Research Article

TRR
JOURNAL OF THE TRANSPORTATION RESEARCH BOARD

Transportation Research Record
2022, Vol. 2676(4) 710–718
© National Academy of Sciences: Transportation Research Board 2022
Article reuse guidelines: sagepub.com/journals-permissions
DOI: 10.1177/03611981211064893
journals.sagepub.com/home/trr
SAGE

# Deep-Learning-Based Fault Occurrence Prediction of Public Trains in South Korea

Angela Caliwag[1], Seok-Youn Han[2], Kee-Jun Park[2], and Wansu Lim[1]

**Abstract**

The reliability and safety of the train system is a critical issue, as it transports many passengers in its daily operation. Most studies focus on fault diagnosis methods to determine the cause of faults in the train system. Aside from fault diagnosis, it is also vital to perceive a fault even before it occurs. In this study, a fault occurrence prediction based on a machine learning model is developed. The fault occurrence prediction method aims to predict the remaining useful life (RUL) of a train subsystem. RUL refers to the remaining amount of time before a fault occurs on a train subsystem. The prediction method developed in this study can be used to clear a fault even before it occurs. In case of inevitable faults, the output from the prediction method can be used to alert the personnel in charge by imposing an alarm. Therefore, the fault occurrence prediction method is expected to increase the reliability of the train system. The deep neural-network-based model is tested on an actual device. Deep neural network is used because of its feature extraction capability, especially in handling big amount of data. The testing results in 90.08% accuracy. In addition, a graphical user interface is developed as an interface between a user and the actual device containing the fault occurrence prediction model.

**Keywords**

data and data science, artificial intelligence and advanced computing applications, artificial intelligence, data analytics, deep learning, machine learning (artificial intelligence), neural networks, supervised learning, rail, rail safety, train

The train system has been a widely used transport system in South Korea. As of February 2020, an analysis performed using a daily use average transportation big data in Seoul, South Korea, revealed that approximately 64% of the volume of traffic is accounted to the public train or subway systems. As of 2019, the average use of public train or subway systems is approximately 605,930 passengers per day (*1*). This amount of daily average use of public train or subway systems has become the motivation for several researchers to exert effort in ensuring the reliability and safety of the passengers. The efforts of the researchers led to the development of the train control and management system (TCMS). TCMS is responsible for: (1) real-time monitoring of train systems; (2) display of onboard equipment fault information; (3) transmission of powering and braking commands; and (4) transmission of commands for passenger information, air conditioning, and other service equipment (*2*, *3*). Among the functions of TCMS, the display of fault information is the most necessary in ensuring the reliability and safety of passengers. Fault information often includes only: (1) type of fault, (2) status of fault; and (3) location of the fault.

To provide reliable fault information, several researches related to train system fault diagnosis were performed. Fault diagnosis is defined as the detection and classification of the fault. The proposed fault diagnosis methods are categorized into three: (1) model-based methods, (2) data-driven methods; and (3) combination of model- and data-driven-based methods. Among the categories of fault diagnosis methods, the

[1]Department of Aeronautics, Mechanical, and Electronic Convergence Engineering, Kumoh National Institute of Technology, Gumi, South Korea
[2]Korea Railroad Research Institute, Uiwang, South Korea

**Corresponding Author:**
Wansu Lim, wansu.lim@kumoh.ac.kr

data-driven methods category is the most prominent (*4*). In Dandan et al., the authors proposed a data-driven fault diagnosis method of the wheelset bearing of high-speed trains (*5*). In the proposed method, the raw vibration signals measured from the wheelset bearing is used to train a convolutional neural network (CNN) to learn the features and classify the health condition of the bearing. In Hexuan et al., the authors proposed a data-driven fault diagnosis method of the bogies of a high-speed train (*6*). In the proposed method, the big data extracted from the bogie of the high-speed train is used to train a deep neural network (DNN) to classify the state of the high-speed train: (1) seven fault conditions, and (2) one normal condition. In Hongtian et al., the authors proposed a data-driven incipient fault diagnosis method of electrical drives of high-speed trains (*7*). Incipient faults are the faults that develop slowly and only show slight fault symptoms. For this reason, efforts must be exerted to detect and diagnose the incipient faults in the electrical drives. In the proposed method, the non-linear principal component analysis (PCA) framework is used to detect incipient faults. PCA is capable of extracting the unique features necessary to detect and discriminate incipient faults. Since the incipient faults only show slight symptoms, PCA is used to extract the slight features from a dataset with high dimensionality. After detection, a support vector machine (SVM) algorithm is used to diagnose and classify the type of incipient fault.

Most of the research available in the literature only focuses on fault diagnosis of a specific component, equipment, or subsystem. The main objective of most researches is to identify the fault type, status, and location on a certain subsystem of a train. However, such fault information is inadequate in ensuring the reliability and safety of passengers. For a complex system that transports many passengers in a day, it is crucial to perceive a fault before it occurs. Yueyue et al. highlighted that a prediction method is necessary for avoiding asset restoration caused by faults in the system (*8*). Therefore, Amin et al. propose a fault prognosis method for an electrical transformer to ensure high reliability and prevent catastrophic failures of valuable machines (*9*).

In this paper, a fault occurrence prediction on a public train using the big data gathered by TCMS is proposed. To the best of the authors' knowledge, no study has been conducted on fault occurrence prediction on the overall train system. To enable fault occurrence prediction on the overall train system, data gathered by TCMS is utilized. Despite the advantage of enabling overall train system fault occurrence prediction, the data gathered by TCMS has high dimensionality. To utilize data with high dimensionality, in-depth knowledge about the data is often necessary. To cope with this disadvantage, a machine learning model is used to automatically learn the features and predict the time occurrence of a fault. Specifically, a deep learning model is used. A deep learning model is a subset of machine learning capable of training itself to perform a certain task.

In summary, the main objective of this paper is to develop a deep-learning-based model for fault occurrence prediction on the overall train system. To the best of the authors' knowledge, no study involving fault occurrence prediction of a train system is available in the literature. The contributions of this paper are listed as follows: (1) Actual data gathered during train operation is used to capture the actual behavior of a certain component, equipment, or subsystem that contributes to the occurrence of a fault in real-time; (2) The data gathered by TCMS is used to take into account all faults in the overall train system. This is unlike other studies which focus only on a specific subsystem; (3) The fault prediction model is uploaded onto an actual device to test its applicability on actual on-board TCMS.

## Methods

TCMS is a microcomputer system that performs centralized on-board information management using state-of-the-art semiconductors, software, and data communication technologies (*10*). It consists of three main parts: main computer or train computer (TC), car computer (CC), and display unit (DU). The TC is installed in each driver's room (both tails of a train). It consists of two computers: LIU1 and LIU2. LIU1 and LIU2 have two main functions: (1) failure check, and (2) backup. Since the train is a complex system, TCMS also needs to perform a self-check. This is done by comparing LIU1 and LIU2. In addition, TCMS needs a backup mechanism. LIU1 is often used as the main computer and LIU2 is reserved as a backup computer. The TC is connected to various switches and equipment and provides access control access to the driver. The TC enables the driver to access and control various switches and equipment of the train. Another part of TCMS is the CC. The CC is installed on each car of the train. It is responsible for the monitoring and control of equipment (such as doors, air conditioning units, etc.) on the car to which it is connected. In case the CC of one car fails, the CC of the adjacent car serves as a backup and is capable of controlling the door of the car with a broken CC. Another part of TCMS is the DU. The DU is installed only with the TC in each driver's room (both tails of a train). A diagram of a train with TCMS is shown in Figure 1.

### *TCMS Fault Information*

TCMS enables monitoring and control of all train subsystems. In addition to that, TCMS also stores fault

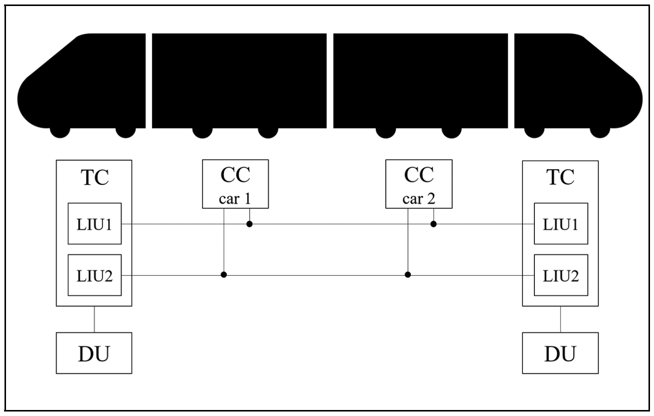


**Figure 1.** Diagram of train system with train control and management system (TCMS).
*Note*: CC = car computer; DU = display unit; TC = train computer.

information that occurred during its operation. Each subsystem of a train is susceptible to a fault (*11*). Using the actual data obtained from TCMS, the most frequent severe faults occurring in a public train in South Korea are summarized as follows: (1) computer communication failure, (2) power cut-off, (3) brake failure, (4) automatic train control (ATC) failure, (5) reversal control failure, and (6) door operation failure. Computer communication failure is further classified into two: CC communication failure and TC-CC communication failure. CC communication failure occurs when the CC loses communication with equipment (such as doors, air conditioning units, etc.) on the car to which it is connected. TC-CC communication failure occurs when the TC loses communication with a CC, and vice versa. Power cut-off occurs when the train or any of its subsystems is suddenly disconnected from its power source. Brake failure is further classified into two: (1) failure of a brake during intended operation, or (2) sudden unintended brake operation. ATC failure is further classified into two: (1) hardware failure, and (2) software failure. Hardware failure occurs when one or more equipment or components fail to operate after the ATC mode is activated. On the other hand, software failure occurs when one or more signals on which the ATC mode depends loses or corrupts. Reversal control failure occurs when the train fails to changes its direction from forward to reverse driving. Door operation failure is further divided into four: (1) failure to close the door, (2) door not completely closed, (3) failure to open the door, and (4) door not completely opened. Failure to close the door means a command to close the door is initiated, but the door remains open. A door not completely closed occurs when a command to close the door is initiated and the door executes the command by closing, but stops somewhere before reaching a completely closed state. Failure to open the door means a command to open the door is initiated, but the door remains closed. A door not completely opened occurs when a command to open the door is initiated and the door executes the command by opening, but stops somewhere before reaching a completely opened state.

## *TCMS Fault Information*

A train consists of several subsystems in which each subsystem is prone to fault. To increase the reliability and safety of passengers, a fault occurrence prediction method for TCMS application is proposed in this study. The overall diagram of the fault occurrence prediction method is shown in Figure 2.

The fault occurrence prediction method developed in this study consists of three phases: (1) data processing, (2) model training, and (3) model testing. In data processing, the input and output data required to train a machine learning model to predict the fault occurrence is obtained from the raw data. The raw data or the data directly obtained from TCMS is in a hexadecimal format. For easier analysis and model training, it is necessary to convert the raw files to a comma-separated values (CSV) file with binary and decimal values. Binary values often represent the state of a component or equipment, while decimal values often represent the measured signals. There are two types of raw files: (1) operation file, (OPER file), and (2) alarm file (ARM file). The OPER file contains all the information about the train and all measured signals from each train subsystem and components. This includes train number, driving mode, door state, train speed, distance traveled, and so forth. A sample plot of the measured signals is shown in Figure 3. The ARM file contains all the information concerning the fault. This includes fault type, fault code, and fault occurrence. The CSV file is obtained by merging and matching the OPER file and ARM file. This is necessary to see the behavior of the signals: (1) before a fault occurs, (2) during a fault occurrence, and (3) after a fault occurrence. After obtaining the CSV file from the raw files, data filtering was performed to remove the anomalies that could affect the training of a machine learning model. The size of data before and after data filtering are listed in Table 1.

The CSV file obtained from the raw data is differentiated from the filtered CSV file by labeling it as CSV file 1 and CSV file 2, respectively. After filtering the data, the time series of the remaining useful life (RUL) is obtained. Here, RUL is defined as the remaining time before a fault occurs. An example of the plot of RUL is shown in Figure 4.

In the figure, the $x$-axis represents the time axis, while the $y$-axis represents the RUL. An RUL value of 0 indicates a fault occurrence. From the figure, the following observations can be made:

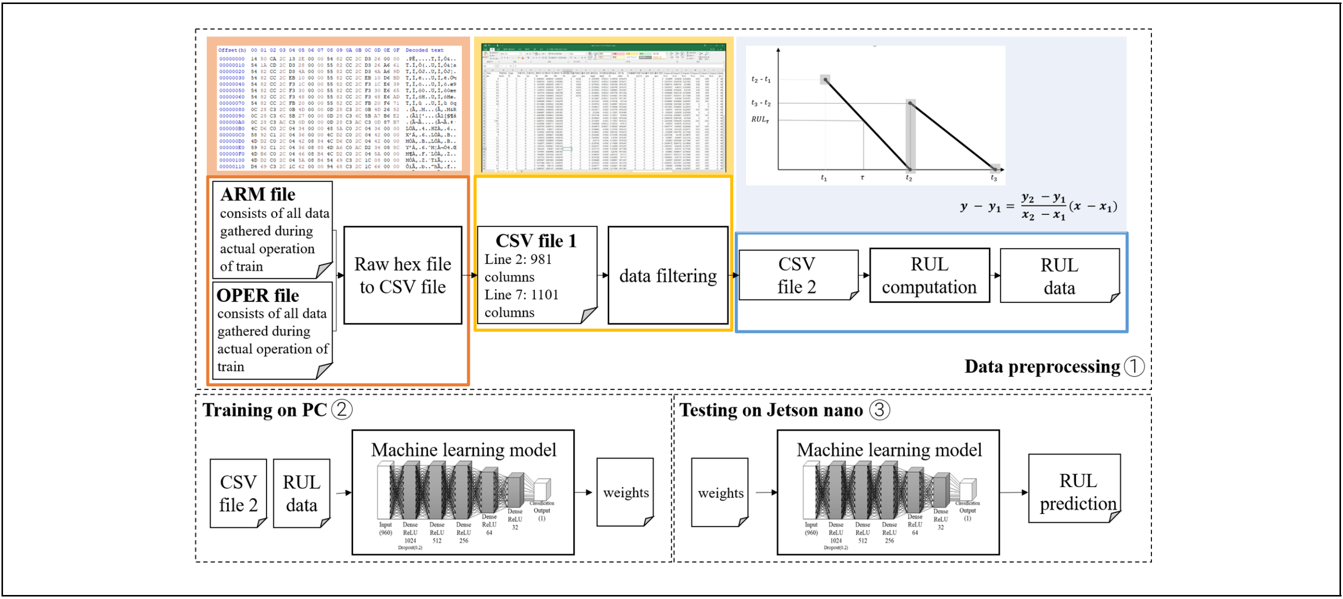


**Figure 2.** Overall fault occurrence prediction method.
*Note*: ARM file = alarm file; CSV file = comma-separated values file; OPER file = operation file; RUL = remaining useful life.

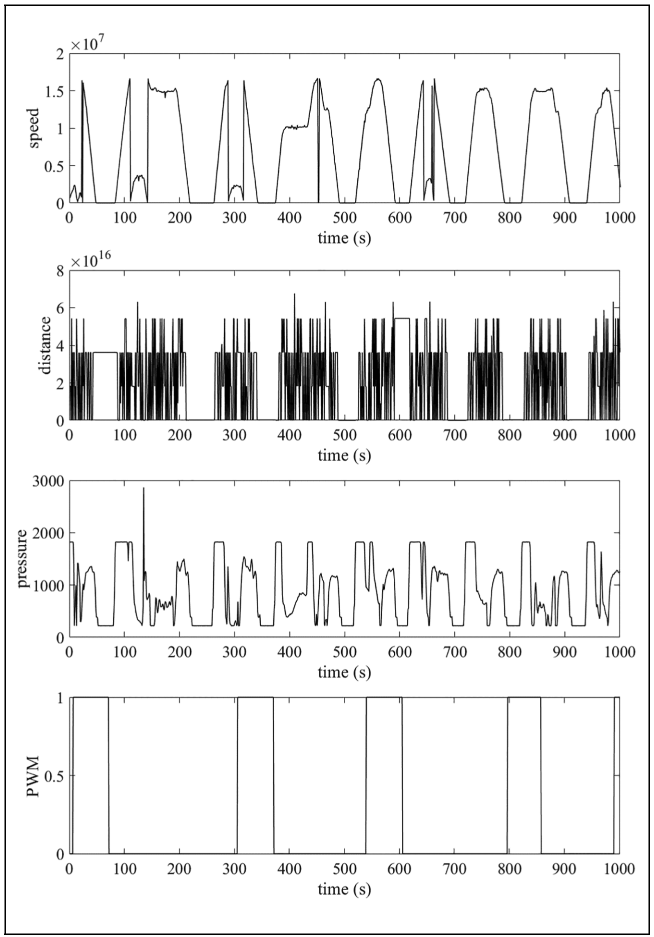


**Figure 3.** Sample plot of the measured signals.
*Note*: PWM = Pulse Width Modulation.

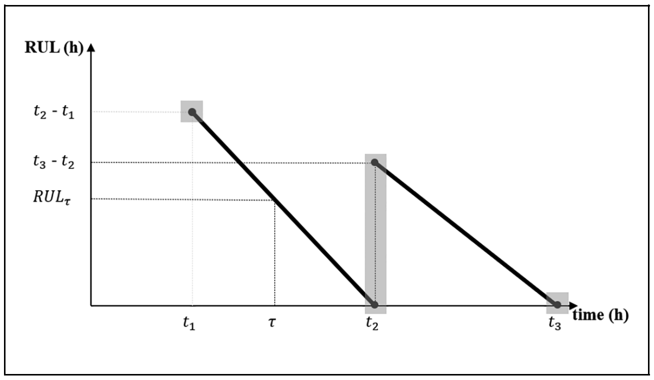


**Figure 4.** Example of remaining useful life (RUL) function.

1. RUL is observed to vary with time. Therefore, it can be considered as a time-series. Each point in the plot in Figure 4 has a corresponding value of RUL with respect to time.
2. RUL degrades linearly with time. That is, the rate at which the RUL decreases until it approaches the fault occurrence time is constant.
3. RUL is dependent on the occurrence of the latest and next faults, and does not depend on the previous fault occurrences.
4. Two points have an RUL value of 0: $t_2$ and $t_3$. Since the RUL is 0, this clearly indicates that a fault occurred at times $t_2$ and $t_3$.
5. The point in time at which the fault occurs is unique since it has two values: 0 and the RUL for the next fault occurrence. In the example shown

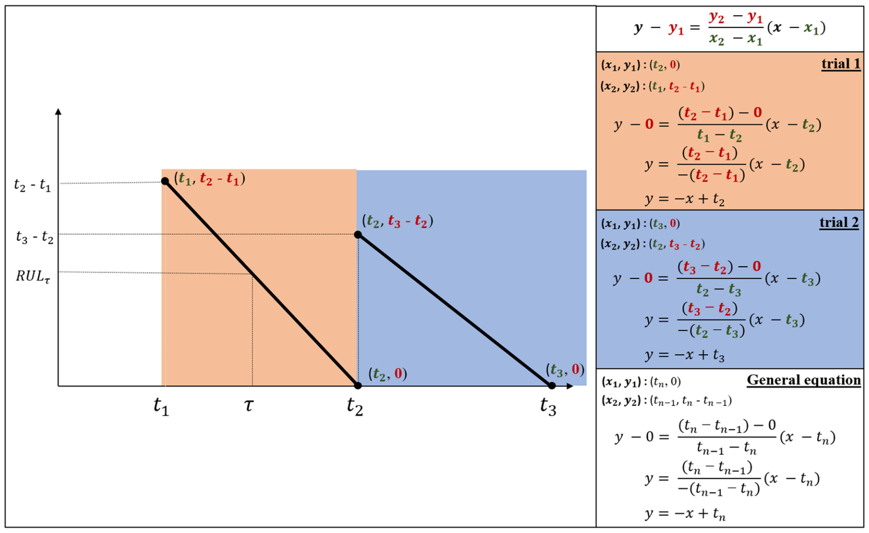

$t_2$ - $t_1$
$t_3$ - $t_2$
$RUL_\tau$
$(t_1, t_2 - t_1)$
$(t_2, t_3 - t_2)$
$(t_2, 0)$
$(t_3, 0)$
$t_1$
$\tau$
$t_2$
$t_3$
$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$
trial 1
$(x_1, y_1) : (t_2, 0)$
$(x_2, y_2) : (t_1, t_2 - t_1)$
$y - 0 = \frac{(t_2 - t_1) - 0}{t_1 - t_2}(x - t_2)$
$y = \frac{(t_2 - t_1)}{-(t_2 - t_1)}(x - t_2)$
$y = -x + t_2$
trial 2
$(x_1, y_1) : (t_3, 0)$
$(x_2, y_2) : (t_2, t_3 - t_2)$
$y - 0 = \frac{(t_3 - t_2) - 0}{t_2 - t_3}(x - t_3)$
$y = \frac{(t_3 - t_2)}{-(t_2 - t_3)}(x - t_3)$
$y = -x + t_3$
General equation
$(x_1, y_1) : (t_n, 0)$
$(x_2, y_2) : (t_{n-1}, t_n - t_{n-1})$
$y - 0 = \frac{(t_n - t_{n-1}) - 0}{t_{n-1} - t_n}(x - t_n)$
$y = \frac{(t_n - t_{n-1})}{-(t_{n-1} - t_n)}(x - t_n)$
$y = -x + t_n$

**Figure 5.** Derivation of remaining useful life (RUL) function.

in Figure 4, time $t_3$ has two corresponding RULs: (1) 0, and (2) $t_3 - t_2$.

The observations made above are supposed by the formulation of a mathematical equation. The mathematical equation represents the function of RUL with respect to time. The derivation of the mathematical equation is shown in Figure 5.

In Figure 5, the RUL function is derived using a two-point form equation. The two points used are: (1) the latest fault occurrence point, and (2) the next fault occurrence point. In trial 1, the latest fault occurrence point is at $(t_1, t_2 - t_1)$ and the next fault occurrence point is at $(t_2, 0)$. The latest fault occurrence point $(t_1, t_2 - t_1)$ indicates that the fault occurs at time $t_1$ with an RUL value of $t_2 - t_1$. Moreover, the next fault occurrence point $(t_2, 0)$ indicates that the fault occurs at time $t_2$ with an RUL value of 0. In trial 2, the latest fault occurrence point is at $(t_2, t_3 - t_2)$ and the next fault occurrence point is at $(t_3, 0)$. The latest fault occurrence point $(t_2, t_3 - t_2)$ indicates that the fault occurs at time $t_2$ with an RUL value of $t_3 - t_2$. Moreover, the next fault occurrence point $(t_3, 0)$ indicates that the fault occurs at time $t_3$ with an RUL value of 0. Furthermore, it can be observed that the next fault occurrence point $(t_2, 0)$ in trial 1 has the same $x$-coordinate value as the latest fault occurrence point $(t_2, t_3 - t_2)$ in trial 2. This indicates that RUL reduces up to 0. An RUL value of 0 indicates a fault occurrence. At the time of that fault occurrence, the value of RUL for the next fault occurrence is reset to a non-zero value. The general equation can be obtained using points $(t_{n-1}, t_n - t_{n-1})$ and $(t_n, 0)$ where $t_{n-1}$ is the latest fault occurrence time and $t_n$ is the next fault occurrence time. The general equation is shown in Equation 1:

$$y = -x + t_n \tag{1}$$

where
$y$ represents RUL, and
$x$ represents the current time.
For example, RUL at time $\tau$ in Figure 5 is shown in Equation 2:

$$y = -\tau + t_2 \tag{2}$$

In computing RUL, first, it is necessary to extract the historical fault occurrence information. Then, the time information in the OPER file is also extracted. The time information is the time step at which the data is obtained. Finally, each time step is simply subtracted from its corresponding next fault occurrence time.

After obtaining the time-series of RUL, the machine learning model is trained. In training the machine learning model, data from CSV file 2 is used as input, and data of RUL is used as output. Specifically, a DNN model is trained to predict RUL using the data from CSV file 2 as input. The network architecture of the DNN model used in this study is shown in Figure 6. As shown in the figure, the model consists of an input layer, four dense layers with rectified linear unit (ReLU) activation functions, and a classifier output. Instead of using traditional methods, a DNN model is used for fault occurrence prediction tasks. Unlike traditional methods, DNN can automatically extract and learn features from a given dataset. That is, DNN eliminates the need for knowledge of the

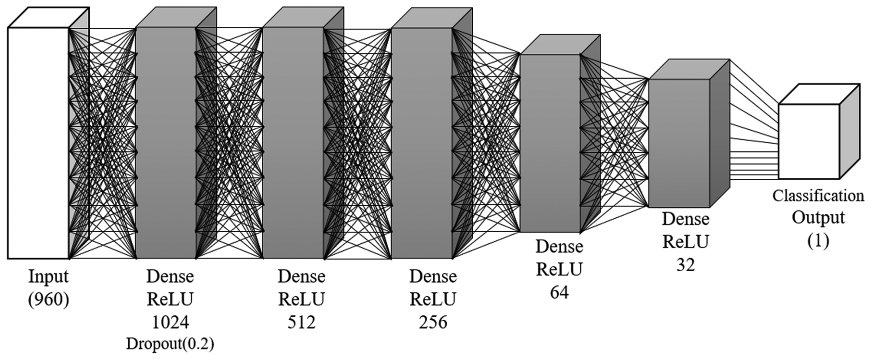


**Figure 6.** Network architecture of the deep neural network (DNN) model for fault occurrence prediction.

**Table 1.** Data Size Before and After Data Filtering

| Line | Size before filtering | Size after filtering | Size difference |
|---|---|---|---|
| 2 | 175.4 GB | 32.3 GB | 143.1 GB |
| 7 | 175.4 GB | 47.2 GB | 128.2 GB |

data and the manual feature selection process. This characteristic is useful, especially when handling a big amount of data. In L'Heureuz et al., challenges encountered by several traditional techniques are complied, summarized, and organized (*12*). In addition, L'Heureuz et al. also discussed deep-learning-based approaches and techniques (*12*). Aside from handling a big amount of data, DNN is capable of capturing the dynamics in the data (*13*). In training, the data with sizes listed in Table 1 are used. Among the data, 70%, 20%, and 10% are used for training, validation, and testing, respectively. After training successfully, the DNN model can then be tested using data from CSV file 2 that has not been used in training, or using newly obtained data from TCMS. RUL prediction is defined as the prediction of the remaining time before a fault occurs. That is, if a fault is expected to after 1 h, the predicted RUL is then 1 h. RUL prediction enables the optimal schedule of maintenance, and prevents undesirable system unavailability and numerous downtime losses (*14*, *15*). RUL prediction usually takes the historical data of failure-time as input and then outputs the predicted future failure-time. On the other hand, fault occurrence prediction is defined as the prediction of the exact time at which a fault will occur. That is, the result of RUL prediction is dependent on the time at which the prediction is made, while the result of fault occurrence prediction is fixed. For instance, given the example in Figure 5, RUL at time $t_1$ is $t_2-t_1$, and RUL at time $\tau$ is $t_2-\tau$. On the other hand, the fault occurrence times at times $t_1$ and $\tau$ are both equal to $t_2$. To obtain not only the fault occurrence time but also the exact remaining time before a fault occurs, the fault occurrence is predicted using the result of RUL prediction. That is, if the predicted RUL at 10:00 a.m. is 10 h, the fault occurrence time is predicted to be at 8:00 p.m. Then, if the predicted RUL at 12 noon is 8 h, the fault occurrence is still predicted to be at 8:00 p.m.

## Results

The data used in this study are obtained from an actual TCMS attached to a train in South Korea. The dataset consists of data from trains with two different travel paths or lines: line 2 and line 7. Because of the limitations of the availability of the data, this study is limited to lines 2 and 7 of the trains in South Korea. Data from line 2 have a total of 214 attributes or features measured from June 6, 2018, to March 20, 2020, while the data from line 7 have a total of 313 attributes or features measured from June 1, 2018, to June 13, 2019. For line 2, five faults with the most frequent occurrence were considered. The line 2 fault codes and descriptions are listed in Table 1. For line 7, 12 faults with the most frequent occurrence were considered. The line 7 fault codes and descriptions are also listed in Table 2.

In predicting RUL, the three prediction ranges are set: (1) 24 h, (2) 72 h, and (3) maximum time interval (MTI). MTI uses the maximum value between two fault occurrences. MTI for each fault code of line 2 and line 7 are listed in Table 3.

All data processing and training methods were performed on the desktop computer, while the testing method is performed on an actual device. Specifically, NVIDIA's Jetson nano was used for testing. The plot of DNN training and testing results for each fault code of line 2 and line 7 are shown in Figure 7.

**Table 2.** Line 2 and Line 7 Fault Codes and Descriptions

| Line | Code | Fault type |
|---|---|---|
| 2 | 11 | Computer communication fault (train computer–car computer [TC–CC]) |
| | 104 | Brake fault (unintentional engagement of emergency brake) |
| | 212 | Brake fault (hardware fault) |
| | 304 | Brake fault (failure of a brake during intended operation) |
| | 501 | Reversal control fault |
| 7 | 35 | Computer communication fault (car computer [CC]) |
| | 130 | Power cut-off |
| | 443 | Brake fault (unintentional engagement of emergency brake) |
| | 641 | Automatic train control (ATC) (hardware failure) |
| | 684 | ATC (software failure) |
| | 703 | Door fault (failure to close the left door on car 1) |
| | 704 | Door fault (failure to close the left door on car 2) |
| | 705 | Door fault (failure to close the left door on car 3) |
| | 706 | Door fault (failure to close the left door on car 4) |
| | 707 | Door fault (failure to close the right door on car 1) |
| | 710 | Door fault (failure to close the right door on car 4) |
| | 711 | Door fault (failure to close all doors) |

**Table 3.** Maximum Time Interval (MTI) for Each Fault Code of Line 2 and Line 7

| Line | Code | Maximum time interval | |
|---|---|---|---|
| | | Seconds | Hours |
| 2 | 11 | 15,552,000 | 7,344 |
| | 104 | 6,226,700 | 1,729 |
| | 212 | 45,434,801 | 12,620 |
| | 304 | 8,733,149 | 2,425 |
| | 501 | 11,652,479 | 3,326 |
| 7 | 35 | 4,403,986 | 1,223 |
| | 130 | 2,636,791 | 732 |
| | 443 | 15,552,000 | 7,344 |
| | 641 | 11,569,94 | 3,221 |
| | 684 | 7,394,670 | 2,054 |
| | 703 | 4,815,330 | 1,337 |
| | 704 | 7,823,937 | 2,173 |
| | 705 | 4,497,009 | 1,249 |
| | 706 | 4,932,262 | 1,370 |
| | 707 | 7,350,683 | 2,041 |
| | 710 | 9,780,738 | 2,716 |
| | 711 | 7,687,797 | 2,135 |

**Table 4.** Average Deep Neural Network Training and Testing Accuracy at 24 h, 72 h, and Maximum Time Interval (MTI) Prediction Range

| 24 h | | 72 h | | MTI | |
|---|---|---|---|---|---|
| Test | Train | Test | Train | Test | Train |
| 90.08% | 89.94% | 59.59% | 82.20% | 56.07% | 90.24% |

The average training and testing accuracies for 24 h, 72 h, and MTI prediction ranges are listed in Table 4.

The testing accuracy shown in Figure 7 and Table 4 are obtained by uploading the DNN-based fault occurrence prediction model on an actual device. Specifically, the model is uploaded on NVIDIA's Jetson nano. In addition to that, a graphical user interface (GUI) is designed to enable users without prior experience in model testing to perform fault occurrence prediction. The GUI is shown in Figure 8.

GUI provides the user with two major functions: (1) fault occurrence prediction, and (2) data transformation. In the fault occurrence prediction function, GUI enables the users to perform the prediction on an actual Jetson nano device using the DNN developed in this study. The fault occurrence prediction function can be accessed by clicking the "Prediction" tab. Aside from the fault occurrence prediction, GUI also provides a data transformation function. The data transformation function takes the OPER and ARM files as input and outputs an equivalent CSV file. The data transformation function can be accessed by clicking the "Transform" tab.

## Discussion

As shown in Figure 7, both training and testing accuracy are highest at a 24 h prediction range, whereas the accuracy of both training and testing at 72 h and MTI prediction ranges varies with the fault code. Looking more closely, the training and testing accuracy are almost equal for each fault code at 24 h prediction range. It is also observed that the plots of training and testing accuracy at 24 h prediction range coincide at some points. This result implies that the model trained to predict the fault occurrence with a 24 h prediction range has the same performance even with two different types of data (line 2

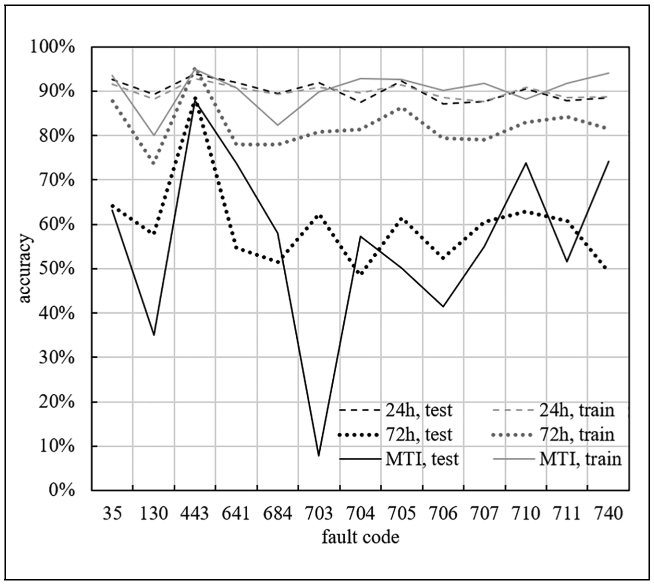


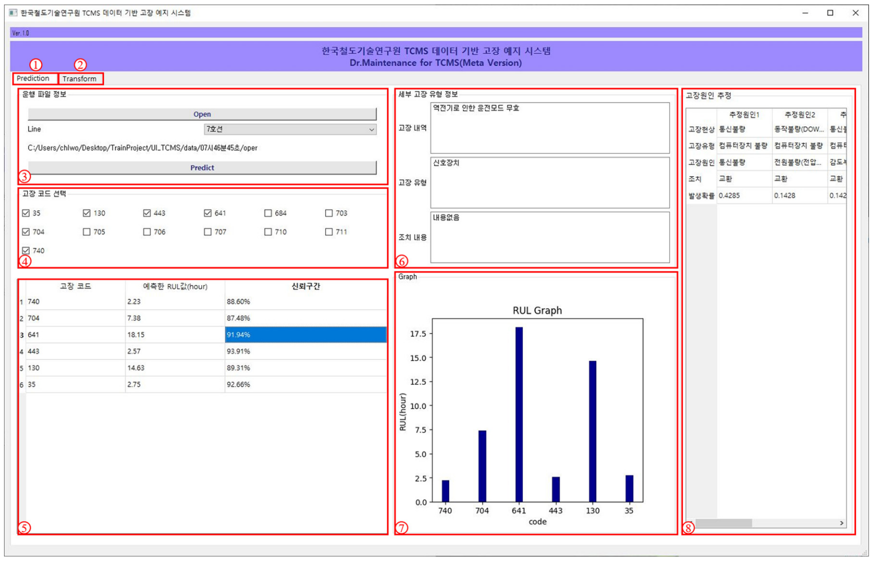
**Figure 7.** Deep neural network (DNN) training and testing accuracy at 24 h, 72 h, and maximum time interval (MTI) prediction ranges.

and line 7). Therefore, the model trained with 24 h prediction range is generalized successfully. On the other hand, the training and testing accuracy have a big gap at 72 h and MTI prediction ranges. It is also observed that the plots of training and testing accuracy do not even coincide. Therefore, the model failed to generalize.

As shown in Table 4, DNN has the highest testing accuracy at 24 h prediction range. On the other hand, the training accuracy at 72 h and MTI prediction ranges are almost equal, with only around 0.33% difference. It is also observed that the models trained with 72 h and MTI prediction ranges are overfitting. That is, the model performs well on training but poorly on testing. Overall, the results presented in this paper show that the DNN-based model has high accuracy and reliability in predicting the occurrence of fault at a 24 h prediction range.

## Conclusions

In this study, a DNN-based fault occurrence prediction for a train system is developed. Data gathered during an actual operation of the train system in South Korea are used to train the model. The data consist of the states of components or equipment, and the values of signals measured from different subsystems of a train, therefore enabling fault occurrence prediction on the overall train system and not only in one equipment or subsystem. Actual data are used to capture the actual behavior of a certain component or equipment in a subsystem. The DNN is trained on a desktop computer and is tested on an actual device. Specifically, a Jetson nano device is used to test the DNN model. In testing the DNN model, a GUI for Jetson nano is also developed. The result obtained from the model testing performed in this study yields an average of 90.08% fault occurrence prediction


한국철도기술연구원 TCMS 데이터 기반 고장 예지 시스템
Ver. 1.0
한국철도기술연구원 TCMS 데이터 기반 고장 예지 시스템
Dr.Maintenance for TCMS(Meta Version)
①
Prediction
②
Transform
운행 파일 정보
Open
Predict
③
고장 코드 선택
④
⑤
세부 고장 유형 정보
⑥
Graph
RUL Graph
RUL(hour)
code
⑦
고장원인 추정
⑧

Line: 7호선

C:/Users/chlwo/Desktop/TrainProject/UI_TCMS/data/07시46분45초/oper

[✓ 35] [✓ 130] [✓ 443] [✓ 641] 684 703
[✓ 704] 705 706 707 710 711
[✓ 740]

| | 고장 코드 | 예측한 RUL값(hour) | 신뢰구간 |
|---|---|---|---|
| 1 | 740 | 2.23 | 88.60% |
| 2 | 704 | 7.38 | 87.48% |
| 3 | 641 | 18.15 | 91.94% |
| 4 | 443 | 2.57 | 93.91% |
| 5 | 130 | 14.63 | 89.31% |
| 6 | 35 | 2.75 | 92.66% |

고장 내역: 역전기로 인한 운전모드 무효

고장 유형: 신호장치

조치 내용: 내용없음

| | 추정원인1 | 추정원인2 | 추 |
|---|---|---|---|
| 고장현상 | 통신불량 | 동작불량(DOW... | 통신 |
| 고장유형 | 컴퓨터장치 불량 | 컴퓨터장치 불량 | 컴퓨터 |
| 고장원인 | 통신불량 | 전원불량(전압... | 감도 |
| 조치 | 교환 | 교환 | 교환 |
| 발생확률 | 0.4285 | 0.1428 | 0.142 |

**Figure 8.** Graphical user interface (GUI) for fault occurrence prediction on Jetson nano.

accuracy. It is expected that this study will enable onboard fault occurrence prediction on the train system, thus increasing the reliability and safety of passengers. For future work, this work can be extended by optimizing the DNN model and deploy real-time fault occurrence prediction on a low-power microcontroller. Moreover, aside from the DNN model, other models and architectures can be explored.

## Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: A. Caliwag, W. Lim; data collection: S. Han, K. Park; analysis and interpretation of results: A. Caliwag; draft manuscript preparation: A. Caliwag, W. Lim. All authors reviewed the results and approved the final version of the manuscript.

## Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Ministry of SMEs and Startups, South Korea (S2829065, S3010704), and by the National Research Foundation of Korea (2020R1A4A101777511, 2021R1I1A3056900). In part, this research was also supported by a grant from the R&D Program of the Korea Railroad Research Institute, Republic of Korea.

## ORCID iDs

Angela Caliwag https://orcid.org/0000-0002-0279-935X
Seok-Youn Han https://orcid.org/0000-0002-9951-3193

## References

1. Seoul Metropolitan Government. *2019 Transportation Data of Seoul Analyzed Through Big Data*. SMG. http://english.seoul.go.kr/2019-transportation-data-of-seoul-analyzed-through-big-data/.
2. Changyuan, L., L. Xiaoming, and Y. Panpan. Train Control Management System Safety Assessment. *Proc., The International Conference on Electrical and Information Technologies for Rail Transportation (EITRT)*, Changchun, China, 2013.
3. Satoru, I., S. Takayuki, S. Keishi, and S. Keita. Latest Train Control and Management System Technologies for Improving Safety and Maintainability. *Latest Developments for Safe and Reliable Railways*, Vol. 67, 2018, pp. 54–58.
4. Hongtian, C., J. Bin, and L. Ningyun. A Newly Robust Fault Detection and Diagnosis Method for High-Speed Trains. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 20, 2019, pp. 2198–2208.
5. Dandan, P., L. Zhiliang, W. Huan, Q. Yong, and J. Limin. A Novel Deeper One-Dimensional CNN With Residual Learning for Fault Diagnosis of Wheelset Bearings in High-Speed Trains. *IEEE Access*, Vol. 7, 2018, pp. 10278–10293.
6. Hexuan, H., T. Bo, G. Xuejiao, W. Wei, and W. Huihui. Intelligent Fault Diagnosis of the High-Speed Train With Big Data Based on Deep Neural Networks. *IEEE Transactions on Industrial Electronics*, Vol. 13, 2017, pp. 2106–2114.
7. Hongtian, C., J. Bin, C. Wen, and Y. Hui. Data-driven Detection and Diagnosis of Incipient Faults in Electrical Drives of High-Speed Trains. *IEEE Transactions on Industrial Electronics*, Vol. 66, 2019, pp. 4716–4725.
8. Yueyue, M., G. Wei, C. Baigen, and Z. Junzheng. Fault Prediction Method of the On-board Equipment of Train Control System Based on Grey-ENN. *Proc., CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS*, Xiamen, China, 2019, pp. 944–949.
9. Amin, Z., K. Kassymzhomart, B. Saeid, and B. Mehdi. Fault Prognosis Using Deep Recurrent Neural Network Over Vibration Signals. *IEEE Transactions on Instrumentation and Measurement*, Vol. 70, 2021, pp. 2502011–2502022.
10. Zhu, L., F. R. Yu, B. Ning, and T. Tang. Communication-Based Train Control (CBTC) Systems with Cooperative Relaying: Design and Performance Analysis. *IEEE Transactions on Vehicular Technology*, Vol. 63, 2014, pp. 2162–2172.
11. Wang, C., L. Wang, H. Chen, Y. Yang, and Y. Li. Fault Diagnosis of Train Network Control Management System Based on Dynamic Fault Tree and Bayesian Network. *IEEE Access*, Vol. 9, 2021, pp. 2618–2632.
12. L'Heureuz, A., K. Grolinger, H. Elyamany, and M. Capretz. Machine Learning With Big Data: Challenges and Approaches. *IEEE Access*, Vol. 5, 2017, pp. 7776–7797.
13. Lu, W., Y. Li, Y. Cheng, D. Meng, B. Liang, and P. Zhou. Early Fault Detection Approach With Deep Architectures. *IEEE Transactions on Instrumentation and Measurement*, Vol. 67, 2018, pp. 1679–1689.
14. Zhang, H., E. Liu, B. Zhang, and Q. Miao. RUL Prediction and Uncertainty Management for Multisensor System Using an Integrated Data-level Fusion and UPF Approach. *IEEE Transactions on Industrial Informatics*, Vol. 17, 2021, pp. 4692–4701.
15. Hu, J., Q. Sun, Z. S. Ye, and Q. Zhou. Joint Modeling of Degradation and Lifetime Data for RUL Prediction of Deteriorating Products. *IEEE Transactions on Industrial Informatics*, Vol. 17, 2021, pp. 4521–4531.