Diagnosing Faults in BWTS based on Machine Learning

Jae Kyun Kim[1], Jae-Hoon Kim[2], and Seong Dae Lee[3†]

[1]Department of Computer Engineering, Korea Maritime & Ocean University, Korea
[2,3]Department of Control and Automation Engineering, Korea Maritime & Ocean University, Korea

Summary
Due to environmental regulations on navigation ships of the IMO (International Maritime Organization), demand for eco-ships and associated equipment is soaring. Eco-ship equipment includes BWTS (ballast water treatment system) and Sox scrubbers. The BWTS is a device for purifying ballast water, which is a major cause of marine pollution. This paper proposes a fault diagnosis system based on machine learning. The proposed system is a classification model that judges the status of BWTS faults through various sensor data sent to the BWTS. The operation data of the BWTS are times series data, and normal state or diverse faults are attached to the data as class. The operation data provided for an experiment in this paper were divided into learning data and evaluation data, and were analyzed through a SVM (Support Vector Machine). The accuracy of each fault cause on the evaluation data was 86.93% on average, and the false alarm rate was 5.9%, signifying room for improvement. Improvements will be made through sufficient collection of learning data, fault data augmentation, and imbalance learning.

Key words:
Eco-Ship, BWTS, Fault-diagnosis, Big-data, Machine Learning

1. Introduction

As international interest in environmental pollution and climate change issues increases, greenhouse gases emission reduction policy and various environmental regulations are being consolidated. The environmental regulations are used as an invisible protective trade barrier to entering the countries concerned, and all industrial fields, such as electrical & electronics, machinery, automobiles, and chemical products including shipbuilding, are included in the scope of such regulations. In the shipbuilding and maritime affairs field, demand for eco-ships is sharply increasing due to environmental regulations of the IMO. Eco-ships have better fuel efficiency and have innovatively reduced marine pollutant emissions/discharges compared to existing ships. Eco-ships refer to ships that can meet EEDI (Energy Efficiency Design Index) of new ships discussed by IMO [1]. In addition, eco-ships refer to ships equipped with technologies suitable for environmental regulations, such as CO2, NOx, Sox, and ballast water discharge regulations [1].

The techniques used for fault diagnosis of machines or equipment including eco-ship equipment can be divided into physical model-based diagnosis and data-driven diagnosis. In addition, the data-driven diagnosis can be classified into signal-based diagnosis and machine learning-based diagnosis [2]. This paper proposes a fault diagnosis and prediction system of BWTS based on machine learning to prevent rejection of ship’s port entry caused by the fault of the BWTS in terms of eco-ship equipment. Ballast water is sea water supplied to maintain ship’s balance depending on cargo loading state, and it becomes a major cause disturbing the marine ecosystem through marine organisms’ migration. Major technologies to treat the ballast water include electrolytic method, ozone spray method, UV disinfection method, and chemical treatment method [3].

In this paper, the machine learning-based fault diagnosis algorithms are proposed using the UV disinfection method of BWTS operation data. To detect or prevent faults of UV lamps (core of UV disinfection method of BWTS), machine learning is performed targeting five-sensor data, including pressure and temperature sensing the operation status of the UV lamp. The proposed system consists of four steps: The first step is pre-processing. Here the BWTS operation data received from a navigation ship are converted into real number values of 0 and 1. In a general machine system, if pressure data scale is not normalized, performance can decrease significantly. This is especially severe for SVMs [4]. The second step is feature generation, whereby time series data are generated as window-based (n-gram). For time series data, if RNN (Recurrent Neural Network) [5] is used, all previous information can be used at the current point in time. This paper was performed to predict information using previous n number of information, rather than the neural network requiring lots of resources, in order to apply to industrial sites. The third step is learning, namely the machine learning model receiving BWTS operation data as learning data. As mentioned above, SVM [6] is used as a machine learning model. The fourth step is an experimental step, in other words, an application of the learned machine learning model to evaluation data. Through this step, system performance is evaluated.

This paper consists of the following: Chapter 2 introduces existing studies on fault diagnosis and Chapter 3 describes the BWTS fault diagnosis system using machine learning. Chapter 4 describes the implementation and experiment of the algorithms, and Chapter 5 discusses conclusions and future paper tasks.

Manuscript received August 5, 2020
Manuscript revised August 20, 2020

2. Related Works

General fault diagnosis methods are introduced in this chapter. A fault diagnosis method, which is machine learning-based fault diagnosis, and anomaly detection are described in brief.

2.1 Fault diagnosis

Techniques used for fault diagnosis of machines or equipment can be divided into physical model-based diagnosis and data-driven diagnosis [2, 7]. The physical model-based fault diagnosis technique is a method judging faults by analyzing differences between the the values measured from various sensors that inspect machines or equipment and the values drawn by the mathematic physical model. However, this technique has a drawback in that generation of a physical model containing numerous input values, output values, and state variables is difficult [8-10]. The data-driven fault diagnosis technique is a method of diagnosis that uses past data containing the status information of a machine or equipment, and it can be divided into signal-based method and machine learning-based method. The signal-based diagnosis mainly uses a signal processing technique for vibration data processing of a rotary machine, and is a fault-detection technique comparing pre-defined threshold values and analyzed signals. The main techniques include time domain, frequency domain, and time-frequency domain. In time domain analysis, a signal's statistical features in the time domain are extracted differently, and such features are analyzed as one-dimensional time domain or the time series signals are interpreted as images and analyzed as two-dimensional time domain. In frequency domain analysis, spectrum analysis using FFT (Fast Fourier Transform) is the most widely used. However, methods based on FFT have a drawback in that transient features cannot be efficiently inspected. To supplement the demerit, the time-frequency domain analysis is used by mixing time and frequency information. It is a method that analyzes transient features by monitoring frequency spectrum in the time domain [11-12].

The fault diagnosis technique based on machine learning has recently gained attention. Machine learning-based diagnosis carries out a feature selection process in which dimensions are reduced for machine learning performance after feature extraction from the data measured through the sensors of machines or equipment or signal processing technologies. Methods including principal component analysis are mainly used for machine learning's dimension reduction. And then learning is performed on the selected features using machine learning algorithms. Algorithms generally used for diagnosis techniques can be categorized into supervised learning and unsupervised learning. Supervised learning is divided into a classification model and a prediction model. The classification model is divided into kNN (k-Nearest Neighbor), SVM, and decision tree. For the prediction model, a regression analysis is typical. For the unsupervised model, a clustering model is a typical model. Clustering algorithms can be divided into partitioning methods and hierarchical methods [13]. For machine learning algorithms for fault diagnosis, SVM and an anomaly detection technique are generally used, and these methods are briefly described in the following chapters.

2.2 Support Vector Machine

SVM is an algorithm that identifies the optimum linear decision boundary that separates data linearly [8]. Decision boundary is a hyper-plane having maximum margin between training data. For example, let's assume learning data as $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$, where $\boldsymbol{x}_i \in R^N$ is the input variable at N dimension, and is a class label or a category having $y_i \in \{-1, +1\}$. The hyper-plane separating the learning data into –1 class (category) and +1 class (category) is called the decision boundary, and it can be indicated as the set of dot (·) x meeting Equation (1).

$$\boldsymbol{w} \cdot \boldsymbol{x} - \boldsymbol{b} = 0 \tag{1}$$

In Equation (1), · is an inner product operator, **w** is the normal vector of the hyper-plane. The support vector refers to a set of learning data existing closest to the given hyper-plane, and it can be divided into the case of $y_i = +1$ and the case of $y_i = -1$. Because no learning data should exist between the hyper-planes, Equation (2) should be established.

$$y_i(W \cdot X_i - b) \geq 1, \forall 1 \leq i \leq n \tag{2}$$

In the hyper-plane margin, margin refers to the distance between each support vector and hyper-plane, and is defined as $\frac{2}{\| \boldsymbol{w} \|}$. As stated above, SVM is a method of identifying the hyper-plane, maximizing the hyper-plane's margin. Namely, SVM's learning process is looking for $(\boldsymbol{w}^*, \boldsymbol{b})$ maximizing the margin as defined in Equation (3).

$$(\boldsymbol{w}^*, \boldsymbol{b}) = \mathrm{argmin}_{(\boldsymbol{w}, \boldsymbol{b})} \| \boldsymbol{w} \| \tag{3}$$

Thus far, descriptions of the case in which learning data can be divided linearly have been made. If any given problem cannot be divided linearly, the nonlinear division problem can be converted into a linear-division problem using the kernel function.

2.3 Anomaly Detection

Anomaly detection scans for data beyond normal data [14]. In other words, it is tasked with finding anomalous data (erroneous data), abnormal events, and defective data. The anomaly detection technique can be classified as a statistical technique and machine learning technique, and the anomaly detection technique based on machine learning has been mainly researched recently. The machine learning-based anomaly detection technique includes a classification-based error detection method [15], a NN (nearest neighbor)-based error detection method [16], and a clustering-based error detection method [17]. In this paper, SVM, which is a machine learning-based classification algorithm, is used for anomaly detection. In classification-based anomaly detection, as anomalous data are remarkably smaller than normal data, a problem of data imbalance occurs [18]. For data imbalance, studies on imbalance learning [19], learning data augmentation [20], and batch sampling [21] are being carried out.

3. Fault-Diagnosis on BWTS

This paper proposes a fault diagnosis system on the BWTS using a UV disinfection method. A UV lamp is a key for the UV disinfection method of BWTS. A system for detection of UV lamp faults has been developed in this paper. Machine learning that targets five sensors, including pressure- and temperature-sensing in the operation status of the UV lamp, is carried out. The proposed machine learning-based fault diagnosis system consists of four steps, as presented in Figure 1.

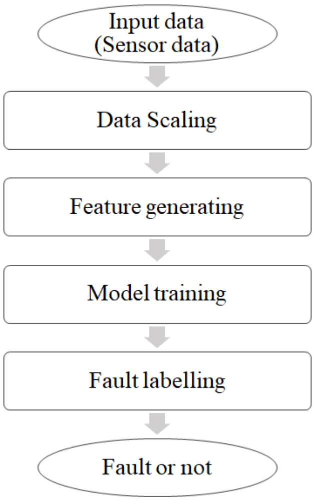

Fig. 1 Work flow of the proposed system

The first step is pre-processing, namely, converting BWTS operation data received from a navigation ship into real values of 0 and 1. The second step is extracting features necessary for machine learning. The third step is learning, whereby machine learning models are learned by receiving BWTS operation data as learning data. Fourth is the experimental step, whereby the learned machine learning models are applied to evaluation data. Each step is described in detail in the following chapters. Through all these steps, the system performance is evaluated.

3.1 Data scaling

Some pre-processing is necessary for input of machine learning without inputting the values measured by sensors. As most machine learning algorithms receive numbers in the pre-processing process, there are a conversion process into numbers and a measured data adjustment process to improve algorithm performance. In this paper, data scope is normalized as part of the measured data adjustment process. This process is called data scaling. Namely it is the step to convert BWTS operation data received from a navigation ship into real numbers like 0 and 1. As values specified in each sensor are different, data scaling is used to ease the difference. The data scaling method includes standard scaling and MinMax scaling [4]. This paper uses MinMax scaling, as shown in Equation (4).

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \qquad (4)$$

In Equation (4), x is the sensor-measured value, namely, raw data value, and $x_{\max}$ and $x_{\min}$ are the maximum and minimum values of the measured values of each sensor.

3.2 Feature generating

Concerning time series data, if RNN (recurrent neural network) [5] is used, all previous information can be used at the current point in time. This paper was composed to predict information using previous n information, rather than the neural network requiring lots of resources, in order to apply to industrial sites.

Input of machine learning algorithms generally consists of real number vector, $x \in R^N$. For time series data, (x_0, x_1..., x_t, ...) is given. Regarding time series data, as diagnosing faults with only currently measured data x_t is not a good method, mainly lots of previous data are used together. In this paper, the data belonging to windows, while moving the n-sized window, are reproduced with input of the machine learning algorithm. For example, if the window size is two, it is recomposed as [(x_0 , x_1), (x_1 , x_2), (x_2 , x_3), ...], and if the window size is three, it is recomposed as [(x_0 , x_1, x_2), (x_1 , x_2, x_3), (x_2 , x_3, x_4), ...]. Therefore, as the window size is larger, the size of the machine learning algorithm becomes larger. The window size was set through an experiment that is described in more detail in Chapter 4.

3.3 Model training

As a machine learning algorithm, SVM is used in this paper. SVM finds data beyond the margin by regression method in many cases, if SVM is used for anomaly detection [22]. This paper regards such a problem as a classification problem. As mentioned in Chapter 2, SVM classifies class or category using margin and decision boundary. SVM is suitable for binary classification. When SVM is applied to multi-class classification, the OvO (One versus One) and OvR (One versus the Rest) methods are used [23]. The OvO method solves the k (k-1)\/2binary classification problem by selecting two classes (categories) in case k classes (categories) exist. It is a method of producing the most-obtained class (category) as the final outcome. In this method, if the number of categories increases, the number of binary classification increases. As for the OvR method, if k classes exist, the binary problems of each class (category) and the rest of the classes (categories) are solved, and the most-obtained class (category) is produced. Although it is OK to solve only k binary classification problems in this method, categories having the same number occur, and so a method to solve this is needed. This paper uses the OvO method.

3.4 Fault Labelling

Category in this paper predicts causes of various faults, as well as the status of faults. Table 1 presents brief descriptions of class (category) according to a variety of causes of faults. Faults were classified into 10 fault types, causes of which are widely used on industrial sites at present.

Table 1: Label description for faults

Label	**Fault name**	**Description**
0	normal	Normal operation
1	Flow 1	Measuring of inlet pressure and DP, but not FLOW
2	Flow 2	Abnormal fluctuation of FLOW
3	Fin 1	Measuring of inlet pressure, but not FLOW
4	Fin 2	Measuring of FLOW, but not inlet pressure
5	Fin 3	Abnormal fluctuation of Inlet pressure
6	DP 1	Continuous increase of DP while backflushing
7	DP 2	Measuring of FLOW and Inlet pressure, but not DP
8	DOSE 1	Abnormal increase of UV DOSE
9	DOSE 2	Abnormal fluctuation of UV DOSE
10	TEMP	Abnormal temperature

% DP: Differential pressure

DOSE: UV dose, energy dosage of ultraviolet radiation

FLOW: the amount of inflow in ballast water

4. Experimental Results

In this chapter, the experimental environment is briefly described, and performance based on evaluation data is analyzed.

4.1 Experimental environment

1) Data gathering

Figure 2 shows GloEn-Patrol, a BWTS of Panasia Co., Ltd. This BWTS is an eco-BWTS that applies a filtering device using a filter and UV disinfection technology, and that effectively treats ballast water. An experiment was performed using the operation data of Panasia's BWTS.

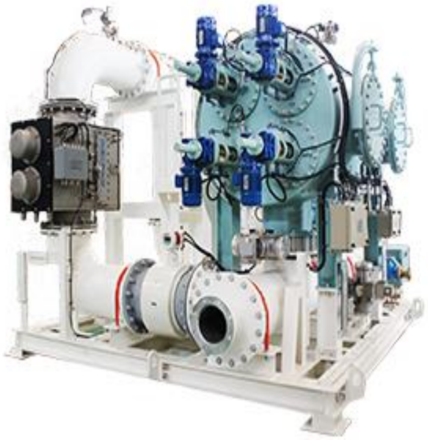

Figure 2. BWTS manufactured by Panasia in Busan

Table 2 is part of raw data gathered from the BWTS, and the data are gathered from five types of sensors – FLOW, FLOW, F_IN, F_DP, DOSE, and TEMP. Flow means the amount of inflow in ballast water, F_IN refers to filter inlet pressure, and F_DP refers to pressure difference. DOSE refers to UV dose, and TEMP refers to temperature. As seen in Table 2, the values measured by each sensor have various scopes according to sensor type, and so there is a need to adjust the data scope.

Table 2: Part of raw data collected from BWTS

Time	FLOW	F_IN	F_DP	DOSE	TEMP	Label
0	513.0	1.05	0.12	317.2	25.7	0
1	493.5	1.22	0.13	324.5	25.8	0
2	491.1	1.25	0.14	332.7	25.8	0
3	487.5	1.26	0.15	334.3	25.8	0
4	486.7	1.27	0.16	335.4	25.8	0
5	484.8	1.27	0.18	335.1	25.7	0
6	483.1	1.29	0.21	335.2	25.8	0
7	480.2	1.30	0.22	337.6	25.8	0
8	478.0	1.31	0.24	338.6	25.9	0
9	479.9	1.33	0.26	339.9	25.8	0

10	478.8	1.33	0.27	341.3	25.8	0
11	477.9	1.34	0.28	341.7	25.9	0
12	476.8	1.34	0.29	341.6	25.9	0
13	475.7	1.35	0.31	344.1	25.9	0
14	488.8	1.26	0.33	339.4	25.8	0
15	490.9	1.24	0.34	334.5	25.8	0
16	491.7	1.23	0.36	332.3	25.9	0
17	1.1	1.29	0.45	335.7	25.9	1
18	1.3	1.07	0.17	335.4	26.0	1
19	1.6	1.00	0.08	328.4	25.9	1
20	1.4	1.00	0.08	328.3	25.9	1

The data gathered in the form as shown in Table 2 are 47,435 in total, and the whole distribution is shown in Table 3. Of the 47,435 data, 46,722 were used as learning data, and the remaining 713 data were used for evaluation (The evaluation data were provide by Panasia Co., Ltd. separately from the learning data). The ratio contained in parentheses in Table 3 refers to the ratio of learning data to evaluation data. Overall, the ratio of learning data (98.5%) and evaluation data (1.5%) shows a slight difference from general machine learning cases. The evaluation data were composed to examine how accurately faults are analyzed, rather than normal data. Although not expressed in Table 3, the ratio of normal state and faults is 97.9% to 2.1%, and it was 29.5% vs. 70% on the evaluation data. Through this, it can be understood that evaluation of this paper concentrates on faults.

Table 3: Statistics of collected raw data

Label	Train	Test	Total
0	45,725 (99.5%)	210 (0.5%)	45,935
1	183 (71.2%)	74 (28.8%)	257
2	98 (57.0%)	74 (43.0%)	172
3	88 (72.7%)	33 (27.3%)	121
4	75 (75.0%)	25 (25.0%)	100
5	77 (52.7%)	69 (47.3%)	146
6	45 (51.7%)	42 (48.3%)	87
7	120 (61.5%)	75 (38.5%)	195
8	86 (76.1%)	27 (23.9%)	113
9	156 (67.5%)	75 (32.5%)	231
10	69 (88.5%)	9 (11.5%)	78
Total	46,722 (98.5%)	713 (1.5%)	47,435

2) Hardware and software

The software (SW) and hardware (HW) environment used for the experiment is shown in Table 4. For hardware, a desktop PC embedded with GPU was used. Mainly Python was used for software, and the Pandas and Numpy modules were mainly used for data analysis. For learning, the Scikit-learn module was used. The Python language cannot be smoothly executed if version numbers are different.

Table 4: HW and SW as experimental environments

H/W	CPU	Intel® Core^{TM} i5-6300HQ CPU @ 2.30GHz
	RAM	4GB
	HDD	128GB SSD, 1TB HDD
	GPU	Geforce GTX 950M
S/W	Module	Ver.
	Python	3.6
	Pandas	1.0.3
	Numpy	1.18.3
	Scikit-learn	0.22.2

4.1 Evaluation of window size

As described above in 3.2, window size needs to be decided to determine the feature set through time series data. Table 5 shows the fault diagnosis system's performance depending on the size change of window targeting the evaluation data. However, the causes of faults were not found in the experiment. The total number of the data in Table 4 shows difference by one. The reason is because one piece of raw data is needed, as the window size increases by one. As Table 5 shows, the best outcome was shown when the window size was 2. This means that excessively dealing with previous information is not significantly helpful for deciding classes (categories).

Table 5: Accuracy variations according to window size

Window size	No. of Correction	No. of Error	Total	Accuracy
1	486	226	713	68.30
2	619	93	712	86.93
3	612	99	711	86.07
4	605	105	710	85.21
5	595	114	709	83.92

4.2 Performance evaluation

1) Accuracy of faults

Table 6 shows the proposed system's performance that classifies all fault causes. The accuracy of fault diagnosis was 86.93%, and there is still room for improvement. In general, macro average is higher than micro average in terms of classification accuracy. However, the macro average is lower than the micro average in this case, for the reason of accuracy deviation being too severe depending on each fault cause (See Table 3).

Table 6: Accuracy of each fault

Label	No. of correction	No. of error	Total	Accuracy
0	183	2	185	98.91
1	60	13	73	82.19
2	62	12	74	83.78
3	33	0	33	100.00
4	21	4	25	84.00
5	55	14	69	79.71
6	13	29	42	30.95
7	74	1	75	98.66
8	27	0	27	100.00
9	57	18	75	76.00
10	34	0	34	100.00
total	619	93		
Macro average			84.92	
Micro average			86.93	

2) False alarm rate and specificity

To analyze false alarm rate, this paper analyzed problems using binary classification, classified by fault and normal state, instead of classifying each cause of fault. The result is revealed as a confusion matrix, as shown in Table 7. In Table 7, TP (true positive) is the frequency that a correct answer predicted faults as normal state, and FP (false positive) is the frequency that a correct answer predicted faults as normal state. Further, FN (false negative) is the frequency that a correct answer predicted normal state as faults, and TN (true negative) is the frequency that a correct answer predicted faults as faults.

Table 7: Confusion matrix for faults

		Correct answer		
		Normal	**Fault**	**Total**
Prediction	**Normal**	152 (TP)	33 (FP)	185
	Fault	2 (FN)	525 (TN)	527
	Total	154	558	712

The system performance can be examined from various aspects when Table 7 is used, and Table 8 reveals the results. Precision is the measurement indicating how precise it is to predict normal state, and the precision of the system is 82.16%; this was high in terms of the ratio of judging normal state as an error. This measurement needs to be greatly improved. Recall (rate) is the measurement of how accurately normal state was predicted. The recall (rate) of the system was 98.7%, which was very high. This signifies that the ratio of judging normal state as faults is very low. F1 measurement (F1 score) is the harmonic mean of precision and recall (rate), and the F1 score of the system was 89.7%. False alarm rate is the ratio predicting faults as normal state, and the system's false alarm rate was 5.9% which was very high. This part should be improved sharply. At this time, there will be some irrationality if the system is applied to industrial sites, as the system can judge faults as normal state. However, this result is caused by a sharp difference between learning data distribution and evaluation data distribution. Specificity is the ratio to predict faults as faults, which can be expressed as 1-FA. The system's specificity was 94.0%. In conclusion, as there is room for improvement in terms of false alarm rate and specificity, improvements need to be made, rather than applying the system to the industrial sites right away. Significant improvements are conjectured by collecting more learning data and applying imbalance learning.

Table 8: Metrics and its values

Metric	Formula	Value
Precision (P)	TP / (TP + FP)	0.822
Recall (R)	TP / (TP + FN)	0.987
F1 score (F1)	2 X P X R / (P + R)	0.897
False Alarm rate (FA)	FP / (FP + TN)	0.059
Specificity (SP)	TN / (FP + TN)	0.941

4.3 Error analysis

Table 9 shows the confusion matrix between classes (categories), and it reveals an analysis of Table 7 in detail. The values diagonally located are the accurately classified numbers. For example, if the value on the first column and 0th row is 13, this means that the wrongly classified number of the class (category) 1 (Flow 1) as category 0 (normal state) is 13. Overall, there are many numbers that are not 0 on the 0th row. This means the errors are classified as normal state, and it is dominant that most errors are classified as normal state, as mentioned in 4.3 above. This seems to be derived from data imbalance.

Table 9. Confusion matrix for classes

Label	0	1	2	3	4	5	6	7	8	9	10
0	183	0	1	0	0	1	0	0	0	0	0
1	13	60	0	0	0	0	0	0	0	0	0
2	12	0	62	0	0	0	0	0	0	0	0
3	0	0	0	33	0	0	0	0	0	0	0
4	4	0	0	0	21	0	0	0	0	0	0
5	10	0	0	0	0	55	0	0	0	0	0
6	29	0	0	0	0	0	13	0	0	0	0
7	1	0	0	0	0	0	0	74	0	0	0
8	0	0	0	0	0	0	0	0	27	0	0
9	18	0	0	0	0	0	0	0	0	57	0
10	0	0	0	0	0	0	0	0	0	0	34

5. Conclusions

This paper proposed a fault diagnosis system of BWTS among eco-ship equipment using SVM. The proposed system judges the status of faults of the BWTS with information received from five types of sensors using the classification model. Through an experiment, the mean accuracy of each fault cause was 86.93%, and the system contains much room for improvement. The false alarm rate was 5.9%, and so the proposed diagnosis system is currently considered unsuitable for application on industrial sites. This paper shows the results of the very initial stages, and therefore great improvements can be made, if more learning data are gathered and the problem of imbalance between classes (categories) is solved. Further, improvement of the diagnosis system can be achieved through data augmentation and imbalance learning. Machine learning algorithms can also be improved by applying a neural network model, such as RNN, rather than SVM.

Acknowledgments

This paper is based on research conducted by the Ministry of Trade, Industry and Energy in 2019 with the support of the Korea Industrial Complex Corporation project to enhance the competitiveness of industrial clusters. And this Work was partially supported by the Ministry of Education of the Republic of Korea and The National Research Foundation of Korea(NRF-2019M3E8A1103533).

References

[1] http://www.imo.org/en/OurWork/Environment/PollutionPrevention/AirPollution/Pages/Technical-and-Operational-Measures.aspx

[2] A. Mouzakitis, “Classification of fault diagnosis methods for control systems”, Measurement and Control, 46(10):303-308, 2013.

[3] EC. Kim, Consideration on the Ballast Water Treatment System Technology and its Development Strategies, Journal of the Korean Society for Marine Environmental Engineering, 15(4):349-356, 2012

[4] A. Géron, Hands-On Machine Learning with Scikit-Learn, Kera and TensorFlow: Concepts, Tools, and Techniques to Build Intelligence Systems, 2nd eds, O’Relliy, 2019.

[5] R. J. Williams, G. E. Hinton, D. E. Rumelhart, “Learning representations by back-propagating errors”, Nature. 323(6088):533–536, 1986.

[6] C. Cortes and V. N. Vapnik, “Support-vector networks”, Machine Learning. 20(3):273–297, 1995.

[7] Y. Yu D. Woradechjumroen, and D. Yu, “A review of fault detection and diagnosis methodologies on air-handling units”, Energy and Buildings 82:550-562, 2014.

[8] D. Filbert and L. Metzger, “Quality test of systems by parameter estimation”, Proceedings of the 9th IMEKO Congress, 1982.

[9] A. Giantomassi, Modeling, Estimation and Identification of Complex System Dynamics: Issues and Solutions, Ph.D. Dissertation, Università Politecnica delle Marche, 2012.

[10] M. Lind and X. Zhang, “Functional modelling for fault diagnosis and its application for NPP”, Nuclear Engineering and Technology, 46(6): 753-772, 2014.

[11] A. Giantomassi, F. Ferracuti, S. Iarlori, G. Ippoliti and S. Longhi, “Signal based fault detection and diagnosis for rotating electrical machines: Issues and solutions”, In Studies in Fuzziness and Soft Computing; Springer: Berlin/Heidelberg, Germany, 319:275–309, 2015.

[12] L. Ciabattoni, F. Ferracuti, A. Freddi, and A. Monbteriu, “Statistical spectral analysis for fault diagnosis of rotating machines”, IEEE Transactions on Industrial Electronics, 65(6):4301-4310, 2018.

[13] R. Xu and D. Wunsch, “Survey of clustering algorithms”, IEEE Transactions on Neural Networks, 16(3):645-678, May 2005.

[14] V. J. Hodge, J. Austin, “A survey of Outlier Detection Methodologies”, Artificial Intelligence Review. 22(2):85–126, 2004

[15] C. Stefano, C. Sansone, and M. Vento, “To reject or not to reject: That is the question– an answer in case of neural classifiers”, IEEE Transactions on Systems, Management and Cybernetics, 30(1): 84–94, 2000

[16] S. Bybers, and A. E. Raftery, Nearest-neighbor clutter removal for estimating features in spatial point processes”, Journal of the American Statistical Association, 93(442): 572-584, 1998.

[17] D. Yu, G. Sheikholeslami, and A. Zhang, “Findout: Finding outliers in very large datasets”, Knowledge and Information Systems, 4(4): 387-412, 2002.

[18] V. Vapnik. The Nature of Statistical Learning Theory, New York: Springer-Verlag New York, 2000.

[19] B. Krawczyk, “Learning from imbalanced data: Open challenges and future directions”, Progress in Artificial Intelligence 5:221–232, 2016.

[20] Y. Hu, X. Bai, P. Zhou, F. Shang, and S. Shen, “Data augmentation imbalance for imbalanced attribute classification”, arXiv:2004.13628, 2020.

[21] E. Rendón, R. Alejo, C. Castorena, F. J. Isidro-Ortega, and E. E. Granda-Gutiérrez, “Data sampling methods to deal with the big data multi-class imbalance problem”, Applied Science, 10, 1276. 2020.

[22] R. Chitrakar and H. Chuanhe, “Anomaly detection using support vector machine classification with k-medoids clustering”, Proceedings of the Third Asian Himalayas International Conference on Internet, pp. 1–5. 2012.

[23] C. M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.

Jae Kyun Kim received the B.S. and now in M.S course in Computer Engineering from Korea Maritime & Ocean University in 2018. His research interests include Natural language processing and Machine learning.

Jae-Hoon Kim received the M.S. and Ph.D. in Computer Engineering from Korea Institute of Science & Technology in 1988 and 1996, respectively. He work at Electronics and Telecommunications Research Institute 1988-1997, He has been a Professor for Korea Maritime & Ocean University since 1997. His research interests include Natural language processing, Information retrieval, Corpus linguistics, Sentiment analysis.

Seong Dae Lee received the M.S. and Ph.D. in Computer Engineering from Korea Maritime & Ocean University in 2001 and 2007, respectively. He has been a research professor in Korea Maritime & Ocean University since 2009. His research interests include Database, Big Data Analysis, Data Mining and Machine Learning.