Enhanced Bearing Fault Detection in Induction Motors Using Projection-Based SVM

Narges Khadem Hosseini, Hamid Toshani, Salman Abdi, Sara Sharifzadeh

***Abstract*— Fine-tuning the hyperparameters of the Support Vector Machine (SVM) is crucial to making it an accurate classifier in fault detection. The convergence rate of the optimiser and its simplicity are also vital for achieving the optimal boundary. Therefore, a highly accurate fault detection algorithm that combines SVM with a simple, easy-to-implement, and stable optimiser called Projection Recurrent Neural Network (PRNN) is proposed in this paper. The algorithm focuses on detecting bearing faults in induction motors using experimentally measured stator currents. The initial dataset is pre-processed using Discrete Wavelet Transform (DWT), Power Spectral Density (PSD), and cepstrum analysis. A feature set is then derived from several statistical metrics, and Kernel Principal Component Analysis (KPCA) is used to select the most salient features. The SVM is trained using these discriminative features, and its optimisation problem is reformulated as Constrained Nonlinear Programming (CNP). The PRNN is introduced to solve the CNP to effectively determine the optimal decision boundary of the SVM. The study evaluates the accuracy of the proposed algorithm and highlights the advantages of using PRNN in SVM, its dependence on the type of kernel function and the selected features. A 4 kW induction motor is used as a prototype machine for experimental data collection under healthy and faulty conditions. The results show that the proposed algorithm leads to more accurate fault detection compared to the conventional SVM and recent fault detection methods.**

***Index Terms*— Bearing fault, Support Vector Machine, Projection Recurrent Neural Network, Kernel Principal Component Analysis, Induction motor.**

I. INTRODUCTION

Induction Motors (IMs) are known for their reliability, robustness, and cost-effectiveness compared to other motor types. Their applications include pumps, compressors, fans, wind generators, and electric vehicles [1]. Bearing faults contribute to 51% of the total downtime experienced by IMs [2]. Therefore, developing accurate and reliable techniques to detect and diagnose the bearing faults in IMs is crucial.

Motor Current Signature Analysis (MCSA) is a common method for detecting bearing faults by examining frequency differences in healthy and faulty data. It identifies faults in theouter bearing part by analysing the motor's current signals [3]. To detect cracks in bearings, a new method was proposed in [4] that treats the signal over time as a combination of weighted sine waves and noise. Using the machine's frequency range, harmonic numbers, and the signal's autocorrelation matrix, thefault can be identified. In [5], the Root Mean Square (RMS), peak, and crest values of the sampled current in the stator phase were analysed. The algorithm compares samples over time, providing an alarm and an end-of-life indication, which shows bearing surface roughness and wear. In [6], the supply and rotation frequencies, along with their harmonics, were identified using MCSA and analysed from both theoretical and practical perspectives. The positions of the signatures in the current spectrum and fault severity were quantified using mean and standard deviation of Gaussian curves.

MCSA is simple and easy to use; nevertheless, its accuracy can be affected by speed changes, voltage differences, and noise [7]. In addition, this method has limitations in its ability to identify faults at an early stage. When multiple faults occur that share similar characteristics, the system faces challenges in accurately distinguishing between these similar faults. Further spectral analysis of current and vibration signals using DWT can be used for bearing fault detection. In [8], entropy analysis of wavelet coefficients from the third order decomposition ofthe residual signal is used for bearing fault classification. In [9], wavelet packet decomposition segments the raw signal into sub-signals. Then, the energy of the most significant coefficients is used as fault discriminators.

Machine learning tools such as SVMs are effective for identifying and classifying faults in IMs. SVMs excel at problem-solving, are resilient to excessive data or noise, and require less configuration than artificial neural networks. They handle diverse data sets and identify dividing lines within the data [10]. In [11], a wavelet transform model was proposed to extract significant engine signal parts. The generated data was used to train an SVM model showing effective performance infault detection. In [12], motor vibrations were analysed using a continuous wavelet transform for fault detection with SVMs.

In [13], variational mode decomposition is used to decompose the measured data into intrinsic mode components. The envelope spectrum entropy of these components was calculated to create the feature space. Bearing fault detection was then implemented using the SVM. The load current'sfrequency spectrum was obtained using FFT in [14]. Thisspectrum was then utilised by SVM, Random Forest, and a

feedforward neural network for bearing fault identification. In [15], FFT analysis was employed to identify the typical faulty signal frequency for cylindrical roller bearings. The extracted features were then used to train SVMs, and the method's accuracy was evaluated for various motor bearings.

In [16], cross-correlation along with various statistical metrics were applied to the current signals. The neighborhood component analysis was used to select the prominent features. To discriminate the bearing fault data, the SVM with multiple kernels, symmetric properties and kernel weights was developed. Fault detection in an induction motor bearing's outer raceway was achieved using SVM in [17]. Feature extraction was performed through FFT, and dimensionality reduction was implemented with PCA. In [18], a wavelet time scattering network based on convolution, modulus, and average pooling operations was proposed to distinguish low-variance features from the main signals. An ensemble binary SVM was then employed to classify the bearing fault data.

Multi-resolution analysis and FFT were employed in [19] to improve feature extraction from raw signals. The feature selection was achieved using grey wolf and heap-based optimisation. A robust SVM-based diagnostic method was developed in [20] that utilised stator current features. This method achieved high accuracy in diagnosing three common faults, with a focus on early detection. In [21], normalised vibration signals and their FFT were used to calculate statistical features. The prominent features were selected using the Fisher score algorithm, which aims to minimise the distance between data points within the same class. In the SVM model, grid search was employed to tune the hyperparameters. In [22], a meta-learning approach with the gradient-by-gradient rule was employed to optimise parameters for bearing fault diagnosis. This approach achieved high accuracy and adaptability even with limited data available.

In the presence of complex patterns in large datasets, solving the constrained optimisation problem within SVM can be challenging. The PRNN as a fast, easy-to-implement, and stable optimiser is an effective approach for linear and nonlinear optimisation problems [23]. This optimiser is used for quadratic programming [24], constrained nonlinear programming [25-26], and constrained L_1 norm optimisation [27]. Integrating PRNN with SVM enhances efficiency for identifying faults in induction motor bearings, which provides a simple, fast, and highly effective approach [28]. In this study, experimental tests were conducted on the prototype machine to obtain current signals at various speeds and loads and under healthy and fault conditions. The time and frequency-related features are extracted from the current signals and ranked using the KPCA algorithm. An enhanced SVM model with an integrated PRNN is developed and used to address the complex nonlinear programming problem with the ability to distinguish between healthy and faulty conditions.

The main contribution of the proposed work is the integration of SVM with PRNN, which provides fast convergence, stability and low computational complexity, ideal for real-time fault detection. The data processing framework in the paper using DWT, PSD and Cepstrum analysis along with several statistical measures provides a comprehensive feature space with both time and frequency information of the stator currents. In addition, the PRNN ensures optimal convergence through an analytical approach that overcomes problems such as slow optimisation and unstable convergence found in traditional SVM implementations. The proposed approach is suitable for practical implementation on resource-constrained systems and provides scalability for real-time applications.

The convergence of this integration towards the optimal hyperplane will be demonstrated. Furthermore, the algorithm processes time-domain signals using DWT, PSD, and Cepstrum analysis to extract critical frequency information. Using KPCA also simplifies SVM training, reduces data redundancy, and enhances the algorithm's accuracy.

The paper's structure is as follows: Section II presents the proposed fault detection method overview. Section III details the experimental setup. A mathematical preliminary regarding the kernel function, preimage problem and PRNN dynamics is presented in Section IV. The details of data pre-processing are given in Section V. Sections VI and VII discuss feature extraction and ranking. Section VIII outlines SVM and PRNN integration for fault detection. Sections X and IX present results and conclusions, respectively.

II. Proposed Fault Detection Algorithm

The proposed methodology, as illustrated in Fig. 1, spans the entire process from data collection to accurate defect detection. Current sensors along with Speedgoat as a high-performance target machine are used to measure stator stator currents. DWT, PSD, and cepstrum are applied to stator current signals to extract frequency information. Various statistical measures are used to characterise the feature space. The KPCA is then used to determine the dominant features, which are then used to train the SVM. To enhance the accuracy and improve effectiveness of the SVM, especially in real-time situations, the SVM is combined with PRNN. The PRNN solves the SVM's constrained optimisation problem.

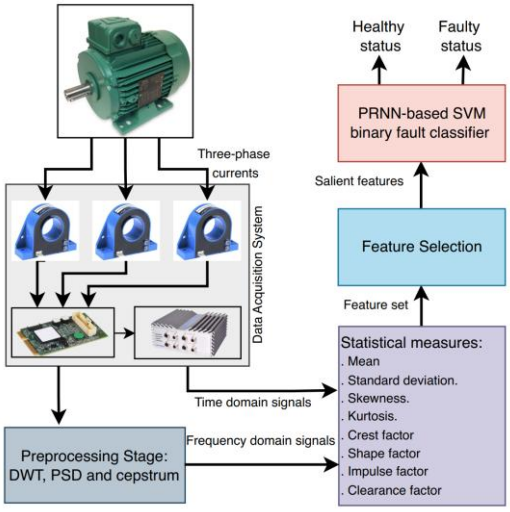

Fig. 1. Block diagram of the proposed fault detection algorithm.

As shown in Fig. 2, motor bearings consist of outer and inner rings with a cage holding evenly spaced balls to prevent contact [17]. In this study, a 6-mm-diameter hole was drilled into the outer race of the faulty bearing. Three-phase electrical current signals are collected from a 4 KW induction motor under both healthy and faulty conditions across various speeds ranging from 1000-1500 rpm and 0-90% load conditions with the step changes of 50 rpm and 10% for speed and load, respectively.

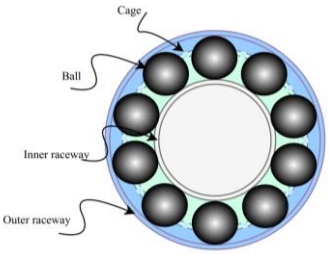

Fig. 2. Main parts of a rolling bearing.

Integrating the PRNN with the SVM addresses certain limitations inherent to traditional SVM implementations, particularly in terms of optimisation speed and convergence stability. This integration can improve the model's ability to find the optimal hyperplane in nonlinear feature spaces, particularly when the data exhibits intricate patterns or noise. The PRNN is a stable and computationally efficient optimiser that ensures fast convergence towards the optimal solution, reduces the risk of local minima traps that often challenge other optimisation methods. Additionally, the simple structure of PRNN minimises computational complexity, which makes it highly scalable for real-time fault detection applications. Using PRNN not only maintains the SVM's generalisation ability but also speeds up the convergence process. Since the proposed method uses only stator currents, it eliminates the need for vibration sensors on the motor, which can be expensive in various motor applications.

III. Experimental Test Rig Set-up

The experimental setup uses a test configuration with a 4 KW induction motor known as LSES112MU, as shown in Fig. 3. A torque transducer is positioned between two identical induction motors to measure the torque exerted by the motor. A data acquisition system consisting of LEM LA series current sensors, a target machine and an input/output (I/O) module is set up to measure the instantaneous three-phase stator currents observed in both normal and faulty scenarios. A low-pass filter was implemented on the board to eliminate high-frequency noise. The Speedgoat is equipped with an Intel 2.0 GHz quad-core central processing unit, a Simulink real-time operating system, 4 GB of DDR3 random access memory, and 4 input/output slots. The IO module is equipped with 13 differential input/output lines that act as channels for sending and receiving electrical signals. Three single-channel Hall effect sensors with a closed-loop bi-directional module measure currents up to 100 A, AC or DC. The sampling time is 1 millisecond and the ADC resolution is 16 bits.

Figure 4 illustrates the connection between the Speedgoat, computer, and induction motor. The real-time three-phase currents are transferred to the Speedgoat target machine using the IO191 module. The Speedgoat communicates with the computer via an Ethernet connection, and the Simulink Real-Time in MATLAB software uses the corresponding libraries to detect and log data.

The proposed method doesn't model the bearing faults, instead they are created experimentally by introducing cracks into the outer raceway of the bearings. The machine is equipped with two single-row, deep groove ball bearings. Each bearing has a width of 14 mm, an outside diameter of 62 mm, an inside diameter of 35 mm, and a pitch diameter of 48 mm. Inside each bearing, there are 11 balls, each with a diameter of 9.52 mm. The contact angle of the balls with the races is 0 degrees. Two identical motors are used, one with healthy bearings, and one with a faulty bearing. To create a localised fault, a 6-mm-diameter hole was drilled into the outer race of the faulty bearing.

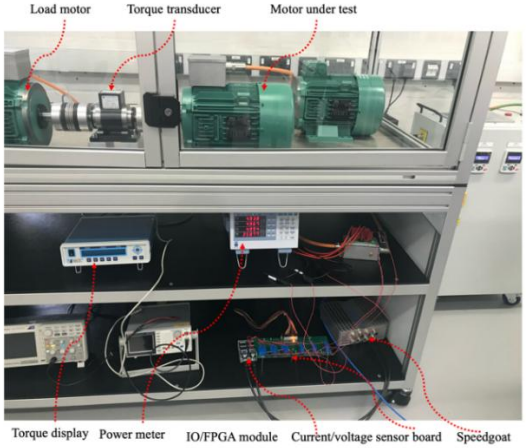

Fig. 3. Experimental test rig and instrumentation system.

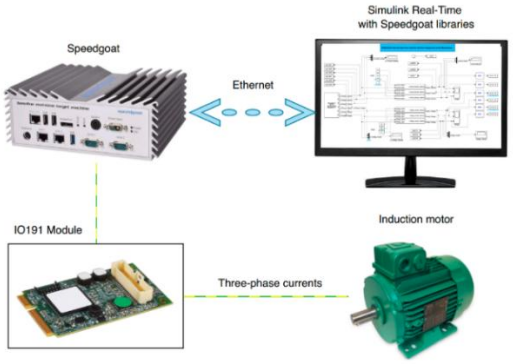

Fig. 4. Connection diagram between the Speedgoat, computer and induction motor.

The rated parameters of the main motor are given in Table I.

TABLE I
RATED PARAMETERS OF MOTOR UNDER TEST

Nominal power	4 kW
Nominal rated voltage	400 V
Nominal rated current	6.8 A
Nominal speed	1456 rpm
Nominal torque	13.3 Nm
Frequency	50 Hz
Pole pairs	2
Stator slots	36
Rotor slots	32
Stator winding connection	Y

IV. Mathematical Preliminaries

A. Kernel function and preimage problem

Let $\mathbf{x} \in R^d$ represent a data point in the original input space. A kernel function $\kappa(x, y)$ implicitly defines a mapping $\phi(x)$ that transform **x** from the input space to a feature space F. The kernel function computes the inner product in the feature space without explicitly computing $\phi(x)$ as [29]

$$\kappa(x, y) = \langle \phi(x), \phi(y) \rangle_F \tag{1}$$

Given a point z in the feature space, the preimage problem involves finding a point $\hat{\mathbf{x}} \in R^d$ in the input space such that $\phi(\hat{\mathbf{x}}) \approx z$. Mathematically, preimage problem can be formulated as follows [30]:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x} \in R^d} \left\| \phi(\mathbf{x}) - z \right\|_F^2 \tag{2}$$

This is the process of finding the closest approximation in the input space that corresponds to a point in the feature space.

B. NCP and PRNN dynamics

Let $\mathbf{x} \in R^n$ be a n-dimensional vector, $f : R^n \to R$ be the continues cost function, $\mathbf{g} : R^n \to R^m$ be a vector-valued continues constrains function, and $\mathbf{D} \in R^{1 \times n}$ be the coefficient vector. A general CNP can be described as follows:

$$\begin{cases} \min \quad f(\mathbf{x}) \\ \text{s.t. } \mathbf{g}(\mathbf{x}) \le 0,\ \mathbf{D}^T \mathbf{x} = 0 \end{cases} \tag{3}$$

An optimal solution $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*)$ to NCP is a feasible point that minimise the cost function $f(\mathbf{x}^*)$ and satisfies $\mathbf{g}(\mathbf{x}) < 0$ and $\mathbf{D}^T \mathbf{x} = 0$. The Karush-Kuhn-Tucker (KKT) optimal conditions for the optimal point is given by [25]

$$\begin{cases} \mathbf{y} \ge 0,\ \mathbf{g}(\mathbf{x}) \le 0,\ \mathbf{D}^T \mathbf{x} = 0 \\ \nabla f(\mathbf{x}) + \nabla \mathbf{g}^T(\mathbf{x}) \mathbf{y} - \mathbf{D}^T \mathbf{z} = 0,\ \mathbf{y}^T \mathbf{g}(\mathbf{x}) = 0 \end{cases} \tag{4}$$

where **y** and **z** are the Lagrange multipliers. For a closed set $\Omega = \left\{ \mathbf{x} \in R^n \,\middle|\, \mathbf{g}(\mathbf{x}) < 0,\ \mathbf{D}^T \mathbf{x} = 0 \right\}$, $\mathbf{x}^*$ is the optimal solution of (4) if and only if $Pr_\Omega(\boldsymbol{\lambda} - \alpha \mathbf{F}(\boldsymbol{\lambda})) = \boldsymbol{\lambda}$, where $\alpha > 0$ is a positive scalar and $\Pr_\Omega(.)$ is the projection operator, which is given $Pr_\Omega(x) = arg \min_{v \in \Omega} |x - v|$. Therefore, the optimal $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*)$ is the solution of the following conditions:

$$\begin{cases} \mathbf{y} = Pr(\mathbf{y} + \mathbf{g}(\mathbf{x})), \mathbf{D}^T \mathbf{x} = 0 \\ \nabla f(\mathbf{x}) + \nabla \mathbf{g}^T(\mathbf{x}) \mathbf{y} - \mathbf{D}^T \mathbf{z} = 0 \end{cases} \tag{5}$$

where $Pr(\delta) = \max\{0, \delta\}$. Thus, the dynamic model of PRNN to meet conditions in (6) is given below:

$$\frac{d}{dt} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \varepsilon \begin{pmatrix} -\nabla f(\mathbf{x}) - \nabla \mathbf{g}^T(\mathbf{x}) \mathbf{y} + \mathbf{D}^T \mathbf{z} \\ -\mathbf{y} + \Pr(\mathbf{y} + \mathbf{g}(\mathbf{x})) \\ -\mathbf{D}\mathbf{x} \end{pmatrix} \tag{6}$$

where $\varepsilon > 0$ is the convergence rate of PRNN.

V. Data preprocessing

Various faults in the motor can appear in different ways in the signals it generates. Some faults may cause abrupt changes in the amplitude of the signal, which appears as sharp spikes or rapid fluctuations when observed over time. Other faults may disrupt the different frequencies of the signal, which can be seen when examining the signal spectra. Investigating time and frequency characteristics of the current signals helps to identify a wider range of potential faults within the motor. The data are collected at various speeds and loads, with an acquisition rate of 1 millisecond. For each scenario, a 20-second segment of data was analysed. DWT, PSD and cepstrum were then applied to measured signals. For a real-time function $x(t)$, DWT is given by [31]

$$DWT(m, n) = \frac{1}{\sqrt[m]{a_0}} \sum_k x(k) \psi \left(\frac{k - n b_0 a_0^m}{a_0^m} \right) \tag{7}$$

where a_0 and b_0 are the scale and translation factors, respectively, *m* is the decomposition level, *n* is the translation step ψ is the wavelet function and $t=kT$, (k=0,1,…) where T is the sampling time. The scale factor a_0 is chosen as 1. The translation factor is the time shifts of the wavelet function at each scale. It's expressed as $b_0 = \{0, 1, ..., N_x / 2 - 1\}$, where N_x is the length of the input signal. The translation step determines the spacing between the wavelet coefficients in the time domain. It's defined as $n = N_x / N_b$ where N_b is the length of translation factor. Low-pass and high-pass wavelet decomposition filters with the biorthogonal spline wavelet are used to obtain the approximate and detail coefficients.

A single level of decomposition is used as it provides a quick and focused analysis by breaking the signal into one approximation and one detail component. This is ideal for real-time processing and initial analysis because it maintains the temporal resolution without requiring excessive computation. The Daubechies 2 (db2) wavelet is chosen because it effectively analyses signals containing both smooth and abrupt changes. The db2 wavelet makes balance between time and frequency localisation. The complexity of the DWT depends on the length of the signal *N*, and the number of decomposition levels *L*. Therefore, for a single-level DWT, the complexity is $O(N)$.

Welch spectrum estimation is used to calculate the PSD. Welch's method was chosen due to its ability to reduce noise and spectral leakage by averaging periodograms from overlapping segments, which improves the overall signal-to-noise ratio. This method has been successfully applied in other studies for similar applications [32]. Welch estimation involves dividing the current signals into overlapping segments, calculating the periodogram for each segment, and averaging to give an estimate of the PSD [33]:

$$PSD(w) = \frac{1}{n_s} \sum_{j=1}^{n_s} \left(\frac{1}{MP} \left| \sum_{k=1}^{M} v(k) x_j(k) e^{-iwk} \right|^2 \right) \tag{8}$$

where n_s is the number of samples, *M* is the number of segments, $v(k)$ is the Hamming window as

$$v(k) = 0.54 - 0.46cos\left(2\pi \frac{k}{N}\right), 0 \le k \le N \qquad (9)$$

where N+1 is the window length, and P denotes the power of the Hamming window as

$$P = \frac{1}{M}\sum_{k=1}^{M}\left|v(k)\right|^2 \qquad (10)$$

The number of segments is chosen as $n_s/4.5$ which is equivalent to dividing the signal into the longest possible segments to obtain as close to but not exceed 8 segments with 50% overlap. The segment length and DFT length are 71 and 256 samples, respectively. The complexity of Welch's method with K segments and window size of M is $O\left(K.M.\log\left(M\right)\right)$.

To calculate the cepstrum of a signal $x(t)$, the initial step involves taking the Fourier transform of $x(t)$. Subsequently, the magnitude of the Fourier transform is calculated, the logarithm of that magnitude is taken, and finally, the logarithm is subjected to the inverse Fourier transform [34].

$$C_x(n) = \frac{1}{N}\sum_{k=0}^{N-1} log\left|\sum_{j=0}^{N-1} x(n) e^{\frac{-i2\pi nj}{N}}\right| e^{\frac{i2\pi nk}{N}} \qquad (11)$$

For a signal of length N, the complexity of cepstrum involves two FFT operations and one logarithmic operation, which is $O(N)$. The overall complexity for cepstrum calculation per signal is $O\left(N.\log\left(N\right)\right)$. Fig. 5 shows an instance of the three-phase currents and their wavelet detail coefficients, PSD, and cepstrum at one of the operational modes of the faulty motor.

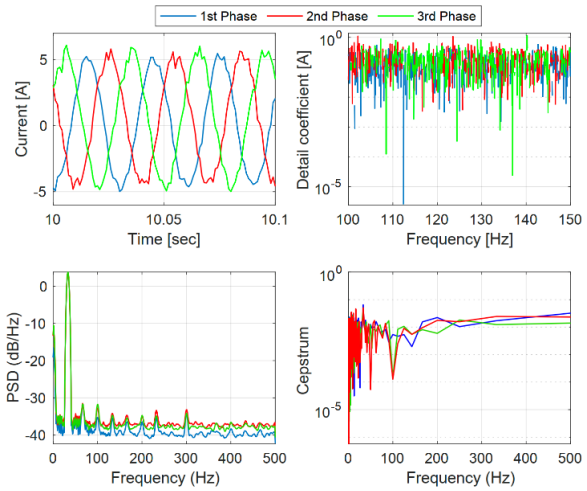

Fig. 5. Three-phase stator currents of the faulty motor, their DWT, PSD and cepstrum.

VI. Feature Extraction

Feature extraction transforms raw data into valuable information, reduces computational requirements, and isolates meaningful patterns by retaining essential information and discarding unnecessary details [35]. In this study, mean, standard deviation, kurtosis, skewness, crest factor, shape factor, impulse factor and clearance factor are calculated, and applied to electrical currents and other signals such as PSD and cepstrum. Table II illustrates the features with their mathematical formulations. Fig. 6 shows an instant of healthy and faulty features which are calculated based on the cepstrum of the three-phase currents.

TABLE II
Statistical Features with their Definition [36]

Feature	Description	Definition
Mean	The average value of a signal over time.	$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$
Standard deviation	A signal's dispersion in terms of variance around the mean.	$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N}}$
Kurtosis	A measure of distribution peakedness.	$k_x = \frac{\sum_{i=1}^{N}(x_i - \mu)^4}{\sigma^4}$
Skewness	A measure of distribution asymmetry.	$s_x = \frac{\sum_{i=1}^{N}(x_i - \mu)^3}{\sigma^3}$
Crest factor	The peak-to-average ratio of a signal.	$crf = \frac{\max_i x_i}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert^2}}$
Shape factor	A measure of the shape of a signal (RMS/mean).	$sf = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert^2}}{\mu}$
Impulse factor	A measure of the sharpness of a signal.	$If = \frac{\max_i x_i}{\mu}$
Clearance factor	A measure of the alignment of motor components.	$clf = \frac{\max_i x_i}{\left[\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert\right]^2}$

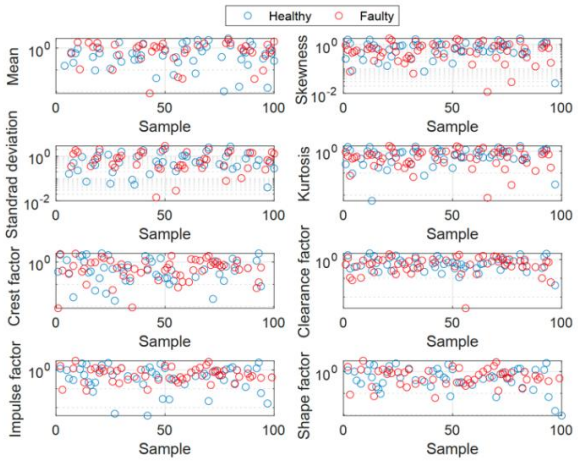

Fig. 6. Healthy and faulty features related to the cepstrum of the three-phase currents.

VII. Feature Selection using KPCA

The goal of feature selection is to identify the dominant features from a data set. This improves the performance of predictive models and simplifies the training process. In this paper, KPCA is used to determine the dominant features. In KPCA, the data is transformed into a high-dimensional feature space using a kernel function, and then PCA is computed in this space, to extract non-linear principal components [37]. Let $\hat{\mathbf{X}}_b$ be the estimated preimage at iteration b obtained using the kernel matrix κ_{wb} and the data matrix $\mathbf{X}$ and $\hat{\mathbf{X}}_b^f$ be the

preimages at iteration b when a specific feature f, the importance of feature f, denoted by imp_f is calculated as [38]

$$imp_f = \sum_{i=1}^{n_b} \frac{\left\| \hat{\mathbf{X}}_b - \hat{\mathbf{X}}_b^f \right\|_F}{n_b} \tag{12}$$

where F denotes the Frobenius norm and n_b is the number of iterations. If Frobenius norm of ($\hat{\mathbf{X}}_b - \hat{\mathbf{X}}_b^f$) is small, there is not much difference in the preimage estimated with and without feature f. Thus, imp_f becomes small and feature f is not important. Similarly, if Frobenius norm of ($\hat{\mathbf{X}}_b - \hat{\mathbf{X}}_b^f$) is high, imp_f is high and feature f becomes important.

In terms of the computational complexity, the kernel matrix computation, which involves calculating the Gaussian kernel for all pairs of data points, has a complexity of $O\left(N^2.d\right)$, where N is the number of data points and d is the number of features. Executing the kernel matrix requires $O\left(N^3\right)$) due to the matrix operations involved. The eigenvalue decomposition, which computes the eigenvalues and eigenvectors of the kernel matrix, also has a complexity of $O\left(N^3\right)$. Finally, projecting the data onto principal components and ranking the features involves matrix multiplications and sorting operations, with the matrix multiplication being $O\left(N^3\right)$. Therefore, the overall complexity of the KPCA algorithm is dominated $O\left(N^3\right)$.

To rank the features, a matrix of 40 features is initially provided, with each feature having 200 values composed of both healthy and faulty data of one single phase of stator currents. The first eight features is related to the time domain signals, while the remaining four groups are related to the DWT approximation and detail coefficients, PSD and cepstrum. Fig. 7 shows the feature ranks for the first phases of the IM. The clearance factor of the electric currents, the clearance factor of PSD and the mean of the DWT approximation coefficients are the three most prominent features. The proposed method's accuracy will be assessed using various sets of features based on their rank determined by the KPCA algorithm.

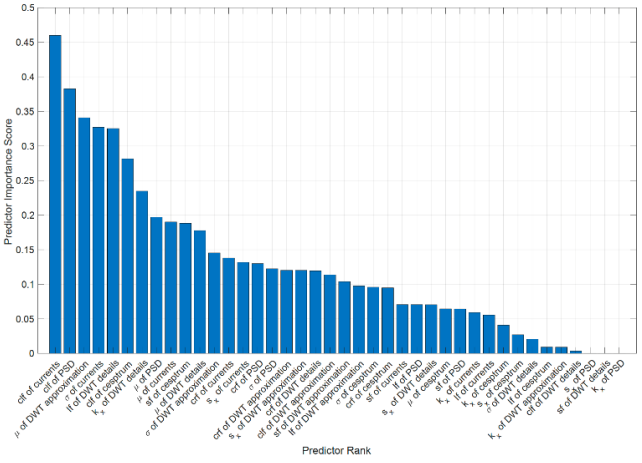

Fig. 7. Ranking of features for the first phase of the induction motor using the KPCA algorithm.

VIII. PRNN-SVM-BASED FAULT DETECTION

SVM is a tool that sorts data into different groups. It does this by providing a hyperplane to separate the groups. For a set of training data $\{(x_1,d_1),\ldots,(x_n,d_n)\}$ where x_i represents the ith input vector and d_i denotes its corresponding class label, the objective of SVM is to identify a hyperplane $(\mathbf{w}, h)$ that satisfies the following optimisation problem [29]:

$$\begin{aligned} &\min_{\mathbf{w}^i} \mathbf{w}^T(\mathbf{w}) \\ &s.t. \quad d_i\left(\boldsymbol{\psi}(\boldsymbol{x_i})\mathbf{w}+h\right) \geq 1,\ i=1,2,...,n \end{aligned} \tag{13}$$

where $\mathbf{w}$ is the weight vector perpendicular to the hyperplane, h is the bias term and $\boldsymbol{\Psi}(\mathbf{x}_i)$ is the kernel function. Consider the dual of (13) using the Lagrange function as

$$\mathrm{L}\left(\mathbf{w},h,\lambda^i\right) = \frac{1}{2}(\mathbf{w})^T(\mathbf{w}) - \sum_{i=1}^{n}\left(\lambda^i\left(d_i\left(\boldsymbol{\psi}(\boldsymbol{x_i})\mathbf{w}+h\right)-1\right)\right) \tag{14}$$

where $\lambda^i \geq 0$ is the Lagrange multiplier. Derivative of (14) to $\mathbf{w}$ and h returns:

$$\begin{cases} \dfrac{\partial \mathrm{L}\left(\mathbf{w},h,\lambda^i\right)}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n}\lambda^i d_i\left(\boldsymbol{\psi}(\boldsymbol{x_i})\right)^T = 0 \\ \dfrac{\partial \boldsymbol{L}\left(\mathbf{w},\boldsymbol{h},\lambda^i\right)}{\partial h^i} = \sum_{i=1}^{n}\lambda^i d_i = 0, i=1,2,...,n \end{cases} \tag{15}$$

Substituting (15) to (14) yields:

$$\mathrm{L}\left(\mathbf{w},h,\lambda^i\right) = \sum_{i=1}^{n}\lambda^i - \frac{1}{2}\sum_{i,j=1}^{n} d_i d_j \lambda^i \lambda^j \left(\boldsymbol{\psi}(\boldsymbol{x_i})\right)\left(\boldsymbol{\psi}(\boldsymbol{x_i})\right)^T \tag{16}$$

Assuming λ^{i*} is the optimal solution of (16), the CNP of SVM can be expressed as follows [29]:

$$\begin{aligned} &\max_{\lambda^i} \sum_{i=1}^{n}\lambda^i - \frac{1}{2}\sum_{i,j=1}^{n} d_i d_j \lambda^i \lambda^j \left(\boldsymbol{\psi}(\boldsymbol{x_i})\right)\left(\boldsymbol{\psi}(\boldsymbol{x_i})\right)^T \\ &s.t. \sum_{i=1}^{n} d_i \lambda^i = 0, \quad \lambda^i \geq 0, i, 1, 2, ..., n \end{aligned} \tag{17}$$

The optimal weight vector and bias term are as follows:

$$\begin{aligned} \mathbf{w}^* &= \sum_{i=1}^{n} d_i \lambda^{i*}\left(\boldsymbol{\psi}(\boldsymbol{x_i})\right)^T \\ h^* &= -\frac{\max_{z_i=1}\left(\boldsymbol{\psi}(\boldsymbol{x_i})\mathbf{w}^*\right) + \min_{z_i=-1}\left(\boldsymbol{\psi}(\boldsymbol{x_i})\mathbf{w}^*\right)}{2} \end{aligned} \tag{18}$$

Consider the cost function $f(\boldsymbol{\lambda})$, inequality constraints function $\mathbf{g}(\boldsymbol{\lambda})$ and the equality function $h(\boldsymbol{\lambda})$ as follows:

$$\begin{aligned} f(\boldsymbol{\lambda}) &= \frac{1}{2}\sum_{i=1}^{n} d_i d_j \lambda^i \lambda^j \left(\boldsymbol{\psi}(\boldsymbol{x_i})\right)\left(\boldsymbol{\psi}(\boldsymbol{x_i})\right)^T - \sum\nolimits_{i=1}^{n}\lambda^i \\ h(\boldsymbol{\lambda}) &= \sum_{i=1}^{n} d_i \lambda^i = 0, \quad g(\boldsymbol{\lambda}) = -\lambda^i \leq 0, i, 1, 2, ..., n \end{aligned} \tag{19}$$

The PRNN dynamics in (6) is then used to obtain the optimal solution of (19). The optimiser in (6) is simple, asymptotically stable and easy to implement. Using (6) and (19), the optimal $\boldsymbol{\lambda}^*$ is first obtained. The optimal weight vector $\mathbf{w}^*$ and bias h^*

are then calculated using (18). Finally, the decision boundary is obtained as $f(\mathbf{x}) = \boldsymbol{\psi}(\mathbf{x})\mathbf{w}^* + h^*$. In the appendix, the convergency of the PRNN dynamics is demonstrated.

Fig. 8 illustrates the block diagram of the PRNN. The input data undergoes initial processing by kernel functions that form the feedforward matrix, denoted by **F**. The PRNN's dynamic involves three distinct recurrent paths designed to compute optimal Lagrangian multipliers in real-time. The projection operator ensures the solution remains within the feasible space.

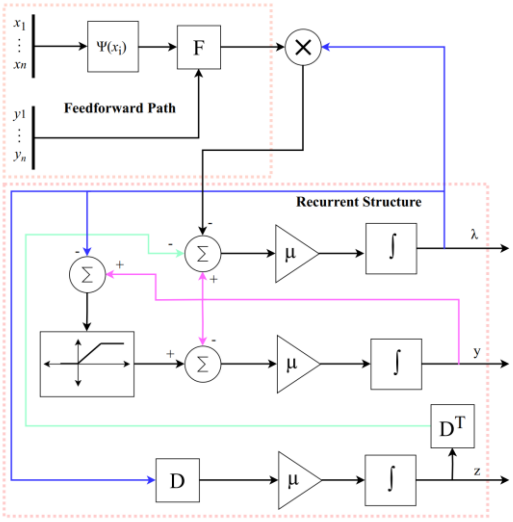

Fig. 8. Block diagram illustrating the feedforward and recurrent structure of the PRNN.

Algorithm 1 outlines the framework of the proposed fault detection algorithm. First, an initial dataset with length N_s is collected. The data is divided into N_d segments to extract key information using DWT, PSD, and cepstrum. KPCA selects the dominant features. The PRNN over N_p iterations, derives optimal Lagrange multipliers, optimal weight vectors and offsets, critical for SVM classification. As a numerical optimiser, rather than a machine learning algorithm, the PRNN dynamically converges to its equilibrium point, which is the optimal solution to the optimisation problem.

The PRNN-based SVM predictive model can be easily deployed on a microprocessor. The inputs for the model are statistical features, which can be computed on the processor. The key consideration is to appropriately assign a time window for segmenting and storing the real-time data the processor's memory. The processor's frequency must be sufficiently high to execute commands efficiently for extracting these features.

In terms of computational complexity, computing the kernel matrix requires $O(n^2.d)$, where d is the dimension of the input space and n is the number of samples. Each iteration of PRNN involves the gradient computations $O(n^2)$ due to the kernel matrix, matrix-vector multiplications $O(n^2)$, and projection operation $O(n)$. For k iterations, the total complexity is $O(k.n^2)$. Final hyperplane computation is $O(n^2)$.

Algorithm 1 PRNN-based SVM fault detection.

Initial dataset ← healthy and faulty three-phase currents

Prepare the time-frequency dataset (initial dataset)

while $i \in N_d$

for $j \in N_s$

$$DWT(m,n) \leftarrow \frac{1}{\sqrt[m]{a_0}} \sum_k x(k)\psi\left(\frac{k - nb_0a_0^m}{a_0^m}\right)$$

$$PSD(w) \leftarrow \frac{1}{n_s}\sum_{j=1}^{n_s}\left(\frac{1}{MP}\left|\sum_{k=1}^{M} y(k)x_j(k)e^{-iwk}\right|^2\right)$$

$$C_x(n) \leftarrow \frac{1}{N}\sum_{k=0}^{N-1} log\left|\sum_{j=0}^{N-1} x(n)e^{\frac{-i2\pi nj}{N}}\right| e^{\frac{i2\pi nk}{N}}$$

end

end

Calculate the features

for $x_i \in \{DWT(m,n), PSD(w), C_x(n)\}$

$$\mu \leftarrow \frac{1}{N}\sum_{i=1}^{N} x_i\,,\ \sigma \leftarrow \sqrt{\frac{\sum_{i=1}^{N}(x_{i-} - \mu)^2}{N}}\,,$$

$$k_x \leftarrow \frac{\sum_{i=1}^{N}(x_{i-} - \mu)^4}{\sigma^4}\,,\ s_x \leftarrow \frac{\sum_{i=1}^{N}(x_{i-} - \mu)^3}{\sigma^3}\,,\ If \leftarrow \frac{\max_i x_i}{\mu}\,,$$

$$cf \leftarrow \frac{\max_i x_i}{\left[\frac{1}{N}\sum_{i=1}^{N}|x_i|\right]^2}\,, crf \leftarrow \frac{\max_i x_i}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}|x_i|^2}}\,,\ sf \leftarrow \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}|x_i|^2}}{\mu}$$

end

Select the dominant features

for $k \in N_f$

$$imp_f = \sum_{i=1}^{n_b}\frac{\left\|\hat{\mathbf{X}}_b - \hat{\mathbf{X}}_b^f\right\|_F}{n_b}$$

end

Obtain the optimal Lagrange multipliers

for $r \in N_p$

$$\begin{pmatrix}\boldsymbol{\lambda}^* \\ \mathbf{y}^* \\ \mathbf{z}^*\end{pmatrix} \leftarrow \frac{d}{dt}\begin{pmatrix}\boldsymbol{\lambda} \\ \mathbf{y} \\ \mathbf{z}\end{pmatrix} = \mu\begin{pmatrix}-\nabla f(\boldsymbol{\lambda}) - \nabla \boldsymbol{g}^T(\boldsymbol{\lambda})\mathbf{y} + \mathbf{D}^T\mathbf{z} \\ -\mathbf{y} + \Pr(\mathbf{y} + \mathbf{g}(\boldsymbol{\lambda})) \\ -\mathbf{D}\boldsymbol{\lambda}\end{pmatrix}$$

end

Calculate the optimal weight and offset for SVM

$$\mathbf{w}^* \leftarrow \sum_{i=1}^{n} d_i\lambda^{i*}\left(\boldsymbol{\psi}(\boldsymbol{x}_i)\right)^T$$

$$h^* \leftarrow -\frac{\max_{z_i=1}\left(\boldsymbol{\psi}(\boldsymbol{x}_i)\mathbf{w}^*\right) + \min_{z_i=-1}\left(\boldsymbol{\psi}(\boldsymbol{x}_i)\mathbf{w}^*\right)}{2}$$

IX. Results and Discussion

To assess the effectiveness of the proposed fault detection method, a dataset containing both normal and faulty scenarios was used. Real-time three-phase currents were measured under different speeds and loads, ranging from 1000-1500 rpm and 0-90% load conditions with the step changes of 50 rpm and 10% for speed and load, respectively. Finally, the experimental tests are performed under 100 operating modes corresponding to 10

different speeds and 10 load levels. A 20-second time window for each data set, leading to 20,000 samples of one single phase of stator current collected for each operating mode.

Each dataset was then divided into five smaller sets with 5,000 samples each. Considering all operating modes, 100 datasets were used for the healthy motor and 100 datasets for the faulty motor. To obtain the feature set, the DWT approximation and detail coefficients, PSD, and cepstrum of each initial dataset were first computed. Then, 40 features were obtained by applying eight different statistical measures (mean, standard deviation, kurtosis, skewness, crest factor, impulse factor, shape factor, and clearance factor) to 5 different kinds of datasets. There were 200 data points for each feature, including 100 healthy and 100 faulty data points. The total duration of the experimental test for each motor speed was 5 minutes.

To train the proposed predictive model for bearing fault detection, 75% of the feature set was used. The dataset used to train and test the proposed model consists of 200 instances, with 140 used for training and 60 for testing. The KPCA feature selection algorithm identified the clearance factor of the electric currents and the clearance factor of PSD signals as the two most important features. To verify this, these features and their corresponding class labels were applied to the proposed PRNN-based SVM algorithm and were used to analyse the accuracy of classifying data into healthy and faulty categories. The results presented in the paper are based on a single-phase stator current to avoid redundancy and overfitting problems.

According to Fig. 9, the classification accuracy is 93.33%. The results highlight the effectiveness of the clearance factor in discriminating between healthy and faulty data. The clearance factor is a statistical feature that characterises the shape of a signal. It is calculated as the ratio of the peak value of a signal to the square mean of the square roots of the absolute amplitudes of the signal. The clearance factor highlights the peaks in the transformed domain, which are often associated with fault-induced resonances or irregularities. A higher clearance factor indicates a more impulsive signal, which indicates a defective outer ring in the induction machine bearing. In addition, a higher clearance factor for the PSD reflects a more significant presence of high frequency components, which are often associated with bearing faults.

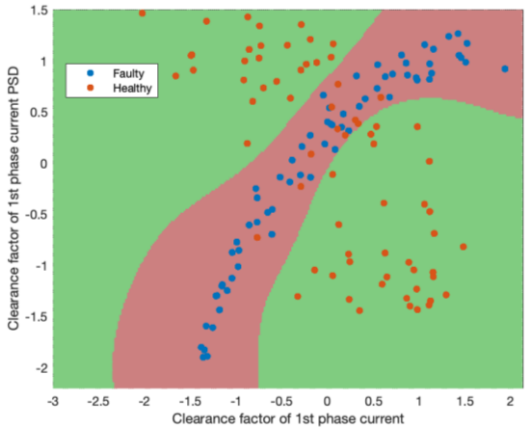

Fig. 9. Bearing fault classification in the training phase using the proposed PRNN-based SVM algorithm.

A key element of the proposed method is the nonlinear kernel function used for both feature selection and fault detection. To verify this aspect, the proposed algorithm was also simulated using linear and polynomial kernel functions as the two widely used functions reported in the literature [39-40]. Fig. 10(a) depicts that the best classification accuracy using linear function is 55% in the training phase, which indicates a poor performance compared to the RBF function. Fig. 10(b) indicates the relatively lower performance of the proposed method with polynomial kernel function compared to the RBF kernel function. The best accuracy in this case is 82.85%. These results confirm the importance of utilising a nonlinear kernel function to classify data in a nonlinear, complex space.

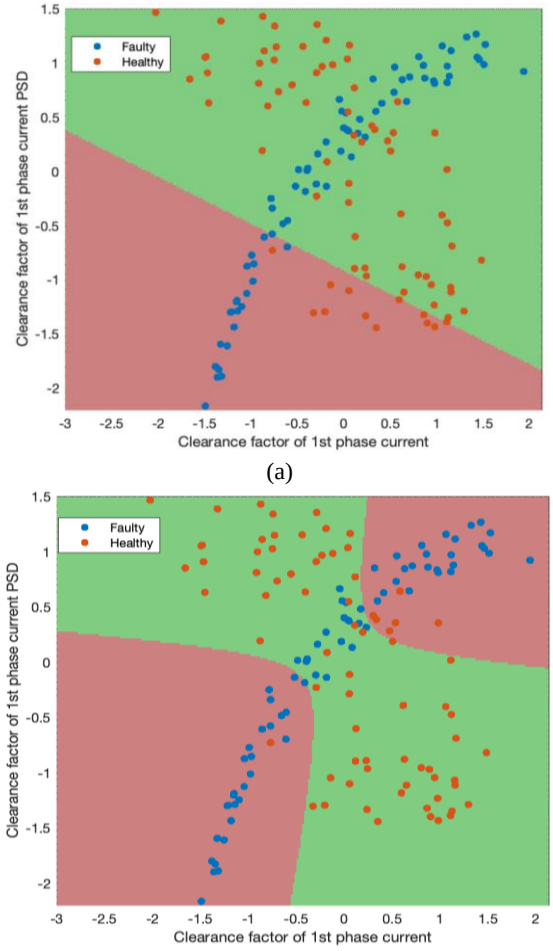

Fig. 10. Bearing fault classification using the proposed algorithm. (a) Linear kernel, (b) Polynomial function.

The next aim is to assess the performance of the conventional SVM algorithm in classifying healthy and faulty data. According to Fig. 11, the accuracy of the conventional SVM is 86.4% in the training phase. In comparison, the proposed method, which utilises the PRNN as a robust and stable optimiser, achieves an accuracy which is approximately 10% higher than the conventional SVM method in the training phase. During the test phase, depicted in Fig. 12, the accuracy of the proposed method with RBF kernel function is 95%. According to Fig. 13, the accuracy of the conventional SVM in detecting bearing faults is 90% in the testing phase.

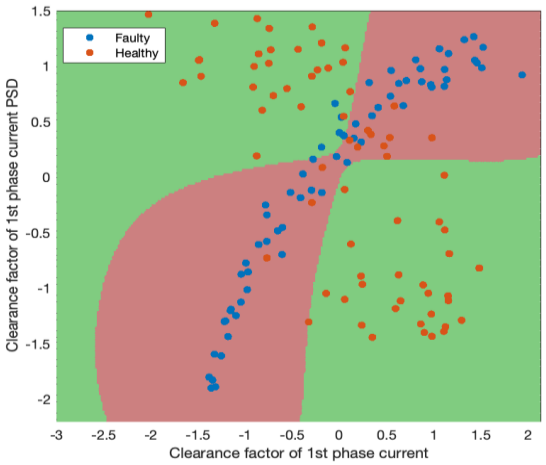

Fig. 11. Bearing fault classification in the training phase using the conventional SVM algorithm.

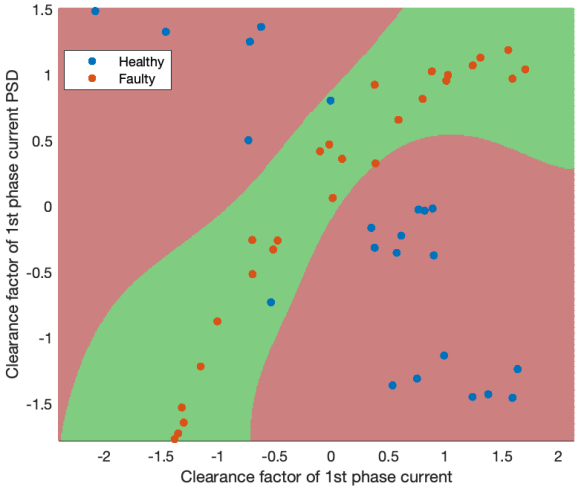

Fig. 12. Bearing fault classification in test phase using the proposed PRNN-based SVM algorithm.

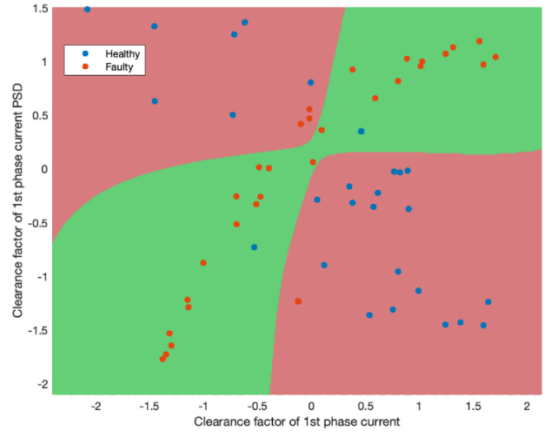

Fig. 13. Bearing fault classification in test phase using the conventional SVM algorithm, utilising only the clearance factor of current and PSD signals as input features.

This superior accuracy in both the training and test phases indicates the acceptable performance of the proposed algorithm in detecting bearing faults using only stator current data. In comparison, the proposed method achieves an accuracy which is approximately 5% higher than the conventional SVM method in the testing phase. Additionally, the proposed method benefits from the PRNN's simple structure, which can be easily implemented using low-cost electric components. The optimiser also has a stable convergence to its equilibrium points, contributing to its robustness.

The performance of the proposed method is also evaluated in comparison to other methods using confusion matrices, precision, recall, and F1 score metrics. Figures 14 and 15 illustrates the confusion matrices obtained by the proposed method with RBF, polynomial, and linear kernel functions, as well as the conventional SVM. The numerical results for both training and test phases are presented in Table III.

The results demonstrate a clear distinction in the performance of different kernel functions for the classification task. The proposed method with the RBF kernel achieved the highest metrics in both training and test sets, with a near-perfect F1 score 0.933 for training and 0.966 for testing. It indicates that the proposed method has a robust model generalisation and minimal overfitting. In comparison, the conventional SVM with lower performance has F1 score of 0.896 in the test set. The results imply a reliable generalisation but less adaptability to intricate decision boundaries. The polynomial kernel has moderate performance, with a decrease in F1 score 0.815 from training to 0.820 in testing phase. The linear kernel performed the worst, with accuracy 0.549 and F1 score 0.606.

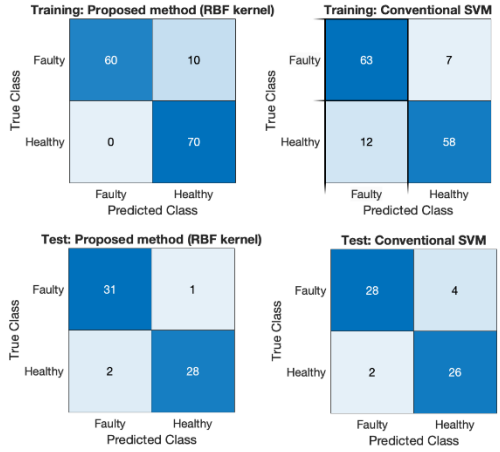

Fig. 14. Confusion matrix for the proposed pethood and conventional SVM during training and test phases.

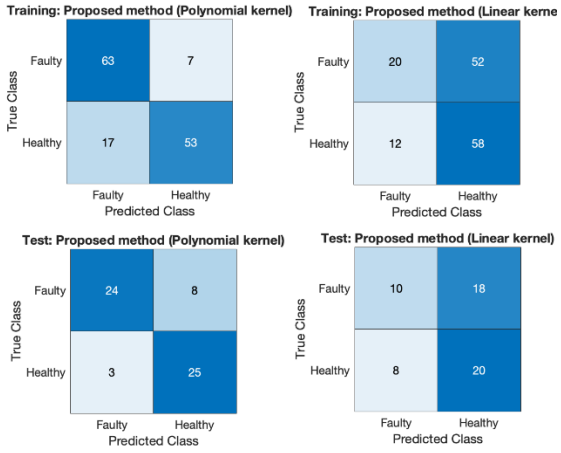

Fig. 15. Confusion matrix for the proposed pethood with different kernel functions during training and test phases.

TABLE III

PERFORMANCE METRICS FOR THE PROPOSED METHOD WITH DIFFERENT KERNEL FUNCTION, AND CONVENTIONAL SVM

Model	Accuracy	Precision	Recall	F1 Score
Proposed method – RBF-Train	0.9286	0.8570	1	0.9333
Proposed method – Polynomial-Train	0.8286	0.8833	0.7571	0.8154
Proposed method – Linear-Train	0.5493	0.5273	0.8286	0.6444
Conventional SVM - Train	0.8643	0.8923	0.8286	0.8593
Proposed method – RBF-Test	0.9677	1	0.9333	0.9655
Proposed method – Polynomial-Test	0.8167	0.7576	0.8929	0.8197
Proposed method – Linear-Test	0.5357	0.5263	0.7143	0.6061
Conventional SVM - Test	0.9	0.8667	0.9286	0.8966

The performance of the proposed method was also evaluated using different sets of features ranked by their importance applied to linear, polynomial and RBF kernel functions. The process involves considering the first two features with the highest rank. This process continued by incrementally adding the next highest-rank feature to the predictive model until all features were included. The analysis was repeated 50 times and the mean and standard deviation of the accuracy for each feature set were computed. According to Fig. 16, the algorithm using only single-phase electric currents achieved its best performance of approximately 95% accuracy in the test phase when using the top four most important features with the RBF kernel. When using the top three most dominant features, the method achieved an accuracy of approximately 83% with the polynomial kernel and 58% with the linear kernel. the results indicate that as more features are incorporated into the predictive model, its performance gradually decreases mainly due to the added complexity to the analysis. This highlights the importance of identifying the most dominant features for effectively training and testing the predictive model.

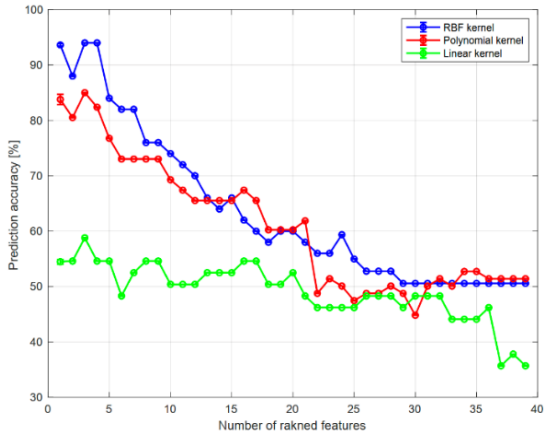

Fig. 16. Comparison of the accuracy obtained from the proposed algorithm for linear, polynomial and RBF kernel functions and different ranked features.

The performance of conventional SVM is investigated under various values of two main parameters: regularisation parameter (C) and Gamma parameter (γ). The parameter C controls the trade-off between achieving a low training error and minimising the norm of the weights. The parameter γ defines how far the influence of a single training example reaches, with low values meaning far and high values meaning close. For each parameter, 50 values between 0.01 and 1 were considered and used to calculate the accuracy of the SVM in the test phase. Fig. 17 depicts the trend of classification accuracy for different C and γ. The results indicate that the same level of performance can be achieved by choosing C values close to 1 and γ values between 0.2 and 1. The best accuracy is 90.6% obtained for C = 0.96 and γ = 0.24.

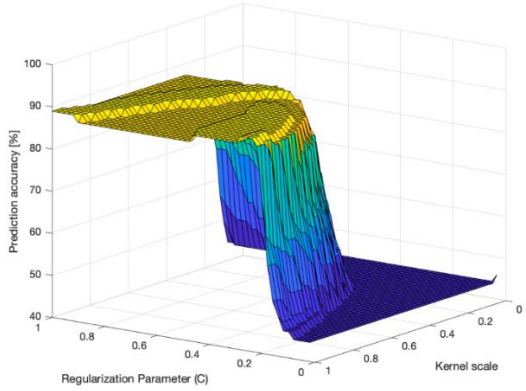

Fig. 17. Accuracy of the conventional SVM under different regularisation and Gamma parameters.

The results of the 10-fold cross-validation for the proposed method are given in Fig. 18. They show an impressive model performance with an average training accuracy of 94.28% and an average testing accuracy of 94.00%. The consistency in the training accuracy values, which ranges from 93.33% to 95.56% indicates that the model effectively captures the underlying patterns in the training data. The testing accuracy, which ranges from 90% to 100%, further supports the model's strong generalisation capabilities.

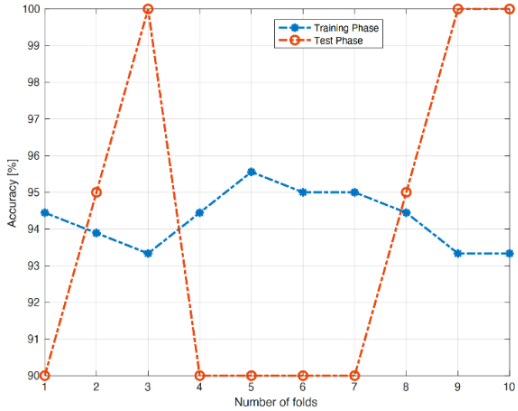

Fig. 18. Accuracy of the proposed algorithm during training and testing phases in the 10-fold cross-validation process.

Finally, the performance of the proposed method is compared to the decision tree algorithm and SVM-based fault detection

proposed in [17]. Figure 19 shows the ROC curves during the 10-fold cross validation. The proposed method with an RBF kernel achieves the highest AUC of 0.9803, which reinforces its effectiveness in classifying the data with minimal false positives. The polynomial kernel also performs well with an AUC of 0.8924. The linear kernel achieves a significantly lower AUC of 0.5055, which confirms its limited ability to effectively separate classes. The decision tree shows a modest AUC of 0.8585, while the SVM from [17] and the conventional SVM both maintain high AUCs of 0.9685 and 0.9508 respectively. The mean accuracy of the method in [17] and decision tree algorithm in the test phase are 90.5% and 75.5%, respectively.

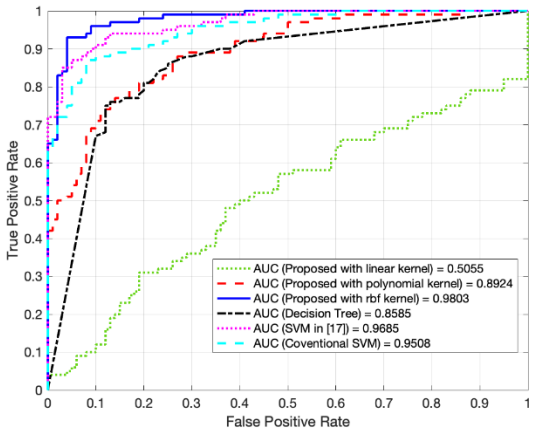

Fig. 19. ROC curve of proposed method with different kernel functions, conventional SVM, decision tree, and SVM in [17].

The box plot results in Figure 20 illustrate the variation in test accuracies over 10-fold cross-validation. The proposed method with the linear kernel shows a relatively low and consistent accuracy with low variance, which reflects its limitations in capturing the complexity of the data. In contrast, RBF kernel methods show much higher accuracies. The proposed method with the RBF kernel function has the least variance and the highest median accuracy close to 100%. The Decision Tree method has a higher variability and a lower median accuracy, which means that it may be more sensitive to data folding. The SVM methods in [17] and conventional SVM perform consistently well, but with slightly lower median accuracies compared to the proposed method.

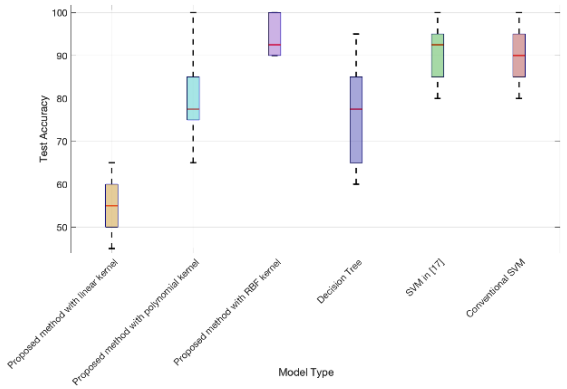

Fig. 20. Accuracies of proposed method with different kernel functions, conventional SVM, decision tree, and SVM in [17].

The proposed approach leverages the strengths of both SVM and PRNN to find the optimal hyperplane. The use of SVM enables the model to distinguish complex and nonlinear patterns through well-tuned kernel functions. The method's feature extraction and selection capabilities make it effective in high-dimensional spaces. The integration of PRNN with SVM enhances the model's convergence speed, stability, and simplicity. Consequently, combining SVM with PRNN, along with advanced feature extraction and selection techniques, provides an efficient solution for detecting not only bearing faults but also other mechanical faults.

Although the proposed method has achieved acceptable accuracy, several strategies could enhance it. Cross-correlation and envelope spectrum entropy could extract more information from current signals. using statistical features such as third and fourth moments, D-factor, RMS, and peak-to-peak values could also enrich the feature set. Using grid search in the PRNN-based SVM could fine-tune the hyperparameters more effectively.

The methodology in this paper can be generalised to address compound faults, which might include inner race faults, outer race faults, and ball defects [41]. To extend the algorithm for compound fault detection, several considerations are necessary. Feature extraction becomes more complex with compound faults due to the potential for overlapping or mixed signal patterns. Techniques like power spectral density, and Hilbert transforms can help capture the interactions between different fault components. Multiscale feature extraction methods, such as empirical mode decomposition, can be used to isolate distinct fault symptoms across various frequency ranges. The model architecture must adapt to the complexity of compound faults. This involves shifting from binary classification to multiclass.

X. Conclusion

This paper presented a machine learning framework that combines SVM with a computational optimisation tool, PRNN, for identifying bearing faults in induction motors. The process involved the acquisition of stator currents, which were then processed by DWT, PSD and cepstrum measures to produce time-frequency representations. Statistical features were derived from these representations to form a feature set. The KPCA algorithm was employed to identify the dominant features. The SVM optimisation task as a non-linear programming challenge was reformulated, and its solution was facilitated by the PRNN. The results demonstrated the remarkable effectiveness of the algorithm to achieve the highest fault detection accuracy. The proposed algorithm features a simple structure, fast convergence, ease of implementation, and asymptotic stability. Compared to the conventional SVM method, decision tree and recent SVM-based fault detection methods, the proposed algorithm exhibited superior performance in terms of accuracy, achieving approximately 5% higher classification accuracy in the training and testing phases, respectively. Detecting fault severity across a range from early fault detection to advanced levels of bearing fault severity, as well as predicting remaining life, presents ongoing challenges. Addressing these aspects will be a focus for future research.

REFERENCES

[1] M. P. Kazmierkowski, *Electric motor drives: Modeling, analysis and control*, R. Krishan, Prentice‐Hall, Upper Saddle River, NJ, 2001, xxviii+ 626 pp. ISBN 0-13-0910147, 2004.

[2] C. Terron-Santiago, J. Martinez-Roman, R. Puche-Panadero, and A. Sapena-Bano, "A review of techniques used for induction machine fault modelling," *Sensors*, vol. 21, no. 14, p. 4855, 2021.

[3] V. Aviña-Corral, J. Rangel-Magdaleno, C. Morales-Perez, and J. Hernandez, "Bearing fault detection in adjustable speed drive-powered induction machine by using motor current signature analysis and goodness-of-fit tests," *IEEE Trans. Industr. Inform.*, vol. 17, no. 12, pp. 8265-8274, 2021.

[4] A. H. Boudinar, N. Benouzza, and A. Bendiabdellah, "Induction motor bearing fault analysis using a root-MUSIC method," *IEEE Trans. Ind. Appl.*, vol. 52, no. 5, pp. 3851-3860, 2016.

[5] M. Afshar, C. Li, and B. Akin, "Real-Time Current-Based Distributed Bearing Faults Detection in Small Cooling Fan Motors," *IEEE Trans. Ind. Appl.*, vol. 60, no. 2, pp. 3188-3199 2023.

[6] Amitabh, P., Bozalakov, D., & De Belie, F. (2024, June). Methodology for Analysis and Quantification of Bearing Fault Signature using MCSA. In *2024 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)* (pp. 1210-1217). IEEE.

[7] M. Yang, N. Chai, Z. Liu, B. Ren, and D. Xu, "Motor speed signature analysis for local bearing fault detection with noise cancellation based on improved drive algorithm," *IEEE Trans. Ind. Electron.*, vol. 67, no. 5, pp. 4172-4182, 2019.

[8] Dubey, R., Rajpoot, V., Chaturvedi, A., Dixit, A., & Maheshwari, S. (2024). Ball-bearing fault classification using comparative analysis of wavelet coefficient based on entropy measurement. *IETE Journal of Research*, *70*(2), 1122-1132.

[9] Laala, W., Guedidi, A., & Guettaf, A. (2023). Bearing faults classification based on wavelet transform and artificial neural network. *International Journal of System Assurance Engineering and Management*, 1-8.

[10] A. Widodo, and B. S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mech. Syst. signal process.*, vol. 21, no. 6, pp. 2560-2574, 2007.

[11] F. B. Abid, S. Zgarni, and A. Braham, A. "Distinct bearing faults detection in induction motor by a hybrid optimized SWPT and aiNet-DAG SVM," *IEEE Trans. Energy Conver.*, vol. 33, no. 4, pp. 1692-1699, 2018.

[12] P. Konar, and P. Chattopadhyay, "Bearing fault detection of induction motor using wavelet and Support Vector Machines (SVMs)," *Appl. Soft Comput.*, vol.11, no. 6, pp. 4203-4211, 2011.

[13] Yang, Y., Liu, H., Han, L., & Gao, P. (2023). A feature extraction method using VMD and improved envelope spectrum entropy for rolling bearing fault diagnosis. *IEEE Sensors Journal*, *23*(4), 3848-3858.

[14] H. Nakamura, and Y. Mizuno, "Diagnosis for slight bearing fault in induction motor based on combination of selective features and machine learning," *Energies*, vol. 15, no. 2, p. 453, 2022.

[15] S. E. Pandarakone, Y. Mizuno, and H. Nakamura, "Distinct fault analysis of induction motor bearing using frequency spectrum determination and support vector machine," *IEEE Trans. Ind. Appl.*, vol. 53, no. 3, pp. 3049-3056, 2016.

[16] Biswas, A., Ray, S., Dey, D., & Munshi, S. (2023). Detection of simultaneous bearing faults fusing cross correlation with multikernel SVM. *IEEE Sensors Journal*, *23*(13), 14418-14427.

[17] B. Brusamarello, J. C. C. da Silva, K. de Morais Sousa, and G. A. Guarneri, "Bearing fault detection in three-phase induction motors using support vector machine and fiber Bragg grating," *IEEE Sens. J.*, vol. 23, no. 5, pp. 4413-4421, 2022.

[18] Mitra, S., & Koley, C. (2024). Real-time robust bearing fault detection using scattergram-driven hybrid CNN-SVM. *Electrical Engineering*, *106*(3), 3615-3625.

[19] C. Y. Lee, T. A. Le, and Y. T. Lin, "A Feature Selection Approach Hybrid Grey Wolf and Heap-Based Optimizer Applied in Bearing Fault Diagnosis," *IEEE Access*, vol. 10, pp. 56691-56705, 2022.

[20] K. Yatsugi, S. E. Pandarakone, Y. Mizuno, and H. Nakamura, "Common Diagnosis Approach to Three-Class Induction Motor Faults Using Stator Current Feature and Support Vector Machine," *IEEE Access*, vol. 11, pp. 24945-24952, 2023.

[21] Kumar, R., & Anand, R. S. (2024). Bearing fault diagnosis using multiple feature selection algorithms with SVM. *Progress in Artificial Intelligence*, *13*(2), 119-133.

[22] J. Chen, W. Hu, D. Cao, Z. Zhang, Z. Chen, and F. Blaabjerg, "A meta-learning method for electric machine bearing fault diagnosis under varying working conditions with limited data," *IEEE Trans. Industr. Inform.*, vol. 19, no. 3, pp. 2552-2564, 2022.

[23] Y. Xia, H. Leung, and J. Wang, J. "A projection neural network and its application to constrained optimization problems," *IEEE Trans. Circuit Syst. I: Fundam. Theory Appl.*, vol. 49, no. 4, pp. 447-458, 2002.

[24] Y. Xia, and J. Wang, J. "A bi-projection neural network for solving constrained quadratic optimization problems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 2, pp. 214-224, 2015.

[25] Y. Xia, J. Wang, and W. Guo, "Two projection neural networks with reduced model complexity for nonlinear programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 2020-2029, 2019.

[26] Huang, B., Liu, Y., Jiang, Y. L., & Wang, J. (2024). Two-timescale projection neural networks in collaborative neurodynamic approaches to global optimization and distributed optimization. *Neural Networks*, *169*, 83-91.

[27] Xia, Y., Wang, J., Lu, Z., & Huang, L. (2022). Two Recurrent Neural Networks With Reduced Model Complexity for Constrained l_1-Norm Optimization. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(9), 6173-6185.

[28] N. K. Hosseini, H. Toshani, and S. Abdi, "A Projection-Based Support Vector Machine Algorithm for Induction Motors' Bearing Fault Detection," in *IEEE 14th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED)* (pp. 186-191). IEEE, 2023.

[29] N. Cristianini, and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[30] Kwok, J. Y., & Tsang, I. H. (2004). The pre-image problem in kernel methods. *IEEE transactions on neural networks*, *15*(6), 1517-1525.

[31] E. C. Lau, and H. W. Ngan, "Detection of motor bearing outer raceway defect by wavelet packet transformed motor current signature analysis," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 10, pp. 2683-2690, 2010.

[32] Wang, H., Sun, W., He, L., & Zhou, J. (2022). Rolling bearing fault diagnosis using multi-sensor data fusion based on 1d-cnn model. *Entropy*, *24*(5), 573.

[33] P. Stoica, and R. L. Moses, *Spectral analysis of signals*. Upper Saddle River, NJ: Pearson Prentice Hall, 2005, vol. 452, pp. 25-26.

[34] Y. R. Hwang, K. K. Jen, and Y. T. Shen, "Application of cepstrum and neural network to bearing fault detection," *J. Mech. Sci. Tech.*, vol. 23, pp. 2730-2737, 2009.

[35] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, J. Saeed, "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction," *J. Appl. Sci. Tech. Trends*, vol. 1, no. 2, pp. 56-70, 2020.

[36] N. Rajapaksha, S. Jayasinghe, H. Enshaei, and N. Jayarathne, "Acoustic analysis based condition monitoring of induction motors: A review," in *IEEE Southern Pow. Elec. Conf.* (SPEC), 2021, pp. 1-10.

[37] Q. Wang, "Kernel principal component analysis and its applications in face recognition and active shape models," *arXiv preprint arXiv:1207.3538*, 2012.

[38] A. Sahu, D. W. Apley, and G. C. Runger, "Feature selection for noisy variation patterns using kernel principal component analysis," *Know-Based Syst.*, vol. 72, pp. 37-47, 2014.

[39] Ö. Karal, " Performance comparison of different kernel functions in SVM for different k value in k-fold cross-validation," in *Innovations in Intelligent Systems and Applications Conference (ASYU)* (pp. 1-5). IEEE, 2020.

[40] S. Xue, L. Zhang, and Z. Zhu, "Design of semi-tensor product-based kernel function for SVM nonlinear classification," *Cont. Theory Tech.*, vol. 20, no. 4, pp. 456-464, 2022.

[41] Suthar, V., Vakharia, V., Patel, V. K., & Shah, M. (2022). Detection of compound faults in ball bearings using multiscale-SinGAN, heat transfer search optimization, and extreme learning machine. *Machines*, *11*(1), 29.

APPENDIX

A. KPCA ALGORITHM

Let $\psi(\mathbf{x})$ represent a nonlinear transformation where each data x_i is then projected to a new point $\psi(x_i)$. Assume that the projected features have zero mean [37]:

$$\frac{1}{N}\sum_{i=1}^{N}\psi(x_i)=0 \tag{A.1}$$

The covariance matrix $\mathbf{C}\in R^{M\times M}$ of the projected features is

$$\mathbf{C}=\frac{1}{N}\sum_{i=1}^{N}\psi(x_i)\psi^T(x_i) \tag{A.2}$$

Considering v_k and λ_k as the eigenvector and eigenvalues of the matrix **C**, it gives

$$\frac{1}{N}\sum_{i=1}^{N}\psi(x_i)\left\{\psi^T(x_i)\mathbf{v}_k\right\}=\lambda_k\mathbf{v}_k \tag{A.3}$$

Consider the eigenvectors as follows:

$$\mathbf{v}_k=\sum_{i=1}^{N}a_{ki}\psi(x_i) \tag{A.4}$$

where a_{ki}, $(i=1,2,...,N)$ are unknown parameters that must be determined properly. Equation (A.3) can be rewritten as [32]

$$\frac{1}{N}\sum_{i=1}^{N}\psi(x_i)\psi^T(x_i)\sum_{j=1}^{N}a_{kj}\psi(x_j)=\lambda_k\sum_{i=1}^{N}a_{ki}\psi(x_i) \tag{A.5}$$

By defining the kernel function as $\kappa(x_i,x_j)=\psi(x_i)\psi^T(x_j)$ and And multiplying it both sides of (A.5), it gets

$$\frac{1}{N}\kappa(x_i,x_j)\sum_{j=1}^{N}a_{kj}\kappa(x_i,x_j)=\lambda_k\sum_{i=1}^{N}a_{ki}\kappa(x_i,x_j) \tag{A.6}$$

In the matrix form, (A.7) can be expressed as

$$\mathbf{K}^2\mathbf{a}_k=\lambda_k N\mathbf{K}\mathbf{a}_k \tag{A.7}$$

where $\mathbf{K}_{i,j}=\kappa(\mathbf{x}_i,\mathbf{x}_j)$, $\mathbf{a}_k=\begin{bmatrix}a_{k1} & a_{k2} & \cdots & a_{kN}\end{bmatrix}^T$. The kernel principal components can be determined using

$$y_k(\mathbf{x})=\psi^T(\mathbf{x})\mathbf{v}_k=\sum_{i=1}^{N}a_{ki}\kappa(\mathbf{x},\mathbf{x}_j) \tag{A.8}$$

For a feature set $F=\{f_1,f_2,...,f_M\}$, a spare random vector ***w***, the number of iterations n_b, the kernel matrix **K**, and the data matrix $\mathbf{x}$, the following Gaussian function with mean zero and variance σ is used [38],

$$\kappa_w(\mathbf{x}_i,\mathbf{x}_j)=\exp\left(\frac{-\left(\mathbf{w}^T\mathbf{x}_i-\mathbf{w}^T\mathbf{x}_j\right)^2}{\sigma}\right) \tag{A.9}$$

where **w** should be normalised to unity length. For each feature in each iteration, a spare random vector $\mathbf{w}_b$ is generated. Then, $w_b^*[f]=0$ if $w_b[f]\neq 0$, otherwise $w_b^*[f]=1$. For each feature at each iteration, matched pairs $\mathbf{w}_b$ and $\mathbf{w}^*{}_b$ are generated. Then, $\mathbf{w}_b$ is used to obtain the kernel matrix $\kappa_w(\mathbf{x}_i,\mathbf{x}_j)$ and determine an estimation of **X** as $\hat{\mathbf{X}}_b$. Similarly, $\mathbf{w}^*{}_b$ is used to obtain $\kappa^*{}_w(\mathbf{x}_i,\mathbf{x}_j)$ and determine an estimation of **X** as $\hat{\mathbf{X}}_b^f$. The importance of preimages $\hat{\mathbf{X}}_b$ and $\hat{\mathbf{X}}_b^f$ is then used to assess the importance of each feature.

B. PRNN CONVERGENCE

Referring to (20), the equality $\nabla f(\boldsymbol{\lambda})=\mathbf{F}\boldsymbol{\lambda}$ holds where $\mathbf{F}\in R^{n\times n}$ is a positively defined matrix is.

$$\mathbf{F}=\begin{bmatrix} d^2\boldsymbol{\psi}^T(\mathbf{x})\boldsymbol{\psi}(\mathbf{x}) & d\,d\,\boldsymbol{\psi}^T(\mathbf{x})\boldsymbol{\psi}(\mathbf{x}) & \cdots & d_1d_n\boldsymbol{\psi}^T(\mathbf{x})\boldsymbol{\psi}(\mathbf{x}_n)\\ d\,d\,\boldsymbol{\psi}^T(\mathbf{x})\boldsymbol{\psi}(\mathbf{x}) & d^2\boldsymbol{\psi}^T(\mathbf{x})\boldsymbol{\psi}(\mathbf{x}) & \cdots & d_2d_n\boldsymbol{\psi}^T(\mathbf{x})\boldsymbol{\psi}(\mathbf{x}_n)\\ \vdots & \vdots & \ddots & \vdots\\ d_nd_1\boldsymbol{\psi}^T(\mathbf{x}_n)\boldsymbol{\psi}(\mathbf{x}_1) & \cdots & \cdots & d_n^2\boldsymbol{\psi}^T(\mathbf{x}_n)\boldsymbol{\psi}(\mathbf{x}_n)\end{bmatrix}$$

The expressions for $\nabla\mathbf{g}(\boldsymbol{\lambda})$ and matrix **D** are given below:

$$\nabla\mathbf{g}(\boldsymbol{\lambda})=-\mathbf{1}_{n\times 1},\ \mathbf{D}=diag\{d_1,d_2,...,d_n\}$$

Consider the following Lyapunov function:

$$\begin{aligned}v(t)=&\frac{1}{2}\left(\boldsymbol{\lambda}-\boldsymbol{\lambda}^*\right)^T\left(\boldsymbol{\lambda}-\boldsymbol{\lambda}^*\right)\\&+\frac{1}{2}\left(\mathbf{y}-\mathbf{W}\mathbf{y}^*\right)^T\left(\mathbf{y}-\mathbf{W}\mathbf{y}^*\right)+\frac{1}{2}\left(\mathbf{z}-\mathbf{W}\mathbf{z}^*\right)^T\left(\mathbf{z}-\mathbf{W}\mathbf{z}^*\right)\end{aligned} \tag{B.1}$$

where $\mathbf{W}$ is a positive definite matrix. The time derivative of equation (B.1) considering (6) is as follows:

$$\begin{aligned}\dot{v}(t)=&\left(\boldsymbol{\lambda}-\boldsymbol{\lambda}^*\right)^T\left(-\mathbf{F}\boldsymbol{\lambda}+\mathbf{y}+\mathbf{D}^T\mathbf{z}\right)\\&+\left(\mathbf{y}-\mathbf{W}\mathbf{y}^*\right)^T\left(-\boldsymbol{\lambda}\right)+\left(\mathbf{z}-\mathbf{W}\mathbf{z}^*\right)^T\left(-\mathbf{D}\boldsymbol{\lambda}\right)\end{aligned} \tag{B.2}$$

Simplification of (B.3) yields the following:

$$\begin{aligned}\dot{v}(t)=&-\boldsymbol{\lambda}^T\mathbf{F}\boldsymbol{\lambda}+\boldsymbol{\lambda}^{*T}\mathbf{F}\boldsymbol{\lambda}-\boldsymbol{\lambda}^{*T}\mathbf{y}-\boldsymbol{\lambda}^{*T}\mathbf{D}^T\mathbf{z}\\&+\mathbf{y}^{*T}\mathbf{W}^T\boldsymbol{\lambda}+\mathbf{z}^{*T}\mathbf{W}^T\mathbf{D}\boldsymbol{\lambda}\end{aligned} \tag{B.3}$$

According to (4), for the optimal solution at the steady-state condition, $\mathbf{y}=\mathbf{F}\boldsymbol{\lambda}-\mathbf{D}^T\mathbf{z}$ can be written. Therefore, Equation (B.4) can be simplified as follows:

$$\dot{v}(t)=-\boldsymbol{\lambda}^T\mathbf{F}\boldsymbol{\lambda}+\mathbf{y}^{*T}\mathbf{W}^T\boldsymbol{\lambda}+\mathbf{z}^{*T}\mathbf{W}^T\mathbf{D}^T\boldsymbol{\lambda}^T \tag{B.4}$$

Rereferring to (4), $\mathbf{F}\boldsymbol{\lambda}^*=\mathbf{y}^*+\mathbf{D}^T\mathbf{z}^*$ holds. Hence, equation (B.5) can be expressed as below:

$$\dot{v}(t)=-\boldsymbol{\lambda}^T\mathbf{F}\boldsymbol{\lambda}+\boldsymbol{\lambda}^{*T}\mathbf{F}^T\mathbf{W}^T\boldsymbol{\lambda} \tag{B.5}$$

With $\mathbf{W}=\mathbf{F}^{-1}$, the following equation is obtained:

$$\dot{v}(t)=-\boldsymbol{\lambda}^T\mathbf{F}\boldsymbol{\lambda}+\boldsymbol{\lambda}^{*T}\boldsymbol{\lambda} \tag{B.6}$$

Since $\Pr(\mathbf{y}+\mathbf{g}(\boldsymbol{\lambda}))=\mathbf{y}+\mathbf{g}(\boldsymbol{\lambda})$, it can be cocluded that $\mathbf{y}\geq\boldsymbol{\lambda}^*$ holds. Therefore, based on $\mathbf{F}\boldsymbol{\lambda}=\mathbf{y}+\mathbf{D}^T\mathbf{z}$, the inquanlty $\mathbf{F}\boldsymbol{\lambda}\geq\boldsymbol{\lambda}^*$ and consequently $\boldsymbol{\lambda}^T\mathbf{F}\boldsymbol{\lambda}\geq\boldsymbol{\lambda}^T\boldsymbol{\lambda}^*$ hold. This implies that $\dot{v}(t)$ in (B.6) is negative. Therefore, the PRNN dynamics in (6) has convergence towards its equilibrium point.