

Received December 26, 2021, accepted January 4, 2022, date of publication January 7, 2022, date of current version January 20, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3141074

Fault Diagnosis of Elevator Doors Using Control State Information

HYUN CHAE¹, JAE EUNG LEE¹, AND KI-YONG OH^{1,2}

¹Department of Mechanical Engineering, Chung-Ang University, Seoul 06974, Republic of Korea

²Department of Mechanical Convergence Engineering, Hanyang University, Seoul 04763, Republic of Korea

Corresponding author: Ki-Yong Oh (kiyongoh@hanyang.ac.kr)

This work was supported in part by the Chung-Ang University Graduate Research Scholarship in 2020; in part by the Research Fund of Hanyang University under Grant HY-20210000002906; and in part by the Core Technology Development Project of Machinery Industry through the Development of Elevator Operation Management Technology Based on Predictive Preservation, funded by the Ministry of Trade, Industry & Energy (MOTIE, South Korea) under Grant 20011249.

ABSTRACT In this study, an integrated framework involving state classification, preprocessing, and classification is proposed for the fault diagnosis of elevator doors using control state information. During state classification, the door state is classified as operational or non-operational; moreover, based on the state information of the control board of an elevator door, the corresponding operational conditions are classified as opening or closing states. In the preprocessing phase, data processing and interpolation and feature manipulation are performed. Data are interpolated to synchronize each measurement with respect to the reference time; then, the state information is manipulated to create distinct features. In the classification phase, the operational states are classified, and nonlinear coordinate transformation is executed to transfer several features into a new nonlinear hyperspace by using an autoencoder. Two manifold features are extracted from the latent layer of the autoencoder; these become the principal axes for state classification using a support vector machine, indicating that three states are possible: normal with and without completion of a full stroke and abnormal stroke. The effectiveness of the proposed method was verified by performing field measurements on a control board during the operation of an elevator. Specifically, the autoencoder in the integrated framework effectively transformed an original space into a highly nonlinear yet efficient manifold, where classification was easy for different operational states. The proposed framework can be effective for real-world detection of elevator door faults because it exclusively uses state information measured from control boards.


INDEX TERMS Autoencoder, elevator door, fault classification, fault diagnosis, SVM.

I. INTRODUCTION

Elevators are a convenient and essential means of transportation in high-rise buildings. Research has shown that 54% of the world's population currently lives in urban areas [1]; thus, elevator systems have become indispensable components of apartments, commercial facilities, and office buildings. Elevators comprise several components, including mechanical, electrical, electronic, automatic control, and construction complexes; thus, faults in any elevator component may result in the stoppage of the elevator service. This downtime causes inconvenience to people and, in the worst case, can result in multiple passenger fatalities [2]–[4]. To guarantee the safe operation of elevators, monitoring the elevator status and

detecting elevator system failures are essential, indicating the necessity of an effective fault diagnosis method.

Several studies have been conducted on the detection of faults in elevator components, such as winding machines, wire ropes, elevator doors, and guide rails [3], [5]–[7]. Among these components, elevator doors are considered as a critical component because if they are affected by faults, multiple passenger fatalities may occur. Therefore, numerous studies have been conducted to detect abnormalities in elevator doors by deploying many effective sensors, including optical cameras, microphones, accelerometers, and gyros [3], [8]–[11]. The hybridization of both acceleration and control information has been suggested to enhance the estimation accuracy [12]. Furthermore, studies on the monitoring and diagnosis of the elevator status in real time using Internet of Things (IoT) information have been

The associate editor coordinating the review of this manuscript and approving it for publication was Jad Nasreddine .

conducted [13]–[15]. In these studies, additional sensor systems were employed to monitor elevator door systems, increasing that the total cost of elevator systems. Because of the competitive nature of the market, where the importance of economic feasibility in the installation of elevators is paramount, the increased costs prevent the regular deployment of fault-detection systems in commercial units. To reduce additional costs, a diagnostic method that uses only elevator state information was proposed [16]. In that study, normal operational states were predefined based on all the control-state information. Then, the state information violating the predefined normal operation was logged and reported to expert systems. These systems evaluated whether the newly sent information was faulty. However, the study did not involve an analysis of the quantitative comparison of the estimation accuracy of the proposed method with that of other methods.

Many novel methods have been studied from an academic perspective by measuring fault data using several sensors. Specifically, many machine-learning methods show excellent classification results and outperform traditional signal-processing-based methods, indicating that fault detection using artificial neural networks (ANNs) is promising [7], [17]–[19]. These methods require the extraction of appropriate features. Hence, statistical methods have been widely deployed for feature extraction from measurements, including mean, root-mean-square (RMS), kurtosis, and crest factor for fault classification [20]–[23]. In addition, deep-learning methods have been recently developed to extract and classify features in an end-to-end manner [24], [25]. Because of these features, minimal engineering effort is required for precise feature extraction and selection if sufficient data are available.

Machine-learning methods are based on empirical risk minimization (ERM) principles [26], indicating that these methods are trained with a focus on cost minimization. However, ERM does not minimize the actual error in the fault diagnosis because of two critical factors. One is the acquisition of insufficient training samples, and the other is the lack of an appropriate architecture [27]. These factors are also problems in end-to-end deep-learning architectures [28], [29], implying that the extraction of features to separate faults is difficult because of insufficient abnormal data. It is difficult to provide sufficient fault data for the training of neural networks in most industrial systems, including elevator door systems; faults rarely occur in real-life instances. Moreover, different industrial systems have different complex characteristics and nonlinearity, suggesting that the selection of an appropriate neural network architecture requires considerable engineering efforts.

To overcome the limitations of both hardware and the method, an integrated framework is proposed in this study for the fault diagnosis of elevator doors using only control state information. The proposed method is cost efficient because it does not require additional sensor systems. The proposed framework is divided into three phases: state classification,

preprocessing, and fault classification. The first phase involves the classification of the door state as operational or non-operational. Distinct features are effectively manipulated during the preprocessing phase. The combination of an autoencoder and a support vector machine (SVM) is implemented in the classification phase as a solution for imbalanced learning [30]. This approach effectively transfers high-dimensional data to a low-dimensional manifold [31], generating reduced low-dimensional data and preserving the key characteristics of high-dimensional inputs. Hence, the proposed framework ensures a high reliability, robustness, and accuracy. Specifically, the autoencoder effectively extracts distinct features and transforms them into a two-dimensional (2D) manifold map. Consequently, abnormal data are effectively separated from normal data because both data are located in different regions in the 2D manifold map. The linear SVM successfully classifies the current state among three different operations and faults with an optimal separating hyperplane (OSH). The effectiveness of the proposed method is verified by performing field measurements on a control board during elevator operation. The results indicate that the proposed framework exhibits excellent accuracy. Furthermore, in-depth discussions of the confusion matrix and quantitative comparisons with other machine-learning methods provide profound insights into the intelligent prognostics and health management of complex industrial applications.

The remainder of this paper is organized as follows. Section II details the experiments for field data measurements and data formats. The integrated framework for fault diagnosis of elevator doors is described in Section III. Section IV provides the analysis and discussion of the results of applying the proposed method to field measurements. Section V summarizes the conclusions and presents possible avenues for future research.

II. EXPERIMENTS

Data were collected for the period from April 01, 2020 to April 30, 2020. A testbed elevator is located in apartment X in Seoul, Korea. This elevator is hereafter designated as “Elevator X” for confidentiality purposes. The total number of the data records is 18 603 367.

The control system of an elevator door measured 17 state information, as listed in Table 1. The data comprised one time step and 16 other state information relevant to door control. Specifically, six sets of data were measured in the form of a digitized integer with a varied measurement range, including door floor, door position, door reference/feedback speeds, and door reference/feedback torques, whereas the others were binary signals, including door open/close, multi-beam, safety edge, open/close limit, open/close command, hatch door limit, and gate door limit. This information is generally measured for elevator door control but not limited to all cases. Some elevator manufacturers may measure different data for control purposes. These signals were measured for door control and not for diagnosis, indicating that the

proposed method can effectively detect faults and classify operational states without any new sensor installation.

TABLE 1. Raw data measured in a control board of an elevator door.

Data name	Description	Data type	Range	Unit
		yy.mm.dd		
Time	Time	+	-	-
		hh.mm.ss		
Door floor	Elevator location (store)		(2–31) (295–840)	floor
Door position	Door location		Min: close Max: open	mm
Reference speed	Command speed for a door motor	Digitized integer number	(0–60)	RPM
Feedback speed	Actual speed for a door		(0–60)	RPM
Reference torque	Command torque for a door motor		(0–40)	A
Feedback torque	Actual torque for a door		(0–40)	A
Close	Door close: 1 Otherwise: 0		(0,1)	-
Open	Door open: 1 Otherwise: 0		(0,1)	-
Multi-beam	Something blocks door: 1 Otherwise: 0		(0,1)	-
Safety edge	Something touches the safety edge : 1 Otherwise: 0		(0,1)	-
Close limit	Door fully closed : 1 Otherwise: 0	Boolean	(0,1)	-
Open limit	Door fully opened : 1		(0,1)	-
Close command	Command door closed: 1 Otherwise: 0		(0,1)	-
Open command	Command door open: 1 Otherwise: 0		(0,1)	-
Hatch door limit	Hatch door closed state : 1 Otherwise: 0		(0,1)	-
Gate door limit	Gate door closed state: 1 Otherwise: 0		(0,1)	-

III. METHODOLOGY

Fig. 1 displays the entire framework of the proposed methodology for the fault diagnosis and classification of elevator doors. The framework comprises three phases. The first phase is the state classification phase, in which the door states are classified as either operational or non-operational. Further, operational conditions are categorized according to either the open or closed state of the door by analyzing the state information on the control board of an elevator door. Subsequently, only the operational conditions are considered to identify the fault states of an elevator door. The second phase is data preprocessing, which involves data interpolation and feature manipulation. First, data are interpolated to synchronize each

measurement with respect to the reference time with a uniform internal for each stroke because actual measurements are randomly varied from 8 to 10 Hz data. Then, the state information with a uniform internal is manipulated to create 28 categories of features based on the state information on a control board. The third phase is the classification of an operational state with distinct features as normal without interruption, normal with interruption, or faulty. In this phase, a nonlinear coordinate transformation is executed to transform several features into a new nonlinear hyperspace by using the autoencoder. Specifically, two major features are extracted from the latent layer of the autoencoder. Then, these 2D features become the main principal axes for state classification by employing the SVM, obtaining the current state among the three states of the elevator door: normal with completion of a full stroke, normal without completion of a full stroke, and abnormal stroke, that is, an elevator door fault. The first state indicates that the elevator door is opened or closed automatically, whereas the second state indicates that the elevator door is opened or closed with interruption during the opening or closing process. The third state indicates a problem or fault in the elevator door.

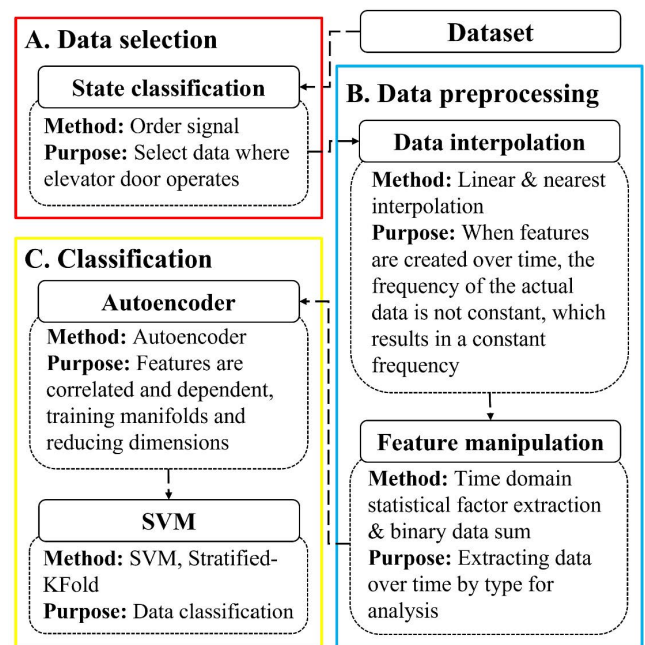


FIGURE 1. Proposed framework for fault diagnosis and classification on elevator doors.

A. PHASE A: STATE CLASSIFICATION

Elevator door states are classified into three categories: non-operational, in operation while a door is opening, and in operation when a door is closing. A non-operational state occurs when the elevator is stationary. An elevator starts to operate when a passenger touches the elevator button from a certain floor. After the elevator reaches the floor where the waiting passenger made a request, the door opens. The passenger enters the elevator and selects the button for the

desired floor, inducing the elevator to move to the selected floor. During the elevator motion, the door is always closed owing to the activated closing command. By contrast, an elevator door can be in an open or closed state when stationary on a certain floor. It is difficult to identify the normal operational state of an elevator door because open or closed operations are frequently interrupted by passengers. Examples of such interruptions include accidental touches by passengers, which can occur for several reasons. This circumstance of elevator operation indicates the potentially varied and complex nature of the open or closed elevator door states. The open state starts when an elevator reaches the selected floor or when passengers inside an elevator touch an open button in a control panel on a certain floor. In this state, the open command is activated, which launches a control protocol for opening a door system. The open limit is activated when the door position reaches a predefined maximum value. Feedback states, including feedback speed and torque, are also measured and used for open-door control. This information is beneficial for the detection of faults, because the difference between the reference and feedback represents an important metric of fault diagnosis. In addition, the close limit, gate door limit, and hatch door limit are released in the open state, implying that this binary-numbered information is also effective for identifying elevator door faults.

The closed state is activated when the elevator is in motion. All state values are monitored to check the door states during this period because opening a door during the operation of an elevator might result in severe accidents, including passenger fatalities. During this period, close, hatch, and gate limits are activated, and the door position is at the minimum value.

Fig. 2 presents the state information measured in the control system during the opening and closing of an elevator door. Once the open command is activated (Fig. 2(a1)), the reference speed and torque are input into the door motor, and the feedback speed and torque of the door motor are measured (Fig. 2(a2)). The feedback signals are used for closed-loop control of the door motor. Hence, the door is opened during the opening period (Fig. 2(a3)). Several Boolean data types are also measured to mitigate safety concerns (Fig. 2(a4)). This process is terminated when an open limit switch is activated. The elevator door closing process inversely mirrors the opening process. Initially, a close command is activated (Fig. 2(b1)). Then, the reference speed and torque are applied to a door motor, and the feedback speed and torque are measured for the closed-loop control of the door motor (Fig. 2(b2)), resulting in the closing process as the door motor changes the door position (Fig. 2(b3)). Several sets of Boolean information are also measured for safety (Fig. 2(b4)) and control perspectives. This closing process terminates when a close-limit switch is activated. Each process is considered to be an independent stroke of the open or closed state, and each stroke is analyzed because operational information provides useful data for the detection of faults compared to operational data during the non-operational period. Note

that the analysis of non-operational data is cost-ineffective because the non-operational time is considerably longer than that considering the operational characteristics of elevators.

Based on the operational data, two operational conditions can be classified: completion of a full stroke and normal without completion of a full stroke. Specifically, states were defined as full strokes when open or closed limits were activated, whereas elevator doors were classified as the other condition when they were inactivated. Then, abnormal strokes were classified by expert systems of the elevator, who actually conducted elevator operation and maintenance, because identifying abnormal strokes would be difficult considering the complex nature of the elevator operation. Abnormal strokes generally show different characteristics than normal strokes. Specifically, the measured information features a slow change in door position over time, a large difference between the feedback speed and reference command speed, no movement in the door position, initial locking of the door by the hatch door and gate doors, and limited movement of the door. These characteristics indicate the diversity of abnormal strokes, which hinders their definition in a single-feature format. Fig. 3(a) shows the state information of an abnormal stroke during an open period. Once the open command was activated (Fig. 3(a1)), the reference speed and torque were input into the door motor, and the feedback speed and torque of the door motor were measured (Fig. 3(a2)). The feedback speed was considerably lower than the reference speed during 4–6 s, resulting in a higher torque (Fig. 3(a2)) than that in the normal state (Fig. 2(a2)). Therefore, the door did not move during 5–8 s and then moved slowly (Fig. 3(a3)). Finally, the opening was interrupted, and the door was stopped (Fig. 3(a3)); therefore, the open-limit signal was not activated at the end of the opening process (Fig. 3(a4)). Other Boolean signals were not activated in Fig. 3(b4), except for the closed feedback signal at the end of the opening process, implying that these signals would provide effective information regarding faults. Fig. 3(b) shows the state information of the abnormal stroke during a closed period. In this abnormal state, the open limit signal is not activated, whereas the door-closing process normally starts with the activation of an open limit signal (Fig. 3(b4)). Specifically, the reference speed and torque were input to the door motor when the close command was activated (Fig. 3(b1)), and the feedback speed and torque of the door motor were measured (Fig. 3(b2)). The door moved slowly compared with the normal state (Fig. 2(b2)). Finally, the closing process ended without completion and the door was stopped (Fig. 3(b3)).

B. PHASE B: PREPROCESSING

This subsection describes phase B (preprocessing), which comprises data interpolation and feature manipulation. The details of each process are as follows.

State information relevant to the elevator door is collected from the door control system. Preliminary analysis of time stamps on state information (Table 1) suggests that the measurement frequency of the data varied

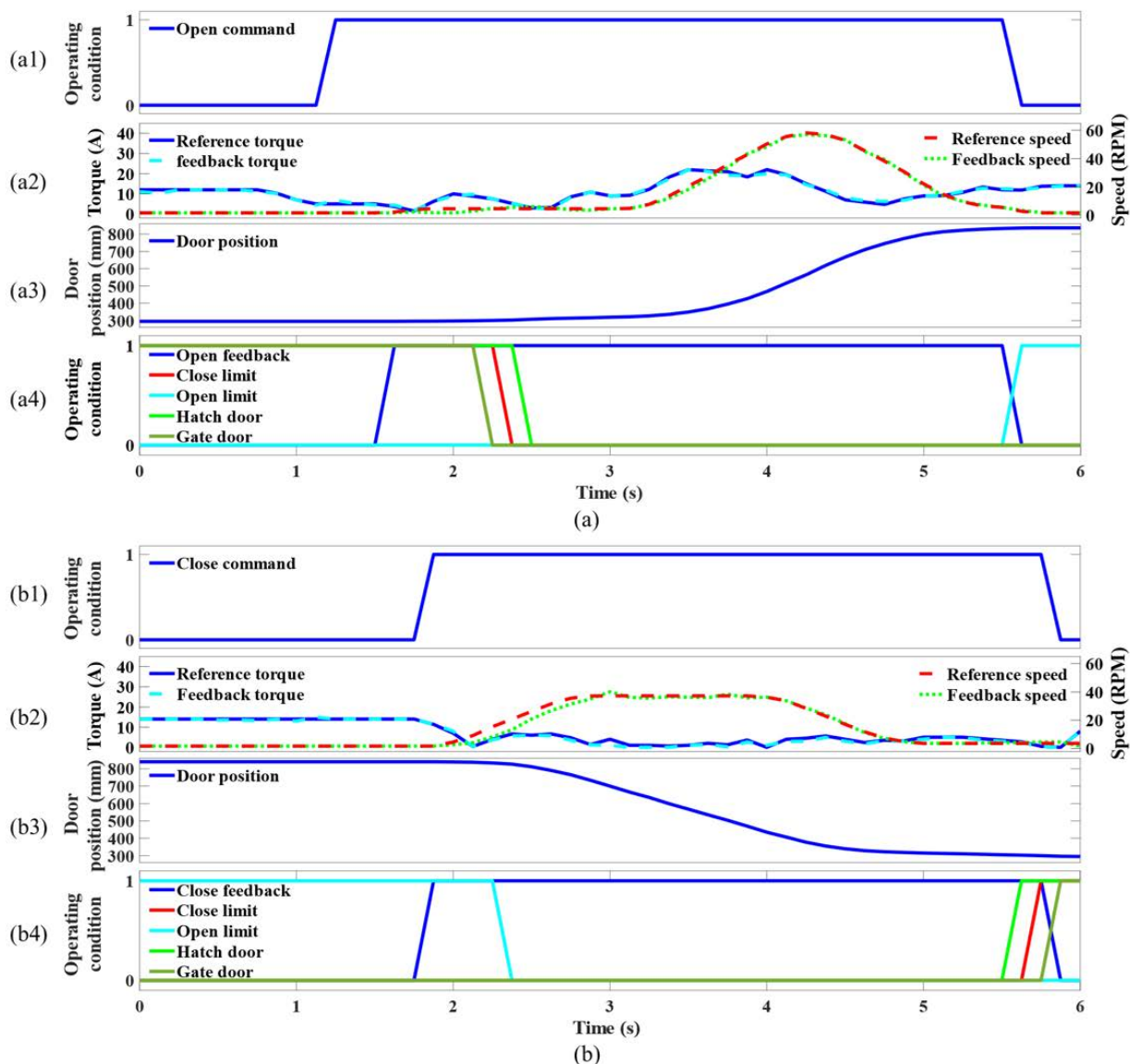


FIGURE 2. State information in a control system of an elevator door during normal (a) open and (b) closed periods.

between 8 and 10 Hz depending on the communication load, although the specification of the door control system indicates that all data were measured at 10 Hz. Hence, all state information was interpolated to ensure a uniform interval of measured states, because it was necessary to synchronize all information to the same time stamp for each stroke using the proposed method. To minimize deterioration and loss of data, the interpolation frequency was set to the lowest frequency of 8 Hz. A linear interpolation method with a round off was executed for a type of digitized integer number, including the door floor, door position, reference/feedback speeds, and reference/feedback torques. The linear interpolation method resulted in small differences, which were negligible when considering the measurement frequency, compared to the results of the actual measurement. Moreover, the Boolean

state information was synchronized using the most accessible method because this is binary information [32].

Distinct features were extracted using a manipulation process. Fig. 4 shows a flowchart of feature manipulation. Overall, 11 measurements were obtained from the 16 state information listed in Table 1. Five state information of the operation floor, open/close command, multi-beam, and safe edge were eliminated because they are less relevant to door faults. The state information for the multi-beam and safe edge was measured to prevent passengers from being trapped between doors. Open or closed commands were always activated during open or closed strokes. Moreover, the operating floor exclusively provided the location of the elevator. Statistical analysis of five state information with a type of digitized integer was conducted to extract statistical features, including

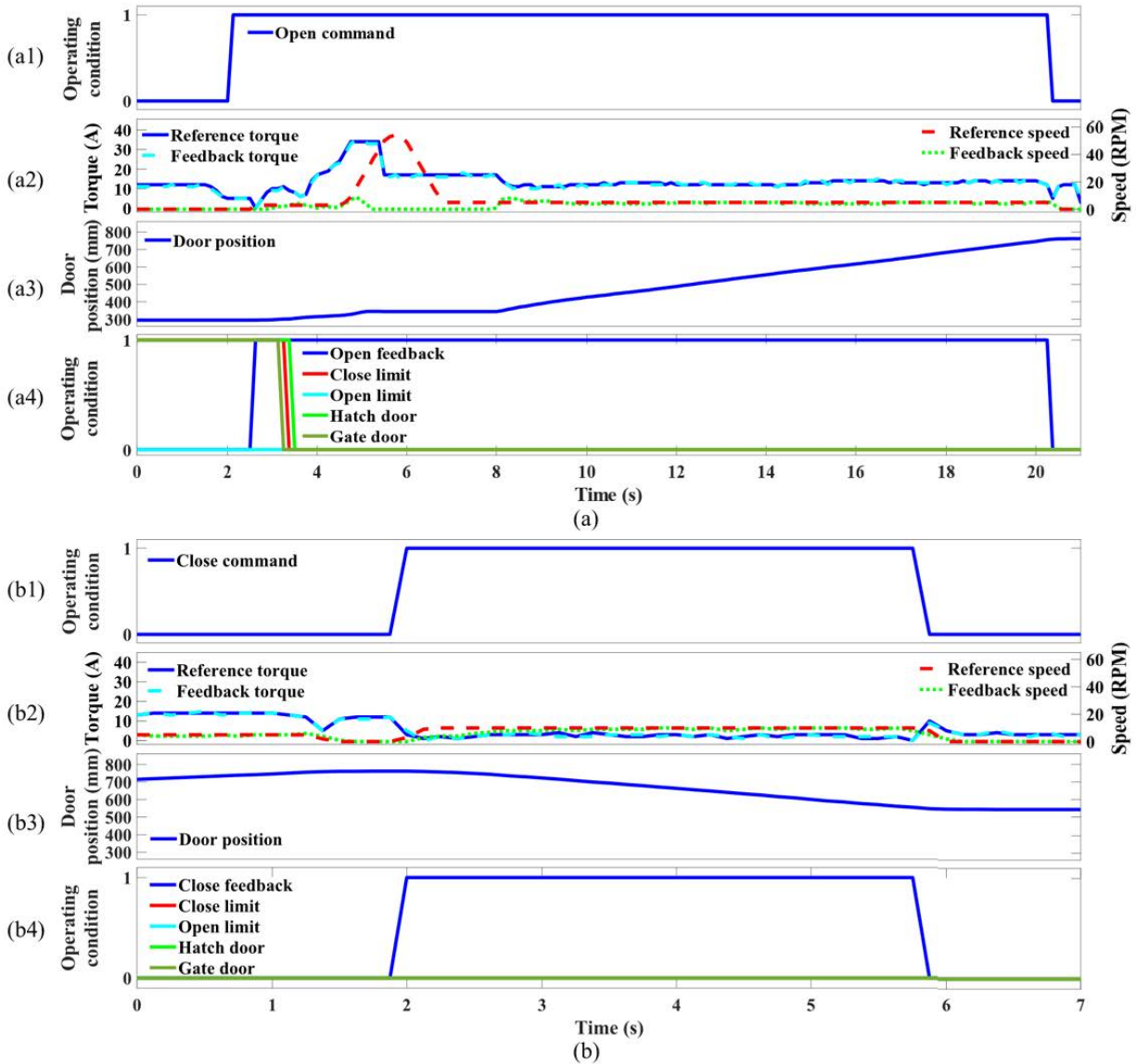


FIGURE 3. State information in a control system of an elevator door during abnormal (a) open and (b) closed periods.

the mean, RMS, and maximum of each feature for each stroke. These three features were addressed because they are effective for fault detection [23], [33]. Differences between the reference speed/torque and feedback speed/torque were also calculated to create two additional features because they provide crucial information regarding elevator door faults, resulting in a total of seven state information. Hence, statistical analysis of this seven-state information resulted in a total of 21 features for each open or closed stroke. In addition, six Boolean operational states were synchronized with respect to a timestamp, which started at zero when the open or close command was first activated. Thereafter, these six sets of binary information were summed to create six features for each open or closed stroke. Additionally, the total operational period of each stroke depends on various situations

and interruptions from passengers; thus, it was calculated as a distinct feature. This indicates that the six features from the binary state information were highly correlated with the total operational period of operation at each stroke. In summary, 28 features were manipulated through this phase, comprising seven features related to operational states and 21 features extracted from statistical analysis.

These features WERE normalized before training because the scales of each feature vary owing to the different physical quantities. Note that convergence is not guaranteed without normalization, and oscillations may occur during the optimization of the autoencoder. The normalization results in a similar effect of the input features on the output prediction in the neural network of the autoencoder. Two methods are widely used for normalization: min-max normalization (1)

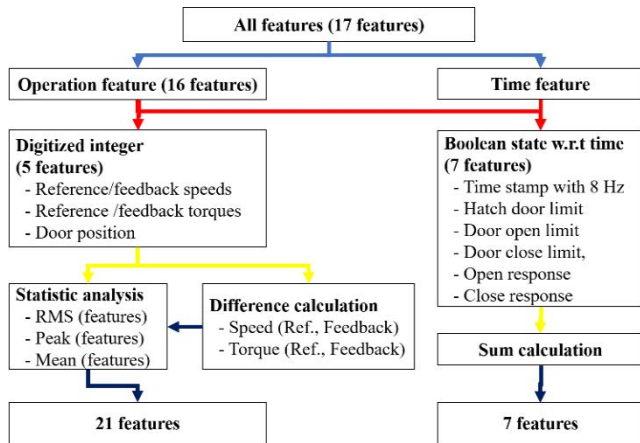


FIGURE 4. Flowchart for feature manipulation.

and z-score normalization (2), as follows:

$$x_{ij} = \frac{x_{ij} - x_{i,min}}{x_{i,max} - x_{i,min}}, \quad (1)$$

where x_{ij} , $x_{i,min}$, and $x_{i,max}$ denote the j th, minimum, and maximum values of feature i , respectively.

$$x_{ij} = \frac{x_{ij} - x_{i,mean}}{\sigma_i}, \quad (2)$$

where $x_{i,mean}$ and σ_i denote the mean value and standard deviation of feature i , respectively.

The min-max normalization changed the range of the features of interest to [0, 1]. This method is generally used for autoencoders with a sigmoid activation function because the range of sigmoid functions is the same. However, this method is difficult to use when the range of a feature is unpredictable. This problem can be solved by assuming an appropriate range for the normal value of each feature, and then considering outliers from a predefined range as the maximum or minimum values. By contrast, in z-score normalization, the range of each feature must not be defined. Moreover, the goal of the autoencoder is to create a manifold by learning more normal data than abnormal data [34], indicating that the scale of the normal data is important. This analysis suggests that z-score normalization is effective for detecting elevator door faults. Hence, the z-score normalization method is addressed in this study. The performance of the z-score normalization method is compared in detail in the Results and Discussion sections.

C. PHASE C: CLASSIFICATION

This subsection describes phase C (classification), comprising the autoencoder and SVM. The autoencoder is a fully connected symmetric neural network composed of an encoder and a decoder. This neural network enables unsupervised learning to reconstruct features with the same characteristics and dimensions as input features [31]. The nonlinear activation functions at each layer enable nonlinear space transformation, implying that data interpretation is easier in a transformed hyperplane. Moreover, the autoencoder has a

role in dimension reduction when features in the latent layers are used. In this study, two major features were used in the latent layer as 2D manifold because they provide sufficient information for door-fault detection.

To optimize the architecture of the autoencoder, the activation function for each layer and the number of nodes per layer should be determined. First, activation functions commonly used in machine learning include the rectified linear unit (ReLU), scaled exponential linear unit (SELU), sigmoid, and hyperbolic tangent. Note that a linear activation function is not used in the autoencoder because this function results in a linear transformation between the input and output and thereby yields the same result as principal component analysis (PCA). Hence, the proposed autoencoder applies only a linear activation function to the last layer to ensure the same range of data. ReLU and SELU were also excluded because the data could not be adequately restored. This factor can be explained by the fact that ReLU and SELU have filter roles below zero in the data. However, the autoencoder transforms the initial coordinates into hyperplanes for easy classification. Therefore, sigmoid and hyperbolic tangents were only considered as candidates for the activation functions of the autoencoder. Second, the number of nodes in each layer must be optimized. These two hyperparameters of the activation functions and the number of nodes at each layer were optimized by trial and error. Specifically, we changed the activation functions and numbers of nodes in each layer and monitored the loss of the object function with the Adam optimizer, which was defined by a mean square error. The architecture of the autoencoder with the smallest value of the object function was selected as the optimum architecture, as shown in Fig. 5(a).

The autoencoder comprises eight layers, with four layers each for the encoder and decoder. Herein, 28-dimensional input features were passed through 32-, 16-, 8-, and 2-dimensional layers in the encoder. Subsequently, 2-dimensional data are decoded into 8-, 16-, and 32-dimensional layers through the decoder and reach the final layer with 28-dimension, which is the same as the input data. The hidden nodes in each layer were trained to minimize the reconstruction errors between the original and reconstructed features. Sigmoid and hyperbolic tangent functions were used as activation functions in the encoder and decoder, respectively. Different activation functions for the encoder and decoder enabled a robust nonlinear space transformation.

Fig. 5(b) depicts the training process of the autoencoder, which comprises three stages: pre-training, unrolling, and fine-tuning. Pre-training sets the initial values of each layer by using a random sampling method. Specifically, the first 32-dimensional layer of the encoder and the seventh layer of the decoder were trained with 28 feature inputs and the same output, resulting in a weight of ω_1 and bias of b_1 . The same procedure was executed with different activation functions to determine the weight of ω_8 and the bias of b_8 . Subsequently, trained weights and biases of 32 dimensions were used to train the second layer of the encoder and the sixth layer of the

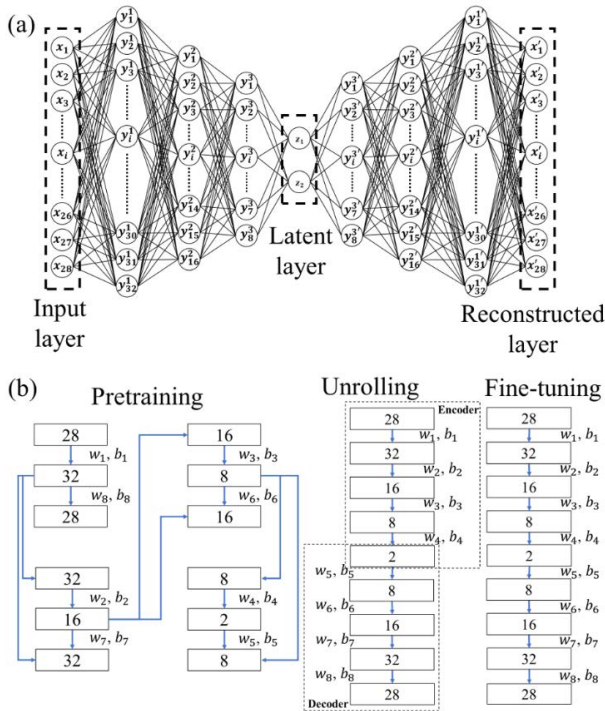


FIGURE 5. (a) Autoencoder used in this study and (b) training process of autoencoder.

decoder with 16 dimensions, resulting in a weight of ω_2 and bias of b_2 . Sigmoid and hyperbolic tangent functions were used to train the second layer of the encoder and decoder, respectively. This resulted in weights of ω_2 and ω_7 , and biases of b_2 and b_7 , for the encoder and decoder, respectively. The process was repeated to determine the weights and biases of the third and fourth layers of the encoder and decoder, respectively. After the pre-training process was completed, the layers on the system were unrolled to construct an entire autoencoder layer. Then, the weights and biases of the layers were fine-trained [31] for accurate prediction, resulting in n -dimensional manifolds for each layer. The architecture of the autoencoder indicates that the weights in the autoencoder form a transformation matrix that nonlinearly transforms distinct features in the original spaces to distinct features in a hyperspace, where the operational states can be easily classified. Furthermore, 2D features from the z -layer (fourth layer) were used in this study for the final classification.

The SVM was addressed in this manifold to classify three operational states: normal with completion of full stroke, normal without completion of full stroke, and abnormal stroke. SVM aims to select the OSH between classes by training samples that lie at the edge of the class distributions, which are referred to as support vectors [35], [36]. In SVM, the training dataset of N cases, represented by $\{x_i, y_i\}, i = 1, \dots, N, y_i \in \{1, -1\}$, is used to form a classifier. The OSH represents the maximum margin between classes, where a hyperplane is defined as

$$w \cdot x + b = 0. \quad (3)$$

Here, x , w , and b denote a point lying on the hyperplane, the weight defining an OSH that is normal to the hyperplane, and the bias, respectively. For a linearly separable space, a separating hyperplane can be defined for two classes. These two equations can be combined as follows:

$$y_i(w \cdot x_i + b) - 1 \geq 0, \quad (4)$$

$$\min \left\{ \frac{1}{2} |w|^2 \right\}. \quad (5)$$

The margin represents the vertical distance from the OSH of the data closest to the OSH that divides the two classes. The margin value is defined as the inverse of (5). The proposed method has three categories, each of which is separated by an OSH. Hence, the three OSHs generated category-specific regions on the 2D feature map through each plane. The determination of the OSH in the SVM is an effective method for elevator door fault detection because SVM learns points in a space to maximize the gap width between each category. This implies that the proposed method can guarantee robustness when data inequalities occur in the training set. Specifically, fault data are relatively small compared with normal operational data in most cases. This serves as the rationale for referring to SVM in this study.

D. EVALUATION

In this study, the K-fold cross-validation method was employed to evaluate the performance of the proposed fault detection method owing to the minimal data available on abnormal strokes in the measurements. Specifically, the main dataset was divided into 5 smaller datasets. One dataset was used as the test set and the other N-1 dataset was used as the training set. This process was iterated N times with different test sets at each iteration. Subsequently, the prediction accuracy was evaluated for each result using the F-score, which is an effective metric when there are minimal abnormal data in the dataset [37], [38]. The F-score is the harmonic mean of the model precision and recall and expressed as

$$F_{score} = \frac{2}{\alpha^{-1} + \beta^{-1}}, \quad (6)$$

where α is the recall and β is the precision. Recall is the fraction of samples classified as positive among the total number of positive samples, whereas precision is the fraction of true-positive samples among the examples classified as positive. Therefore, recall is defined as the number of true positives divided by the number of true positives and false negatives, whereas precision is defined as the number of true positives divided by the number of false positives and true positives. Equation (6) can be applied to the problem of only two categories; however, this study aimed to classify these three categories. Therefore, we used the modified F-score. Specifically, one category was set and the F-score was computed by grouping the remaining categories. Three iterations were executed for each category. The F-score for

each category was calculated using the following expression:

$$F_{score} = \frac{\sum_{k=1}^N F_k}{N}, \quad (7)$$

where F_k is the F-score of each category and N is the total number of categories. Equation (7) is a macro F-score that calculates the F-score of more than two categories [39]. Therefore, the F-score obtained from (7) represents the overall performance of all categories. Finally, the calculated F-score was averaged for all iterations in which the K-fold cross-validation method was used. Moreover, the confusion matrix was addressed for a quantitative and reasonable analysis.

E. BAYESIAN OPTIMIZATION

The hyperparameters of the classifiers used in this study should be optimized to ensure the best results achieved for fair comparison. Hence, in this study, Bayesian optimization (BO) was applied to optimize the hyperparameters of all classifiers, which is a simple yet effective method widely used for hyperparameter optimization in machine-learning methods [40]–[42].

BO functions intervene in the construction of the posterior distribution because of the general fit of the Gaussian process to the results at a given data point, and select and use a new parameter space that is likely to be used in the next iteration. The choice of the second parameter at each iteration is determined by search strategies such as expectation improvement (EI) and defect posterior distribution, which determine the local maxima of the acquisition function. This process was repeated to determine the optimal variable. BO uses the initial values as prior knowledge, whose similar inputs produce similar outputs, and uses this learning to quickly refine and find optimal parameters. Specifically, $\mu(x)$ is the predictive mean function and $k(x, x')$ is the predictive variance function under the prior Gaussian process. Based on the current data point, D_n can predict the expected function, $f_{n+1}(x)$:

$$f_{n+1}(x) \sim GP(\mu(x), k(x, x')). \quad (8)$$

A new data point, x_{n+1} , is set based on function $f_{n+1}(x)$.

The new set data point is

$$x_{n+1} = \arg \max_x (\max\{0, f_{n+1}(x) - f_{max}\} | D_n). \quad (9)$$

It is the data point with the greatest expected improvement obtained by utilizing the expectation function, $f_{n+1}(x)$, and f_{max} is the maximum output value for the current data point. Equations (8) and (9) were repeatedly applied to search for the improved optimal value, x . Subsequently, the models were compared using the optimal hyper-parameter, x , when training was completed.

IV. RESULTS AND DISCUSSION

A. PHASE A: STATE CLASSIFICATION

The first phase of data selection in the proposed method (Fig. 1) resulted in 23 574 open strokes and 23 556 closed

strokes based on the open and close commands. This represents only 0.13% of the total data, suggesting that the elevator was mostly in a non-operational state because it started to operate only when passengers touched a button. Specifically, there were 22 466, 1 043, and 65 open strokes, and 22 466, 1 060, and 30 closed strokes, in the normal with completion of a full stroke, normal without completion of a full stroke, and abnormal states, respectively. This classification suggests the presence of two factors. First, random interruptions from passengers when the elevator door is in the process of opening or closing account for 5% of the total interruptions. Second, the probability of faults is less than 0.3%, suggesting that gathering fault data is significantly difficult. Validation of the proposed method is also challenging because there are significantly minimal fault data, suggesting that the K-fold cross-validation method should be used to ensure the reliability of the validation [43], [44]. The hyperparameters of the models were optimized using 20% of the BO data. Then, k-fold cross-validation was performed five times, except for the data used for BO, and it was divided into 80% training set and 20% test set for each learning. For the open stroke, the training set consisted of 14,378, 657, and 42 strokes, whereas the test set consisted of 3,594, 167, and 10. In the closed stroke, the training set consisted of 14,378, 678, and 19 strokes, whereas the test set consisted of 3 595, 170, and 5 strokes. Note that the number of open strokes was slightly larger than that of closed strokes because some closed strokes measured below 8 Hz were deleted considering communication problems.

B. PHASE B: FEATURE PREPROCESSING

1) STATISTICAL METHODS

Several statistical methods are effective for feature manipulation for fault classification, including skewness, kurtosis, crest factor, and shape factor [20], [22]. One common observation is that the estimation accuracy of fault diagnosis and classification may increase when more statistical features are addressed. Because this study addressed only three statistical features, we compared one extreme case with the proposed method to analyze this hypothesis. In the extreme case, nine additional statistical methods were applied to seven features, in addition to the mean, RMS, and peak, which were used in the proposed method, resulting in 84 features for the operational features of the digitized integer. Table 8 in the appendix shows the formulas of the statistical methods used in this extreme case. The input and reconstructed layer dimensions were changed to 91 by the proposed autoencoder according to the shape of the input data developed by additional time-related static factors. In this subsection, the proposed method refers to the first case, whereas the extreme case refers to the second case. Table 2 presents the classification results.

In the first case, all abnormal strokes were classified as abnormal in both open and closed states. All normal patients with complete stroke were correctly classified. Five of these states were identified as normal, without completion of the full stroke in closed states. Similarly, seventeen cases of normal without completion of full stroke were

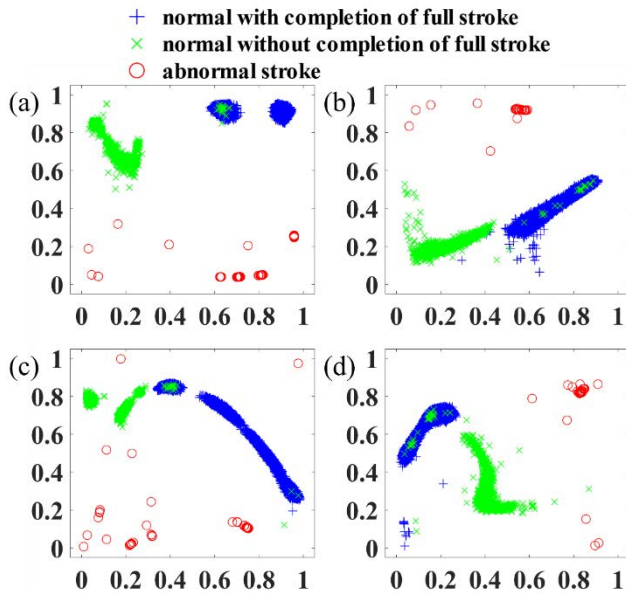


FIGURE 6. Manifold maps with three statistical features at the (a) open and (b) closed states and with 12 statistical features at the (c) open and (d) closed states.

TABLE 2. Classification results by applying different statistical methods; +, x, and o denote normal with completion of full stroke, normal without completion of full stroke, and abnormal stroke, respectively.

Statistical Method	State		True			Mean F-score	Average
			+	x	o		
Mean, RMS, and peak	Open	Predict	+	17967	17	0	0.996
			x	0	817	0	
			o	0	0	52	
	Closed	Predict	+	17968	16	0	0.996
			x	5	832	0	
			o	0	0	24	
12 methods in Table VII	Open	Predict	+	17967	17	0	0.989
			x	0	832	2	
			o	0	0	49	
	Closed	Predict	+	17965	23	0	0.988
			x	5	825	0	
			o	0	1	24	

classified as normal with completion of full stroke in the open state. Therefore, the average F-score of the proposed method was 0.996, suggesting that the proposed method identifies faults in elevator doors with high accuracy, as illustrated in Fig. 6(a) and (b). All abnormal strokes were less than 0.5, for the second primary axis, that is, the vertical axis in Fig. 6(a), whereas two normal strokes exceeded 0.5, for this axis for closed states (Fig. 7(b)). This suggests that the characteristics of abnormal strokes differ from those of normal strokes in the transformed manifold. However, a small amount of normal data without the completion of full stroke overlapped with the normal data with the completion of full stroke in the transformed manifold because the measured data for both

were similar. Specifically, interruptions from a passenger were triggered at the end of stroke completion, and thus did not have any effect on the operation of the elevator door. All the characteristics of the features in this case were very similar to that of the normal data with completion of full stroke, resulting in misclassification because the area overlapped in the manifold. However, this area was far from the abnormal stroke in the manifold and, thus, did not affect the classification of elevator door faults.

In the second case, two abnormal strokes were misclassified in the open state. Moreover, 17 and 29 misclassifications occurred in the open and closed states, respectively, resulting in an average F-score of 0.983, as shown in Fig. 6(c) and (d). Three abnormal strokes were located close to clusters of normal strokes without completion of full stroke in the open state, even though the abnormal stroke did not overlap with the normal strokes. This has resulted in the misclassification of abnormal strokes as normal strokes. One normal stroke was located between clusters of normal and abnormal states in the closed state. However, it was closer to abnormal strokes, resulting in misclassification. Some normal strokes also overlapped in this manifold in both the open and closed states, resulting in misclassification.

In summary, the dimensions of the input features cannot be increased owing to the challenges in hyperplane transformation. Thus, the three strokes were not clearly separated in the transformed manifold. A comparison of the two cases clearly shows that the characteristics of abnormal strokes differ from those of normal strokes when the hyperplane transformation is examined through the autoencoder. However, many input features encounter challenges in hyperplane transformation because of the dimension increase and the subsequent decrease in prediction accuracy for fault classification in the transformed manifold. Hence, the selection of appropriate features is important in accurate fault diagnosis.

2) FEATURE EXTRACTION

This subsection details the contribution of the features extracted in the fault classification. The proposed method employed 28 features, comprising 21 statistical features from digitized integers and 7 features from Boolean state information. Moreover, 21 statistical features, including 15 features 6 statistical features from 3 statistical methods, were applied to 5 operational data and 2 difference calculations. Four extreme cases were compared with the proposed method to demonstrate the effectiveness of the feature extraction. In the first case, 15 operational features with 7 Boolean features were used for fault classification. This case provided the contributions of the 6 difference calculations. In the second case, 15 features from digitized integers were exclusively used to analyze the contribution of the difference calculation, and 7 features were acquired from the Boolean state information. In the third case, 21 features from digitized integers were used, whereas 7 features relevant to the Boolean state were used in the fourth case. Different input features resulted in varied classification results, indicating the contribution of

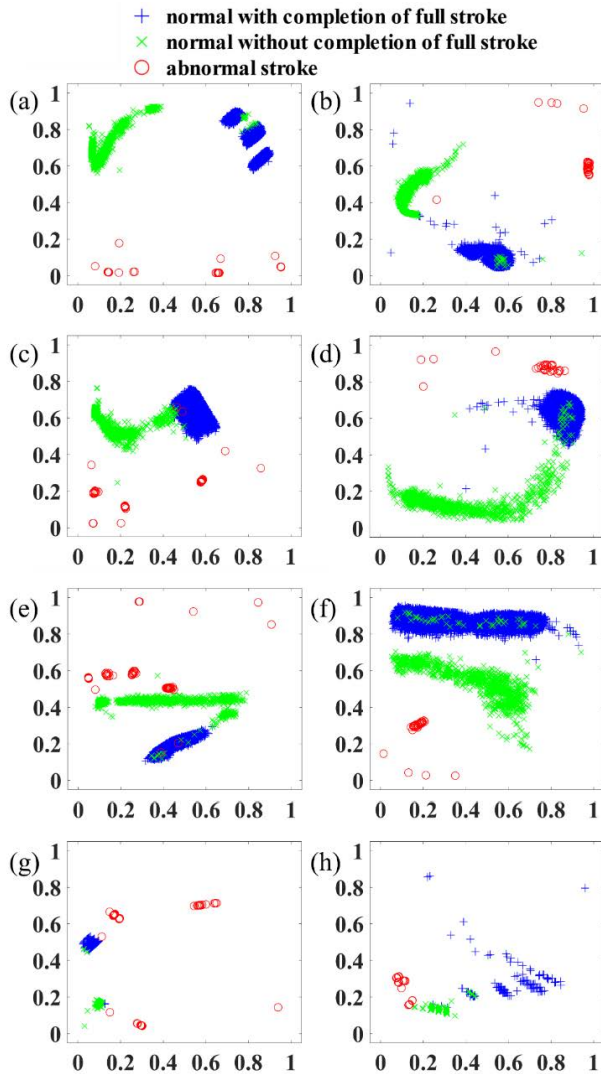


FIGURE 7. Manifold maps with the proposed method at open state with (a) 15 operational features + 7 Boolean features, (c) 15 operational features, (e) 15 operational features, and (g) 7 Boolean features. Manifold maps with the proposed method at closed state with (b) 15 operational data + Boolean features, (d) 15 operational features, (f) 15 operational features, and (h) 7 Boolean features.

each feature. The input and reconstructed layer dimensions of the autoencoder were converted into numbers of corresponding inputs for each case. Table 3 presents the classification results.

In the first case, apart from one abnormal stroke in the closed state, all abnormal strokes were classified as abnormal in the open and closed states. Seventeen instances of normal without completion of a full stroke were misclassified in the open state, whereas 12 and 16 instances of normal with and without completion of a full stroke, respectively, were misclassified in the closed state, resulting in an average F-score of 0.990. This score was lower than that of the proposed method, suggesting that six features from the difference calculation contributed to the increased fault classification accuracy. This result is explained by the manifold maps shown in Fig. 7(a) and (b). Normal strokes were separated from

TABLE 3. Classification results by applying different features; +, x, and o denote normal with completion of full stroke, normal without completion of full stroke, and abnormal stroke, respectively.

Case #	Features	State	Predict	True			Mean F-score	Average	
				+	x	o			
1	15 operational features + 7 Boolean features	Open	Predict	+	17972	17	0	0.996	0.990
				x	0	814	0		
	o	0	0	51					
	Closed	Predict	+	17966	16	0	0.985		
x			12	832	1				
				o	0	0	23		
2	15 operational features	Open	Predict	+	17972	17	2	0.983	0.988
				x	0	816	1		
	o	0	1	49					
	Closed	Predict	+	17969	26	0	0.994		
x			4	822	0				
				o	0	0	24		
3	21 operational features	Open	Predict	+	17971	11	2	0.988	0.990
				x	1	826	0		
	o	0	1	50					
	Closed	Predict	+	17973	38	0	0.992		
x			1	810	0				
				o	0	0	24		
4	7 Boolean features	Open	Predict	+	17969	17	1	0.988	0.972
				x	3	817	1		
	o	0	0	50					
	Closed	Predict	+	17970	14	0	0.955		
x			3	825	0				
				o	0	7	24		

abnormal strokes in both the feature maps of the open and closed states, similar to those in the first case. However, one abnormal stroke was located close to clusters of two normal strokes and far from a cluster of abnormal strokes in the closed state, even though the abnormal stroke did not overlap with the normal strokes. This has resulted in the misclassification of abnormal strokes as normal strokes. The manifold maps also demonstrated the overlap area for the two normal strokes, resulting in misclassification.

In the second case, all the abnormal strokes were classified as abnormal strokes in the closed state. However, two abnormal strokes were considered normal with the completion of full stroke, and one abnormal stroke was considered normal without the completion of full stroke in the open state. Several misclassifications occurred in the two normal stroke classes in both the open and closed states, resulting in an average F-score of 0.988. These results are explained by the manifold maps shown in Fig. 7(c) and (d). Specifically, two abnormal strokes were in a clustered area of normal with the completion of full stroke, and one abnormal stroke was very close to the clustered area of normal without the completion of full stroke, resulting in this misclassification. Classification accuracy decreased by neglecting the features of the difference calculation and Boolean states.

The third and fourth cases also had two misclassifications of abnormal strokes in the open state and several misclassifications in normal strokes, resulting in average

F-scores of 0.990 and 0.972, respectively. The corresponding manifold maps in Fig. 7(e)–(h) show that misclassified abnormal strokes overlapped or were near the clusters of normal strokes.

In summary, a comparison of four cases with the proposed method demonstrated that the characteristics of abnormal strokes differed from those of normal strokes in the manifold transformed through the autoencoder. However, the limited availability of information decreased the accuracy of fault classification. In addition, we observed that adding Boolean features increased classification accuracy, implying that the sum of time for the Boolean state was effective in improving estimation accuracy. Furthermore, difference calculations provided additional effective information for fault diagnosis and classification, thereby increasing the estimation accuracy.

3) NORMALIZATION METHOD

This subsection presents the classification results based on the normalization methods addressed in this study. A case study was conducted because the fault classification depends on the normalization method [45], [46]. Normalization has a critical role in the rescaling of inputs that measure different physical quantities, thereby resulting in similar weights for all features. Two normalization methods widely used in autoencoders are the z-score and min-max methods. Because the z-score was used in the proposed method, we employed the min-max method, expressed by (2), for comparison. Table 4 presents the results of this study.

In the case of addressing the min-max normalization method, three abnormal states were misclassified in both the open and closed states. Specifically, two abnormal strokes were classified as normal with completion of full stroke, and one abnormal stroke was considered normal without completion of full stroke in the open state, whereas three abnormal strokes were considered normal without completion of full stroke in the closed state. Moreover, 20 and 38 misclassifications occurred for normal strokes in open and closed states, respectively, resulting in an average F-score of 0.947. Fig. 8 presents the results of the misclassifications. Three abnormal strokes occurred in clusters of normal strokes in the open state. Interestingly, many misclassifications occurred in this area, implying that the min-max normalization method can only separate non-identical characteristics in this area, rather than identical characteristics. One abnormal stroke was also in a cluster of normal without completion of full stroke, and two abnormal strokes were located close to the misclassified abnormal stroke in the closed state. These three abnormal strokes were far from a cluster of abnormal strokes, suggesting that normalizing features via the min-max normalization method is less effective in the creation of distinct manifolds through the autoencoder.

A comparison of the normalization methods clearly shows that the selection of an appropriate normalization is important for securing fault diagnosis and classification, and the z-score normalization method is more appropriate than the min-max normalization method for the data used in this study.

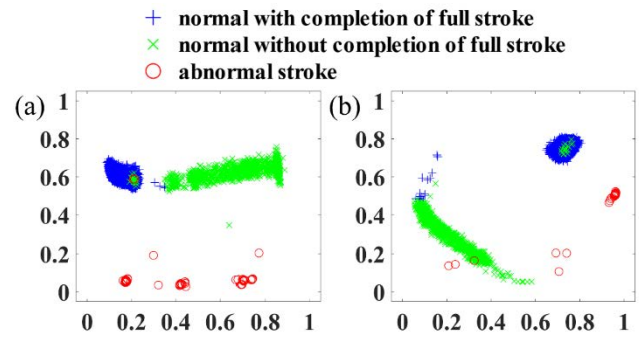


FIGURE 8. Manifold maps with the proposed method at (a) open and (b) closed states using min-max normalization method.

TABLE 4. Classification by using Min-max normalization method; +, x, and o denote normal with completion of full stroke, normal without completion of full stroke, and abnormal stroke, respectively.

Normalization method	State		True			Mean F-score	Average	
			+	x	o			
Min-max	Open	Predict	+	17969	17	2	0.985	0.947
			x	3	817	1		
			o	0	0	49		
	Closed	Predict	+	17960	14	0	0.914	
			x	13	826	3		
			o	0	11	21		

The superiority of the z-score normalization method can be explained by the fact that the feature was normalized with a standard deviation. Compared with other features, this allows other data in this feature to be less affected by outliers when converting physical quantities to the same scale. In contrast, the min-max normalization method is significantly affected by abnormal data, which likely originate from a fault. For example, one abnormal datum, that is, an outlier, is added when the value of two features is already set equal. These abnormal data exceed the range of the current data in one feature but fall within the range of the current data in another feature. In z-score normalization, abnormal data have a minimal effect on the scale of features that are out of range. However, the min-max normalization method has a significant impact on the scale of the out-of-range features. Abnormal data in existing data, which are well-found features by the changed scale feature, may not be well found by the changing feature scale. This results in improved performance of the z-score normalization method in learning autoencoders for abnormal detection.

4) AUTOENCODER

The autoencoder projects data in a high-dimensional space onto data in a low-dimensional space [47], [48]. The non-linear activation function enables a nonlinear hyperplane transformation. Apart from the last layer, the same activation function is typically used for each layer to match the range of the output and input data when training the AE. However, the proposed method addresses varying activation functions of encoders and decoders to impose more nonlinear characteristics in the hyperplane transformation. Three cases were

compared to analyze the contribution of this strategy. The first method is the proposed method. In the second case, the sigmoid function was used for both the encoder and decoder to train the autoencoder, whereas the hyperbolic tangent function was used for both the encoder and decoder in the third case. In all cases, the output layer used a linear activation function. The classification results are listed in Table 5.

In the first case, all abnormal states were detected and the average F-score was 0.996, as mentioned in Section 4.2.1. The average margins of all iterations between the normal with complete stroke and abnormal stroke and between normal without complete stroke and abnormal stroke were 0.29 and 0.10 in open states and 0.18 and 0.09 in closed states, respectively.

In the second case, all abnormal strokes were classified as abnormal strokes in closed states, whereas one abnormal stroke was considered normal without completion of full stroke. Moreover, 29, 78, 6, and 18 misclassifications occurred during normal stroke with completion of full stroke in the open state, normal stroke with completion of full stroke in the closed state, and normal stroke without completion of full stroke in the closed state, respectively, resulting in an average F-score of 0.975. The average margins of all iterations between the normal with complete stroke and the abnormal stroke and between the normal without complete stroke and the abnormal stroke were 0.08 and 0.02 in open states and 0.05 and 0.04 in closed states, respectively. This observation implies that the margin of the second case is lower than that of the first case, suggesting that the distribution of the data according to the state of each data point is closer than that of the first case. The result for the misclassification is clearly identified by the manifold maps presented in Fig. 9(a) and (b). Specifically, some abnormal strokes were located close to clusters of normal strokes, even though the abnormal strokes did not overlap with normal strokes. This resulted in the misclassification of abnormal strokes as normal and normal strokes as abnormal.

In the third case, the classification of all abnormal strokes was similar to that in the second. All abnormal strokes were classified as abnormal strokes in closed states, whereas one abnormal stroke was considered normal without the completion of full stroke. However, 16, 15, and 16 misclassifications occurred at normal stroke without completion of full stroke in the open state, normal with completion of full stroke in the closed state, and normal stroke without completion of full stroke in the closed state, respectively, resulting in an average F-score of 0.993. The misclassification results were clearly identified by the manifold maps presented in Fig. 9(c) and (d). Specifically, normal without the completion of full stroke clusters between clusters of abnormal states and one abnormal state in the open state. This would be closer to normal strokes, resulting in misclassification.

A comparison of the three cases clearly shows that the characteristics of abnormal strokes differ from those of normal strokes when the hyperplane transformation is con-

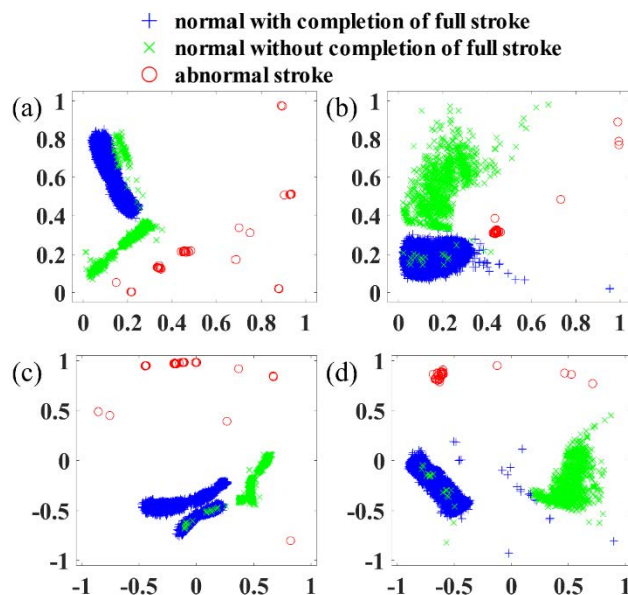


FIGURE 9. Manifold maps with the proposed method at open state with (a) sigmoid and (c) hyperbolic tangent activation function methods. Manifold maps with the proposed method at closed state with (b) sigmoid and (d) hyperbolic tangent activation function methods.

ducted through the autoencoder. However, the consideration of one type of activation function in the autoencoder results in difficulties in space transformation, thereby decreasing the prediction accuracy for fault classification in the transformed manifold. Addressing different nonlinear activation functions on the encoder and decoder strengthens the capability of the nonlinear hyperplane transformation, exhibiting the best accuracy. Specifically, the normal data were trained to not cross the region of the abnormal data when addressing the sigmoid activation function, whereas the margin between the normal and abnormal data was observed to be large when addressing the hyperbolic tangential activation function. The proposed method achieved the advantages of both activation functions, suggesting that addressing different activation functions for the encoder and decoder is effective for the separation of complex yet similar operational strokes in a transformed manifold.

5) SVM

Several methods, including SVMs, ANNs, and decision trees, are widely used for classification [49]. This subsection compares the two other methods to the SVM as a classifier. For a fair comparison of each model, 20% of the data were optimized through BO [40]–[42], implying that the hyperparameters of SVM, decision tree, and ANN would be designated as suitable values for the data used. The hyperparameters of each classifier are presented in Table 6.

In the case of ANNs, all abnormal strokes were classified as abnormal strokes in both open and closed states. Similarly, all normal with completion of full stroke were classified as being in open states, whereas five of these states were identified as normal with completion of full stroke in closed states. Seventeen normal cases without completion of full stroke

TABLE 5. Classification by using different activation function methods; symbols +, ×, and ○ denote normal with completion of full stroke, normal without completion of full stroke, and abnormal stroke, respectively.

Activation function method	State		True			Mean F-score	Average	
			+	×	○			
Sigmoid activation function	Open	Predict	+	17943	77	0	0.971	0.975
			×	29	766	1		
			○	0	1	51		
	Closed	Predict	+	17969	17	0	0.980	
			×	4	831	0		
			○	2	1	24		
Hyperbolic tangent activation function	Open	Predict	+	17972	16	0	0.993	0.993
			×	0	818	1		
			○	0	0	51		
	Closed	Predict	+	17958	16	0	0.994	
			×	15	832	0		
			○	0	0	24		

TABLE 6. Hyperparameter optimization Result by BO by using different classifier methods: SVM, ANN, and Decision tree.

SVM	
Kernel	Linear
Regularization parameter C	100
Class weight	None
Decision function shape	One vs one
Tolerance	1e-3
Decision tree	
Minimal cost-complexity pruning alpha	1e-5
Criterion	Gini
Max. depth	5
Max. features	2
Max. leaf nodes	5
Min. samples leaf	4
Class weight	Balanced
Min. impurity split	0
ANN	
Num. of hidden layers	2
Activation function	Hyperbolic tangent
Num. of hidden nodes	5
Batch size	107
Epochs	995
Learning rate	6.6e-3
Loss function	Sparse categorical crossentropy
Early stopping	20

were classified as normal with completion of full stroke in the open state. Sixteen normal cases without completion of

TABLE 7. Classification by using different classifier methods: ANN and Decision tree; symbols +, ×, and ○ denote normal with completion of full stroke, normal without completion of full stroke, and abnormal stroke, respectively.

Classifier method	State		True			Mean F-score	Average	
			+	×	○			
ANN	Open	Predict	+	17972	17	0	0.996	0.996
			×	0	817	0		
			○	0	0	52		
	Closed	Predict	+	17968	16	0	0.996	
			×	5	832	0		
			○	0	0	24		
Decision tree	Open	Predict	+	17969	17	0	0.996	0.995
			×	3	817	0		
			○	0	0	52		
	Closed	Predict	+	17957	15	0	0.994	
			×	16	833	0		
			○	0	0	24		

full stroke were classified as normal with completion of full stroke, resulting in an average F-score of 0.996. These values were the same as those of the SVM.

In the case of the decision tree, all abnormal strokes were classified as abnormal strokes in both the open and closed states. However, 3 cases of normal with completion of full stroke were classified as normal without completion of full stroke, and 17 cases of normal without completion of full stroke were classified as normal with completion of full stroke in open states. Sixteen cases of normal with completion of full stroke were classified as normal without completion of full stroke, and 15 cases of normal without completion of full stroke were classified as normal with completion of full stroke in closed states, resulting in an average F-score of 0.995.

A comparison of these cases shows that the SVM, ANN, and decision-tree classifiers have similar accuracies. In addition, all abnormal strokes were classified as abnormal strokes in all cases. These results again confirm that the autoencoder effectively transforms the original spaces into manifold maps, where two types of normal and abnormal data can be separated.

However, the calculation times of the SVM, decision tree, and ANN were 0.078, 0.041, and 685 s, respectively, suggesting that a deep neural network of ANN requires 1000 times heavier computational effort than SVM and decision tree. Moreover, the decision tree requires two more hyperparameters to be selected than the SVM, implying that more engineering effort is required to optimize the decision tree. This result also suggests that the decision tree easily converges to a local minimum and thereby has estimation accuracy. For example, misclassifications for normal with and without completion of full stroke are larger in the decision tree compared to those in ANN and SVM, even after conducting BO.

TABLE 8. Additional time-related statistical factors.

Feature name	Equation
Mean	$\frac{\sum x_i}{N}$
RMS	$\sqrt{\frac{\sum x_i^2}{N}}$
Standard deviation	$\sqrt{\frac{\sum (x_i - \bar{x})^2}{N}}$
Peak	$\max(x)$
Skewness	$\frac{\sum (x_i - \bar{x})^3}{(N - 1) \left(\sqrt{\frac{\sum (x_i - \bar{x})^2}{N - 1}} \right)^3}$
Kurtosis	$\frac{\frac{1}{N} \sum (x_i - \bar{x})^4}{\left(\frac{1}{N} \sum (x_i - \bar{x})^2 \right)^2}$
Crest factor	$\frac{ x_{peak} }{\frac{x_{rms}}{x_{peak}}}$
Clearance factor	$\frac{x_{rms}}{\left(\frac{\sum \sqrt{ x_i }}{N} \right)^2}$
Shape factor	$\frac{1}{N} \sum x_i $
Impulse factor	$\frac{x_{peak}}{\frac{1}{N} \sum x_i }$
Peak-to-peak	$\max(x) - \min(x)$
Root sum of square	$\sqrt{\sum x_i^2}$

Therefore, the SVM is the most accurate classifier among the three analyzed.

V. CONCLUSION

In this study, a fault diagnosis and classification method is proposed for elevator doors based on the exclusive use of control state information. The proposed method comprises three phases. The first phase extracts operational state information from the control state information of the elevator doors. This phase is intended to minimize computational efforts for fault detection because elevators are in a nonoperational state in most periods. In the second phase, distinct features are manipulated for fault diagnosis. In the third phase, autoencoders and an SVM are used to effectively classify the three operational states. Actual field measurements indicated that the proposed method has high estimation accuracy for fault diagnosis. A systematic analysis of the field measurements provides the following findings.

- The selection of appropriate statistical methods increases the estimation accuracy of fault diagnosis methods. The use of numerous statistical methods increases the dimensions related to problem solving, creating difficulties in nonlinear hyperplane transformations.
- Feature manipulation, including difference calculation and normalization, has a critical role in fault diagnosis because a high correlation between distinct features and different operational conditions ensures a high estimation accuracy.

- The autoencoder effectively transforms the original space into manifold space for fault diagnosis. Moreover, the implementation of different activation functions in the autoencoder improves the capability of the nonlinear hyperplane transformation.

Future work includes the validation of the proposed method by acquiring data from elevator doors and applying them to other complex systems to demonstrate the effectiveness and robustness of the proposed method.

APPENDIX

See Table 8.

REFERENCES

- [1] N. United, "World urbanization prospects: The 2014 revision: Population division," United Nations, New York, NY, USA, Tech. Rep., 2014.
- [2] G. Zhang, S. Huang, and Y. Yuan, "The study of elevator fault diagnosis based on multi-agent system," in *Proc. Int. Conf. Comput. Intell. Softw. Eng.*, Dec. 2009, pp. 3–7, doi: 10.1109/CISE.2009.5363218.
- [3] S. Yuan, M. Ge, H. Qiu, J. Lee, and Y. Xu, "Intelligent diagnosis in electromechanical operation systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004, vol. 2004, no. 3, pp. 2267–2272, doi: 10.1109/robot.2004.1307399.
- [4] W. L. Chan, A. T. P. So, and S. K. Liu, "A costeffective remote monitoring and communication system in elevator technology," in *Proc. Elevcon*, 1998, pp. 1–6.
- [5] O. Yaman, M. Baygin, and M. Karakose, "A new approach based on image processing for detection of wear of guide-rail surface in elevator systems," *Int. J. Appl. Math. Electron. Comput.*, vol. 4, no. 1, p. 296, 2016, doi: 10.18100/ijamec.270559.
- [6] O. Yaman and M. Karakose, "Auto correlation based elevator rope monitoring and fault detection approach with image processing," in *Proc. Int. Artif. Intell. Data Process. Symp. (IDAP)*, Sep. 2017, pp. 1–5, doi: 10.1109/IDAP.2017.8090176.
- [7] G. Niu, S. S. Lee, B. S. Yang, and S. J. Lee, "Decision fusion system for fault diagnosis of elevator traction machine," *J. Mech. Sci. Technol.*, vol. 22, no. 1, pp. 85–95, 2008, doi: 10.1007/s12206-007-1010-0.
- [8] A. Q. Flores, J. B. Carvalho, and A. J. M. Cardoso, "Mechanical fault detection in an elevator by remote monitoring," in *Proc. 18th Int. Conf. Electr. Mach.*, Sep. 2008, pp. 1–5, doi: 10.1109/ICELMACH.2008.4800064.
- [9] I. O. Olalere and M. Dewa, "Early fault detection of elevators using remote condition monitoring through IoT technology," *South Afr. J. Ind. Eng.*, vol. 29, no. 4, pp. 17–32, Dec. 2018.
- [10] C.-Y. Hsu, Y. Qiao, C. Wang, and S.-T. Chen, "Machine learning modeling for failure detection of elevator doors by three-dimensional video monitoring," *IEEE Access*, vol. 8, pp. 211595–211609, 2020, doi: 10.1109/ACCESS.2020.3037185.
- [11] K. M. Mishra and K. Huhtala, "Elevator fault detection using profile extraction and deep autoencoder feature extraction for acceleration and magnetic signals," *Appl. Sci.*, vol. 9, no. 15, p. 2990, Jul. 2019, doi: 10.3390/app9152990.
- [12] J. Yan and J. Lee, "Degradation assessment and fault modes classification using logistic regression," *ASME J. Manuf. Sci. Eng.*, vol. 127, pp. 912–914, Nov. 2005, doi: 10.1115/1.1962019.
- [13] S. C. Hui, M. K. H. Leung, and F. Wang, "Elevview: Remote intelligent elevator monitoring system," *Int. J. Comput. Appl.*, vol. 26, no. 2, pp. 111–118, 2004, doi: 10.2316/journal.202.2004.2.202-1206.
- [14] I. Skog, I. Karagiannis, A. B. Bergsten, J. Harden, L. Gustafsson, and P. Handel, "A smart sensor node for the Internet-of-Elevators—Non-invasive condition and fault monitoring," *IEEE Sensors J.*, vol. 17, no. 16, pp. 5198–5208, Aug. 2017, doi: 10.1109/JSEN.2017.2719630.
- [15] L.-D. Van, Y.-B. Lin, T.-H. Wu, and Y.-C. Lin, "An intelligent elevator development and management system," *IEEE Syst. J.*, vol. 14, no. 2, pp. 3015–3026, Jun. 2020, doi: 10.1109/JSYST.2019.2919967.
- [16] J. Bao, H. Wu, and Y. Yan, "A fault diagnosis system-PLC design for system reliability improvement," *Int. J. Adv. Manuf. Technol.*, vol. 75, nos. 1–4, pp. 523–534, Oct. 2014, doi: 10.1007/s00170-014-6166-z.
- [17] B. Samanta and K. R. Al-Balushi, "Artificial neural network based fault diagnostics of rolling element bearings using time-domain features," *Mech. Syst. Signal Process.*, vol. 17, no. 2, pp. 317–328, Mar. 2003, doi: 10.1006/mssp.2001.1462.

- [18] M. Ge, R. Du, G. Zhang, and Y. Xu, "Fault diagnosis using support vector machine with an application in sheet metal stamping operations," *Mech. Syst. Signal Process.*, vol. 18, no. 1, pp. 143–159, 2004, doi: [10.1016/S0888-3270\(03\)00071-2](https://doi.org/10.1016/S0888-3270(03)00071-2).
- [19] Y. Qian, X. Li, Y. Jiang, and Y. Wen, "An expert system for real-time fault diagnosis of complex chemical processes," *Expert Syst. Appl.*, vol. 24, no. 4, pp. 425–432, 2003, doi: [10.1016/S0957-4174\(02\)00190-2](https://doi.org/10.1016/S0957-4174(02)00190-2).
- [20] U. E. Akpudo and J.-W. Hur, "A feature fusion-based prognostics approach for rolling element bearings," *J. Mech. Sci. Technol.*, vol. 34, no. 10, pp. 4025–4035, Oct. 2020.
- [21] V. K. Gupta, S. R. K. Nielsen, and P. H. Kirkegaard, "A preliminary prediction of seismic damage-based degradation in RC structures," *Earthquake engineering structural dynamics*, vol. 30, no. 7, pp. 981–993, 2001.
- [22] Z. Liu, N. Fnaiech, L. Saidi, B. Chebel-Morello, and F. Fnaiech, "Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals," *Appl. Acoust.*, vol. 89, no. 3, pp. 16–27, Mar. 2015.
- [23] J.-T. Kim, J.-H. Park, and B.-J. Lee, "Vibration-based damage monitoring in model plate-girder bridges under uncertain temperature conditions," *Eng. Struct.*, vol. 29, no. 7, pp. 1354–1365, Jul. 2007.
- [24] T. Han, Y.-F. Li, and M. Qian, "A hybrid generalization network for intelligent fault diagnosis of rotating machinery under unseen working conditions," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.
- [25] T. Han, C. Liu, L. Wu, S. Sarkar, and D. Jiang, "An adaptive spatiotemporal feature learning approach for fault diagnosis in complex systems," *Mech. Syst. Signal Process.*, vol. 117, pp. 170–187, Feb. 2019.
- [26] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1997.
- [27] V. Vapnik, *The Nature of Statistical Learning Theory*. Switzerland: Springer, 2013.
- [28] T. Glasmachers, Y.-K. Noh, and M.-L. Zhang, "Limits of end-to-end learning," in *Proc. Asian Conf. Mach. Learn.*, 2017, pp. 17–32.
- [29] A. Webb, C. Reynolds, W. Chen, H. Reeve, D. Iliescu, M. Lujan, and G. Brown, "To ensemble or not ensemble: When does end-to-end training fail?" in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2020, pp. 109–123.
- [30] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
- [31] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [32] X. Liu, Z. Zheng, Z. Zhang, and Z. Cao, "A statistical learning framework for the intelligent imputation of offshore wind farm missing scada data," in *Proc. 8th Renew. Power Gener. Conf. (RPG)*, 2019, vol. 2019, no. CP764, pp. 68–71, doi: [10.1049/cp.2019.0615](https://doi.org/10.1049/cp.2019.0615).
- [33] T. R. Fasel, H. Sohn, G. Park, and C. R. Farrar, "Active sensing using impedance-based ARX models and extreme value statistics for damage detection," *Earthquake engineering structural dynamics*, vol. 34, no. 7, pp. 763–785, Jun. 2005, doi: [10.1002/eqe.454](https://doi.org/10.1002/eqe.454).
- [34] Y. S. Chong and Y. H. Tay, "Abnormal event detection in videos using spatiotemporal autoencoder," in *Proc. Int. Symp. Neural Netw.*, 2017, pp. 189–196.
- [35] M. Kallas, C. Francis, L. Kanaan, D. Merheb, P. Honeine, and H. Amoud, "Multi-class SVM classification combined with kernel PCA feature extraction of ECG signals," in *Proc. 19th Int. Conf. Telecommun. (ICT)*, Apr. 2012, pp. 1–5, doi: [10.1109/ICTEL.2012.6221261](https://doi.org/10.1109/ICTEL.2012.6221261).
- [36] A. Mathur and G. M. Foody, "Multiclass and binary SVM classification: Implications for training and classification users," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 2, pp. 241–245, Apr. 2008, doi: [10.1109/LGRS.2008.915597](https://doi.org/10.1109/LGRS.2008.915597).
- [37] R. Laza, R. Pavón, M. Reboiro-Jato, and F. Fdez-Riverola, "Evaluating the effect of unbalanced data in biomedical document classification," *J. Integrative Bioinf.*, vol. 8, no. 3, pp. 105–177, 2011, doi: [10.1515/jib-2011-177](https://doi.org/10.1515/jib-2011-177).
- [38] Y. Sun, M. Robinson, R. Adams, P. Kaye, A. Rust, and N. Davey, "Integrating binding site predictions using non-linear classification methods," in *Proc. Int. Workshop Deterministic Stat. Methods Mach. Learn.*, 2004, pp. 229–241.
- [39] I. Visentini, L. Snidaro, and G. L. Foresti, "Diversity-aware classifier ensemble selection via f-score," *Inf. Fusion*, vol. 28, pp. 24–43, Mar. 2016, doi: [10.1016/j.inffus.2015.07.003](https://doi.org/10.1016/j.inffus.2015.07.003).
- [40] Y. Xia, C. Liu, Y. Li, and N. Liu, "A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring," *Expert Syst. Appl.*, vol. 78, pp. 225–241, Jul. 2017, doi: [10.1016/j.eswa.2017.02.017](https://doi.org/10.1016/j.eswa.2017.02.017).
- [41] M. Feurer, J. T. Springenberg, and F. Hutter, "Using meta-learning to initialize Bayesian optimization of hyperparameters," in *Proc. MetaSel@ ECAI*, 2014, pp. 3–10.
- [42] W. Jia, C. Xiu-Yun, Z. Hao, X. Li-Dong, L. Hang, and D. Si-Hao, "Hyperparameter optimization for machine learning models based on Bayesian optimization," *J. Electron. Sci. Technol.*, vol. 17, no. 1, pp. 26–40, 2019, doi: [10.11989/JEST.1674-862X.80904120](https://doi.org/10.11989/JEST.1674-862X.80904120).
- [43] S. Sen, E. A. Sezer, C. Gokceoglu, and S. Yagiz, "On sampling strategies for small and continuous data with the modeling of genetic programming and adaptive neuro-fuzzy inference system," *J. Intell. Fuzzy Syst.*, vol. 23, no. 6, pp. 297–304, 2012, doi: [10.3233/IFS-2012-0521](https://doi.org/10.3233/IFS-2012-0521).
- [44] M. Tripathi, "K-fold cross-validation machine learning approach on data imbalance for wireless sensor network," *Int. J. Sci. Res. Eng. Trends*, vol. 5, no. 5, pp. 1590–1595, 2019.
- [45] J. Sola and J. Sevilla, "Importance of input data normalization for the application of neural networks to complex industrial problems," *IEEE Trans. Nucl. Sci.*, vol. 44, no. 3, pp. 1464–1468, Jun. 1997, doi: [10.1109/23.589532](https://doi.org/10.1109/23.589532).
- [46] S. Nayak, B. B. Misra, and H. S. Behera, "Impact of data normalization on stock index forecasting," *Int. J. Comput. Inf. Syst. Ind. Manage. Appl.*, vol. 6, no. 2014, pp. 257–269, 2014.
- [47] H. Babaoumail, R. Hou, G. T. Gnitou, and B. Ayugi, "Novel statistical downscaling emulator for precipitation projections using deep convolutional autoencoder over Northern Africa," *J. Atmos. Sol.-Terr. Phys.*, vol. 218, Jul. 2021, Art. no. 105614, doi: [10.1016/j.jastp.2021.105614](https://doi.org/10.1016/j.jastp.2021.105614).
- [48] R. Savitha, S. Suresh, and N. Sundararajan, "Projection-based fast learning fully complex-valued relaxation neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 4, pp. 529–541, Apr. 2013, doi: [10.1109/TNNLS.2012.2235460](https://doi.org/10.1109/TNNLS.2012.2235460).
- [49] D. Bertsimas and J. Dunn, "Optimal classification trees," *Mach. Learn.*, vol. 106, no. 7, pp. 1039–1082, Jul. 2017, doi: [10.1007/s10994-017-5633-9](https://doi.org/10.1007/s10994-017-5633-9).

HYUN CHAE received the B.S. degree in mechanical engineering from Chung-Ang University, Seoul, South Korea, in 2020, where he is currently pursuing the M.S. degree.

His research interests include signal preprocessing, deep learning, machine learning, and its applications in prognostics and health management.



JAE EUNG LEE received the B.S. degree in mechanical engineering from Chung-Ang University, Seoul, South Korea, in 1981, and the M.S.E. and Ph.D. degrees in mechanical engineering from the University of Michigan, Ann Arbor, MI, USA, in 1986 and 1991, respectively.

He joined Chung-Ang University, in 1993, where he is currently a Professor with the School of Mechanical Engineering. His research interests include dynamics, vibration, and signal analyses in mechanical systems.



KI-YONG OH received the B.S. degree in mechanical engineering from Hanyang University, Seoul, South Korea, in 2005, the M.S. degree in mechanical engineering from the KAIST, in 2006, and the Ph.D. degree in mechanical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2016.



He joined the School of Mechanical Engineering, Hanyang University, in 2019, where he is currently employed as an Assistant Professor. His teaching and research interests include applied dynamics, and AI and mobility toward prognostics and health management in the field of complex energy systems.

• • •