Journal of Nondestructive Evaluation (2024) 43:114
https://doi.org/10.1007/s10921-024-01131-3

Multiple Fault Diagnosis in a Wind Turbine Gearbox with Autoencoder Data Augmentation and KPCA Dimension Reduction

Leonardo Oldani Felix[1] · Dionísio Henrique Carvalho de Sá Só Martins[1] · Ulisses Admar Barbosa Vicente Monteiro[1] · Luiz Antonio Vaz Pinto[1] · Luís Tarrataca[2] · Carlos Alfredo Orfão Martins[1]

Received: 5 June 2024 / Accepted: 23 September 2024

Abstract

Gearboxes, as critical components, often operate in demanding conditions, enduring constant exposure to variable loads and speeds. In the realm of condition monitoring, the dataset primarily comprises data from normal operating conditions, with significantly fewer instances of faulty conditions, resulting in imbalanced datasets. To address the challenges posed by this data disparity, researchers have proposed various solutions aimed at enhancing the performance of classification models. One such solution involves balancing the dataset before the training phase through oversampling techniques. In this study, we utilized the Sparse Autoencoder technique for data augmentation and employed Support Vector Machine (SVM) and Random Forest (RF) for classification. We conducted four experiments to evaluate the impact of data imbalance on classifier performance: (1) using the original dataset without data augmentation, (2) employing partial data augmentation, (3) applying full data augmentation, and (4) balancing the dataset while using Kernel Principal Component Analysis (KPCA) for dimensionality reduction. Our findings revealed that both algorithms achieved accuracies exceeding 90%, even when employing the original non-augmented data. When partial data augmentation was employed both algorithms were able to achieve accuracies beyond 98%. Full data augmentation yielded slightly better results compared to partial augmentation. After reducing dimensions from 18 to 11 using KPCA, both classifiers maintained robust performance. SVM achieved an overall accuracy of 98.72%, while RF achieved 96.06% accuracy.

Keywords Gearbox · Fault diagnosis · Autoencoder · KPCA · SVM · Random Forest

1 Introduction

Gearboxes serve as vital components in a myriad of transmission devices, including helicopter transmission reducers, the main propulsion reducer of a ship, and wind power generator sets, among others. The wind power industry stands out as a promising sector within renewable energy, representing a predominant approach to harnessing renewable resources [1, 2].

The challenging operating conditions in wind turbine gearboxes frequently lead to failures of critical components, such as gears and bearings. Statistical data indicate that gearbox failures often cause the most substantial economic losses and extended downtime for wind turbines [1]. The challenging operating conditions in wind turbine gearboxes frequently lead to failures of critical components, such as gears and bearings. Statistical data indicate that gearbox failures often cause the most substantial economic losses and extended downtime for wind turbines [2].

Consequently, research into gearbox fault diagnosis holds significant practical importance [3], prompting numerous authors to concentrate their efforts on this area. Many studies have centered on fault diagnosis of wind turbine gearboxes through vibration monitoring employing machine learning techniques [4] and, more recently, researchers have also explored the application of deep learning techniques in this domain [3, 5]. For processing vibration dada, numerous

✉ Dionísio Henrique Carvalho de Sá Só Martins
dionisiohmartins@gmail.com

[1] Federal University of Rio de Janeiro (UFRJ), Ocean Engineering Program (PENO), Centro de Tecnologia, Bloco I – 108, Cidade Universitária, Ilha Do Fundão, Rio de Janeiro, RJ ZIP 20945-970, Brazil

[2] Federal Center for Technological Education of Rio de Janeiro, Petrópolis, Rio de Janeiro, RJ ZIP-25620-003, Brazil

Published online: 07 October 2024

researchers have employed a range of techniques and feature extraction, such as Wavelet Transform [6] and Empirical Mode Decomposition [7].

In gearbox condition monitoring, acquiring a substantial volume of monitoring data in normal operational states across various working conditions is relatively straightforward. However, the availability of monitoring data in fault states is limited, resulting in a scarcity of fault samples for training fault diagnosis models. This imbalance in the sample set significantly undermines the performance of the classifier, making the resolution of unbalanced sample sets a highly pertinent task [8]. Various solutions have been proposed to address the challenge posed by unbalanced data, aiming to enhance the performance of classification models [9]. Among these solutions, two prominent approaches stand out. One revolves around algorithmic framework. This method offers the advantage of improving diagnostic accuracy for a few samples at the algorithm level, eliminating the need to directly handle unbalanced data. The other approach centers on data frameworks [10]. The objective of this method is to balance the dataset prior to classification by using an oversampling technique [11].

Two commonly employed oversampling techniques include the Synthetic Minority Oversampling Technique (SMOTE), pioneered by Chawla et al. [12]. SMOTE increases the number of minority class by generating virtual samples through random interpolation to achieve a balanced dataset. To enhance the algorithmic performance of SMOTE, Han et al. [13] introduced Borderline-SMOTE, which synthesizes samples along the boundary between the majority and minority classes. Another oversampling method is the adaptive synthetic sample approach, proposed by He et al. [14], known as ADASYN. ADASYN dynamically generates minority data samples based on their distributions, prioritizing the generation of synthetic data for minority class samples that are harder to learn compared to those that are easier to learn. Furthermore, Generative Adversarial Networks (GANs), introduced by Goodfellow et al. [15], represent another oversampling avenue. This concept has since evolved to encompass various GAN variants, including conditional Generative Adversarial Networks (cGAN) [16].

While the aforementioned methods have demonstrated some efficacy, they still exhibit room for improvement. Both SMOTE and ADASYN, in their endeavors to augment the number of minority samples, overlook the distribution area, thereby increasing the risk of classifier overfitting. Conversely, training GAN networks proves to be unstable and requires time-consuming hyperparameter optimization [10]. In contrast, autoencoders, which are unsupervised learning approaches, preserve nonlinear similarities within input data and accurately reproduce them in the generated data [17]. In Ref. [18], the Sparse Autoencoder technique was employed to enhance vibration signal data for diagnosing gear pitting wear. By integrating the sparse autoencoder data augmentation method with a classification algorithm, the study achieved an accuracy of 99%.

In paper [19], a Variational Autoencoder (VAE) was used to generate synthetic instances across nine distinct datasets to enhance the performance of classification models. The study assessed the effectiveness of two classifiers, Support Vector Machine (SVM) and K-Nearest Neighbors (K-NN), by comparing their performance on both the original datasets (without data augmentation) and the augmented datasets containing synthetic instances. The analysis examined three different configurations for the number of neurons per layer in the VAE. The results showed that augmenting the datasets with synthetic instances from the VAE consistently improved the performance of both classifiers.

In work [20], a dataset containing ten distinct types of faults from a wind farm in China was utilized. To address the imbalance in the dataset, a Stacked Autoencoder was employed to generate synthetic samples, thereby augmenting the training set. The feature vector, consisting of 68 dimensions, included data on environmental conditions, operating time, and the operational status of the units, all obtained from a SCADA system. The results of the study show that incorporating synthetic data alongside the real data significantly enhanced the performance of the classification algorithm.

In paper [21], it was demonstrated that synthetic data generated by the Variational Autoencoder (VAE) closely mimic the characteristics of real data in both the time and frequency domains. This approach surpassed other data augmentation methods evaluated in the study. By integrating synthetic data with real data for training a convolutional neural network (CNN), the performance of the classification algorithm was significantly improved.

Feature extraction plays a crucial role in fault identification within condition monitoring. Improper feature extraction inevitably leads to misclassification [22]. While Principal Component Analysis (PCA) is a commonly utilized method for data dimension reduction, its linear space transformation may not effectively address nonlinear problems. In contrast, Kernel Principal Component Analysis (KPCA) offers a solution by employing a kernel function to map the original data to a high-dimensional feature space. Within this space, linear principal components of the mapped data are extracted, facilitating nonlinear fault feature extraction and dimension reduction [23].

This study aims to introduce a novel approach for classifying a wind turbine gearbox dataset containing complex real faults. Our methodology utilizes the deep learning technique Sparse Autoencoder for data augmentation, alongside the KPCA method for dimensionality reduction. Two classification approaches are also analyzed, namely, one using Support Vector Machines (SVM) and another based on Random Forests (RF). A comparative analysis is conducted on

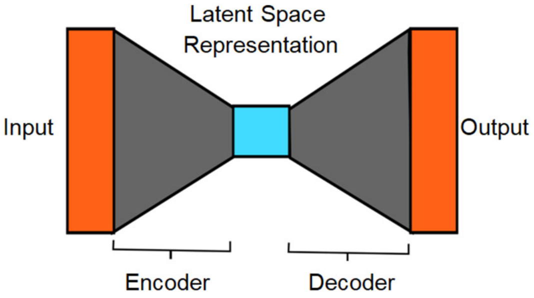

Fig. 1 Architecture of an autoencoder

the performance of models trained on the original dataset, without employing data augmentation or dimension reduction techniques.

The subsequent sections of this paper are organized as follows: Sect. 2 provides an overview of Autoencoder and Sparse Autoencoder, the KPCA method, and the SVM and Random Forest classifiers. Section 3 outlines the methodology employed in this study, encompassing signal acquisition, signal processing, and feature extraction techniques. Section 4 presents the obtained results and, finally, Sect. 5 offers concluding remarks.

2 Theorical Background

2.1 Sparse Autoencoder

An Autoencoder is a type of unsupervised neural network designed to replicate its input as its output. Its conventional architecture comprises two main components: the encoder, which condenses the original (x) into a new representation known as the ‘latent vector’ or ‘compressed representation’, and the decoder, which reconstructs this representation into a new output ($x\prime$) based on the correlations among the input features. Internally, it features a hidden layer that encapsulates the code utilized to represent the input [17, 24, 25]. Figure 1 illustrates the architecture of an autoencoder, with the encoder and decoder functions depicted in Eqs. (1) and (2), respectively.

$$h = f(x) = \sigma(wx + b) \tag{1}$$

$$x\prime = g(h) = \sigma(w\prime x + b\prime) \tag{2}$$

where w and b denote the weights and bias values for the encoder model and $w\prime$ and $b\prime$ denote the weights and bias values for the decoder model and σ is the activation function [25]. The network is trained by minimizing a loss function between the original input and the reconstructed input $L(x, x\prime)$ [17].

Sparse Autoencoder stands out as a state-of-the-art technique that has demonstrated success in enhancing equipment fault diagnosis through data augmentation [17].

When regularization constraints are introduced to the loss function of artificial neural networks, the resulting autoencoder can be termed as a Sparse Autoencoder [18]. To impose a bottleneck in the information flow, the Sparse Autoencoder randomly restricts the number of active nodes during training. This enforced sparsity within the hidden layers prevents the autoencoder from capturing noise in the data and reproducing the input verbatim, enabling it to discern intrinsic features from the input data [17, 24]. In a Sparse Autoencoder, the aim is to minimize the average output of hidden layer, ensuring that most neurons remain inactive [18]. This technique can be classified as a deep learning algorithm because both the encoder and decoder components may consist of multiple hidden layers of neurons, as discussed in references [17, 18]. Equation (3) represents the average activation $\widehat{\rho}$ of hidden layer neurons j:

$$\widehat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}[a_j^{(l)}(x_i)] \tag{3}$$

where $\widehat{\rho}_j$ represents the average activation of the j-th neuron, and m represents the total number of samples in the dataset. $a_j^{(l)}(x_i)$ indicates the degree of activation of the hidden layer neuron j given the input data is x_i, and l indicates the l-th layer of the neural network [18]. The expression of the Sparse Autoencoder loss function is:

$$\begin{aligned} J_{sparse}(W, b) &= J(W, b) \\ &\quad + \beta \sum_{j=1}^{s_l} KL(\rho||\hat{\rho}_j)\hat{\rho}_j \\ &= \frac{1}{m}\sum_{i=1}^{m}\left[a_j^{(l)}(x_i)\right] \end{aligned} \tag{4}$$

where J_{sparse} represents the loss function of the Sparse Autoencoder, W represents the weight matrix, b represents the bias term, KL represents the relative entropy (KL divergence), β is the weight of the sparsity penalty, ρ is the sparsity parameter and s_l is number of cells in the l-th layer. The first term in Eq. (4) is the loss function of the Autoencoder without sparse constraints, including the mean squared term and the weighted decay term [18]. The concrete expression of $J(W, b)$ is:

$$\begin{aligned} J(W, b) = &\left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\|y_i\prime - y_i\|^2\right)\right] \\ &+ \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(W_{ji}^{(l)}\right)^2 \end{aligned} \tag{5}$$

where λ is the weight attenuation coefficient, n_{l-1} is the network layer number and s_l is the number of units in the corresponding layer. The second term in Eq. (4) is the sparse penalty term, and the KL distance expression is [18]:

$$KL\left(\rho||\hat{\rho}_j\right) = \rho\,log\frac{\rho}{\hat{\rho}_j} + (1-\rho)log\frac{1-\rho}{1-\hat{\rho}_j} \qquad (6)$$

where the weighting coefficient for controlling the sparsity penalty is β and the sparsity parameter ρ is usually a smaller value close to zero [18].

2.2 Kernel Principal Component Analysis (KPCA)

Principal Component Analysis (PCA) stands as a renowned unsupervised linear technique for dimensionality reduction and feature extraction. It operates by linearly transforming a high-dimensional input vector into a lower-dimensional one, yielding uncorrelated components known as Principal Components (PCs). These PCs are derived by computing the eigenvectors of the covariance matrix of the original inputs [26, 27].

PCA may not produce optimal results for nonlinear data. Specifically, it is most effective when applied to observations generated from a Gaussian distribution or when the data can be adequately described by linear second-order correlations [28, 29]. Hence, the Kernel Principal Component Analysis (KPCA), which uses the kernel trick K to project data to a higher feature space so that the data can be linearly separated, becomes useful [30], as shown in Eq. (7).

$$K(x_a, x_b) = \phi(x_a)^T\phi(x_b) \qquad (7)$$

where x_a and x_b are two arbitrary data points [27]. Only non-stationary kernel functions were utilized, as they represent the most general class of such functions, as documented in [31].

KPCA operates by initially mapping the original input vectors x_i to a higher-dimensional feature space $\phi(x_i)$ using the kernel method [26]. Commonly used kernels in KPCA include linear, polynomial, Gaussian, and sigmoid. Following the application of the kernel method, PCA is employed on the transformed data, which now exhibits linear separability, to reduce dimensionality [27]. Among the kernel functions, the Gaussian kernel is known for its superior performance, and is often the preferred choice in practical applications of KPCA.

To streamline the testing of various models for Kernel Principal Component Analysis (KPCA), a literature review was conducted to identify kernel functions commonly associated with strong performance. Based on this review, the Gaussian kernel was selected as the KPCA kernel function [30, 32–34].

Accordingly, in this study, the Gaussian kernel was selected as the KPCA kernel function [30].

$$K(x_a, x_b) = exp\left(-\gamma\|x_a - x_b\|^2\right) \qquad (8)$$

where γ, is a parameter [35, 36]. In this work, the value of γ was found by testing a range of different values among the value set [0.01; 0.1; 1; 10; 100].

2.3 Support Vector Machine (SVM)

The Support Vector Machine (SVM) stands as a prominent supervised machine learning algorithm employed extensively in classification tasks. Its core principle involves seeking an optimal hyperplane that maximally separates two classes in the input space. In scenarios where the classes are not linearly separable, SVM employs nonlinear kernel functions to map the input space into a higher-dimensional feature space, facilitating effective classification. The input feature vectors that lie on the boundary plane are called support vectors, and they help to decide the hyperplane orientation and position [37]. As they contain the needed information, the support vectors are sufficient to model the classifier [37].

To optimize the SVM classifier, a grid search technique was employed to select the optimal value for the regularization constant C. The search space for C was defined as {2^{-5}, 2^{-3}, 2^{-1}, ..., 2^{13}, 2^{15}} and a Gaussian kernel function was used.

2.4 Random Forest (RF)

The Random Forest algorithm is a powerful machine learning ensemble method composed of multiple decision trees (DT). Unlike a single decision tree, which is constructed greedily by selecting the best feature splits at each node, Random Forest employs bagging to build its decision trees. Bagging involves creating T bootstrap sets, which are subsets of the N training data formed by random sampling with replacement of training instances [38]. About 2/3 of each bootstrap dataset is used to build each tree, while the remaining 1/3, referred to as out-of-bag data, is used to estimate the classification error of each tree [38]. After constructing the random forest, the classification of an instance involves passing it through each decision tree within the forest. The output of each decision tree serves as a vote, collectively contributing to the final classification. The overall output of the random forest is determined by aggregating these individual decisions, typically by selecting the class with the majority of votes [38].

To optimize the Random Forest (RF) classifier, a grid search technique was used to determine the optimal number of trees. The search space for the number of trees was

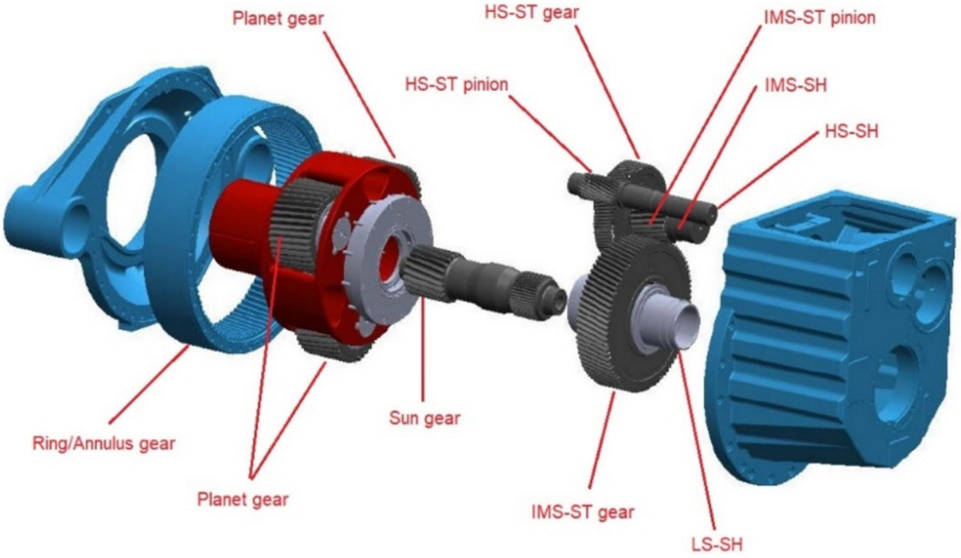

Fig. 2 Expanded view of the gearbox [39]

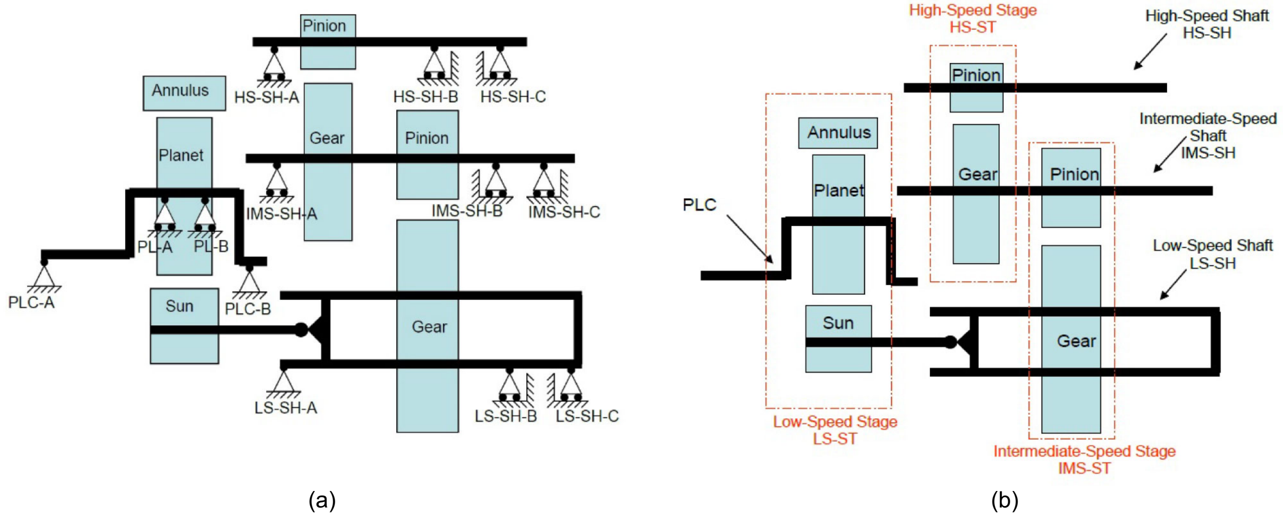

Fig. 3 Wind Turbine Gearbox; **a** bearings location; **b** gearbox arrangement [39]

defined as {5, 10, 15, ..., 40, 50} with 5 observations per leaf.

3 Methodology

3.1 Data Acquisition

The dataset was collected using a three-blade test turbine rated at 750 kW, featuring upwind and stall control. Data acquisition was performed with the National Instruments PXI-4472B high-speed data acquisition (DAQ) system. Measurements were taken with the main shaft operating at 22.09 rpm and the high-speed shaft running at its nominal speed of 1800 rpm. To minimize the risk of critical gearbox failure, a nominal power setting of 50% was used during these measurements.

The dataset was obtained from two gearboxes of identical design—one in a ‘healthy’ state and the other ‘damaged’—through GRC dynamometer tests. Vibration data were sampled at 40 kHz over a ten-minute period and subsequently divided into ten separate files for each test condition [39].

The complete drivetrain and nacelle assembly were installed in the National Renewable Energy Laboratory (NREL) Drive Train Facility (DTF) and securely anchored to the floor, excluding the hub, rotor, yaw bearing, and yaw drives. The system was operated using the actual field controller to manage start-up procedures and ensure safety

Table 1 Gears and bearings basic information [39]

Gears				Bearings			
	Teeth	Speed (Hz)	GMF (Hz)		Type	Local	Quantity
HS-ST				*HS-SH*			
Pinion	22	30	660		CRB	Upwind	1
Gear	88	7.5			TRB	Downwind	2
IMS-ST				*IMS-SH*			
Pinion	23	7.5	172.5		CRB	Upwind	1
Gear	82	2.1			TRB	Downwind	2
Planetary				*Planetary*			
Sun Gear	21	2.1	36.45		CRB	Upwind	1
Planet Gear	39	1.3			CRB	Downwind	1
Ring Gear	99	×		*Planet Carrier*			
					fcCRB	Upwind	1
					fcCRB	Downwind	1

Table 2 Faults and their locations [39]

Component	Fault
HS-ST gear set	Scuffing
HS-SH downwind bearings	Overheating
IMS-ST gear set	Fretting Corrosion, scuffing, polishing wear
IMS-SH upwind bearings	Assembly damage, scuffing, dents
IMS-SH downwind bearings	Assemby damage, dents
Annulus/ring gear, or sun pinion	Scuffing and polishing, fretting corrosion
Planet carrier upwind bearing	Fretting corrosion

responses. To measure turbine vibration levels, accelerometers were installed on various components: the lower and upper radial ring gears, LS-SH radial, IMS-SH radial, HS-SH radial, HS-SH radial upwind bearing, HS-SH upwind bearing, radial wind support downwind, and the radial support. Additionally, a tachometer was installed on the HS-SH to measure rotational speed [39].

After completing its run-in period on the dynamometer, one of the gearboxes underwent field testing at a wind farm, where it encountered two instances of oil loss leading to damage in internal components like bearings and gear elements. Following these incidents, the gearbox was returned to the National Renewable Energy Laboratory (NREL) for further assessment. Here, condition monitoring (CM) equipment was installed, and the gearbox underwent controlled load testing to prevent catastrophic damage [39]. The dataset provided by NREL thus encompasses real-world faults, reflecting compound issues affecting the gearbox. The two gearboxes were originally taken from the field and they are composed of one low speed planetary stage and two parallel stages, with an overall ratio of 1:81.49 [39].

As an innovative extension of prior research [40], this study broadened the scope by introducing six distinct conditions, encompassing both healthy and damaged states of: (I) the High-Speed Shaft; (II) Intermediate Shaft Speed; and (III) Planetary components.

Figure 2 shows an overview of the expanded gearbox conditions, while Fig. 3a, b illustrate the locations of the bearings and the arrangement of the gearbox, respectively.

As can be seen in Fig. 3a, each planet gear is supported by two identical cylindrical roller bearings (CRB) and the planet carrier is supported by two full-component cylindrical roller bearings (fcCRB). Each parallel shaft is supported by a CRB on the upwind side and by two back-to-back mounted tapered roller bearings (TRB) on the downwind side of the assembly [39].Table 1 provides the location of those bearings and information about all the gears of the gearbox and Table 2 shows the faults and their locations.

3.2 Signal Processing

The signals collected for the parallel axes (High Speed and Intermediate Speed shafts) and for the planetary in the conditions described above, were processed using MATLAB software.

Each condition of the parallel axes comprises 10 signals, while each condition of the planetary has 20 signals, resulting in an unbalanced dataset. To assess classifier performance, four experiments were conducted to compare the original unbalanced dataset with the same dataset subjected

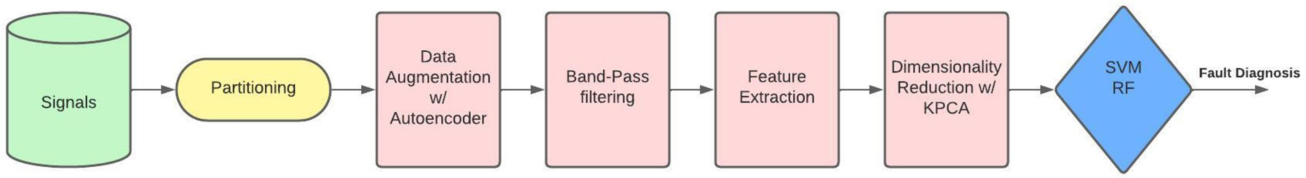

Fig. 4 Flowchart of the proposed method

Table 3 Time domain features extracted

No	Feature Name	Definition	No	Feature Name	Definition
1	Mean	$\frac{1}{N}\sum_{i=1}^{N} x_i$	10	Impulse Factor	$\frac{max}{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert}$
2	RMS	$\sqrt{\frac{1}{N}\left[\sum_{i=1}^{N}(x_i)^2\right]}$	11	Peak to Peak	$\max(x_i) - \min(x_i)$
3	Standard Deviation	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$	12	Minimum value	$\min(x_i)$
4	Peak	$\max(abs(x_i))$	13	Sum	$\sum_{i=1}^{N} x_i$
5	Skewness	$\frac{N\sum_{i=1}^{N}(x_i-\mu)^3}{\sigma^3}$	14	Variance	$\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$
6	Kurtosis	$\frac{N\sum_{i=1}^{N}(x_i-\mu)^4}{\left[\sum_{i=1}^{N}(x_i-\mu)^2\right]^2}$	15	Standard Error	$\frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2}}{\sqrt{N}}$
7	Crest Factor	$\frac{max}{RMS}$	16	Median	$median(x_i)$
8	Clearence Factor	$\frac{max}{\frac{1}{N}\sum_{i=1}^{N}(x_i)^2}$	17	Energy	$\sum_{i=1}^{N}\lvert x_i\rvert^2$
9	Shape Factor	$\frac{RMS}{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert}$	18	Entropy	$-\sum_{i=1}^{N} p_i log_2(p_i)$ $where p_i = \frac{E_i}{E}$

to data augmentation and dimensionality reduction techniques. Figure 4 depicts the flowchart outlining the proposed methodology, as employed in the fourth experiment.

In the four experiments, a bandpass filter between 10 and 2000 Hz was used in all signals according to [41], and after that, 18 time domain features, which are shown in Table 3, were extracted from the filtered signals [40].

3.3 Data Augmentation

Data augmentation is employed to enhance predictive model performance by expanding the original dataset through generated data. We conducted four experiments to evaluate the impact of data imbalance on classifier performance: (1) using the original dataset without data augmentation, (2) employing partial data augmentation, (3) applying full data augmentation, and (4) balancing the dataset while using Kernel Principal Component Analysis (KPCA) for dimensionality reduction. In this work, the Sparse Autoencoder was utilized for data augmentation. Given the substantial sample count within the time series, each signal in the original dataset was partitioned into six equal segments, as showed in in Table 4. Subsequently, we used the partitioned dataset as input for the autoencoder to produce new signals.

Table 4 Initial dataset and dataset partitioned in 6 equal parts

Original dataset	Original dataset partitioned in 6 parts
10 HSS healthy signals	60 HSS healthy signals
10 HSS damaged signals	60 HSS damaged signals
10 IMSS healthy signals	60 IMSS healthy signals
10 IMSS damaged signals	60 IMSS damaged signals
20 Planetary healthy signals	120 Planetary healthy signals
20 Planetary damaged signals	120 Planetary damaged signals

The number of planetary signals (healthy or damaged) served as the reference for determining the necessary signal augmentation to ensure that each condition comprised 20% of the dataset, thus balancing it.

Initially, data augmentation was applied only partially, resulting in an improved but still unbalanced dataset, as depicted in the first column of Table 5. Then, augmentation was extended, achieving dataset balance, as shown in the second column of Table 5.

Regarding the autoencoder structure, we used a single hidden layer with an L2 weight regularizer set to 0.01. The

Table 5 Dataset with partial data augmentation and full data augmented and balanced

Partial data augmentation	Full data augmented and balanced
270 HSS healthy signals	300 HSS healthy signals
270 HSS damaged signals	300 HSS damaged signals
270 IMSS healthy signals	300 IMSS healthy signals
270 IMSS damaged signals	300 IMSS damaged signals
300 Planetary healthy signals	300 Planetary healthy signals
300 Planetary damaged signals	300 Planetary damaged signals

Table 6 MSE and PCC values for the generated data

Signals	MSE value	PCC value
HSS healthy A	0.7306	0.9462
HSS healthy B	0.5278	0.9629
HSS damaged A	0.2501	0.9927
HSS damaged B	0.7667	0.9771
IMSS healthy A	0.0939	0.9942
IMSS healthy B	0.7363	0.9553
IMSS damaged A	0.2448	0.9939
IMSS damaged B	0.7252	0.9825
Planetary healthy B	0.5058	0.9284
Planetary damaged B	0.5503	0.9662

loss function employed was msesparse, with Logsig as the encoder transfer function and Purelin as the decoder transfer function. The training was conducted with a maximum of 400 epochs. For the number of neurons, we configured 12 neurons for HSS Healthy, 22 for HSS Damaged, 18 for IMSS Healthy, 7 for Planetary Healthy, and 20 for Planetary Damaged.

3.4 Data Validation

To assess the fidelity of the generated data, two metrics were employed: Mean Square Error (MSE) and the Pearson Correlation Coefficient (PCC). This approach ensures that any significant deviation between the generated and real data is detected, as excessive disparities could compromise classification accuracy.

MSE measures the amount of error in statistical models assessing the average squared difference between the observed and the predicted values. When a model has no error, the MSE value is equal to 0, so the closer to 0 the value of the error is, better is the model. PCC computes the linear relationship between the two signals and its magnitude ranges from − 1 to + 1. The closer to the extremes greater is the correlation between the signals. A value of 0 or close to 0, indicates a weak or no correlation between signals [42]. Table 6 shows the average of the MSE and PCC values for the generated data.

As indicated in Table 6, the Mean Square Error (MSE) values approached zero, and the Pearson Correlation Coefficient (PCC) values neared unity, suggesting an acceptable correspondence between the generated and real data. Figure 5 visually illustrates this relationship between the real and generated data.

Figures 5, 6, and 7 display the training curves of the autoencoder used to generate synthetic data for HSS Healthy, IMSS Healthy, and Planetary Healthy signals. Figures 8, 9, and 10 compare the measured and synthetic vibration signals in both the time and frequency domains for HSS, IMSS, and Planetary.

4 Results and Discussion

4.1 Results

As previously mentioned, four experiments were conducted. After filtering the vibration signals, 18 time-domain features (refer to Table 3) were extracted. These experiments were as follows: the first utilized the data from Table 4 without any partitioning, data augmentation, or balancing; the second employed the dataset from Table 5, incorporating partial data augmentation; the third utilized the balanced and augmented dataset from Table 5; and the fourth, utilizing the same dataset as the third experiment, alongside dimensionality reduction via KPCA.

In each experiment, data from both sides of the tables were amalgamated, followed by a stratified fivefold cross-validation to split the dataset into training and testing subsets. Subsequently, SVM and RF classifiers were trained and evaluated on the respective datasets. The results of the first, second, and third experiments, along with the best-performing results of SVM and RF models from the fourth experiment, are summarized in Table 7.

Figures 8, 9 and 10 show the comparison of the measured and synthetic vibration signals, in time and frequency domains, for HSS, IMSS and Planetary.

The confusion matrices of each experiment are presented in Figs. 11, 12, 13 and 14, where the columns represent the predicted values.

Upon analyzing the results presented in Table 7, it becomes evident that even with a small and unbalanced dataset in the first experiment, both SVM and RF achieved accuracies exceeding 90%. This performance can likely be attributed to the effectiveness of the extracted features in

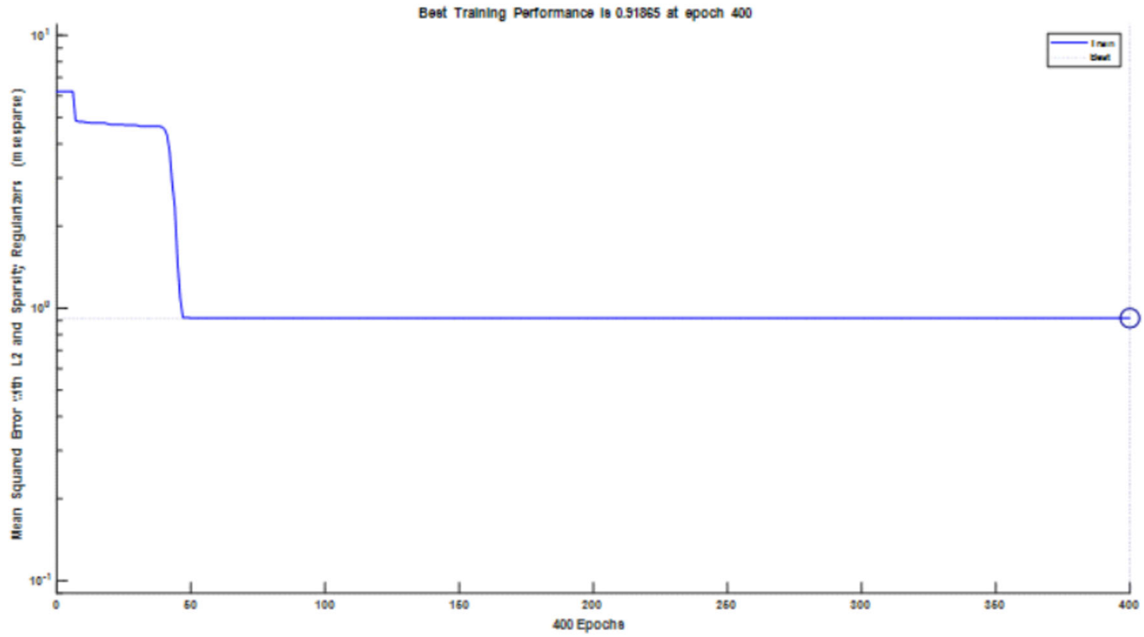

Fig. 5 Autoencoder training curve for generating synthetic data for HSS healthy

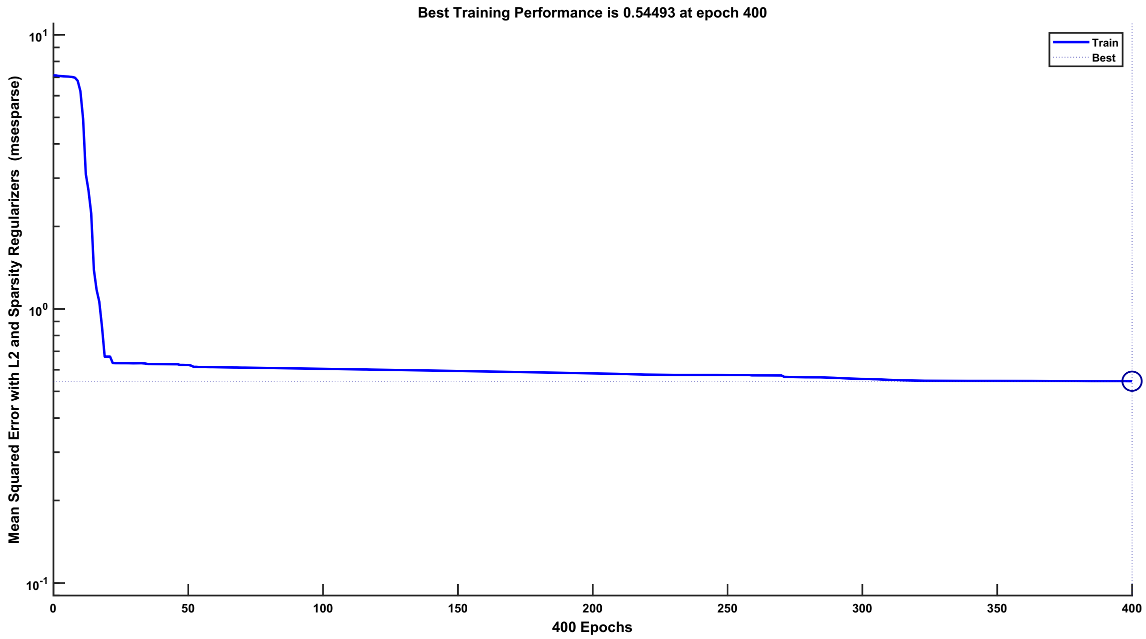

Fig. 6 Autoencoder training curve for generating synthetic data for IMSS healthy

accurately representing the vibration patterns associated with different conditions.

In the second and third experiments, as illustrated in Figs. 11 and 12 respectively, the introduction of dataset augmentation enhanced classifier performance. Moreover, dataset balancing further improved performance, affirming the quality of the generated data.

Furthermore, upon comparison of the confusion matrices, it is apparent that the most significant classification errors occurred within the parallel shaft conditions, likely due to consistent damage patterns in the gears. Conversely, minimal classification errors were observed between the planetary conditions or between planetary conditions and any one of the parallel shaft conditions, owing to slight variations in vibration patterns between the planetary and parallel shafts.

The KPCA experiment achieved good results. Both SVM and Random Forest models reduced their dimensionality to 11. For the SVM model, the KPCA parameter γ was set to

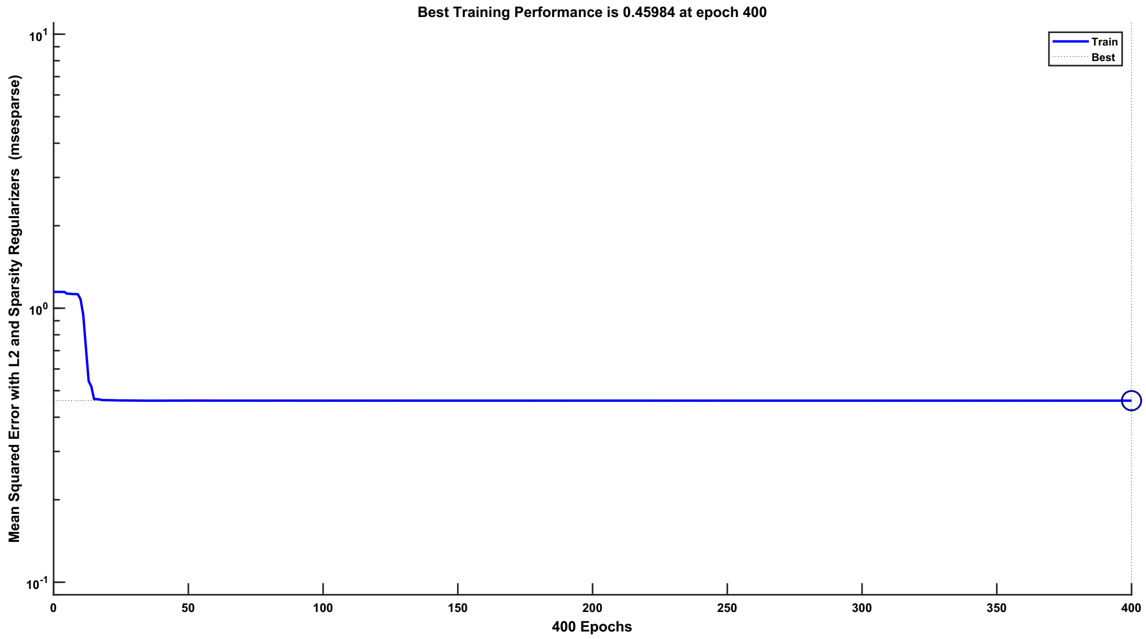

Fig. 7 Autoencoder training curve for generating synthetic data for Planetary healthy

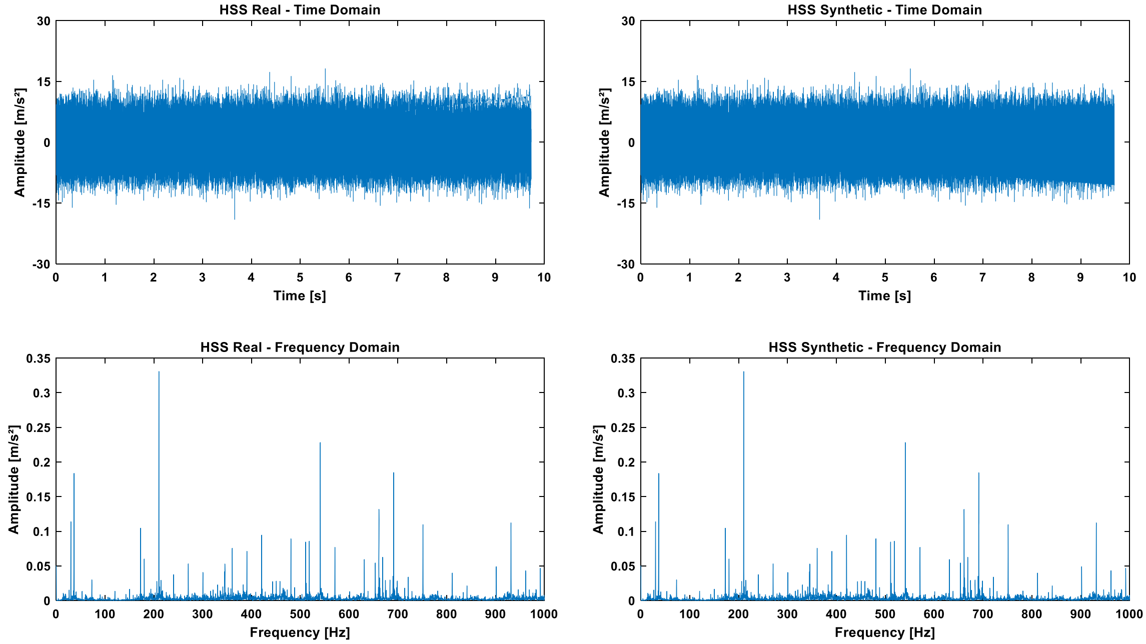

Fig. 8 Comparison of measured and synthetic vibration signals in time and frequency domains

1, and it achieved the same overall accuracy value (98.72%) as that achieved in the third experiment, which used all the 18 time domain features, using the Gaussian kernel and the value of the regularization index $C = 32$ as the SVM hyperparameters.

For the RF model, the KPCA parameter γ was set to 0.1, and it achieved an overall accuracy value of 96.06%, which is very close to those achieved in the second and third experiments, using the Random Search as the search method, comprising of 146 trees with 5 observations per leaf.

While accuracy and the confusion matrix serve as primary metrics for comparing model performance, providing an overview of classification correctness, additional metrics were calculated to compare the best models from the fourth experiment, as depicted in Fig. 15.

It can be seen in Fig. 15 that both models yielded similar results in all calculated metrics, except for the computational

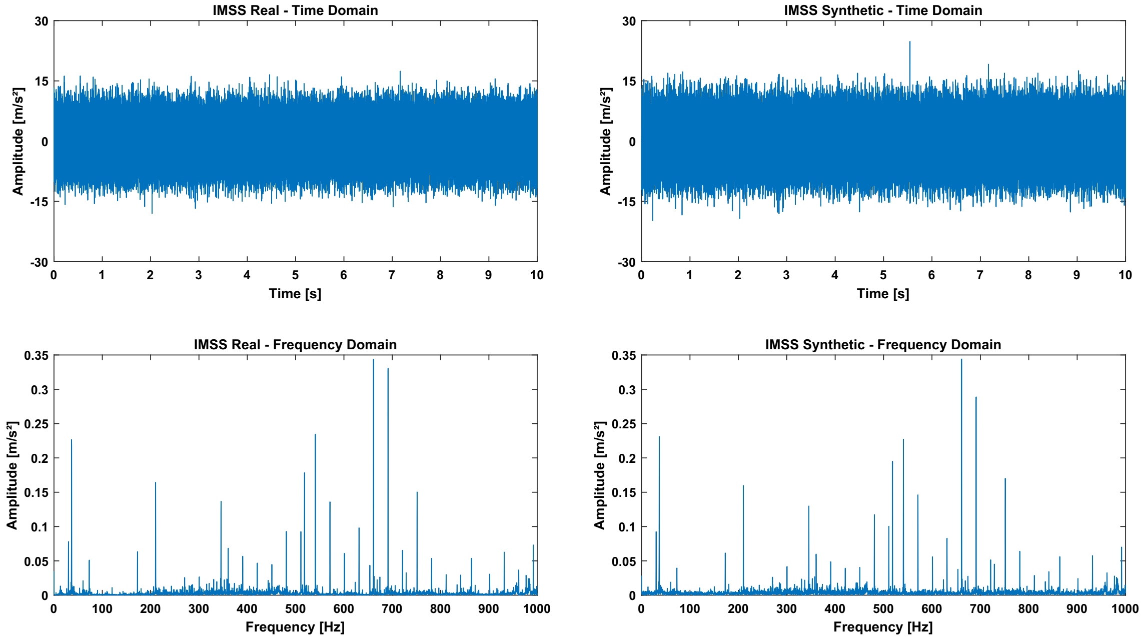

Fig. 9 Comparison of measured and synthetic vibration signals in time and frequency domains

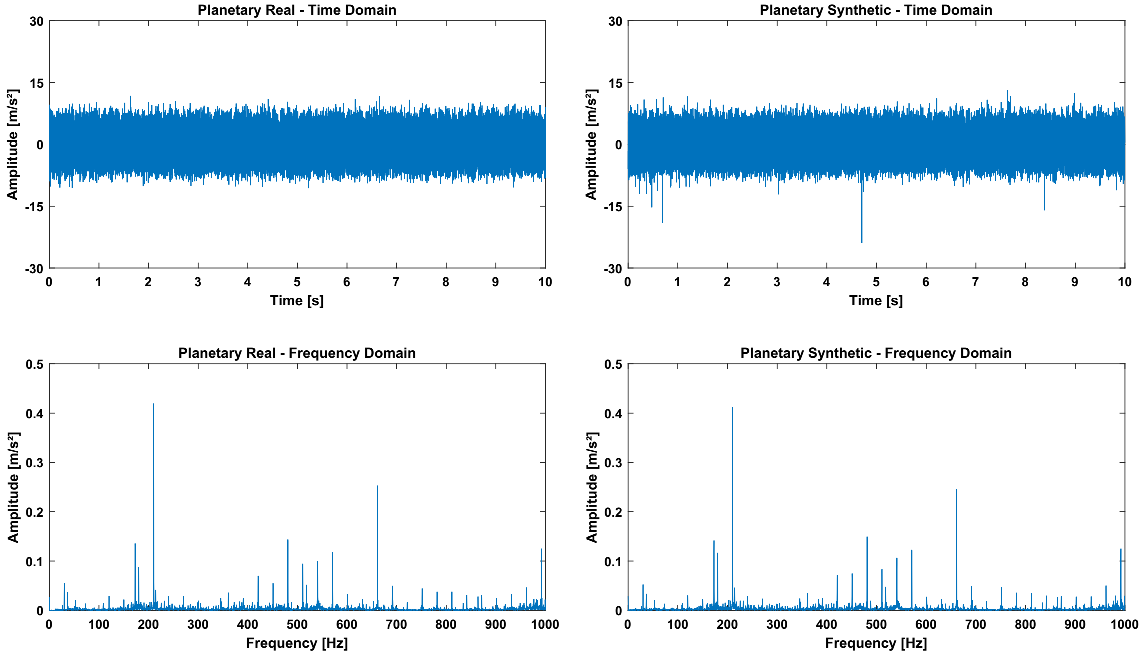

Fig. 10 Comparison of measured and synthetic vibration signals in time and frequency domains

Table 7 Results of the experiments

		SVM	RF
Experiment 1	Accuracy (%)	90.83	95.00
Experiment 2	Accuracy (%)	98.03	99.83
Experiment 3	Accuracy (%)	98.72	99.94
Experiment 4	Accuracy (%)	98.72	96.06

	IMSS Heathy	IMSS Damaged	HSS Healthy	HSS Damaged	Planetary Healthy	Planetary Damaged
IMSS Heathy	**90**	0	0	0	5	0
IMSS Damaged	0	**80**	0	10	0	0
HSS Healthy	0	0	**100**	0	0	0
HSS Damaged	0	10	0	**80**	0	0
Planetary Healthy	10	0	0	0	**95**	0
Planetary Damaged	0	10	0	10	0	**100**

(a)

	IMSS Heathy	IMSS Damaged	HSS Healthy	HSS Damaged	Planetary Healthy	Planetary Damaged
IMSS Heathy	**100**	0	0	0	0	0
IMSS Damaged	0	**70**	0	0	0	0
HSS Healthy	0	0	**100**	0	0	0
HSS Damaged	0	0	0	**100**	0	0
Planetary Healthy	0	0	0	0	**100**	0
Planetary Damaged	0	30	0	0	0	**100**

(b)

Fig. 11 Confusion matrices of the first experiment (%); **a** SVM; **b** Random Forest

	IMSS Heathy	IMSS Damaged	HSS Healthy	HSS Damaged	Planetary Healthy	Planetary Damaged
IMSS Heathy	**96.67**	0	3.33	0	0	0
IMSS Damaged	0	**97.41**	0	1.85	0	0
HSS Healthy	3.33	0	**96.67**	0.37	0	0
HSS Damaged	0	2.59	0	**97.78**	0	0
Planetary Healthy	0	0	0	0	**99.67**	0
Planetary Damaged	0	0	0	0	0.33	**100**

(a)

	IMSS Heathy	IMSS Damaged	HSS Healthy	HSS Damaged	Planetary Healthy	Planetary Damaged
IMSS Heathy	**99.63**	0	0	0	0.67	0
IMSS Damaged	0	**100**	0	0	0	0
HSS Healthy	0.37	0	**100**	0	0	0
HSS Damaged	0	0	0	**100**	0	0
Planetary Healthy	0	0	0	0	**99.33**	0
Planetary Damaged	0	0	0	0	0	**100**

(b)

Fig. 12 Confusion matrices of the second experiment (%); **a** SVM; **b** Random Forest

	IMSS Heathy	IMSS Damaged	HSS Healthy	HSS Damaged	Planetary Healthy	Planetary Damaged
IMSS Heathy	**98.33**	0	1.67	0	0	0
IMSS Damaged	0	**99.67**	0	2.67	0	0
HSS Healthy	1.67	0	**98.33**	0.33	0	0
HSS Damaged	0	0.33	0	**97**	0	0
Planetary Healthy	0	0	0	0	**99.33**	0.33
Planetary Damaged	0	0	0	0	0.67	**99.67**

(a)

	IMSS Heathy	IMSS Damaged	HSS Healthy	HSS Damaged	Planetary Healthy	Planetary Damaged
IMSS Heathy	**100**	0	0.33	0	0	0
IMSS Damaged	0	**100**	0	0	0	0
HSS Healthy	0	0	**99.67**	0	0	0
HSS Damaged	0	0	0	**100**	0	0
Planetary Healthy	0	0	0	0	**100**	0
Planetary Damaged	0	0	0	0	0	**100**

(b)

Fig. 13 Confusion matrices of the third experiment (%); **a** SVM; **b** Random Forest

	IMSS Heathy	IMSS Damaged	HSS Healthy	HSS Damaged	Planetary Healthy	Planetary Damaged
IMSS Heathy	**97.67**	0	1.33	0.33	0	0
IMSS Damaged	0	**98**	0	2.33	0	0
HSS Healthy	2.33	0.67	**98.67**	0	0	0
HSS Damaged	0	1.33	0	**97.33**	0	0
Planetary Healthy	0	0	0	0	**99.33**	0.67
Planetary Damaged	0	0	0	0	0.67	**99.33**

(a)

	IMSS Heathy	IMSS Damaged	HSS Healthy	HSS Damaged	Planetary Healthy	Planetary Damaged
IMSS Heathy	**95.33**	0	4.67	0.67	0	0.33
IMSS Damaged	0	**97.33**	0	4.33	0	0
HSS Healthy	4	0	**94**	0.67	0	0.33
HSS Damaged	0.67	2.67	0.33	**94.33**	0	0
Planetary Healthy	0	0	1	0	**98.67**	2.67
Planetary Damaged	0	0	0	0	1.33	**96.67**

(b)

Fig. 14 Confusion matrices of the fourth experiment (%); **a** SVM; **b** Random Forest

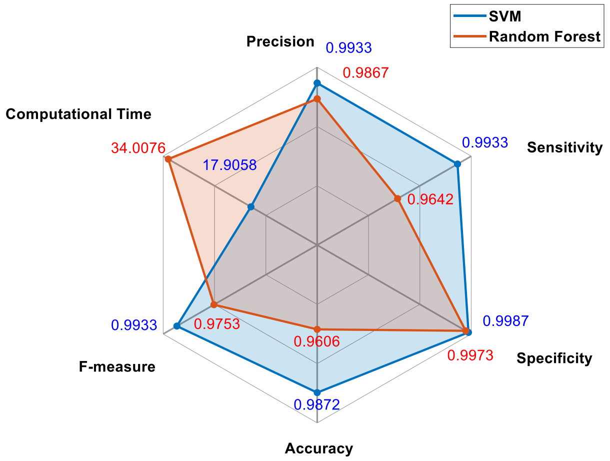

Fig. 15 Radar plot of the comparison metrics between the best models of the fourth experiment

time. The SVM model achieved shorter computation time, 17.9058 s compared with 34.0076 s of the Random Forest.

4.2 Comparison with Previous Works

Numerous studies in the literature have addressed the classification of mechanical faults in wind turbine gearboxes. In [43], the authors proposed a Convolutional Neural Network (CNN) for fault classification, employing cyclostationary and kurtogram analyses on acquired data. Despite not employing data augmentation or dimensionality.

Similarly [42], proposed SVM with RBF kernel function, utilizing PCA for dimensionality reduction, which achieved 92% accuracy. However, their focus was solely on the planetary stage of a wind turbine gearbox, unlike our approach, which evaluated the entire gearbox. By employing data augmentation and dimensionality reduction, our method achieved superior results, in identifying and classifying faults in both parallel shafts and planetary stages. In [44], the authors proposed a CNN for fault classification, incorporating a variant of GAN, termed improved spectrally normalized-GAN, for data augmentation, achieving an impressive 98.5% accuracy. However, their study concentrated on three distinct types of gear faults. In contrast, our proposed method focuses on compound faults, mirroring real-world scenarios with simultaneous faults in gearbox internal components such as bearings and gears. Table 8 provides a summary of these comparative results.

In this paper, statistical features were extracted directly from the vibration signal without using the empirical mode decomposition technique used in work [40], thus allowing an increase in speed in the training stage of the machine learning models used in this paper.

Dealing with unbalanced datasets is a common issue in equipment failure diagnosis, as there are often very few instances related to failure classes. This study examines the impact of using synthetic data generated by the variational autoencoder technique on the training of machine learning algorithms and the effectiveness of equipment diagnosis. This approach was not applied in paper [40].

In this paper, an additional contribution to the previous work [40] is the identification of two more scenarios: planetary healthy signals and planetary damaged signals. Consequently, the present paper deals with the identification of six different scenarios, while the work referenced in [40] aimed to identify only four scenarios.

In this study, we evaluated the effectiveness of using the KPCA technique to reduce the dimensionality of the feature vector. This process helped in reducing model training time

Table 8 Summary results of previous works

Work	Classifier	Data Augmentation	Dimensionality reduction	Accuracy (%)
[42]	SVM	–	Dynamic PCA	92.0
[43]	Cyclostationary-based CNN	–	–	88.9
[44]	CNN	SN-GANs	–	98.5

and improving diagnostic accuracy. On the other hand, the outhors of [40] used the mRMR, ReliefF, Decision Tree, and Chi-square techniques to create a feature selection committee for choosing the most relevant features for the machine learning models.

5 Conclusion

This study utilized a gearbox dataset provided by NREL, featuring compound real faults, to classify six state conditions: Healthy High-Speed Shaft (HSS), Damaged HSS, Healthy Intermediate Shaft Speed (IMSS), Damaged IMSS, Healthy Planetary, and Damaged Planetary stages. Due to the small size and imbalance nature of the dataset, data augmentation was performed using the Sparse Autoencoder.

The augmentation process, validated through MSE and PCC values, involved splitting the original dataset in half and generating additional signals. Following augmentation, the signals underwent bandpass filtering, from which 18 time-domain features were extracted.

Four experiments were conducted: (1) utilizing the original unbalanced dataset, (2) augmenting the dataset without balancing, (3) augmenting and balancing the dataset, and (4) augmenting, balancing, and employing KPCA for dimensionality reduction. SVMs and RFs algorithms were employed for classification in all experiments.

Results demonstrated that despite the small and unbalanced dataset, both SVM and Random Forest models achieved over 90% overall accuracy. Dataset augmentation and subsequent balancing notably improved classifier performance, with experiments 2 and 3 yielding comparable results. Furthermore, dimensionality reduction via KPCA enhanced the performance of both SVM and RF, achieving an overall accuracy of 98.72 and 96.06%, respectively, with the dimensions of each model being reduced from 18 to 11. Notably, SVM exhibited faster computational times compared to Random Forest, completing the analysis in 17.9058 s versus 34.0076 s.

The method proposed in this work evaluated the whole gearbox, identifying and classifying faults in the parallel shafts and planetary stage using data augmentation and dimensionality reduction outperforming previous works.

For future research, the proposed method could be tested under variable speed conditions and extended to differentiate between gear and bearing faults.

Author Contribution Each author contributed to the research presented in this manuscript, approved the contents now presented, and agreed to the compliance with ethical standards. Experimentation: Leonardo Félix; Original draft writing: Leonardo Félix and Dionísio Martins. Review and editing: Ulisses Monteiro, Luiz Vaz, Luís Tarrataca and Carlos Martins.

Funding The authors declare that no funds, grants, or other support was received during the preparation of this manuscript.

Data availability The datasets analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

References

1. Shen, G., Xiang, D., Zhu, K., Jiang, L., Shen, Y., Li, Y.: Fatigue failure mechanism of planetary gear train of wind turbine gearbox. Eng. Fail. Anal. **87**, 96–110 (2018). https://doi.org/10.1016/j.engfailanal.2018.01.007
2. Gong, Y., Fei, J.L., Tang, J., Yang, Z.G., Han, Y.M., Li, X.: Failure analysis on abnormal wear of roller bearings in gearbox for wind turbine. Eng. Fail. Anal. **82**, 26–38 (2017). https://doi.org/10.1016/j.engfailanal.2017.08.015
3. Su, Y., Meng, L., Kong, X., Xu, T., Lan, X., Li, Y.: Small sample fault diagnosis method for wind turbine gearbox based on optimized generative adversarial methods. Eng. Fail. Anal. (2022). https://doi.org/10.1016/j.engfailanal.2022.106573
4. Straczkiewicz, M., Barszcz, T.: Application of artificial neural network for damage detection in planetary gearbox of wind turbine. Shock. Vib. (2016). https://doi.org/10.1155/2016/4086324
5. Zhang, X., Han, P., Xu, L., Zhang, F., Wang, Y., Gao, L.: Research on bearing fault diagnosis of wind turbine gearbox based on 1DCNN-PSO-SVM. IEEE Access **8**, 192248–192258 (2020). https://doi.org/10.1109/ACCESS.2020.3032719
6. Saravanan, N., Ramachandran, K.I.: Incipient gearbox fault diagnosis using discrete wavelet transform (DWT) for feature extraction and classification using artificial neural networks (ANN). Expert Syst. Appl. **37**(6), 4168–4418 (2010). https://doi.org/10.1016/j.eswa.2009.11.006

7. Desavale, R.G., Jadhav, P.M., Dharwadkar, N.V.: Dynamic response analysis of gearbox to improve fault detection using empirical mode decomposition and artificial neural network techniques. J. Risk Uncertain. Eng. Syst. (2021). https://doi.org/10.1115/1.4051344
8. Wang, Y., Sun, G., Jin, Q.: Imbalanced sample fault diagnosis of rotating machinery using conditional variational autoencoder generative adversarial network. Appl. Soft Comput. J. (2020). https://doi.org/10.1016/j.asoc.2020.106333
9. Wan, Z., Zhang, Y., He, H.: Variational autoencoder based synthetic data generation for imbalanced learning. IEEE Symp. Ser. Comput. Intell. (2017). https://doi.org/10.1109/SSCI.2017.8285168
10. Xing, Z., Liu, Y., Wang, Q., Li, J.: Intelligent diagnosis of rolling bearings with imbalanced data based on the ConVAE-CNN model. J. Phys. Conf. Ser. (2022). https://doi.org/10.1088/1742-6596/2369/1/012001
11. Babaei, K., Chen, Z.Y., Maul, T.: Data augmentation by autoencoders for unsupervised anomaly detection. Cornell University (2019). https://doi.org/10.48550/arXiv.1912.13384
12. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority oversampling technique. J. Artif. Intell. Res. **16**(1), 321–357 (2002). https://doi.org/10.1613/jair.953
13. Han, H., Wang, W.Y., Mao, B.H.: Boderline-smote: a new oversampling method in imbalanced datasets learning. Adv. Intell. Comput. (2005). https://doi.org/10.1007/11538059_91
14. He, H., Bai, Y., Garcia, E.A., Li, S.: ADASYN: Adaptive synthetic sampling approach for imbalanced learning. IEEE Int. Joint Conf. Neural Netw. **3**, 1322–1328 (2008). https://doi.org/10.1109/IJCNN.2008.4633969
15. Goodfellow, I.J., Abadie, J.P., Mirza, M., Xu, B., Farley, D.W., Ozair, S., Courville, A., Bengio. Y.: Generative Adversarial Networks. https://doi.org/10.48550/arXiv.1406.2661
16. Douzas, G., Bacao, F.: Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Syst. Appl. **91**, 464–471 (2018). https://doi.org/10.1016/j.eswa.2017.09.030
17. Delgado, J.M.D., Oyedele, L.: Deep learning with small datasets: using autoencoder to address limited datasets in construction management. Appl. Soft Comput. (2021). https://doi.org/10.1016/j.asoc.2021.107836
18. Li, X., Li, J., Qu, Y., He, D.: Semi-supervised gear fault diagnosis using raw vibration signal based on deep learning. Chin. J. Aeronaut. **33**(2), 418–428 (2020). https://doi.org/10.1016/j.cja.2019.04.018
19. Goubeaud, M., Joußen, P., Gmyrek, N., Ghorban, F., Schelkes, L., Kummert, A.: Using variational autoencoder to augment sparse time series datasets. In: 7th International Conference on Optimization and Applications (ICOA), Wolfenbüttel, Germany, pp. 1–6 (2021). https://doi.org/10.1109/ICOA51614.2021.9442619
20. Ma, Z., Cao, M., Deng, Yi., Jiang, Y., Tian, Ye., Wang, X.: Typical fault prediction method for wind turbines based on an improved stacked autoencoder network. Energy Harvest. Syst. **11**(1), 1–11 (2024). https://doi.org/10.1515/ehs-2023-0072
21. Zhao, D., Liu, S., Gu, D., Sun, X., Wang, L., Wei, Y., Zhang, H.: Enhanced data-driven fault diagnosis for machines with small and unbalanced data based on variational auto-encoder. Meas. Sci. Technol. (2019). https://doi.org/10.1088/1361-6501/ab55f8
22. Shao, R., Hu, W., Wang, Y., Qi, X.: The fault feature extraction and classification of gear using principal component analysis and kernel principal component analysis based on wavelet packet transform. Measurement **54**, 118–132 (2014). https://doi.org/10.1016/j.measurement.2014.04.016
23. Cheng, G., Chen, X., Li, H., Li, P., Liu, H.: Study on planetary gear fault diagnosis based on entropy feature fusion of ensemble empirical mode decomposition. Measurement **91**, 140–154 (2016). https://doi.org/10.1016/j.measurement.2016.05.059
24. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
25. Bedi, J., Toshniwal, D.: Energy load time-series forecast using decomposition and autoencoder integrated memory network. Appl. Soft Comput. J. (2020). https://doi.org/10.1016/j.asoc.2020.106390
26. Zhang, Y., Zuo, H., Bai, F.: Classification of fault location and performance degradation of a roller bearing. Measurement **46**(3), 1178–1189 (2013). https://doi.org/10.1016/j.measurement.2012.11.025
27. Anowar, F., Sadaoui, S., Selim, B.: Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE). Comput. Sci. Rev. (2021). https://doi.org/10.1016/j.cosrev.2021.100378
28. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. **10**(5), 1299–1319 (1998). https://doi.org/10.1162/089976698300017467
29. Lee, J.-M., et al.: Nonlinear process monitoring using kernel principal component analysis. Chem. Eng. Sci. **59**(1), 223–234 (2004). https://doi.org/10.1016/j.ces.2003.09.012
30. Wang, W., Zhang, M., Wang, D., Jian, Y.: Kernel PCA feature extraction and the SVM classification algorithm for multiple-status, through-wall, human being detection. EURASIP J. Wirel. Commun. Netw. (2017). https://doi.org/10.1186/s13638-017-0931-2
31. Genton, M.G.: Classes of kernels for machine learning: a statistics perspective. J. Mach. Learn. Res. **2**, 299–312 (2001)
32. Harkat, M.-F., et al.: Machine learning-based reduced kernel PCA model for nonlinear chemical process monitoring. J. Control Autom. Electr. Syst. **31**(5), 1196–1209 (2020). https://doi.org/10.1007/s40313-020-00604-w
33. Attouri, K., et al.: Wind power converter fault diagnosis using reduced kernel PCA-based BiLSTM. Sustainability **15**(4), 1–19 (2023). https://doi.org/10.3390/su15043191
34. Gwashavanhu, B.K., Oberholster, A.J., Heyns, S.P.: A comparative study of principal component analysis and kernel principal component analysis for photogrammetric shape-based turbine blade damage analysis. Eng. Struct. **318**, 1–18 (2024). https://doi.org/10.1016/j.engstruct.2024.118712
35. Olsson D (2011) Applications and implementation of kernel principal component analysis to specific data sets. Master's Thesis Report, University of Florida.
36. Chang, Q., Chen, Q., Wang, X.: Scaling Gaussian RBF kernel width to improve SVM classification. In: International Conference on Neural Networks and Brain (2005). https://doi.org/10.1109/ICNNB.2005.1614559
37. Syed, S.H., Muralidharan, V.: Feature extraction using discrete wavelet transform for fault classification of planetary gearbox—a comparative study. Appl. Acoust. (2022). https://doi.org/10.1016/j.apacoust.2021.108572
38. Quiroz, J.C., Mariun, N., Mehrjou, M.R., Izadi, M., Misron, N., Radzi, M.A.M.: Fault detection of broken rotor bar in LS-PMSM using random forests. Measurement **116**, 273–280 (2018). https://doi.org/10.1016/j.measurement.2017.11.004
39. Sheng S (2012) Wind turbine gearbox vibration condition monitoring benchmarking datasets. National Renewable Energy Laboratory NREL/TP-5000–54530.
40. Felix, L.O., de Sá Só Martins, D.H., Monteiro, U.A., et al.: A feature selection committee method using empirical mode decomposition for multiple fault classification in a wind turbine Gearbox. J. Nondestruct. Eval. **42**, 85 (2023). https://doi.org/10.1007/s10921-023-00996-0
41. International Standard, ISO 10816-21:2015: Mechanical vibration—evaluation of machine vibration by measurements on nonrotating parts—Horizontal axis wind turbines with gearbox (2015)
42. Kordestani, M., Rezamand, M., Orchard, M., Carriveau, R., Ting, D.S.K., Saif, M.: Planetary gear faults detection in wind turbine

gearbox based on a ten years historical data from three wind farms. IFAC-PapersOnLine **53**(2), 10318–10323 (2020). https://doi.org/10.1016/j.ifacol.2020.12.2767

43. Amin, A., Bibo, A., Panyam, M., Tallapragada, P.: Vibration based fault diagnostics in a wind turbine planetary gearbox using machine learning. Wind Eng. **47**(1), 175–189 (2023). https://doi.org/10.1177/0309524X221123968
44. Shen, C., Wang, J., Chen, J., Zhang, B.: Gearbox fault diagnosis for wind turbine based on data augmentation using improved generative adversarial networks. In: International Conference on Electrical Materials and Power Equipment (ICEMPE) (2021). https://doi.org/10.1109/ICEMPE51623.2021.9509056

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.