IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 19, 2021, accepted April 21, 2021, date of publication April 29, 2021, date of current version May 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3076571

Global and Local Feature Extraction Using a Convolutional Autoencoder and Neural Networks for Diagnosing Centrifugal Pump Mechanical Faults

ALEXANDER E. PROSVIRIN, ZAHOOR AHMAD, AND JONG-MYON KIM, (Member, IEEE)
School of Electrical, Electronics, and Computer Engineering, University of Ulsan, Ulsan 44610, South Korea

Corresponding author: Jong-Myon Kim (jmkim07@ulsan.ac.kr)

This work was supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry, and Energy (MOTIE) of the Republic of Korea under Grant 20181510102160.

ABSTRACT Centrifugal pumps are important types of electro-mechanical machines used for fluid and energy conveyance. Mechanical faults in centrifugal pumps lead to abnormal impacts in the vibration signal of the system. Those impacts induce nonstationarity in vibration signals and hence complex time-frequency domain signal analysis techniques are required to investigate the mechanical fault features of centrifugal pumps. In this paper, an end-to-end pipeline for diagnosing faults in centrifugal pumps is proposed. To create a two-dimensional representation of the transients that appear in the vibration signals due to centrifugal pump operating conditions, first, a 1/3-binary tree fast kurtogram is computed. Next, a convolutional autoencoder and convolutional neural network are trained to autonomously extract global and local features from the kurtograms. Then, global, and local features are merged to form a joined feature vector that contains different visual features that are extracted using convolutional deep architectures using their specific loss functions during the training. Finally, this feature vector is propagated to a shallow-structured artificial neural network to accomplish fault identification. The proposed framework has been validated by the dataset collected from a real industrial centrifugal pump test rig. The results obtained during the series of experimental trials demonstrated that the introduced method achieved high classification accuracies when diagnosing faults based on signals collected under 3.0 and 4.0 bars of pressure.

INDEX TERMS Artificial neural network, centrifugal pump, convolutional autoencoder, convolutional neural network, deep learning, fault diagnosis, kurtogram, vibration signals.

I. INTRODUCTION

Centrifugal pumps (CPs) are an important electro-mechanical energy conversion machine and have become an important part of everyday business. According to the study made by the European energy agency, 65% of the energy produced in Europe is consumed by machines driven by electric motors, out of which CPs account for 80% of the energy consumption [1]. CPs are versatile, cheap, simple in construction, and reliable in operation. However, their unexpected failure may lead to severe consequences which include economic losses, energy losses, costly repairs, threats to the safety of operating staff, and long downtimes. To avoid these consequences and to ensure the reliable operation of CPs, health management has been gaining attention.

Several health management strategies (HMS), such as reactive maintenance, preventive maintenance, and predictive maintenance have been developed in past research [2]. Among these, HMS predictive maintenance, also called condition-based maintenance (CBM), recommends maintenance based on the data collected through condition monitoring. CBM maximizes the running time of the machine with minimum cost [3]. Considering these advantages, this study uses the CBM strategy for centrifugal pump fault diagnosis.

Research on CPs divides faults into two categories: fluid flow-related or hydraulic faults (HFs) and mechanical faults (MFs). However, HFs and MFs may not be completely independent of each other [4]. It means that a fault of one category

The associate editor coordinating the review of this manuscript and approving it for publication was Moussa Boukhnifer.

can be the source of fault occurrence in the other category. Mckee *et al.* [5] conducted a study on CP failure modes and explained that heavy pitting can lead to impeller imbalance and may result in HFs or MFs in the CP. Chittora [6] in the thesis work used failure modes, their effects, and criticality analysis (FMECA) to explain all the possible failure modes of mechanical seals, CPs, and the vibration response of a CP under failure. That work also stated that mechanical seal defects are responsible for 39% of CP failures. To keep the CP healthy, early detection of impeller and mechanical seal faults is of great interest. Hence, in this paper, MFs, which include mechanical seal faults (MSF) and impeller faults (IF) are carefully considered.

An MF or failure in the CP components results in abnormal vibration and impacts the signal. Due to this fact, it is common to use the vibration signal as an analytical signal [7], [8]. However, an abnormal vibration signal with a low amplitude can be disturbed by substantial background noise. To extract useful information about the fault from the raw vibration signals, signal preprocessing is required [9], [10]. For this reason, several signal processing techniques in time (TD), frequency (FD), and time-frequency domain (TFD) are developed in the past decades.

An MF in the CP results in a shock because of the stiffness around the mechanical component. These shocks produce variations in the amplitude of the vibration signal [11]. Statistical indicators, such as the root mean square (RMS), peak value, variance, etc. can be obtained from the TD vibration signal for fault detection in the CP [12]. TD statistical features are also useful for fault diagnosis of the gearbox [13] and bearings [14]. However, TD statistical features are not sensitive to subtly varying severity faults and can result in less discriminating information.

In order to extract discriminant features from raw vibration signals, it is necessary to overcome the shortcomings of TD statistical features. The FD is more appropriate for detecting CP faults since an unnoticeable change in the CP will produce an impulse in the corresponding frequency spectrum (FS). Statistical features are obtained from the FS of the CP vibration and were used for fault detection in [15]. Still, the vibration signals obtained from the CP are complex and non-stationary, while the Fourier transform (FT) applies only to stationary ones. Nonetheless, the FS is helpful for fault symptom identification and characterization. Jia *et al.* [16] utilized maximum correlated kurtosis deconvolution for the identification of periodic fault transients and spectral kurtosis for identifying these transients in the FD. Sun *et al.* [17] performed cyclic spectral analysis for the identification of frequency components related to cavitation and seal damage. Still, identifying and interpreting the fault characteristic frequencies from the FS is a complex process because it needs both experimental and mathematical validation.

The FD is appropriate for stationary signals; however, vibration signals obtained during the CP operation are non-stationary [18]. This causes limitations in applying FD-based signal analysis techniques to the valuable feature extraction from non-stationary signals. To address this problem, researchers have proposed various TFD-based techniques that have been efficiently applied for diagnosing mechanical faults based on nonstationary signals. Kang *et al.* [19] utilized time-varying and multiresolution envelope analysis for diagnosing rolling-element bearings. Prosvirin *et al.* [20] applied ensemble empirical mode decomposition for identifying blade rub-impact faults using nonstationary vibration signals collected from a nonlinear rotor system. Additionally, wavelet packet transformations enhanced with different base selection techniques have been employed for diagnosing CP faults in [4]. TFD approaches allow for a precise analysis of the complex nonstationary signals; however, reliable feature extraction models should still be used in conjunction with these types of techniques to properly characterize the mechanical faults.

In a conventional fault diagnosis pipeline, once the fault features have been extracted, decision-making on the state of the system should be performed. For this, supervised machine learning algorithms, such as support vector machines (SVM) [21], [22], k-nearest neighbor [23], decision trees [24], and naïve Bayes classifiers [25] are widely applied in the field of mechanical fault diagnosis. Different principles for decision-making are employed inside these algorithms; however, these techniques have a common issue. The significant drawback is that the classification performance, which can be measured by various metrics depending on the task and needs, is dependent on the quality of the features that are used for making a decision about the state of the system being investigated. Unfortunately, extracting or handcrafting the informative features requires a good level of field expertise and even in this scenario, it is almost impossible to guarantee that these features are optimal ones allowing for the high classification performance.

Deep learning-based approaches can help to address both the problems of informative feature extraction and fault classification. These techniques autonomously learn the features from the data representations and achieve a high level of generalization. The most popular deep learning techniques are convolutional neural networks (CNNs) [26]–[28] that can extract the highly-discriminative features from image patterns and accomplish the classification; various types of autoencoders (AEs) [29]–[32] that can be used for data or image compression, anomaly detection, and data generation by learning informative features from the input data; and generative adversarial networks (GANs) [33]–[35] that are frequently applied for data augmentation by learning the distribution of the input data. All of these techniques are very powerful themselves, but their performance depends on the quality of the input data representation. Additionally, neural component analysis shows promising results in the field of industrial processes monitoring [36].

In this paper, we propose a combined deep learning model for extracting features from the representations of the vibration signals collected from the CP and for decision making (i.e., fault classification). Here, the multiple convolutional

kernels of CNN that slide across the inputs to detect the patterns in specific local regions, covered by kernels, are used to extract the one part of a joined feature vector and this part is called ''local features'' in the text of this manuscript. Simultaneously, the latent coding located in the bottleneck layer of trained convolutional autoencoder (CAE) is considered as ''global features'' in the text of this manuscript, because these high-level features characterize the input data and allow CAE to accurately reconstruct it. This latent coding represents as the second part of a joined feature vector. Finally, this joined feature vector consisting of both global and local features is used for assessing the health state of the CP system using a shallow structured artificial neural network (ANN). The specific contributions of this work are as follow:

1) To improve the discriminatory properties of the vibration signals, their time-domain sequences are converted into 1/3-binary tree kurtogram forms to ensure the high quality of representation and uncover the detailed information about ongoing transient processes in the system;

2) To cope with the problems of conventional feature extraction approaches, the combined deep neural architecture consisting of CAE and CNN, further referred to as CAE-CNN, is proposed for autonomous fault feature extraction;

3) The ANN is built up on top of the CAE-CNN model to perform the decision-making by processing the feature vector containing the global and local features extracted by the CAE-CNN model. Combined, the introduced deep neural architecture represents end-to-end deep learning model that performs both fault feature extraction and decision making on a health state of a rotary system.

The remainder of this paper is organized as follows. Section II introduces the data collection system and the proposed fault feature extraction and diagnosis framework. Section III contains the experimental results and a discussion. Finally, Section IV concludes this manuscript.

II. PROPOSED METHODOLOGY

The proposed methodology for diagnosing CP mechanical faults is presented in Fig. 1 and contains four essential steps. First, the vibration acceleration signals are collected from the testbed using a data acquisition system. Next, the collected signals are converted into a kurtogram padding to explore the frequency changes in different subbands of the signals and to form the discriminative image-like patterns that can be used for fault detection and identification. In the third step, the tiny CNN and CAE are used in parallel, where the CNN is used to extract the local fault features from the kurtogram padding of the signal, and the CAE is utilized to detect the global features of the pattern using the data compression property of autoencoders. The difference between the extracted features is driven by the different types of loss functions used by the CNN and CAE during the training process. Finally, the local and global features are merged and pushed into the input layer of a shallow ANN that uses these discriminative local and global features for mechanical seal fault identification.

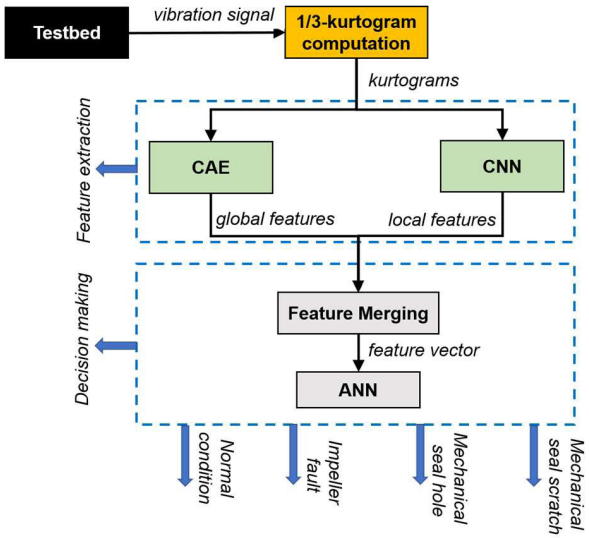

FIGURE 1. The proposed methodology for diagnosing centrifugal pumps.

A. EXPERIMENTAL PROCEDURE AND TEST RIG SETUP

In Fig. 2 a self-developed test rig was constructed that consists of several parts: a PMT- 4008 CP, which is driven by a 5.5 kW motor, a control panel, which contains controls for the speed, flowrate, temperature, water supply, ON/OFF switch, and a display screen. To keep a consistent net positive suction head at the pump inlet, two water tanks (a main tank and buffer tank) were used. The water tanks were placed at a sufficient height for the normal operation of the CP. The two tanks and the CP are connected through clear steel pipes having pressure gauges and a valve installed on the steel pipes. The test rig schematics are shown in Fig. 3 after establishing the test rig, the CP was operated at a constant speed of 1,733 rpm and the vibration data from the CP was collected using four accelerometers. Two of the accelerometers were installed on the pump casing, one was fixed close to the mechanical seal, and the other was fixed near the impellers using adhesive. Each accelerometer records the vibration of the CP using an independent channel. The recorded vibration of the CP was passed through a National Instruments 9234 device, which digitizes the acquired CP vibration signal. The details for the accelerometer and the digitization of the acquired CP vibration signal are given in Table 1.

TABLE 1. Specifications for the CP vibration data collection.

Accelerometer (622b01)	0.42 up to 10kHz frequency range 100 mV/g (10.2mV/(m/s^2)) ± 5% of sensitivity
(NI 9234) DAQ System	0 to 13.1MHz frequency range A generator having 4 analog input channels with 24 bits ADC resolution

The CP vibration data is acquired for 300 sec with a sampling rate of 25.6 kHz. First, a total of 1,200 samples,

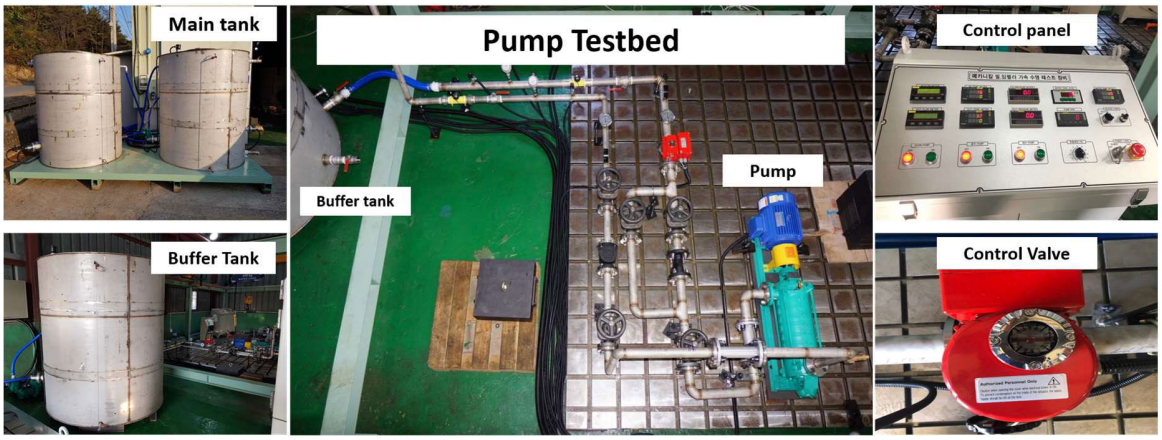

FIGURE 2. The experimental test rig used for data collection.

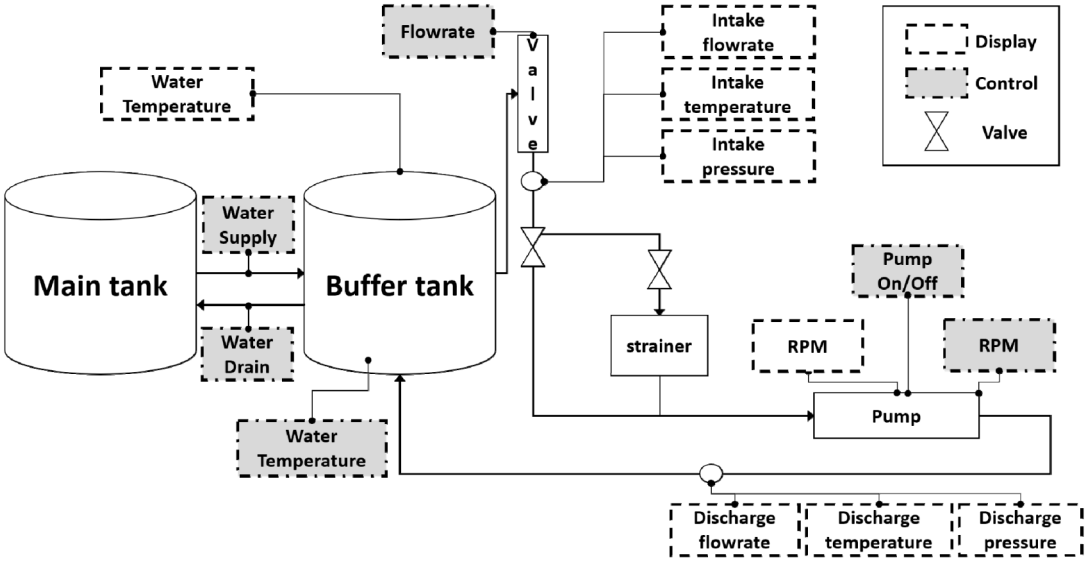

FIGURE 3. A schematic block-diagram of the experimental testbed.

each with a sample length of 25,600, were collected from the CP under different operation conditions having a 3.0 bar of pressure at the inlet of the CP. After obtaining samples from 3.0 bar pressure, the pressure level was increased to 4.0 bar at the inlet and a total of 1,200 samples, each with a sample length of 25,600, were collected from the CP under different operation conditions.

In this paper, the CP was operated under normal conditions, the mechanical seal hole defect condition, mechanical seal scratch defect condition, and impeller defect condition. These faults were introduced one at a time and the vibration data were collected from the CP. The specific properties of the collected dataset including the fault severity types are presented in Table 2.

TABLE 2. Number of samples collected under different faults types of closely varying severities.

Pressure level at the Inlet	Fault Type	No. of collected Samples	Fault severity
3.0 bars	Mechanical seal hole	300	Soft
4.0 bars	Mechanical seal hole	300	Soft
3.0 bars	Mechanical seal scratch	300	Soft
4.0 bars	Mechanical seal scratch	300	Soft
3.0 bars	Impeller fault	300	Soft
4.0 bars	Impeller fault	300	Soft

1) MECHANICAL SEAL FAULT

a: MECHANICAL SEAL HOLE (MSHF)

In this paper, mechanical seals with a 38mm inner diameter are used. A hole, 2.8 mm in diameter with a depth of 2.8 mm, was created in the rotating part of the seal to study the behavior of CP under the MSHF condition. This fault is depicted in Fig. 4.

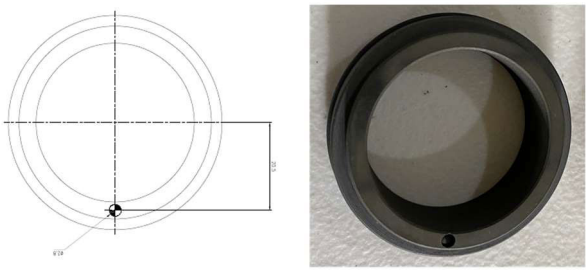

FIGURE 4. The mechanical seal hole fault.

b: MECHANICAL SEAL SCRATCH (MSSF)

For the scratch data, mechanical seals with a 38mm inner diameter are used. A severe fault due to a scratch of diameter 2.5 mm having a depth of 2.8 mm was created in the rotating part of the seal. This allows the study of the fault of a CP under the MSSF condition. This fault condition is presented in Fig. 5.

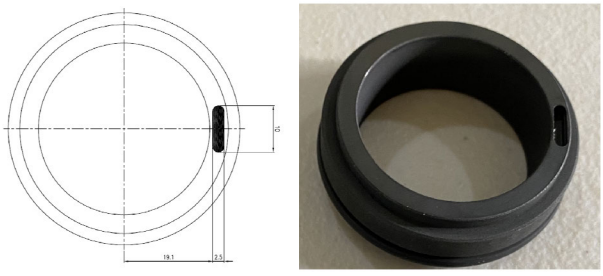

FIGURE 5. The mechanical seal scratch fault.

2) IMPELLER FAULT

In this paper, seven cast iron impellers of a diameter equal to 161 mm were used. Six of them were new ones and were kept free from defects. In the seventh impeller, a defect was created by removing some portion of the metal. The diameters of the faults were 2.5 mm, with a length of 18 mm and a depth of 2.8 mm. This allows the study of the fault of the CP under the defective impeller fault (IF) condition. This fault condition is demonstrated in Fig. 6.

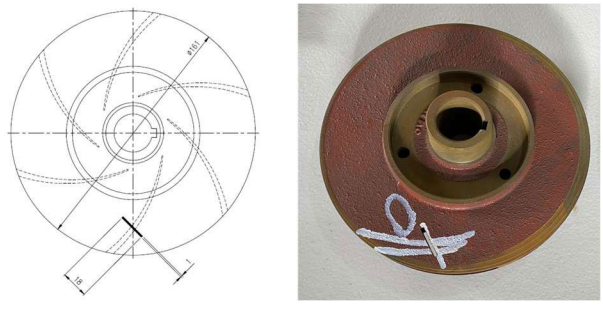

FIGURE 6. The impeller fault.

Additionally, the vibration response of CP under different operating conditions is shown in Fig.7. Details for the experimental setup can be found in [38].

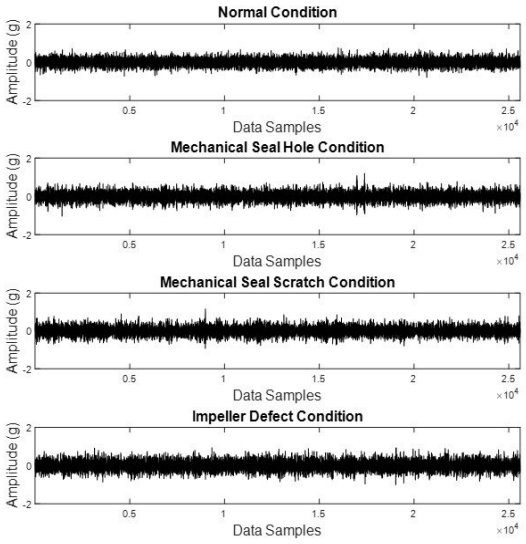

FIGURE 7. Vibration responses of CP under different operating conditions (a) Normal condition (b) Mechanical seal hole fault condition (c) Mechanical seal scratch fault condition (d) Impeller fault condition.

B. VIBRATION SIGNAL REPRESENTATION AS A 1/3-BINARY TREE KURTOGRAM

The kurtogram is a signal processing technique widely applied for the analysis of non-stationary and nonlinear signals. The kurtogram is usually represented as a <frequency/frequency resolution> padding in the form of a dyadic grid. However, in the case of complex signals, the original sampling of the kurtograms (i.e., decomposition by banks of low-pass/high-pass filters in form of a binary tree) cannot be representative enough and the narrow-band transients might not be appropriately detected. Furthermore, the industry provides strict requirements to a computational time for its algorithms used for fault feature extraction and identification. Hence, in this paper, the extension of the original kurtogram computation algorithm, called the 1/3-binary tree fast kurtogram [37], for analyzing the vibration signals of mechanical seals is utilized.

First, let us briefly discuss how the original binary tree fast kurtogram is formed. In the original approach, each sequence (including the original signal) is decomposed using two quasi-analytic low-pass and high-pass filters with the frequency ranges [0; 1/4] and [1/4;1/2] and then, downsampled by a factor of 2. To obtain the kurtogram representation, which is dyadic padding, this process should be iterated from the level $l = 0$ to level $L - 1$ and the kurtosis values of the outputs of decomposition should be computed. This idea can be expressed using the following formula:

$$\begin{aligned} Seq_l^m(n) \rightarrow decomp \rightarrow \downarrow 2 \rightarrow \begin{cases} Seq_{l+1}^{2m}(n) \\ Seq_{l+1}^{2m+1}(n), \end{cases} \\ s.b.t.l = 0, \ldots, L-1 \\ m = 0, \ldots, 2^l - 1 \end{aligned} \quad (1)$$

where *decomp* states for 'decomposition' using two quasi-analytic bandpass filters, m is the order number of filter,

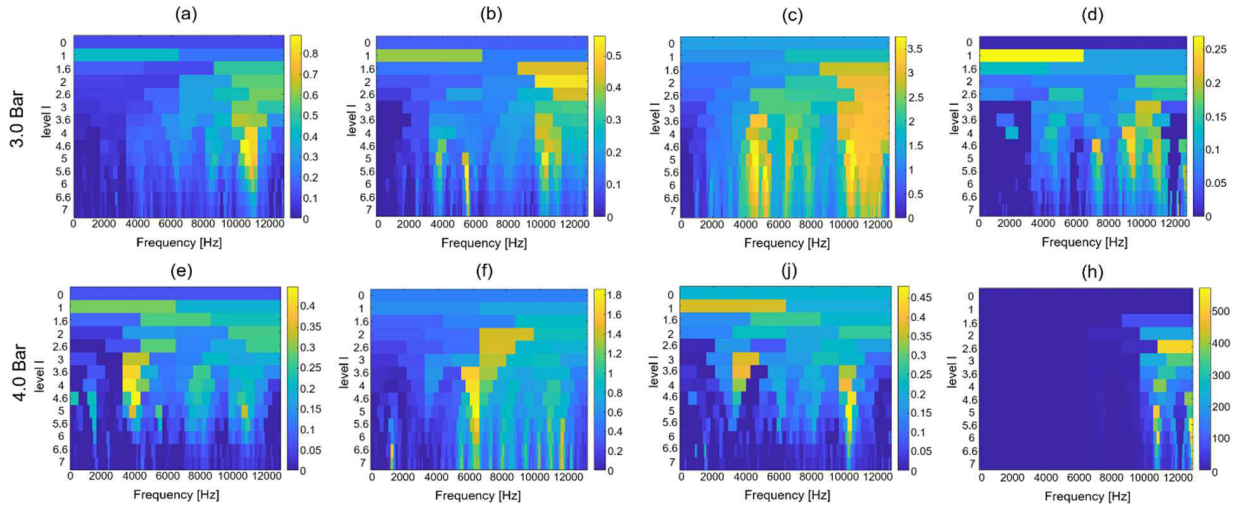

FIGURE 8. The 1/3-binary tree kurtogram patterns computed for the signals corresponded to (a) IF under 3.0 bar pressure, (b) MSHF under 3.0 bar pressure, (c) MSSF under 3.0 bar pressure, (d) normal system state under 3.0 bar pressure, (e) IF under 4.0 bar pressure, (f) mechanical seal hole under 4.0 bar pressure, (j) mechanical seal scratch under 4.0 bar pressure, and (h) normal system state under 4.0 bar pressure, respectively.

l states for the level of decomposition. The meaning of (1) is that each input sequence $Seq_l^m(n)$ after being decomposed and downsampled produces two new subsequences at the next level $l+1$. It can be understood that at each level, the number of decomposed sequences increases by a factor of 2, but their length decreases by the same factor. Note that at the level $1=0$, the $Seq_0(n)$ is equal to the original time-series signal, $x(n)$.

Regarding 1/3-binary tree fast kurtogram, its main idea can be summarized as follows. Unlike the original kurtogram, in the 1/3-binary tree kurtogram three additional quasi-analytic band-pass filters with the frequency ranges of [0;1/6], [1/6;1/3], and [1/3;1/2], are used for further decomposing $Seq_l^m(n)$ into three subsequences. After decomposition, those newly obtained subsequences are downsampled by a factor of 3 and are inserted between the levels l and $l+1$ of the original binary tree kurtogram (at a level $l+0.6$). This conception can be formulated as below:

$$Seq_l^m(n) \rightarrow decomp \rightarrow \downarrow 3 \rightarrow \begin{cases} Seq_{l,6}^{m+j}(n) \\ Seq_{l,6}^{m+j}(n) \\ Seq_{l,6}^{m+j}(n) \end{cases}$$
$$s.b.t.l = 0,\ldots,L-1$$
$$m = 0,\ldots,2^l-1$$
$$j = 0,1,2 \qquad (2)$$

Here, j identifies the low- (0), medium-(1), or high-frequency (2) part of the frequency interval $\left[m\cdot 2^{-l-1};(m+1)\cdot 2^{-l-1}\right]$. The meaning of the subscript $l,6$ is that there are $3\times 2^{l+0.6}$ sub-sequences inserted between the levels l and $l+1$ of the original binary tree.

Finally, the computation of the kurtogram is completed by computing the kurtosis values of all the sub-sequences obtained in the result of the decomposition process. For the details on fast kurtogram computation algorithm in general, and the 1/3-binary tree kurtogram extension specifically, the reader is encouraged to proceed to the source paper in [37].

In this paper, the level of decomposition is assigned to 7 because the CP vibration signals are known as highly nonstationary ones and a large order number is needed to properly detect and characterize the fault signatures of the CP. In the result of vibration signal decomposition into 1/3-binary tree kurtograms, the discriminative image-like patterns for each state of the system are obtained. The 1/3-binary tree kurtograms computed for the signals correspond to the normal operating state of the system, mechanical seal faults, and impeller faults under 3.0 and 4.0 bar pressure are depicted in Fig. 8.

From Fig.8 (a), it is evident that when an impeller fault occurs in the CP at a pressure of 3.0 bar, an abnormal shock, with a maximum kurtosis value of close to 1 at level $l=4.6$, appears. Similarly, when a mechanical seal hole fault occurs in the CP at a pressure of 3.0 bar at the inlet, an abnormal shock, with a maximum kurtosis of around 0.6 at level 5.6, reveals its presence, as can be seen from Fig.8 (b). Regarding the CP mechanical seal scratch fault at the 3.0 bar pressure, an abnormal transient with maximum kurtosis of around 4 at level 5 appears in the kurtogram pattern, which is demonstrated in Fig.8 (c). However, it can be noticed from Fig.8 (d) that when the CP operates under the normal condition with a normal pressure of 3.0 bar at the inlet, the maximum kurtosis value demonstrated in the kurtogram has been reduced to 0.3. It is worth noting from Fig.8 (e), (f), (g), (h) that when the pressure of the fluid increases from 3.0 to 4.0 bars at the inlet, the kurtosis value of around 0.5 at level $l=4.6$ appears for the impeller fault, a kurtosis of around 1.9 at level $l=5$ appears for the mechanical seal hole fault, a kurtosis of around 0.5 at level $l=5$ appears for a mechanical seal

scratch fault and a kurtosis of around 600 at level $l = 5.6$ appears for the normal condition. The changes in the kurtosis indicator for the CP fault conditions and normal condition are observed due to an increase in the pressure level at the pump inlet, which causes pressure pulsations in the CP, as can be observed from Fig.8 (h) - the high value of the kurtosis under the normal operating condition at level $l = 5.6$.

C. GLOBAL AND LOCAL FEATURE EXTRACTION

In this paper, the combined deep learning approach is proposed to extract both the local and global fault features from the kurtogram patterns of the vibration signals. Specifically, the local fault features are extracted using a CNN, whereas the global features are extracted using the property of image compression of the CAE. The following subsections provide the details on each of these techniques that are used in parallel.

All of the deep neural architectures used in this research were implemented and tested using the Keras Deep Learning library for the Python programming language.

1) LOCAL FAULT FEATURE EXTRACTION USING A CNN

Originally, CNNs appeared based on the study of the visual cortex of the brain. Now, with the rapid increase of computational capabilities of computers, they represent an important type of representation learning techniques that are frequently applied for image and natural language processing. However, due to their excellent performance on those tasks, CNNs also drew attention in the field of fault diagnosis and the condition monitoring of rotating machinery. Specifically, their property of extracting valuable high-level features that are insensitive to shifts and rotations from the image patterns of the signals made CNNs a great tool that allows automating of the feature extraction and fault classification process.

In this paper, the CNN represents a part of the fault feature extraction model and serves the role of extracting local fault features from the kurtogram patterns delivered from the vibration signals. In general, a CNN consists of a set of convolutional and pooling layers that are responsible for extracting the features, followed by a fully-connected layer, where the local extracted features are flattened, and an output layer.

In the convolutional layer, various convolutional filters are used to perform the convolution operation on the input kurtogram pattern. It is important to notice that in convolutional layers applied to 3D images, the convolutional layers are applied to each channel of the image separately. At the output of convolutional layers, a set of feature maps are obtained by activation functions. The operation of each convolution layer m can be generalized using the following expression:

$$\mathbf{x}_c^m = a^m\left(\sum\nolimits_{k=1}^{K^{m-1}} \mathbf{W}_{k,c}^m * \mathbf{x}_k^{m-1} + \mathbf{b}_c^m\right), \tag{3}$$

where m corresponds to the order number of the current convolutional layer; $*$ denotes the two-dimensional convolutional process of the channel $k = 1, \ldots, K^{m-1}$ on the input into this convolutional layer $\mathbf{x}_k^{m-1}$, $\mathbf{W}_{k,c}^m$ stands for the weights of the filter of the c^{th} filter in layer m; $\mathbf{b}_c^m$ corresponds to the bias term of filter c in convolutional layer m, and $a^m(\cdot)$ is a nonlinear activation function used to obtain the feature map (the output of the m^{th} convolutional layer). As a nonlinear activation function, the rectified linear unit (ReLU) activation function [38] is used in this paper.

In most of the known CNN architectures, as in the proposed one, the convolutional layers are followed by pooling layers. The main purpose of the pooling operation is to extract valuable information from feature maps while reducing the amount of data needed to be processed, and hence, reduce the time and memory requirements needed for the operation of CNNs. There are several types of pooling, including mixed pooling, spectral pooling, average pooling, max pooling, and others [39]. The general expression of the feature maps obtained after the pooling layer can be expressed as below:

$$\mathbf{x}_c^{\mathbf{m}} = \boldsymbol{\beta}_c^m down\left(\mathbf{x}_c^{m-1}\right) + \mathbf{b}_c^m. \tag{4}$$

Here, $down(\cdot)$ states for the downsampling operation; $\mathbf{x}_c^{\mathbf{m}}$ is the output of a pooling layer; $\mathbf{x}_c^{m-1}$ is the output of the previous layer and the input of the current pooling layer; $\boldsymbol{\beta}_c^m$ is a multiplicative bias, whereas $\mathbf{b}_c^m$ corresponds to an additive bias. In this paper, the max-pooling operation is used to reduce the size of feature maps by partitioning it into a set of segments and outputting the maximum value of each segment.

After completing a series of convolutional and pooling operations, the high-level local featured are computed. These features are then flattened as $\mathbf{x}^m = \text{vec}(\mathbf{x}^{m-1})$ and pushed into a fully-connected layer, where they are weighted and their activation function is computed. In this work, the CNN used for extracting local features contains only one fully-connected layer, which is also an output layer of this network. The formula of this output fully-connected layer is shown below:

$$\mathbf{x}^m = a^m\left(\boldsymbol{W}^m\mathbf{x}^{m-1} + \mathbf{b}^m\right). \tag{5}$$

Here, $\mathbf{x}^m$ is the output of a fully-connected output layer, a is an activation function, and $\mathbf{W}^m$ and $\mathbf{b}^m$ are the weight matrix and bias of the fully-connected layer, respectively. Since mechanical seal fault diagnosis is a multi-class classification problem, the activation function $\alpha(\cdot)$ of the output fully-connected layer is assigned to be a SoftMax activation that allows computing the probabilities of a sample belonging to the particular class. The SoftMax activation function is given as follows:

$$\hat{P}_l = exp\left(\mathbf{s}_l(x)\right) \Big/ \sum\nolimits_{j=1}^{L} exp\left(\mathbf{s}_j(x)\right), \tag{6}$$

where L is the total number of classes and $\mathbf{s}(x)$ is a vector with the scores of every available class for the specific data sample x. The input data sample is assigned to the class with the highest estimated probability $\hat{P}_l$ (i.e., the class that has the highest computed score for this instance).

To train the proposed CNN to extract high-level local features from the data sample, the categorical cross-entropy loss

TABLE 3. The detailed architecture of CNN submodule.

Layer #, Type	# Filters / # Nodes	Kernel Size	Output Shape	Activation
#1, Conv.	8 Filters	3x3	128x128x8	ReLU
#2, MaxPool	8 Filters	2x2	64x64x8	-
#3, Conv.	8 Filters	3x3	64x64x8	ReLU
#4, MaxPool	8 Filters	2x2	32x32x8	-
#5, Conv.	8 Filters	3x3	32x32x8	ReLU
#6, MaxPool	8 Filters	2x2	16x16x8	-
#7, Conv.	8 Filters	3x3	16x16x8	ReLU
#8, MaxPool	8 Filters	2x2	8x8x8	-
#9, Flatten	512 Nodes	-	512	-
#10, Output	4 Nodes	-	4	SoftMax

function is used with the outputs of the SoftMax activation to perform a decision making about the state of the system. The categorical cross-entropy loss can be formulated as below:

$$Loss\,(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{L} y_l^i \log\left(\hat{P}_l\right), \tag{7}$$

where θ is the set of model parameters and y_l^i and $\hat{P}_l$ are the target and estimated probabilities that the $i-th$ data sample belongs to the class l, respectively.

To prevent the overfitting of the CNN model, dropout regularization is applied in this study. Specifically, a dropout with a rate equal to 0.2 is applied to the fully-connected layer, where the local features extracted by convolution-pooling operations were flattened. A rate of 0.2 means that at each epoch of the training process 20% of features randomly were 'dropped out' and did not participate in the learning procedure. Once the training is completed, the activations of this fully-connected layer are considered as local features and are further merged with the global features to perform fault diagnosis of mechanical seals. The detailed architecture of CNN utilized for local fault feature extraction in this paper is depicted in Table 3. Note, that through the whole network the zero-padding and a stride of size equal to 1 are used.

2) GLOBAL FAULT FEATURE EXTRACTION USING CAE

a: BASIC AUTOENCODER BACKGROUND

The basic autoencoder is usually represented as a neural network that consists of three layers: input, hidden, and output. Here, the input and hidden layers comprise the encoder network, while the same hidden layer in conjunction with the output layer produces a decoder network. The goal of the encoder is to learn the latent coding (hidden representation) of the input data. Opposite to the encoder, the decoder part is used for reconstructing the input data from the learned hidden representation. The details are provided as follows.

The encoder network receives the input data $\mathbf{x}$ and transforms it into a hidden representation $\mathbf{h}$ using the mathematics of hidden layers:

$$\mathbf{h} = a\left(\mathbf{W_1}\mathbf{x} + \mathbf{b_1}\right), \tag{8}$$

where $\mathbf{W_1}$ and $\mathbf{b_1}$ are the weight matrix and a bias vector of hidden (bottleneck layer) and a is a nonlinear activation function. The decoder network uses the learned hidden representation $\mathbf{h}$ to reconstruct the input data $\mathbf{x}$ using the equation below:

$$\hat{\mathbf{x}} = a\left(\mathbf{W_2}\mathbf{h} + \mathbf{b_2}\right), \tag{9}$$

where $\hat{\mathbf{x}}$ is a reconstructed input of the autoencoder and $\mathbf{W_2}$ and $\mathbf{b_2}$ are the weight matrix and a bias vector of the output layer, respectively.

For training the autoencoders, usually, the mean squared error (MSE) loss function is used. Thus, the autoencoders attempt to minimize the reconstruction error between the input data and the result obtained at the output layer. Therefore, the more discriminative and of better quality this learned latent coding is, the smaller the reconstruction error can be achieved. Due to these properties, under 'global' features, the latent codings learned by autoencoder are understood in this paper. The loss function for training the autoencoder is represented below:

$$Loss\left(\mathrm{x},\hat{x}\right) = \left\|\mathbf{x} - \hat{x}\right\|^2 \tag{10}$$

The overall autoencoder operation can be summarized by:

$$\begin{aligned} encoder &: \mathbf{h} \leftarrow \mathbf{x} \\ decoder &: \hat{\mathbf{x}} \leftarrow \mathbf{h} \\ encoder, decoder &= \mathrm{argmin}_\theta\, Loss\left(\mathbf{x}, \hat{\mathbf{x}}\right) \end{aligned} \tag{11}$$

Above, the basic three-layer autoencoder operation has been described; however, the number of hidden layers can be easily increased, and the number of nodes in these hidden layers can be changed to create a deep autoencoder architecture to resolve the specific class of problem. The only important restriction is that the number of nodes in the output layer should match the number of nodes in the input layer to compute the loss function and perform the training of the deep learning architecture.

b: CONVOLUTIONAL AUTOENCODER (CAE)

A CAE is a convolutional autoencoder that is used for extracting the global features from the kurtogram representations of the vibration signals in this paper using the property of data compression inherent in autoencoders. Since the CAE receives the image at its input, the encoder part of CAE consists of a set of convolutional layers followed by pooling layers and one fully-connected layer. Symmetrically, the decoder part of CAE consists of one fully-connected layer followed by a set of transposed convolutional layers used to perform the deconvolution operation [40], [41], and this transforms the latent coding in the bottleneck layer of the CAE into an image pattern. The exact architecture of the CAE utilized in this paper is provided in Table 4.

TABLE 4. The detailed architecture of the CAE submodule.

Layer #, Type	Filters / Nodes #	Kernel Size	Output Shape	Activation
#1, Conv.	8 Filters	3x3	128x128x8	ReLU
#2, MaxPool	8 Filters	2x2	64x64x8	-
#3, Conv.	8 Filters	3x3	64x64x8	ReLU
#4, MaxPool	8 Filters	2x2	32x32x8	-
#5, Conv.	8 Filters	3x3	32x32x8	ReLU
#6, MaxPool	8 Filters	2x2	16x16x8	-
#7, Conv.	8 Filters	3x3	16x16x8	ReLU
#8, MaxPool	8 Filters	2x2	8x8x8	-
#9, Flatten	512 Nodes	-	512	-
#10, Reshape	-	-	8x8x8	-
#11, ConvT	8 Filters	3x3	16x16x8	ReLU
#12, ConvT	8 Filters	3x3	32x32x8	ReLU
#13, ConvT	8 Filters	3x3	64x64x8	ReLU
#14, ConvT	8 Filters	3x3	128x128x8	ReLU
#15, Conv.	3 Filters	3x3	128x128x3	ReLU

As can be seen in Table 4, the encoder part of the CAE resembles the architecture of a CNN, described in subsection 2.C.1), with the only difference that the fully-connected layer, 512 nodes in size, which is a bottleneck layer of CAE, is not connected to the output layer. However, in the decoder part of the CAE, another type of layer - transposed convolutional layers (ConvT) can be observed. The idea behind ConvT layers is opposite to one of the conventional convolutional layers [39]. The convolutional layer maps multiple input activations to a single one, while the transposed convolutional layer associates a single activation with multiple output activations. Specifically, the ConvT layer first upsamples its input by a factor equal to a stride value with padding and then performs a convolutional operation on the upsampled input. Thus, to reconstruct the input vibration signal pattern (kurtogram) from the latent coding, the stride value was assigned to 2 in ConvT layers and the padding was kept equal to zero.

To briefly describe the process of transposed convolution, let us assume that $\mathbf{L}^m$ is the convolution matrix corresponded to a convolution and summation operations of the layer m and multiple feature maps (i.e., the outputs) of the previous layer are transformed into a vector $\mathbf{s}^{m-1}$. Then, the output $\mathbf{x}^m$ of a convolution layer can be represented as follows:

$$\mathbf{x}^m = \mathbf{a}^m\left(\mathbf{L}^m\mathbf{s}^{m-1} + \mathbf{b}^m\right). \tag{12}$$

To obtain a reconstruction of the dimensions of $\mathbf{s}^{m-1}$ from the output $\mathbf{x}^{\mathbf{m}}$, the convolution matrix $\mathbf{L}^m$ should be transposed to perform a backward pass of a convolution process (i.e., transposed convolution) [42]. It is important to note that the transposed convolution process does not deliver the same output as the input, but delivers the output of the same spatial dimensionality and energy. However, in deep learning, this is usually not considered as the issue because the network parameters and weights of kernels (filters) are adjusted during the training process. Concerning (12), the operation of transposed convolution layer can be roughly represented as bellow:

$$\hat{\mathbf{y}}^m = \mathbf{a}^m\left((\mathbf{L}^m)^T\mathbf{x}^m + \mathbf{b}^m\right), \tag{13}$$

where $\hat{\mathbf{y}}^m$ is the output of transposed convolution layer of the same spatial dimensions as $\mathbf{s}^{m-1}$ from equation (12).

3) THE CAE-CNN MODEL FOR GLOBAL AND LOCAL FEATURE EXTRACTION

In Fig. 9, the complete model for extracting global and local features is represented. As can be seen from the figure, the local features are extracted from the input kurtogram using a CNN, while the global ones are extracted using a CAE. After training the CNN and CAE models, the outputs of their fully-connected layers, 512 nodes in size (global and local features extracted from the kurtogram), are merged. These features are used for accomplishing the fault identification problem using the shallow ANN architecture with a SoftMax classifier.

D. FAULT IDENTIFICATION BY ANN

The ANN used for fault identification in this study has a shallow architecture and consists of three layers: one input later that receives the global and local features extracted by the CNN-CAE model in its input, one hidden layer used for linear transformation of the features using the matrix multiplication operator, and an output layer with a SoftMax activation function. This ANN is needed to perform the decision about the state of the rotor system using the features extracted by the CNN-CAE model from the input kurtogram of the vibration signal; hence, the categorical cross-entropy is a loss function used for accomplishing the multiclass classification task.

The operation of the output layer can be described similarly, as was presented in (5) and (6). However, a small difference exists in the input and hidden layer of the ANN. In this work, the activation function is not applied to these layers and thus, the formulation of operation of the input and hidden layers of the ANN can be represented using the following formula:

$$\mathbf{x}^m = \mathbf{W}^m\mathbf{x}^{m-1} + \mathbf{b}^m, \tag{14}$$

where $\mathbf{x}^m$ is the output of the m^{th} layer that can be an input or hidden one, $\mathbf{x}^{m-1}$ is the output of the previous layer, and $\mathbf{W}^m$ and $\mathbf{b}^m$ are the respective weight matrix and bias vector of the m^{th} layer. Note, that when this equation is applied to the input layer of ANN, as the output of the previous layer ($\mathbf{x}^{m-1}$) the vector comprising merged global and local features should be understood, so $\mathbf{x}^{m-1} = \{\mathbf{x}^{FC\_CNN} \cup \mathbf{x}^{FC\_CAE}\}$, where $\mathbf{x}^{FC\_CNN}$ are the features obtained in flattened features extracted by CNN and $\mathbf{x}^{FC\_CAE}$ are the flattened features delivered by CAE.

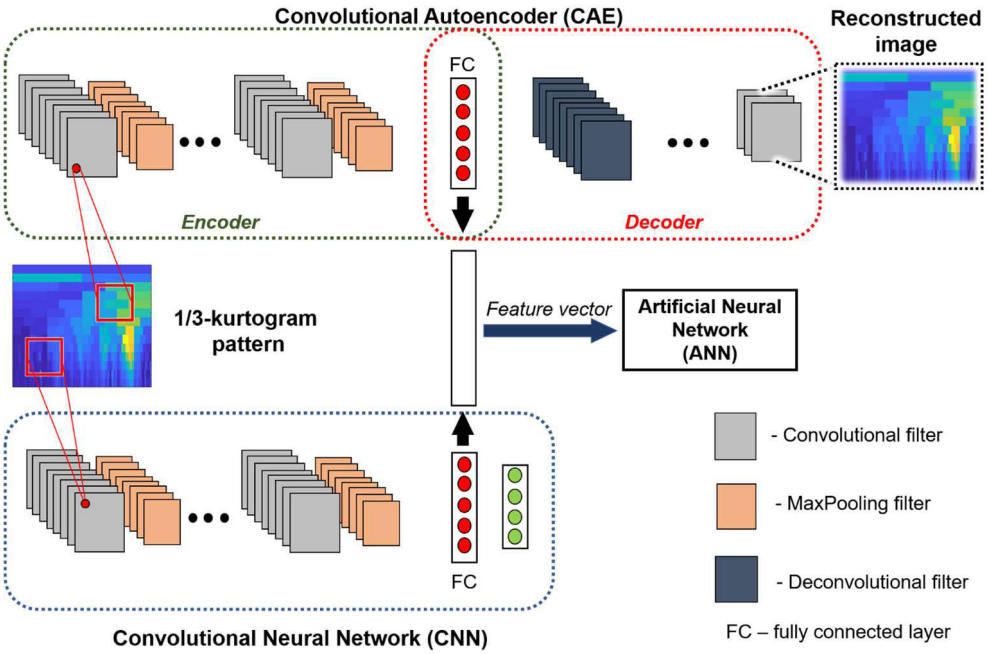

FIGURE 9. The proposed CAE-CNN model for extracting local and global features from the kurtogram patterns.

TABLE 5. The detailed architecture of the ANN submodule.

Layer #, Type	# Nodes	Activation	Dropout (rate)
#1, Input	1024	-	Yes, 0.2
#2, Hidden	512	-	Yes, 0.2
#3, Output	4	SoftMax	-

The exact architecture of the ANN used for fault identification in this study is presented in Table 5. To avoid overfitting during the training, dropout regularization at a rate of 0.2 is applied to the input and hidden layers.

E. THE COMPLETE MODEL FOR DIAGNOSING FAULTS

To provide a better understanding of the proposed feature extraction and fault diagnosis model operation, the mathematical summary is provided in this subsection. To recover, at the first step, the CNN and CAE modules of the proposed model work on the feature extraction task. Let $\mathbf{x}_k^{inp}$ be an input image pattern of the kth color channel, then the process of extracting local features by CNN using (3)-(5) and Table 3 can be summarized as below:

$$\begin{aligned}
\text{Layer 1}: \mathbf{x}_c^1 &= a^1\left(\sum\nolimits_{k=1}^{K^{inp}} \mathbf{W}_{k,c}^1 * \mathbf{x}_k^{inp} + \mathbf{b}_c^1\right)\\
\text{Layer 2}: \mathbf{x}_c^2 &= \boldsymbol{\beta}_c^2 down\left(\mathbf{x}_c^1\right) + \mathbf{b}_c^2\\
&\cdots\\
\text{Layer 7}: \mathbf{x}_c^7 &= a^7\left(\sum\nolimits_{k=1}^{K^6} \mathbf{W}_{k,c}^7 * \mathbf{x}_k^6 + \mathbf{b}_c^7\right)\\
\text{Layer 8}: \mathbf{x}_c^8 &= \boldsymbol{\beta}_c^8 down\left(\mathbf{x}_c^7\right) + \mathbf{b}_c^8\\
\text{Layer 9}: \mathbf{x}^{FC_CNN} &= \text{vec}(\mathbf{x}^8)\\
\text{Layer 10}: \mathbf{x}^m &= softmax\left(\boldsymbol{W}^m \mathbf{x}^{FC_CNN} + \mathbf{b}^m\right).
\end{aligned} \tag{15}$$

Despite CNN has an output layer with a softmax activation function, this output layer is needed for training the deep neural architecture. Out of these equations, $\mathbf{x}^{FC\_CNN}$ is of our interest which represents a flattened vector of local features extracted by series of convolutional-maxpooling layers of the CNN.

Simultaneously with CNN, CAE processes the input $\mathbf{x}_k^{inp}$ to derive the latent coding which is considered as a vector of global features in this manuscript. The operation of CAE can be summarized using the (3-5) as for CNN, (12-13) for transposed convolution operation, and the architecture presented in Table 4. This summary is represented as below:

$$\begin{aligned}
\text{Layer 1}: \mathbf{x}_c^1 &= a^1\left(\sum\nolimits_{k=1}^{K^{inp}} \mathbf{W}_{k,c}^1 * \mathbf{x}_k^{inp} + \mathbf{b}_c^1\right)\\
\text{Layer 2}: \mathbf{x}_c^2 &= \boldsymbol{\beta}_c^2 down\left(\mathbf{x}_c^1\right) + \mathbf{b}_c^2\\
&\cdots\\
\text{Layer 7}: \mathbf{x}_c^7 &= a^7\left(\sum\nolimits_{k=1}^{K^6} \mathbf{W}_{k,c}^7 * \mathbf{x}_k^6 + \mathbf{b}_c^7\right)\\
\text{Layer 8}: \mathbf{x}_c^8 &= \boldsymbol{\beta}_c^8 down\left(\mathbf{x}_c^7\right) + \mathbf{b}_c^8\\
\text{Layer 9}: \mathbf{x}^{FC_CAE} &= \text{vec}(\mathbf{x}^8)\\
\text{Layer 10}: \mathbf{x}_c^R &= \text{vec}_{8,8,8}^{-1}(\mathbf{x}^{FC_CAE})\\
\text{Layer 11}: \hat{\mathbf{x}}^{11} &= a^{11}\left((\mathbf{L}^{11})^T \mathbf{x}^R + \mathbf{b}^{11}\right)
\end{aligned}$$

$$\begin{aligned} &\cdots \\ \text{Layer 14} : \hat{\mathbf{x}}^{14} &= a^{14}\left((\mathbf{L}^{14})^{T}\mathbf{x}^{13} + \mathbf{b}^{14}\right) \\ \text{Layer 15} : \mathbf{x}_c^{15} &= a^{15}\left(\sum_{k=1}^{K^{14}} \mathbf{W}_{k,c}^{15} * \mathbf{x}_k^{14} + \mathbf{b}_c^{8}\right). \end{aligned} \tag{16}$$

Here, layers 1 to 8 are similar to ones of CNN and correspond to the encoder part of CAE. The vector $\mathbf{x}^{FC\_CAE}$ is of our interest since it represents the latent coding learned by CAE from which it is possible to reconstruct the input image patterns. These latent codings are considered as global features in this manuscript. Layers 10-15 are the layers of the decoder parts of CAE which are used to reconstruct the inputs using the learned global features. Here, layer 10 represents the reshaping operation on the vector of global features and layers 11 to 14 are the layers performing transposed convolution operation and layer 15 is a single convolution layer used to recover the spatial dimensionality of the original inputs $\mathbf{x}_k^{inp}$.

Once the local features $\mathbf{x}^{FC\_CNN}$ are extracted by CNN and global features $\mathbf{x}^{FC\_CAE}$ are extracted by CAE, these features are fed into ANN to complete the fault classification. The operation of ANN used in this paper can be summarized as follows:

$$\begin{aligned} \mathbf{x}^{FM} &= \{\mathbf{x}^{FC\_CNN} \cup \mathbf{x}^{FC\_CAE} \\ \text{Layer 1} : \mathbf{x}^{\mathbf{inp}} &= \left(\boldsymbol{W}^{inp}\mathbf{x}^{FM} + \mathbf{b}^{inp}\right) \\ \text{Layer 2} : \mathbf{x}^{h} &= \left(\boldsymbol{W}^{h}\mathbf{x}^{inp} + \mathbf{b}^{h}\right) \\ \text{Layer 3} : \mathbf{x}^{out} &= softmax\left(\boldsymbol{W}^{\mathbf{out}}\mathbf{x}^{h} + \mathbf{b}^{out}\right). \end{aligned} \tag{17}$$

That is, after the features were merged, the simple structured ANN accomplishes the decision-making on the health state of the system.

III. EXPERIMENTAL RESULTS

In this section, the proposed model for diagnosing mechanical seals fault is evaluated using the datasets corresponding to two different pressure levels and compared against other counterpart methodologies.

A. TRAINING, VALIDATION, AND TESTING DATA CONFIGURATION

For evaluating the fault diagnosis performance of the framework introduced in this paper, the training, validation, and testing subsets should be organized. For this, first, the complete dataset consisting of 1200 kurtograms (300 kurtograms for each data class) is randomly split into training and testing subsets at a ratio of 8:2. Next, the training subset is randomly split again at the same ratio to create the final training subset and validation subset that will be used to observe the improvements of the validation loss during the model training. Hence, the final training subset consisted of 960 data samples. The validation subset comprised 192 data instances and the testing dataset was represented by the remaining 240 kurtograms that were unseen during the model training. The training-validation-testing subset organization of methods used in the comparison is held at the same rate as the ones presented for the proposed framework. To eliminate the effect of randomness, the experiment was performed ten times where, in each experiment, the complete dataset has been randomly split.

B. CONVERGENCE ANALYSIS

In this subsection, the convergence of the proposed methodology is analyzed and the possible directions for further improvement are presented. To analyze the convergence properties, the training scenario of the proposed technique should be defined first. In this work, the training of each module of the model is left under the control of the early stopping algorithm. The idea of this algorithm is to stop the training process once the value of the validation loss function stops decreasing or starts increasing. The minimal improvement of validation loss as a stopping criterion for training has been assigned to 2×10^{-5} experimentally. This value is a tradeoff between the validation classification accuracy and convergence speed in this work.

Once the training scenario was set, the training and validation procedure was repeated ten times. It is important to mention that the CAE and CNN modules of the proposed model were trained in parallel for extracting global and local features from the kurtogram representations of the vibration signals. The validation loss convergence curves for each module of the proposed model obtained during experiment #2 for the 4.0 bar pressure dataset are presented in Fig. 10.

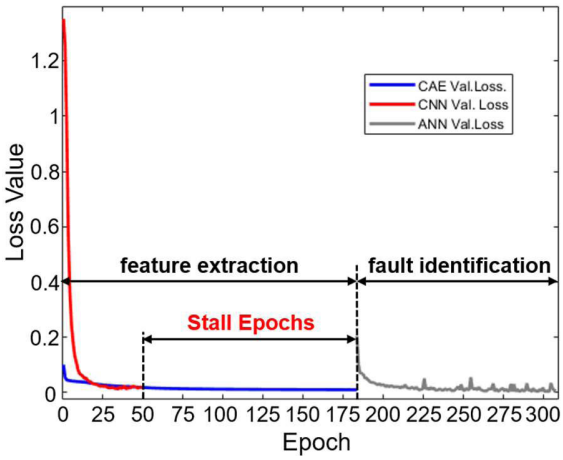

FIGURE 10. The validation loss curves obtained by the proposed technique during experiment #2 on the dataset collected under 4.0 bar pressure.

From Fig. 10 it can be seen that the loss curves corresponding to each of the modules of the proposed framework successfully converged to the values close to zero. However, due to the difference in the complexity of the training procedure of the CNN and CAE modules, the number of epochs needed until convergence is different. This caused the stalling issue when one feature extraction module completed its training and to proceed further with fault diagnosis, time is required to complete the training of the second feature extraction module.

TABLE 6. The convergence loss function end-values in ten experiments For 3.0 and 4.0 bar pressure datasets.

Pressure 3.0 Bar											
Experiment	CAE			CNN			ANN			Total	Stall
	Epochs	Tr. Loss	Val. Loss	Epochs	Tr. Loss	Val. Loss	Epochs	Tr. Loss	Val. Loss	Epochs	Epochs
1	138	0.0088	0.0083	101	0.0184	0.0429	77	0.0718	0.0244	215	37
2	142	0.0087	0.0090	125	0.0083	0.0273	69	0.0689	0.0210	211	17
3	195	0.0080	0.0083	70	0.0420	0.0561	108	0.0339	0.0052	303	125
4	158	0.0083	0.0081	99	0.0073	0.0743	53	0.1167	0.1124	211	59
5	175	0.0089	0.0082	62	0.0416	0.1010	110	0.0652	0.1190	285	113
6	206	0.0079	0.0080	71	0.0241	0.0648	84	0.1252	0.0837	290	135
7	171	0.0088	0.0088	85	0.0046	0.0321	70	0.0576	0.0180	241	86
8	170	0.0090	0.0090	72	0.0260	0.0484	79	0.0739	0.0524	249	98
9	149	0.0086	0.0083	101	0.0167	0.0290	78	0.0638	0.0292	227	48
10	160	0.0087	0.0089	68	0.0203	0.0892	65	0.0674	0.0522	225	92
Average	**166.4**	**0.0085**	**0.0084**	**85.4**	**0.0209**	**0.0565**	**79.3**	**0.0744**	**0.0517**	**245.7**	**81**
Pressure 4.0 Bar											
Experiment	CAE			CNN			ANN			Total	Stall
	Epochs	Tr. Loss	Val. Loss	Epochs	Tr. Loss	Val. Loss	Epochs	Tr. Loss	Val. Loss	Epochs	Epochs
1	153	0.0103	0.0104	93	0.0038	0.0649	83	0.0466	0.1025	236	60
2	184	0.0108	0.0106	50	0.0155	0.0189	125	0.0567	0.0051	309	134
3	152	0.0102	0.0104	101	0.0018	0.0036	77	0.0665	0.0146	229	51
4	166	0.0100	0.0097	62	0.0092	0.0295	124	0.0261	0.0064	290	104
5	170	0.0097	0.0092	109	0.0020	0.0045	109	0.0339	0.0017	279	61
6	149	0.0104	0.0103	57	0.0053	0.0481	77	0.0905	0.1154	226	92
7	199	0.0101	0.0097	71	0.0168	0.0322	90	0.0419	0.0196	289	128
8	177	0.0098	0.0101	57	0.0051	0.0153	115	0.0304	0.0277	292	120
9	157	0.0104	0.0100	79	0.0049	0.0410	73	0.0520	0.0337	230	78
10	176	0.0100	0.0101	51	0.0330	0.0745	33	0.0602	0.1002	209	125
Average	**168.3**	**0.0101**	**0.0100**	**73**	**0.0097**	**0.0332**	**90.6**	**0.0504**	**0.0426**	**258.9**	**95.3**

More details related to convergence analysis of the proposed technique, when trained on both datasets collected under different pressures, are presented in Table 6.

Despite the good convergence obtained for the two datasets collected under different pressure levels during ten experiments, it can be seen that to speed up the overall training procedure, the number of stalling epochs should be reduced, or the proposed methodology should be improved to eliminate this problem.

C. FAULT IDENTIFICATION PERFORMANCE AND COMPARISONS

To evaluate the efficacy of the proposed fault identification methodology, it is compared with five various counterpart techniques that include the conventional fault diagnosis framework and the recent deep learning-based solutions. The first two approaches described are the methods from the literature that were explicitly designed for diagnosing CP fault types. The first method used for the comparison comprises a wavelet packet transformation (WPT) used for vibration signal preprocessing and principal component analysis (PCA) applied for selecting the optimal WPT bases and to perform the extraction of statistical feature parameters [4]. This method is further referred to as WPT-PCA-MSVM. In the second approach, the WPT is first used for signal preprocessing and the best energy criteria is next applied for selecting the optimal bases for extracting the valuable features [4]. This method is further referred to as WPT-BE-MSVM. The third scheme from the family of conventional fault diagnosis pipelines represents the statistical feature parameter extraction from the time-domain vibration signal and SVM for accomplishing the fault identification task. This technique is further referred to as TSF-SVM in the text of this manuscript. Since the proposed technique in this paper represents the family of the deep learning methods, it is decided to add one of the state-of-the-art from the same family of techniques for comparison purposes. Specifically, this is the method that has been successfully applied for diagnosing mechanical faults in rolling-element bearings utilizing envelope power spectrum analysis and one-dimensional CNN [43]. This technique is further referred to in the text as EPS-CNN. In the final method used in this comparison, the CNN architecture from the proposed scheme alone is applied for mechanical seal fault diagnosis with the kurtogram of the vibration signal as its input. This approach is further referred to as Kurt-CNN and allows us to investigate whether the addition of the features extracted by CAE to ones extracted by CNN affects the fault diagnosis performance or not.

The fault diagnosis results for the proposed and referenced methods are expressed in terms of weighted recall, precision, and f1-score, along with the fault classification accuracy that can be computed using the formulas below:

$$Recall = \frac{\sum_m^M n_m \times \left(TP_m / \left(TP_m + FN_m\right)\right)}{N} \tag{18}$$

$$Precision = \frac{\sum_m^M n_m \times \left(TP_m / \left(TP_m + FP_m\right)\right)}{N} \tag{19}$$

TABLE 7. The Results averaged over ten experiments For 3.0 and 4.0 bar pressure datasets.

Methods	PRESSURE 3.0 BAR				PRESSURE 4.0 BAR			
	Metrics (%)				Metrics (%)			
	Recall	Precision	F1-Score	Accuracy	Recall	Precision	F1-Score	Accuracy
Proposed	98.67	98.72	98.67	98.67	99.42	99.43	99.42	99.42
Kurt-CNN	97.08	97.18	97.09	97.08	98.88	98.90	98.87	98.88
EPS-CNN	98.58	98.62	98.59	98.58	97.83	97.88	97.83	97.83
WPTPCA-SVM	98.20	98.23	98.20	98.20	98.95	98.97	98.95	98.95
WPTBE-SVM	95.04	95.11	95.03	95.04	99.29	99.30	99.29	99.29
TSF-SVM	96.66	96.73	96.66	96.66	92.91	98.94	92.89	92.91

$$\boldsymbol{F1} = \frac{1}{N}\sum_{\boldsymbol{m}}^{\boldsymbol{M}} \boldsymbol{n_m} \times 2 \times \frac{(\boldsymbol{Recall_m} \times \boldsymbol{Precision_m})}{(\boldsymbol{Recall_m} + \boldsymbol{Precision_m})} \tag{20}$$

$$Accuracy = \frac{\sum_m^M TP_m}{N} \tag{21}$$

Here, TP_m, FP_m, and FN_m are the true positive, false positive, and false negative values computed for the data samples belonging to the class m; n_m is the total number of samples belonging to the class m; M stands for the total number of classes (equal to four in this study); and N is the total number of data instances in the testing datasets. The metrics of the experimental results, presented in (18-21), are averaged over ten experimental trials and are tabulated in Table 7.

Fig. 11 shows the statistics of testing accuracies demonstrated by all the techniques during the ten experiments. The black cross inside the boxes in this figure corresponds to the mean fault classification accuracy demonstrated by the methods. Additionally, the confusion matrices presenting the performance of all the methods on 3.0 and 4.0 bar datasets and averaged over 10 experimental trials are depicted in Fig. 12 and 13.

From Table 7 and Fig. 11, it can be seen that the proposed methodology outperformed the referenced fault diagnosis frameworks used in this comparison. Specifically, the fault classification accuracy for the proposed method averaged over ten experimental trials is equal to 98.67% for the data corresponding to 3.0 bar pressure and 99.42% for the instances belonging to 4.0 bar pressure datasets, respectively. Furthermore, during all the experiments using both datasets, the fault classification accuracy demonstrated by the proposed technique did not fall below 97.9%. Fig. 14 represents the linear discriminants obtained by applying Linear Discriminant Analysis to the joint feature vector containing features extracted by CAE and CNN. From this figure it can be seen that despite there is a some degree of overlap between feature projections, overall, the features belonging to different classes are well-separable so ANN module used for decision-making in this work can achieve high classification accuracy.

On the contrary, Kurt-CNN (representing the same CNN network as one used in the proposed CAE-CNN model) achieved an averaged fault identification accuracy of 97.08% and 98.88% for the data samples corresponding to datasets collected under 3.0 bar and 4.0 bar pressure, respectively. Also, from the boxplots depicted in Fig.11, it can be observed that during ten experiments the fault diagnosis accuracy experienced a drop until a value of 93.8%. Furthermore, some outliers deviating from the mean and median accuracy can be seen. Comparing to the proposed technique, the boxplots corresponding to Kurt-CNN have longer whiskers which highlights the deviations of results for some of the experiment trials from the mean accuracy value. Based on these

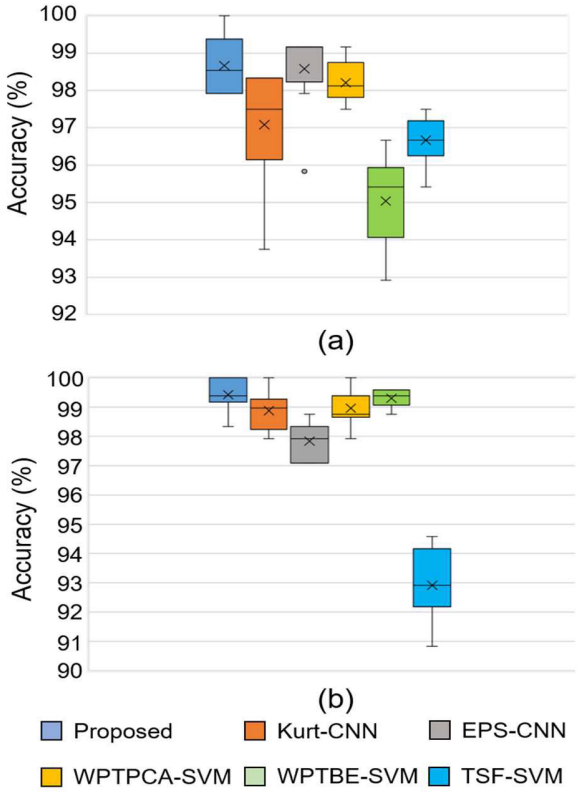

FIGURE 11. The statistics of fault classification accuracy demonstrated by various techniques during ten experiments on (a) 3.0 bar pressure dataset and (b) 4.0 bar pressure dataset, respectively.

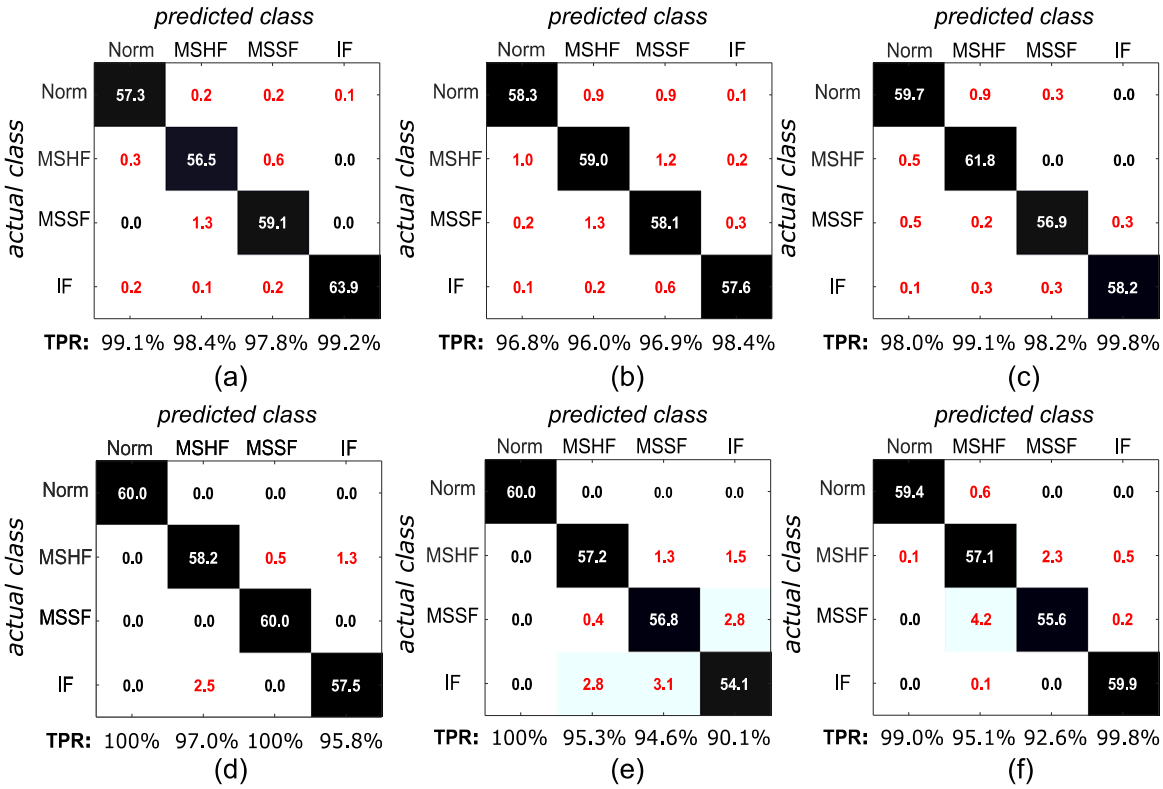

FIGURE 12. The confusion matrices averaged over 10 experimental trials under 3.0 bar pressure for (a) proposed, (b) Kurt-CNN, (c) EPS-CNN, (d) WPTPCA-SVM, (e) WPTBE-SVM, and (f) TSF-SVM techniques, respectively.

observations it is possible to conclude that the simultaneous application of the global and local features extracted by CAE and CNN (CAE-CNN model), in conjunction with a tiny ANN for decision making, can increase the fault classification accuracy and stabilize the fault diagnosis performance (reduce the deviations from the mean accuracy value when the experiment repeated multiple times with random dataset permutations).

Talking about EPS-CNN, from the experimental results it can be observed that this technique reached fault classification accuracy values averaged over 10 experimental trials of 98.58% and 97.83% on data collected under 3.0 and 4.0 bar pressure, respectively. In this technique, the envelope power spectra of vibration signals are computed and then are fed into a one-dimensional CNN for extracting features and diagnosing faults. Based on the experimental results it is concluded that despite one-dimensional CNN is capable of achieving high fault classification accuracy, the use of envelope power spectra as the patterns for representation learning problem might not be discriminative enough for a task of diagnosing CP mechanical faults. One of the reasons for this behavior is that the fault frequency components of CP are spread in a wide frequency range, including a high-frequency one, which may cause CNN to extract non-discriminative features from these one-dimensional sequences.

In regard to WPTPCA-MSVM, this method demonstrated the averaged fault classification accuracy of 98.20% and 98.95% for the data instances belonging to datasets collected under 3.0 and 4.0 bar pressure, respectively. In this approach, the vibration signal is decomposed by the WPT up to 2 levels having 4 bases. The number of bases here corresponds to the dimensions of the data. The application of PCA for dimensionality reduction requires the assumption that the most valuable information hidden in the data is contained in the components with high values of variance. After PCA, the first two higher-variance WPT bases, which contributed 70% of data covariance, have been selected for statistical feature extraction. The degraded fault classification performance in comparison with the proposed model can be explained by the information loss during the PCA dimensionality reduction as well as the common issues inherent to the WPT, such as the dependence of the result on the mother wavelet function.

Regarding the WPTBE-MSVM approach, this method demonstrated the averaged fault classification accuracy of 96.66% and 92.91% for the data instances belonging to datasets collected under 3.0 and 4.0 bar pressure, respectively. Here, the vibration signal was first decomposed up to 2 levels having 4 bases using WPT. Then, the energy values were calculated for each of the bases, and bases are arranged in descending order concerning the energy values.

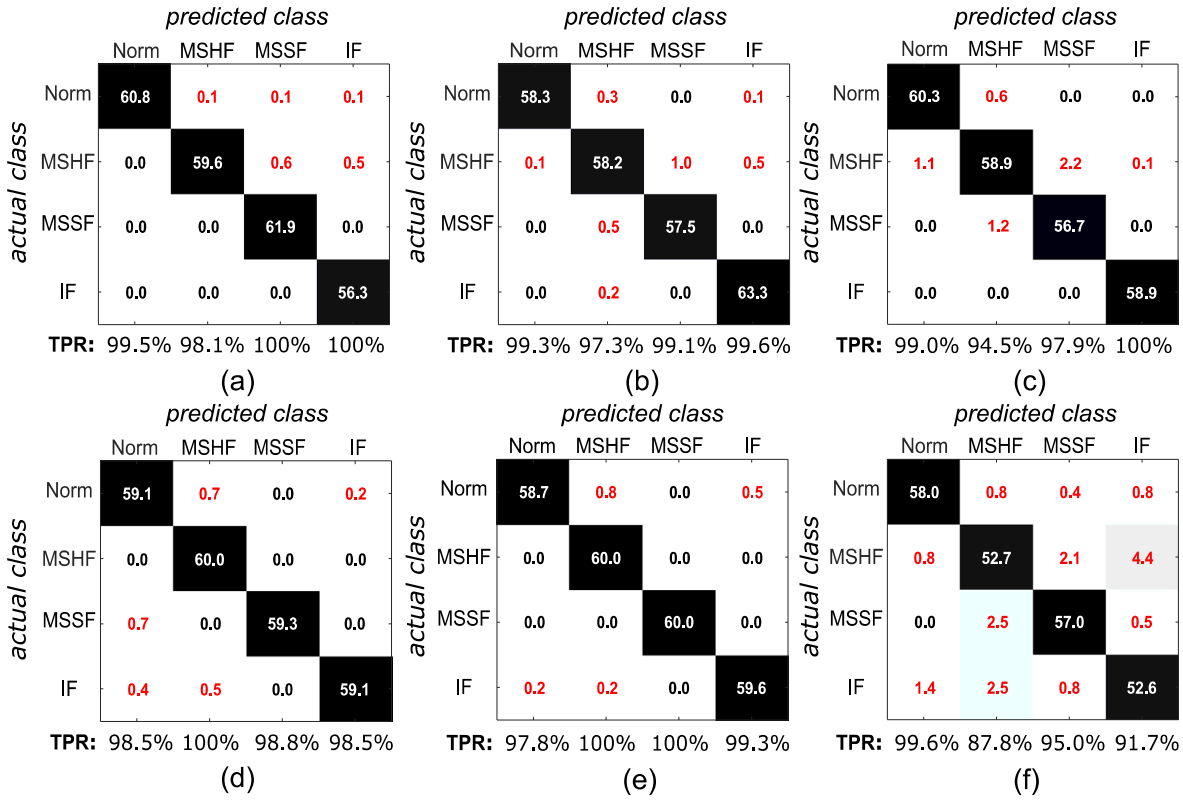

FIGURE 13. The confusion matrices averaged over 10 experimental trials under 4.0 bar pressure for (a) proposed, (b) Kurt-CNN, (c) EPS-CNN, (d) WPTPCA-SVM, (e) WPTBE-SVM, and (f) TSF-SVM techniques, respectively.

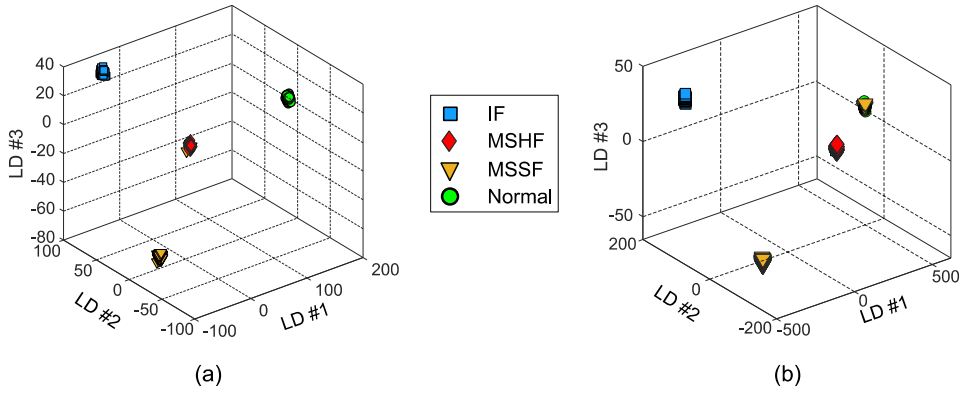

FIGURE 14. The linear discriminants of the joined feature vector consisting of global and local features obtained for datasets collected under (a) 3.0 bar and (b) 4.0 bar pressure levels, respectively.

After arranging the WPT bases, statistical features were used to grade each of the bases based on the achieved classification accuracy. There are two main reasons for the observed degradation of classification performance during the analysis. First, it has been observed that the selection of the decomposition bases based on the energy criterion proposed in the referenced literature may lead to a loss of valuable information about several fault subtypes. The second reason is similar to the one discussed for the WPT-PCA-MSVM technique and this is the need for finding an appropriate mother wavelet function to achieve high-quality decomposition results.

In regard to the TSF-SVM technique, where the statistical feature parameters for diagnosing CP faults have been extracted from the time-domain vibration signals, this approach demonstrated an averaged fault classification accuracy of 95.04% and 92.91% for the data instances belonging to datasets collected under 3.0 and 4.0 bar pressure, respectively. The classification accuracy underperformance for this technique can be explained by the fact that the

vibration signals collected from the testbed include the random fluctuations caused by the CP faults. Thus, the features obtained from these signals without filtering or preprocessing might fail to properly describe the fault processes being investigated.

Overall, it can be concluded that the proposed methodology solves the problem of centrifugal pump fault diagnosis at high classification accuracy levels under different data permutations that also provide some insights about the capabilities of delivering stable performance during the repeated experiment trials. However, from the convergence analysis, it can be seen that still there is a window for the improvement of the proposed technique such as addressing the problem of training stall.

IV. CONCLUSION

In this paper, a novel deep learning-based approach is proposed for fault feature extraction and diagnosing mechanical faults in centrifugal pump systems. Specifically, first, the vibration signals collected from the system are converted into a two-dimensional kurtogram representation to reveal insights into the processes ongoing in the centrifugal pump system and to detect transients. Next, the convolutional autoencoder and convolutional neural network are learned simultaneously to extract global and local features from the kurtogram patterns. Finally, those extracted features are merged and propagated to a shallow artificial neural network to complete fault identification. The experimental results demonstrated that the proposed framework reached a high classification accuracy of 98.67% and 99.42% averaged over ten experimental trials with datasets collected under 3.0 and 4.0 bar pressure, respectively. In future research, we will focus on resolving the training stall issue of the current version of the technique as well as comparing its efficacy when other types of signals, such as acoustic emission ones, are used for assessing the condition of the system.

REFERENCES

[1] H. Vogelesang, "An introduction to energy consumption in pumps," *World Pumps*, vol. 2008, no. 496, pp. 28–31, 2008, doi: 10.1016/S0262-1762(07)70434-0.

[2] R. B. Randall, Ed., *Vibration-Based Condition Monitoring: Industrial, Aerospace and Automotive Applications*, 1st ed. Hoboken, NJ, USA: Wiley, 2011.

[3] H. Nandi and A. K. Ahmed, *Condition Monitoring With Vibration Signals: Compressive Sampling and Learning Algorithms for Rotating Machines*, vol. 1. Hoboken, NJ, USA: Wiley, 2019.

[4] J. S. Rapur and R. Tiwari, "Experimental fault diagnosis for known and unseen operating conditions of centrifugal pumps using MSVM and WPT based analyses," *Measurement*, vol. 147, Dec. 2019, Art. no. 106809, doi: 10.1016/j.measurement.2019.07.037.

[5] K. McKee, G. Forbes, M. I. Mazhar, R. Entwistle, and I. Howard, "A review of major centrifugal pump failure modes with application to the water supply and sewerage industries," in *Proc. ICOMS Asset Manage. Conf.*, Gold Coast, QLD, Australia, 2011, pp. 1–12.

[6] S. M. Chittora, "Monitoring of mechanical seals in process pumps," KTH Roy. Inst. Technol., Stockholm, Sweden, Tech. Rep. TRITA-ITM-EX 2018:686, 2018.

[7] A. K. Panda, J. S. Rapur, and R. Tiwari, "Prediction of flow blockages and impending cavitation in centrifugal pumps using support vector machine (SVM) algorithms based on vibration measurements," *Measurement*, vol. 130, pp. 44–56, Dec. 2018, doi: 10.1016/j.measurement.2018.07.092.

[8] Z. Yunlong and Z. Peng, "Vibration fault diagnosis method of centrifugal pump based on EMD complexity feature and least square support vector machine," *Energy Procedia*, vol. 17, pp. 939–945, Dec. 2012, doi: 10.1016/j.egypro.2012.02.191.

[9] Y. Si, Y. Wang, and D. Zhou, "Key-performance-indicator-related process monitoring based on improved kernel partial least squares," *IEEE Trans. Ind. Electron.*, vol. 68, no. 3, pp. 2626–2636, Mar. 2021, doi: 10.1109/TIE.2020.2972472.

[10] X. Ma, Y. Si, Z. Yuan, Y. Qin, and Y. Wang, "Multistep dynamic slow feature analysis for industrial process monitoring," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 12, pp. 9535–9548, Dec. 2020, doi: 10.1109/TIM.2020.3004681.

[11] M. A. Abu-Zeid and S. M. Abdel-Rahman, "Bearing problems' effects on the dynamic performance of pumping stations," *Alexandria Eng. J.*, vol. 52, no. 3, pp. 241–248, Sep. 2013, doi: 10.1016/j.aej.2013.02.002.

[12] A. R. Al-Obaidi, "Investigation of effect of pump rotational speed on performance and detection of cavitation within a centrifugal pump using vibration analysis," *Heliyon*, vol. 5, no. 6, Jun. 2019, Art. no. e01910, doi: 10.1016/j.heliyon.2019.e01910.

[13] L. Cao, J. Zhang, J. Wang, and Z. Qian, "Intelligent fault diagnosis of wind turbine gearbox based on long short-term memory networks," in *Proc. IEEE 28th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2019, pp. 890–895, doi: 10.1109/ISIE.2019.8781108.

[14] X. Yan, Y. Liu, M. Jia, and Y. Zhu, "A multi-stage hybrid fault diagnosis approach for rolling element bearing under various working conditions," *IEEE Access*, vol. 7, pp. 138426–138441, 2019, doi: 10.1109/ACCESS.2019.2937828.

[15] J. S. Rapur and R. Tiwari, "Automation of multi-fault diagnosing of centrifugal pumps using multi-class support vector machine with vibration and motor current signals in frequency domain," *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 40, no. 6, pp. 1–21, Jun. 2018, doi: 10.1007/s40430-018-1202-9.

[16] F. Jia, Y. Lei, H. Shan, and J. Lin, "Early fault diagnosis of bearings using an improved spectral kurtosis by maximum correlated kurtosis deconvolution," *Sensors*, vol. 15, no. 11, pp. 29363–29377, Nov. 2015, doi: 10.3390/s151129363.

[17] H. Sun, S. Yuan, and Y. Luo, "Characterization of cavitation and seal damage during pump operation by vibration and motor current signal spectra," *Proc. Inst. Mech. Eng., A, J. Power Energy*, vol. 233, no. 1, pp. 132–147, Feb. 2019, doi: 10.1177/0957650918769761.

[18] S. Li, N. Chu, P. Yan, D. Wu, and J. Antoni, "Cyclostationary approach to detect flow-induced effects on vibration signals from centrifugal pumps," *Mech. Syst. Signal Process.*, vol. 114, pp. 275–289, Jan. 2019, doi: 10.1016/j.ymssp.2018.05.027.

[19] M. Kang, J. Kim, L. M. Wills, and J.-M. Kim, "Time-varying and multiresolution envelope analysis and discriminative feature analysis for bearing fault diagnosis," *IEEE Trans. Ind. Electron.*, vol. 62, no. 12, pp. 7749–7761, Dec. 2015, doi: 10.1109/TIE.2015.2460242.

[20] A. E. Prosvirin, M. M. M. Islam, and J.-M. Kim, "An improved algorithm for selecting IMF components in ensemble empirical mode decomposition for domain of rub-impact fault diagnosis," *IEEE Access*, vol. 7, pp. 121728–121741, 2019, doi: 10.1109/ACCESS.2019.2938367.

[21] F. Piltan, A. E. Prosvirin, M. Sohaib, B. Saldivar, and J.-M. Kim, "An SVM-based neural adaptive variable structure observer for fault diagnosis and fault-tolerant control of a robot manipulator," *Appl. Sci.*, vol. 10, no. 4, p. 1344, Feb. 2020, doi: 10.3390/app10041344.

[22] A. Prosvirin, B. P. Duong, and J.-M. Kim, "SVM hyperparameter optimization using a genetic algorithm for rub-impact fault diagnosis," in *Advances in Computer Communication and Computational Sciences*, vol. 924, S. K. Bhatia, S. Tiwari, K. K. Mishra, and M. C. Trivedi, Eds. Singapore: Springer, 2019, pp. 155–165.

[23] R. N. Toma, A. E. Prosvirin, and J.-M. Kim, "Bearing fault diagnosis of induction motors using a genetic algorithm and machine learning classifiers," *Sensors*, vol. 20, no. 7, p. 1884, Mar. 2020, doi: 10.3390/s20071884.

[24] F. Piltan, A. E. Prosvirin, I. Jeong, K. Im, and J.-M. Kim, "Rolling-element bearing fault diagnosis using advanced machine learning-based observer," *Appl. Sci.*, vol. 9, no. 24, p. 5404, Dec. 2019, doi: 10.3390/app9245404.

[25] V. Muralidharan and V. Sugumaran, "A comparative study of Naïve Bayes classifier and Bayes net classifier for fault diagnosis of monoblock centrifugal pump using wavelet analysis," *Appl. Soft Comput.*, vol. 12, no. 8, pp. 2023–2029, Aug. 2012, doi: 10.1016/j.asoc.2012.03.021.

[26] A. Prosvirin, J. Kim, and J.-M. Kim, "Bearing fault diagnosis based on convolutional neural networks with kurtogram representation of acoustic emission signals," in *Advances in Computer Science and Ubiquitous Computing*, vol. 474, J. J. Park, V. Loia, G. Yi, and Y. Sung, Eds. Singapore: Springer, 2018, pp. 21–26.
[27] G. Jiang, H. He, J. Yan, and P. Xie, "Multiscale convolutional neural networks for fault diagnosis of wind turbine gearbox," *IEEE Trans. Ind. Electron.*, vol. 66, no. 4, pp. 3196–3207, Apr. 2019, doi: 10.1109/TIE.2018.2844805.
[28] R. Liu, G. Meng, B. Yang, C. Sun, and X. Chen, "Dislocated time series convolutional neural architecture: An intelligent fault diagnosis approach for electric machine," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1310–1320, Jun. 2017, doi: 10.1109/TII.2016.2645238.
[29] A. E. Prosvirin, F. Piltan, and J.-M. Kim, "Blade rub-impact fault identification using autoencoder-based nonlinear function approximation and a deep neural network," *Sensors*, vol. 20, no. 21, p. 6265, Nov. 2020, doi: 10.3390/s20216265.
[30] X. Wu, G. Jiang, X. Wang, P. Xie, and X. Li, "A multi-level-denoising autoencoder approach for wind turbine fault detection," *IEEE Access*, vol. 7, pp. 59376–59387, 2019, doi: 10.1109/ACCESS.2019.2914731.
[31] E. Principi, D. Rossetti, S. Squartini, and F. Piazza, "Unsupervised electric motor fault detection by using deep autoencoders," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 2, pp. 441–451, Mar. 2019, doi: 10.1109/JAS.2019.1911393.
[32] M. Cui, Y. Wang, X. Lin, and M. Zhong, "Fault diagnosis of rolling bearings based on an improved stack autoencoder and support vector machine," *IEEE Sensors J.*, vol. 21, no. 4, pp. 4927–4937, Feb. 2021, doi: 10.1109/JSEN.2020.3030910.
[33] S. A. Khan, A. E. Prosvirin, and J.-M. Kim, "Towards bearing health prognosis using generative adversarial networks: Modeling bearing degradation," in *Proc. Int. Conf. Advancements Comput. Sci. (ICACS)*, Lahore, Pakistan, Feb. 2018, pp. 1–6, doi: 10.1109/ICACS.2018.8333495.
[34] W. Mao, Y. Liu, L. Ding, and Y. Li, "Imbalanced fault diagnosis of rolling bearing based on generative adversarial network: A comparative study," *IEEE Access*, vol. 7, pp. 9515–9530, 2019, doi: 10.1109/ACCESS.2018.2890693.
[35] Q. Liu, G. Ma, and C. Cheng, "Data fusion generative adversarial network for multi-class imbalanced fault diagnosis of rotating machinery," *IEEE Access*, vol. 8, pp. 70111–70124, 2020, doi: 10.1109/ACCESS.2020.2986356.
[36] Z. Lou and Y. Wang, "New nonlinear approach for process monitoring: Neural component analysis," *Ind. Eng. Chem. Res.*, vol. 60, no. 1, pp. 387–398, Jan. 2021, doi: 10.1021/acs.iecr.0c02256.
[37] J. Antoni, "Fast computation of the kurtogram for the detection of transient faults," *Mech. Syst. Signal Process.*, vol. 21, no. 1, pp. 108–124, 2007, doi: 10.1016/j.ymssp.2005.12.002.
[38] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," p. 8.
[39] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018, doi: 10.1016/j.patcog.2017.10.013.
[40] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proc. Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 2018–2025, doi: 10.1109/ICCV.2011.6126474.
[41] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1520–1528, doi: 10.1109/ICCV.2015.178.
[42] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," 2016, *arXiv:1603.07285*. Accessed: Mar. 11, 2021. [Online]. Available: http://arxiv.org/abs/1603.07285
[43] D. K. Appana, A. Prosvirin, and J.-M. Kim, "Reliable fault diagnosis of bearings with varying rotational speeds using envelope spectrum and convolution neural networks," *Soft Comput.*, vol. 22, no. 20, pp. 6719–6729, 2018, doi: 10.1007/s00500-018-3256-0.

ALEXANDER E. PROSVIRIN received the degree in engineering from the Moscow State University of Mechanical Engineering (MSUME-MAMI) (currently, Moscow Polytechnic University), Moscow, Russia, in 2013, with a focus on control and informatics in technical systems, and the Ph.D. degree in computer engineering with the University of Ulsan, Ulsan, South Korea, in 2021.

He is currently a Postgraduate Research Fellow with the Ulsan Industrial Artificial Intelligence (UIAI) Laboratory, Department of Electrical, Electronics, and Computer Engineering, University of Ulsan. He has authored or coauthored more than 13 scientific articles in academic journals and four book chapters. His research interests include data-driven fault diagnosis and anomaly detection of complex engineering systems using artificial intelligence, advanced signal processing, fault feature extraction, and feature engineering.

Dr. Prosvirin has served as a reviewer for several recognized journals. He is also an editorial board member of an academic journal.

ZAHOOR AHMAD received the B.S. degree in computer engineering from the COMSATS Institute of Information Technology (CIIT) [currently, Comsats University Islamabad (CUI)], Attock, Pakistan, in 2016, the M.S. degree in electronics and information engineering from Korea Aerospace University (KAU), Goyang, South Korea. He is currently pursuing the Ph.D. degree in computer engineering with the University of Ulsan, South Korea.

Since 2019, he has been a Graduate Research Assistant with the Ulsan Industrial Artificial Intelligence (UIAI) Laboratory. He worked on the development of advanced algorithms for UAV path planning. His current research interests include artificial intelligence, signal processing, fault diagnosis, vibration-based condition monitoring of industrial machinery, and fault feature extraction.

Mr. Ahmad received the Institution's Highest Prize of Gold Medal for the B.S. degree from CUI, the Fully Funded Scholarship for the M.S. degree from the Korean Government Project, the National Research Foundation (NRF) of Korea Research Fellowship Scholarship, and the University of Ulsan President Excellence Scholarship for pursuing the Ph.D. degree.

JONG-MYON KIM (Member, IEEE) received the B.S. degree in electrical engineering from Myongji University, Yongin, South Korea, in 1995, the M.S. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2000, and the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2005.

He is currently a Professor with the Department of IT Convergence, University of Ulsan, Ulsan, South Korea. His research interests include multimedia-specific processor architecture, fault diagnosis and condition monitoring, parallel processing, and embedded systems.

Dr. Kim is a member of the IEEE Industrial Electronics Society.