Accepted manuscript (author version)

To appear in:

International Journal of Mathematical Modelling & Computations

Online ISSN: 2228-6233 Print ISSN: 2228-6225

Received: 16- November-2025

Revised: 30- December-2025

Accepted: 12- February-2026

DOI: https://doi.org/10.57647/ijm2c.2026.1604.22

ORIGINAL RESEARCH

Fault detection and severity classification of inter-turn short-circuit faults in PMSMs using a combination of projection-based support vector machine and Bayesian neural network

Abbas Darvishi[a], Seyed Mohsen Seyed Moosavi*[a], Ebrahim Aghagari[a], Behzad Moshiri[b]
[a]Department of Electrical Engineering, Ahv.C., Islamic Azad University, Ahvaz, Iran
[b]Department of Engineering, School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran

Corresponding author: Seyed Mohsen Seyed Moosavi (Email: Mohsenmoosavi@iau.ac.ir)
https://orcid.org/0000-0002-2224-4768

Abstract

This paper presents a novel approach for detecting and classifying the severity of Inter-Turn Short Circuit (ITSC) faults in a Permanent Magnet Synchronous Machine (PMSM) using a combination of Dempster-Shafer (DS), Support Vector Machine (SVM), Projection Recurrent Neural Network (PRNN), and Bayesian Neural Network (BNN). The dataset comprises three-phase current signals gathered under healthy and faulty conditions at six severity levels. Feature extraction is performed using Discrete Wavelet Transform (DWT) and Power Spectral Density (PSD), followed by Kernel Principal Component Analysis (KPCA) for dimensionality reduction. For ITSC fault detection, an optimized classifier is developed by integrating DS, SVM, and PRNN to ensure high convergence speed and robust decision-making under uncertainty. For fault severity classification, a BNN is employed to provide probabilistic predictions and enhance reliability in distinguishing fault severity levels. The proposed method achieves 98.10% accuracy in fault detection and demonstrates a strong correlation between classification performance and increasing fault severity. The BNN model outperforms LSTM by achieving 97.33% accuracy at the highest severity level and ensures high precision, recall, and F1-score across all severity levels. The results confirm that the proposed approach provides a reliable and interpretable solution for ITSC fault detection and severity estimation in PMSMs.

Keywords: Inter-turn short circuit, Permanent magnet synchronous machine, Support vector machine, Dempster-Shafer theory, Bayesian neural network.

I. Introduction

PMSMs are widely used in various applications due to their high efficiency, durability, and reliability. However, faults in their drive systems, such as problems with inverters, stator windings, or sensors, can lead to performance degradation and inefficiency. These faults can cause problems such as insulation breakdown, torque irregularities, overcurrent conditions, and, in severe cases, complete system failure [1]. The critical role of PMSMs, particularly in safety-critical applications such as electric vehicles, underlines the importance of early fault detection. Timely fault detection is essential to ensure safety, maintain system reliability, optimize performance, and implement preventive measures to extend motor lifespan [2].

ITSC faults are a significant concern in PMSMs, being the most common electrical fault. These faults occur when the insulation between adjacent turns in the stator winding deteriorates, causing a short circuit. ITSC faults can result in mechanical stress damage to the shaft and winding ends due to transient impulse currents and may also cause irreversible demagnetization of the permanent magnet [3]. Adverse effects include increased current, localized heating, distortion of the magnetic field, and torque ripples. Designing the motor with high inductance to limit short-circuit current is crucial [4]. Accurate detection, diagnosis, and mitigation strategies are essential to maintain the reliability and longevity of PMSMs.

An effective approach to assessing PMSM's ITSC faults involves structural analysis using the dynamic model of the system. A fault model incorporating PWM voltages was presented in [5], which showed that PWM harmonics can exceed 50% of the fundamental component in fault currents. The impact of back EMFs and PWM voltages on fault currents across speed ranges was also analyzed. A method for the detection of early-stage ITSC faults in PMSMs used in elevators is presented in [6]. It employed a nonlinear unknown input observer combined with least squares estimation to enhance fault detection sensitivity. A physics-based method using an open-loop back EMF estimator was introduced in [7], which demonstrated high sensitivity to varying speeds and harmonic loads. It was validated by experiments and Finite Element Analysis (FEA) to offer greater accuracy in modeling ITSC faults. An FEA-based equivalent circuit model for analyzing short-circuit currents by deriving inductance matrices and back-EMF is presented in [8], where faults were simulated by disconnecting specific turns and current sources.

Unlike model-based approaches, deep learning algorithms are increasingly used for detecting ITSC faults in PMSMs, relying on measured data for fault diagnosis. A fault diagnosis method using a conditional generative adversarial net and an optimized sparse autoencoder was proposed in [9]. An improved deep Q-network framework is introduced in [10] enhancing feature extraction and achieving high diagnostic accuracy. A Convolutional Neural Network (CNN) for diagnosing permanent magnet damage under various operational conditions is presented in [11]. A Siamese CNN for diagnosing ITSC faults is introduced in [12]. This method uses advanced data processing techniques to enhance diagnosis accuracy. A multiscale convolutional residual neural network incorporating attention mechanisms is presented in [13], achieving high accuracy in early diagnosis of ITSC faults. A mathematical model for PMSMs under ITSC conditions is developed in [14], which employed SVM and CNN for feature selection and diagnosis accuracy comparison.

BNNs offer a significant advantage over traditional deep neural networks (DNNs) in fault detection applications due to their ability to quantify uncertainty. Unlike DNNs that provide deterministic outputs, BNNs produce probability distributions over predictions, allowing for reliable estimation of uncertainty in fault detection scenarios [15]. A method combining experimental setups with deep learning algorithms for automatic feature extraction from stator current signals was presented in [16]. A fault diagnosis method combining multiscale residual dilated CNN with a bidirectional LSTM network is proposed in [17]. A deep learning algorithm optimized through Bayesian optimization is introduced in [18]. It outperformed other methods such as LSTM and CNN variants for PMSM fault detection. Besides BNNs, Dempster–Shafer (DS) theory also provides an effective approach for uncertainty estimation in data-driven methods. A technique combining interval-valued belief entropies with Shannon entropy is proposed in [19] to improve decision-making under incomplete or uncertain information.

A critical issue in fault detection is efficiently classifying fault severity. Accurate detection of the severity of faults in PMSMs is crucial for preventing catastrophic failures and minimizing downtime. Early detection allows for optimized maintenance schedules, reducing unnecessary inspections and maximizing the motor's lifespan. Furthermore, understanding fault severity provides insights into underlying degradation causes, enabling proactive measures [20]. A method for estimating the number of shorted turns using low AC voltage excitation is proposed in [21]. An effective method for monitoring fault severity using a proportional-integral observer is presented in [22]. A control strategy named maximum torque per circulating current is introduced in [23], which retained output torque while keeping the circulating current within safe limits. An approach for estimating the number of shorted turns using low sinusoidal voltage excitation is proposed in [24]. This method simplifies failure characterization and enhances mitigation strategies.

Despite significant advances in ITSC fault detection in PMSMs, several challenges remain. Model-based approaches are computationally intensive and may not fully capture the complex dynamics of faults under varying conditions. Deep learning methods, although accurate, require extensive training data and can struggle with limited or noisy datasets. These approaches often produce deterministic outputs and cannot quantify the uncertainty of fault severity.

In this paper, a new combination of SVM, DS, and PRNN is proposed to detect ITSC faults in a PMSM. Unlike previous methods, the proposed approach ensures a simple structure and guaranteed convergence. DS theory provides a mechanism for combining hypotheses under uncertain and incomplete information. DS theory has been used in electric motor fault classification, including broken rotor bars, bearing faults, and rotor imbalance [25], bearing fault detection with ensemble approaches [26], and fusion with deep neural networks [27].

A challenge in DS theory is handling conflicting evidence during fusion. An improved probability adjustment method for combining multiple single-label classifiers is proposed in [28], which adjusts scores based on ranks, and demonstrating superior performance across datasets. The DS output function, described in a linear regression form, is reformulated as a constrained classification problem in the SVM. The SVM's immunity to local errors, ability to classify high-dimensional data, and low noise sensitivity are exploited.

To solve the constrained optimization problem, a PRNN is used as a numerical optimizer [29]. The dynamic equations of PRNNs are based on the theory of variational inequalities to solve general optimization problems [30]. PRNNs offer simple structure, high convergence speed, and easy implementation. They have been developed for solving linear programming [31], quadratic programming [32], nonlinear programming [33], and optimal control [34].

This paper also proposes an approach that combines feature extraction and feature selection with BNN for robust and accurate severity detection of ITSC faults in PMSMs. Specifically, the proposed method introduces a pre-processing stage where the DWT, and PSD are applied to the time-domain data. A feature extraction process is then employed to capture critical statistical measures indicative of fault conditions, while a KPCA feature selection method is utilized to prioritize the most significant features affecting fault severity. A BNN is trained to classify the ITSC fault severity. The integration of a BNN further improves the system's reliability by not only classifying fault severity but also providing confidence levels in its predictions, and addressing uncertainty in fault diagnosis. The key contributions of this paper are

- A pre-processing stage using DWT, and PSD on the time-domain data.
- Feature extraction which includes statistical measures.
- KPCA feature selection to prioritize the most significant features impacting fault severity.
- A new fault detection approach using a combination of DS, SVM and PRNN.
- Simple structure and fast convergence of the optimiser.
- Address uncertainty in the fault severity prediction using the BNN.

The structure of the paper is as follows: Section II introduces the methodology developed for assessing the severity of ITSC faults in PMSMs. Section III includes the data pre-processing, which highlights the DWT and PSD results. Sections IV and V describe the feature extraction and feature selection stages. Fault detection using the combination of DS, SVM and PRNN is presented in section VI. Section VII includes the BNN architecture along with the simulation results for fault severity classification. Finally, Section VIII concludes the paper.

II. Proposed fault severity detection algorithm

Fault severity detection, particularly the identification of low-level fault severity, remains a complex and challenging problem in the field of fault detection. This paper proposes a probability-based, data-driven approach to the detection of ITSC fault severity in PMSMs. The approach is outlined in Fig. 1, which shows the main components and interrelationships of the proposed method. The process initiates with the preprocessing of raw datasets, which include three-phase stator current signals captured under various operating conditions. In this stage, advanced frequency domain techniques, such as DWT and PSD analysis, are applied to the stator current signals.

This stage is crucial to get insights into the characteristics of ITSC fault severity by capturing transient and oscillatory behaviours. Feature extraction includes a series of statistical measures such as mean, standard deviation, skewness, kurtosis and crest factor. These measures are applied to both time and frequency domain data.

To further refine the feature space and eliminate redundant features, a KPCA algorithm is employed as a nonlinear feature selection method. To detect the ITSC faults in the PMSM, a combination of DS, SVM and PRNN is introduced. The proposed classifier combines a straightforward structure with fast convergence, strong tolerance to local errors. A BNN as the predictive mode is finally trained to classify the ITST severities in a PMSM. A dataset of stator currents with six levels of severities are used. The proposed algorithm provides a reliable framework for uncertainty quantification in the fault severity predictions in a PMSM.

III. Preprocessing stage

Electric machine components often operate under extreme and dynamic conditions, where the monitored signals are typically nonlinear, nonstationary, and heavily affected by noise, leading to high uncertainty and unpredictability. These challenges necessitate effective preprocessing, where methods like DWT and PSD analysis help eliminate noise and highlight critical features, ultimately enhancing the reliability of fault detection systems [35].

DWT is particularly advantageous due to its capability to decompose signals into various frequency components, which facilitates the identification of anomalies. PSD is valuable for analysing the power distribution across different frequencies, which is crucial for detecting subtle changes associated with ITSC faults. By integrating these techniques, the performance of predictive maintenance systems is significantly enhanced, leading to timely and accurate fault diagnosis.

In this paper, Welch's method is used to for estimating the PSD of the signal. It works by dividing the signal into overlapping segments, applying a window function to each segment to minimize spectral leakage, and then computing the Discrete Fourier transform for each windowed segment. The resulting periodograms are squared and averaged across all segments to produce the final PSD estimate.

The dataset used in this paper contains three-phase stator current measurements collected from an experimental setup under healthy and faulty operating conditions of a 3 kW PMSM, as reported in [36]. The data acquisition setup includes a Hioki CT6700 current sensor on each phase (U, V, and W), with the signals recorded through an NI-9775 module. All current measurements are sampled at 100 kHz over a 120-second operating period. The tests are performed at a constant speed of 3000 rpm and a load torque equal to 15% of the rated limit.

The PMSM used in this study is part of a wider experimental dataset that includes three motors rated at 1.0 kW, 1.5 kW, and 3.0 kW, all sharing similar characteristics such as three-phase configuration, four-pole structure, and identical loading and speed conditions. The fault type considered in this work is an ITSC, applied to the PMSM at six discrete severities: 1.78%, 2.13%, 2.65%, 3.5%, 9.81%, and 17.68%. These severity levels represent the percentage of the affected turns within one phase. They are selected to provide a balance between capturing meaningful variations in fault progression and maintaining manageable complexity in data analysis.

Fig. 2 presents the three-phase stator current signals for the healthy condition and the 9.81% severity ITSC fault, highlighting the variations introduced by the fault.

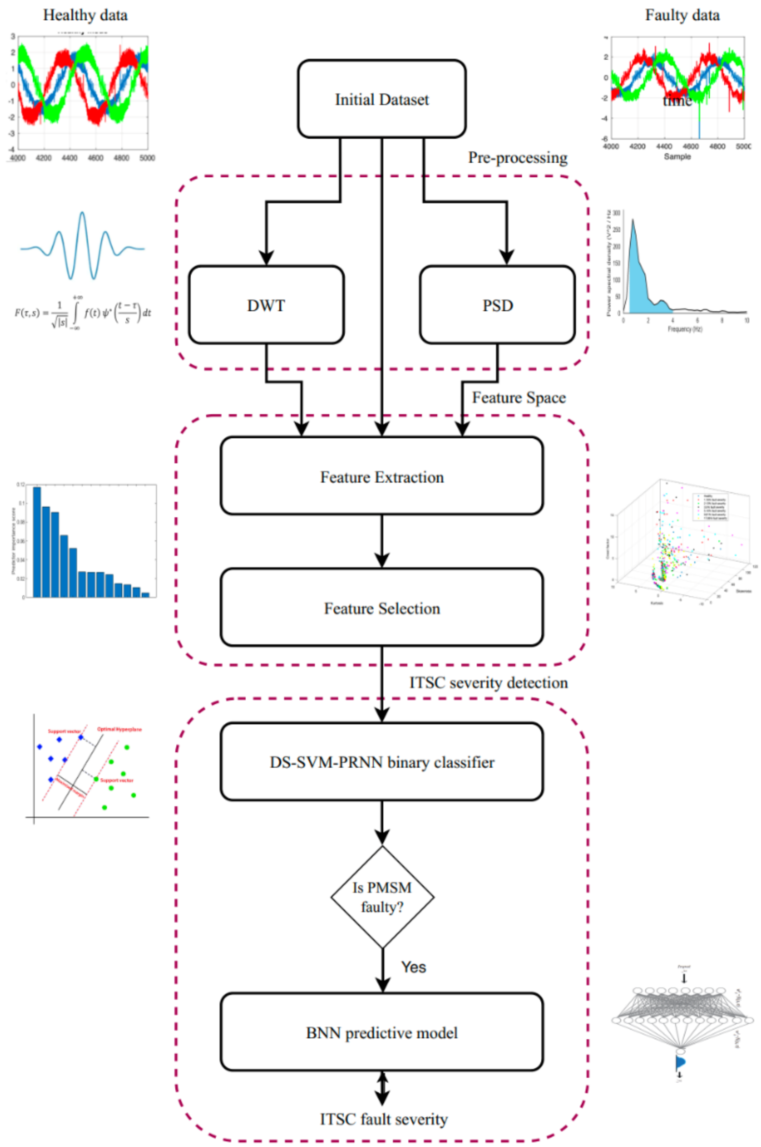


Fig. 1. Block diagram of the proposed ISTC fault severity detection algorithm

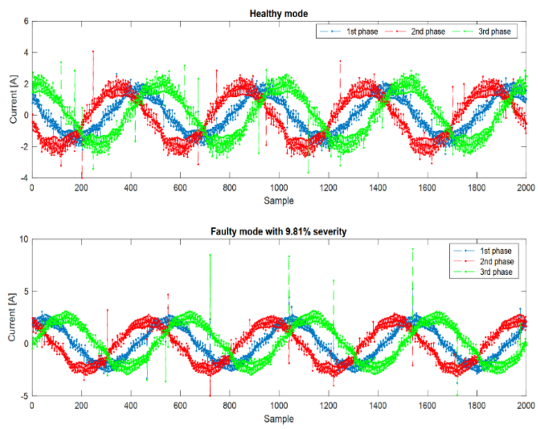
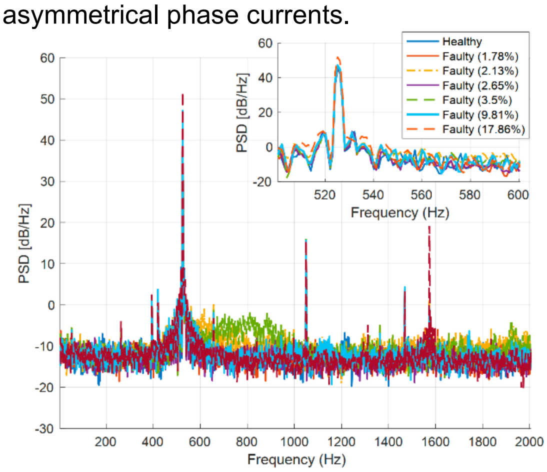
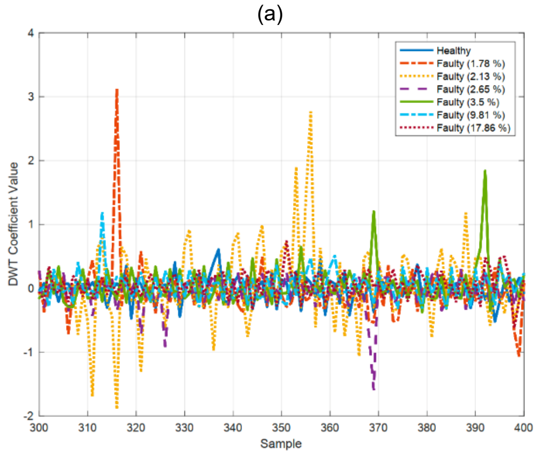


Fig. 2. Three-phase current signals for both healthy and faulty data with 9.81% severity

Fig. 3 (a) presents the PSD of the three-phase currents for different fault severities using the Welch method. A distinct peak is observed near 500 Hz in both healthy and faulty conditions, but its amplitude increases with fault severity. This indicates that fault progression leads to increased harmonic content due to asymmetrical phase currents.



(a)



(b)

Fig 3. a) PSD of three-phase currents for different fault severities using the Welch method. The inset highlights spectral differences in the 520–600 Hz range, where fault-induced frequency shifts become more evident. b) DWT detail coefficients over a selected sample range, showing the increasing fluctuations with fault severity.

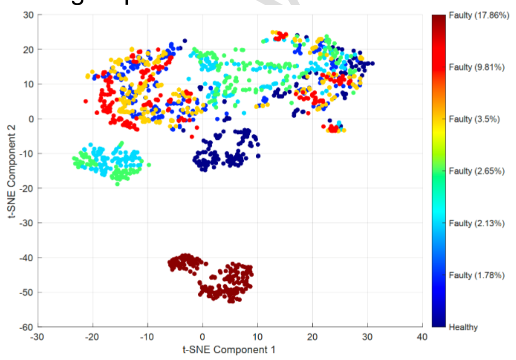
The inset zooms into the 520–600 Hz range, highlighting the spectral differences across severities. Notably, the most severe fault (17.86%) exhibits a broader frequency spread and higher power levels, which indicates increased distortion in the current signal. Figure 3 (b) depicts the wavelet detail coefficients for different severities, indicating localized variations in the current signal. The healthy condition exhibits a relatively uniform distribution, whereas faulty cases show sharp fluctuations, especially at specific sample points. The most severe faults (9.81% and 17.86%) present the most pronounced deviations, which confirms the presence of high-frequency distortions.

IV. Feature extraction

Effective feature extraction is essential for detecting and diagnosing ITSC faults in PMSMs. These faults are among the most prevalent and damaging in PMSMs, potentially causing severe operational disruptions if not detected early. There are a variety of statistical measures to create a feature set. In this paper, mean as the average value of a signal over time, standard deviation as the signal's dispersion in terms of variance around the mean, kurtosis as a measure of distribution peakedness, skewness as a measure of distribution asymmetry, crest factor as the peak-to-average ratio of a signal, and clearance factor as a measure of the alignment of motor components are chosen [37].

To create the feature space, first both healthy and faulty current signals are segmented to small packages with size of 5000 samples. Then, DWT and PSD of each segment are derived. Finally, the statistical features are obtained by applying equations in (6) on the segmented data. Finally, 24 features are calculated with 1400 samples corresponding to each feature.

Figure 4 presents a t-distributed Stochastic Neighbour Embedding (t-SNE) projection, which reduces the feature space into two dimensions while preserving the high-dimensional structure. Each point represents a segment, and results correspond to different severity levels, from healthy (dark blue) to the most severe fault (dark red). Clustering behaviour can be observed, which indicates that healthy data is well-separated from the faulty cases, and different severities also form distinct groups.



(b)

Fig 4. t-SNE projection of extracted features, which shows the separability of healthy and faulty data in a two-dimensional space.

V. Feature selection using KPCA

Kernel Principal Component Analysis (KPCA) is an advanced feature selection technique that is particularly useful when dealing with data that has non-linear relationships. This higher dimensional transformation allows KPCA to reveal complex patterns in the data, making it more effective for both feature extraction and dimensionality reduction [38]. By focusing on the features associated with the largest eigenvalues in this transformed space, KPCA improves the identification of the most relevant features. Fig. 5 shows the ranked features using KPCA algorithm. The three most important features are clearance factor of DWT approximation and detail coefficients, and stator currents. The heatmap in the most important features are used to train the fault detection algorithm.

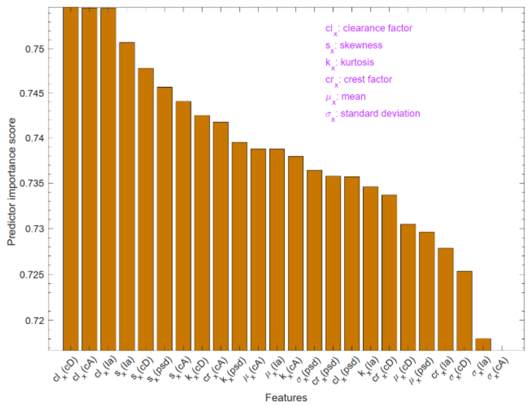


Fig 5. KPCA-based feature ranking for fault classification.

VI. DS fusion rule as a regression form

The Dempster-Shafer method offers the flexibility of classifying data without mapping a function for each element in a set, which is useful to model uncertainties. In a fault detection system, a detection framework is a set of *n* classes in the pattern space. Despite its many advantages, DS theory lacks adaptability and can be challenging for data classification in high-dimensional spaces. In addition, the paradox problem between propositions is a fundamental challenge in this area. In this paper, we adopt a new strategy in which the DS fusion rule is suitably described based on SVM in order to exploit the advantages of this method in data classification tasks. SVM offers relatively simple training, does not suffer from local minima, excels in high-dimensional data classification, and is less sensitive to noise. The basic assignment function for the first stage is described as follows:

$$\mathbf{m}^1 = \boldsymbol{\varphi}^1\mathbf{w}^1 \tag{1}$$

where

$$\mathbf{m}^1 = \begin{bmatrix} m_1^1 \\ m_2^1 \end{bmatrix},\ \boldsymbol{\varphi}^1 = s^1 = \exp\left(-\frac{\left\|\mathbf{x}^1 - \overline{\mathbf{x}}^1\right\|}{\sigma_1^2}\right),\ \mathbf{w}^1 = \begin{bmatrix} a_1^1 \\ a_2^1 \end{bmatrix}$$

where $\overline{\mathbf{x}}^1$ and σ_1 represent the center and width of the mass function (1). We define the activation vector as

$$\boldsymbol{\mu}^1 = \begin{bmatrix} \mu_1^1 & \mu_2^1 \end{bmatrix} = \begin{bmatrix} m_1^1 & m_2^1 \end{bmatrix} = \boldsymbol{\varphi}^1\mathbf{w}^1 \tag{2}$$

The first and second components of the DS fusion rule can be expressed as follows:

$$\begin{cases} \mu_1^2 = \mu_1^1 m_1^2 + \mu_1^1 m_3^2 + m_3^1 m_1^2 \\ \mu_2^2 = \mu_2^1 m_2^2 + \mu_2^1 m_3^2 + m_3^1 m_2^2 \end{cases} \tag{3}$$

The components of the basic assignment function for the second stage are given below:

$$\mathbf{m}^2 = \boldsymbol{\varphi}^2\mathbf{w}^2,\ \mathbf{m}^2 = \begin{bmatrix} m_1^2 \\ m_2^2 \end{bmatrix},\ \boldsymbol{\varphi}^2 = s^2,\ \mathbf{w}^2 = \begin{bmatrix} a_1^2 \\ a_2^2 \end{bmatrix} \tag{4}$$

The activation vector for the second stage is

$$\boldsymbol{\mu}^2 = \overline{\boldsymbol{\varphi}}^2\mathbf{w}^2 + \mathbf{b}^2 \tag{5}$$

where

$$\boldsymbol{\mu}^2 = \begin{bmatrix} \mu_1^2 \\ \mu_2^2 \end{bmatrix},\ \overline{\boldsymbol{\varphi}}^2 = \begin{bmatrix} \mu_1^1 + \mu_3^1 & 0 \\ 0 & \mu_2^1 + \mu_3^1 \end{bmatrix}\boldsymbol{\varphi}^2,$$

$$\mathbf{b}^2 = \begin{bmatrix} \mu_1^1 m_3^2 \\ \mu_2^1 m_3^2 \end{bmatrix},\ m_3^2 = \left(1 - s^1\right)\left(1 - s^2\right)$$

For the *n*th step, we can write:

$$\boldsymbol{\mu}^n = \overline{\boldsymbol{\varphi}}^n\mathbf{w}^n + \mathbf{b}^n \tag{6}$$

where

$$\boldsymbol{\mu}^n = \begin{bmatrix} \mu_1^n \\ \mu_2^n \end{bmatrix},\ \overline{\boldsymbol{\varphi}}^n = \begin{bmatrix} \mu_1^{n-1} + \mu_3^{n-1} & 0 \\ 0 & \mu_2^{n-1} + \mu_3^{n-1} \end{bmatrix}\boldsymbol{\varphi}^n,$$

$$\mathbf{b}^n = \begin{bmatrix} \mu_1^{n-1} m_3^n \\ \mu_2^{n-1} m_3^n \end{bmatrix},\ m_3^n = \prod_{i=1}^{n}\left(1 - s^i\right)$$

Remark: At step *i*, $\mathbf{w}^{i-1}$ is used as a known information. It means that the dimensions of $\boldsymbol{\varphi}^i$ and the $\mathbf{w}^i$ will always remain constant.

The final dimension of the activation vector in each stage is equal to 3, while the dimension of the target vector, i.e., the class labels considering binary classification, is equal to 2. Therefore, the output vectors for each class are defined as follows:

$$\begin{cases} P_{v,1}^i = \mu_1^i + v\mu_3^i \\ P_{v,2}^i = \mu_2^i + v\mu_3^i \end{cases},\ i = 1, 2, \ldots, n \tag{7}$$

where 0 < *v* < 1. Choosing values of 0, 1 and 0.5 for *v* represents credibility, reasonableness and probability of error in the diagnostic statements of the faults, respectively. Using (6), the regression form of (7) is:

$$\overline{\mathbf{P}}^i = \overline{\boldsymbol{\varphi}}^i\mathbf{w}^i + \overline{\mathbf{b}}^i,\ i = 1, 2, \ldots, n \tag{8}$$

where

$$\overline{\mathbf{P}}^i = \begin{bmatrix} P_{v,1}^i \\ P_{v,2}^i \end{bmatrix} - \mathbf{b}^i = \begin{bmatrix} p_1^i \\ p_2^i \end{bmatrix},\ \overline{\mathbf{b}}^i = \begin{bmatrix} \mu_3^2 \\ \mu_3^2 \end{bmatrix}v,\ i = 1, 2, \ldots, n$$

Between the two parameterized components in (8), the one with the higher probability assigns the data to the first class. For the binary classification space in the SVM, the separating boundary is

$$p_1^i = \varphi_1^i w_1^i + b_1^i,\ \begin{cases} \varphi_1^i = \left(\mu_1^{i-1} + \mu_3^{i-1}\right)s^i \\ b_1^i = \mu_3^i v \end{cases} \tag{9}$$

1. DS fusion output to a SVM problem

Consider $\Gamma = \left\{\left(x_1, z_1\right), \left(x_2, z_2\right), \ldots, \left(x_n, z_n\right)\right\}$ as a dataset. The aim is to find the hyperplane

$\left(w_1^i, b_1^i\right)$ by maximizing the following optimization problem:

$$\max \sum_{i=1}^{n} \alpha^i - \frac{1}{2}\left(\sum_{i=1}^{n} z^i \alpha^i \varphi_1^i\left(x^i\right)\right)\left(\sum_{j=1}^{n} z^j \alpha^j \varphi_1^j\left(x^j\right)\right) \quad (10)$$
$$s.t. \ \sum_{i=1}^{n} z^i \alpha^i = 0, \quad \alpha^i \geq 0 \ , \ i = 1,2,...,n$$

where $\varphi_1^i\left(x^i\right)$ represents the non-linear mapping function. α^i is the Lagrange multiplier. For the separable training dataset Γ, and assuming that α^{i*} is the optimal solution of (10), we can define the following dual optimization problem [39].

The optimal weight vector w_1^{i*} is given by $w_1^{i*} = \sum_{i=1}^{n} z^i \alpha^{i*} \varphi_1^i\left(x^i\right)$ where the maximum boundary hyperplane is $1/\left|w_1^{i*}\right|$. The optimal value of the offset b_1^{i*} can be determined as follows:

$$b_1^{i*} = -\frac{\max_{z^i=-1}\left(\varphi_1^i\left(x^i\right) w_1^{i*}\right) + \min_{z^i=1}\left(\varphi_1^i\left(x^i\right) w_1^{i*}\right)}{2} \quad (11)$$

Consequently, the function $sign\left(\varphi_1^i\left(x^i\right) w_1^{i*} + b_1^{i*}\right)$ is the binary classifier.

VII. PRNN to solve SVM problem

The main reasons to extend the PRNN to obtain the optimal solution of the nonlinear optimization problem (10) is because the PRNN has a high convergence rate, simple structure, ease of practical implementation, a guarantee of solution convergence, and the ability to be extended to different optimization problems. Consider the dual of the nonlinear optimization problem in (10) below:

$$\min_{\alpha^i} \ f\left(\boldsymbol{\alpha}\right) = \frac{1}{2}\left(\sum_{i=1}^{n} z^i \alpha^i \varphi_1^i\left(x^i\right)\right)^2 - \sum_{i=1}^{n} \alpha^i$$
$$s.t. \ g\left(\boldsymbol{\alpha}\right) = -\boldsymbol{\alpha} \leq 0 \ , \ h\left(\boldsymbol{\alpha}\right) = \mathbf{Z}^T \boldsymbol{\alpha} = 0$$
$$(11)$$

Based on optimality conditions for a Lagrange function with multiplayers $\left(\boldsymbol{\eta}, \rho\right)$ [33]:

$$\begin{cases} -\boldsymbol{\beta}^T \boldsymbol{\alpha} \boldsymbol{\beta} + \mathbf{1} + \boldsymbol{\eta} + \mathbf{Z}\rho = 0, \\ -\boldsymbol{\alpha} \leq 0, \boldsymbol{\eta} \geq 0, \mathbf{Z}^T \boldsymbol{\alpha} = 0, \boldsymbol{\eta}^T \boldsymbol{\alpha} = 0 \end{cases} \quad (12)$$

where

$\mathbf{Z} = \left[z^1 \quad z^2 \quad \cdots \quad z^n\right]^T$, $\boldsymbol{\alpha} = \left[\alpha^1 \quad \alpha^2 \quad \cdots \quad \alpha^n\right]^T$,

$\boldsymbol{\beta} = \left[z^1 \varphi_1^1 \quad z^2 \varphi_1^2 \quad \cdots \quad z^n \varphi_1^n\right]^T$

The projection operator $\Pr_\Omega(.)$ is defined as follows:

$$\Pr_\Omega\left(\boldsymbol{\alpha}\right) = \max\left\{0 \, , \boldsymbol{\alpha}\right\} \quad (13)$$

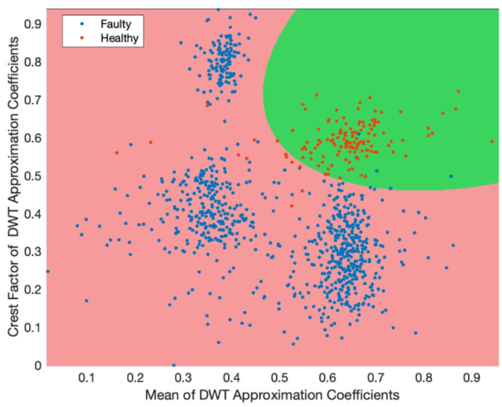
The PRNN dynamics to satisfy conditions in (13) and obtain the optimal solution $\boldsymbol{\alpha}^{i*}$ is given below:

$$\frac{d}{dt}\begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\eta} \\ \rho \end{pmatrix} = \varepsilon \begin{pmatrix} -\boldsymbol{\beta}^T \boldsymbol{\alpha} \boldsymbol{\beta} + \mathbf{1} + \boldsymbol{\eta} + \mathbf{Z}\rho \\ -\boldsymbol{\eta} + \Pr_\Omega\left(\boldsymbol{\eta} - \boldsymbol{\alpha}\right) \\ \mathbf{Z}^T \boldsymbol{\alpha} \end{pmatrix} \quad (14)$$

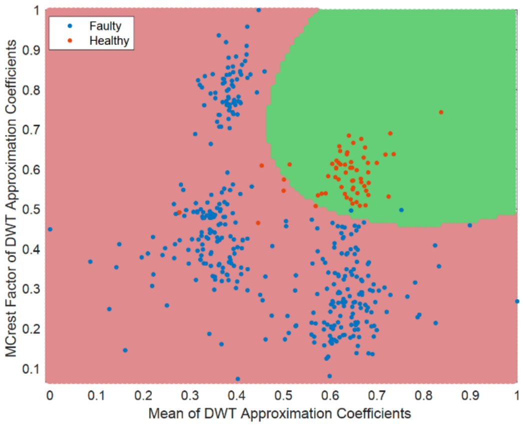
where $\varepsilon > 0$ is the learning rate of PRNN. The dataset to evaluate the proposed fault detection algorithm is divided into a training set (70%) and a testing set (30%).

Two most salient features are used as the inputs of the predictor model. Several key metrics including accuracy, precision, recall, specificity and F1 score are used to characterise the performance of the fault detection algorithm. Based on feature selection method, we assessed the proposed method with different number of features. The best results are obtained using mean and crest factor of DWT approximation coefficients. The classification boundaries learned by the proposed method is shown in Fig. 6.

It provides clear decision regions for the two classes with faulty conditions represented in blue and healthy conditions in red. The training phase decision boundary in Figure 6 (a) shows a well-separated distribution of the faulty and healthy samples, indicating that the model effectively learned the classification pattern. Similarly, in the testing phase decision boundary in Figure 6 (b), the model successfully generalizes its learned decision boundary to unseen data, showing minimal misclassification.



(a)



(b)

Fig 6. a) ITSC Fault classification using the proposed method in training phase using mean and crest factor of DWT approximation coefficients. b) ITSC Fault classification using the proposed method in testing phase using mean and crest factor of DWT approximation coefficients.

According to the confusion matrices in Figure 7, in the training phase, the model correctly classified 835 faulty samples and 125 healthy samples, with only 6 healthy misclassified as faulty and 14 faulty misclassified as healthy. This suggests that the model has a strong learning capability with minimal misclassification errors. For the testing phase, the SVM model correctly identified 357 faulty samples and 55 healthy samples, with only 2 false positives and 6 false negatives.

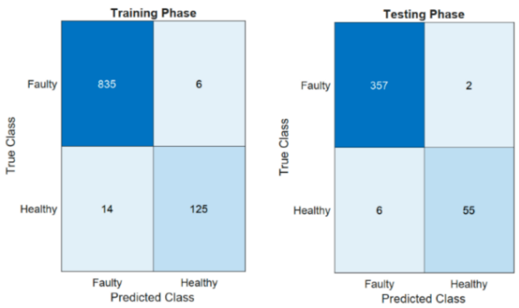


Fig 7. Confusion matrices for training and testing phases.

A summary of key metrics to evaluate the performance of the proposed fault detection algorithm is given in Table 1. The high accuracy values, with 97.96% in training and 98.10% in testing, indicate that the model correctly classifies most samples in both phases. Precision is 99.29% for training and 99.44% for testing, showing that when the model predicts a sample as faulty, it is highly likely to be correct.

Table 1: Summary of metrics used to evaluate the model in the training and testing phases.

Metric	Training Phase	Testing Phase
Accuracy	97.96%	98.1%
Precision	99.29%	99.44%
Recall	98.35%	98.35%
F1 Score	98.82%	98.89%

VIII. Fault severity classification using BNN

By integrating Bayesian inference into neural network architectures, BNNs offer probabilistic predictions that quantify uncertainty. The capacity of BNNs to handle uncertainty and offer probabilistic outputs represents a significant advancement over traditional neural networks in the domain of fault classification and diagnostic reliability. Unlike traditional networks, the weights and biases in a BNN are treated as random variables, each associated with a probability distribution.

According to Figure 8, the BNN architecture comprises an input layer, two fully connected Bayesian layers, a dropout layer for regularization, a SoftMax layer for probability estimation, and an output layer for the final classification decision. Architecture of the BNN used in the paper, consisting of an input layer, two fully connected layers, two activation layers, a dropout layer and an output layer.

For class fault condition, the BNN demonstrated a precision of 85.19%, a recall of 87.34%, and an F1 score of 86.25%. The higher recall value indicates that the model is particularly adept at identifying actual faults. On the testing dataset, the BNN maintained an accuracy of 86.00%, which is slightly higher than the training accuracy. This means that the model generalizes well to new, unseen data and is not overfitting to the training set. One of the main benefits of using BNN is fault classification is its ability to describe the predicted outputs in terms of probability distributions. Figure 12 shows the 2D and 3D views of distribution of the predicted outputs for test phase.

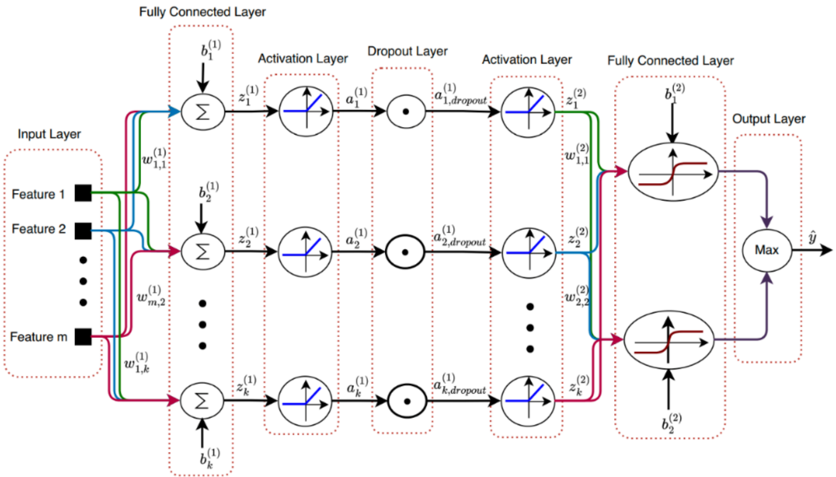


Fig 8. Architecture of the BNN used in the paper, consisting of an input layer, two fully connected layers, two activation layers, a dropout layer and an output layer.

To distinguish the fault severity, six BNNs are developed where each BNN detects one of the fault severities against the heathy data. We first demonstrate the performance of BNN for the first fault severity which is 1.78% fault level. According to confusion matrix results, 59 true positives and 10 false negatives for non-fault conditions, along with 69 true positives and 12 false positives for fault conditions have been obtained. The distribution of these values shows that the BNN is well-calibrated to detect faults accurately while maintaining a reasonable balance between false positives and false negatives. During training, the BNN achieved an accuracy of 85.33%. The metrics for healthy condition show a precision of 85.51%, a recall of 83.10%, and an F1 score of 84.29%.

To further asses the BNN in terms of the prediction uncertainty, we discuss the distribution of predicted labels for healthy and faulty classes of the first test sample. According to Figure 9, the two distributions including healthy and faulty are well-separated. It indicates that the BNN has effectively learned to distinguish between these classes. The healthy class distribution is centered around a lower predicted output value μ_h while the faulty class is centered around a higher value μ_f. The narrow spread of the distributions around their respective means, marked by low standard deviations, reflects high confidence in the model's predictions and low uncertainty. Figure 10 shows the presence of minimal overlap between the two distributions further underscores the model's robustness in differentiating between healthy and faulty conditions.

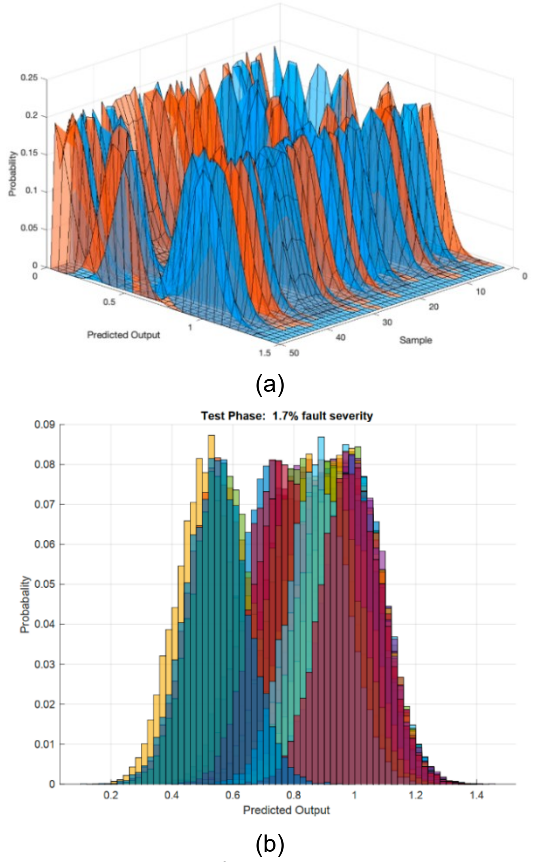


Fig. 9. Distribution of predicted outputs during the test phase for detecting ITSC with a fault severity of 1.78%.

If the predicted output for this sample aligns closely with the faulty class mean μ_f and falls within the range of one standard deviation $\mu_f \pm \sigma_f$, it would indicate a high-confidence correct classification. The clear separation and narrow distribution width prove that the BNN is highly reliable in fault detection. The fault severity classification results using the BNN (Table 1) show a clear trend of increasing accuracy as fault severity rises. At lower severity levels, the model achieves 85.33% accuracy in both training and testing, with all other metrics remaining constant. As severity increases, the model's performance improves significantly.

At 2.13% severity, accuracy rises to 88.67% (train) and 88.00% (test), with Class 0 (healthy) reaching 93.55% precision and 81.69% recall, while Class 1 (faulty) achieves 94.94% recall. This trend continues at 2.65% severity, where accuracy improves to 89.33% (train) and 90.00% (test), demonstrating better generalization.

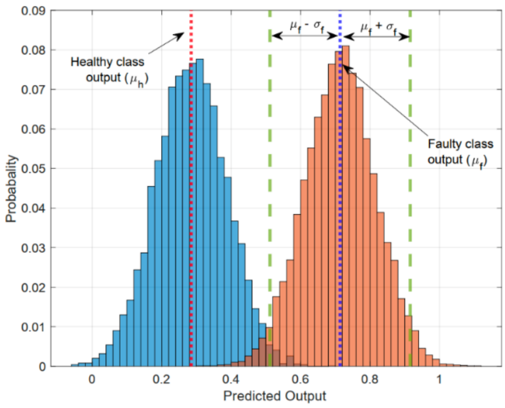


Fig. 10. Distribution of predicted outputs in the test phase for classifying the first sample.

Precision and recall values indicate that the model effectively distinguishes between healthy and faulty conditions as severity increases. At 3.5% severity, accuracy further improves to 91.33% (train) and 92.00% (test), with Class 0 achieving 93.94% precision and 87.32% recall, while Class 1 reaches 94.94% recall. The most significant improvement is observed at 9.81% severity, where accuracy reaches 94.00%, with consistently high precision and recall across both classes. Finally, at 17.68% severity, the model attains 97.33% accuracy in both training and testing, with all classification metrics also reaching 97.33%, indicating near-perfect performance. These results demonstrate that the BNN model is highly effective at detecting severe faults.

Table 2: The performance of BNN models in classifying fault severities based on several metrics.

Fault severity	Accuracy (train)	Accuracy (test)	Precision (Class 0)	Recall (Class 0)	F1 Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1 Score (Class 1)
1.78 %	85.33%	85.33%	85.33%	85.33%	85.33%	85.33%	85.33%	85.33%
2.13%	88.67%	88.00%	93.55%	81.69%	87.22%	85.23%	94.94%	89.82%
2.65%	89.33%	90.00%	92.31%	84.51%	88.24%	87.06%	93.67%	90.24%
3.5%	91.33%	92.00%	93.94%	87.32%	90.51%	89.29%	94.94%	92.02%
9.81%	94.00%	94.00%	96.97%	90.14%	93.43%	91.67%	97.47%	94.48%
17.68%	97.33%	97.33%	97.33%	97.33%	97.33%	97.33%	97.33%	97.33%

In the following, the multiscale convolutional neural network with adaptive feature weighting presented in [46] was implemented directly on the motor current signals, and its performance was evaluated. This assessment aimed to establish a fair comparison between a deep learning framework and the feature-based classification strategy proposed in this dissertation, regarding the balance between fault sensitivity and healthy-state specificity. As illustrated in Figure 11, the method in [46] achieves reasonably high levels of accuracy, precision, recall, and F1-score in the binary healthy–faulty classification task. However, the confusion matrix in Figure 12 reveals a substantial imbalance in decision outcomes: while 98.9% of faulty samples are correctly detected, only 26.7% of healthy samples are classified correctly, with 73.3% of healthy instances incorrectly flagged as faulty. This indicates a conservative decision bias toward fault declaration, whereby natural variations in healthy current signals are occasionally misinterpreted as fault-related patterns.

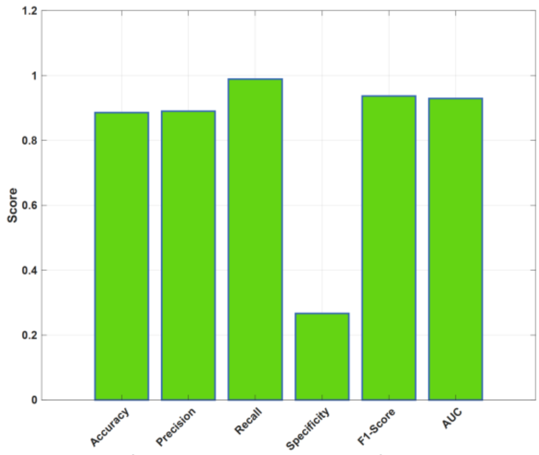


Fig. 11. Performance evaluation of the multiscale convolutional neural network with adaptive feature weighting in binary healthy–faulty classification based on various statistical metrics [46].

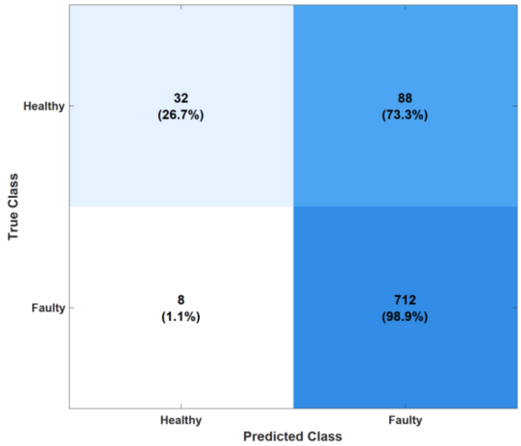


Fig. 12. Confusion matrix for binary healthy–faulty classification based on motor current signals, illustrating the algorithm's performance in detecting faulty conditions and the corresponding false-alarm rate for healthy samples [46].

The predicted probability distributions presented in Figure 13 further support this finding. Faulty samples cluster predominantly in high-probability regions, demonstrating strong confidence in fault detection, whereas healthy samples show significant overlap with the decision boundary and, in several cases, fall within the faulty prediction domain. In contrast, the proposed approach achieves a more stable balance between detecting faulty conditions and correctly identifying healthy states. Moreover, the reduced overlap between classes in the feature space demonstrates improved separability relative to the method in [46].

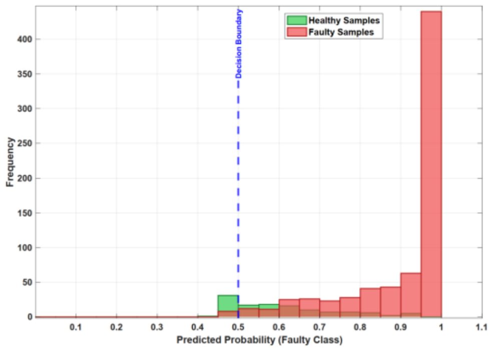


Fig. 13. Predicted probability distribution of the faulty class for healthy and faulty samples, along with the decision boundary of the algorithm in binary classification based on motor current signals [46].

For the 1.78% fault severity level representing incipient and 1.78% faults the multiscale CNN with adaptive feature weighting [46] demonstrates acceptable overall accuracy but limited effectiveness in reliably distinguishing early-stage faults. As shown in Figure 14, although the accuracy remains relatively high, the recall and F1-score drop noticeably, indicating that a non-negligible portion of faulty samples is misclassified as healthy.

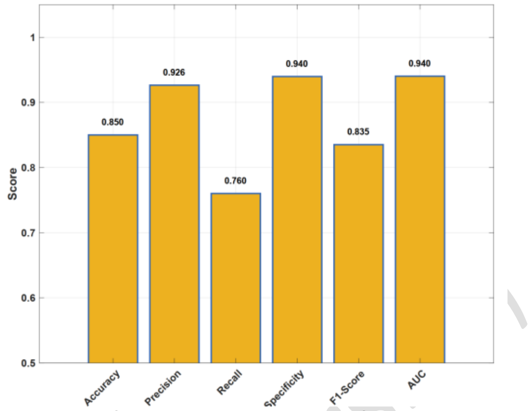


Fig. 14. Performance evaluation of the multiscale convolutional neural network with adaptive feature weighting [178] in classifying healthy data and 1.78% fault severity based on accuracy, precision, recall, specificity, and F1-score.

This limitation becomes clearer in the confusion matrix of Figure 15, where the model correctly identifies approximately 94% of healthy samples but detects only 76% of faulty samples at 1.78% severity. The resulting 24% false-negative rate is critical, as it reflects the model's difficulty in capturing weak fault signatures.
The predicted probability distribution in Figure 16 further confirms substantial overlap between healthy and faulty samples near the decision boundary, highlighting the model's sensitivity to threshold selection and its instability in borderline cases. Figure 15 illustrates the two-dimensional t-SNE projection of the features learned by the multiscale CNN [178]. Although healthy and 1.78% faulty samples form several clusters, noticeable overlap appears in the boundary regions. This indicates that the network does not achieve fully stable feature separation for low-severity faults.

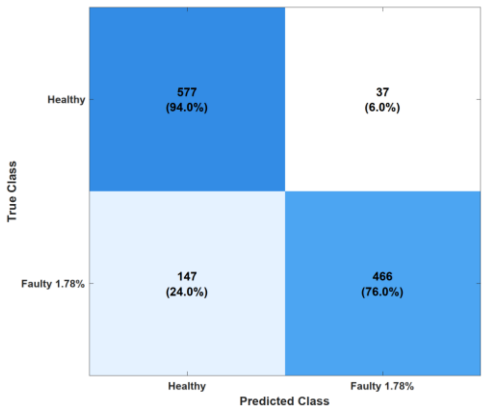


Fig. 15. Confusion matrix of the multiscale convolutional neural network [178] for classifying healthy data and short-circuit fault data with 1.78% severity.

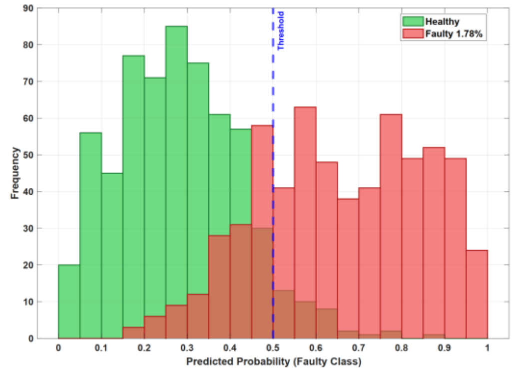


Fig. 16. Predicted probability distribution of the faulty class for healthy data and 1.78% fault severity using the multiscale convolutional neural network.

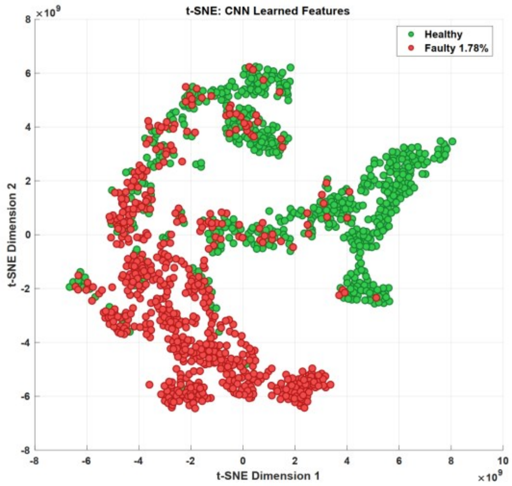


Fig. 17. Two-dimensional visualization of the learned features using t-SNE for separating healthy data and 1.78% fault severity, obtained from the multiscale convolutional neural network [46].

The proposed method's performance is also compared with the Long Short-Term Memory (LSTM) network from [17] for fault severity classification. Figure 11 illustrates the accuracy of both methods against various severity levels, showing that the proposed model consistently outperforms LSTM. At 1.78% severity, the proposed method achieves 86.00% accuracy, surpassing LSTM's 74.00% by 16.2%. This trend continues at 2.13% severity, where the proposed model reaches 88.00% accuracy, 10.2% higher than LSTM's 78.00%. As severity increases, the performance gap becomes more pronounced. At 9.81% severity, the proposed method achieves 94.00% accuracy, marking a 6.8% improvement over LSTM's 88.00%. Finally, at 17.68% severity, the proposed model attains 96.00% accuracy, exceeding LSTM's 92.00% by 4%, demonstrating its superior classification capability across all severity levels.

The confusion matrices of LSTM in Figure 12 illustrate the performance of the classification model against various fault severity levels. At lower severities (1.78% and 2.13%), there is a higher misclassification rate, particularly for Class 2 (faulty), where some faulty instances are predicted as healthy. As severity increases (2.65% to 17.68%), the classification performance improves, with fewer misclassifications and more accurate predictions for both classes. At 9.81% and 17.68% severity, the model achieves near-perfect classification for Class 2, with all faulty instances correctly identified. This trend indicates that the model struggles more with subtle faults but significantly improves as fault severity increases.

The current study uses data from only one PMSM and one type of fault, and the operating conditions are fixed. Some parts of the method, such as choosing the parameters of the mass functions and selecting signal features, still depend on manual decisions. In addition, the PRNN model has only been tested in a laboratory environment, and the proposed framework has not yet been evaluated with other useful signals like vibration or temperature. To guide future work, we suggest testing the model with real industrial data, using more advanced hybrid learning models, improving performance when only limited training data are available, exploring different types of signal features, reducing computational cost, using probabilistic tools to manage uncertainty, and studying how the model behaves when multiple faults occur together.

IX. Conclusion

This paper proposed a hybrid fault detection and severity classification method for ITSC faults in PMSMs. The approach integrates Dempster-Shafer theory, SVM, projection recurrent neural network and BNN to improve both fault detection accuracy and severity classification reliability. The fault detection stage applied DS-SVM-PRNN and achieved 98.10% accuracy in testing, which confirms its ability to differentiate between healthy and faulty conditions. The fault severity classification which was performed using BNN, demonstrated increasing accuracy as fault severity levels increases. At 1.78% fault severity, the BNN achieved 85.33% accuracy, while at 17.68% severity, it reached 97.33% accuracy, significantly outperforming the LSTM network. The BNN model's ability to quantify uncertainty further enhances decision reliability in real-world applications. A key finding is that higher fault severity levels are easier to classify, whereas lower-severity faults exhibit greater misclassification due to subtle variations in stator currents. The proposed approach provided a fast, robust, and interpretable solution for PMSM fault diagnostics.

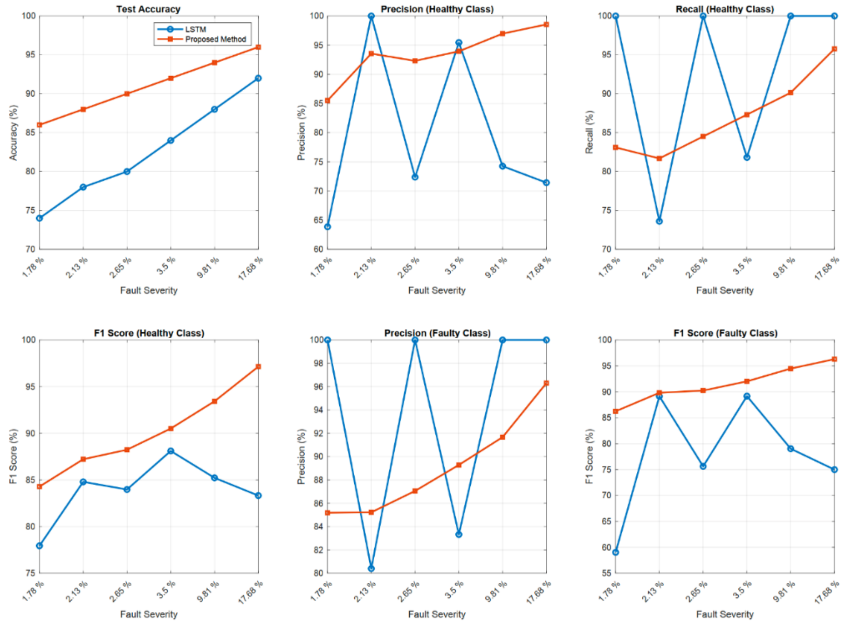


Fig 11. Comparison of the proposed method and LSTM [17] based on various evaluation metrics in the testing phase.

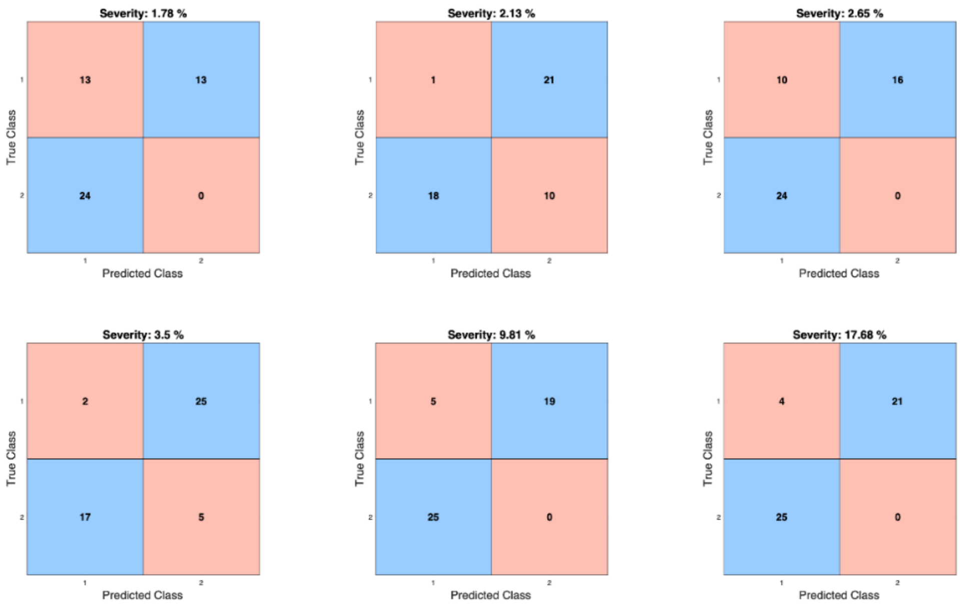


Fig 12. Confusion matrices of the LSTM algorithm from [17] in the testing phase for classifying different fault severity levels.

Funding No funding was received to assist with the preparation of this manuscript.

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

Declarations

Conflicts of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Competing interests The authors have no financial or proprietary interests in any material discussed in this article.

References

[1] Bhuiyan EA, Akhand MMA, Das SK, Ali MF, Tasneem Z, Islam MR, Moyeen SI (2020) A survey on fault diagnosis and fault tolerant methodologies for permanent magnet synchronous machines. Int J Autom Comput 17:763–787
[2] Kilic A, Weiss F (2023) Early detection of PMSM faults at static and dynamic operating points by using artificial neural network for automated electric vehicle. J Braz Soc Mech Sci Eng 45(9):493
[3] Chen X, Qin P, Chen Y, Zhao J, Li W, Mao Y, Zhao T (2022) Inter-turn short circuit fault diagnosis of PMSM. Electronics 11(10):1576
[4] Wang D, Chen Y (2020) Fault-tolerant control of coil inter-turn short-circuit in five-phase permanent magnet synchronous motor. Energies 13(21):5669
[5] Qin, Y., Li, G. J., Jia, C., & McKeever, P. (2023). Investigation of inter-turn short-circuit fault of PM machines using PWM voltage-based modeling. *IEEE Transactions on Transportation Electrification*, *10*(1), 1324-1334.
[6] Niu D, Song D (2023) Model-based robust fault diagnosis of incipient ITSC for PMSM in elevator traction system. IEEE Trans Instrum Meas
[7] Sarikhani A, Mohammed OA (2012) Inter-turn fault detection in PM synchronous machines by physics-based back electromotive force estimation. IEEE Trans Ind Electron 60(8):3472–3484
[8] Qi Y, Bostanci E, Gurusamy V, Akin B (2018) A comprehensive analysis of short-circuit current behavior in PMSM interturn short-circuit faults. IEEE Trans Power Electron 33(12):10784–10793
[9] Li Y, Wang Y, Zhang Y, Zhang J (2021) Diagnosis of inter-turn short circuit of permanent magnet synchronous motor based on deep learning and small fault samples. Neurocomputing 442:348–358
[10] Li Y, Wang R, Mao R, Zhang Y, Zhu K, Li Y, Zhang J (2023) A fault diagnosis method based on an improved deep Q-network for the inter-turn short circuits of a permanent magnet
[21] Zeng Z, Zhang M, Chen L, Huang Q, Sun D (2022) A new method to estimate the severity of interturn short-circuit fault for PMSM. In: 2022 IEEE Int Conf Power Syst Electr Technol (PSET), pp 149–154
[22] Qi Y, Zafarani M, Gurusamy V, Akin B (2019) Advanced severity monitoring of interturn short circuit faults in PMSMs. IEEE Trans Transp Electric 5(2):395–404
[23] Huang S, Aggarwal A, Strangas EG, Khoshoo B, Li K, Niu F (2021) Mitigation of interturn short-circuits in IPMSM by using MTPCC control adaptive to fault severity. IEEE Trans Power Electron 37(4):4685–4696
[24] Qi Y, Bostanci E, Zafarani M, Akin B (2018) Severity estimation of interturn short circuit fault for PMSM. IEEE Trans Ind Electron 66(9):7260–7269
[25] Yang BS, Kim KJ (2006) Application of Dempster–Shafer theory in fault diagnosis of synchronous motor. IEEE Trans Transp Electrific
[11] Skowron M, Orlowska-Kowalska T, Kowalski CT (2022) Detection of permanent magnet damage of PMSM drive based on direct analysis of the stator phase currents using convolutional neural network. IEEE Trans Ind Electron 69(12):13665–13675
[12] Wu X, Geng Y, Li M, Wang W, Tu M (2024) Inter-turn short circuit diagnosis of permanent magnet synchronous motor based on Siamese convolutional neural network under small fault samples. IEEE Sens J. 26982 – 26993.
[13] Chen Q, Dai X, Song X, Liu G (2023) ITSC fault diagnosis for five phase permanent magnet motors by attention mechanisms and multiscale convolutional residual network. IEEE Trans Ind Electron
[14] Shih KJ, Hsieh MF, Chen BJ, Huang SF (2022) Machine learning for inter-turn short-circuit fault diagnosis in permanent magnet synchronous motors. IEEE Trans Magn 58(8):1–7
[15] MacKay DJ (1992) A practical Bayesian framework for backpropagation networks. Neural Comput 4(3):448–472
[16] Mirshekali H, Santos AQ, Shaker HR (2023) A survey of time-series prediction for digitally enabled maintenance of electrical grids. Energies 16(17):6332
[17] Yan J, Senemmar S, Zhang J (2024) Inter-turn short circuit fault diagnosis and severity estimation for wind turbine generators. J Phys Conf Ser 2767(3):032021
[18] Song Q, Wang M, Lai W, Zhao S (2022) On Bayesian optimization-based residual CNN for estimation of inter-turn short circuit fault in PMSM. IEEE Trans Power Electron 38(2):2456–2468
[19] Xue Y, Deng Y (2021) Interval-valued belief entropies for Dempster–Shafer structures. Soft Comput 25(13):8063–8071
[20] Hang, J., Yan, D., Xia, M., Ding, S., & Wang, Q. (2019). Quantitative fault severity estimation for high-resistance connection in PMSM drive system. *IEEE Access*, *7*, 26855-26866.
induction motors using vibration and current signals. Mech Syst Signal Process 20(2):403–420
[26] Wang Z, Wang R, Gao J, Gao Z, Liang Y (2020) Fault recognition using an ensemble classifier based on Dempster–Shafer theory. Pattern Recognit 99:107079
[27] Zhang Z, Jiang W, Geng J, Deng X, Li X (2020) Fault diagnosis based on non-negative sparse constrained deep neural networks and Dempster–Shafer theory. IEEE Access 8:18182–18195
[28] Jebari C, Jin L, Yager RR (2024) A new probability adjustment method for combining conflicting evidences: application in classifier combination. Soft Comput 28(20):12175–12190
[29] Joya G, Atencia MA, Sandoval F (2002) Hopfield neural networks for optimization: study of the different dynamics. Neurocomputing 43(1-4):219–237

[30] Kinderlehrer D, Stampacchia G (2000) An introduction to variational inequalities and their applications. SIAM
[31] Liu Q, Wang J (2008) A one-layer recurrent neural network with a discontinuous activation function for linear programming. Neural Comput 20(5):1366–1383
[32] Xia, Y., & Wang, J. (2015). A bi-projection neural network for solving constrained quadratic optimization problems. *IEEE transactions on neural networks and learning systems*, *27*(2), 214-224.
[33] Xia Y, Wang J, Guo W (2019) Two projection neural networks with reduced model complexity for nonlinear programming. IEEE Trans Neural Netw Learn Syst 31(6):2020–2029
[34] Toshani H, Farrokhi M (2023) Optimal discrete-time sliding-mode control based on recurrent neural network: a singular value approach. Soft Comput 27(6):3083–3094
[35] Buabeng A, Simons A, Frempong NK, Ziggah YY (2024) Hybrid intelligent predictive maintenance model for multiclass fault classification. Soft Comput 28(15):8749–8770
[36] Jung W, Yun SH, Lim YS, Cheong S, Park YH (2023) Vibration and current dataset of three-phase permanent magnet synchronous motors with stator faults. Data Brief 47:108952
[37] Rajapaksha N, Jayasinghe S, Enshaei H, Jayarathne N (2021) Acoustic analysis based condition monitoring of induction motors: a review. In: 2021 IEEE South Power Electron Conf (SPEC), pp 1–10
[38] Gao Y, Ma C, Sheng A (2023) Compound fault diagnosis for cooling dehumidifier based on RBF neural network improved by kernel principal component analysis and adaptive genetic algorithm. Soft Comput 27(3):1599–1613
[39] Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press
[40] Alessio SM (2016) Discrete wavelet transform (DWT). In: Digital Signal Processing and Spectral Analysis for Scientists: Concepts and Applications, pp 645–714
[41] Rahi PK, Mehra R (2014) Analysis of power spectrum estimation using Welch method for various window techniques. Int J Emerg Technol Eng 2(6):106–109
[42] Shen K, Wang H, Chaudhuri A, Asgharzadeh Z (2023) Automatic Gaussian bandwidth selection for kernel principal component analysis. In: Int Conf Knowl Sci Eng Manag, pp 15–26
[43] Koksalmis E, Kabak Ö (2020) Sensor fusion based on Dempster–Shafer theory of evidence using a large scale group decision making approach. Int J Intell Syst 35(7):1126–1162
[44] Tang Y, Wu L, Huang Y, Zhou D (2024) An improved approach for incomplete information modeling in the evidence theory and its application in classification. Soft Comput 28(17):10187–10200
[45] Graves A (2011) Practical variational inference for neural networks. Adv Neural Inf Process Syst 24
[46] Han S, et al. (2024) Deep residual multiscale convolutional neural network with attention mechanism for bearing fault diagnosis under strong noise environment. IEEE Sens J 24(6):9073–9081

Appendix

A. DWT and PSD mathematics

The wavelets are sampled at discrete intervals, which allows DWT to decompose the signal into multiple levels of detail. The technique involves filtering the signal through a series of high-pass and low-pass filters, which split the signal into different frequency ranges. Given an input signal *x* of length *N*, two sets of coefficients are computed as approximation coefficients c_A and detail coefficients c_D. This decomposition is achieved through two parallel processes. First, the signal *s* is convolved with a lowpass filter L_D. The resulting filtered signal is then down sampled by a factor of 2 to obtain c_A. Second, the signal *s* is convolved with a high pass filter H_D. The resulting filtered signal is then down sampled by a factor of 2 to obtain c_D. The DWT uses a mother wavelet function $\psi(t)$, which is scaled and translated to create a family of wavelets [40]:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-a}{b}\right) \tag{15}$$

where *a* is the scaling parameter and *b* is the translation parameter. Alongside the wavelet function, a scaling function $\varphi(t)$ is used to approximate the signal at different resolutions. Mathematically, the approximation coefficients c_A, and detail coefficients c_D, are computed as:

$$\begin{aligned} c_A &= \int f(t)\varphi(t)\,dt, \\ c_D &= \int f(t)\psi(t)\,dt \end{aligned} \tag{16}$$

. The signal $x(n)$ with length N is first divided into k segments, each of length L with an overlap D samples between consecutive samples. Each segment $x_k(n)$ is multiplied by a window function $w(n)$ of length L. The DFT of each windowed segment is calculated as [41]

$$x_k(m) = \sum_{n=0}^{L-1} x_k^{'}(n)\, e^{-j\frac{2\pi}{L}mn} \tag{17}$$

The periodogram of the *k*th segment is

$$P_k(m) = \frac{1}{L}\left|x_k(m)\right|^2 \tag{18}$$

Finally, the estimated PSD is calculated by averaging the periodograms of all segments:

$$\hat{P}(m)=\frac{1}{K}\sum_{k=0}^{K-1}P_k(m) \tag{19}$$

B. KPCA mathematics

Let $x_i, i=1,2,...,n$ be the input data points. These points are then projected to a new space using a function $\boldsymbol{\psi}(x_i)$. It's assumed that the projected features have zero mean [42]:

$$\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\psi}(x_i)=0 \tag{20}$$

The covariance matrix $\mathbf{C}\in R^{m\times m}$ of the projected features is defined as:

$$\mathbf{C}=\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\psi}(x_i)\boldsymbol{\psi}^T(x_i) \tag{21}$$

Considering $\mathbf{v}_k$ and λ_k as the eigenvector and eigenvalues of the matrix **C**, we have:

$$\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\psi}(x_i)\boldsymbol{\psi}^T(x_i)\mathbf{v}_k=\lambda_k\mathbf{v}_k \tag{22}$$

The eigenvectors can be expressed as:

$$\mathbf{v}_k=\sum_{i=1}^{n}a_{ki}\boldsymbol{\psi}(x_i) \tag{23}$$

where a_{ki} (*i* = 1, 2, ..., *N*) are unknown parameters. Consider the following kernel function:

$$\kappa(x_i,x_j)=\boldsymbol{\psi}(x_i)\boldsymbol{\psi}^T(x_j) \tag{24}$$

This allows us to reformulate the problem in terms of the kernel matrix **K**, leading to:

$$\mathbf{K}^2\mathbf{a}_k=\lambda_k n\mathbf{K}\mathbf{a}_k \tag{25}$$

where $K_{ij}=\kappa(x_i,x_j)$, $\mathbf{a}_k=\begin{bmatrix}a_{k1} & a_{k2} & \ldots & a_{kn}\end{bmatrix}^T$.

For a feature set $F=\{f_1,f_2,...,f_m\}$, and $\mathbf{w}$ as a spare random vector, a Gaussian kernel function is used to measure similarity between data points:

$$\kappa(x_i,x_j)=\exp\left(-\frac{\left(\left\|x_i-x_j\right\|_F^2\right)}{\sigma}\right) \tag{26}$$

To select the most appropriate features, we rewrite (26) as:

$$\kappa(x_i,x_j)=\exp\left(-\frac{\left(\mathbf{w}^T x_i-\mathbf{w}^T x_j\right)^2}{\sigma}\right) \tag{27}$$

For each feature in each iteration, a spare random vector $\mathbf{w}_b$ is generated and is converted to $\mathbf{w}_b^*$:

$$\mathbf{w}_b^*=\begin{cases}0 & \text{if } w_b[f]\neq 0\\ 1 & \text{else}\end{cases} \tag{28}$$

Using the kernel function, $\mathbf{w}_b$ and $\mathbf{w}_b^*$ are used to determine $\hat{\mathbf{X}}_b$ and $\hat{\mathbf{X}}_b^f$ as estimation of **X**, respectively. Finally, for each feature *f*, an importance measure h_f is calculated as follows:

$$h_f=\sum_{i=1}^{n_b}\frac{\left\|\hat{\mathbf{X}}_b-\hat{\mathbf{X}}_b^f\right\|}{n_b} \tag{29}$$

C. Dempster-Shefer fusion rule

Consider the detection framework as follows:

$$\Theta=\{w_1,w_2,...,w_n\}\begin{bmatrix}\mu_1^2\\ \mu_2^2\end{bmatrix}$$
$$=\begin{bmatrix}\mu_1^1+\mu_3^1 & 0\\ 0 & \mu_2^1+\mu_3^1\end{bmatrix}\begin{bmatrix}m_1^2\\ m_2^2\end{bmatrix}+\begin{bmatrix}\mu_1^1 m_3^2\\ \mu_2^1 m_3^2\end{bmatrix} \tag{30}$$

where *w* represents a class in the pattern space. Suppose that 2^Θ can be expressed as a set consisting of 2^n propositions in the detection framework:

$$2^\Theta=\{\Phi,H_1,H_2,...,H_n,...,\{H_1\cup H_2\cup,...,H_n\}\} \tag{31}$$

where *Φ* denotes the empty set and every proposition in a subset of Θ. The Basic Probability Assignment (BPA) in the set 2^Θ is called the Basic Belief Assignment (BBA) and is described by the function *m*: $2^\Theta\longrightarrow[0,1]$. It must satisfy the following conditions [43, 44]:

$$\begin{cases}m(\Phi)=0\\ \sum_{A\subset\Theta}m(A)=1\end{cases} \tag{32}$$

where *A* is a proposition in the set 2^Θ containing one or more hypotheses. Furthermore, *m*(*A*) represents the prior probability for proposition *A*. The belief function and the plausibility function, abbreviated as *Bel* and *Pl*, are defined in the set 2^Θ as follows:

$$\begin{cases}Bel(A)=\sum_{A\subset\Theta}m(A)\\ Pl(A)=\sum_{B\cap A\neq\Theta}m(B)\end{cases} \tag{33}$$

where *A* and *B* are propositions in the set 2^Θ. *Bel*(*A*) provides the minimum measure of uncertainty for proposition *A*, while *Pl* provides the maximum measure of uncertainty for proposition *A*. Considering $\bar{A}$ as the complement set of proposition *A*, the relationship between *Bel*(*A*) and *Pl*(*A*) is defined as follows:

$$\begin{cases}Bel(A)\leq Pl(A)\\ Pl(A)=1-Bel(\bar{A})\end{cases} \tag{34}$$

The interval [*Bel*(*A*), *Pl*(*A*)] defines the degree of uncertainty or imprecision in DS theory. Suppose $\{m_1, m_2, ..., m_n\}$ represents *n* independent BBA functions from *n* different sensors in the same detection framework. The DS fusion rule is described as follows:

$$m = m_1 \oplus m_2 \oplus ... \oplus m_n \tag{35}$$

The combination in (22) can be defined for any two arbitrary *BBA* functions:

$$m_{ij}(A) = \frac{1}{K-1} \sum_{A_i \cap A_j} m_i(A_i) m_j(A_j) \tag{36}$$

where *K* is the general discount factor and represents the total conflict between the evidence m_i and m_j. Suppose $\Omega = \{c_1, c_2\}$ denotes healthy and faulty classes. The goal is to assign patterns $x \in R^p$ to one of two classes. A vector x_i that is sufficiently close to pattern *x* in class c_j is a BBA as $m_i(c_j),\ j = 1,2$. The greater the distance between *x* and x_i, the lower the belief that *x* belongs to class c_j. Therefore, the *BPA* can be expressed as a Gaussian function based on the distance d_i between *x* and x_i as follows

$$m^i(w_j) = a^i \exp\left(-\frac{(x-\mu^i)}{2\sigma_i^2}\right) \tag{37}$$

where μ_i is the center and σ_i is the width of the Gaussian functions. a^i is the correction factor with a value between zero and one. Therefore, the equations governing the DS can be expressed as

$$\mu^1 = m^1 \ ,...,\ \mu^n = \mu^{n-1} \oplus m^n \tag{38}$$

D. BNN mathematics

The first fully connected layer consists of k neurons, each with Bayesian weights. These weights and biases are modelled as Gaussian distributions. i.e. $\mathbf{W}^{(1)} \sim N(\mu_w^{(1)}, \sigma_w^{(1)})$, and $\mathbf{b}^{(1)} \sim N(\mu_b^{(1)}, \sigma_b^{(1)})$.

The allows the network to capture the uncertainty inherent in the data [15]. The output of the first layer, denoted by $\mathbf{z}^{(1)}$, is computed as:

$$\mathbf{z}^{(1)} = \mathbf{W}^{(1)}.\mathbf{x} + \mathbf{b}^{(1)} \tag{39}$$

Following this, Rectified Linear Unit (ReLU) as a nonlinear activation function $\phi(.)$ is applied to $\mathbf{z}^{(1)}$ to produce the activated outputs $\mathbf{a}^{(1)}$. Since $\mathbf{z}^{(1)}$ is a random variable due to the Bayesian nature of the weights, $\mathbf{a}^{(1)}$ also represents a distribution over activations rather than a fixed value. To prevent overfitting and enhance the generalization capabilities of the network, a dropout layer is applied after the first fully connected layer. During training, the layer deactivates a randomly selected 50% of the input units by assigning them zero. This regularization technique helps in mitigating the over-reliance on particular neurons. Mathematically, the dropout operation is represented as:

$$\mathbf{a}_{\text{dropout}}^{(1)} = \mathbf{q} \odot \mathbf{a}^{(1)} \tag{40}$$

where $\mathbf{q}$ is a binary mask vector sampled from a Bernoulli distribution with probability $p = 0.5$, and $\odot$ denotes element-wise multiplication. The second fully connected layer, which also employs Bayesian weights, consists of two neurons corresponding to the two output classes.

The weights $\mathbf{W}^{(2)}$ and biases $\mathbf{b}^{(2)}$ are drawn from Gaussian distributions, i.e., $\mathbf{W}^{(2)} \sim N(\mu_w^{(2)}, \sigma_w^{(2)}),\ \mathbf{b}^{(2)} \sim N(\mu_b^{(2)}, \sigma_b^{(2)})$. The output of this layer, $\mathbf{z}^{(2)}$ is given below:

$$\mathbf{z}^{(2)} = \mathbf{W}^{(2)}.\mathbf{a}_{\text{dropout}}^{(1)} + \mathbf{b}^{(2)} \tag{41}$$

To convert these logits into a probability distribution, the SoftMax function is applied. This function normalizes the logits so that they sum to one, allowing them to be interpreted as probabilities. The SoftMax output for each class c is given by:

$$p(y = c | x, W) = \frac{e^{z_c^{(2)}}}{e^{z_1^{(2)}} + e^{z_2^{(2)}}},\ c \in \{1,\ 2\} \tag{42}$$

where $z_c^{(2)}$ represents the logit for class c. The final classification decision is made by selecting the class with the highest probability, determined by the argmax operation:

$$\hat{y} = \arg\max_c p(y = c | x,\ W) \tag{43}$$

Training the BNN involves optimizing the model to minimize the negative evidence lower bound (ELBO) as a common objective function in variational inference. The ELBO consists the expected negative log-likelihood of the data under the approximate posterior and the Kullback-Leibler (KL) divergence between the approximate posterior distribution $q(W)$ and the prior distribution $p(W)$ [45]:

$$\begin{aligned} L(W) &= E_{q(W)}\left[-\log p(y|x,W)\right] \\ &+ KL(q(W) \| p(W)) \end{aligned} \tag{44}$$

The first term represents the cross-entropy loss where E denotes the expectation. The second term ensures that the learned weight distributions do not deviate significantly from the prior. The notation $KL(p \| q)$ represents the KL divergence of the probability distribution q from the probability distribution q. For optimization, we employ Stochastic Gradient Descent with Momentum, a variant of gradient descent that accelerates convergence by considering the previous gradients. The training is conducted over a maximum of 500 epochs with a mini-batch size of 32, which allows for efficient updates and effective use of computational resources.

