symmetry

MDPI

Article

An Imbalanced Data Handling Framework for Industrial Big Data Using a Gaussian Process Regression-Based Generative Adversarial Network

Eunseo Oh and Hyunsoo Lee *

School of Industrial Engineering, Kumoh National Institute of Technology, Gumi 39177, Korea; chss014@kumoh.ac.kr

* Correspondence: hsl@kumoh.ac.kr; Tel.: +82-54-478-7661

Received: 20 March 2020; Accepted: 20 April 2020; Published: 23 April 2020

Abstract: The developments in the fields of industrial Internet of Things (IIoT) and big data technologies have made it possible to collect a lot of meaningful industrial process and quality-based data. The gathered data are analyzed using contemporary statistical methods and machine learning techniques. Then, the extracted knowledge can be used for predictive maintenance or prognostic health management. However, it is difficult to gather complete data due to several issues in IIoT, such as devices breaking down, running out of battery, or undergoing scheduled maintenance. Data with missing values are often ignored, as they may contain insufficient information from which to draw conclusions. In order to overcome these issues, we propose a novel, effective missing data handling mechanism for the concepts of symmetry principles. While other existing methods only attempt to estimate missing parts, the proposed method generates a whole set of data set using Gaussian process regression and a generative adversarial network. In order to prove the effectiveness of the proposed framework, we examine a real-world, industrial case involving an air pressure system (APS), where we use the proposed method to make quality predictions and compare the results with existing state-of-the-art estimation methods.

Keywords: missing data generation; industrial big data; air pressure system; generative adversarial network; Gaussian process regression

1. Introduction

Failure analysis and process predictions for manufacturing processes have significant impacts on improving product quality and process reliability. A number of studies on failure cause analysis and classification using machine learning or deep learning methods are actively been conducted, which will lead to improvements in product quality and reliability. For instance, a failure of the brakes in a vehicle may lead to a significant accident, so it is important to predict any potential failures. In heavy vehicles such as trucks excavators, the performance of the brakes is very important due to the heavy weight they must bring to a halt. The brakes of a truck are driven by an air pressure system, and therefore require pressurized air to operate. If the air pressure in the system does not reach a certain level, its brakes may not work precisely and then serious accidents can occur. This research focuses on quality prediction of an air compressor and pressure system found in vehicles. The air pressure system (APS), an essential part of the vehicle, stops or decelerates the vehicle by applying compressed air to the brakes. Figure 1 depicts the APS structure.

As shown in Figure 1, the air compressor compresses air that is initially at atmospheric pressure and maintains the air at optimum pressure through the air compressor governor. The compressed air moves to the air reservoir. When the driver presses the brake pedal, the brake valve closes and compressed air stored in the air reservoir is used to brake by applying pressure to the brake chamber through the line.

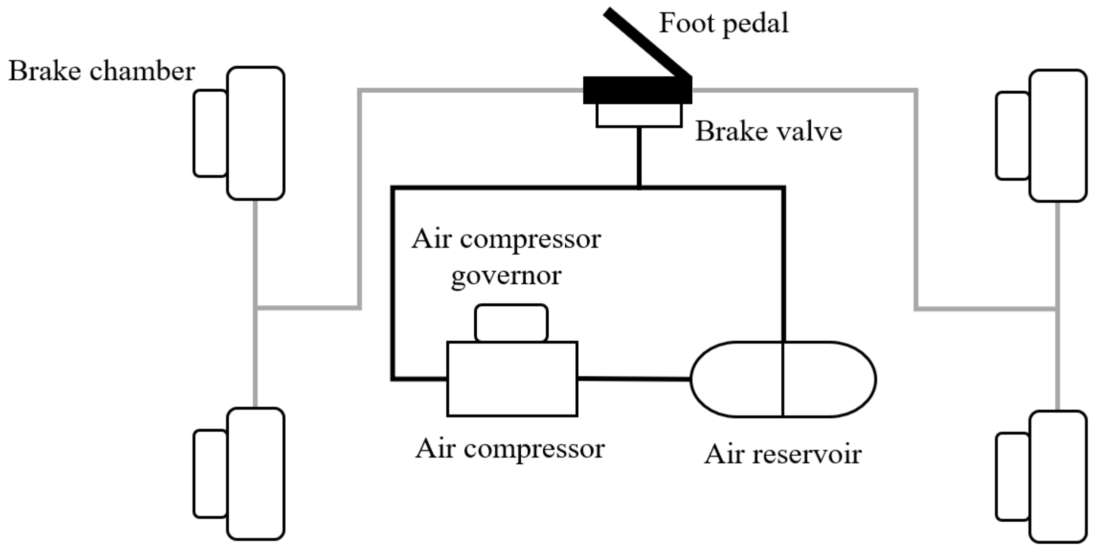
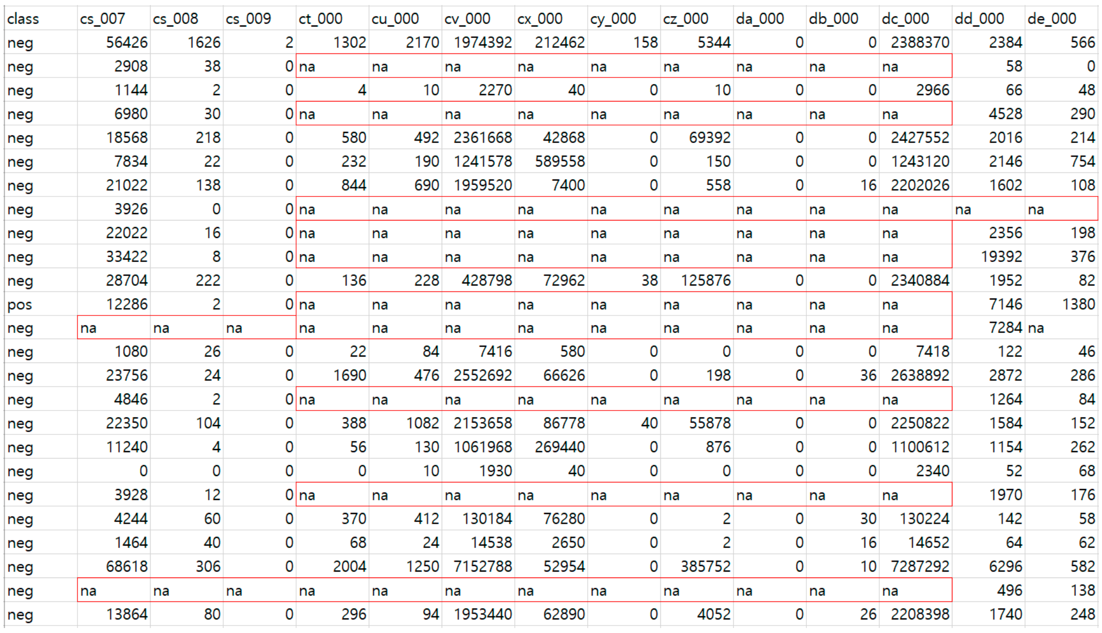


Figure 1. Air pressure system (APS) structure.

It is important to predict defects and malfunctions in the APS. Well considered analysis of data leads to reliable predictions of potential APS failures and contributes to the minimization of dangers, as well as reduces maintenance costs. In this study, the data set for the failure analysis is from APS operations and the relevant quality data is from the Scania trucks data set [1]. This data set consists of the operating sensor attributes from a broken Scania truck, in which the attributes are anonymized. In order to investigate the causes of any APS failure, these attributes and their data are analyzed using several machine learning and statistical methods.

However, the data have a number of missing values due to sensor failures. Missing data can mean that there is insufficient information for analysis by existing methods and can make it difficult to identify the actual cause of any failure. In addition, missing values cause disproportionately distributed data, and for models trained on properly proportioned data this reduces the accuracy of the classification performance. Figure 2 shows the issue of missing values in a real APS failure incident in the Scania trucks data set. Missing values are marked as not available (na). The data are used for the failure study of a vehicle. The positive class (pos) of the data set indicates that the failure is in the APS, and the negative class (neg) indicates that the failure is not related to the APS.

In APS failure analysis, the classification performance depends heavily on the data's completeness level. The more missing values in the data, the lower the completeness level is. Unfortunately, the Scania APS data contains a number of missing values. In addition, these missing values may cause a lack of data for analysis. Therefore, it is important to have a framework that handles incomplete data. Missing values and data imbalances are a primary cause for less accurate quality prediction. In addition, several kinds of multivariate statistical analysis cannot be applied to this data set due to the issues of missing values. This situation makes missing value estimations the most important preprocessing steps. A number of existing studies [2–13] have proposed methods to overcome this data imbalance issue. Table 1 summarizes several existing missing value estimation methods and their applications.



class	cs_007	cs_008	cs_009	ct_000	cu_000	cv_000	cx_000	cy_000	cz_000	da_000	db_000	dc_000	dd_000	de_000
neg	56426	1626	2	1302	2170	1974392	212462	158	5344	0	0	2388370	2384	566
neg	2908	38	0	na	na	na	na	na	na	na	na	na	58	0
neg	1144	2	0	4	10	2270	40	0	10	0	0	2966	66	48
neg	6980	30	0	na	na	na	na	na	na	na	na	na	4528	290
neg	18568	218	0	580	492	2361668	42868	0	69392	0	0	2427552	2016	214
neg	7834	22	0	232	190	1241578	589558	0	150	0	0	1243120	2146	754
neg	21022	138	0	844	690	1959520	7400	0	558	0	16	2202026	1602	108
neg	3926	0	0	na	na	na	na	na	na	na	na	na	na	na
neg	22022	16	0	na	na	na	na	na	na	na	na	na	2356	198
neg	33422	8	0	na	na	na	na	na	na	na	na	na	19392	376
neg	28704	222	0	136	228	428798	72962	38	125876	0	0	2340884	1952	82
pos	12286	2	0	na	na	na	na	na	na	na	na	na	7146	1380
neg	na	na	na	na	na	na	na	na	na	na	na	na	7284	na
neg	1080	26	0	22	84	7416	580	0	0	0	0	7418	122	46
neg	23756	24	0	1690	476	2552692	66626	0	198	0	36	2638892	2872	286
neg	4846	2	0	na	na	na	na	na	na	na	na	na	1264	84
neg	22350	104	0	388	1082	2153658	86778	40	55878	0	0	2250822	1584	152
neg	11240	4	0	56	130	1061968	269440	0	876	0	0	1100612	1154	262
neg	0	0	0	0	10	1930	40	0	0	0	0	2340	52	68
neg	3928	12	0	na	na	na	na	na	na	na	na	na	1970	176
neg	4244	60	0	370	412	130184	76280	0	2	0	30	130224	142	58
neg	1464	40	0	68	24	14538	2650	0	2	0	16	14652	64	62
neg	68618	306	0	2004	1250	7152788	52954	0	385752	0	10	7287292	6296	582
neg	na	na	na	na	na	na	na	na	na	na	na	na	496	138
neg	13864	80	0	296	94	1953440	62890	0	4052	0	26	2208398	1740	248

Figure 2. Issues from missing values in APS data for Scania trucks.

Table 1. Existing research studies and applications for estimating missing data.

Missing Value Handling Method	Research Studies and Applications	Main Estimation Techniques and Frameworks
Imputation	Paul [2]	Multiple imputation (MI)-based missing value estimation
	Dempster, Laird, and Rubin [3]	Probability modeling maximum likelihood estimation (MLE)-based estimation Expectation maximization-based estimation [4]
	Hastie et al. [5], Troyanskaya et al. [6]	Singular value decomposition (SVD) and K-nearest neighbor (KNN)-based missing data imputation
	Zhang [7]	Regression-based imputation
	Gondara and Wang [8]	Deep denoising autoencoder-based imputation
	Gemmeke et al. [9]	Missing data imputation using sparse imputation-based compressive sensing (CS)
Estimation	Oba et al. [10]	Bayesian network-based preprocessing Gene profiling expression
	Little and Rubin [11]	Least squares-based missing data analysis
	Gondek, Hafner, and Sampson [12]	Missing value imputation using random forest and feature engineering
	Perepu and Tangirala [13]	Missing value estimation using a CS method with adaptive dictionary
	Chodosh, Wang, and Lucey [14]	Estimating a dense depth map using a CS method and alternating direction neural networks

Most research methods resort to data imputation, where a data set with missing values is ignored. However, these kinds of methods cause distortion of the captured data-based probability distribution. In addition, the variance of the data that is used becomes heavily distorted. In order to overcome the issue, this paper proposes a novel, effective framework to generate a complete data set using a generative adversarial network (GAN) and Gaussian processes regression (GPR). The proposed framework is based on the symmetry principles, which the original data and the generated data have. Scania APS data are used as exemplary test data and the proposed method is compared with other existing methods.

Section 2 examines the background of the key techniques we use (GPR and GAN) and reviews the relevant literature. Section 3 proposes an overall framework for generating a data set using GPR and GAN. Section 4 shows the proposed method's effectiveness with experimental analysis of the proposed framework and provides comparisons with other classification models using the APS data.

2. Background Knowledge and Literature Reviews

2.1. Gaussian Processes Regression

GPR [15–19] is a Bayesian algorithm and has the ability to provide a statistical uncertainty measure. Since it can provide high uncertainty prediction measurements in changing environments, the GPR algorithm has been applied in various research fields, as shown in Table 2.

Table 2. Research studies and applications using Gaussian processes regression (GPR).

Research Studies Using GPR	Application Areas
Jochem et al. [20]	Automated spectral band analysis
Ak et al. [21]	The time and space prediction of an infectious diseases
Luttinen and Ilin [22]	Sea level temperature reconstruction using GPR
Nguyen and Peters [23]	Kinetics model estimation
Nguyen, Hu and Spanos [24]	Efficient building field formation using an estimation of indoor environment fields
Chen et al. [25]	Wind prediction for energy efficiency
Oh and Lee [26]	Estimation of pheromone values based on ant colony optimization

Figure 3 depicts the general process of GPR. The latent variable f_i is derived from the input value X_i with the observed value Y_i. The distribution of the test observation value Y_* is estimated using the Gaussian field f_* for the input value X_*.

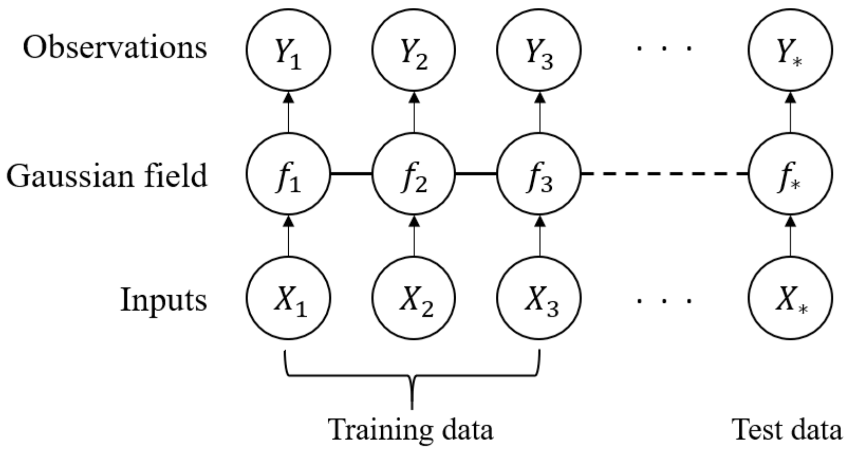


Figure 3. Graphical model of Gaussian process regression.

The training data set is $\left\{\left(x^{(i)}, y^{(i)}\right)\right\}_{i=1}^{l}$, while Equations (1) and (2) summarize the general GPR model.

$$y = f + \epsilon \tag{1}$$

where, $\epsilon \sim \mathrm{N}\left(0, \sigma_{\mathrm{y}}^{2}\mathrm{I}\right)$

$$f \sim \mathrm{GP}(\mathrm{m}(x), \mathrm{k}(x, x_{*})) \tag{2}$$

where $\mathrm{m}(x) = \mathrm{E}[f(x)]$, and $\mathrm{k}(x, x_{*}) = \mathrm{E}[(f(x) - \mathrm{m}(x))(f(x_{*}) - \mathrm{m}(x_{*}))]$.

In Equation (1), ϵ is a noise parameter that follows a Gaussian distribution, with variance of σ_{y}^{2} and a mean of 0. Here, I is an identity matrix constructed according to the data's dimensions, f is the transformation relation-based value for the equivalent input vector X, and Y is the observed output vector. Equation (2) is the "distribution over functions" [19] used to derive the test target value Y using the Gaussian model. The distribution consists of a mean $\mathrm{m}(x)$ and variance covariance $\mathrm{k}(x, x_{*})$ derived by sampling from a multivariate Gaussian distribution. The covariance function $\mathrm{k}(x, x_{*})$ is commonly parameterized by a kernel parameter and models the dependence between existing observed input points x and new test input points to predict x_{*}. Equation (3) is a radial basis kernel function that calculates a similarity measure between both data instances. The kernel is a closeness measure of data points. It is not just used to model the dependence of observed and unobserved points, but all points.

$$\mathrm{k}(x, x_{*}) = \exp(\gamma \cdot |x - x_{*}|_{2}) \tag{3}$$

where γ is a hyper parameter of the kernel function.

The parameter γ is related with the variance of a data set. After the mean function and kernel types are selected, the Gaussian process produce predictions are made based on previous observations. However, any actual data obtained from measurements include a lot of noise in general. Therefore, an observed output vector Y with its noise is expressed as in Equation (4).

$$Y \sim \mathrm{N}\left(0, \mathrm{k} + \sigma_{y}^{2}\mathrm{I}\right) \tag{4}$$

In Equation (4), the observed output vector Y with a mean 0 and covariance $\mathrm{k} + \sigma_{y}^{2}\mathrm{I}$ follows the properties of multivariate Gaussian distribution and is used as a prediction model. Equation (5) denotes the prior distribution for $f_{*}(X_{*})$ with the noise condition, $f_{*}(X_{*})$ is a function that outputs a predicted vector Y_{*} for a vector X_{*}, which has a new input point x_{*}. It is a prior distribution model that considers the noise generated when the test data vector X_{*} is input and the output data are predicted through the function value $f_{*}(X_{*})$.

$$\begin{bmatrix} Y \\ f_{*} \end{bmatrix} \sim \mathrm{N}\left(0, \begin{bmatrix} \mathrm{k}(X, X) + \sigma_{y}^{2}\mathrm{I} & \mathrm{k}(X, X_{*}) \\ \mathrm{k}(X_{*}, X) & \mathrm{k}(X_{*}, X_{*}) \end{bmatrix}\right) \tag{5}$$

The maximum likelihood estimator (MLE) and variance of the predicted distribution derived using the newly updated Gaussian process is expressed as seen in Equations (6) and (7).

$$m(x) = \mathrm{k}(x, X)(\mathrm{k}(X, X) + \sigma_{y}^{2}\mathrm{I})^{-1}Y \tag{6}$$

$$\mathrm{k}(x, x_{*}) = \mathrm{k}(x, x_{*}) - \mathrm{k}(\mathrm{x}, X)(\mathrm{k}(X, X) + \sigma_{y}^{2}\mathrm{I})^{-1}\mathrm{k}(X, x_{*}) \tag{7}$$

In this study, the introduced GPR framework is used to estimate missing values in our real-world imbalanced data set. The GPR process helps produce more accurate modeling from the data. The following section explains the GAN method, which generates a new data set using the GPR-based missing value estimations.

2.2. Generative Adversarial Network

GAN [27] is a generative model that uses the neural network architecture. A general GAN framework includes two types of model, both of which are neural networks: a generator (G) and a discriminator (D). Both models generate data that become closer to the real data set by competing with each other. GANs have been used in many research fields, as shown in Table 3.

Table 3. Research studies and applications using a generative adversarial network (GAN).

Research Studies Using GAN	Application and Characteristics
Kim and Lee [28]	Missing data generation of semiconductor manufacturing processes data method: Oversample → GAN based data generation
Yoon, Jordon, and Schaar [29]	Missing data imputation of breast cancer, spam, letter recognition, credit, news data GAN-based hint generation
Kim and Lee [30]	Missing data generation of steel Plates faults data Estimate the missing value by adding missing term based on the GAN
Shang et al. [31]	Image generation GAN-based missing view imputation
Mao et al. [32]	Image generation Least squares loss function-based discriminator in a GAN
Zhao, Mathieu, and Le Cun [33]	Image generation Energy value allocation according to data density-based a discriminator in a GAN
Li et al. [34]	Object detection GAN-based high-quality image generation

As shown in Figure 4, the generator G takes a vector z extracted from random noise as its input and attempts to generate data which is close to the real data. The discriminator D learns how to distinguish between real data and the generated fake data. During training, while this process is being repeated, the generator minimizes the probability that the discriminator can distinguish real from generated data, and the discriminator maximizes the probability of distinguishing real data from the generated data.

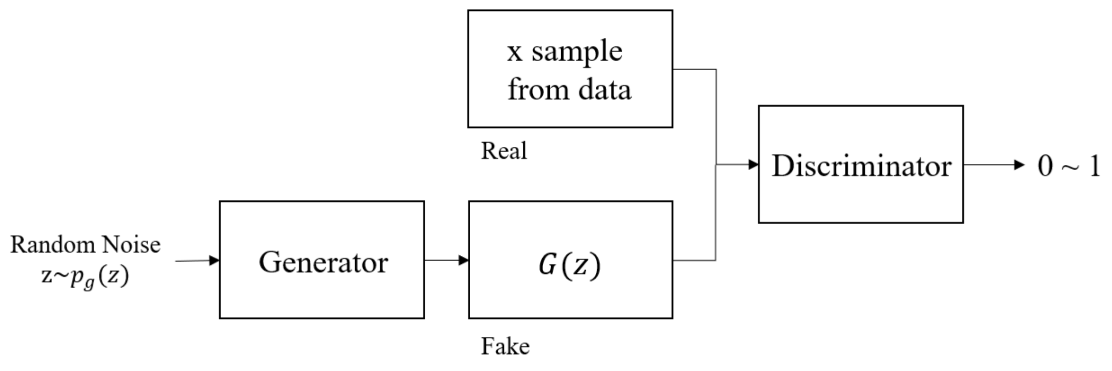


Figure 4. General architecture of generative adversarial network.

As shown in Figure 4, the sample data extracted from the real data are represented by x and the distribution of the real data are $p_{data(x)}$. The distribution of the data from the generator is p_g and the input noise variable is $p_g(z)$. The discriminator and the generator are differential multilayer perceptrons with θ_d and θ_g as parameters, respectively.

$D(x)$ is the probability that x comes from the real data distribution. $D(G(z))$ is the probability that $G(z)$ comes from the p_g, which is not from the real data distribution. $D(x)$ should point to 1 and $D(G(z))$ should point to 0, resulting in a min-max problem as shown in Equation (8). The objective function of GAN is shown in Equation (8).

$$\min_G \max_D V(D,G) = E_{x\sim p_{data(x)}}[\log D(x)] + E_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{8}$$

where $f_D\left(x^d\right) = E_{x \sim p_{data(x)}}[\log D(x)]$, $f_G(x^g) = E_{z \sim p_z(z)}[\log(1 - D(G(z)))]$.

According to the distributions (p_g and $p_{data(x)}$), the discriminator learns to distinguish between the real and the fake, and the generator also learns to produce a distribution that is similar to the real data in order to prevent the discriminator from easily distinguishing what is fake. If this learning process is repeated, $p_g = p_{data(x)}$, so we get to the point where the discriminator cannot distinguish anymore. Then, the converged $D(x)$ follows Equation (9).

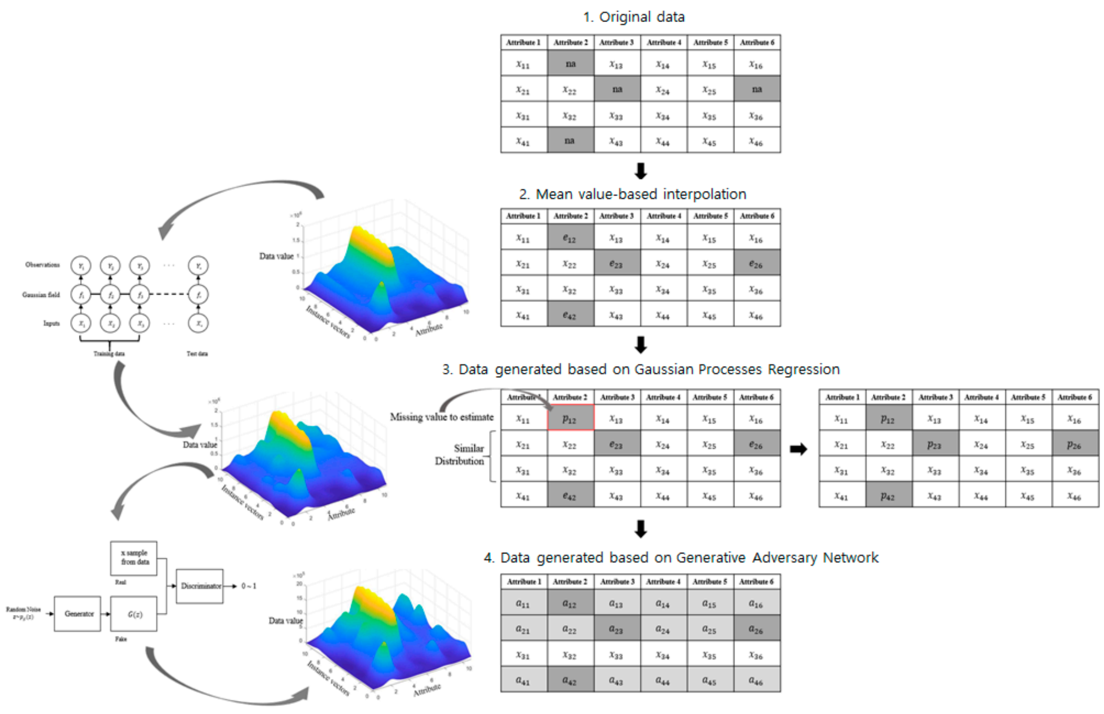
$$D^*(x) = \frac{p_{data(x)}}{p_{data(x)} + p_g(x)} \tag{9}$$

In Equation (8), the optimum value is obtained at $p_g = p_{data(x)}$ so, the value of $D^*(x)$ is $\frac{1}{2}$. The generator proceeds learning in a way so that $D^*(x)$ becomes close to $\frac{1}{2}$.

This research applies the GAN framework, which is used for the correction of missing data after GPR correction. The detailed framework is provided in Section 3.

3. Generative Adversarial Network-Based Missing Value Estimation Framework

In general, real data from manufacturing processes contains a number of missing values. This causes a lack of data and data imbalance as a result. These issues such as data shortages and data imbalances make it difficult to analyze the industrial data accurately. This section explains a new and effective framework to estimate the missing values and generate data that is closer to the real data distribution. Figure 5 shows the detailed procedures for generating a data set, which includes missing values.



1. Original data

Attribute 1	Attribute 2	Attribute 3	Attribute 4	Attribute 5	Attribute 6
x_{11}	na	x_{13}	x_{14}	x_{15}	x_{16}
x_{21}	x_{22}	na	x_{24}	x_{25}	na
x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}
x_{41}	na	x_{43}	x_{44}	x_{45}	x_{46}

2. Mean value-based interpolation

Attribute 1	Attribute 2	Attribute 3	Attribute 4	Attribute 5	Attribute 6
x_{11}	e_{12}	x_{13}	x_{14}	x_{15}	x_{16}
x_{21}	x_{22}	e_{23}	x_{24}	x_{25}	e_{26}
x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}
x_{41}	e_{42}	x_{43}	x_{44}	x_{45}	x_{46}

3. Data generated based on Gaussian Processes Regression

Attribute 1	Attribute 2	Attribute 3	Attribute 4	Attribute 5	Attribute 6
x_{11}	p_{12}	x_{13}	x_{14}	x_{15}	x_{16}
x_{21}	x_{22}	e_{23}	x_{24}	x_{25}	e_{26}
x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}
x_{41}	e_{42}	x_{43}	x_{44}	x_{45}	x_{46}

Attribute 1	Attribute 2	Attribute 3	Attribute 4	Attribute 5	Attribute 6
x_{11}	p_{12}	x_{13}	x_{14}	x_{15}	x_{16}
x_{21}	x_{22}	p_{23}	x_{24}	x_{25}	p_{26}
x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}
x_{41}	p_{42}	x_{43}	x_{44}	x_{45}	x_{46}

4. Data generated based on Generative Adversary Network

Attribute 1	Attribute 2	Attribute 3	Attribute 4	Attribute 5	Attribute 6
a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}
a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{26}
x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}
a_{41}	a_{42}	a_{43}	a_{44}	a_{45}	a_{46}

Figure 5. Missing value estimation and data generation procedures using the proposed framework.

As shown in Figure 5, the missing values of the original data are indicated as not available (na). First, if na exists, it is replaced with the average value of the attribute data that are missing. The average value e_{hl} is derived according to Equation (10).

$$e_{hl} = \frac{\sum_{h=1}^{n} a_h}{n} \tag{10}$$

where n is the number of remaining values, except for the missing value in the attribute, where the missing value comes from; h is the number of instance vectors and l is the number of attributes.

Then, approximate estimation of the missing value is achieved using GPR. In this case, GPR is applied to an instance vector.

$$f(Y_*) \sim \mathrm{N}(\mathrm{m}(X), \mathrm{k}(X, X_*)) \tag{11}$$

Equation (11) is derived using the prediction procedure provided in Section 2.1. The missing value is estimated by predicting a new estimate Y_* through Equations (12)–(14). Missing values are predicted based on GPR and updated to e_{hl}.

$$X = [x_{h1}, x_{h2}, \ldots, e_{hk}, \ldots x_{hl}] \tag{12}$$

$$f(e_{hk}) \sim \mathrm{N}([\mathrm{m}(e_{hk})], [\mathrm{k}(\mathrm{X_k}, \mathrm{X_1}) \cdots \mathrm{k}(\mathrm{X_k}, \mathrm{X_l})]) \tag{13}$$

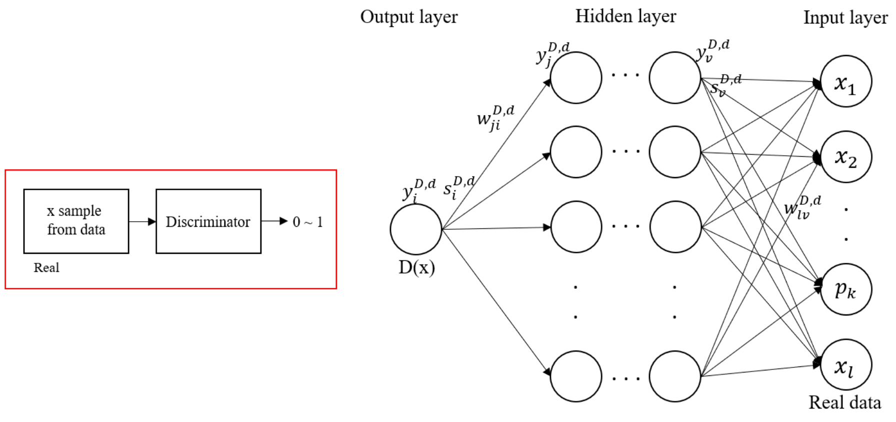
$$p_{hk} = Y_k = f(e_{hk}) + \epsilon \tag{14}$$

In Equation (12), X is an input vector with l attributes and k is the missing index among these attributes. Equation (13) is the distribution of the latent variables in the k^{th} attribute where the missing value occurs, and the missing value correction value p_{hk} using GPR is the same as Equation (13). Then, the e_{hk} is transformed to Y_k using Equation (14).

Finally, GAN is used to estimate missing values. The discriminator distinguishes the real instance vector distribution from the generated instance vector distribution, and the generator produces a new instance vector distribution based on the error generated by the discriminator.

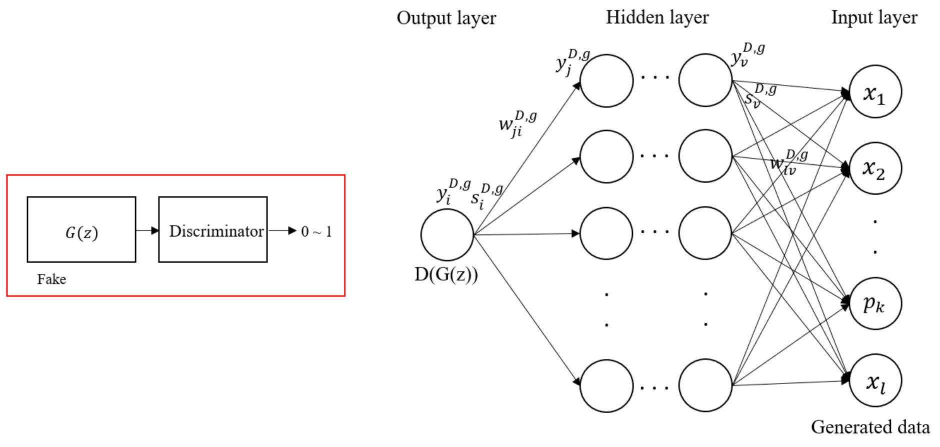
The discriminator derives gradients using the backpropagation algorithm to maximize Equation (8), while the generator derives the relevant gradient using the backpropagation algorithm to minimize $f_G(x^g)$ during the learning process. In this study, the gradient is derived to maximize $E_{z \sim p_z(z)}[\log(D(G(z)))]$ to increase the convergence speed of learning. The learning process using the backpropagation algorithm for the discriminator and generator is shown in Figure 6.

Figure 6a shows the discriminator's learning process that is used to distinguish whether the input data are from real data or are generated data. Figure 6b shows the learning process of a discriminator to distinguish whether the input data are real or not. Figure 6c shows the learning process in which the generator generates data from random noise values.

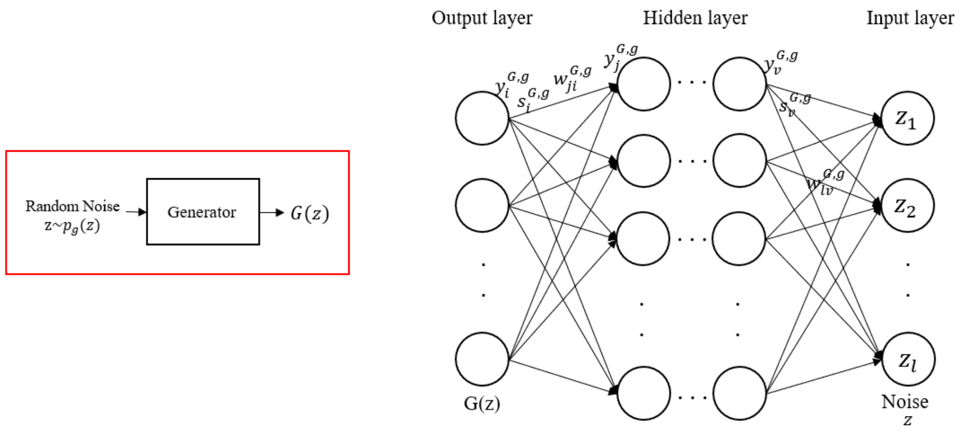


(a)

Figure 6. *Cont.*



(**b**)



(**c**)

Figure 6. Backpropagation-based learning process for the ggenerative adversarial network (GAN). (**a**) architecture of the discriminator from real data; (**b**) architecture of the discriminator from fake data; (**c**) architecture of the generator.

$$\frac{\partial V}{\partial \theta^{(D)}} = \frac{\partial f\left(x^d\right)}{\partial \theta^{(D)}} + \frac{\partial f(x^g)}{\partial \theta^{(D)}} \tag{15}$$

$$\frac{\partial f_D\left(x^d\right)}{\partial w_{ji}^{D,d}} = \frac{\partial f_D\left(x^d\right)}{\partial y_i^{D,d}} \cdot \frac{\partial y_i^{D,d}}{\partial s_i^{D,d}} \cdots \frac{\partial s_v^{D,d}}{\partial w_{ji}^{D,d}} \tag{16}$$

$$\frac{\partial f_D\left(x^d\right)}{\partial w_{Nj}^{D,d}} = \frac{1}{D(x)} \cdot \left(y_i^{D,d}\right)\prime \cdot w_{ji}^{D,d} \cdot \left(y_j^{D,d}\right)\prime \cdot x^d,\ \frac{\partial f_G(x^g)}{\partial w_{ji}^{D,g}} = \frac{\partial f_G(x^g)}{\partial y_i^{D,g}} \cdot \frac{\partial y_i^{D,g}}{\partial s_i^{D,g}} \cdots \frac{\partial s_v^{D,g}}{\partial w_{ji}^{D,g}} \tag{17}$$

$$\frac{\partial f_G(x^g)}{\partial w_{Nj}^{D,g}} = \frac{1}{1 - D(G(z))} \cdot \left(y_i^{D,g}\right)\prime \cdot w_{ji}^{D,g} \cdot \left(y_j^{D,g}\right)\prime \cdot x^g,\ \frac{\partial V}{\partial \theta^{(G)}} = \frac{\partial f_G(x^g)}{\partial \theta^{(G)}} \tag{18}$$

$$\frac{\partial f_G(x^g)}{\partial w_{ji}^{G,g}} = \frac{\partial f_G(x^g)}{\partial y_i^{G,g}} \cdot \frac{\partial y_i^{G,g}}{\partial s_i^{G,g}} \cdots \frac{\partial s_v^{G,g}}{\partial w_{ji}^{G,g}},\ \frac{\partial f_G(x^g)}{\partial w_{Nj}^{G,g}} = \frac{1}{D(G(z))} \cdot \left(y_i^{G,g}\right)\prime \cdot w_{ji}^{G,g} \cdot \left(y_j^{G,g}\right)\prime \cdot z \tag{19}$$

where $y_i = \frac{1}{1+e^{-s_i}}$, $s_i = \sum\limits_{N=1} y_i w_{ji}$ $y_j = \frac{1}{1+e^{-s_j}}$, $s_j = \sum\limits_{N=1} y_j w_{Nj}$.

Equations (15)–(17) summarize the backpropagation processes of the discriminator, and Equation (18) and Equation (19) summarize the backpropagation process of the generator. The backpropagation process of the discriminator derives the gradients for $D(x)$ and $D(G(z))$, as shown in respective Equation (16) and Equation (17).

w is the weight of the neural network, and the sigmoid function is used as its activation function in this paper. Based on the output derived by the discriminator learning, the generator updates w through the gradient to produce a newly generated data. When $D(x)$ converges through repetition, the estimation process is terminated.

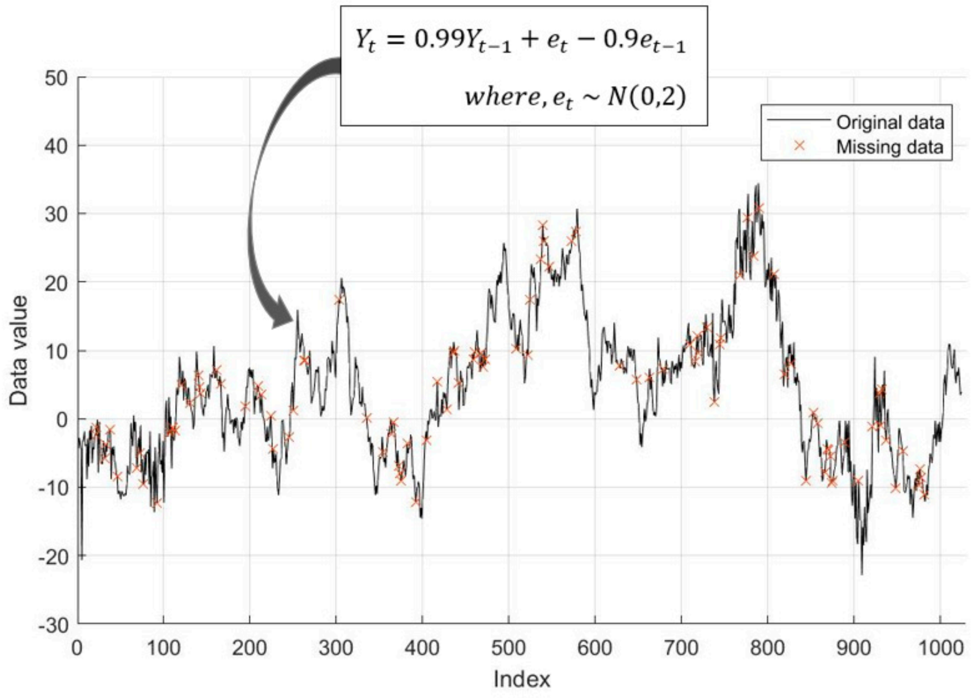
$$X = [x_{h1}, x_{h2}, \ldots, p_{hk}, \ldots x_{hl}] \tag{20}$$

$$\frac{\partial f_G(x^g)}{\partial X} = \frac{1}{D(G(z))} \cdot \left(y_i^{G,g}\right)' \cdot w_{ji}^{G,g} \cdot \left(y_j^{G,g}\right)' \cdot z \tag{21}$$

The missing value p_{hk} in Equation (20) is estimated and a new data set is generated using Equation (21). Using these processes, a new set of data are generated. The generated data are considered balanced data.

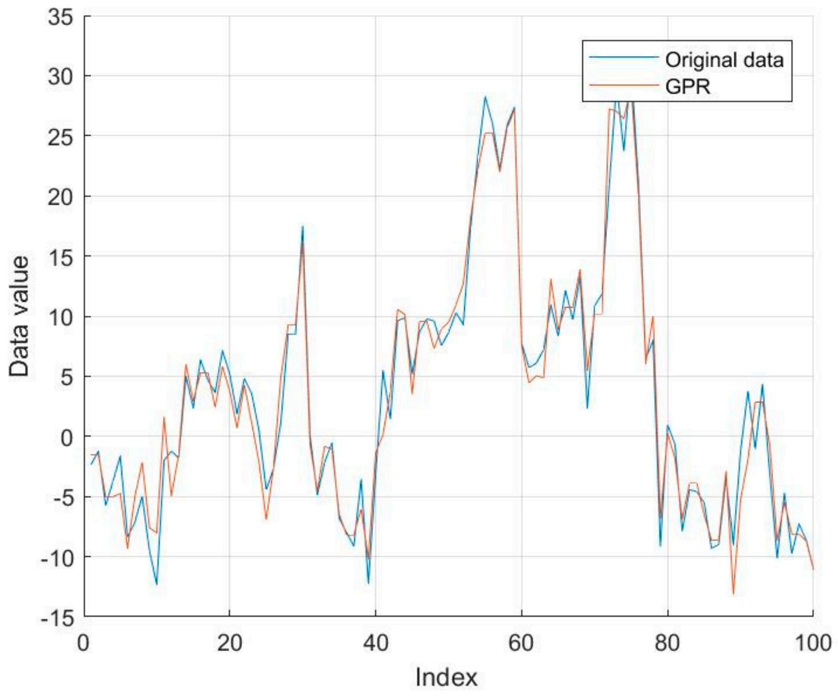
In order to show the effectiveness of GPR-based GAN, time series data with missing values were tested. Figure 7a shows original time-series data, including Autoregressive Moving-Average (ARMA) model - ARMA(1,1) with the Gaussian noise N(0,2). The number of data points is 1024 (N = 1024). Among them, 100 points are randomly picked as missing parts.

Then, GPR-GAN is applied to estimate the missing value. Figure 7b shows the data gap between the original data and the newly generated data. In order to measure the accuracy of the generated method, Equation (22) is applied.



(**a**)

Figure 7. *Cont.*



(**b**)

Figure 7. A time series-based numerical analysis using GPR-GAN: (**a**) randomly generated original data; (**b**) comparison between actual data and data estimated using GPR.

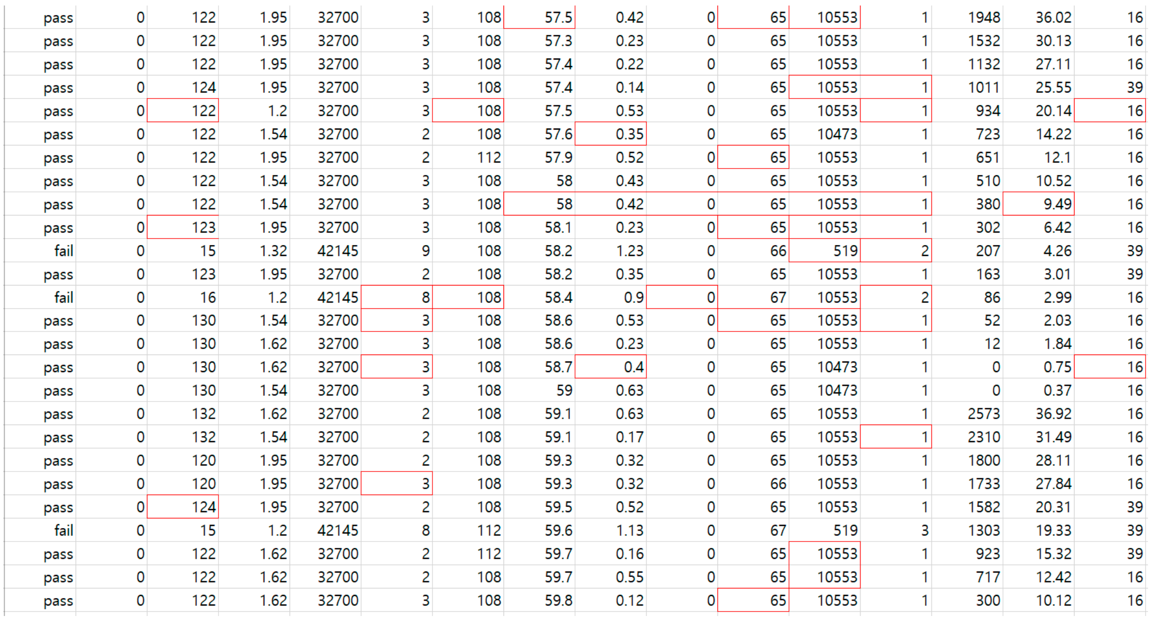
$$\text{Accuracy } (\%) = \left(1 - \sum_{i=1}^{N} \frac{O_i - G_i}{O_i}\right) \cdot 100 \tag{22}$$

The proposed method has 91.23% accuracy in the provided numerical test.

In order to show the effectiveness of the proposed framework, randomly generated data and their pass/fail outputs are considered. Figure 8a shows a data set from a randomized time series with Gaussian noise $\mathrm{N}(0, 2)$. The data set size is 50 data points and each data point is composed of 20 attributes. Their outputs are divided randomly into 44 passes (1) and 6 fails (0). Then, as shown in Figure 8b, several selected sections are considered as the sections with missing values.

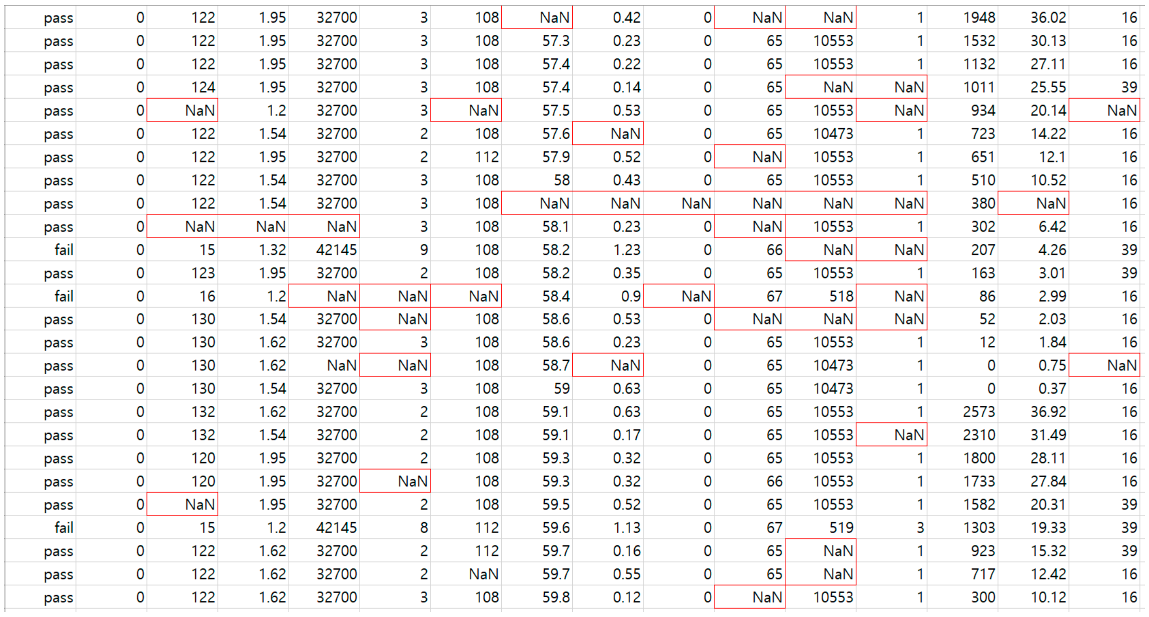
Figure 8c shows a newly generated data set using the proposed framework. Finally, the pass/fail predictions using the randomly generated original data and the generated data are conducted as shown in Figure 8d.

The following section shows how the proposed framework is effective using the real data set.



pass	0	122	1.95	32700	3	108	57.5	0.42	0	65	10553	1	1948	36.02	16
pass	0	122	1.95	32700	3	108	57.3	0.23	0	65	10553	1	1532	30.13	16
pass	0	122	1.95	32700	3	108	57.4	0.22	0	65	10553	1	1132	27.11	16
pass	0	124	1.95	32700	3	108	57.4	0.14	0	65	10553	1	1011	25.55	39
pass	0	122	1.2	32700	3	108	57.5	0.53	0	65	10553	1	934	20.14	16
pass	0	122	1.54	32700	2	108	57.6	0.35	0	65	10473	1	723	14.22	16
pass	0	122	1.95	32700	2	112	57.9	0.52	0	65	10553	1	651	12.1	16
pass	0	122	1.54	32700	3	108	58	0.43	0	65	10553	1	510	10.52	16
pass	0	122	1.54	32700	3	108	58	0.42	0	65	10553	1	380	9.49	16
pass	0	123	1.95	32700	3	108	58.1	0.23	0	65	10553	1	302	6.42	16
fail	0	15	1.32	42145	9	108	58.2	1.23	0	66	519	2	207	4.26	39
pass	0	123	1.95	32700	2	108	58.2	0.35	0	65	10553	1	163	3.01	39
fail	0	16	1.2	42145	8	108	58.4	0.9	0	67	10553	2	86	2.99	16
pass	0	130	1.54	32700	3	108	58.6	0.53	0	65	10553	1	52	2.03	16
pass	0	130	1.62	32700	3	108	58.6	0.23	0	65	10553	1	12	1.84	16
pass	0	130	1.62	32700	3	108	58.7	0.4	0	65	10473	1	0	0.75	16
pass	0	130	1.54	32700	3	108	59	0.63	0	65	10473	1	0	0.37	16
pass	0	132	1.62	32700	2	108	59.1	0.63	0	65	10553	1	2573	36.92	16
pass	0	132	1.54	32700	2	108	59.1	0.17	0	65	10553	1	2310	31.49	16
pass	0	120	1.95	32700	2	108	59.3	0.32	0	65	10553	1	1800	28.11	16
pass	0	120	1.95	32700	3	108	59.3	0.32	0	66	10553	1	1733	27.84	16
pass	0	124	1.95	32700	2	108	59.5	0.52	0	65	10553	1	1582	20.31	39
fail	0	15	1.2	42145	8	112	59.6	1.13	0	67	519	3	1303	19.33	39
pass	0	122	1.62	32700	2	112	59.7	0.16	0	65	10553	1	923	15.32	39
pass	0	122	1.62	32700	2	108	59.7	0.55	0	65	10553	1	717	12.42	16
pass	0	122	1.62	32700	3	108	59.8	0.12	0	65	10553	1	300	10.12	16

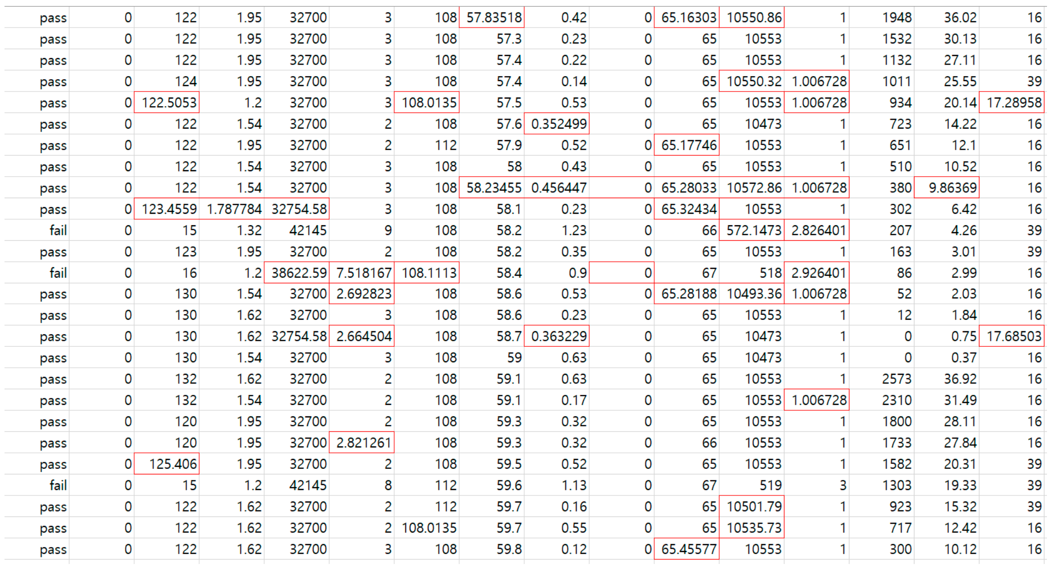
(**a**)



pass	0	122	1.95	32700	3	108	NaN	0.42	0	NaN	NaN	1	1948	36.02	16
pass	0	122	1.95	32700	3	108	57.3	0.23	0	65	10553	1	1532	30.13	16
pass	0	122	1.95	32700	3	108	57.4	0.22	0	65	10553	1	1132	27.11	16
pass	0	124	1.95	32700	3	108	57.4	0.14	0	65	NaN	NaN	1011	25.55	39
pass	0	NaN	1.2	32700	3	NaN	57.5	0.53	0	65	10553	NaN	934	20.14	NaN
pass	0	122	1.54	32700	2	108	57.6	NaN	0	65	10473	1	723	14.22	16
pass	0	122	1.95	32700	2	112	57.9	0.52	0	NaN	10553	1	651	12.1	16
pass	0	122	1.54	32700	3	108	58	0.43	0	65	10553	1	510	10.52	16
pass	0	122	1.54	32700	3	108	NaN	NaN	NaN	NaN	NaN	NaN	380	NaN	16
pass	0	NaN	NaN	NaN	3	108	58.1	0.23	0	NaN	10553	1	302	6.42	16
fail	0	15	1.32	42145	9	108	58.2	1.23	0	66	NaN	NaN	207	4.26	39
pass	0	123	1.95	32700	2	108	58.2	0.35	0	65	10553	1	163	3.01	39
fail	0	16	1.2	NaN	NaN	NaN	58.4	0.9	NaN	67	518	NaN	86	2.99	16
pass	0	130	1.54	32700	NaN	108	58.6	0.53	0	NaN	NaN	NaN	52	2.03	16
pass	0	130	1.62	32700	3	108	58.6	0.23	0	65	10553	1	12	1.84	16
pass	0	130	1.62	NaN	NaN	108	58.7	NaN	0	65	10473	1	0	0.75	NaN
pass	0	130	1.54	32700	3	108	59	0.63	0	65	10473	1	0	0.37	16
pass	0	132	1.62	32700	2	108	59.1	0.63	0	65	10553	1	2573	36.92	16
pass	0	132	1.54	32700	2	108	59.1	0.17	0	65	10553	NaN	2310	31.49	16
pass	0	120	1.95	32700	2	108	59.3	0.32	0	65	10553	1	1800	28.11	16
pass	0	120	1.95	32700	NaN	108	59.3	0.32	0	66	10553	1	1733	27.84	16
pass	0	NaN	1.95	32700	2	108	59.5	0.52	0	65	10553	1	1582	20.31	39
fail	0	15	1.2	42145	8	112	59.6	1.13	0	67	519	3	1303	19.33	39
pass	0	122	1.62	32700	2	112	59.7	0.16	0	65	NaN	1	923	15.32	39
pass	0	122	1.62	32700	2	NaN	59.7	0.55	0	65	NaN	1	717	12.42	16
pass	0	122	1.62	32700	3	108	59.8	0.12	0	NaN	10553	1	300	10.12	16

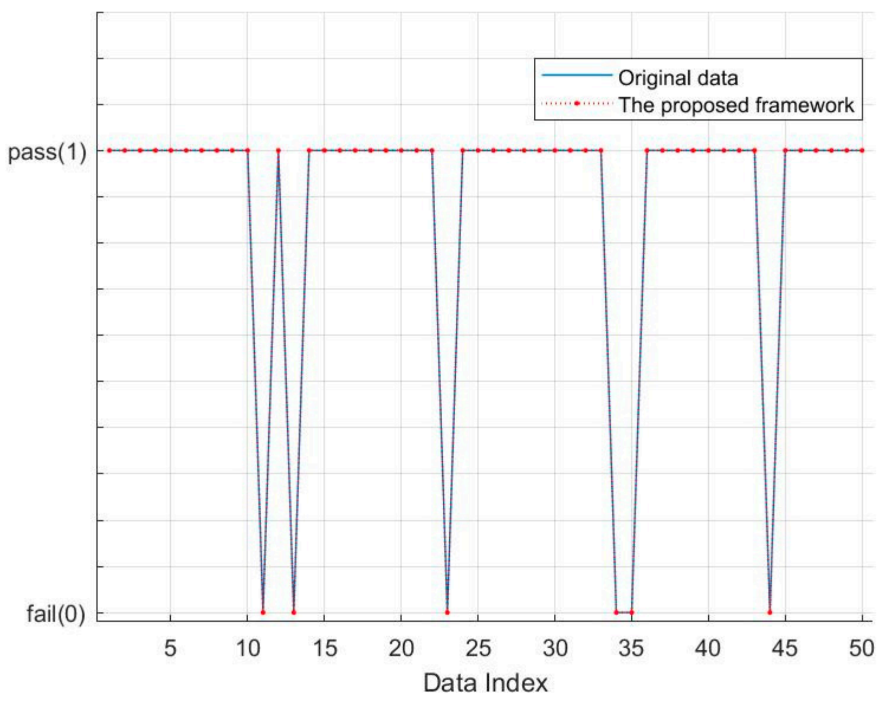
(**b**)

Figure 8. *Cont.*



pass	0	122	1.95	32700	3	108	57.83518	0.42	0	65.16303	10550.86	1	1948	36.02	16
pass	0	122	1.95	32700	3	108	57.3	0.23	0	65	10553	1	1532	30.13	16
pass	0	122	1.95	32700	3	108	57.4	0.22	0	65	10553	1	1132	27.11	16
pass	0	124	1.95	32700	3	108	57.4	0.14	0	65	10550.32	1.006728	1011	25.55	39
pass	0	122.5053	1.2	32700	3	108.0135	57.5	0.53	0	65	10553	1.006728	934	20.14	17.28958
pass	0	122	1.54	32700	2	108	57.6	0.352499	0	65	10473	1	723	14.22	16
pass	0	122	1.95	32700	2	112	57.9	0.52	0	65.17746	10553	1	651	12.1	16
pass	0	122	1.54	32700	3	108	58	0.43	0	65	10553	1	510	10.52	16
pass	0	122	1.54	32700	3	108	58.23455	0.456447	0	65.28033	10572.86	1.006728	380	9.86369	16
pass	0	123.4559	1.787784	32754.58	3	108	58.1	0.23	0	65.32434	10553	1	302	6.42	16
fail	0	15	1.32	42145	9	108	58.2	1.23	0	66	572.1473	2.826401	207	4.26	39
pass	0	123	1.95	32700	2	108	58.2	0.35	0	65	10553	1	163	3.01	39
fail	0	16	1.2	38622.59	7.518167	108.1113	58.4	0.9	0	67	518	2.926401	86	2.99	16
pass	0	130	1.54	32700	2.692823	108	58.6	0.53	0	65.28188	10493.36	1.006728	52	2.03	16
pass	0	130	1.62	32700	3	108	58.6	0.23	0	65	10553	1	12	1.84	16
pass	0	130	1.62	32754.58	2.664504	108	58.7	0.363229	0	65	10473	1	0	0.75	17.68503
pass	0	130	1.54	32700	3	108	59	0.63	0	65	10473	1	0	0.37	16
pass	0	132	1.62	32700	2	108	59.1	0.63	0	65	10553	1	2573	36.92	16
pass	0	132	1.54	32700	2	108	59.1	0.17	0	65	10553	1.006728	2310	31.49	16
pass	0	120	1.95	32700	2	108	59.3	0.32	0	65	10553	1	1800	28.11	16
pass	0	120	1.95	32700	2.821261	108	59.3	0.32	0	66	10553	1	1733	27.84	16
pass	0	125.406	1.95	32700	2	108	59.5	0.52	0	65	10553	1	1582	20.31	39
fail	0	15	1.2	42145	8	112	59.6	1.13	0	67	519	3	1303	19.33	39
pass	0	122	1.62	32700	2	112	59.7	0.16	0	65	10501.79	1	923	15.32	39
pass	0	122	1.62	32700	2	108.0135	59.7	0.55	0	65	10535.73	1	717	12.42	16
pass	0	122	1.62	32700	3	108	59.8	0.12	0	65.45577	10553	1	300	10.12	16

(**c**)



(**d**)

Figure 8. Numerical analysis using the proposed framework: (**a**) randomly generated data and their pass/fail outputs; (**b**) randomly selected missing values in the data set; (**c**) newly generated data using the proposed framework; (**d**) classification results between the original data and the generated data using the proposed framework.

4. Data Issues in Air Pressure System and Numerical Analysis

This section proves the effectiveness of the proposed framework and compares it with other existing methods. As discussed in previous sections, the proposed framework has the advantage of high performance in classification that comes from using accurate interpolations of missing values using the GPR-based GAN framework.

In order to show the effectiveness of the proposed framework, real-world data are used, specifically the APS failure Scania trucks data set [1]. The data consist of 170 attributes and 60,000 instance vectors. Training data are divided into 59,000 negative classes of APS-based faults and 1000 positive cases. Test data are divided into 15,625 negative cases and 375 positive cases. Several attributes in each piece of data have missing values. Table 4 shows the number of missing values in the APS data.

Table 4. Missing data issues in APS data.

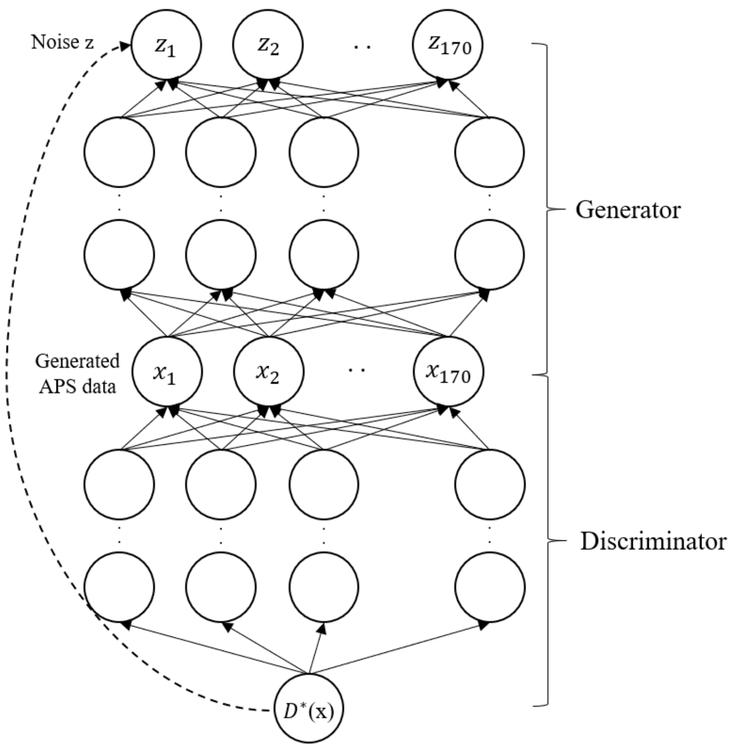
	Categories	**Total Size**	**Data Item with Missing Values**	**Percentages of Missing Items**
Training data	Number of attributes	170	<u>170</u>	100%
	Data set	60,000	<u>59,998</u>	99.99%
Test data	Number of attributes	170	<u>169</u>	99.41%
	Data set	16,000	<u>16,000</u>	100%

In order to estimate the missing values, the proposed framework is applied. Figure 9a shows the applied structure of the proposed framework. The generator regenerates the data by reflecting the discriminator's objective value that distinguishes whether the data points generated by the generator are real data points or not. The deep neural network (DNN) is trained and tested to produce interpolated data, as shown in the red box in Figure 9b. A pass (1) or fail (0) is diagnosed using the DNN model. The DNN model has one input layer, multiple hidden layers, and one output layer. The hyperparameter for the experiment is set to 0.001 for its learning rate, 28 for the mini-batch size, 100 for max-epochs, and 0.5 for momentum.

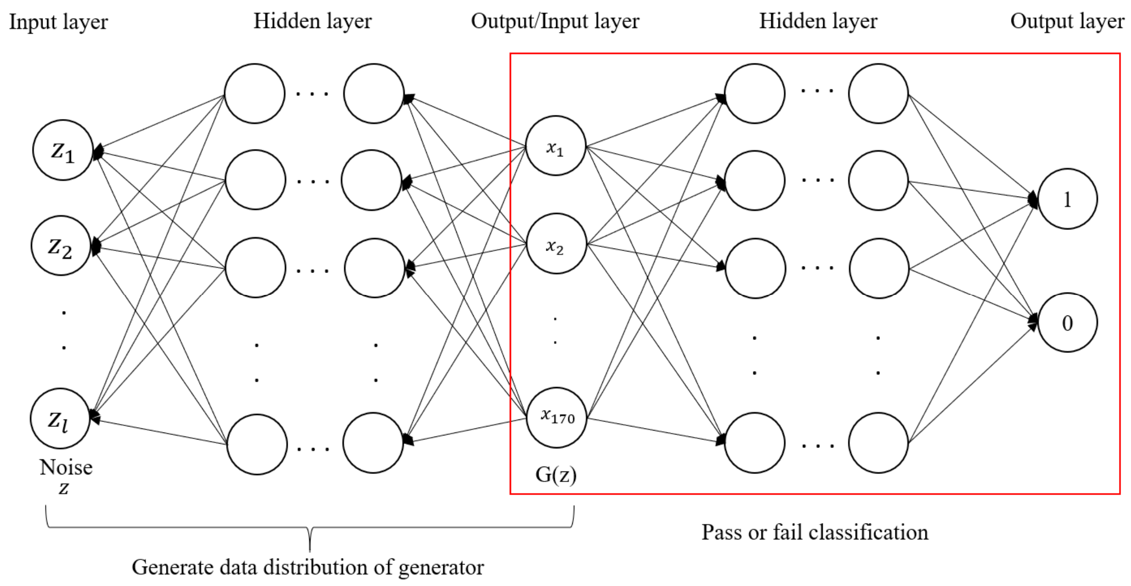
In order to verify the effectiveness of the proposed framework, it is compared with other existing methods, including the classification and regression tree (CART), GPR, K-means, mean-based GAN, and compressed sensing (CS) methods. Table 5 summarizes these models and the relevant parameters.

Table 5. Models and parameters for each testing algorithm. Note: classification and regression tree = CART; compressed sensing = CS.

Tested Frameworks	**Equation**	**Parameter**
GPR-based GAN (Proposed framework)	$\min_G \max_D V(D,G) = E_{x\sim p_{data(x)}}[\log D(x)] + E_{z\sim p_z(z)}[\log(1-D(G(z)))]$	Max epochs = 100 Learning rate = 0.1 Momentum = 0.5
CART	$I_G(f) = 1 - \sum_{i=1}^{m} f_i^2$ $Cost\ function = \frac{m_{left}}{m} I_G\left(f_{left}\right) + \frac{m_{right}}{m} I_G\left(f_{right}\right)$	Tree's depth = 50,000 Costs of misclassification = 1:1 Occupation percentage = 1:1
GPR	$f \sim \mathrm{GP}(\mathrm{m}(x), \mathrm{k}(x, x_*))$	Nonparametric
K-means	$\arg\min_S \sum_{i=1}^{k} \sum_{x\in S_i} \lVert x-\mu_i \rVert^2$	Maximum number of runs = 100 Distance calculation method = Euclidean Surface pretreatment = Normalization
Mean-based GAN	$\min_G \max_D V(D,G) = E_{x\sim p_{data(x)}}[\log D(x)] + E_{z\sim p_z(z)}[\log(1-D(G(z)))]$	Max epochs = 100 Learning rate = 0.1 Momentum = 0.5
CS	$\min_x \lVert x \rVert_1\ subject\ to\ y = wx$	Max epochs = 100 Learning rate = 0.1



(**a**)



(**b**)

Figure 9. GAN-based DNN structure: (**a**) structure of the proposed framework; (**b**) classification structure using DNN.

Table 6 shows the results of the confusion matrix experiment with test data for each classification model.

Table 6. Confusion matrix for each classification model.

		Pass (Predicted)	**Fail (Predicted)**
Proposed Framework	**Pass (Actual)**	**15,418**	**207**
	Fail (Actual)	**59**	**316**
		Pass (Predicted)	Fail (Predicted)
CART	Pass (Actual)	14,084	1541
	Fail (Actual)	259	116
		Pass (Predicted)	Fail (Predicted)
GPR	Pass (Actual)	14,625	1000
	Fail (Actual)	214	161
		Pass (Predicted)	Fail (Predicted)
K-means	Pass (Actual)	11,249	4376
	Fail (Actual)	295	80
		Pass (Predicted)	Fail (Predicted)
Mean-based GAN	Pass (Actual)	14,654	971
	Fail (Actual)	146	229
		Pass (Predicted)	Fail (Predicted)
CS	Pass (Actual)	14,782	843
	Fail (Actual)	106	269

As shown in Table 6 and Figure 10, the proposed GPR-based GAN framework shows the lowest rates of type-I and type-II errors.

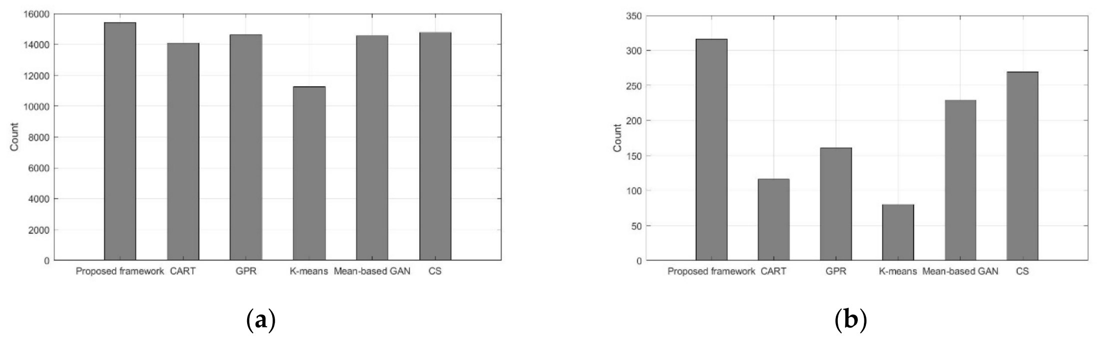


Figure 10. Comparison graphs among each classification model: (**a**) comparisons of "an actual passes versus predicted passes" in the negative class; (**b**) comparison of "an actual failures versus predicted failures" in the positive class.

Table 7 summarizes several performance evaluation indicators using the confusion matrix from Table 6. A true positive (TP) is given if a pass is indicated in real data and a pass is indicated by the classification model. A false negative (FN) is given if a pass is indicated in real data but a fail is indicated by the classification model. A false positive (FP) is given if a fail is indicated in real data but a pass is indicated by the classification model. A true negative (TN) is given if a fail is indicated in real data and a fail is indicated by the classification model.

Table 7. Performance evaluation indicators.

	Proposed Framework	CART	GPR	K-Means	Mean-Based GAN	CS
Precision	0.996	0.981	0.985	0.974	0.99	0.992
Recall	0.986	0.901	0.936	0.719	0.937	0.946
Fall-out	0.157	0.690	0.570	0.786	0.389	0.282
Accuracy	0.983	0.887	0.924	0.708	0.93	0.94

As outlined in Table 7, the definition of "precision" is the number of TP divided by the number of TP plus FP, while the "recall" is the number of TP divided by the number of TP plus FN. The "precision" and the "recall" handle the cases where the classification model classifies a pass when the real data indicates a pass. The "fall-out" is the number of FP divided by the number of TN plus FP. It handles misclassifications by the classification model when the real data indicate a fail but the classification model gives a pass. The "accuracy" is the number of TP plus TN divided by the sum of TP, TN, FP, and FN. "Accuracy" handles cases where both passes and fails are correctly classified. This is used as the main performance evaluation indicator.

As shown in Table 7 and Figure 11, the accuracy of the proposed framework is 98.3% and the fall-out is 15.7%, thus it is experimentally proved that the missing value handling using the proposed framework has better performance than other existing methods.

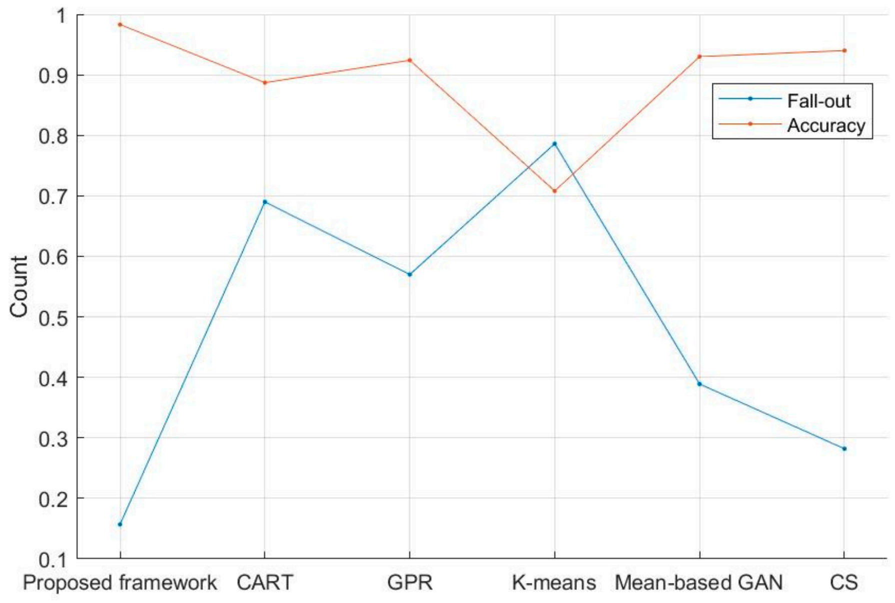


Figure 11. Comparison graph of "fall-out" and "accuracy" among four test models.

5. Conclusions

Failure analysis and relevant predictions generated from industrial big data are essential processes for industry in order to produce high quality and reliable products. However, most industrial big data sets are incomplete due to various issues. In these situations, existing algorithms fail to provide accurate corrections for the missing data. Therefore, any classification task executed on these kinds of incomplete data sets shows very poor performance.

In order to overcome this issue, the proposed framework generates a new complete data set using the proposed GPR-based GAN framework. The provided framework is based on the symmetry properties. First, the missing values are replaced with the mean value of the appropriate attribute. Then, the missing value estimates are refined by applying GPR. The data characteristics from this GPR process are linked to the GAN. Finally, the GAN is applied to generate further refinements to generate new data that are similar to the real data. The generated data are used as training data and help to overcome any data imbalances in the input data set.

In order to prove the performance of the proposed framework, it is compared with existing classification models using a real industrial data set related to APS failure in Scania trucks. Numerical analysis shows that the proposed framework has higher accuracy and lower fall-out than existing classification models. Through numerical analysis, it was confirmed that the proposed framework is effective compared with existing classification models.

The proposed framework can be used to estimate missing values in a data set with a high frequency of missing data. In addition, industrial data sets that are highly distorted with large data imbalances can be successfully analyzed using the proposed framework. In future studies, we hope to incorporate more efficient computation methods to handle data from multiple industrial areas.

Author Contributions: E.O. and H.L. conceptualized the framework and developed the methodologies. E.O. implemented and validated the framework. H.L. supervised the overall research processes. E.O. wrote the manuscript. H.L. reviewed and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Priority Research Centers Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education, Science, and Technology (2018R1A6A1A03024003).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. UCI Machine Learning Repository. 2017. Available online: https://archive.ics.uci.edu/ml/datasets/APS+Failure+at+Scania+Trucks (accessed on 8 December 2017).
2. Paul, D.A. *Missing Data*; Sage Publications Inc.: Thousand Oaks, CA, USA, 2002; pp. 27–74.
3. Dempster, P.; Laird, N.M.; Rubin, D.B. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. R. Stat. Soc.* **1997**, *39*, 1–22. [CrossRef]
4. Moon, T.K. The expectation-maximization algorithm. *IEEE Signal Process. Mag.* **1996**, *13*, 47–60. [CrossRef]
5. Hastie, T.; Tibshirani, R.; Sherlock, G.; Eisen, M.; Brown, P.; Bolstein, D. *Imputing Missing Data for Gene Expression Arrays*; Technical Report; Standford University Press: Standford, CA, USA, 1999.
6. Troyanskaya, O.; Cantor, M.; Sherlock, G.; Brown, P.; Hastie, T.; Tibshirani, R.; Botstein, D.; Altman, R. Missing value estimation methods for DNA microarrays. *Bioinformatics* **2001**, *17*, 520–525. [CrossRef] [PubMed]
7. Zhang, Z. Missing data imputation: Focusing on single imputation. *Ann. Transl. Med.* **2016**, *4*, 9. [CrossRef] [PubMed]
8. Gondara, L.; Wang, K. Multiple imputation using deep denoising autoencoders. *arXiv* **2017**, arXiv:1705.02737.
9. Gemmeke, J.F.; Hamme, H.V.; Cranen, B.; Boves, L. Compressive Sensing for Missing Data Imputation in Noise Robust Speech Recognition. *IEEE J. Sel. Top. Signal Process.* **2010**, *4*, 272–287. [CrossRef]
10. Oba, S.; Sato, M.; Takemasa, I.; Monden, M.; Matsubara, K.; Ishii, S. A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics* **2003**, *19*, 2088–2096. [CrossRef]
11. Little, R.J.; Rubin, D.B. *Statistical Analysis with Missing Data*, 3rd ed.; NJ John Wiley & Sons Inc.: Hoboken, NJ, USA, 2020; pp. 29–42.
12. Gondek, C.; Hafner, D.; Sampson, O.R. Prediction of Failures in the Air Pressure System of Scania Trucks using a Random Forest and Feature Engineering. *Adv. Intell. Data Anal.* **2016**, *9897*, 398–402. [CrossRef]
13. Perepu, S.K.; Tangirala, A.K. Reconstruction of missing data using compressed sensing techniques with adaptive dictionary. *J. Process Control* **2016**, *47*, 175–190. [CrossRef]
14. Chodosh, N.; Wang, C.; Lucey, S. Deep Convolutional Compressed Sensing for LiDAR Depth Completion. *Comput. Vis. ACCV* **2018**, *11361*, 499–513. [CrossRef]
15. Williams, C.K.I.; Rasmussen, C.E. *Gaussian Processes for Machine Learning*; The MIT Press: London, UK, 2006; pp. 7–128.
16. Williams, C.K.I.; Rasmussen, C.E. Gaussian Processes for Regression. *Adv. Neural Process. Syst.* **1996**, *8*, 514–520.
17. Rasmussen, C.E. Gaussian Processes in Machine Learning. *Adv. Lect. Mach. Learn.* **2004**, *3176*, 63–71. [CrossRef]
18. Chu, W.; Ghahramani, Z. Gaussian Processes for Ordinal Regression. *J. Mach. Learn. Res.* **2005**, *6*, 1019–1041.
19. Schulz, E.; Speekenbrink, M.; Krause, A. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *J. Math. Psychol.* **2017**, *85*, 1–16. [CrossRef]
20. Jochem, V.; Juan, P.R.; Anatoly, G.; Jesus, D.; Jose, M.; Gustau, C. Spectral band selection for vegetation properties retrieval using Gaussian processes regression. *Int. J. Appl. Earth Obs. Geoinf.* **2016**, *52*, 554–567. [CrossRef]

21. Ak, C.; Ergonul, O.; Sencan, I.; Torunoglu, M.A.; Gonen, M. Spatiotemporal prediction of infectious diseases using structured Gaussian processes with application to Crimean-Congo hemorrhagic fever. *PLoS Negl. Trop. Dis.* **2018**, *12*, e0006737. [CrossRef] [PubMed]
22. Luttinen, J.; Ilin, A. Efficient Gaussian process inference for short-scale spatio-temporal modeling. In Proceedings of the 15th International Conference on Artificial Intelligence and Statistics, La Palma, Canary Islands, 21–23 April 2012; pp. 741–750.
23. Nguyen, D.; Peters, J. Learning Robot Dynamics for Computed Torque Control using Local Gaussian Processes Regression. In Proceedings of the ECSIS Symposium on Learning and Adaptive Behaviors for Robotic Systems, Edinburgh, UK, 6–8 August 2008; pp. 59–64.
24. Nguyen, L.; Hu, G.; Spanos, C.J. Spatio-temporal environmental monitoring for smart buildings. In Proceedings of the 13th IEEE International Conference on Control and Automation, Ohrid, Macedonia, 3–6 July 2017; pp. 277–282.
25. Chen, N.; Qian, Z.; Meng, X.; Nabney, I.T. Short-term wind power forecasting using Gaussian processes. In Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013; pp. 2790–2796.
26. Oh, E.; Lee, H. Development of a Convolution-Based Multi-Directional and Parallel Ant Colony Algorithm Considering a Network with Dynamic Topology Changes. *Appl. Sci.* **2019**, *9*, 3646. [CrossRef]
27. Goodfellow, I.J.; Pouget, J.; Mirza, M.; Xu, B.; WardeFarley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the NIPS 2014, Montreal, QC, Canada, 8–13 December 2014; pp. 1–9.
28. Kim, H.; Lee, H. Fault Detect and Classification Framework for Semiconductor Manufacturing Processes using Missing Data Estimation and Generative Adversary Network. *J. Korean Inst. Intell. Syst.* **2018**, *28*, 393–400. [CrossRef]
29. Yoon, J.; Jordon, J.; Schaar, M. GAIN: Missing Data Imputation using Generative Adversarial Nets. *arXiv* **2018**, arXiv:1806.02920.
30. Kim, H.; Lee, H. Generative Adversarial Networks based Data Generation Framework for Overcoming Imbalanced Manufacturing Process Data. *J. Korean Inst. Intell. Syst.* **2019**, *29*, 1–8. [CrossRef]
31. Shang, C.; Palmer, A.; Sun, J.; Chen, K.; Lu, J.; Bi, J. VIGAN: Missing view imputation with generative adversarial networks. In Proceedings of the IEEE International Conference on Big Data, Boston, MA, USA, 11–14 December 2017; pp. 766–775.
32. Mao, X.; Li, Q.; Xie, H.; Lau, R.Y.K.; Wang, Z.; Smolley, S.P. Least Squares Generative Adversarial Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2794–2802.
33. Zhao, J.; Mathieu, M.; LeCun, Y. Energy-based generative adversarial network. *arXiv* **2016**, arXiv:1609.03126.
34. Li, J.; Liang, X.; Wei, Y.; Xu, T.; Feng, J.; Yan, S. Perceptual Generative Adversarial Networks for Small Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Hawaii, HI, USA, 21–26 July 2017; pp. 1222–1230.