Hindawi
Journal of Advanced Transportation
Volume 2020, Article ID 8861942, 15 pages
https://doi.org/10.1155/2020/8861942

Research Article

Generative Adversarial Network-based Missing Data Handling and Remaining Useful Life Estimation for Smart Train Control and Monitoring Systems

Hyunsoo Lee,[1] **Seok-Youn Han**,[2] **and Kee-Jun Park**[2]

[1] *School of Industrial Engineering, Kumoh National Institute of Technology, Gumi 39177, Republic of Korea*
[2] *Urban Transit Research Group, Advanced Railroad Vehicle Division, Korea Railroad Research Institute, Uiwang, Republic of Korea*

Correspondence should be addressed to Hyunsoo Lee; hsl@kumoh.ac.kr

Received 29 July 2020; Revised 5 November 2020; Accepted 17 November 2020; Published 27 November 2020

Academic Editor: Ladislav Routil

As railway is considered one of the most significant transports, sudden malfunction of train components or delayed maintenance may considerably disrupt societal activities. To prevent this issue, various railway maintenance frameworks, from "periodic time-based and distance-based traditional maintenance frameworks" to "monitoring/conditional-based maintenance systems," have been proposed and developed. However, these maintenance frameworks depend on the current status and situations of trains and cars. To overcome these issues, several predictive frameworks have been proposed. This study proposes a new and effective remaining useful life (RUL) estimation framework using big data from a train control and monitoring system (TCMS). TCMS data is classified into two types: operation data and alarm data. Alarm or RUL information is extracted from the alarm data. Subsequently, a deep learning model achieves the mapping relationship between operation data and the extracted RUL. However, a number of TCMS data have missing values due to malfunction of embedded sensors and/or low life of monitoring modules. This issue is addressed in the proposed generative adversarial network (GAN) framework. Both deep neural network (DNN) models for a generator and a predictor estimate missing values and predict train fault, simultaneously. To prove the effectiveness of the proposed GAN-based predictive maintenance framework, TCMS data-based case studies and comparisons with other methods were carried out.

1. Introduction

Railway infrastructure has been one of the essential infrastructures not only at a national level, but also across continents. In terms of ground cargo and freight transport, the railway system is the most important infrastructure. A number of research studies have focused on detections of aberrant situations in trains. For instance, unexpected failures in train components may catastrophically harm the passengers' safety. Moreover, a maintenance delay may result in subsequent heavy delays in overall train schedule. Thus, maintenance frameworks for railway infrastructure have received significant attention. A number of existing research studies have proposed various railway maintenance frameworks and relevant applications for more reliable railroad operations.

Early railway maintenance frameworks are based on periodic time-based maintenance [1], which is still an effective technique for checking railway components. Monthly or quarterly inspection belongs to this type of maintenance. Recently, maintenance framework has evolved to "preventive maintenance" in contemporary railroad systems. Preventive maintenance is classified into "time-based maintenance" and "distance-based maintenance" in general. Most railway operators utilize both maintenance frameworks, simultaneously.

Moreover, the maintenance framework is evolving with the development of technologies of the fourth industrial revolutions. Of these technologies, Internet of Things (IoT) technology is the most relevant for enhancing railway maintenance. Fraga-Lamas et al. [2] summarized the utilization of IoT technologies in train maintenance. IoT-based embedded systems enable the detection of abnormal status of railway components in real time, where the signals are subsequently transferred to a secured database system. In general, most of the train systems have their own management systems, such as train control and monitoring system (TCMS) for storing various train data and for managing trains. Based on TCMS-based research studies [3, 4], TCMS is a system with control, communication, and management functions for all train platforms and applications. As the system collects operation and management-based data for trains and their connected cars, a huge amount of data can be collected and analysed. Several research studies [5, 6] used TCMS data for estimating energy consumption of trains or for controlling train doors safely. However, relatively fewer studies on predictive maintenance were carried out.

This study focuses on developing a new and effective predictive maintenance framework using TCMS data. In this study, remaining useful lives (RUL) of various train modules are predicted using a proposed deep learning method. In order to measure RUL of train modules, this study predicts time periods to the relevant trains' faults and malfunctions. In this paper, this time to failure (TTF) for a certain train fault is defined a RUL of a certain fault. However, prerequisite conditions are necessary to handle data issues in TCMS. As TCMS data could include missing values, their handling mechanism must be embedded in a relevant RUL estimation framework.

This study applies a generative adversarial network (GAN) to handle missing values in TCMS. The following section provides relevant background knowledge and literature review. Section 3 examines TCMS data and relevant data issues. Sections 4 and 5 present a GAN-based predictive maintenance framework and its verifications using various numerical analyses, respectively.

2. Background and Literature Review

This study utilized TCMS data to estimate train component status and predict their RULs. The proposed framework is classified as a predictive maintenance framework in train systems. As discussed in the previous section, the maintenance paradigm in train transportation has converged with the technologies of the fourth industrial revolution. The time and distance-based maintenance frameworks have been combined with monitoring-based methods. Several sensing systems have been developed and installed for more detailed examinations of trains' components. Sharma et al. [7] detected breakage of railway tracks using vibration sensors. Sireesha et al. [8] used a radio frequency-based method to detect rails' broken status. These sensing systems have integrated with Internet of Things (IoT-) based frameworks. Lee [9, 10] developed various Industrial IoT (IIoT) systems to monitor abnormal manufacturing signals and estimate production performance indices in multiple supply chains. The detected signals are transmitted to a cloud server, where industrial big data analytics analyses the collected data and takes preventive measures for better production controls. These technologies and frameworks have been applied to various train systems and their relevant monitoring-based/condition-based maintenance. Hitachi [11] proposed Lumada IoT Platform© as a monitoring-based maintenance system for its railway system.

While various monitoring-based methods for detecting abnormal status of railway components have been introduced, deep learning methods and relevant data analytics have been integrated into predictive maintenance. Corman et al. [12] applied a data-driven method to estimate the remaining life of a light rail braking system in a train. McKinsey [13] suggested similar approaches to enhance rail operations using digital maintenance technologies. Atamuradov et al. [17] and Liden [18] summarized comprehensive overviews on railway infrastructure maintenance. Table 1 provides various time-based, monitoring-based, and data-driven maintenance frameworks and their applications.

As shown in Table 1, data driven analytics has been introduced for better rail maintenance. Among a number of data sets in a train framework, the TCMS data is the most comprehensive data, as it includes operation, parameter settings, and other information on train components. Figure 1 shows the various TCMS subcomponents that are installed in Korean trains and cars.

In general, TCMS is an essential system for controlling electrical multiple units (EMU) in each train and car in a train system. Thus, control parameters and operation data are stored in TCMS. While TCMS is mainly used to control trains and cars, the usage of TCMS data for various purposes has been suggested. Table 2 shows various applications that use TCMS data. As shown in Table 2, most of the applications that use TCMS data have focused on monitoring-based maintenance.

While TCMS data have been used comparatively less with more advanced maintenance analytics, several industrial projects including Shift2Rail [21] have suggested predictive maintenance frameworks using TCMS data. However, these projects provide only conceptual frameworks or experimental-level demonstrations. In particular, big data analytics and more advanced data mining methods are seldom applied in TCMS-based predictive maintenance frameworks. To address this issue, this study proposes a new and effective predictive maintenance using deep learning methods and real-time TCMS data handling modules.

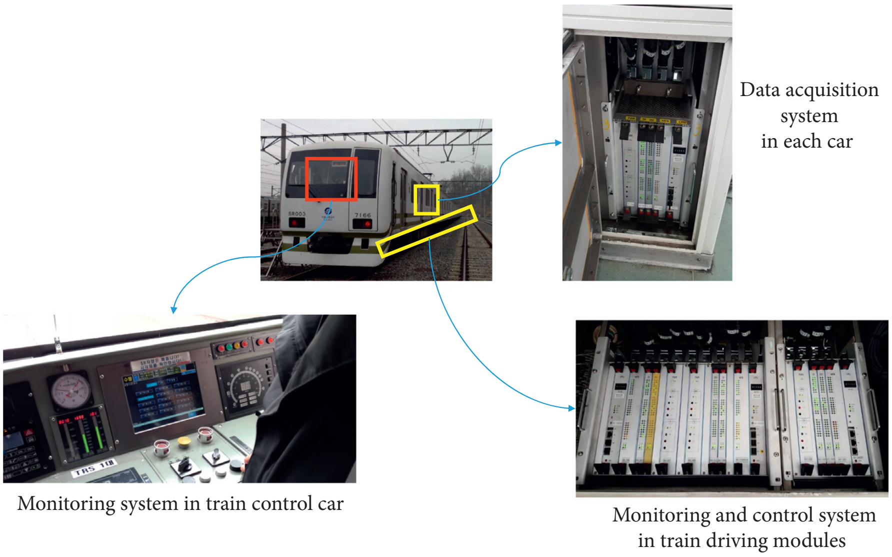
3. Missing Value Issues in TCMS Data for Predictive Maintenance Framework

The proposed predictive maintenance framework uses TCMS data for predicting RULs in a certain breakage. As shown in (1), the RUL (RUL_j, $j \in J$; J is a set of integers) of a certain breakage (j) is estimated using the TCMS data (X) and used as a main reference for setting up train and cars maintenance schedules. The function $f(\cdot)$ is modelled using a deep learning-based network architecture and is explained in the following section.

$$\mathrm{RUL}_j = f\left(x_{i\in I}\right). \tag{1}$$

Table 1: Existing time, monitoring, and data-driven maintenance applications in rail systems.

Existing research studies	Target railway components	Methods and characteristics	Maintenance type		
			Time/ distance	Monitoring	Data-driven
Faiz and Singh [14]	Railway track	(i) Detection of track geometry (ii) Usage of rail profile-based regression model	O	—	—
Sharma et al. [7]	Railway	(i) Vibration sensor-based estimation of railway breakage	—	O	—
Shaikh et al. [15]	Solid axle wheel sets	(i) Installation of additional sensors (vibration sensors for capturing lateral and yaw dynamics) (ii) Vibration model-based simulation	—	O	—
Letot et al. [16]	Railway track point machine	(i) Degradation assessment and data-based RUL estimation	—	—	O
Corman et al. [12]	Train breaking system	(i) Work, maintenance, and failure data-based reliability estimation (ii) Usage of Weibull distribution	O	O	O



Data acquisition system in each car

Monitoring system in train control car

Monitoring and control system in train driving modules

Figure 1: Train control and monitoring system (TCMS) in Korean trains and cars. (a) Monitoring system in train control car. (b) Monitoring and control system in train driving modules. (c) Data acquisition system in each car.

Table 3 summaries the TCMS data used in this study and the general specifications of the proposed RUL prediction framework.

As shown in Table 3, the input data of the proposed predictive maintenance framework is the TCMS data. Figure 2(a) shows a part of the TCMS data, which is an encrypted data. In general, TCMS data is classified into two types: operation data (oper) and alarm data (arm) as shown in Figure 2(a). The TCMS data is stored with each car no., the date, and time. While the “oper” data describes information such as train identification and operation and other relevant train parameters, the recorded “arm” data includes various warning signals and relevant alarm codes. These alarm level information and other warning data are written using the predefined criteria, such as status levels of train components and other relevant sensor measuring ranges. Thus, “oper” data is used as input data, while the output of the proposed RUL estimation framework is driven by the “arm” data. If “arm” data can be predicted using a series of “oper” data, a real-time predictive maintenance can be applied. Hence, the proposed predictive maintenance framework uses “oper” data as input vector. The RUL variable is extracted from the “arm” data and fault/maintenance history. The fault/maintenance data clarifies the relationship among “arm” data and a certain train defect. RUL data is extracted from the “arm” data and its relationship to a certain defect is obtained using the mapping between both data.

However, TCMS data cannot be directly used owing to their encrypted formats and missing value issue. As shown in Figure 2(a), both types of data are encrypted for various reasons, such as data protection, data size reduction, and sensor driver encryption. This indicates that the data need to be decrypted prior to any further data processing.

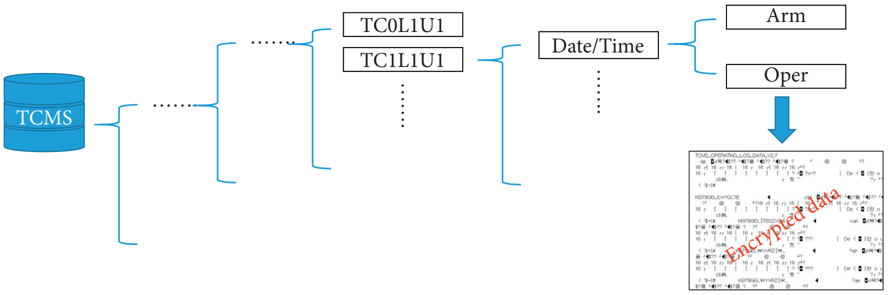
To decrypt the data, hex data-based decoding is performed as an essential prerequisite procedure using the encryption rule for TCMS. Then, the hex-formatted data are converted into number-formatted data for subsequent deep learning processing. Figure 2(b) shows a program developed

TABLE 2: Applications that use TCMS data.

Research studies	Applications	TCMS data
Ito et al. [6]	(i) Safe door operation (ii) Automatic power changer-based driver advisory system	(i) EMU functions in TCMS
Neil [19]	(iii) Railway safety monitoring-based maintenance	(ii) Transaction data in TCMS
Kim et al. [13]	(iv) Analysis of train energy consumption considering driving patterns	(iii) Driving time (iv) Train's driving speed (v) Railway track data
Xu et al. [20]	(v) Queuing theory-based maintenance cycle scheduling for an urban rail transit system	(vi) Running distance, velocity, and mileage data Maintenance schedule
Shift2Rail project [21]	(vi) Monitoring of cargo condition (vii) A conceptual and experimental project	(vii) TCMS data (viii) Additional sensing systems (e.g., ultrasonic sensor and other wireless sensors)

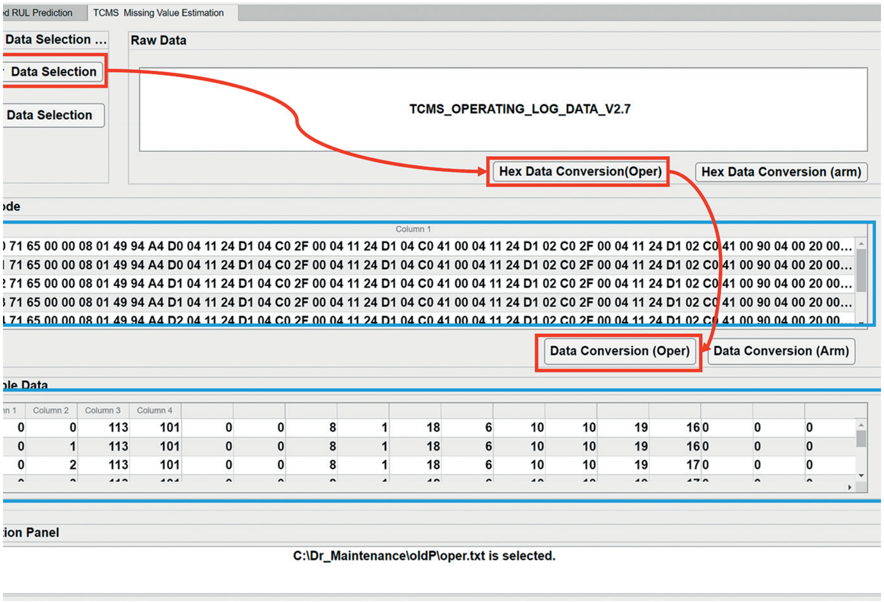
TABLE 3: Specification of the proposed predictive maintenance framework.

Content	Classification	Issues
Data specification	(i) Data source: TCMS data (2018.6~2019.05) (ii) Data from the seventh line in subway system, Republic of Korea	Big data
TCMS data specification	(i) Number of attributes: 2643 per one record Existence of a number of missing values in one record (ii) Data format: encrypted data	(i) Data decryption is needed (ii) Missing value handling is needed
Fault/alarm data	(i) Number of attributes: 56 (ii) Data format: encrypted text data	(i) Data decryption is needed
Predictive maintenance framework	(i) Data input: TCMS data (ii) Output: the estimated RUL (iii) Mechanism: GAN-based deep neural network	—



(a)

FIGURE 2: Continued.



```
71 65 00 00 08 01 49 94 A4 D0 04 11 24 D1 04 C0 2F 00 04 11 24 D1 04 C0 41 00 04 11 24 D1 02 C0 2F 00 04 11 24 D1 02 C0 41 00 90 04 00 20 00...
71 65 00 00 08 01 49 94 A4 D0 04 11 24 D1 04 C0 2F 00 04 11 24 D1 04 C0 41 00 04 11 24 D1 02 C0 2F 00 04 11 24 D1 02 C0 41 00 90 04 00 20 00...
71 65 00 00 08 01 49 94 A4 D1 04 11 24 D1 04 C0 2F 00 04 11 24 D1 04 C0 41 00 04 11 24 D1 02 C0 2F 00 04 11 24 D1 02 C0 41 00 90 04 00 20 00...
71 65 00 00 08 01 49 94 A4 D1 04 11 24 D1 04 C0 2F 00 04 11 24 D1 04 C0 41 00 04 11 24 D1 02 C0 2F 00 04 11 24 D1 02 C0 41 00 90 04 00 20 00...
71 65 00 00 08 01 49 94 A4 D2 04 11 24 D1 04 C0 2F 00 04 11 24 D1 04 C0 41 00 04 11 24 D1 02 C0 2F 00 04 11 24 D1 02 C0 41 00 90 04 00 20 00...
```

n 1	Column 2	Column 3	Column 4													
0	0	113	101	0	0	8	1	18	6	10	10	19	16	0	0	0
0	1	113	101	0	0	8	1	18	6	10	10	19	16	0	0	0
0	2	113	101	0	0	8	1	18	6	10	10	19	17	0	0	0

C:\Dr_Maintenance\oldP\oper.txt is selected.

(b)

Figure 2: Encrypted TCMS data sample and the developed conversion tool. (a) A TCMS data sample and its encrypted texts. (b) The decryption and conversion software program developed in this study.

in this study, which converts hex-formatted data to number-formatted interpretable data. The "oper" data has 2,643 attributes, which include train identification, operation time, station time, velocity, sensor data, and other information. The "arm" data has 56 attributes such as system failure, warning, and alarm information.

The data conversion process is performed as a preprocessing step. However, the main issue is the frequent occurrence of missing values in the TCMS data. Missing values in the TCMS data exist due to various reasons (e.g., sudden breakdown of sensors, malfunction of devices, and/or sudden changes of electric current). Missing input and output data issues have to be resolved prior to training a deep learning-based predictive maintenance model. This is one of the most common issues in manufacturing [22–24], transportation, and other data handling processes. Table 4 summaries various methods for handling missing values.

As shown in Table 4, most of the early relevant research studies tried to remove records with missing values. While these methods provided complete data for input, it could result in the lack of a training data. However, this limitation can be overcome by estimating missing values. The simplest estimation method is to consider data in the same attribute and then extract a probability density function using the data associated with the same attribute. For instance, the Gaussian mixture model (GMM) can be applied for capturing the characteristics of the data [30]. Then, a random number generated using the reasoned GMM model is used as an estimator for a missing value. However, in this method, relationships among other attributes are ignored. To address this issue, another estimation method, missing value estimation method, which considers the overall dependencies among the data attributes can be used. In general, multivariate statistical approaches, regression model, or multivariate nearest neighbour methods [31] can be applied for describing data dependencies. Then, the missing values can be estimated and substituted using Markov Chain Monte Carlo (MCMC)-based random number generation methods.

While these methods have worked only for estimating missing values, the recent methods tend to generate not only missing data, but also overall data. The generative adversarial network (GAN) is the representative method among them. In industrial big data, one of the issues is the lack of certain fault data. The lack of certain types of data may lower training performances of applied learning mechanisms. While the initial purpose of adversarial network (G) in GAN is to increase the classification ability of a classification network (D), a well-trained adversarial network can generate data which fits to an objective. Table 5 summarizes the learning algorithm of GAN.

The gradients for G and D are driven using

$$\min_{G}\max_{D} f_D(X') + f_G(Z). \qquad (2)$$

As shown in (2), $f_D(X')$ and $f_G(Z)$ denote $E_{X'}[\log D(X)]$ and $E_Z[\log(1 - D(G(Z)))]$, respectively. Several research studies applied GAN for generating fault data in automotive [22], semiconductor [23], and steel production processes [24]. This study used GAN to handle the missing value issues in the TCMS data as well as to predict RULs in train components.

4. Generative Adversarial Network-Based Predictive Maintenance Framework

The proposed RUL estimation framework predicts the RUL of a certain defect or a malfunction. To focus on major

Table 4: Various methods for handling missing values.

Methods for handling missing values	Detailed methods	Related research studies
Removals of data sets with missing values	(i) Ignorance of records with missing values (ii) Data without missing values are used only for an input vector	A number of research studies including [25]
Estimation of missing values (I)	(iii) Estimation of missing values using mean, MCMC, and nearest neighbours (iv) Estimation considering only the attribute that has missing values	Moldovan et al. [26]
Estimation of missing values (II), multiple imputation	(v) Data estimation considering overall attributes' dependency (vi) Missing values estimation using regression and other statistical methods	Hruschka et al. [27] Yuan [28]
Generation of a new data set	(vii) Generative adversarial network- (GAN-) based data generation (viii) Replacement of the data having missing values with newly generated data	Kim and Lee [23, 24] Douzas and Bacao [29]

Table 5: General learning algorithm of GAN.

Input/parameters	(i) Training data: X (ii) Learning epoch: $k1$/· Training epoch: $k2$ (iii) Step length: η (iv) Mini-batch size: m
Output	(v) Optimal parameters for G: $\widehat{\theta_G}$ (vi) Optimal parameters for D: $\widehat{\theta_D}$

Learning algorithm	for 1:$k1$
	  Initialize θ_G, θ_D
	  for 1:$k2$
	    mini-batch partitioning from X, $X' = \{x_1, \ldots, x_m\}$
	    calculate gradient for D and update θ_D
	    $\theta_D' = \theta_D + \eta \cdot (\partial f_D / \partial \theta_D)$
	    Generate random vector, $Z' = \{z_1, \ldots, z_m\}$
	    Calculate gradient for G and update θ_G
	    $\theta_G' = \theta_G + \eta \cdot (\partial f_G / \partial \theta_G)$
	  end
	end

malfunctions during train operations, 49 defects are extracted from the "arm" data in TCMS based on defect frequency and severity. Figure 3 shows the defect frequency. The records are gathered by Korean Railroad Research Institute.

Each defect's RUL is calculated using the TCMS "arm" data and relevant fault/maintenance data. The "arm" data includes the identification number, occurrence date, and other relevant information for each defect. Figure 4 shows an occurrence history of a specific defect (defect code no. 442–fault of electronic control unit (ECU)). As shown in Figure 4, the X and the Y axes indicate the occurrence date and defect code, respectively.

From the information, $\mathrm{Fault}_{i,j}(t)$ is extracted. $\mathrm{Fault}_{i,j}(t)$ indicates the jth occurrence time of the ith defect in the TCMS data. Then, inter-defect time, $\mathrm{RUL}_{i,j}(t)$, is calculated using

$$\mathrm{RUL}_{i,j}(t) = \left(\mathrm{Fault}_{i,j}(t) - \mathrm{Fault}_{i,j-1}(t)\right). \tag{3}$$

As shown in (3), $\mathrm{Fault}_{i,j}(t)$ denotes the jth sensing time of the ith defect in TCMS, and $\mathrm{RUL}_{i,j}(t)$ indicates the inter-defect time in day between the jth occurrence and (j–1)th occurrence of the ith defect. The obtained RUL is used as output data for prediction. Subsequently, the RUL is predicted using TCMS's operation data ($X(t)$) and a deep learning framework as shown in (4).

$$\mathrm{RUL}_{i,j}(t) = f_n\left(w_n \cdot \left(f_{n-1}\left(\ldots f_1\left(\sum_{i,j} w_i \cdot x_i(t) + b_i\right)\right)\right)\right. \\ \left. + b_n\right), \tag{4}$$

$$X(t) = \left(x_1(t), \ldots, x_i(t), \ldots, x_{2643}\right). \tag{5}$$

As denoted in (4) and (5), $x_i(t)$ is the value of the ith attribute at time t in the TCMS "oper" data, w_i is the weight value of $x_i(t)$, b_i is the i^{th} bias, and f_i is the i^{th} activation function.

While a general predictive maintenance estimates RUL using (4), the equation cannot be directly applied in the TCMS-based data mining owing to the missing value issue discussed in the previous section. To overcome this issue, (4) is converted by

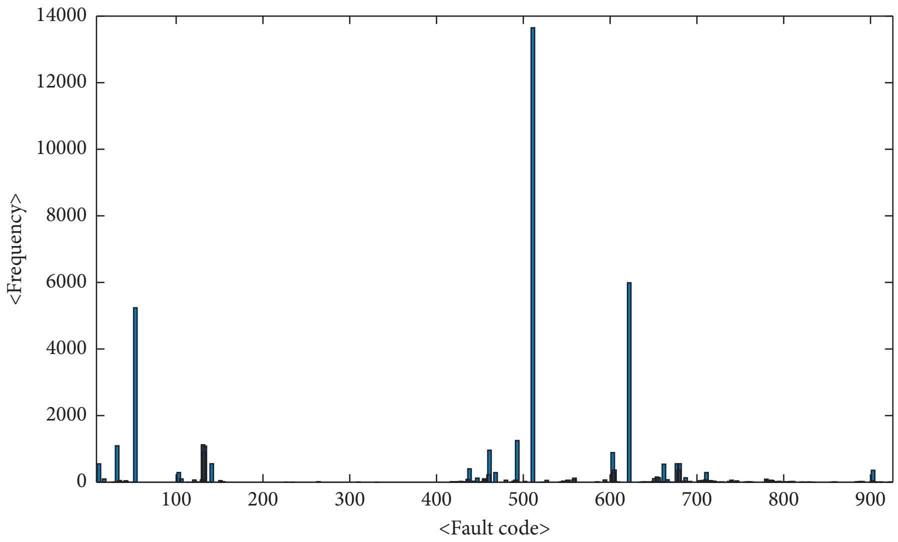


Figure 3: Defect frequency (each fault code is designated by Korean Railroad Research Institute).

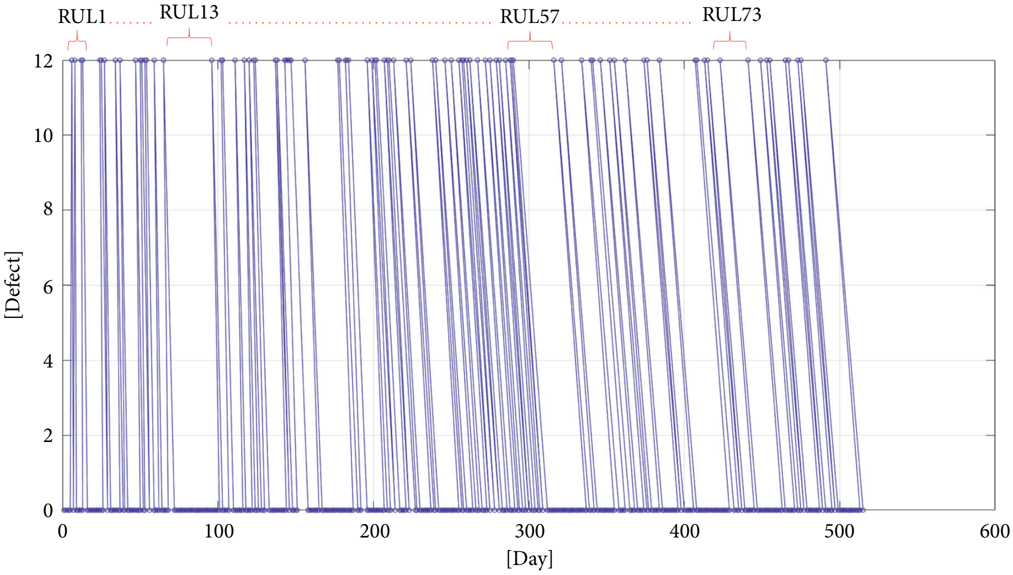


Figure 4: RUL extraction for a specific defect (code no. 442, fault of ECU) from the TCMS "arm" data.

introducing a GAN in the RUL estimation. Figure 5 shows the overall RUL prediction framework using GAN.

As shown in Figure 5, the proposed RUL prediction framework consists of two phases: learning stage and prediction stage. The main objective of the first stage is to obtain a discriminator (D) using a deep learning-based architecture. The proposed framework applies a deep neural network for the discriminator. The discriminator uses a complete TCMS data ($X'(t)$), where $X'(t)$ is generated using $X(t)$ and a generator (G) in the proposed GAN. $X'(t)$ is complete data, while $X(t)$ is a data set with missing values. As discussed in the previous section, the TCMS data ($X(t)$) of a certain train's fault may have missing values owing to various reasons. As shown in (6), these missing values are estimated initially using multivariate GMM, $p(\theta|x_i)$ where θ is the extracted RUL data, x_i is the i^{th} attribute's data over the entire time considering $x_i(t)$, and $|\text{oper}|$ is the data size of x_i.

$$p(\theta|x_i) = \sum_{i=1}^{k} \phi_i \cdot N(\mu_i, \Sigma_i), \tag{6}$$

where $N(\mu_i, \Sigma_i) = e^{-(1/2)\cdot(x_i - mu_i)^T \cdot \Sigma^{-1} \cdot (x_i - mu_i)} / \sqrt{2\pi^{|oper|} \cdot |\Sigma_i|}$.

The missing value is generated using Gibbs sampling method [30, 31]. The initial completed data ($X'(t)$) is inputted to a generator $G(\cdot)$. The output of G is the regenerated data ($X''(t)$). The generator has another deep neural network architecture similar to discriminator $D(\cdot)$ as shown in (2). Then, D generates $X''(t)$ that satisfies (1) better

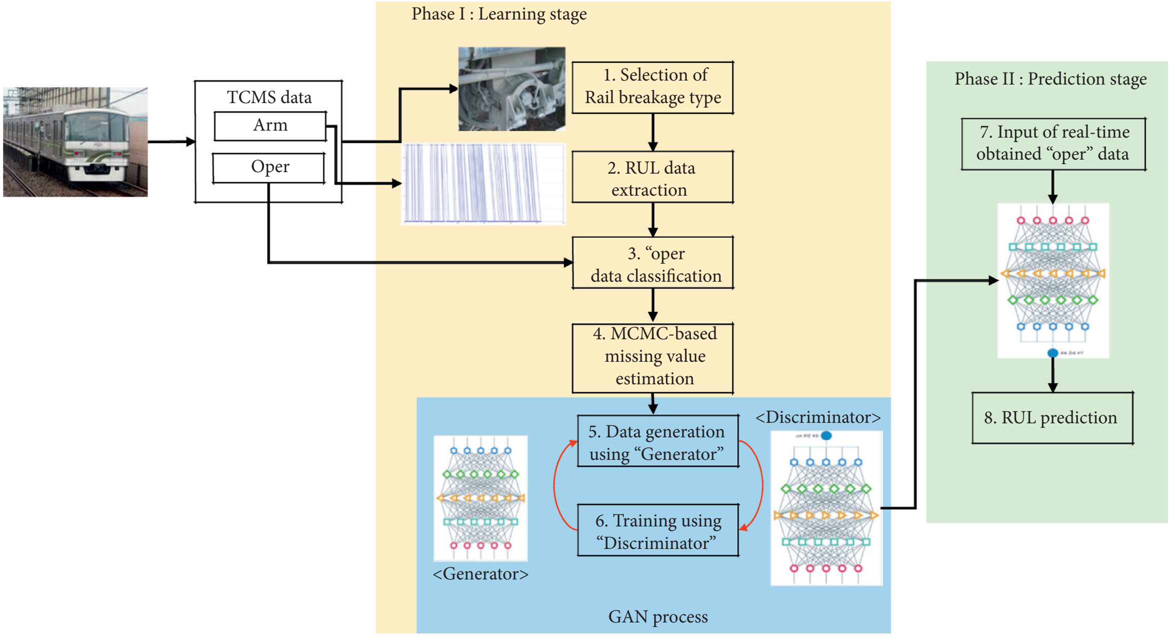

Phase I : Learning stage
TCMS data
Arm
Oper
1. Selection of Rail breakage type
2. RUL data extraction
3. "oper data classification
4. MCMC-based missing value estimation
5. Data generation using "Generator"
6. Training using "Discriminator"
<Generator>
<Discriminator>
GAN process
Phase II : Prediction stage
7. Input of real-time obtained "oper" data
8. RUL prediction

Figure 5: GAN-based missing value handling and RUL prediction framework.

than the previous estimated $X'(t)$. The updated $X''(t)$ is then inputted to $D(X''(t))$.

As presented in Figure 5, the GAN process shows these learning processes. Both network models have the deep learning parameters $\theta^{(D)}$ and $\theta^{(G)}$ as weight vectors. The updating procedures $\theta^{(D)}$ are achieved using the gradients indicated in (7)–(9).

$$\frac{\partial V}{\partial \theta^{(D)}} = \frac{\partial f_D(X''(t))}{\partial \theta^{(D)}} + \frac{\partial f_G(X'(t))}{\partial \theta^{(D)}}. \tag{7}$$

As shown in (7), V denotes $f_D(X'(t)) + f_G(X'(t))$.

$$\frac{\partial f_D(X''(t))}{\partial \theta^{(D)}} = \frac{\partial f_D(X''(t))}{\partial y_i^D} \cdot \frac{\partial y_i^D}{\partial v_i^D} \cdot \frac{\partial v_i^D}{\partial y_{i-1}^D}, \ldots, \varphi(v_1) \cdot X''(t). \tag{8}$$

v_i denotes $(w_i \cdot y_{i-1}) + b_i$, and y_i is the output of the i^{th} deep learning layer, $\phi(v_i)$.

$$\frac{\partial f_G(X'(t))}{\partial \theta^{(D)}} = \frac{\partial f_G(X'(t))}{\partial y_i^D} \cdot \frac{\partial y_i^D}{\partial v_i^D} \cdot \frac{\partial v_i^D}{\partial y_{i-1}^D} \cdot, \ldots, \cdot \varphi(v_1) \cdot X'(t). \tag{9}$$

$\theta^{(G)}$ is updated using the same procedures. Finally, the learned $D(\cdot)$ is obtained after the overall learning iterations.

In the second phase (prediction stage), the RUL of a certain defect is estimated using real-time "oper" data. While real-time data $(X(t))$ may have missing values, the RUL is estimated successfully with $D(X''(t))$. The proposed GAN-based RUL prediction framework considers data with various missing values that exist frequently in TCMS data. While TCMS may generate data with missing values due to various issues, the proposed framework is considered an effective predictive maintenance framework for handling missing values. The following section proves the effectiveness of the proposed framework using case studies and TCMS data analyses.

5. Verification and Analysis of GAN-Based RUL Prediction Framework

To prove the effectiveness of the proposed TCMS-based predictive maintenance framework, this section provides prediction performances of several train faults and compares prediction accuracies with other methods. As explained in the previous section, each fault type's RUL was estimated using its GAN-based framework.

The prediction accuracy and analyses were performed using the data of the Korean Railroad Research Institute (KRRI), which is a Korean government-funded railroad institute. The TCMS data from June, 2018, to May, 2019, in Seoul Metropolitan Subway were used as training and testing data. Sixteen defects were selected to predict their RULs. Table 6 provides the fault types and their information. The fault code ID and other relevant information were recorded in the TCMS data of the Seoul Metropolitan Subway.

To prove the verification using the provided GAN-based missing value estimation, the proposed method was compared with the other three existing methods: (1) ARIMA-based RUL estimation, (2) estimation with pruning of missing values, and (3) RUL prediction using mean-value estimation. Table 7 summaries the architecture, characteristics, and parameters of the proposed method and the three existing methods.

The GAN architecture and other relevant parameters of the proposed method are provided in Table 8. As mentioned

Table 6: Fault types and their information for RUL predictions.

No.	Fault code ID	Fault information and relevant location
1	31	LIU1 communication error in TC1
2	32	LIU2 communication error in TC1
3	34	LIU2 communication error in TC0
4	38	LIU1 hardware malfunction in TC0
5	39	LIU2 hardware malfunction in TC
6	231	SIV inverter malfunction
7	434	Break malfunction
8	442	ECU malfunction
9	635	TC MFB card/ATC vital malfunction
10	636	Tachometer error
11	640	Main ATC hardware malfunction
12	641	Secondary ATC hardware malfunction
13	647	FSB/ATC error
14	669	ATO-ATC communication error
15	670	ATO-ATC 1 communication error
16	684	ATC DBAU hardware error

Table 7: Four RUL prediction methods.

RUL prediction method	GAN-based RUL estimation (the proposed method)	ARIMA-based RUL estimation	RUL estimation using "missing value pruning"	RUL estimation with "mean-value estimation" of missing values
Missing value handling mechanism	*O* (GAN-based data generation)	*X* (removal of records with missing values)	*X* (removal of records with missing values)	O (mean-value estimation)
RUL estimation method	Classification using GAN	ARIMA-based RUL estimation	Deep neural network	Deep neural network
Detailed parameters	Refer to Table 8	ARIMA (6, 2, 5)	(i) Learning epoch: 1000 (ii) Learning rate: 10^{-3} (iii) Number of layers:10 (iv) Used activation functions =(leaky RU for final layer, Sigmoid for layers #1–#9)	—

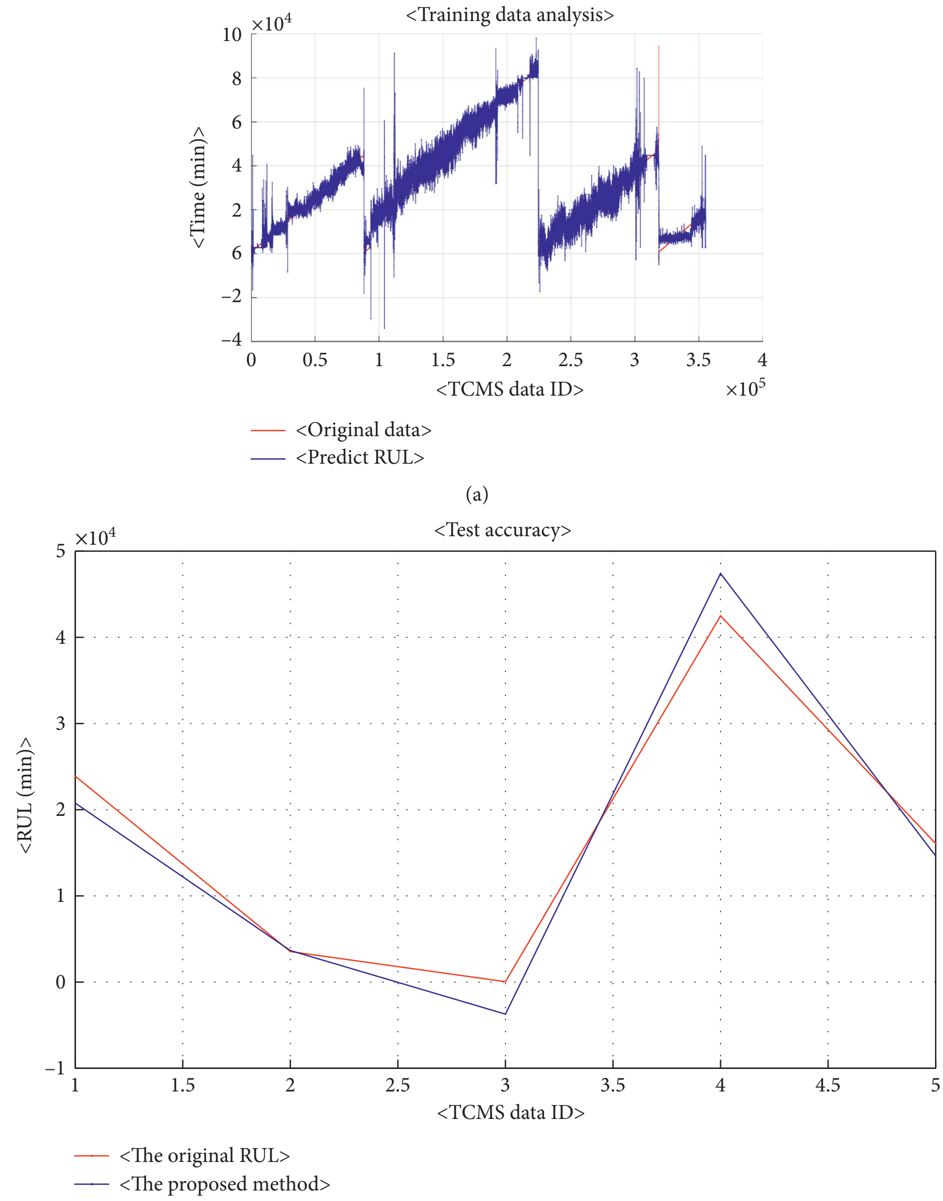
Table 8: Detailed architecture and relevant parameters of the GAN-based RUL estimation (case for fault code ID 31).

Classification	Detailed architectures
General learning parameters	(i) Learning epoch: 1000 (ii) Learning rate: 10^{-3}
Discriminator $(D(\cdot))$	(i) Number of Layers:10 (ii) Number of nodes in each layer =(1, 50, 100, 200, 500, 1500, 2500, 3000, 3500, 2653) (iii) Used activation functions =(leaky RU for final layer, Sigmoid for layers #1–#9)
Generator $(G(\cdot))$	(i) Number of layers: 7 (ii) Number of nodes in each layer =(2653, 2700, 2800, 3000, 3200, 3500, 2653) (iii) Used activation functions: Sigmoid function for each layer

in the previous section, the GAN architecture varies for every defect. The parameters are provided in Table 8.

The parameters shown in Tables 7 and 8 were determined by applying numerical tests on the provided TCMS data. Figure 6 shows the training and test accuracies of the proposed RUL prediction for fault code ID 31 (LIU1 communication error in TC1).

As shown in Figure 6, the proposed method performed 99.9% and 83.5% for the training and test accuracies, respectively. The accuracy was calculated using (10) and (11). The root mean squared error $(\mathrm{RMSE}(\mathrm{RUL})_i)$ for a certain type (i) of RUL (RUL_i) was used as a test metric, where $\widehat{\mathrm{RUL}_{i,j}}$ is the j^{th} predicted value using the proposed framework and $\mathrm{RUL}_{i,j}$ is the j^{th} original RUL value, and n is the size of the test data.



(a)

(b)

Figure 6: The training and the test accuracies of the proposed RUL prediction (for fault code ID 31). (a) The training accuracy. (b) The test accuracy.

$$\mathrm{RMSE}\left(\mathrm{RUL}_i\right)=\sqrt{\frac{\sum_{j=1}^{n}\left(\widehat{\mathrm{RUL}_{i,j}}-\mathrm{RUL}_{i,j}\right)^2}{n}}, \quad (10)$$

$$\text{Accuracy }(\%)=\left[1-\mathrm{RMSE}\left(\mathrm{RUL}_i\right)\right]\cdot 100. \quad (11)$$

The test data was sampled from the original TCMS data. As the fault occurrence was very low, the amount of test data was limited. The data-based numerical tests were carried out by comparison with the other methods. Figure 7 shows the test accuracies using the four methods: the proposed method and the other three benchmarking methods shown in Table 7.

As shown in Figure 7, the proposed method had the highest accuracy compared with the other existing methods. Table 9 provides the test accuracy for each method.

The numerical analysis indicates that the missing value issue was critical for the fault prediction using the TCMS data. In addition, the estimation of the missing values strongly influenced the RUL predictions. Using the proposed framework, the RUL prediction system for the 16 train faults was developed as shown in Figure 8. The software program was implemented using MFC© and MATLAB© on Windows 10©. Figure 9 shows the test accuracies of the various TCMS faults.

As shown in Figure 9, the proposed framework and its implemented software resulted in RUL predictions of over 82.07% for all TCMS fault types. Table 10 shows the test

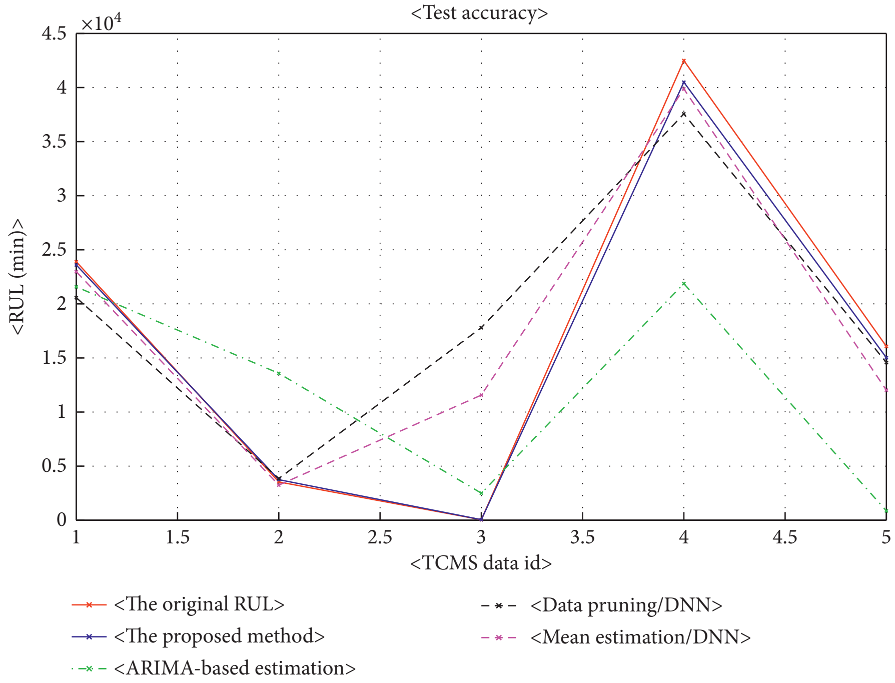
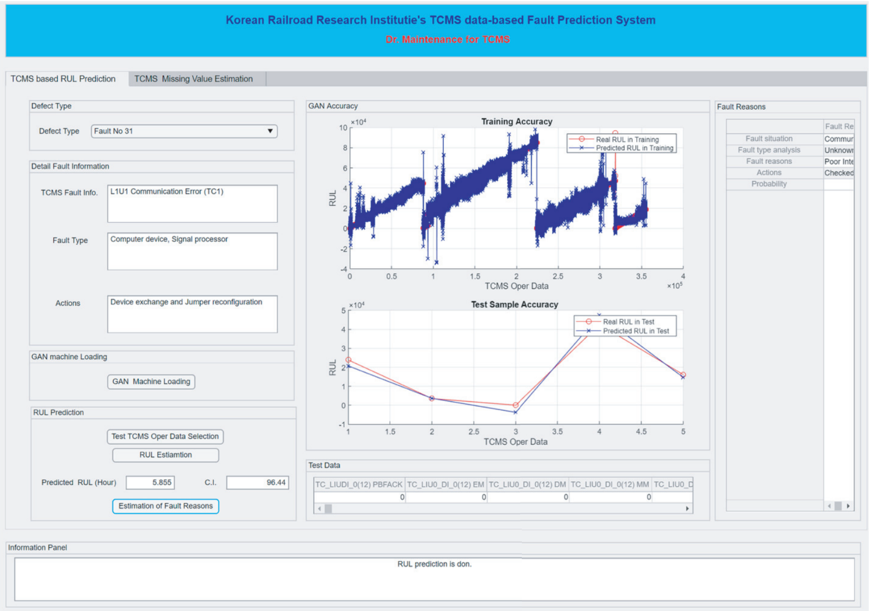


Figure 7: Test accuracies using the four benchmarking methods.

Table 9: Test accuracy for each RUL estimation method (case for fault code ID 31).

RUL prediction method	GAN-based RUL estimation (the proposed method)	ARIMA-based RUL estimation (%)	RUL estimation using "missing value pruning" (%)	RUL estimation using "mean-value estimation" of missing values (%)
Test accuracy	83.5%	56.7	69.7	76.3



Defect Type: Fault No 31

TCMS Fault Info.: L1U1 Communication Error (TC1)

Fault Type: Computer device, Signal processor

Actions: Device exchange and Jumper reconfiguration

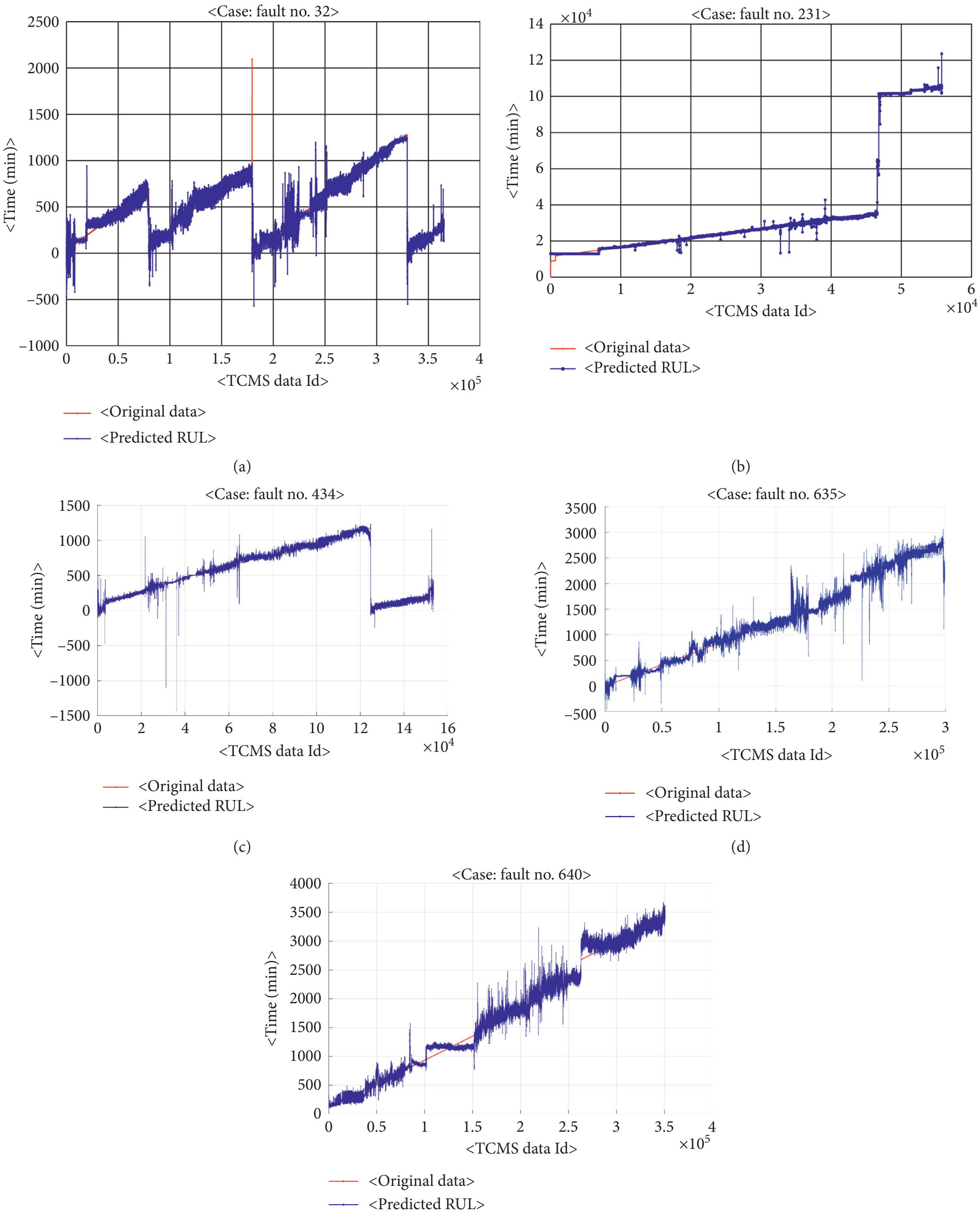
Predicted RUL (Hour): 5.855 C.I.: 96.44

TC_LIUDI_0(12) PBFACK	TC_LIU0_DI_0(12) EM	TC_LIU0_DI_0(12) DM	TC_LIU0_DI_0(12) MM	TC_LIU0_D
0	0	0	0	

	Fault Re
Fault situation	Commur
Fault type analysis	Unknowr
Fault reasons	Poor Inte
Actions	Checked
Probability	

RUL prediction is don.

Figure 8: The implementation of GAN-based missing value handling and RUL prediction framework.



(a)

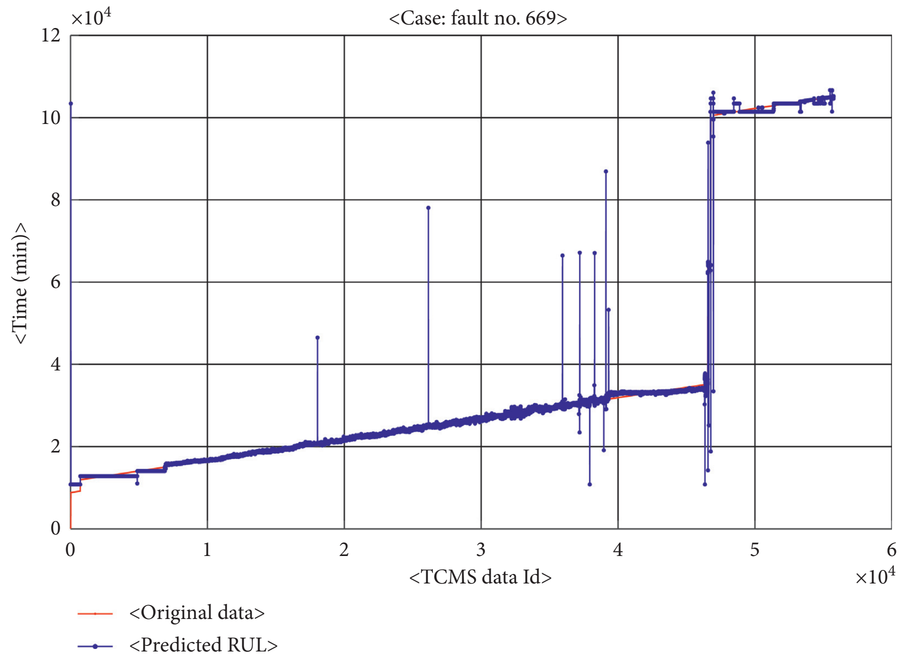
(b)

(c)

(d)

(e)

Figure 9: Continued.



(f)

Figure 9: RUL prediction results using the proposed method. (a) Fault code ID 34 (LIU2 communication error in TC0). (b) Fault code ID 231 (SIV inverter malfunction). (c) Fault code ID 434 (break malfunction). (d) Fault code ID 635 (TC MFB malfunction). (e) Fault code ID 640 (main ATC hardware malfunction LIU2). (f) Fault code ID 669 (ATO-ATC communication error).

Table 10: Test accuracy for each defect using the proposed framework.

Fault code id.	Test accuracy	Fault code ID	Test accuracy	Fault code ID	Test accuracy	Fault code ID	Test accuracy (%)
31	83.50	32	88.08	34	88.94	38	82.47
39	86.98	231	99.91	434	96.75	442	95.68
635	85.24	636	85.60	640	92.21	641	89.15
647	93.14	669	97.56	670	82.07	684	80.46

accuracy of the proposed framework for each defect in the TCMS data.

6. Conclusions and Further Study

The transportation maintenance has gained substantial attention owing to its significance in societies. This study focused on the RUL prediction-based railway maintenance framework. While initial railway maintenance concentrated on periodic maintenance framework such as predefined time-based maintenance or distance-based scheduling, railway maintenance has evolved to condition-based maintenance owing to the advancement in monitoring devices and information technologies. This framework detects abnormal status using state-of-the-art sensors in a train system. The monitored signals are transferred to a server, TCMS.

This study proposes a new and effective RUL prediction framework using TCMS data. In general, TCMS data are classified into operation data and alarm data. To predict the remaining life of a certain fault or malfunction, this paper selected 16 faults based on their significance and severity. A deep learning-based mechanism was developed for each fault. Firstly, RUL of the target train fault was extracted using the TCMS alarm data. Then, the data was used as the predicted output of the proposed deep neural network. However, the system has a critical issue, which is common in most sensor-based systems: the existence of missing values. Existence of missing values in TCMS data could be due to various reasons such as sensor malfunction and low life of monitoring modules. Among several estimation methods for replacing missing values, this study used a GAN model to estimate missing values and predict RULs, simultaneously. The developed GAN framework can generate new data that cover missing values using the prediction objectives. Initially, the missing values are estimated using GMM and the estimated data are refined with the proposed GAN framework. In addition, the discriminator in the GAN model has better predictive performances in generating more accurate data. The effectiveness of the proposed maintenance framework was investigated by comparing it with other existing methods. The proposed framework is a new and effective train predictive maintenance framework that addresses missing value issue and predicts fault detection in real time.

For further studies, various optimization and metaheuristics methods can be applied to the proposed framework. As TCMS data is classified as big data, its learning could take longer time. Moreover, the framework requires comparatively higher computational burden. While the proposed framework uses two deep neural network models for a generator and a discriminator in its GAN module, it is expected that application of several optimization methods could reduce learning time and computation burden. In addition, prediction of each train fault requires a different GAN-based framework. While it has an advantage focusing on the defined fault, real-time prediction may require significant computational burden. To resolve this issue, a new architecture for multiple-fault prediction will be considered in future studies.

Data Availability

The used data is supported by Korea Railroad Research Institute.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by The Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Republic of Korea (grant number: NRF-2018R1D1A3B07047113), and by a research grant from R&D Program of the Korea Railroad Research Institute (KRRI), Republic of Korea.

References

[1] T. Lidén, “Railway infrastructure maintenance - a survey of planning problems and conducted research,” *Transportation Research Procedia*, vol. 10, pp. 574–583, 2015.

[2] P. Fraga-Lamas, T. M. Fernandez-Carames, and L. Castedo, “Towards the internet of smart trains: a review on industrial IoT-Connected Railways,” *Sensors*, vol. 17, no. 9, pp. 1–44, 2017.

[3] H. Zhao, Z. Huang, and Y. Mei, “High-speed EMU TCMS design and LCC technology research,” *Engineering*, vol. 3, no. 1, pp. 122–129, 2017.

[4] J. Han and C. Kim, “A conceptual design of maintenance information system interface for real-time diagnosis of driverless EMU,” *Journal of the Korea Academia-Industrial Cooperation Society*, vol. 18, no. 10, pp. 63–68, 2017.

[5] K. Kim, K. Lee, and S. An, “An analysis about consumed energy of electric multiple unit used TCMS data on the condition of safety driving,” *Journal of the Korean Society of Safety*, vol. 27, no. 6, pp. 31–42, 2012.

[6] S. Ito, T. Suzuki, K. Suzuki, and K. Suzuki, “Train control and management system technologies for improving safety and maintainability,” *Hitachi Review*, vol. 67, no. 7, pp. 52–58, 2018.

[7] K. Sharma, S. Maheshwari, R. Solanki, and V. Khanna, “Railway track breakage detection method using vibration estimation sensor network,” in *Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics*, pp. 2355–2362, New Delhi, India, September 2014.

[8] R. Sireesha, K. B. Ajay, G. Mallikarjunaiah, and K. B. Bharath, “Broken rail detection system using RF technology,” *Proceedings of SSRG International Journal of Electronics and Communication Engineering*, vol. 2, no. 4, pp. 11–15, 2015.

[9] H. Lee, “Framework and development of fault detection classification using IoT device and cloud environment,” *Journal of Manufacturing Systems*, vol. 43, no. 2, pp. 257–270, 2017.

[10] H. Lee, “Effective dynamic controls strategy of key supplier with multiple downstream manufacturers using industrial Internet of Things and cloud system,” *Processes*, vol. 7, no. 3, pp. 1–18, 2019.

[11] Use IoT to advance railway predictive maintenance. (https://www.hitachivantara.com 2020).

[12] F. Corman, S. Kraijema, M. Godjevac, and G. Lodewijks, “Optimizing preventive maintenance policy: a data-driven application for a light rail braking system,” *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol. 231, no. 5, pp. 534–545, 2017.

[13] Mckinsey, *Using Analytics to Get European Rail Maintenance on Track*, McKinsey & Company, New York, NY, USA, 2020, https://www.mckinsey.com/.

[14] R. B. Faiz and S. Singh, “Time based predictive maintenance management of UK rail track,” in *Proceedings of the 2009 International Conference on Computing, Engineering and Information*, Fullerton, CA, USA, April 2019.

[15] K. Shaikh, I. H. Kalwar, B. S. Chowdhry, K. Kazi, and B. A. Arain, “Modeling and simulation of predictive maintenance scheme for high speed railway vehicles,” *Indian Journal of Science and Technology*, vol. 9, no. 1, pp. 1–6, 2016.

[16] C. Letot, P. Dersin, M. Pugnaloni et al., “A data driven degradation-based model for the maintenance of turnouts: a case study,” *IFAC-PapersOnLine*, vol. 48, no. 21, pp. 958–963, 2015.

[17] V. Atamuradov, K. Medjaher, P. Dersin, B. Lamoureux, and N. Zerhouni, “Prognostics and health management for maintenance practitioners–review, implementation and tools evaluation,” *International Journal of Prognostics and Health Management*, vol. 8, pp. 1–31, 2017.

[18] T. Liden, “Railway infrastructure maintenance–a survey of planning problems and conducted research,” *Transportation Research Procedia*, vol. 10, pp. 574–583, 2018.

[19] G. Neil, “On board train control and monitoring systems,” in *Proceedings of the 9th Institution of Engineering and Technology Professional Development Course on Electric Traction Systems*, Manchester, UK, November 2006.

[20] Y. Xu, Q. Qiao, R. Wu, and Z. Zhou, “Advanced maintenance cycle optimization of urban transit vehicle,” *Advances in Mechanical Engineering*, vol. 11, no. 2, pp. 1–7, 2019.

[21] 2020 Innovative monitoring and predictive maintenance solutions on lightweight wagon, http://newrail.org/innowag/wp-content/uploads/2017/12/INNOWAG_D1.1_Benchmark-market-drivers.pdf.

[22] E. Oh and H. Lee, “An imbalanced data handling framework for industrial big data using Gaussian Process Regression-based Generative Adversarial Network,” *Symmetry*, vol. 12, no. 4, pp. 1–19, 2020.

[23] H. Kim and H. Lee, “Fault detect and classification framework for semiconductor manufacturing processes using missing data estimation and Generative Adversary Network,” *Journal*

of Korean Institute of Intelligent Systems, vol. 28, no. 4, pp. 393–400, 2018.

[24] H. Kim and H. Lee, "Generative adversarial networks based data generation framework for overcoming imbalanced manufacturing process data," *Journal of Korean Institute of Intelligent Systems*, vol. 29, no. 1, pp. 1–8, 2019.

[25] S. Munirathinam and B. Ramadoss, "Predictive models for equipment fault detection in the semiconductor manufacturing process," *International Journal of Engineering and Technology*, vol. 8, no. 4, pp. 273–285, 2016.

[26] D. Moldovan, T. Cioara, I. Anghel, and I. Salomie, "Machine learning for sensor-based manufacturing process," in *Proceedings of the 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 147–154, Cluj-Napoca, Romania, September 2017.

[27] E. R. Hruschka, E. R. Hruschka, and N. E. F. Ebecken, "Missing values imputation for a clustering genetic algorithm," *Lecture Notes in Computer Science*, vol. 3612, pp. 245–254, 2005.

[28] Y. C. Yuan, "Multiple imputation for missing data: concepts and new development," SAS Institute Inc, Cary, NC, USA, 2019, http://support.sas.com/rnd/app/stat.

[29] G. Douzas and F. Bacao, "Effective data generation for imbalanced learning using conditional generative adversarial networks," *Expert Systems with Applications*, vol. 91, pp. 464–471, 2018.

[30] S. Choo and H. Lee, "Learning framework of multimodal Gaussian-Bernoulli RBM handling real-value input data," *Neurocomputing*, vol. 275, no. 1, pp. 1813–1822, 2018.

[31] G. Casella and R. L. Berger, *Statistical Inference*, Cengage Learning, Boston, MA, USA, 2nd edition, 2002.