}(t-1) + \left(W_{ij}(t-1) - W_{ij}(1)\right) g_{ij}(t), 0\right) \quad (i = 1, \dots, N_i, j = 1, \dots, N_h) \quad (4)$$

$$\theta_{ij}(t) = \theta_{ij}(t-1) + g_{ij}(t) \quad (i = 1, \dots, N_i, j = 1, \dots, N_h) \quad (5)$$

$$W_{ij}(t) = W_{ij}(1) + \Delta W_{ij}(t) \quad (i = 1, \dots, N_i, j = 1, \dots, N_h) \quad (6)$$

$$\Delta W_{ij}(t) = \frac{\theta_{i,j}(t)}{L_{ij}(t) \max\left(G_{ij}(t) + L_{ij}(t), \alpha L_{ij}(t)\right)} \left(L_{ij}(t) + Reward_{ij}(t)\right) \quad (i = 1, \dots, N_i, j = 1, \dots, N_h) (7)$$

where $g_{ij}(t)$ is a negative partial gradient of LF_d with respect to W_{ij} at the tth iteration, LF_d is a loss function value of the dth data, $W_{ij}(t)$ is a autoencoder parameter from the ith input unit to the jth hidden unit at the tth iteration, N_i is the number of input units, N_h is the number of hidden units, $L_{ij}(t)$ is the maximum absolute value of the partial gradient of LF_d with respect to W_{ij} up to the tth iteration, $G_{ij}(t)$ is the sum of absolute values of the partial gradients of LF_d with respect to W_{ij} up to the tth iteration, $Reward_{ij}(t)$ is corresponding to gambler's wealth at the tth iteration [16], $\theta_{ij}(t)$ is the sum of the gradients with respect to W_{ij} up to the tth iteration, $\Delta W_{ij}(t)$ is an update value of $W_{ij}(t)$, α is a constant value in order to limit autoencoder parameters at early iterations.

Autoencoder parameter update formulas in the coin betting between hidden and output layers are expressed using almost the same equations. The coin betting is a method based on the optimal betting strategy of gamblers betting on coin flip repeatedly. The betting strategy changes a rate of betting money in possession for each coin flip. The rates are increased as long as each coin flip result is the same side of the coin. On the other hand, the rates are decreased in case of a coin flip result differing from the last result. Using the strategy, the betting money in each coin flip is tuned automatically. In case of using the coin betting for updating autoencoder parameters, results of the coin flip are linked to gradients of loss function. In fact, in case of updating the autoencoder parameters based on the coin betting, signs of the gradients are focused on each iteration. Update rates of the autoencoder parameters are increased as long as signs of the gradients are the same as the last sign of the gradients. On the other hand, update rates of the autoencoder parameters are decreased in case of signs of the gradients differing from the last sign of the gradients. Autoencoder parameters are updated based on the strategy. Therefore, in the coin betting, ΔW in (6) can be tuned automatically during learning without using hyperparameters related to a learning rate.

Fig. 2 shows an example of minimizing a function $y = |x-10|$ using the coin betting. x starts from zero and update rates of x are increased until the sign of the gradient differs from the last sign of the gradient. In the figure, the update rate of x at the third update is increased because the signs of the first and the second gradients are the same. Similarly, the update rates of x at the fourth and fifth update are increased. On the other hand, the update rate of x at the sixth update is decreased because the signs of the fourth and fifth gradients are different. In this way, the update rates can be tuned automatically during learning. Learning algorithm by an autoencoder with coin betting is shown below. In the algorithm, D is the number of data and T_{max} is the maximum iteration number.

Step.1 Create initial autoencoder parameters with uniform random numbers. Learning data number $d = 1$. Iteration number $t = 1$.
Step.2 Calculate a loss function value for the dth learning data.
Step.3 Update autoencoder parameters using (1) to (7).
Step.4 If $d = D$, go to Step.5. Otherwise, $d = d + 1$ and return to Step.2.
Step. 5 If $t = T_{max}$, go to Step.6. Otherwise, $t = t + 1$, $d = 1$ and return to Step.2.
Step. 6 Output an autoencoder model.

III. Fualt Detection by an Autoencoder with Coin Betting and Maximum Correntoropy Criterion

A. Maximum Correntorpyp Criterion

For conventional autoencoders, least square error (LSE) has been usually utilized as a loss function for learning. The LSE in autoencoders is expressed with the following equation:

$$\min LF_d = \frac{1}{U}\sum_{u=1}^{U}(i_{du} - o_{du})^2 \quad (d = 1, \dots, D) \qquad (8)$$

where U is the number of units in input and output layers, i_{du} is an input value of the uth input unit in the dth data, o_{du} is an output value of the uth output unit in the dth data.

Fig. 3 shows an example of the loss function by the LSE. For learning, the LSE utilizes a whole error range. The larger error values become, the larger loss function values become using the LSE. The LSE minimize the sum of square errors between input and output values of the autoencoder. Therefore, the autoencoder using the LSE focuses to make large errors smaller. In actual store data, the outliers may be included in the refrigerated showcase data. In case of including the outliers in learning data, errors for the outliers become large. Therefore, the large errors are focused and reduced by learning in the autoencoder using the LSE. This may cause to decrease fault detection accuracy. In order to tackle the challenge, W. Liu, et al. proposed the MCC in 2006 [17]. The MCC in autoencoders is expressed with the following equation:

$$\max LF_d = \frac{1}{U}\sum_{u=1}^{U}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(i_{du} - o_{du})^2}{2\sigma^2}\right) \quad (d = 1, \dots, \mathrm{D}) \qquad (9)$$

where σ is a kernel size.

The MCC utilizes a limited error range for learning using a Gaussian kernel function. Fig. 4 shows an example of the loss function by the MCC. When error values become large, loss function values become close to zero by the MCC. Therefore, the autoencoder using the MCC can learn without affection by the outliers. As shown in Fig. 4, only data with errors within plus or minus three are utilized for learning and data with errors greater than plus or minus three are ignored and not utilized for learning. As described before, in case of including the outliers in learning data, errors of the outliers become large. Therefore, the large errors are ignored for learning in the autoencoder using the MCC.

Fig. 5 shows decision boundaries at learning and test stages by autoencoders using the LSE and the MCC when the outliers are included in learning data. The autoencoder using the LSE focuses the outliers which have large errors and learns to reduce the errors. Therefore, the autoencoder using the LSE learn to fit not only normal measured data but also the outliers which have large errors. This may cause an incorrect decision boundary for fault detection of the refrigerated showcases. Consequently, in case of including the outliers in learning data, the refrigerated showcase faults cannot be detected with high accuracy for test data by the autoencoder using the LSE (Fig.5(a)). On the other hand, the autoencoder using the MCC utilized a limited error range. Therefore, even in case of including the outliers in learning data, the autoencoders can learn without affection by the outliers which have large errors. This may cause a correct decision boundary for fault detection of the refrigerated showcases. Consequently, even in case of including the outliers in learning data, the refrigerated showcase faults can be detected with high accuracy for test data by the autoencoders using the MCC (Fig.5(b)).

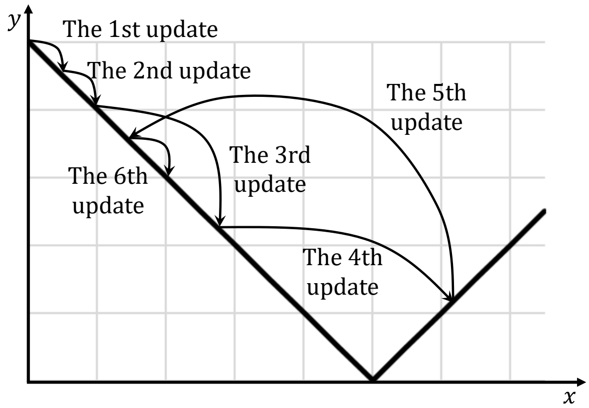


Fig. 2 An example of minimizing a function $y = |x-10|$ using the coin betting.

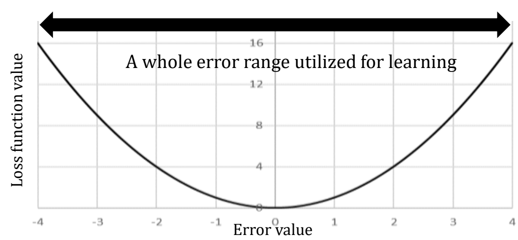


Fig.3 An example of the loss function by the LSE.

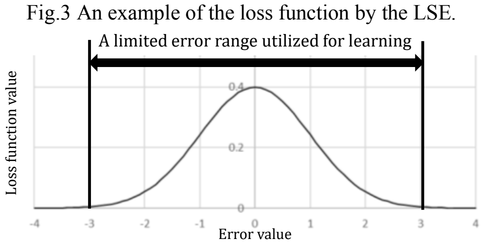
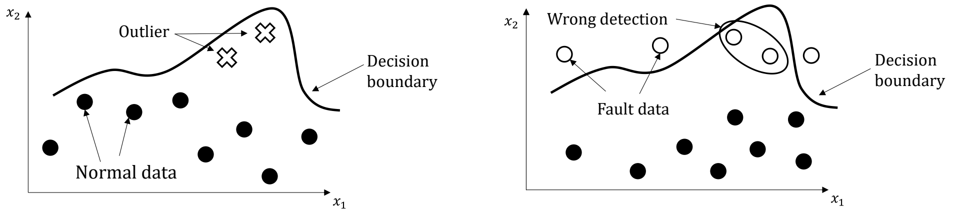
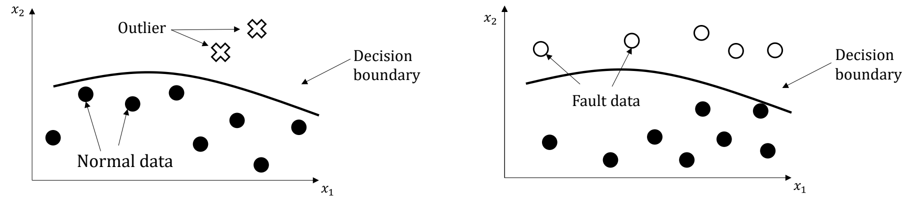


Fig. 4 An example of the loss function by the MCC.



(a) Decision boundaries at a learning stage (left) and a test stage (right) by an autoencoder using the LSE in case of including the outliers in learning data.



(b) Decision boundaries at a learning stage (left) and a test stage (right) by an autoencoder using the MCC in case of including the outliers in learning data.

Fig. 5 Decision boundaries at a learning stage and a test stage by autoencoders using the LSE and the MCC in case of including the outliers in learning data.

B. An Algorithn of the Proposed Method

For the refrigerated showcases, an algorithm for generating a fault detection model by an autoencoder with coin betting, the MCC, and the adaptive kernel size tuning is shown below. In the algorithm, the adaptive kernel size tuning method can be found in [15].

Step.1 Create initial autoencoder parameters with uniform random numbers. Set an initial σ. $d = 1$. $t = 1$.

Step.2 Calculate a loss function value for the dth learning data using (9).

Step.3 Update autoencoder parameters using (1) to (7).

Step.4 If $d = D$, go to Step.5. Otherwise, $d = d + 1$ and return to Step.2.

Step.5 Update the kernel size in (9) using the following equations [15]:

$$\sigma(t) = \frac{E_{order}(DON(t))}{2\sqrt{2}} \tag{10}$$

$$DON(t) = roundup(R(t) \times D) \tag{11}$$

$$R(t) = \frac{R_{max}}{T_{max}} \times t \tag{12}$$

where $\sigma(t)$ is the kernel size at the tth iteration, $E_{order}(a)$ is the ath absolute error of learning data between input and output values in a descending order vector, $DON(t)$ is a descending order number of the $E_{order}(a)$ at the tth iteration, $roundup(r)$ is a function to round up a real number r to an integer, $R(t)$ is a rate of a top percentage of errors to be utilized to tune the kernel size at the tth iteration, R_{max} is the final rate of the top percentage of errors to be utilized to tune the kernel size.

Step.6 If $t = T_{max}$, go to Step.7. Otherwise, $t = t + 1$, $d = 1$ and return to Step.2.

Step.7 Output an autoencoder model and check test data whether a fault occurs at the refrigerated showcase or not using the model.

IV. SIMULATION

A. Simulation Conditions

The proposed autoencoder with coin betting and MCC (the proposed method), an autoencoder with SGD and LSE (the comparative method 1), an autoencoder with SGDM and MCC (the comparative method 2) [15], an autoencoder with Adam and MCC (the comparative method 3), and an autoencoder with AdaGrad and MCC (the comparative method 4) are applied for the refrigerated showcase fault detection with actual refrigerated showcase data. Fault detection results of the above four methods are compared. Simulation conditions are shown below.

- Initial autoencoder parameters are changed 30 times.
- 10-fold cross validation is performed.
- The outliers are set to include 10% in learning data randomly (outlier rate: 10%).
- Learning data consist of 70% of data under normal conditions.
- Test data consist of all data under fault conditions and 30% of data under normal conditions.
- As autoencoder activation functions, sigmoid function are utilized.

Hyperparameters of the proposed and comparative methods are shown below.

- Common hyperparameters of all methods:
  - The number of hidden layers : 1, The number of hidden layer units : 2, T_{max} : 1000, R_{max} : 5 (for 0% outlier rate) and 15 (for 10% outlier rate)
- A hyperparameter of the proposed method:
  - α : 100 (a default value)
- Hyperparameters of the comparative method 1:
  - η : 0.01, where η is a learning rate
- Hyperparameters of the comparative method 2:
  - η : 0.01, c : 0.01

The following equations are utilized for updating autoencoder parameters by SGDM [15].

$$W_{ij}(t) = W_{ij}(t-1) + \Delta W_{ij}(t) \tag{13}$$

$$\Delta W_{ij}(t) = \eta \frac{\partial LF_d}{\partial W_{ij}(t-1)} + c\Delta W_{ij}(t-1) \tag{14}$$

where $\Delta W_{ij}(t)$ is an update value of W_{ij} at the tth iteration, c is a momentum coefficient.

- Hyperparameters of the comparative method 3:
  - γ : 10^{-4}, β_1 : 0.9, β_2 : 0.999, ε : 10^{-8}

The following equations are utilized for updating autoencoder parameters by Adam [18].

$$W_{ij}(t) = W_{ij}(t) - \gamma(t)\frac{m(t)}{\left(\sqrt{v(t)} + \varepsilon\right)} \quad (15)$$

$$\gamma(t) = \gamma\frac{\sqrt{1-{\beta_2}^t}}{1-{\beta_1}^t} \quad (16)$$

$$m(t) = \beta_1 m(t) + (1-\beta_1)\frac{\partial LF_d}{\partial W_{ij}(t)} \quad (17)$$

$$v(t) = \beta_2 v(t) + (1-\beta_2)\left(\frac{\partial LF_d}{\partial W_{ij}(t-1)}\right)^2 \quad (18)$$

where $\gamma(t)$ is a step size at the tth iteration, $m(t)$ is a biased first moment estimate at the tth iteration, $v(t)$ is a biased second raw moment estimate at the tth iteration, β_1 is an exponential decay rate for m, β_2 is an exponential decay rate for v, ε is a small positive number, γ is a step size coefficient.

- Hyperparameters of the comparative method 4:
  - η : 0.07, ε : 10^{-8}

The following equations are utilized for updating autoencoder parameters by AdaGrad [19].

$$W_{ij}(t) = W_{ij}(t-1) + \frac{\eta}{\sqrt{h_{ij}+\varepsilon}}\frac{\partial LF_d}{\partial W_{ij}(t-1)} \quad (19)$$

$$h_{ij} = h_{ij} + \left(\frac{\partial LF_d}{\partial W_{ij}(t-1)}\right)^2 \quad (20)$$

where h_{ij} is sum of squared gradients with respect to W_{ij}.

As evaluation indicators for the refrigerated showcase fault detection, “accuracy” and “recall” are utilized. The accuracy is a rate of correct detection results for all test data (Acc.). The recall is a rate of correct detection results for fault test data. In actual situations, store personnel find the refrigerated showcase faults lately if a specific refrigerated showcase is in fault conditions but the fault is undetected. This may cause deterioration of food quality and sales opportunities may be lost. Since it is important to detect the refrigerated showcase faults correctly as fault, the recall is considered more important than the accuracy.

B. Simulation Results

Table I shows average accuracy and recall values by the proposed method and comparative method 1, 2, 3 and 4, and p-values by the Friedman test using the comparative method 2, 3, 4 and the proposed method. It is confirmed that the accuracy value by the comparative method 1 is extremely worse than those by other methods when learning data include the outliers (bold number). Therefore, since the comparative method 1 is affected by the outliers largely, the method using the LSE is not suitable for the refrigerated showcase fault detection that may include the outliers. It is confirmed that the other methods except the comparative method 1 can detect the refrigerated showcase faults with almost the same accuracy even when the outliers are included in learning data. Therefore, the methods using the MCC are suitable for the refrigerated showcase fault detection that may include the outliers. In both cases of including and not including the outliers in learning data, the recall values of the comparative method 2 are the highest and those of the proposed method are the second highest. The Friedman test is applied for revealing a significant difference among four suitable methods (the comp. 2, 3, 4, and the proposed methods). As a result of the Friedman test, according to p-values in both cases, it is confirmed that there are significant differences at a significance level of 5% among the three suitable comparative methods using the MCC and the proposed method for the recall values.

In the Friedman test, significant differences between the specific two methods are not clear when three or more methods are compared. Therefore, revealing a significant difference between the specific two methods, a post hoc test using the Wilcoxson signed-rank sum test with the Holm correction is applied to all combinations of the proposed method and the comparative method 2, 3, and 4. Table II and III show p-values by the post hoc test for the recall values with and without the outliers in learning data. As a result of the post hoc test, when the outliers are not included in learning data, it is confirmed that there are significant differences at a significance level of 5% between any two methods. Moreover, when the outliers are included in learning data, it is confirmed that there are significant differences between the proposed method and all comparative methods. Consequently, from the recall point of view, it is clear that recall of the comparative method 2 is the highest, and recall of the proposed method is the second highest.

Although there are significance differences at a significant level of 5% between the proposed method and the comparative method 2, the differences of the recall values with and without the outliers are small (italic numbers). Actually, the differences of the recall values between the comparative method 2 and the proposed method are 4.016E-05 without the outliers, and 4.017E-05 with the outliers. Moreover, the comparative method 2 has two hyperparameters related to a learning rate. Fig.6 shows

TABLE I. AVERAGE ACCURACY AND RECALL VALUES BY THE THE PROPOSED METHOD AND COMPARATIVE METHODS 1, 2, 3, AND 4, AND P-VALUES BY THE FRIEDMAN TEST USING THE COMPARATIVE METHOD 2, 3, 4, AND THE PROPOSED METHOD.

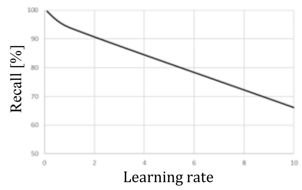
Outlier Rate		0[%]		10[%]	
Evaluation [%]		Acc.	Recall	Acc.	Recall
Method	Comp.1	85.02	98.35	**62.75**	98.78
	Comp.2	84.44	*99.56*	86.34	*99.85*
	Comp. 3	86.68	99.47	85.36	99.82
	Comp. 4	86.90	99.41	84.71	99.81
	The Proposed Method	85.36	*99.55*	82.24	*99.84*
P-value		9.7E-19		9.1E-14	

TABLE II. P-VALUES BY THE POST HOC TEST FOR THE RECALL WITHOUT THE OUTLIERS IN LEARNING DATA (OUTLIER RATIO:0%).

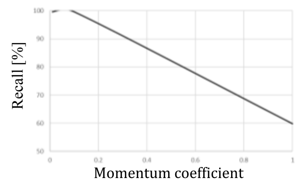
	The Proposed Method	Comp. 2	Comp. 3
Comp. 2	0.02		
Comp. 3	4.1E-07	5.1E-08	
Comp. 4	0	9.4E-15	3.0E-03

TABLE III. P-VALUES BY THE POST HOC TEST FOR THE RECALL WITH THE OUTLIERS IN LEARNING DATA (OUTLIER RATIO:10%).

	The Proposed Method	Comp. 2	Comp. 3
Comp. 2	8.0E-07		
Comp. 3	8.0E-10	7.0E-03	
Comp. 4	1.1E-10	1.5E-04	0.20



(a) Parameter sensitivity analysis of η.



(b) Parameter sensitivity analysis of c.

Fig. 6 Parameter sensitivity analysis of the comparative method 2.

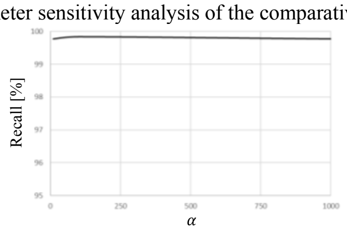


Fig. 7 Parameter sensitivity analysis of the proposed method.

parameter sensitively analysis of the comparative method 2. It is confirmed that the comparative method 2 cannot detect the refrigerated showcase faults correctly as faults when the two hyperparameters are getting larger. Therefore, it is required to tune the hyperparameters appropriately for each showcase to detect the refrigerated showcase faults with high recall values using the comparative method 2. On the other hand, the proposed method has a hyperparameter α which limits autoencoder parameters at early iterations. Fig.7 shows parameter sensitively analysis of the proposed method. It is confirmed that the proposed method can detect the refrigerated showcase faults correctly as faults with almost the same correctness even if the hyperparameter α is getting larger. Therefore, it is not required to tune the hyperparameter α for each showcase for detection of the refrigerated showcase faults with high recall values using the proposed method. Since huge number of refrigerated showcases are utilized in the world, the proposed method is suitable for the refrigerated showcase fault detection according to high recall values and no need for hyperparameter tuning. Consequently, the proposed method is a more practical refrigerated showcase fault detection method than the comparative methods from the requirement five, namely, tuning hyperparameters easily, point of view.

V. CONCLUSIONS

This paper proposes refrigerated showcase fault detection by an autoencoder with coin betting and MCC. Practicability of the proposed method is verified by comparison with an autoencoder with SGD and the LSE, an autoencoder with SGDM and the MCC, an autoencoder with Adam and the MCC, and an autoencoder with AdaGrad and the MCC. The results are verified by the Friedman test, a post hoc test using the Wilcoxson signed-rank sum test with the Holm correction, and parameter sensitivity analysis.

As future works, for improvement in accuracy and to achieve more efficient hyperparameter tuning, various learning methods will be investigated.

REFERENCES

[1] K. Arai, Y. Fukuyama, K. Murakami, and T. Matsui,"Unsupervised fault detection for refrigeration showcase systems with kernel principal component analysis based multivariate statistical process control using feature selection with maximal information coefficient," *IEEJ Journal of Industry Applications*, vol. 10, no. 5, pp. 598-605, 2021.

[2] K. Mihara, Y. Aono, T. Ohkawa, N. Komoda, and F. Miyasaka, "Diagnosis of air conditioning system by stochastic quliative reasoning," *IEEE Symposium on Emerging Technologies and Factory Automation*, pp. 86-91, 1994.

[3] K. Mihara, Y. Aono, T. Ohkawa, N. Komoda, and F. Miyasaka, "Stochastic qualitative reasoning and its application to diagnosis of air conditioning system," *Proc. of IRCON'94-20th Annual Conference of IEEE Industrial Electronics*, pp. 1401-1406, 1997.

[4] S. Youk, S. Yoo, Y. Kim, and G. Park,"Fault detection system using directional classified rule-based in AHU," *Proc. of Future Generation Communication and Networking (FGCN)* , 2007.

[5] D. Han, and Y. Chang,"Automated fault diagnosis method for a variable air volume air handling unit," *Proc. of International Conference on Control, Automation and Systems - Society of Instrument and Control Engineers* (ICCAS-SICE) , pp. 2038-2041, 2009.

[6] K. Choi, S. M. Namburu, M. Azam, J. Luo, K. R. Pattipati, and A. Patterson-Hine,"Fault diagnosis in HVAC chillers," *IEEE Instrumentation and Measurement Magazine*, Vol. 8, Issue3, pp. 24-32, 2005.

[7] V. Martinez-Viol, E. M. Urbano, K. Kampouropoulos, M. Delgado-Prieto, and L. Romeral,"Support vector machine based novelty detection and FDD framework applied to building AHU systems," *Proc. of 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1759-1754, 2020.

[8] A. Santana, Y. Fukuyama, K. Murakami, and T. Matsui,"Machine learning application for refrigeration showcase fault discrimination", Proc. of IEEE TENCON, pp.10-13 2016.

[9] A. Santana, K. Murakami, T. Iizaka, T. Matsui, and Y. Fukuyama,"One class data mining approaches for leakage fault detection in refrigeration showcases", Proc. of Annual Conference of SICE, pp.1514-1519 2018.

[10] N. Otaka, Y. Fukuyama, et al.,"Refrigerated showcase fault analysis by a Correntropy based artificial neural network using particle swarm optimization," *Annual Conference of SICE*, pp. 1508-1513, 2018.

[11] N. Otaka, Y. Fukuyama, et al.,"Refrigerated showcase fault analysis by a Correntropy based artificial neural network using modified brain swarm optimization," *Proc. of IEEE SSCI*, 2019.

[12] N. Otaka, Y. Fukuyama, et al.,"Refrigerated showcase fault analysis by a Correntropy based artificial neural network using fast brain swarm optimization," *Proc. of International Conference on Swarm Intelligence*, 2019.

[13] N. Otaka, Y. Fukuyama, et al.,"Refrigerated showcase fault detection by a pasting based artificial neural networks using parallel multi-population modified brain storm optimization and Correntropy," *Proc. of IEEE CEC*, 2020.

[14] M. Igarashi, Y. Fukuyama, et al.,"Refrigerated showcase fault detection by an artificial neural network using Correntropy with improved adaptive kernel size tuning," *Proc. of integration of power conversion, electric machine, and control*, 2022.

[15] M. Igarashi, Y. Fukuyama, et al.,"Refrigerated Showcase Fault Detection by an Autoencoder with the Adaptive Kernel Size Tuning using Maximum Correntropy Criterion," *Proc. of IEEE SSCI*, 2022.

[16] F. Orabona, et. al, "Training Deep Networks without Learning Rates Through Coin Betting," Proc. of 31st Conference on Neural Information Processing Systems, 2017.

[17] W. Liu, P. P. Pokharel, and J. C. Principe, "Correntropy: a localized similarity measure", Proc. of IJCNN, pp.4919-4924 2006.

[18] D. P. Kingma and J. L. Ba: "Adam: A Method for Stochastic optimization", 3rd International Conferrence for Learning Representations (2014).

[19] J. Duchi, E. Hazen, and Y.Singer: "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization", Journal of Machine Learning Research 12, pp. 2121-2159 (2011).