

Article

Exploiting Generative Adversarial Networks as an Oversampling Method for Fault Diagnosis of an Industrial Robotic Manipulator

Ziqiang Pu ¹, Diego Cabrera ², René-Vinicio Sánchez ², Mariela Cerrada ², Chuan Li ³ and José Valente de Oliveira ^{4,*}

¹ National Research Base of Intelligent Manufacturing Service, Chongqing Technology and Business University, China and Universidade do Algarve, 8005-139 Faro, Portugal; a66835@ualg.pt

² GIDTEC Research Group, Universidad Politécnica Salesiana, Cuenca 010105, Ecuador; dcabrera@ups.edu.ec (D.C.); rsanchezl@ups.edu.ec (R.-V.S.); mcerrada@ups.edu.ec (M.C.)

³ National Research Base of Intelligent Manufacturing Service, Chongqing Technology and Business University, Chongqing 400067, China; chuanli@ctbu.edu.cn

⁴ Universidade do Algarve and with the Center of Intelligent Systems, IDMEC, Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisboa, Portugal

* Correspondence: jvo@ualg.pt; Tel.: +351-289-800-10

Received: 30 September 2020; Accepted: 28 October 2020; Published: 31 October 2020



Abstract: Data-driven machine learning techniques play an important role in fault diagnosis, safety, and maintenance of the industrial robotic manipulator. However, these methods require data that, more often than not, are hard to obtain, especially data collected from fault condition states and, without enough and appropriated (balanced) data, no acceptable performance should be expected. Generative adversarial networks (GAN) are receiving a significant interest, especially in the image analysis field due to their outstanding generative capabilities. This paper investigates whether or not GAN can be used as an oversampling tool to compensate for an unbalanced data set in an industrial manipulator fault diagnosis task. A comprehensive empirical analysis is performed taking into account six different scenarios for mitigating the unbalanced data, including classical under and oversampling (SMOTE) methods. In all of these, a wavelet packet transform is used for feature generation while a random forest is used for fault classification. Aspects such as loss functions, learning curves, random input distributions, data shuffling, and initial conditions were also considered. A non-parametric statistical test of hypotheses reveals that all GAN based fault-diagnosis outperforms both under and oversampling classical methods while, within GAN based methods, an average accuracy difference as high as 1.68% can be achieved.

Keywords: feature extraction; generative adversarial network; random forest; unbalance data; fault diagnosis

1. Introduction

For data-driven machine learning techniques [1], it plays an important role in the machinery fault diagnosis and prognostics [2,3]. Recently, deep learning [4] emerges as one that has been progressively adopted to develop health monitoring systems for the machinery. One way of looking at deep learning is as a feature engineering method [5] that automatically extracts features from the collected signals. Propagating these signals from the input layer to layers with less and fewer neurons, the neural network is forced to represent the input data space into a lower dimensional feature space, which, in general, reduces overfitting and increases the accuracy.

Deep learning models such as convolutional neural networks [6], deep autoencoders [7], deep belief networks [8], and deep Boltzmann machines [9] have achieved outstanding results in fault diagnosis, and other fields essentially due to the above-described dimensionality reduction effect. In the industrial manipulator field, Nho Cho et al. [10] proposed an algorithm based on the multilayer perceptrons for the manipulator actuator fault detection. Wang et al. [11] proposed a multi-data fusion with an optimized convolutional neural network (CNN) for fault detection of rotating machinery. Ma et al. [12] proposed the convolutional multi-time scale echo state network (ESN) to efficient classification. Three different ESNs with different time scales were used to allow the recurrent neural network to successively refine the features in a similar way of a kernel in a CNN. Due to the fast training of ESN, Long et al. [13] proposed a deep echo state network optimized by particle swarm optimization to fault diagnosis of a wind turbine gearbox. Hu et al. [14] present an approach with deep Boltzmann machine and multi-grained scanning forest to effectively deal with industrial fault diagnosis. Wang et al. [15] proposed a new deep neural network model based on a deep Boltzmann machine for condition prognosis. Lee et al. [16] proposed a real-time fault diagnosis model using a deep neural network. Shao et al. [17] presented a continuous deep belief network (CDBN) for bearings fault detection. Shen et al. [8] used a deep belief network with an optimized function of Nesterov momentum (NM) for bearing fault diagnosis. Polic et al. [18] presented a new method of a convolutional neural network encoder for feature extraction in tactile robotics. D'Elia et al. [19] based on the study of how the power flows inside the time synchronous average of the ring gear and a modified statistical parameter for planet gears fault diagnosis. Zaidan et al. [20] use a Bayesian hierarchical model with utilizing fleet data from multiple assets to perform probabilistic estimation of remaining useful life for civil aerospace gas turbine engines.

However, all the above models are critically dependent on a representative, balanced, large enough data set that require data that, more often than not, are hard to obtain [21]. While data from a healthy state are abundant, data from faulty states are rare, sparse, and hardly representative of all possible faults. This could lead to low diagnosis precision in these intelligent fault diagnosis techniques. Without appropriated data, these machine learning methods simply do not have acceptable performance [22], and it is important to consider a method that mitigates such a data shortage.

As Robotic manipulators, they are widely used in the industry as they can be used in a range of tasks such as assembly, painting, or welding [23]. However, the transmission system of a manipulator is prone to faults due to prolonged working periods [24]. Typically, these faults manifest in the connection parts, bearings, gears, or gear shafts. A faulty manipulator will be less precise, less efficient, less productive, and less secure—even though it could be a tricky task to obtain fault data for such precision machinery. Therefore, the monitoring of the manipulator health condition with limited data sources is of paramount interest.

For mitigating this, some fault diagnosis works resorted to the SMOTE. SMOTE stands for Synthetic Minority Oversampling Technique and aims at compensating for the data unbalance of a given class by increasing the number of samples in that class. Roughly, it creates a new synthetic sample by interpolating two existing similar samples of the same class, the new sample having the same class of the two originating samples [25]. SMOTE is a popular method but sometimes, by increasing the number of samples, it will also increase the overlapping between classes. An alternative approach to cope with the unbalanced data set problem is to resort to deep learning and in particular to generative adversarial networks (GAN). The idea is to use the generative model of a GAN to generate enough samples for effective training of the diagnoser. GAN was first proposed by Goodfellow [26] and consists of two adversarial models (neural networks): a generator and a discriminator.

The learning process can be described as a min-max game. The generator produces synthetic examples while the discriminator tries to decide whether the current input is either a real or a synthetic example. Both models improve their performance simultaneously up to a Nash equilibrium using a gradient-based optimization technique. The number of GAN applications has been steadily increasing. A sample of recent applications is offered next. In the image processing field, Ledig et al. [27] presented SRGAN, a generative adversarial network for image super-resolution. Isola et al. [28] demonstrated

that the conditional adversarial network is a promising approach for image-to-image translation tasks. Based on the above comment, Zhu et al. [29] presented an approach for learning to translate an image from a source domain to a target domain in the absence of paired examples. Shao et al. [30] develop an auxiliary classifier GAN (ACCGAN) to learn from mechanical sensor signals and generate realistic one-dimensional raw data. Li et al. [31] proposed a novel fault detection method for 3D printers using GAN which consider only normal condition signals with outstanding performance. Mao et al. [32] used a GAN for an unbalanced data driven fault diagnosis of rolling bearings. In addition, for rolling bearings, Jiang et al. [33] proposed a novel anomaly detection approach based on GAN with only health data. Li et al. [34] applied a GAN for the feature space learning in fault diagnosis of 3D printers using only one sample of each faulty state. Wang et al. [35] proposed a method based on a conditional variational auto-encoder and a generative adversarial network for unbalanced fault diagnosis of a planetary gearbox.

To the best of our knowledge, no work reported model based generation of synthetic samples for fault diagnosis of robotic manipulators. Hence, the main contributions of our work are: (1) The application of GAN to generate synthetic examples (signals) representing fault states for mitigating the presence of an unbalanced data set in a fault diagnosis task of an industrial robotic manipulator. More concretely, GAN generates a synthetic wavelet packet transform based feature of a vibrational signal as acquired by an accelerometer; (2) A comprehensive study taking into account six different scenarios for mitigating the unbalanced data, including classical under and oversampling (SMOTE) methods as well as for assessing the effect of factors such as generator selection, the number of training examples in each class, data shuffling in training data, the distribution used for sampling input random data and initial conditions.

The rest of this paper is organized as follows. The proposed GAN based fault diagnosis scheme is specified in Section 2. The manipulator experiment was presented in Section 3. The fault diagnosis of the manipulator was analyzed in Section 4 to validate the experiment result. Finally, the conclusions and the future work were detailed in Section 5.

2. Methodology

2.1. Feature Extraction

Wavelet packet transform (WPT) can be viewed as a time frequency conversion technique of a non-stationary signal [36]. It complements the shortage of wavelet transforms that only decomposes the low frequency components but cannot extract high-resolution on high frequency components. The discrete wavelet transform of the discrete signal $f(t)$ is given by [37]:

$$w_{m,n}(t) = \langle f(t), \Psi_{m,n}(t) \rangle = \frac{1}{\sqrt{m}} \int_{-\infty}^{+\infty} \Psi_{m,n}\left(\frac{t-n}{m}\right) f(t) dt, \quad (1)$$

where m is the scaling factor and n is the sifting factor, which are given, respectively, by:

$$m = a_0^m, \quad (2)$$

$$n = b_0 a_0^m. \quad (3)$$

When $a_0 = 2, b_0 = 1$, (1) can be re-written as

$$\Psi_{m,n}(t) = 2^{-\frac{m}{2}} \Psi(2^{-m}t - n) \quad (4)$$

In the wavelet transform, the signal $u(t)$ can be separated in the Hilbert space by a scaling and by a wavelet function. The scaling function $\Phi(t)$ corresponds to the low frequency part of the original

signal while the wavelet function $\varphi(k)$ corresponds to the high frequency part of the original signal with initial conditions:

$$\Psi_{m,n}^0(t) = \Phi(t) \quad (5)$$

$$\Psi_{m,n}^1(t) = \varphi(t) \quad (6)$$

Figure 1 shows a 3-level decomposition of a WPT of a signal $u(t)$. This is decomposed in a high-frequency part $h_k(t)$ and in a low-frequency part $g_k(t)$. Each part is computed by a filter, i.e.,

$$h_k(t) = \frac{t_k + t_{k+1}}{2}, \quad (7)$$

$$g_k(t) = \frac{t_k - t_{k+1}}{2}, \quad (8)$$

The function using the above filters can be given by:

$$u_{2n}(t) = \sum_k h_k u_n(2t - k), \quad (9)$$

$$u_{2n+1}(t) = \sum_k g_k u_n(2t - k) \quad (10)$$

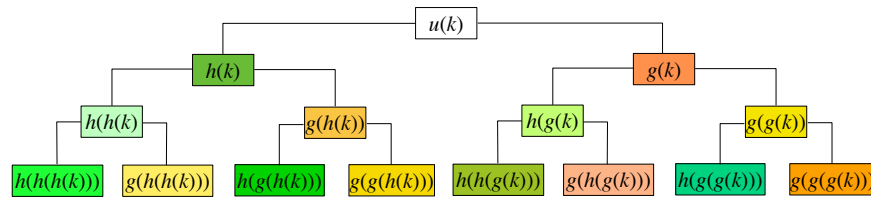


Figure 1. The decomposition levels of a wavelet packet transform of the signal $u(t)$.

As illustrated in the figure, this procedure can be recursively applied to both low and high-frequency parts. However, the number of decomposition levels will be limited by the actual application. Due to its smoothness and nonlinear characteristics, in this paper, we applied the Daubechies WPT with seven levels (Db7).

We further compute an informative statistics from the WPT as follows [38]:

$$p(m) = \sqrt{\sum_{n=1}^N (w_{m,n}(t))^2} \quad (11)$$

where N denotes the number of data in each node of the 7th decomposition level and $w_{m,n}(t)$ and is given by (1). Hence, a feature vector \mathbf{p} can be defined for the signal $u(t)$ as follows:

$$u(t) \leftrightarrow \mathbf{p} = [p(1), p(2), \dots, p(d)]^T \quad (12)$$

where d is the number of features, i.e., with seven levels $d = 2^7 = 128$.

2.2. Generative Adversarial Network

A Generative adversarial network consists of two models: the generative model $G(z)$ and the discriminative model $D(x)$. The goal of the generative model is to produce synthetic samples such that the discriminate model could not distinguish them from the real samples. At the same time, the objective of the discriminative model is to accept real samples and reject synthetic ones with the highest possible accuracy. In equilibrium, the discriminative model cannot identify the source of the

data, meaning that synthetic data are indistinguishable from real data. Figure 2 shows a block diagram of a GAN as used in this work

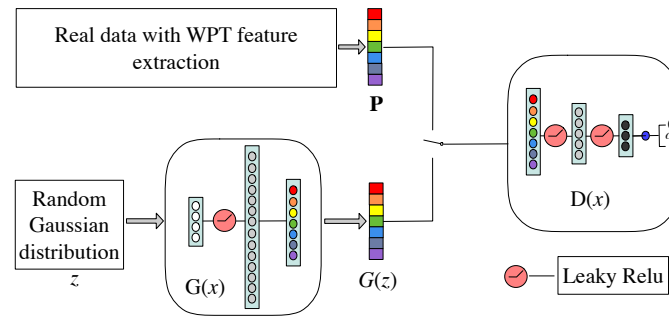


Figure 2. A block diagram of a GAN as used in this work.

For learning, a GAN implements an adversarial competition between the generator $G(z)$ and the discriminator $D(x)$. Initially, a real sample $u(t)$ is processed by the above described WPT feature extraction technique, and (12) is set as the real input of $D(x)$. A random signal $z(n)$ with a given distribution is input into the generator which in turn produces a synthetic feature vector $G(z)$. The discriminator is trained with a target value of 1 when a real sample is presented at its input, and with 0 for a synthetic example. This process repeats until a Nash equilibrium is reached. In general terms, the GAN optimization problem can be given by:

$$\mathcal{G}^* = \arg \min \max V(G, D) \quad (13)$$

where max stands for the maximization of the probability of the generator G while min refers to the minimization of the probability in the discriminator D ; $V(G, D)$ is the GAN objective function that can be given by:

$$V(G, D) = \mathbb{E}_{x \sim P_{data}} [\log(D(x))] + \mathbb{E}_{x \sim P_G} [\log(1 - D(G(z)))] \quad (14)$$

where $E_{x \sim P_{data}}$ represents the expectation of real probability distribution, whereas $E_{x \sim P_G}$ represents the expectation of the random distribution. Since x and z are real-value random variables on the probability space, the expect values of x and z can defined as the integral of x and z , respectively. Therefore, (14) can be re-written as

$$V(G, D) = \int_x [P_{data}(x) \log D(x) + P_G(x) \log(1 - D(x))] dx \quad (15)$$

where $\int_x P_{data}(x) dx$ stands for the expectation of $E_{x \sim P_{data}}(\cdot)$ and $\int_x P_G(x) dx$ denote the expectation of $E_{x \sim P_G}(\cdot)$. Let $f(x) = P_{data}(x) \log D(x) + P_G(x) \log(1 - D(x))$; then, the derivative of $f(x)$ is given by

$$\dot{f}(x) = \frac{P_{data}(x)}{\log D(x)} + \frac{P_G(x)}{\log(1 - D(x))} \quad (16)$$

where $P_{data}(x)$ is the distribution of the real data while $P_G(x)$ is the distribution of generated data. When $\dot{f} \rightarrow 0$, D tends to the maximum value, which is given by

$$D^* = \frac{P_{data}(x)}{P_{data}(x) + P_G(x)} \quad (17)$$

When $P_G(x) = 0$, D^* becomes 1 meaning that the discriminator can effectively recognize synthetic data, when $P_G(x)$ is close to $P_{data}(x)$, D^* tends to the optimal value of 0.5, which means that synthetic data are indistinguishable from real data. Plugging in (17) into (15), one has:

$$V(G, D^*) = \int_x [P_{data}(x) \log \frac{P_{data}(x)}{P_{data}(x) + P_G(x)} + P_G(x) \log(1 - \frac{P_{data}(x)}{P_{data}(x) + P_G(x)})] dx \quad (18)$$

As the objective of generative part is to shrink the distance between real and generated data, the loss function of generative model can be defined as

$$V(G, D^*) = \int_x [P_G(x) \log(1 - \frac{P_{data}(x)}{P_{data}(x) + P_G(x)})] dx \quad (19)$$

Under the above loss and, in general, the training process of a GAN is not stable and gives rise to model collapse. To mitigate this problem, the Wasserstein GAN was proposed where the KL divergence of the classic GAN was replaced by the 1-Wasserstein distance [39]. The Wasserstein GAN loss function is therefore given by:

$$V(G, D) = \mathbb{E}_{x \sim P_{data}} [D(x)] + \mathbb{E}_{x \sim P_G} [(D(G(z)))] \quad (20)$$

This change in the loss function can make the convergence of the generator faster, but it can be further improved. In [40], an improved Wasserstein GAN was proposed in which an additional gradient penalty was added to (20), i.e.,

$$\begin{aligned} V(G, D) = & \mathbb{E}_{x \sim P_{data}} [D(x)] + \mathbb{E}_{x \sim P_G} [(D(G(z)))] \\ & + \lambda \mathbb{E}[(|\delta D(\alpha x - (1 - \alpha G(z)))| - 1)^2] \end{aligned} \quad (21)$$

where α is a user-defined scaling factor and λ stands for the gradient penalty coefficient.

In another approach, Cabrera et al. [41] proposed a metric aiming at keeping track of the best current generator while training progresses. The metric is given by:

$$\|D_r - D_g\| = (R_{mean} - G_{mean})^2 + (R_{std} - G_{std})^2 \quad (22)$$

where R_{mean} and G_{mean} are the centroids of the real and generated data clusters, respectively, while R_{std} and G_{std} are the real and generated data dispersion (standard deviation), respectively. The smaller (22) the closer the generated data are from real data. In each training step, (22) is computed for the current generator, and the generator exhibiting the lower current distance is viewed as the best current generator. Hereafter, we refer to this process as (model) generator selection.

A Vapnik Loss Inspired GAN

The loss function is a key issue for model selection. Therefore, we are adopting a recently proposed loss function within the GAN and more concretely within the Wasserstein GAN framework. This loss function, first proposed by Vapnik et al. [42], considers the geometry distance between the predict and original data. In brief, the classical loss function used in regression, given a set of N examples, can be given by:

$$L_c = \frac{1}{N} \sum_{i=1}^N (h(x_i) - y_i)^2 + \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N (h(x_i) - y_i) I(h(x_i) - y_i) \quad (23)$$

where x_i and y_i are the i -th independent and dependent observation, respectively, $h(x)$ is the regressor hypothesis, and I is the identity matrix. Based on the VC theory, in [42], the identity matrix is replaced by the so-called V-matrix, i.e., (23) becomes:

$$L_v = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N (h(x_i) - y_i) V(h(x_i) - y_j) \quad (24)$$

where V is the V-matrix. For data in \mathbb{R}^d , the V-matrix can be computed for all $i, j = 1, \dots, d$ as

$$V(i, j) = \sum_{k=1}^d (c_k - \max(x_i^k - x_j^k)) \quad (25)$$

where d denotes the number of data dimensions, $0 \leq x_i \leq c_i$ and c_1, \dots, c_d are non-negative constants. This approach has shown good results in the regression problems. Motivated by both the theoretical background and the experimental results obtained in regression problems, including in the framework of SVR [43], we are proposing to apply this loss function in the framework of GAN as a regularization term in (22), which becomes:

$$Rf_v(G, D) = E_{x \sim P_{data}}[D(x)] + E_{x \sim P_G}[1D(x)] + \lambda E[(|\delta D(\alpha x - (1 - \alpha G(z)))| - 1)^2] + L_v \quad (26)$$

where L_v is of the form (24).

2.3. Random Forests for Fault Classification

Ensemble learning uses a group of algorithms to get a better prediction than any of its base algorithms. A random forest (RF) is a homogeneous ensemble classifier that uses a set of decision trees (DT) [44,45]. Each DT is grown independently using the Bagging technique. In addition, and to increase diversity (reducing the correlation between trees), an RF grows each tree from a random selection of data features. Once trained, an RF uses a majority voting mechanism for making its classification (or regression) prediction.

The CART algorithm is frequently used to grow a decision tree. In the CART algorithm, the Gini index is the metric used for selecting the data set feature to be used in a given node of the tree. Given a node m and the estimated probability $p(c|m)$ ($c = 1, 2, 3, \dots, C$), the Gini impurity index is defined as:

$$G(m) = \sum_{c_1 \neq c_2} p(c_1|m)p(c_2|m) = 1 - \sum_{c=1}^C p^2(c|m) \quad (27)$$

Let n be the splitting point of node m that separates the node into two portions in which a proportion P_a of the samples in m is assigned to m_a and a proportion P_b is assigned to m_b , i.e., $P_a + P_b = 1$. Thus, the decrease in the Gini impurity index is defined as follows:

$$\delta G(m, n) = G(m) - P_a G(m_a) - P_b G(m_b) \quad (28)$$

The optimal feature j^* and the optimal splitting point n^* that produce the largest decrease in the Gini impurity corresponds to

$$n^*, j^* = \arg \max \delta G(m, n) \quad (29)$$

The flowchart for building an RF is shown in Figure 3.

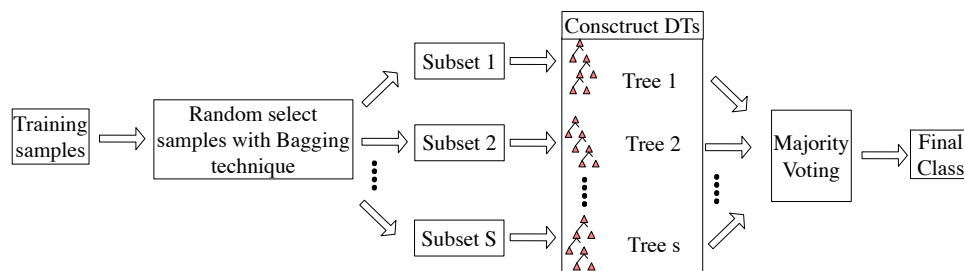


Figure 3. Steps for building a random forest.

2.4. Data Generation for Fault Classification

Based on the feature extraction process that uses Wavelet Packet Transform, illustrated in Section 2.1, the GAN network described in Section 2.2 and a random forest classifier detailed in the previous section, one can setup the learning scheme for the manipulator fault diagnoser. As shown in Figure 4, a real data observation of from each class is sent to the WPT to extract the vector of features (12). Meanwhile, a random signal z is input into the generator that will produce a vector of synthetic features $G(z)$.

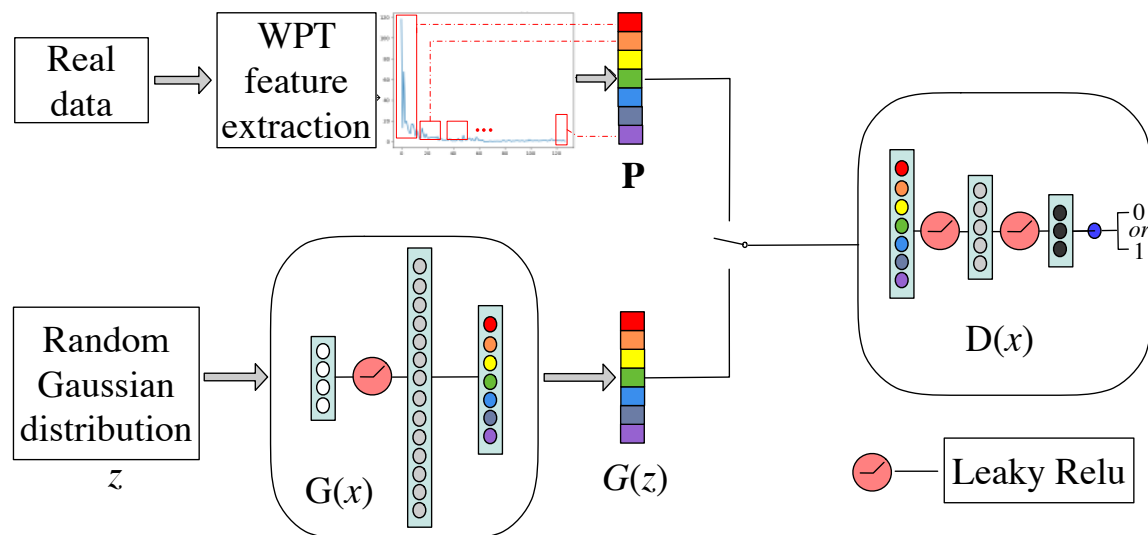


Figure 4. The learning scheme of the fault diagnoser.

The goal of the discriminator $D(x)$ is to distinguish between the real vector of features, outputting a 1, and the synthetic vector of features, outputting a 0. The learning process described in Section 2.2 is applied. Once both the generator and discriminator are trained, the generator can be used to generate as many synthetic data as required. Notice that the learning process and subsequent synthetic generation are carried out for each faulty class, i.e., for each class, we need to increase the number of observations. Finally, (health and faulty) real data together with faulty generated data are used in the random forest classifier for fault classification. Based on the above description, the procedure can be outlined in the following flow chart (Figure 5).

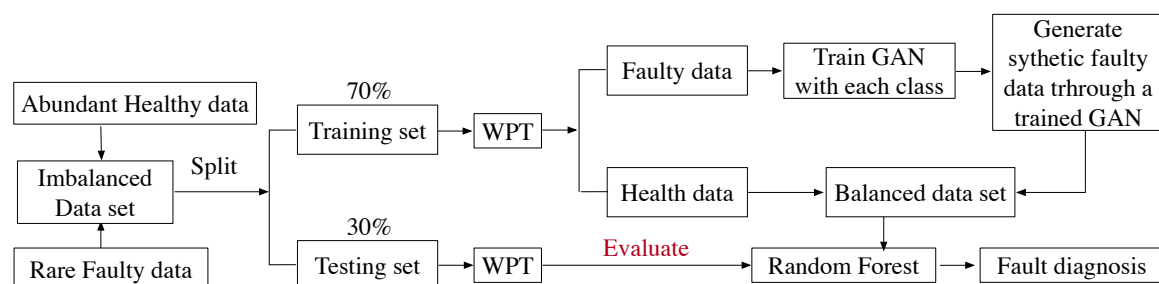


Figure 5. The flow chart of the procedure of the proposed approach.

In addition, this whole process is illustrated in Figure 6.

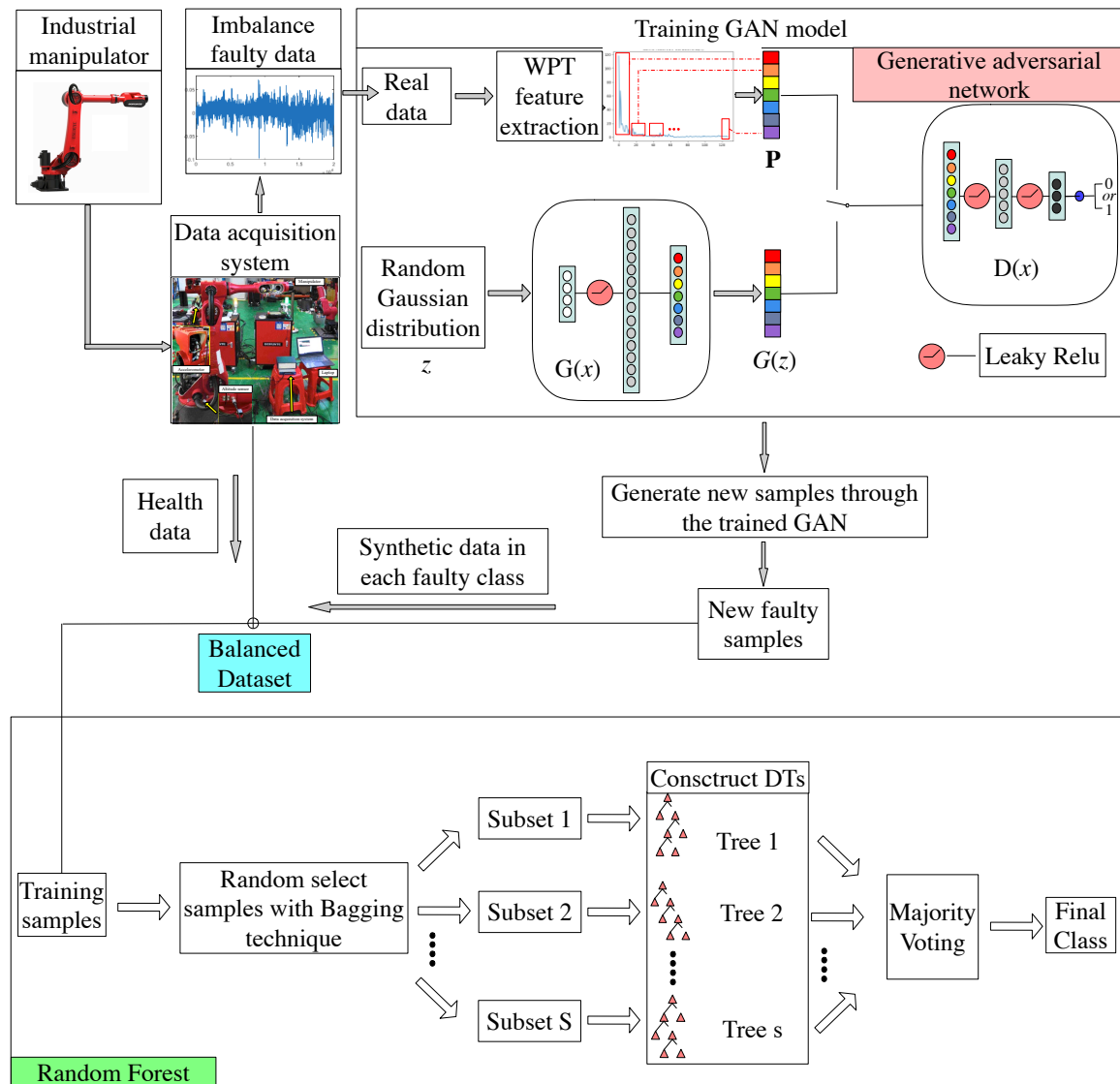


Figure 6. The complete data pipeline for fault diagnosis of the manipulator.

3. Experiment

3.1. Experimental Test-Rig

Experiments were carried out in the gear shaft system of a Brtirus1510A 6 degrees of freedom industrial manipulator. The gear system, which is the main driver component of the manipulator, consists of two planetary gears and two sun gears. The objective is to monitor the health state of gears by measuring the vibrational signals using an PCB 622B01 accelerometer. The accelerometer was installed on the basis of the sixth axis of the manipulator. See Figure 7 for its exact location. Cracking, pitting, and broken tooth are the main gear fault types in manipulators. Table 1 shows the fault type we are interested in. Figure 8 shows an example of each one of the four types of faults.

The robot is moved by the motors, and the teaching box gives the instructions to the robot to start its next movement. At the beginning of the process, the robot is in its original position of 0 degree. Firstly, it will start back and forth movement from -115 degrees to 140 degrees of the limit range point in the first axis. Secondly, the same movement and the same limit range which is from -50 degrees to 35 degrees. Thirdly, the robot will move from -60 degrees to 90 degrees. Fourthly, the same configuration of movement is from -180 degrees to 180 degrees. Fifthly, the movement range will be decreased such that the range is from -90 degrees to 90 degrees. Finally, the robot will move

from -180 degrees to 180 degrees and stop in the original place. This series of dynamic movement is only one experiment process. In the next step, we replace the faulty part in Table 1 to restart the above movement for the next experiment. Finally, the signal in each channel is collected by the NI acquisition system which is an analog-to-digital conversion system that the digital samples are collected with an interface on the laptop.

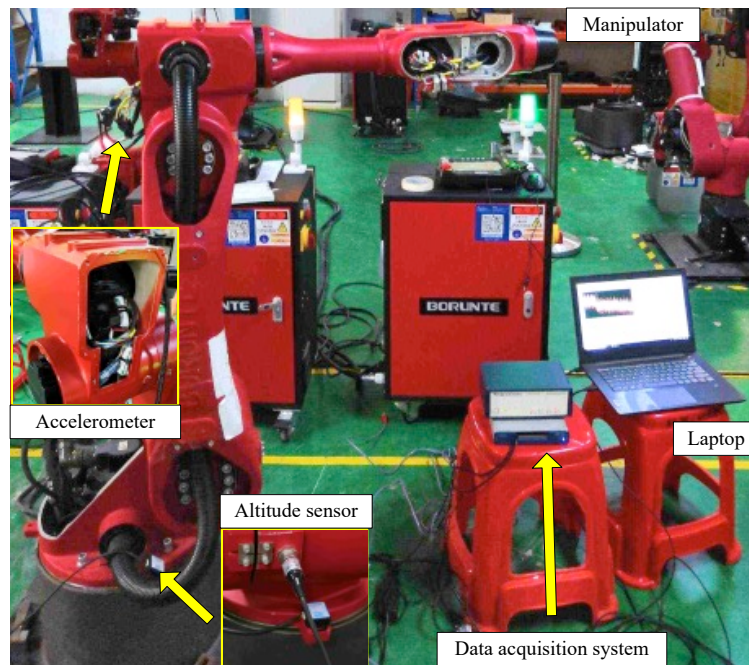


Figure 7. Monitoring the health state of the gears using vibrational signals acquired by an accelerometer located in the basis of the sixth axis of the manipulator.

Table 1. Different fault patterns in the industrial manipulator.

Fault Id	Part	Fault Type
A	None	Healthy
B	Sun gear 1	Pitting
C	Sun gear 1	Broken tooth
D	Planetary gear 1	Cracking

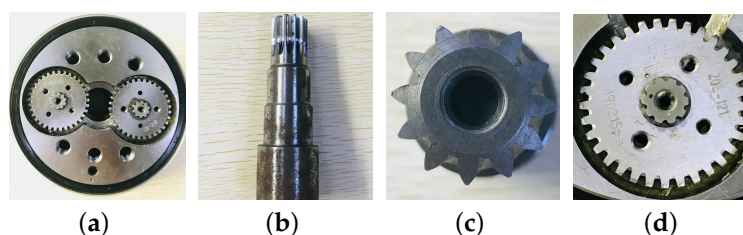


Figure 8. Examples of each one of the 4 monitoring conditions: (a) Healthy state; (b) Pitting in Sun gear 1; (c) Broken tooth in Sun gear 1; and (d) Cracking in Planetary gear 2.

3.2. Collected Vibration Signals

As mentioned above, the experimental measurements include four fault types shown in Table 1. The sampling rate is 100 kHz . The duration of each measurement is 20 s . The sampling interval was set to 0.2 s . Thus, $20,000$ observations were obtained in each fault type, and 20 k points are chosen for each observations. Therefore, a dataset of $80,000 \times 20\text{ k}$ can be acquired during the experiment. Figure 9 shows an example of a vibration signal acquired in each one of the fault types.

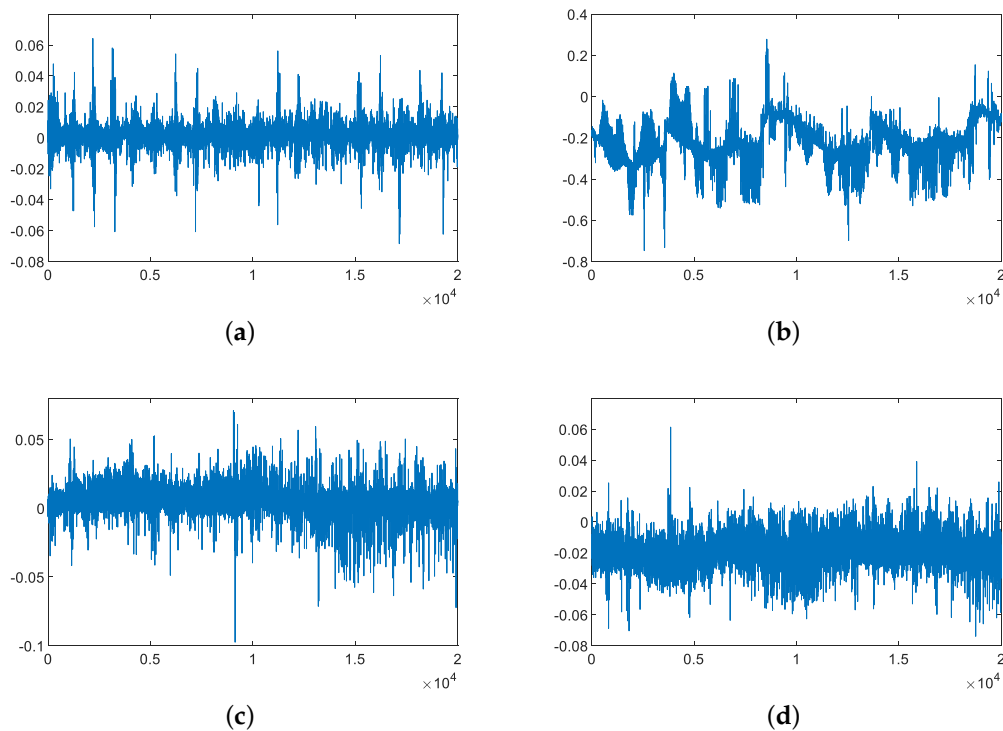


Figure 9. Vibration signals acquired in the following fault conditions: (a) Healthy state; (b) Pitting in Sun gear 1; (c) Broken tooth in Sun gear 1; and (d) Cracking in Planetary gear 2.

For hold-out validation, the data set was divided into two disjoint subsets, the training and the test (sub)sets. The training set has 70% of data while the test set has the remaining 30%. Table 2 shows the number of observations for each one of these sets.

Table 2. Number of observations in the training and test sets used for hold out validation. The training set is clearly unbalanced.

Fault Id	Training Set	Test Set
A	14,000	6000
B	140	6000
C	140	6000
D	140	6000

4. Results and Discussion

4.1. On Different Scenarios for Dealing with the Unbalanced Data Set

Several scenarios were considered to assess the effectiveness of the proposed model. In all these scenarios, we all used an RF with 1000 trees for classification. The different scenarios identified as follows: RF-i denotes a random forest trained with the unbalanced dataset described by Table 2; RF-b2 denotes a random forest trained with a subsampled balanced dataset with only 140 observations per condition; RF-GAN and RF-GAN2 stands rand forests trained with data sets that have the real 14,000 healthy observations and 14,000 faulty observations, the only difference between these scenarios is that RF-GAN uses the technique described in Section 2.2 to select the best current model for generating samples while RF-GAN2 uses the model obtained in the last iteration (i.e., iteration 20,000) of training process to generate the samples. RF-GAN1 is similar to RF-GAN with the difference that only 13,860 synthetic faulty states were generated while the remaining 140 are the original faulty states presented in the training data set. For comparison purposes, we also considered a random forest

trained with a data set previously processed by SMOTE, a popular oversampling method for dealing with unbalanced data sets.

In all GAN based models, the generators are multi-layer-perceptrons with a 64:1014:128 fully connected topology while discriminators are also multi-layer-perceptrons with 128:1024:2048:1 fully connected topology. These were selected empirically after some preliminary tests. The Adam optimizer is used with its key parameter settings of $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1 \times 10^{-8}$. The learning rate for the generators is set to 1×10^{-5} , while, for the discriminators, is set to 1×10^{-4} . The α and λ are set to 1×10^{-4} and 1.0, respectively. A maximum number of 10,000 iterations was set for training.

Figure 10 shows the distribution of the obtained accuracy for each scenario using boxplots. A boxplot summarizes a data distribution stressing five of its characteristic values: minimum, lower quartile, median, upper quartile, and maximum value. The red line denotes the median value. The distribution pertains to 20 independent repetitions.

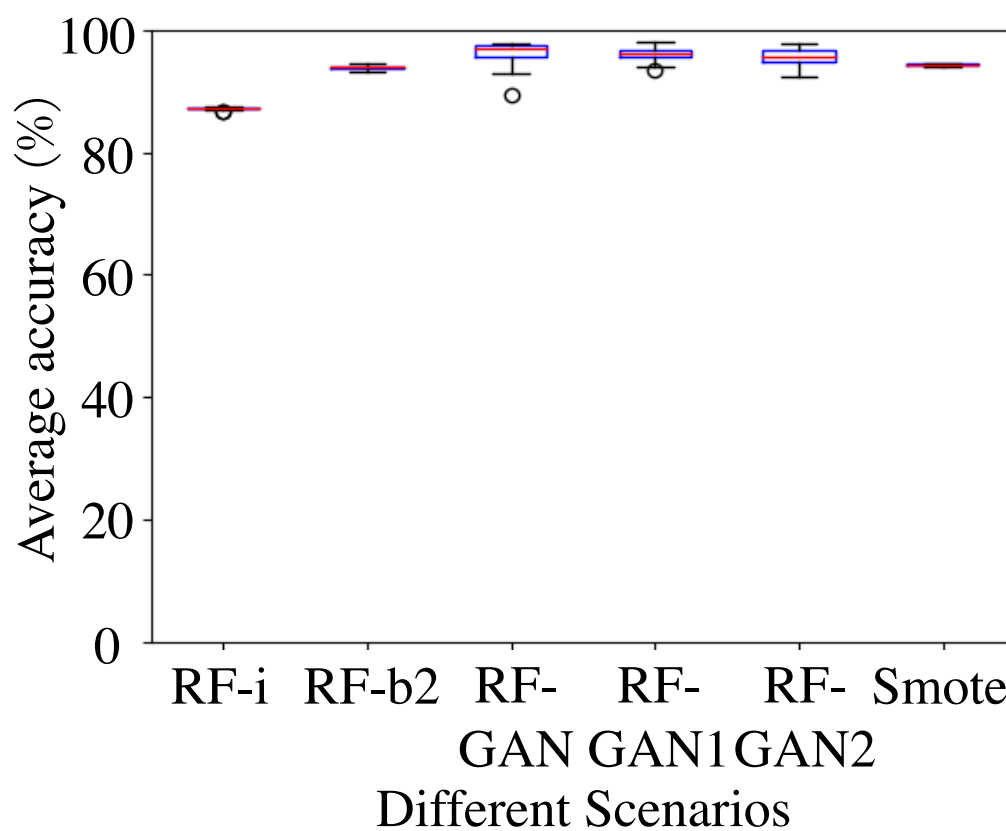


Figure 10. Boxplots exhibiting the relative distributions of accuracy obtained with the different scenarios considered for fault classification. See text for details.

The results presented in Figure 10 were analyzed by the Friedman test, a non parametric statistic test of hypotheses to evaluate whether or not there is a statistically significant difference between the results (boxplots) of the different scenarios. The Friedman null hypothesis is that there is no statistically significant difference between the results of the different scenarios. Given a significant level, α , this hypothesis cannot be rejected whenever the p_{Friedman} , the p -value generated by the test, satisfies $p_{\text{Friedman}} > \alpha$. The null hypothesis is rejected otherwise, meaning that there is a statistically significant difference between the analyzed scenarios. In such a case, we can detect which of the scenario is responsible for such difference resorting to a pairwise *posthoc* test. A ranking can be obtained by counting the number of times that a method was a winner in the pairwise comparison. See [46,47] for further details. Here, we are using the usual $\alpha = 0.05$ and the Wilcoxon test as *posthoc*.

Applying Friedman to the results in Figure 10 yielded $p_{\text{Friedman}} = 1.865 \times 10^{-19} < 0.05$, meaning there is a statistically significant difference between the six scenarios. Table 3 shows the subsequent *posthoc* results. From these, one should conclude that there is no statistically significant difference between scenarios RF-GAN and RF-GAN1 and that these outperform all the others. This is an interesting observation as both RF-GAN and RF-GAN1 use the technique described in Section 2.2 to select the best current generator and that the only difference between these scenarios is that in RF-GAN all the faulty state data are synthetic while in RF-GAN1 only 13,860 faulty examples are synthetic while the remaining 140 are the original faulty states presented in the training data set. This further endorses the quality of the obtained generators.

Table 3. Wilcoxon *posthoc* pairwise tests for the different scenarios.

Pair	<i>p</i> -Value	Winner
RF-i vs. RF-b2	8.857×10^{-5}	RF-b2
RF-i vs. RF-GAN	8.857×10^{-5}	RF-GAN
RF-i vs. RF-GAN1	8.857×10^{-5}	RF-GAN1
RF-i vs. RF-GAN2	8.857×10^{-5}	RF-GAN2
RF-i vs. SMOTE	8.857×10^{-5}	SMOTE
RF-b2 vs. RF-GAN	8.857×10^{-5}	RF-GAN
RF-b2 vs. RF-GAN1	8.857×10^{-5}	RF-GAN1
RF-b2 vs. RF-GAN2	8.857×10^{-5}	RF-GAN2
RF-b2 vs. SMOTE	1.204×10^{-5}	SMOTE
RF-GAN vs. RF-GAN1	0.079	—
RF-GAN vs. RF-GAN2	8.857×10^{-5}	RF-GAN
RF-GAN vs. SMOTE	8.857×10^{-5}	RF-GAN
RF-GAN1 vs. RF-GAN2	8.845×10^{-5}	RF-GAN1
RF-GAN1 vs. SMOTE	8.844×10^{-5}	RF-GAN1
		RF-GAN/RF-GAN1

The average accuracy of RF-i was 87.88% while, with an undersampling balanced dataset, RF-b2 reached 94.5%. RF-GAN had an average accuracy of 97.06%, RF-GAN1 97.75%, and RF-GAN2 95.38%. SMOTE had an averaged accuracy of 95.17%. We observe that the GAN based average accuracies are all higher than the unbalanced (RF-i), undersampling (RF-b2), and oversampling (SMOTE) scenarios. Within the GAN based scenarios, RF-GAN has shown a difference of 1.68% relatively to RF-GAN2 in terms of average accuracy.

Curiously enough, we observed no advantage on the application of (24). For the moment, we keep in mind that (26) is not the only possibility to use the Vapnik-loss in a GAN and that other forms are currently being studied.

4.2. On of the Performance in Each Class

To further analyze the above results, the recall indicator of each fault class is studied. As it is shown in Figure 11, the recall indicator in the RF-i model of the health state reaches 100% while, for the other three faulty classes, comes down to 63.88% in gear pitting; 84.3% in gear broken tooth, and 63.35% in gear cracking for an average of 77.88%. This clearly shows the effect of the unbalanced data set. For RF-GAN, the recall indicator in each class is 99.53%, 99.73%, 99.63%, and 99.87%, respectively. In RF-GAN1, the recall indicator in each class is 98.63%, 98.67%, 98.28%, and 95.40%, respectively. For the RF-GAN2 model, the recall indicator in each class is 99.30%, 93.08%, 98.15%, and 90.08%, respectively. The high recall indicators are due to the existence of sufficient examples in each class. This is also visible in the recall of SMOTE.

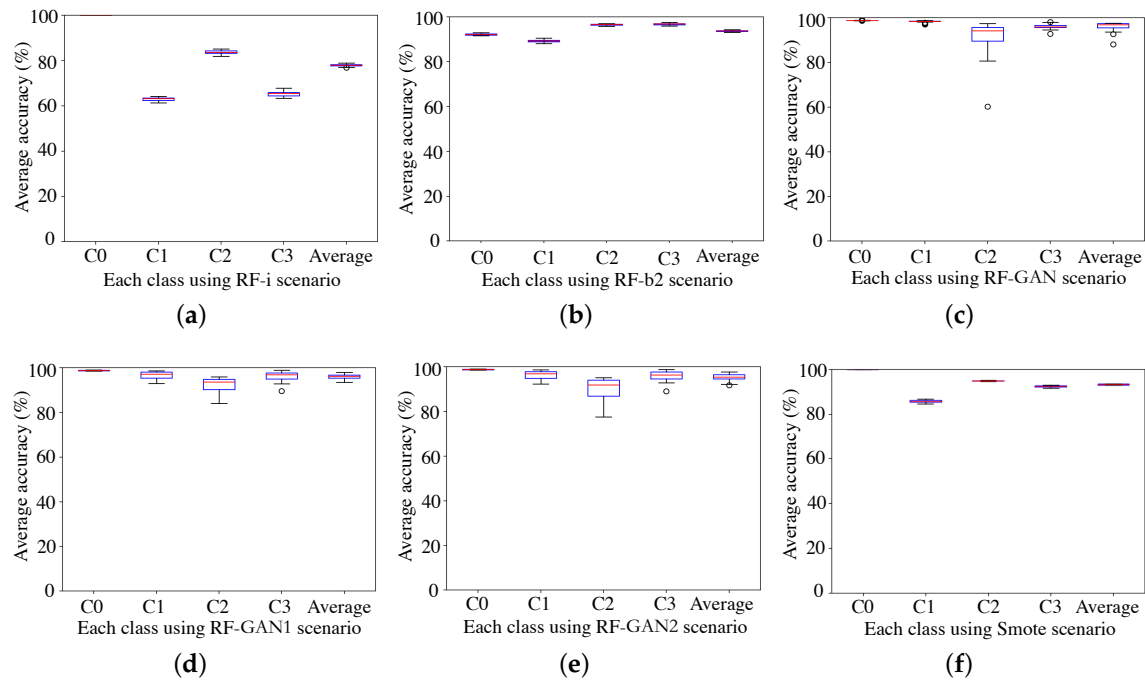


Figure 11. Recall indicators for the different scenarios: (a) RF-i; and (b) RF-b2; (c) RF-GAN; (d) RF-GAN1; (e) RF-GAN2; and (f) SMOTE.

The fit score is another metric that can be used to compare the relative performance of the different scenarios in each class. From Figure 12, one can see that the performance of RF-i as measured by the F1-score is 70.38% in fault A, 77.54% in fault B, 90.03% in fault C, and 76.49% in fault D. The performance of RF-b2 in each faulty condition is 91.05% in fault A, 92.25% in fault B, 96.50% in fault C and 94.3% in fault D, respectively. The performance of RF-GAN is 98.52% in fault A, 98.83% in fault B, 95.66% in fault C and 95.23 in fault D. The performance of RF-GAN1 is 98.55% in fault A, 98.95% in fault B, 96.87% in fault C and 96.62 in fault D. The performance of RF-GAN2 is 95.70% in fault A, 96.54% in fault B, 90.01% in fault C, and 94.57% in fault D. For SMOTE, the performance is 89.72% in fault A, 93.08% in fault B, 97.56% in fault C, and 95.28% in fault D.

For completeness, the confusion matrices are presented in Figure 13. All these matrices consider the 6000 test observations as per Table 2. For RF-i (Figure 13a), one can see a high number of misclassifications in non-healthy states due to the unbalanced training set. These misclassifications are strongly reduced (especially in the GAN-based scenarios) when enough data are generated and used for training.

4.3. On the Training Set

4.3.1. Learning Curves

The performance of the proposed model under different data set sizes is now considered. Figure 14 shows the average performance over 20 independent runs in the testing set for the scenario RF-GAN when only a given percentage of faulty observations are available for training. More concretely, the following percentages were considered $i = 1, 2, 4, 6, 8, 10, 20, 60, 80$, and 100 %. For instance, when $i = 4\%$, the number of training examples in each faulty state is $14,000 \times 0.04 = 560$.

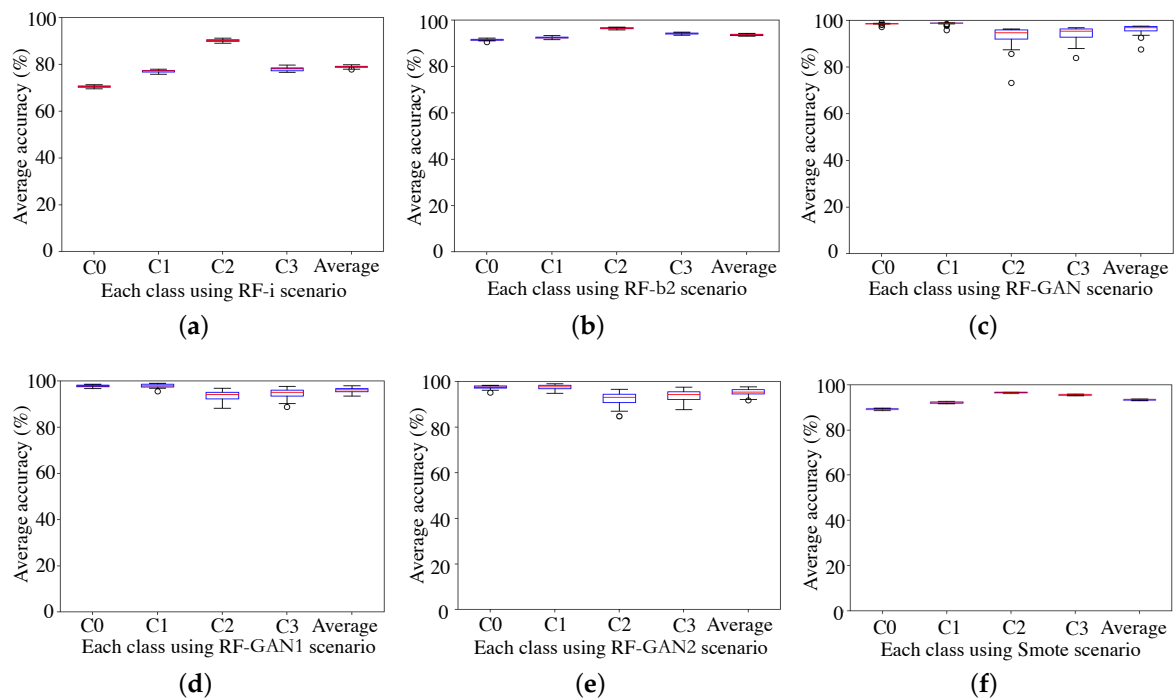


Figure 12. F1-score for scenario: (a) RF-i; and (b) RF-b2; (c) RF-GAN; (d) RF-GAN1; (e) RF-GAN2; and (f) SMOTE.

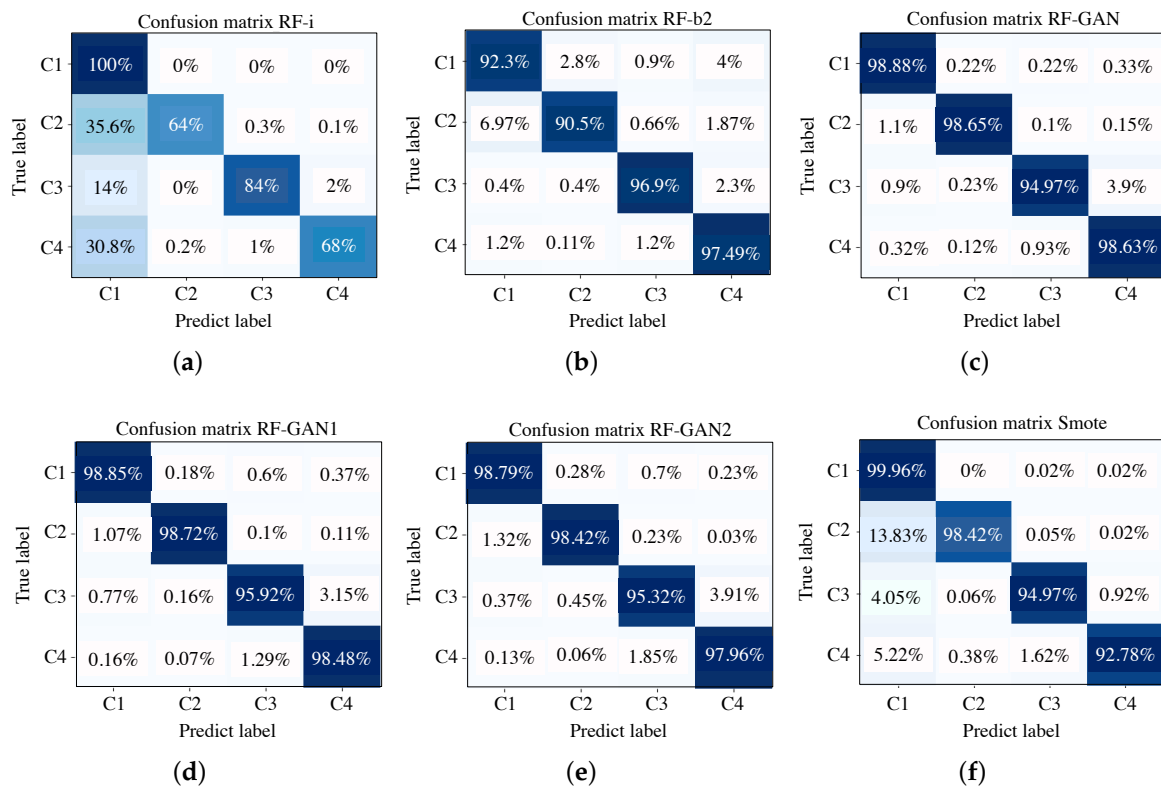


Figure 13. The confusion matrix for: scenario: (a) RF-i; and (b) RF-b2; (c) RF-GAN; (d) RF-GAN1; (e) RF-GAN2; and (f) SMOTE.

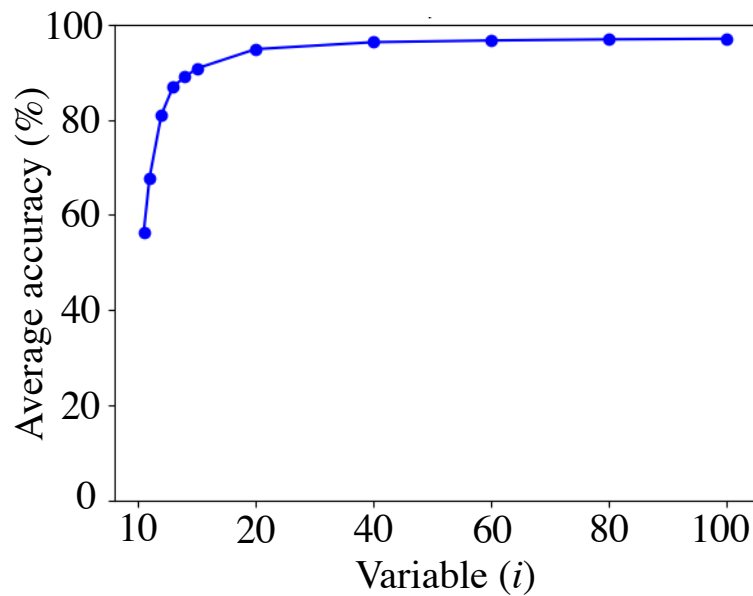


Figure 14. Learning curve of scenario RF-GAN for $i = 1, 2, 4, 6, 8, 10, 20, 60, 80$, and 100% of the training set.

It can be seen that the average accuracy increased from 56.25% to 97.05% by increasing the availability of faulty data from 1% (140 examples) to 100% (14,000 examples) in each fault type. There was a strong increased in performance up to 20%; after that point, the improvement in accuracy was slower and slower until about 80%. After this value, the improvement was neglectable. That is, adding more data after a certain point hardly improves the performance.

4.3.2. Shuffling Data

When generating training examples from a GAN based model, shuffling the training data are a key important factor for obtaining an acceptable performance. Figure 15 illustrates the importance of data shuffling. The results presented in this figure were obtained with exactly the same RF-GAN configuration, the only difference being the way data are presented to the GAN for training. For non shuffled data, the model is simply not able to generate no matter the other initial conditions (weights).

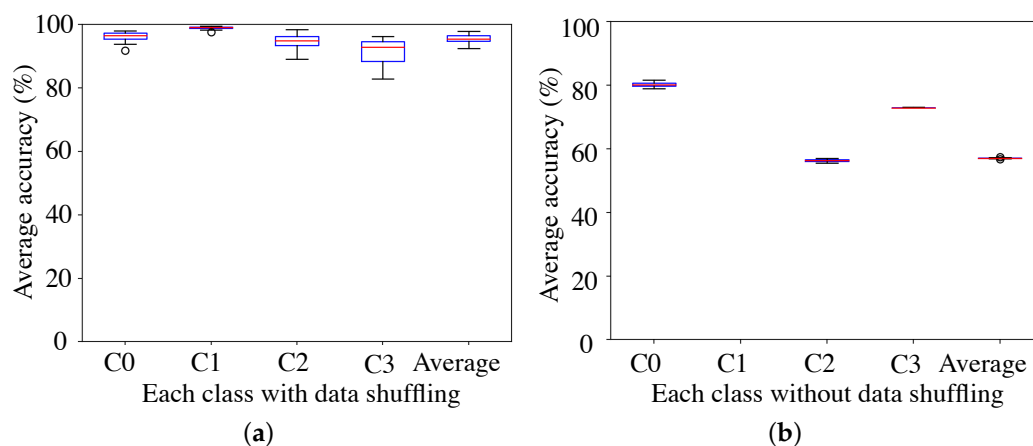


Figure 15. The effect of shuffling input data for training a GAN based model: (a) with shuffling and (b) without shuffling.

4.4. On the Distribution Used for Sampling Random Inputs

In a GAN based model, the z signal presented to the generator (recall Figure 4) can be drawn from any distribution. However, for this particular application, some distributions are better than others for the training process. Figure 16 shows the classification results for RF-GAN when (a) z is sampled from the standard normal distribution (0 mean and variance 1) and (b) a uniform distribution in $[-1,1]$. Undoubtedly, the former outperforms the latter.

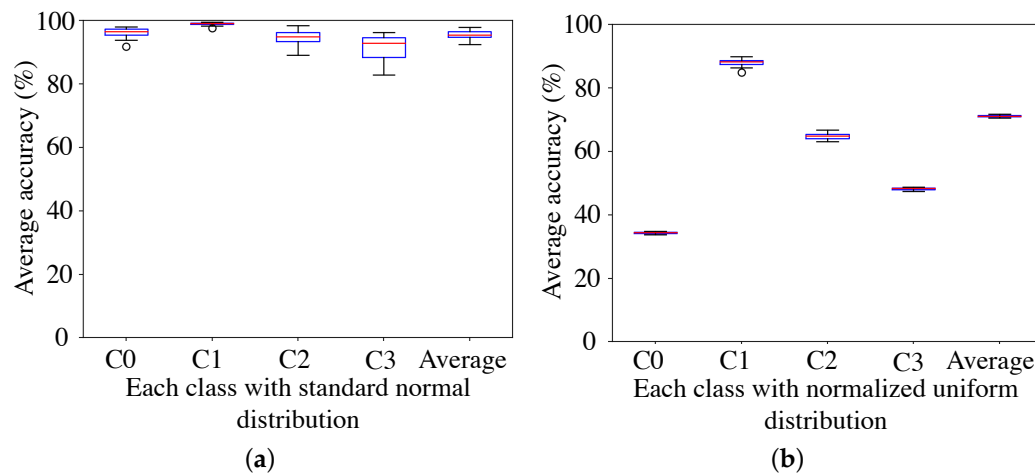


Figure 16. The effect of the distribution used for sampling the input z of the GAN generator: (a) standard normal distribution and (b) normalized uniform distribution.

4.5. On the Initial Conditions

GAN is trained using a gradient based method that is sensitive to initial conditions (weights). Figure 17 illustrates the impact of the initial conditions on the fault classification results. As shown in that figure, RF-GAN was able to produce an acceptable accuracy for any of the initial set of weights used. However, and as expected for local optimization methods, in 30 independent runs (initializations), it was possible to identify a particular set of initial random weights that outperformed all the others.

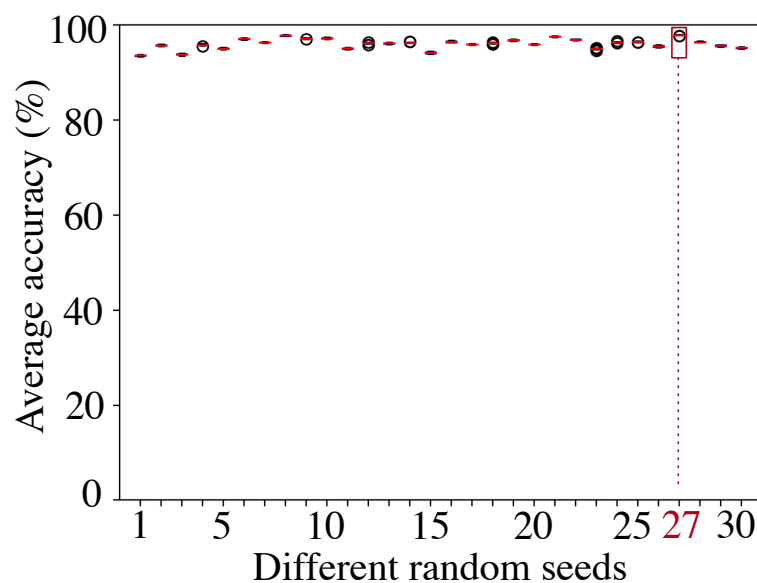


Figure 17. The effect of initial weights (generated from different random generator seeds) on the classification accuracy of RF-GAN.

5. Conclusions

Robotic manipulators are widely used in the industry and their maintenance and monitoring systems are resorting more and more to data intensive machine learning methods. Methods such as multilayer perceptrons, convolutional neural network, echo state networks, or deep Boltzmann machines have all been used for such endeavors. However, all of these methods rely on a representative, balanced, and large enough training set, which, due to the very nature of (some) faults, is very hard to collect from the equipment. Motivated by the recent success of generative adversarial network (GAN), in this work, we have exploited, for the first time, this type of generative model as an oversampling method for fault classification in an industrial robotic manipulator.

A comprehensive empirical analysis was performed taking into account six different scenarios for mitigating the unbalanced data, including classical under and oversampling (SMOTE) methods. In all of these, a wavelet packet transform combined with GAN is used for feature generation while a random forest is used for fault classification. Studies were also conducted for assessing the sensibility of aspects such as generator selection, the number of training examples in each class, training data shuffling, the distribution used for sampling input random data, and initial conditions.

The main conclusion is that it was possible to increase the performance of the fault diagnoser for an industrial robotic manipulator for any of GAN based models over classical undersampling and oversampling (SMOTE) methods. This is accomplished at the expense of a much higher design and computational effort. Training a GAN is not an easy task due to the model collapse and other factors, and it is certainly a quite time-consuming process. After training a GAN for each fault, one will have a set of generators able to produce as much synthetic data as required in an efficient way though.

Within the GAN based models, those that keep track of the best current generator during training yielded the best results. No statistically significant difference was observed between the scenario that uses exclusively synthetic data for the faulty states and the scenario that uses the available real data for such states. This is yet another piece of evidence on the quality of the obtained GAN generators.

In many cases like prognostics and health management (PHM), enough data can effectively improve the fault monitoring capability of the industrial system. However, it can not completely be executed due to a lack of faulty data. GAN is an efficient tool to get rid of the limitation of data imbalance state, which can enhance the monitoring capability in PHM. Therefore, this approach can provide a data background for PHM. However, this approach still has limitations, one is that this model can only learn the data distribution from a limited faulty data source while there are many kinds of faulty data in the industrial system. For the new faulty data, they need to be sent to this model to learn the new distribution again to generate enough data that will bring time cost for training. Another one is that this approach is trained by sending one single faulty class as an input to GAN in order to obtain a generator. This means we need to train several GAN models for several faulty classes, which is computationally demanding and time consuming.

Author Contributions: Writing—original draft preparation, and investigation, Z.P.; methodology, D.C.; writing—review and supervision, J.V.d.O.; funding acquisition, C.L.; investigation, R.-V.S.; investigation, M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is financed by national funds through FCT—Foundation for Science and Technology, I.P., through IDMEC, under LAETA, project UIDB/50022/2020. This work was supported in part by the National Natural Science Foundation of China under Grant 51775112, the Natural Science Foundation of Chongqing cstc2019cyj-zdxmX003, the CTBU Project under Grant KFJJ2019060.

Acknowledgments: The authors are thankful for the finance support from FCT—Foundation for Science and Technology, I.P., through IDMEC, under LAETA and in part to the National Natural Science Foundation of China.

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

References

1. Jin, X.; Que, Z.; Sun, Y.; Guo, Y.; Qiao, W. A data-driven approach for bearing fault prognostics. *IEEE Trans. Ind. Appl.* **2019**, *55*, 3394–3401. [\[CrossRef\]](#)
2. Schmidt, S.; Heyns, P.S.; Gryllias, K.C. A discrepancy analysis methodology for rolling element bearing diagnostics under variable speed conditions. *Mech. Syst. Signal Process.* **2019**, *116*, 40–61. [\[CrossRef\]](#)
3. Zaidan, M.A.; Harrison, R.F.; Mills, A.R.; Fleming, P.J. Bayesian hierarchical models for aerospace gas turbine engine prognostics. *Expert Syst. Appl.* **2015**, *42*, 539–553. [\[CrossRef\]](#)
4. He, M.; He, D. Deep learning based approach for bearing fault diagnosis. *IEEE Trans. Ind. Appl.* **2017**, *53*, 3057–3065. [\[CrossRef\]](#)
5. Liu, H.; Liu, C.; Huang, Y. Adaptive feature extraction using sparse coding for machinery fault diagnosis. *Mech. Syst. Signal Process.* **2011**, *25*, 558–574. [\[CrossRef\]](#)
6. Wu, C.; Jiang, P.; Ding, C.; Feng, F.; Chen, T. Intelligent fault diagnosis of rotating machinery based on one-dimensional convolutional neural network. *Comput. Ind.* **2019**, *108*, 53–61. [\[CrossRef\]](#)
7. Zhang, Y.; Li, X.; Gao, L.; Chen, W.; Li, P. Intelligent fault diagnosis of rotating machinery using a new ensemble deep auto-encoder method. *Measurement* **2020**, *151*, 107232. [\[CrossRef\]](#)
8. Shen, C.; Xie, J.; Wang, D.; Jiang, X.; Shi, J.; Zhu, Z. Improved hierarchical adaptive deep belief network for bearing fault diagnosis. *Appl. Sci.* **2019**, *9*, 3374. [\[CrossRef\]](#)
9. Deng, S.; Cheng, Z.; Li, C.; Yao, X.; Chen, Z.; Sanchez, R.V. Rolling bearing fault diagnosis based on Deep Boltzmann machines. In Proceedings of the 2016 Prognostics and System Health Management Conference (PHM-Chengdu), Chengdu, China, 19–21 October 2016; pp. 1–6.
10. Cho, C.N.; Hong, J.T.; Kim, H.J. Neural network based adaptive actuator fault detection algorithm for robot manipulators. *J. Intell. Robot. Syst.* **2019**, *95*, 137–147. [\[CrossRef\]](#)
11. Wang, H.; Li, S.; Song, L.; Cui, L. A novel convolutional neural network based fault recognition method via image fusion of multi-vibration-signals. *Comput. Ind.* **2019**, *105*, 182–190. [\[CrossRef\]](#)
12. Ma, Q.; Chen, E.; Lin, Z.; Yan, J.; Yu, Z.; Ng, W.W. Convolutional Multitimescale Echo State Network. *IEEE Trans. Cybern.* **2019**. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Long, J.; Zhang, S.; Li, C. Evolving deep echo state networks for intelligent fault diagnosis. *IEEE Trans. Ind. Inform.* **2019**, *16*, 4928–4937. [\[CrossRef\]](#)
14. Hu, G.; Li, H.; Xia, Y.; Luo, L. A deep Boltzmann machine and multi-grained scanning forest ensemble collaborative method and its application to industrial fault diagnosis. *Comput. Ind.* **2018**, *100*, 287–296. [\[CrossRef\]](#)
15. Wang, J.; Wang, K.; Wang, Y.; Huang, Z.; Xue, R. Deep Boltzmann machine based condition prediction for smart manufacturing. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 851–861. [\[CrossRef\]](#)
16. Lee, K.P.; Wu, B.H.; Peng, S.L. Deep-learning-based fault detection and diagnosis of air-handling units. *Build. Environ.* **2019**, *157*, 24–33. [\[CrossRef\]](#)
17. Shao, H.; Jiang, H.; Li, X.; Liang, T. Rolling bearing fault detection using continuous deep belief network with locally linear embedding. *Comput. Ind.* **2018**, *96*, 27–39. [\[CrossRef\]](#)
18. Polic, M.; Krajacic, I.; Lepora, N.; Orsag, M. Convolutional autoencoder for feature extraction in tactile sensing. *IEEE Robot. Autom. Lett.* **2019**, *4*, 3671–3678. [\[CrossRef\]](#)
19. D’Elia, G.; Mucchi, E.; Coconcelli, M. On the identification of the angular position of gears for the diagnostics of planetary gearboxes. *Mech. Syst. Signal Process.* **2017**, *83*, 305–320. [\[CrossRef\]](#)
20. Zaidan, M.A.; Mills, A.R.; Harrison, R.F.; Fleming, P.J. Gas turbine engine prognostics using Bayesian hierarchical models: A variational approach. *Mech. Syst. Signal Process.* **2016**, *70*, 120–140. [\[CrossRef\]](#)
21. Ali, J.B.; Fnaiech, N.; Saidi, L.; Chebel-Morello, B.; Fnaiech, F. Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals. *Appl. Acoust.* **2015**, *89*, 16–27.
22. Yan, K.; Ji, Z.; Lu, H.; Huang, J.; Shen, W.; Xue, Y. Fast and accurate classification of time series data using extended ELM: Application in fault diagnosis of air handling units. *IEEE Trans. Syst. Man, Cybern. Syst.* **2017**, *49*, 1349–1356. [\[CrossRef\]](#)
23. Iqbal, J.; Islam, R.U.; Abbas, S.Z.; Khan, A.A.; Ajwad, S.A. Automating industrial tasks through mechatronic systems—A review of robotics in industrial perspective. *Teh. Vjesn.* **2016**, *23*, 917–924.

24. Caccavale, F.; Cilibrizzi, P.; Pierri, F.; Villani, L. Actuators fault diagnosis for robot manipulators with uncertain model. *Control Eng. Pract.* **2009**, *17*, 146–157. [\[CrossRef\]](#)
25. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [\[CrossRef\]](#)
26. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems 27, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
27. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.
28. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
29. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
30. Shao, S.; Wang, P.; Yan, R. Generative adversarial networks for data augmentation in machine fault diagnosis. *Comput. Ind.* **2019**, *106*, 85–93. [\[CrossRef\]](#)
31. Li, C.; Cabrera, D.; Sancho, F.; Sánchez, R.V.; Cerrada, M.; Long, J.; de Oliveira, J.V. Fusing convolutional generative adversarial encoders for 3D printer fault detection with only normal condition signals. *Mech. Syst. Signal Process.* **2020**, *147*, 107108. [\[CrossRef\]](#)
32. Mao, W.; Liu, Y.; Ding, L.; Li, Y. Imbalanced fault diagnosis of rolling bearing based on generative adversarial network: A comparative study. *IEEE Access* **2019**, *7*, 9515–9530. [\[CrossRef\]](#)
33. Jiang, W.; Hong, Y.; Zhou, B.; He, X.; Cheng, C. A GAN-based anomaly detection approach for imbalanced industrial time series. *IEEE Access* **2019**, *7*, 143608–143619. [\[CrossRef\]](#)
34. Li, C.; Cabrera, D.; Sancho, F.; Sánchez, R.V.; Cerrada, M.; de Oliveira, J.V. One-shot fault diagnosis of 3D printers through improved feature space learning. *IEEE Trans. Ind. Electron.* **2020**, *147*, 107108.
35. Wang, Y.R.; Sun, G.D.; Jin, Q. Imbalanced sample fault diagnosis of rotating machinery using conditional variational auto-encoder generative adversarial network. *Appl. Soft Comput.* **2020**, *92*, 106333. [\[CrossRef\]](#)
36. Gokhale, M.; Khanduja, D.K. Time domain signal analysis using wavelet packet decomposition approach. *Int. J. Commun. Netw. Syst. Sci.* **2010**, *3*, 321. [\[CrossRef\]](#)
37. Bruce, L.M.; Koger, C.H.; Li, J. Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction. *IEEE Trans. Geosci. Remote Sens.* **2002**, *40*, 2331–2338. [\[CrossRef\]](#)
38. Li, C.; Sanchez, R.V.; Zurita, G.; Cerrada, M.; Cabrera, D.; Vásquez, R.E. Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals. *Mech. Syst. Signal Process.* **2016**, *76*, 283–293. [\[CrossRef\]](#)
39. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein gan. *arXiv* **2017**, arXiv:1701.07875.
40. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved training of wasserstein gans. In Proceedings of the Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 4–9 December 2017; pp. 5767–5777.
41. Cabrera, D.; Sancho, F.; Long, J.; Sánchez, R.V.; Zhang, S.; Cerrada, M.; Li, C. Generative adversarial networks selection approach for extremely imbalanced fault diagnosis of reciprocating machinery. *IEEE Access* **2019**, *7*, 70643–70653. [\[CrossRef\]](#)
42. Vapnik, V.; Izmailov, R. Rethinking statistical learning theory: Learning using statistical invariants. *Mach. Learn.* **2019**, *108*, 381–423. [\[CrossRef\]](#)
43. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [\[CrossRef\]](#)
44. Friedl, M.A.; Brodley, C.E. Decision tree classification of land cover from remotely sensed data. *Remote Sens. Environ.* **1997**, *61*, 399–409. [\[CrossRef\]](#)

- 45. Liaw, A.; Wiener, M. Classification and regression by randomForest. *R News* **2002**, *2*, 18–22.
- 46. Demsar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
- 47. Pacheco, F.; Valente de Oliveira, J.; Sánchez, R.V.; Cerrada, M.; Cabrera, D.; Li, C.; Zurita, G.; Artés, M. A statistical comparison of neuroclassifiers and feature selection methods for gearbox fault diagnosis under realistic conditions. *Neurocomputing* **2016**, *194*, 192–206. [[CrossRef](#)]

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).