Model-based Fault Detection and Diagnosis for HVAC Systems Using Convolutional Neural Network

Shohei Miyata¹, Yasunori Akashi¹, Jongyeon Lim¹, Yasuhiro Kuwahara², Katsuhiko Tanaka³
¹The University of Tokyo, Tokyo, Japan
²MTD CO. Ltd., Tokyo, Japan
³Tokyo Electric Power Company Holdings, Inc., Tokyo, Japan

Abstract

In building heating, ventilation, and air conditioning (HVAC) systems, fault detection and diagnosis (FDD) is crucial for achieving high energy efficiency. In this study, a novel method for FDD is proposed, which includes fault database generation by detailed simulation, convolutional neural network (CNN) training using a database, and FDD of real data using the trained CNN. The CNN is a classifier with sufficiently high accuracy for diagnosing the subtle fault features emerging in the fault behavior data of HVAC systems. It was confirmed that FDD of real data was possible by the trained CNN, in addition to learning the generated database with high accuracy. Thus, this methodology can assist in analyzing real data because it is possible to locate the fault and assume its relative severity approximately.

Introduction

Faults that occur in heating, ventilation, and air conditioning (HVAC) systems degrade the energy efficiency and indoor environment. Some faults, such as equipment malfunction, affect the system supplying the heat load, whereas other faults deteriorate the system performance, although the indoor environment is properly regulated. These are the target faults that are detected and diagnosed in this study. Faults generally cause 5–30% degradation in commercial buildings (Katipamula and Brambley, 2005; Roth et al., 2005; Fernandez et al., 2017). Therefore, it is crucial to detect, diagnose, and eliminate them. Fault detection and diagnosis (FDD) is defined as the detection of the presence of faults and their location (Hyvärinen et al., 1996).

An optimally controlled system does not have faults such as inappropriate controls. Thus, the control optimization is included in the concept of FDD and fault elimination. However, to avoid complicated discussions, we define it as the standard condition calculated by the simulation, with reference to the design specifications.

The ideal FDD scheme is depicted in Figure 1. The existence of faults is first detected, followed by the diagnosis of the type of fault and location, after which the severities of the diagnosed faults are determined. The severity of the fault determines the fault that is to be eliminated first; hence, identification is included in this scheme. As we had previously addressed fault detection (Miyata et al., 2016), the objective of this study is diagnosis.

FDD methods are mainly classified into three classes: anomaly detection, rule-based method, and model-based method. For anomaly detection, Yoshida and Kumar (1999) proposed methods to determine abrupt faults using an autoregressive exogenous model and extended Kalman filter. However, anomaly detection cannot detect faults other than the abrupt ones. For the rule-based method, Schein and Bushby (2006) presented a flow chart of the hierarchical FDD algorithm and Veronica (2013) computerized an expert system logic for detecting control faults. However, the constructed rules cannot be applied to all systems, and rule-construction for a target system is expensive. For the model-based method, Wang et al. (2009) proposed a method for detecting sensor faults using a regression model and principal component analysis. Frank et al. (2016) modelled a VAV system and classified the fault data obtained by simulation, using a machine learning method such as the support vector machine. However, its accuracy was insufficient, and it was not applied to real data. The model-based method has difficulty in generating data with sufficient quality and quantity, and in processing the data appropriately.

Based on the development of machine learning and system simulation, we decided to adopt the model-based method. We considered the FDD as a classification problem in the machine learning field and diagnosed the fault features emerging in the system behavioral data using a convolutional neural network (CNN). The CNN requires training data, which includes the fault behavior with the fault label. Real data cannot be used as training data because the training data should be labelled appropriately. Hence, to generate training data for the CNN, we performed dynamic simulation for generating complicated system behaviors with faults. In addition, we analyzed the influence of the fault severity on the diagnosis results.

![Figure 1: FDD scheme](image-url)
Methods

Target system

We targeted a real system in an office building whose area is approximately 160,000 m² in Tokyo, Japan. The target system was water-side of the HVAC system, i.e., a heat-source system (Figure 2) comprising four chillers and thermal water storage tanks. The system utilizes sewage from a treatment plant adjacent to the building, instead of cooling towers, and charges heat from 22:00 to 08:00 and discharges from 8:00 to 22:00.

Heat source system simulation

To generate the training data for the CNN, we constructed a dynamic model of the system (Figure 3) following the first principles model for generating complicated system behaviors with faults (Bourdouxhe et al., 1998). The input values included the heat load, sewage temperature, and a few set values. The pumps and valves were controlled every minute, based on the input and control logic, and system behaviors, such as the water flow and temperature, were also calculated every minute.

In addition, control logic, including the heat charge and discharge, and proportional-integral (PI) controls were incorporated in the simulation. The pumps and valves were regulated by PI controls, according to set values. The threshold and waiting time for the controllers to change the number of operating pumps were also incorporated.

The flow rate was calculated considering the pressure and flow balance in the pipe network. The calculation is based on Kirchhoff’s laws: one is that the sum of the water quantity flowing toward any node and away from it is zero, while the other is that the sum of the pressure in any loop is zero.

Figure 2: Target system.

Figure 3: Flow of the system simulation.
The total pump head and flow rate were determined from the specification curve, which was rescaled based on the inverter frequency (Figure 4). The pressure loss in the pipes was calculated using the Darcy–Weisbach equation (1), while that at the valves was calculated with the degree of opening and flow rate in an equal proportion.

\[ \Delta P = \frac{\lambda}{D^2} \frac{L}{2} \nu^2 \]  
(1)

where \( \Delta P \) is the pressure loss [Pa], \( \lambda \) is the flow coefficient [-], \( L \) is the pipe length [m], \( D \) is the hydraulic diameter [m], \( \rho \) is the density of the fluid [kg/m³], and \( \nu \) is the flow velocity [m/s].

The temperature in the tanks and heat exchanger were calculated theoretically and the outlet temperature of the heat exchanger was calculated using equations (2)–(4), as follows:

\[ \dot{Q} = K \cdot A \cdot (\text{MTD}) \]  
(2)

\[ \dot{Q} = \dot{G}_h \cdot c_h \cdot (T_{h,in} - T_{h,out}) \]  
(3)

\[ \dot{Q} = \dot{G}_c \cdot c_c \cdot (T_{c,out} - T_{c,in}) \]  
(4)

where \( \dot{Q} \) is the exchanged heat [W], \( K \) is the heat transfer coefficient, \( A \) is the heat exchange area [m²], \( \text{MTD} \) is the logarithmic mean temperature difference [°C], \( \dot{G} \) is the flow [kg/s], and \( c \) is the specific heat at constant pressure. Subscripts \( h \) and \( c \) refer to the hotter and the colder side, respectively, and \( in \) and \( out \) refer to the inlet and outlet, respectively.

The performances of the chillers were calculated using the specification curve (Figure 5) that expresses the relationship among the partial load, condenser-water-outlet temperature, and coefficient of performance (COP) (generated heat/power consumption); however, the outlet temperature was calculated using the COP and the inlet temperature. Therefore, the curve for the chillers requires convergence calculation.

Finally, 120 items, including the flow rate, temperature, and power, were output every minute. The monitored data and simulation results of the chilled water temperature and flow rate in a representative week were compared (Figure 6). It was confirmed that the behaviors were similar, and that the simulation results exhibited the same phenomenon as in the real system. Because both data corresponded to 1-min intervals, their values oscillated sharply, when the operation number of pumps was controlled.

### Table 1: Faults generated by the system simulation

<table>
<thead>
<tr>
<th>Label</th>
<th>Fault type</th>
<th>Fault detail and severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>F0</td>
<td>No faults</td>
<td>First principles model without faults</td>
</tr>
<tr>
<td>F1</td>
<td>Performance of chillers</td>
<td>Due to condenser fouling, the pressure loss of condenser increases by 50% and efficiency deteriorates by 10%</td>
</tr>
<tr>
<td>F2</td>
<td>Sewage pump set value</td>
<td>The approach temperature is set to 1.0 °C from 2.0 °C</td>
</tr>
<tr>
<td>F3</td>
<td>Heat exchanger efficiency (CHEX)</td>
<td>Heat exchange area becomes half</td>
</tr>
<tr>
<td>F4</td>
<td>Temperature sensor (outlet of CHEX)</td>
<td>The sensor measures the value lower than the true value by 1.0 °C</td>
</tr>
</tbody>
</table>

Fault database generation

A condition without faults and conditions with four types of faults were calculated (Table 1). In addition to these five conditions, the combinations of these faults were calculated, and the fault database for 16 conditions was generated. The simulation period was from July to September in 2014, which included 122 days. Therefore,
faulty behaviors for 1,952 days were generated. The input data of the simulation was assembled from the real data for the corresponding period. The fault database was utilized to train the CNN such that the CNN can diagnose real data from July to September in 2015, which is a year after the learning-data period.

For analyzing the impact of the faults on energy efficiency, system coefficient of performance (COPsys) (supplied heat/power consumption) was calculated. F1, F2, and F3 decreased the COPsys by 9.18%, 3.22%, and 2.88%, respectively (Table 2), whereas F4 improved it by 2.48%. The supplied heat was almost the same from F0–F4, each fault was under the control of supplying heat load.

With respect to the system behavior, the power consumed by chiller, TR2, the sewage pumps (SPs), and chilled water pump, CHP5, increased because of F1, F2, and F3, respectively (Figure 7). The SPs are controlled as per the outlet condenser water temperature of the heat exchanger (SHEX); therefore, its set value affects the pump power and condenser water temperature, which is a chiller performance parameter. The chilled water supply pump, SHP5, is controlled as per the outlet chilled water temperature of the heat exchanger (CHEX); therefore, the reduction in the heat exchange efficiency of the CHEX resulted in greater flow and power at CHP5. F4 can be regarded as a fault because the chilled-water-supply-temperature increased, even though the COPsys improved.

Fault detection and diagnosis by CNN

To utilize the generated fault database as the learning data for the CNN, it is necessary to preprocess the database appropriately. In this study, we converted 24-h data into an image because a day is the minimum cycle for the HVAC system behavior, and decision making for fixing the fault would be implemented in a period longer than a day. Real data was collected for 15 min, and the database was converted from 1-min data to 15-min data by averaging the values.

Each value was then standardized from 0 to 1 and converted into images (Figure 8). As the column-direction represents the time axis and the 15-min data for a day is the data for the image, the number of values in column-direction was 96. The number of rows was set to 120, which is the number of items output by the simulation. In this study, by applying imaged data to the CNN, it was possible to learn the fault features in the column direction for the time series feature, and in the row direction for the relationship between items. Because all the images appear similar, it was assumed that it would be difficult for a conventional neural network or other machine learning methods to extract the features. Therefore, a CNN, which recognizes images using a network modelled with reference to the visual cortex of the brain (Fukushima, 1982), was adopted in this study.

The CNN was structured with reference to Simonyan (2014) (Figure 9). It has six layers including two convolution layers, two max pooling layers and two full connection layers. The number of filters was set to 36. The parameters for convolution, max pooling and number of filters were decided based on try and error.

### Table 2: Impact of faults on energy performance.

<table>
<thead>
<tr>
<th></th>
<th>Supplied heat [GJ]</th>
<th>Power consumption [MWh]</th>
<th>COPsys</th>
<th>COPsys ratio [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>F0</td>
<td>24338</td>
<td>1394</td>
<td>4.85</td>
<td>-</td>
</tr>
<tr>
<td>F1</td>
<td>24338</td>
<td>1535</td>
<td>4.40</td>
<td>-9.18</td>
</tr>
<tr>
<td>F2</td>
<td>24338</td>
<td>1441</td>
<td>4.69</td>
<td>-3.22</td>
</tr>
<tr>
<td>F3</td>
<td>24338</td>
<td>1436</td>
<td>4.71</td>
<td>-2.88</td>
</tr>
<tr>
<td>F4</td>
<td>24338</td>
<td>1360</td>
<td>4.97</td>
<td>2.48</td>
</tr>
</tbody>
</table>

![Figure 7: System behavior with faults.](image)

![Figure 8: Imaged data.](image)

The CNN that diagnose real data outputs the probability of each label and it expresses which types of fault behavior the real data is close to. The meaning of the value of the probability is discussed later.
We divided the database into training and validation data. Each fault included 122 data, among which 20 were assigned randomly as validation data and the remaining as training data. To moderate the bias of this data assignment, the assignment was implemented 30 times and 30 CNNs were trained. In the test phase, the trained 30 CNNs diagnosed the real data from 2015, during the period corresponding to the learning data. The program was coded using TensorFlow (Abadi et al., 2016).

Results
Training accuracy of CNN
The average validation accuracy of 30 CNNs was 98.4% (Figure 10). Hence, it can be regarded that the CNN learnt the fault features sufficiently.

Diagnosis characteristics of the trained CNN
When the trained CNN diagnoses real data, it outputs certain probability values. Considering the feature extraction process in the CNN, the probability indicates the degree of the fault pattern recognized in the real data. Hence, the probability was called diagnosis probability and its characteristics were analyzed.

![Figure 9: Convolutional Neural Network.](image)

![Figure 10: Learning curve.](image)

![Figure 11: Impact of faults with different severities on energy performance.](image)

![Figure 12: Impact of faults with different severity on diagnosis probability.](image)
To observe the diagnosis characteristics of the trained CNN, fault behaviors simulated with different severities were generated and diagnosed by the trained CNN. The severity was defined as the amount by which the fault differed from the condition without faults. A severity of 0% indicates no faults and 100% means the same as that described in Table 1. The severity was expressed as 20–180% in steps of 20%. For example, 20% severity of F4 is that the sensor measures the value lower than the true value by 0.2 °C. The input values for this simulation were obtained from the real data in 2015.

The impact of faults with different severities on the COPs system was nonlinear, although the severities differed linearly (Figure 11). As the HVAC system is highly complicated, the behavior was not linearly affected by the fault severity. This can influence the diagnosis probability, when the fault severity in the real system is different from the assumptions in the simulation for the learning data. System behaviors generated with different severities were then diagnosed by the trained CNN (Figure 12); the displayed diagnosis probability indicates the average diagnosis probability during the target period.

For a single fault, the diagnosis probability decreased as the severity reduced, for severities less than 100% (a-1,2,4). For F3, the diagnosis probability of F1 emerged with an increase in the severity of F3 (a-3) because when F3 is more severe, the performance of the chillers, whose feature is similar to F1, decreases.

To investigate the diagnostic characteristics of multiple faults, F1 and F2 were generated simultaneously, and the severity of F1 was manipulated (b-1). The diagnosis probabilities of F1 and F2 were almost the same at 50%, when the severity of F1 was more than 100% and vice versa (b-2). Similarly, the other combinations were investigated, and the same tendency was observed, except for the combination of F3 and F4.

When F3 and F4 were generated simultaneously and the severity of F3 was manipulated, the more severe F3 was, the less was the diagnosis probability of F4 and vice versa (b-3,4). F3 and F4 both occur around the CHEX with opposing influences; therefore, the features of F3 and F4 can cancel out each other.

From the above, the diagnosis probability indicates the existence of the fault feature and the value reflects the severity of the fault. However, if multiple faults occur simultaneously, the values approximately represent the relative severities of the faults. It is difficult to utilize the value of the diagnosis probability directly for the assumption of the fault severity. Therefore, an identification process is required, as described in Figure 1.

**Diagnosis of real data by the trained CNN**

Real data were diagnosed by the trained CNN (Figure 13, Table 3). Figure 13 and Table 3 shows the diagnosis probabilities, where values of multiple faults were proportionally divided. The result indicates that the real system presented F3 primarily. F1 and F4 were diagnosed with lower probabilities throughout the period, and F2 was diagnosed at a specific time.

### Table 3: Average diagnosis probabilities for F0-F4.

<table>
<thead>
<tr>
<th>Average diagnosis probability [%]</th>
<th>F0</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
</tr>
</thead>
<tbody>
<tr>
<td>for F0-F4</td>
<td>22.4</td>
<td>14.6</td>
<td>9.3</td>
<td>40.3</td>
<td>13.4</td>
</tr>
</tbody>
</table>

**Figure 13:** Diagnosis of the real data.

**Figure 14:** Diagnosis probability and fault in real data.
Analysis of the real data

The diagnosed fault probabilities differed each day. Further, the fault severities of the real data, the assumed fault severities in the simulation (Table 1), and the diagnosis results were compared (Figure 14). For F3, as the real data and the assumed severity were very close, the diagnosis probability was high. For F1, the closer the real data was to the assumed severity, the higher was the diagnosis probability. For F2, when the real data was higher than the no-fault value, the diagnosis probability was almost zero. For F4, the real data was considerably lower than the assumed severity, and the diagnosis probability was also low. The sensor error cannot be observed because its true value never be observed. For F4, a temperature sensor error in the real data was assumed based on the temperature at the outlet of the CHEX and at the chilled water header (see Figure 2).

Discussion

Considering the 98.4% training accuracy, the CNN performed sufficiently in extracting the features of the faulty behaviors of the heat source systems. The results of the real data diagnosis by the CNN corresponded to the real data analysis; therefore, the proposed FDD method is effective for real data. It should be noted that the output of the CNN was called the diagnosis probability, which indicates the existence of a fault feature and the approximate relative severity of the fault. Therefore, to determine the accurate severity of the fault diagnosed by the proposed method, an identification process is required. In Figure 13, F0, which is the case without faults, was diagnosed second. This may be because of the presence of other faults, which were not available in the fault database. In addition, F3 and F4 can cancel out the fault effects. Therefore, the CNN may have misdiagnosed the status that includes F3 and F4 simultaneously as F0. From the above, expanding the fault types and diagnosing multiple faults considering fault characteristics, such as cancelling out, need to be undertaken in future.

Conclusion

A general FDD methodology for building-heat-source systems had not been established because these systems include various equipment and complex controls, depending on the building. Hence, this study proposed an FDD method involving the generation of a fault database by detailed simulation, CNN training using the generated database, and FDD of the real data by the trained CNN. It was demonstrated that FDD of real data was possible by the trained CNN, in addition to learning the generated database with high accuracy. This methodology can assist in analyzing real data because it indicates the faults emerging in real data with the probability. For example, if the result indicates that the system has a fault in the heat exchanger, the fault can be confirmed by checking the data related to the heat exchanger, instead of checking all the data, based on expert knowledge.

However, as discussed earlier, there is possibility for the CNN to misdiagnose faults and hence, expansion of fault types and diagnosis of multiple faults considering fault characteristics, such as cancelling out, is essential. Considering the cost for generating the fault database and CNN learning, appropriate generation of the database is crucial. It is intended to develop a framework for generating the fault database in future.

Acknowledgements

This work was supported by Grant-in-Aid for JSPS Research Fellow, Grant number JP17J08140 and JSPS KAKENHI, Grant Number JP 18K13879.

References


