INTERNATIONAL JOURNAL OF AUTOMOTIVE AND MECHANICAL ENGINEERING
ISSN: 2229-8649 e-ISSN: 2180-1606
VOL. 21, ISSUE 1, 2024, 11074 – 11084
DOI: https://doi.org/10.15282/ijame.21.1.2024.11.0857

RESEARCH ARTICLE

A Comparative Study and Improved Bearing Fault Classifier Using Raw Vibration Data Under Limited Training Samples

J.S. Yap*, M.H. Lim, and M. Salman Leong

Institute of Noise and Vibration, Universiti Teknologi Malaysia, Kuala Lumpur 54100, Malaysia

ABSTRACT – Artificial intelligence is gaining traction in bearing fault detection and diagnosis. Generally, signal processing and feature selection are carried out to facilitate the fault classification process; however, classification accuracy tends to degrade under limited training data. In this paper, various artificial intelligence (AI) classification models are studied and compared using raw vibration data without signal processing and feature engineering. A Cosine k-Nearest Neighbours (CosKNN)-based classification model is optimized by integrating a Segmentive Mechanism, resulting in an overall classification F1-score improvement to 90.8% compared to the original classifier's 76.9%. The comparative findings show that the proposed model is suitable for circumstances with limited availability of training data, signal processing tools, and feature engineering tuning.

ARTICLE HISTORY

Received	:	18th Sept. 2023
Revised	:	30th Dec. 2023
Accepted	:	25th Jan. 2024
Published	:	20th Mar. 2024

KEYWORDS

Bearing fault
Fault detection & diagnosis
Vibration signals
Small data
Cosine KNN

1.0 INTRODUCTION

Bearings are commonly used in rotating machinery such as electrical motors, pumps, engines, and fans. They function as crucial components, enabling rotational motion by reducing friction loss, subsequently improving energy consumption, preventing overheating, and minimizing wear and tear effects [1], [2]. Among the many types of bearings, Rolling Element Bearings (REBs) are considerably common, consisting of four major elements: outer race, inner race, cage and ball [3], [4].

Bearing faults are the main reason for machinery failure, according to literature statistics, ranging from 30% to 55% according to various studies [3]-[5]. Bearing faults can result in machinery breakdowns, unexpected maintenance, inefficient operations, and extreme noise and vibrations, subsequently causing economic losses and even risking the safety and health of humans[5], [6]. Therefore, Fault Detection and Diagnosis (FDD) is crucial to determine the state of health of the REB [3], [6].

In this paper, the performance of different Artificial Intelligence (AI)-assisted FDD models was evaluated by analyzing raw vibration signals, as well as under limited training data. An improved model was developed to tackle the challenges of classification under the scarcity of signal processing, feature engineering, and training data.

FDD is applicable in different maintenance strategies, namely Breakdown, Preventive, Predictive, and Proactive Maintenance [7]. FDD in Predictive and Proactive Maintenance, including Condition-Based Monitoring (CBM), is potentially beneficial for detecting bearing faults in advance, allowing for in-time maintenance planning and rectification to minimize the mentioned machinery failure consequences [7], [8]. A simplified representation of the steps in FDD is presented in Figure 1.Figure 1.

1.1 Data Acquisition

Mainstream data acquisition of bearing information, in the form of vibration, is performed during FDD, although other parameters such as motor current, acoustic emission, temperature, and oil analysis can be used to evaluate the bearing conditions [3], [8]. In general, an accelerometer is placed in close proximity to the REB, and raw vibration data in the time series of the interested oscillatory motion body is recorded [1]. With the advancement of computational resources, Artificial Intelligence (AI) is utilized in FDD, either replacing or assisting maintenance professionals [9], [10].

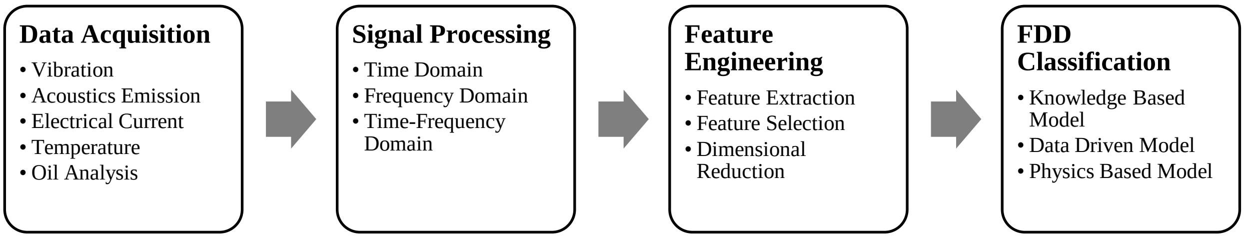

Figure 1. The simplified steps of Fault Detection and Diagnosis (FDD)

***CORRESPONDING AUTHOR | Yap Jee Siang** | ✉ yapjeesiang@gmail.com

1.2 Signal Processing

Raw vibration data typically undergoes signal processing to yield useful or visually presentable forms [1], [3] as illustrated in Table 1. These forms are mainly categorized into three domains: Time, Frequency, and Time-Frequency, where unwanted signals or noise are filtered [10]. It is worth noting that performing FDD without signal processing on raw vibration signals has become possible due to recent advancements in AI. This brings benefits such as more efficient data storage, computational resources, and reduced dependency on expert knowledge inputs, all while maintaining FDD classification accuracy over 90% [6], [11], [12].

Table 1. Signal processing method for vibration data of rolling element bearing (REB)[1], [13], [14]

Domain	Example	Benefits	Drawbacks
Time	Averaging Peak hold Threshold Trending Log-scale	Requires relatively less computational resources.	Unable to detect REB element passing frequency. Limitation for detecting early fault generation.
Frequency	Fast Fourier Transform (FFT) Envelope Spectrum Power Spectrum Cepstrum	Suitable to detect REB element passing frequency.	Less sensitive to non-stationary faults. Requires stable & continuous data (for FFT). Requires proper knowledge in windowing selection.
Time-Frequency	Short-time Fourier Transform (STFT) Empirical Model Decomposition (EMD) Continuous Wavelet Transform (CWT)	More sensitive to non-stationary faults	Requires relatively more computational resources. Requires proper knowledge in windowing selection (STFT).

1.3 Feature Engineering

Traditionally, raw data and processed signals are considered high-dimensional and require feature engineering to obtain discriminative attributes for FDD [3], [8]. Feature engineering, including the extraction and selection of useful features, presents a challenge due to the necessity of prior knowledge, and traditional AI is incapable of processing large and high-dimensional features in fault classification [5], [10]. Statistical feature extraction, such as Root-Mean-Square (RMS), Kurtosis, Crest Factor, and Entropy, is conventionally applied in FDD [1], [6], however, a high level of expertise input is not exempted in this method [5]. AI is used as a Supervised Learning Algorithm for Feature Selection, namely Filter-based, Embedded-based, and Wrapper-based methods [15]. Wrapper-based and Embedded-based methods are found to be computationally inefficient, while Filter-based methods require additional redundancy analysis mechanisms [15].

1.4 FDD Classification

Traditionally, the extracted or selected features are reviewed using a manual approach (human-machine interaction), including association, reasoning, and decision-making techniques[4], [8]. A subset of AI known as Machine Learning (ML), such as Artificial Neural Networks (ANN), Support Vector Machine (SVM), and Decision Trees, provides an alternative to manual approaches [1], [4], [8], [10]. Generally, a model-based classifier is developed using an ML technique by training with labeled features [1], [8]. Provided with proper signal processing and feature engineering, the majority of ML approaches are capable of achieving satisfactory classification accuracy above 90% [4].

2.0 DEEP LEARNING IN FDD

Deep Learning (DL) is a subset of AI with automated feature learning capability, in contrast to classical ML, further reducing human or machine-assisted tuning as illustrated in Figure 2 [3], [8], [10]. The DL's advantage over ML is due to the rise of Big Data, Internet of Things (IoT), Wireless Sensor Networks (WSN), and Computer Processors [1], [6], [16]. Conversely, DL, in general, struggles in situations with limited training data (especially faulty data) and computational hardware due to financial and energy efficiency limitations [3], [4], [6].

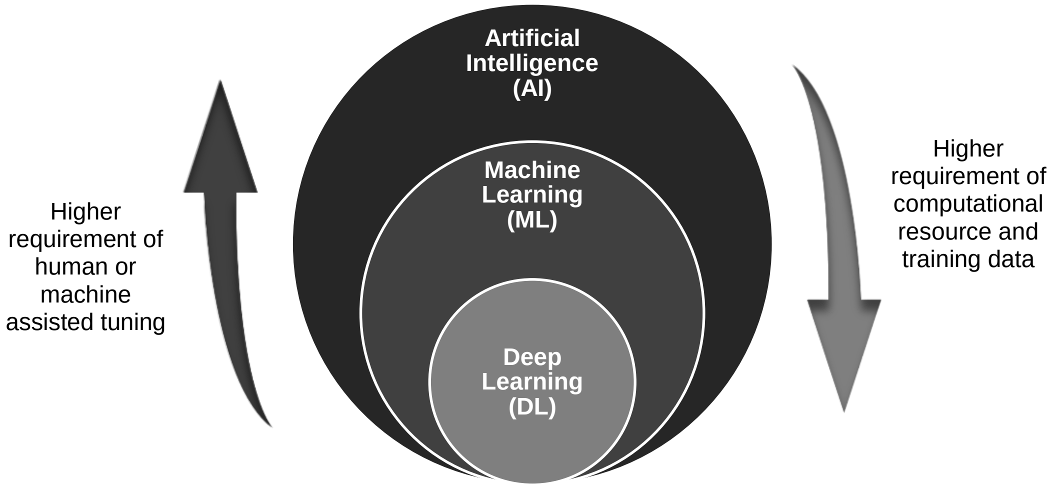

Figure 2. Relationship between AI, ML and DL

2.1 Limited Training Data in DL

Limited data, especially for REB faulty conditions, is due to the fact that faulty data generally takes a lengthy time to develop. Additionally, industrial systems are usually forced to stop for repair to avoid amplified damage [4], [17]. Real-world data is also imbalanced, with healthy data being much more substantial than faulty data. This imbalance causes the trained AI model to be sensitive to healthy data but less sensitive to faulty data [3], [18]. To tackle these shortcomings, various approaches were studied, including few-shot learning and extreme learning [1], [3], [6]. Numerous publications by researchers have used Case Western Reserve University (CWRU) datasets in FDD [3], [6], and recent studies addressing limited data are listed in Table 2. Based on the recent studies in Table 2, classification accuracies were found to be adequately high; however, most approaches still require signal processing efforts [3], [4], [6].

2.2 Computational Resources in DL

Signal processing and A.I. model training are associated with expensive data acquisition boards and complex software, increasing the cost of the condition monitoring system [19], [20]. Erica et al. explored a low-cost and small-size alternative by measuring acoustic noise and sampling it with a microcontroller. However, signal processing, such as Fast Fourier Transform, and top-flat windowing, were inevitable in the system [19]. Thani et al. proposed an A.I. model that analyzes raw vibration data using Auto-Encoder as a feature extractor. A classification of 90.3% (F1-score) was achieved; however, the model training time was longer [21].

Table 2. Recent study of limited training data deep learning in FDD

Reference	[18]	[22]	[23]	[24]	[25]	[26]
Training Data Sample						
Normal	2828	400	30	10	10	60
Outer Race Fault	243	12	30	10	10	60
Inner Race Fault	323	12	30	10	10	60
Ball Fault	323	12	30	10	10	60
Signal Processing Method						
Domain	Time-Frequency	Frequency	Time	Time	Frequency	Frequency
Signal Processing	Wavelet Time-Frequency Diagram	Spectrum	Normalized & 2D Imaging	Raw	FFT	FFT

Table 2. (cont.)

Reference	[18]	[22]	[23]	[24]	[25]	[26]
Deep Learning Method						
Integrated Feature Learning	Wide First-layer Kernels Convolutional Neural Networks (WDCNN)	Stacked Capsule Auto-encoder (CaAE)	Conditional Generative Adversarial Network (CGAN)	Generative Adversarial Network (GAN) & Attention-Weighted Multidepth Feature Fusion	Statistical Similarity Measurement (SSM)	Varying Coefficient Transfer Learning (VCTL) & Bayesian Network (BN)
Integrated Classifier			2D Convolutional Neural Networks (2D-CNN)	Softmax Classifier		
Classification Accuracy (%)	94.72	98.28	97.14	99.14	100	90.30

3.0 RESEARCH METHODOLOGY

This research is partitioned into 2 core parts: (A) Investigate and compare the classification performance of various AI methods without signal processing, feature engineering, and knowledge-based tuning, as illustrated in Figure 3; (B) Develop and compare an optimized AI model with improved classification performance.

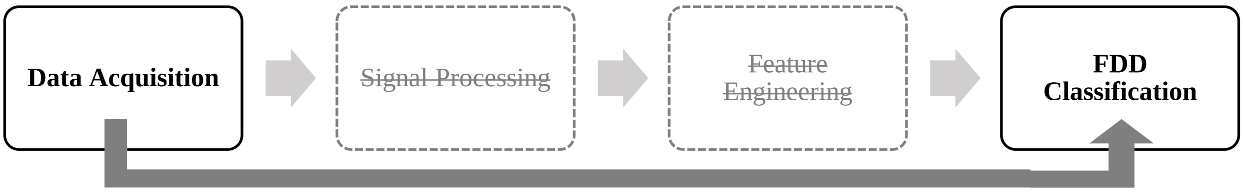

Figure 3. Proposed comparative study of AI methods without Signal Processing and Feature Engineering

3.1 Datasets Description

Case Western Reserve University Bearing Data Center (CWRU)[27] and a purchased Spectaquest Machinery Fault Simulator (MFS) are used in this study. The description of the datasets is shown in Figure 4 and Table 3. The Rolling Element Bearing (REB) faults were artificially seeded with electro-discharge machining (EDM), and the vibration signals were collected by a 16-channel DAT recorder with accelerometers at 12,000 samples per second for CWRU, while the MFS signals were acquired with a 16-channel OROS recorder with a Wilcoxon 100mV/g accelerometer at 8,000 samples per second.

Table 3. REB Datasets descriptions

Dataset	CWRU		MFS	
Bearing Model	6205-2RS JEM SKF, deep groove ball bearing @ Drive End		Spectraquest custom bearing	
No. of Balls	9		8	
Motor Load (HP)	0		n.a.	
Shaft Speed (RPM)	1,797		1,800	
Data Acquisition	Vibration			
Sampling Rate (kHz)	12		8	
Data Points per Sample	1,024			
Condition	Fault Size (inch)	No. of Samples	Fault Size (inch)	No. of Samples
Normal	n.a.	300	n.a.	300
Outer Race Fault	0.007	100	0.03	100
	0.014	100	0.06	100
	0.021	100	0.09	100

Table 3. (cont.)

Dataset	CWRU		MFS	
Inner Race Fault	0.007	100	0.03	100
	0.014	100	0.06	100
	0.021	100	0.09	100
Ball Fault	0.007	100	0.03	100
	0.014	100	0.06	100
	0.021	100	0.09	100

Figure 4. REB datasets: CWRU (left); MFS (right)

The datasets are split into training and testing samples, with each sample consisting of 1,024 data points of raw vibration signals. Different scenarios of AI model training were carried out, identified as T600, T480, T360, and T240, corresponding to total samples of 600, 480, 360, and 240 used as training input. These samples were labeled into 4 classes: Normal, Outer Race Fault, Inner Race Fault, and Ball Fault. The fault sizes are artificially seeded using electro-discharge machining (EDM) and distributed evenly in the mentioned scenarios. For example, in T360 of CWRU, the Inner Race Fault consists of 30 samples for 0.007, 0.014, and 0.021 inches, respectively, totaling 90 samples. For a consistent comparison result, the testing samples are identical and consist of 150 samples for each class. The details are described in Table 4.

Table 4. Description of training and testing samples for CWRU and MFS datasets

Scenario	T600	T480	T360	T240
Training Samples				
Normal	150	120	90	60
Outer Race Fault	150	120	90	60
Inner Race Fault	150	120	90	60
Ball Fault	150	120	90	60
Total	600	480	360	240
Testing Samples				
Normal	150	150	150	150
Outer Race Fault	150	150	150	150
Inner Race Fault	150	150	150	150
Ball Fault	150	150	150	150
Total	600	600	600	600

4.0 PROPOSED COMPARISON OF AI METHODS

MATLAB’s Classification Learner App offers multiple and simplified classification functions, including Support Vector Machine (SVM), k-Nearest Neighbors (KNN), and Artificial Neural Networks (ANN). Training data can be fed into the program to train different models for evaluation and can be exported to real-world applications (reference: matworks.com/help/stats/classification-learner-app.html). In this study, a total of 31 models were built with default presets and hyperparameters to compare and evaluate the models’ performance. The type of classifier/model and its default hyperparameters are tabulated in Table 5.

Table 5. Type of classifier and corresponding default hyperparameter of MATLAB's Classification Learner App

Type of Classifier	Hyperparameters
Fine Tree (FT)	Maximum split no.: 100 Split criterion: Gini's diversity index Surrogate decision splits: Off
Medium Tree (MT)	Maximum split no.: 20 Split criterion: Gini's diversity index Surrogate decision splits: Off
Coarse Tree (CT)	Maximum split so.: 4 Split criterion: Gini's diversity index Surrogate decision splits: Off
Linear Discriminant (LD)	Covariance structure: Full
Quadratic Discriminant (QD)	Covariance structure: Diagonal
Gaussian Naïve Bayes (GNB)	Distribution name for numerical Predictor: Gaussian
Kernel Naïve Bayes (KNB)	Distribution name for numerical Predictor: Kernel Kernel type: Gaussian Support: Unbounded
Linear SVM (LSVM)	Kernel function: Linear Kernel scale: Automatic Box constraint: 1 Multiclass method: One-vs-One
Quadratic SVM (QSVM)	Kernel function: Quadratic Kernel scale: Automatic Box constraint: 1 Multiclass method: One-vs-One
Cubic SVM (CUBSVM)	Kernel function: Cubic Kernel scale: Automatic Box constraint: 1 Multiclass method: One-vs-One
Fine Gaussian SVM (FGSVM)	Kernel function: Gaussian Kernel scale: 8 Box constraint: 1 Multiclass method: One-vs-One
Medium Gaussian SVM (MGSVM)	Kernel function: Gaussian Kernel scale: 32 Box constraint: 1 Multiclass method: One-vs-One
Coarse Gaussian SVM (CGSVM)	Kernel function: Gaussian Kernel scale: 130 Box constraint: 1 Multiclass method: One-vs-One
Fine KNN (FKNN)	Number of neighbors: 1 Distance metric: Euclidean Distance weight: Equal
MEDIUM KNN (MKNN)	Number of neighbors: 10 Distance metric: Euclidean Distance weight: Equal
COARSE KNN (CKNN)	Number of neighbours: 100 Distance metric: Euclidean Distance weight: Equal
Cosine KNN (CosKNN)	Number of neighbors: 10 Distance metric: Cosine Distance weight: Equal
Cubic KNN (CubKNN)	Number of neighbors: 10 Distance metric: Minkowski Distance weight: Equal
Weighted KNN (WKNN)	Number of neighbors: 10 Distance metric: Euclidean Distance weight: Squared inverse
Ensemble Boosted Trees (EBooT)	Ensemble method: AdaBoost Learner type: Decision tree Maximum split no.: 20 Learners no.: 30
Ensemble Bagged Trees (EBagT)	Ensemble method: Bag Learner type: Decision tree Maximum split no.: 599 Learners no.: 30
Ensemble Subspace Discriminant (ESubD)	Ensemble method: Subspace Learner type: Discriminant Learners no.: 30 Subspace dimension:512
Ensemble Subspace KNN (ESubKNN)	Ensemble method: Subspace Learner type: Nearest Neighbours Learners no.: 30 Subspace dimension:512
Ensemble RUSBoosted Trees (ERUST)	Ensemble method: RUSBoost Learner type: Decision tree Maximum split no.: 20 Learners no.: 30
Narrow Neural Network (NNN)	Fully connected layers no.: 1 First layer size: 10 Activation: ReLU
Medium Neural Network (MNN)	Fully connected layers no.: 1 First layer size: 25 Activation: ReLU
Wide Neural Network (WNN)	Fully connected layers no.: 1 First layer size: 100 Activation: ReLU
Bilayered Neural Network (BNN)	Fully connected layers no.: 2 First layer size: 10 Second layer size: 10 Activation: ReLU
Trilayered Neural Network (TNN)	Fully connected layers no.: 3 First layer size: 10 Second layer size: 10 Second layer size: 10 Activation: ReLU
SVM Kernel (SVMK)	Learner: SVM Expansion dimensions no.: Auto Kernel scale: Auto Multiclass method: One-vs-One
Logistic Regression Kernel (LRK)	Learner: Logistic Regression Expansion dimensions no.: Auto Kernel scale: Auto Multiclass method: One-vs-One

4.1 Quantifications of AI Methods' Performance

The key measurements of AI Methods are classification accuracy as follows[28]:

$$Precision\ (\%) = \frac{TP}{TP + FP}\ X\ 100$$

$$Recall\ (\%) = \frac{TP}{TP + FN}\ X\ 100$$

$$F1 - score\ (\%) = 2\frac{[Precision\ x\ Recall]}{[Precision + Recall]} \tag{1}$$

where

TP = number of true positive classifications
TN = number of true negative classifications
FP = number of false positive classifications
FN = number of false negative classifications

F1- is selected as the metric for classification performance because it balances precision and recall compared to accuracy. his metric is particularly useful when the testing data are imbalanced [29]. Since there are 2 datasets in the comparative analysis, the averaging of metrics for both CWRU and MFS is carried out. Additionally, to evaluate the performance of different scenarios of limited training data, an overall indicator is calculated by averaging the accuracy of scenarios T600, T480, T360, and T240.

4.2 Results and Discussion of AI Methods

Table 6 tabulates the comparative result of classification accuracy using different AI methods. In general, the majority of the overall F1-score is below 75%, demonstrating the challenges of classifiers without the setting of signal processing and feature engineering. CosKNN and MGSVM established slightly higher overall accuracy of 76.9% and 75.5%, respectively. The F1-scores were observed to be trending down with lesser training data, MGSVM saw a reduction of 6% from 78.6% (T600) to 72.6% (T240), while CosKNN reduced by 4.4% from 81.4% (T600) to 77.0% (T240). The reason for the greater degradation of accuracy in CosKNN is believed to be the default setting of the nearest neighbors' number (k) of 10, which is in a higher ratio compared to the sample number, resulting in the k-number being filled by true negative (TN) samples when insufficient true positive (TP) samples are available. Although the setting can be adjusted to lower the k-number, this would require prior knowledge before developing the classification model.

Table 6. Comparative result (*F1-score*) in relation to the scenario of limited training data

Data-Sets	AI Method	T600	T480	T360	T240	Overall	AI Method	T600	T480	T360	T240	Overall
CWRU	FT	40.7%	37.2%	38.6%	35.4%	38.0%	CosKNN	83.7%	79.7%	76.1%	73.0%	78.1%
MFS		37.6%	37.8%	37.8%	38.2%	37.8%		79.1%	74.1%	68.7%	80.9%	75.7%
Average		39.1%	37.5%	38.2%	36.8%	37.9%		81.4%	76.9%	72.4%	77.0%	76.9%
CWRU	MT	34.1%	34.0%	38.0%	33.4%	34.9%	CubKNN	17.0%	16.4%	15.6%	15.3%	16.1%
MFS		31.2%	30.1%	27.8%	32.9%	30.5%		19.5%	16.9%	17.1%	13.2%	16.7%
Average		32.6%	32.0%	32.9%	33.1%	32.7%		18.3%	16.6%	16.3%	14.2%	16.4%
CWRU	CT	29.9%	23.7%	28.1%	33.1%	28.7%	WKNN	17.0%	20.5%	22.5%	19.5%	19.9%
MFS		23.2%	20.3%	27.8%	20.6%	23.0%		23.9%	21.7%	20.1%	15.9%	20.4%
Average		26.6%	22.0%	27.9%	26.8%	25.8%		20.5%	21.1%	21.3%	17.7%	20.1%
CWRU	LD	28.9%	29.5%	29.8%	29.5%	29.4%	EBooT	47.4%	42.8%	47.6%	43.1%	45.2%
MFS		32.3%	35.0%	32.5%	31.8%	32.9%		43.0%	45.1%	40.9%	44.3%	43.3%
Average		30.6%	32.3%	31.1%	30.6%	31.2%		45.2%	44.0%	44.2%	43.7%	44.3%
CWRU	QD	70.2%	69.9%	70.4%	71.3%	70.4%	EBagT	53.5%	53.2%	49.7%	42.2%	49.7%
MFS		52.8%	50.9%	49.8%	49.8%	50.8%		61.2%	61.2%	56.2%	46.7%	56.3%
Average		61.5%	60.4%	60.1%	60.5%	60.6%		57.4%	57.2%	52.9%	44.4%	53.0%
CWRU	GNB	70.2%	69.9%	70.4%	71.3%	70.4%	ESubD	29.4%	31.5%	33.4%	30.1%	31.1%
MFS		52.8%	50.9%	49.8%	49.8%	50.8%		36.6%	51.8%	37.2%	37.7%	40.8%
Average		61.5%	60.4%	60.1%	60.5%	60.6%		33.0%	41.7%	35.3%	33.9%	36.0%
CWRU	KNB	63.7%	64.9%	65.5%	67.2%	65.3%	ESubKNN	31.8%	31.4%	25.6%	23.7%	28.2%
MFS		52.1%	52.5%	50.7%	50.3%	51.4%		50.2%	46.0%	41.5%	37.0%	43.7%
Average		57.9%	58.7%	58.1%	58.7%	58.3%		41.0%	38.7%	33.6%	30.4%	35.9%

Table 6. (cont.)

Data-Sets	AI Method	T600	T480	T360	T240	Overall	AI Method	T600	T480	T360	T240	Overall
CWRU	LSVM	28.7%	25.8%	22.5%	22.6%	24.9%	ERUST	34.1%	26.4%	36.5%	36.1%	33.2%
MFS		30.6%	29.1%	27.0%	27.0%	28.4%		30.0%	28.6%	26.8%	28.3%	28.5%
Average		29.6%	27.4%	24.8%	24.8%	26.7%		32.0%	27.5%	31.7%	32.2%	30.8%
CWRU	QSVM	43.3%	37.1%	36.3%	32.7%	37.3%	NNN	51.3%	38.9%	42.2%	31.5%	41.0%
MFS		58.2%	52.4%	46.8%	41.2%	49.7%		57.0%	56.0%	26.8%	41.0%	45.2%
Average		50.8%	44.8%	41.5%	36.9%	43.5%		54.2%	47.5%	34.5%	36.3%	43.1%
CWRU	CubSVM	40.4%	33.9%	32.6%	30.5%	34.3%	MNN	54.7%	45.4%	42.1%	33.2%	43.8%
MFS		59.7%	52.4%	45.1%	39.3%	49.1%		59.3%	62.7%	52.8%	45.4%	55.0%
Average		50.0%	43.1%	38.9%	34.9%	41.7%		57.0%	54.1%	47.5%	39.3%	49.4%
CWRU	FGSVM	25.9%	25.4%	25.8%	24.2%	25.3%	WNN	54.2%	47.6%	45.4%	34.7%	45.5%
MFS		50.1%	53.8%	52.4%	52.1%	52.1%		64.1%	54.9%	51.8%	45.2%	54.0%
Average		38.0%	39.6%	39.1%	38.2%	38.7%		59.1%	51.3%	48.6%	40.0%	49.7%
CWRU	MGSVM	80.6%	80.1%	76.7%	76.9%	78.6%	BNN	57.6%	46.2%	48.1%	35.3%	46.8%
MFS		76.7%	74.4%	70.1%	68.2%	72.3%		58.1%	51.3%	49.3%	45.5%	51.0%
Average		78.6%	77.2%	73.4%	72.6%	75.5%		57.9%	48.7%	48.7%	40.4%	48.9%
CWRU	CGSVM	34.7%	30.4%	25.6%	20.5%	27.8%	TNN	52.1%	57.6%	50.7%	41.2%	50.4%
MFS		41.5%	41.4%	38.7%	36.4%	39.5%		58.9%	57.1%	45.3%	48.9%	52.6%
Average		38.1%	35.9%	32.1%	28.5%	33.7%		55.5%	57.4%	48.0%	45.0%	51.5%
CWRU	FKNN	36.8%	35.3%	27.5%	25.8%	31.3%	SVMK	64.2%	64.7%	64.6%	58.0%	62.9%
MFS		50.3%	45.9%	40.2%	36.5%	43.2%		76.9%	73.2%	70.4%	67.1%	71.9%
Average		43.5%	40.6%	33.8%	31.1%	37.3%		70.6%	69.0%	67.5%	62.6%	67.4%
CWRU	MKNN	20.3%	20.0%	21.8%	17.1%	19.8%	LRK	59.0%	57.1%	55.1%	53.6%	56.2%
MFS		17.9%	16.9%	16.9%	13.5%	16.3%		73.9%	70.5%	63.6%	62.0%	67.5%
Average		19.1%	18.4%	19.4%	15.3%	18.0%		66.5%	63.8%	59.3%	57.8%	61.9%
CWRU	CKNN	10.0%	10.0%	10.0%	10.0%	10.0%						
MFS		20.1%	29.5%	10.5%	10.0%	17.5%						
Average		15.0%	19.7%	10.3%	10.0%	13.8%						

5.0 IMPROVEMENT OF CosKNN CLASSIFIER

The comparison showed CosKNN achieves the highest overall accuracy with limited training data, the AI method is established on cosine distance (*Dc*) between 2 vectors as follows

$$Dc\ (A, B) = 1 - Sc(A, B) \tag{2}$$

$$= 1 - \frac{A\ .\ B}{||A||\ .\ ||B||}$$

$$= 1 - \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2\ .\ \sum_{i=1}^{n} B_i^2}}$$

where,

Dc = Cosine Distance
Sc = Cosine Similarity
A = Tested Sample in Vector form
B = Trained Sample in Vector form
i = *i*-th components of Vectors

The vectors are raw signal samples, each with *i*-th numbers consisting of 1024 data points. Two vectors are categorized as the same class when the computed *Dc* value is closest to 0, and vice versa (*Dc* ≈ -1 or 1). The predicted output is based on the preset k-number without weighting of data points. The main benefit of KNN is the absence of assumptions about attributes, making it trainable using elemental procedures and local approximation [3].

A Segmentive CosKNN (SCosKNN) classification model is proposed to improve classification accuracy. A Segmentive Mechanism is integrated with a CosKNN classifier, and the data samples are segmented and arranged as shown in Figure 5. This segmentation mechanism virtually increases the number of training inputs, allowing a greater opportunity for *Dc* coordination. The trained model is packaged for testing, and a consolidated prediction is generated from segmented 5 outputs, as shown in Figure 6.

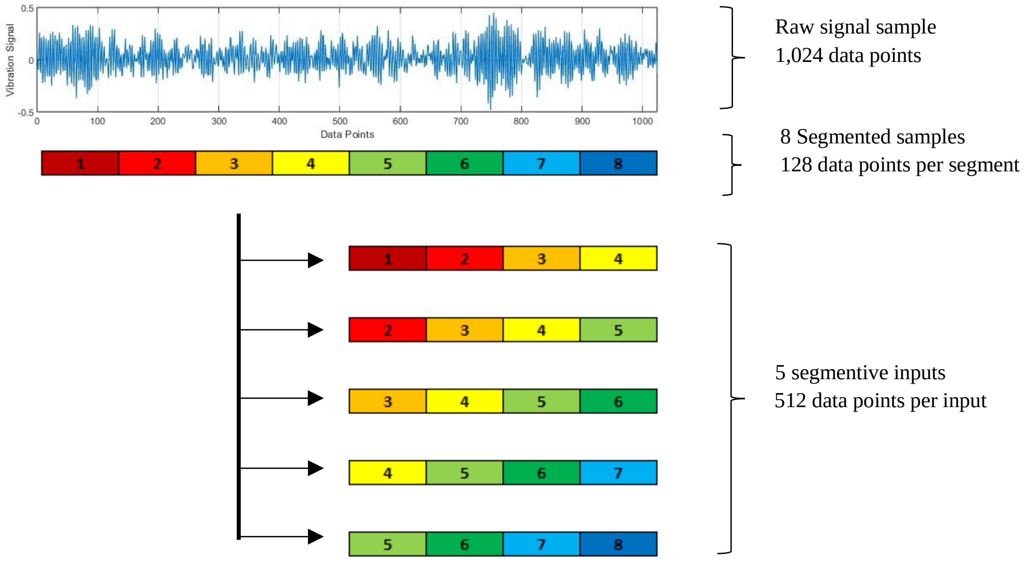

Figure 5. Segmentive mechanism of the proposed SCosKNN model

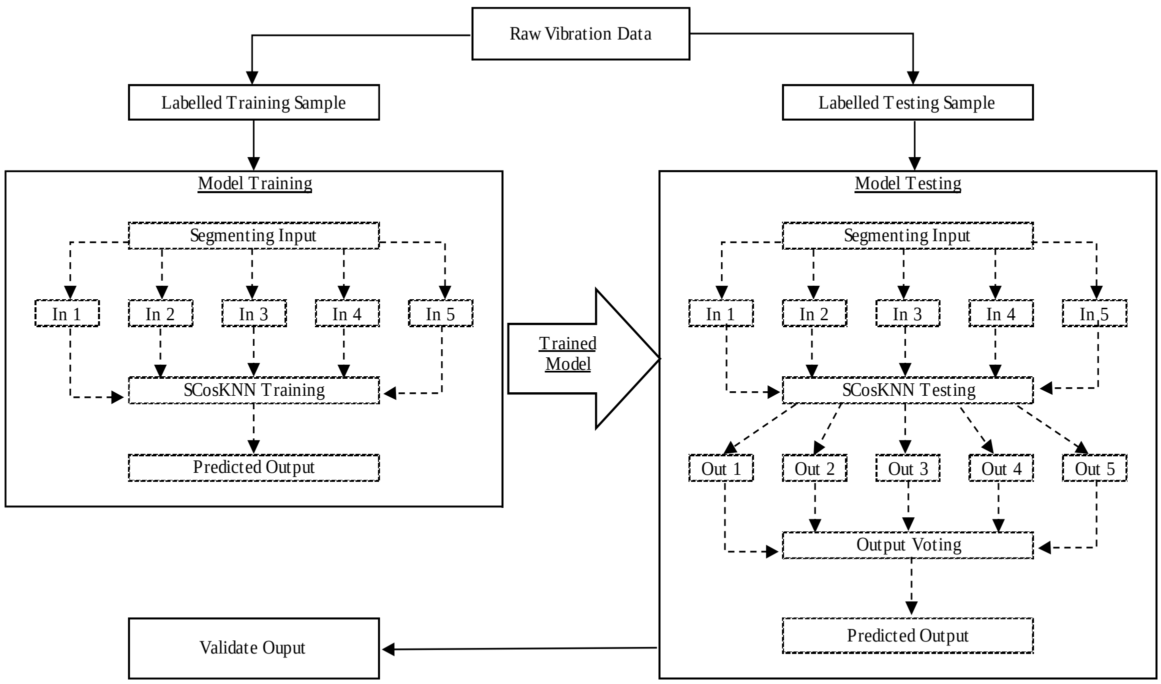

Figure 6. Framework of proposed SCosKNN model

5.1 Results and Discussion of CosKNN Classifier

The F1-score of SCosKNN is presented and compared with CosKNN in Table 7 and Figure 7. The overall accuracy of SCosKNN is 90.8%, an increase of 13.9% compared to CosKNN (76.9%).

Table 7. Classification performance of CosKNN and SCosKNN

Data-Sets	AI Method	T600	T480	T360	T240	Overall	AI Method	T600	T480	T360	T240	Overall
CWRU	CosKNN	83.7%	79.7%	76.1%	73.0%	78.1%	SCosKNN	97.5%	96.8%	96.3%	92.4%	95.8%
MFS		79.1%	74.1%	68.7%	80.9%	75.7%		89.7%	88.8%	83.7%	80.9%	85.8%
Average		81.4%	76.9%	72.4%	77.0%	76.9%		93.6%	92.8%	90.0%	86.7%	90.8%

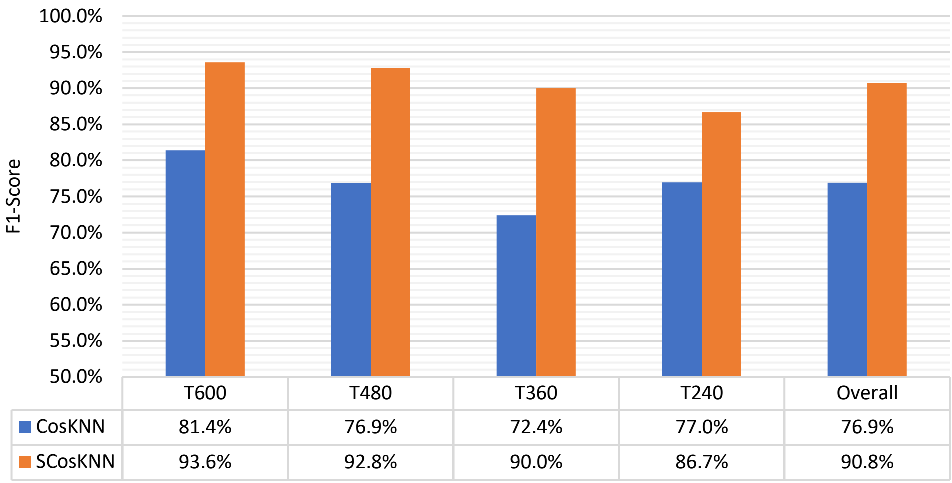

Figure 7. Classification performance of CosKNN and SCosKNN in limited training data scenarios

The Segmentive Mechanism integrated into the proposed model is capable of enhancing the classification performance as follows:

a) Virtually increases training data
For the T240 scenario, the virtual data samples are five times higher after segmentation, providing more opportunities for *Dc* coordination within the same class without adjusting the k-number.

b) Reduces input dimensionality
The data points per input are reduced by half, thereby reducing dimensionality. Referring to equation *(2)*, higher dimensions (*i*-th) potentially increase combinatorial effects.

c) Support collective multi-output
Multiple outputs from segmented input computations are produced, subsequently consolidated to establish a collective predicted output.

Despite the benefits of the proposed model, several aspects should be considered when implementing it in FDD. For example, it requires larger computational memory due to the increased virtual data size. Compared with the literature, a 91.3% *F1-score* is considered a moderate performance in FDD; however, the proposed model is potentially applicable for fault screening, where classification can be carried out without human input, signal processing, and feature engineering.

6.0 CONCLUSION

This paper presents a comparative study of Rolling Element Bearing (REB) fault classification using different AI methods without signal processing and feature engineering in scenarios of limited training data. CosKNN outperformed other AI methods with an overall *F1-score* of 76.9%; however, the overall accuracy is not considered adequate for Fault Detection and Diagnosis (FDD), especially with greater degradation in scenarios of reduced training data. An improved model, integrating the Segmentive Mechanism (SCosKNN), is proposed, achieving a higher overall accuracy of 90.8% and experiencing less degradation in scenarios of limited training data. The comparative findings suggest that the proposed model is suitable for circumstances with limited availability of training data, signal processing tools, and feature engineering tuning.

7.0 ACKNOWLEDGEMENT

The authors would like to extend their greatest gratitude to the Institute of Noise and Vibration, Universiti Teknologi Malaysia for funding the current study under the Higher Institution Centre of Excellence (HICoE) Grant Scheme of Advanced Health Monitoring for Turbomachinery (R.K130000.7843.4J227) and Health Monitoring and Integrity Assessment of Ageing Assets (R.K130000.7843.4J228).

8.0 REFERENCES

[1] A. Anwarsha and T. N. Babu, "Recent advancements of signal processing and artificial intelligence in the fault detection of rolling element bearings: a review," *Journal of Vibroengineering*, vol. 24, no. 6, pp. 1027–1055, 2022.

[2] S. Zhang, S. Zhang, B. Wang, and T. G. Habetler, "Machine Learning and Deep Learning Algorithms for Bearing Fault Diagnostics - A Comprehensive Review," *arXiv preprint*, arXiv:1901.08247, 2019.

[3] M. Hakim, A. A. B. Omran, A. N. Ahmed, M. Al-Waily, and A. Abdellatif, "A systematic review of rolling bearing fault diagnoses based on deep learning and transfer learning: Taxonomy, overview, application, open challenges, weaknesses and recommendations," *Ain Shams Engineering Journal*, vol. 14, no. 4, p. 101945, 2023.

[4] S. Zhang, S. Zhang, B. Wang, and T. G. Habetler, "Deep learning algorithms for bearing fault diagnosticsx - A Comprehensive Review," *IEEE Access*, vol. 8, pp. 29857–29881, 2020.

[5] M. Mohiuddin and M. S. Islam, "Rolling element bearing faults detection and classification technique using vibration signals," *Engineering Proceedings*, vol. 27, no. 1, p. 53, 2022.

[6] S. R. Saufi, Z. A. Bin Ahmad, M. S. Leong, and M. H. Lim, "Challenges and opportunities of deep learning models for machinery fault detection and diagnosis: A review," *IEEE Access*, vol. 7, pp. 122644–122662, 2019.

[7] S. Gawde, S. Patil, S. Kumar, and K. Kotecha, "A scoping review on multi-fault diagnosis of industrial rotating machines using multi-sensor data fusion," *Artificial Intelligence Review*, vol. 56, no. 5, pp. 4711–4764, 2023.

[8] V. Singh, P. Gangsar, R. Porwal, and A. Atulkar, "Artificial intelligence application in fault diagnostics of rotating industrial machines: a state-of-the-art review," *Journal of Intelligent Manufacturing*, vol. 34, no. 3, pp. 931–960, 2023.

[9] M. A. Gultekin and A. Bazzi, "Review of fault detection and diagnosis techniques for AC Motor Drives," *Energies*, vol. 16, no. 15, p. 5602, 2023.

[10] S. R. Saufi, M. F. Isham, Z. A. Ahmad, and M. D. A. Hasan, "Machinery fault diagnosis based on a modified hybrid deep sparse autoencoder using a raw vibration time-series signal," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 4, pp. 3827–3838, 2023.

[11] J. Zhang, Y. Sun, L. Guo, H. Gao, X. Hong, and H. Song, "A new bearing fault diagnosis method based on modified convolutional neural networks," *Chinese Journal of Aeronautics*, vol. 33, no. 2, pp. 439–447, 2020.

[12] J. He, P. Wu, Y. Tong, X. Zhang, M. Lei, and J. Gao, "Bearing fault diagnosis via improved one-dimensional multi-scale dilated CNN," *Sensors*, vol. 21, no. 21, p. 7319, 2021.

[13] D. Khan and R. Burdzik, "Measurement and analysis of transport noise and vibration: A review of techniques, case studies, and future directions," *Measurement*, vol. 220, p. 113354, 2023.

[14] S. Singh and M. Vishwakarma, "A review of vibration analysis techniques for rotating machines." *International Journal of Engineering Research and Technology*, vol. 4, pp. 757-761, 2015.

[15] D. Asir, S. Appavu, and E. Jebamalar, "Literature Review on Feature Selection Methods for High-Dimensional Data," *International Journal of Computer Applications*, vol. 136, no. 1, pp. 9–17, 2016.

[16] R. R. Swain, T. Dash, and P. M. Khilar, "Automated fault diagnosis in wireless sensor networks: A Comprehensive Survey," *Wireless Personal Communications*, vol. 127, no. 4, pp. 3211–3243, 2022.

[17] A. Zhang, S. Li, Y. Cui, W. Yang, R. Dong, and J. Hu, "Limited data rolling bearing fault diagnosis with few-shot learning," *IEEE Access*, vol. 7, pp. 110895–110904, 2019.

[18] Y. Kang, G. Chen, W. Pan, X. Wei, H. Wang, and Z. He, "A dual-experience pool deep reinforcement learning method and its application in fault diagnosis of rolling bearing with unbalanced data," *Journal of Mechanical Science and Technology*, vol. 37, no. 6, pp. 2715–2726, 2023.

[19] E. Raviola and F. Fiori, "A low-cost, small-size, and bluetooth-connected module to detect faults in rolling bearings," *Applied Sciences (Switzerland)*, vol. 10, no. 16, p. 5645, 2020.

[20] Y. Hu, Q. Xiong, Q. Zhu, et al., "Few-shot transfer learning with attention for intelligent fault diagnosis of bearing," *Journal of Mechanical Science and Technology*, vol. 36, no. 12, pp. 6181–6192, 2022.

[21] M. A. Tnani, P. Subarnaduti, and K. Diepold, "Efficient feature learning approach for raw industrial vibration data using two-stage learning framework," *Sensors*, vol. 22, no. 13, p. 4813, 2022.

[22] Z. Ren, Y. Zhu, Y. Fu, C. Zhang, K. Yan, and J. Hong, "Fault diagnosis with imbalanced data based on auto-encoder," in *Proceedings - 11th International Conference on Prognostics and System Health Management, PHM-Jinan 2020,* Institute of Electrical and Electronics Engineers Inc., pp. 97–102, 2020.

[23] J. Yang, J. Liu, J. Xie, C. Wang, and T. Ding, "Conditional GAN and 2-D CNN for bearing fault diagnosis with small samples," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-12, 2021.

[24] T. Zhang, S. He, J. Chen, T. Pan, and Z. Zhou, "Toward small sample challenge in intelligent fault diagnosis: Attention-weighted multidepth feature fusion net with signals augmentation," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, p. 3502413, 2022.

[25] I. Attoui, "Novel fast and automatic condition monitoring strategy based on small amount of labeled data," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 2, pp. 1266–1275, 2022.

[26] Y. Hou, A. Yang, W. Guo et al., "Bearing fault diagnosis under small data set condition: A Bayesian network method with transfer learning for parameter estimation," *IEEE Access*, vol. 10, pp. 35768–35783, 2022.

[27] Case School of Engineering, "Bearing Data Center," Case Western Reserve University Bearing Data Center. Accessed: Sep. 10, 2023. [Online]. Available: https://engineering.case.edu/bearingdatacenter

[28] N. F. Waziralilah, A. Abu, M. H. Lim, L. K. Quen, and A. Elfakharany, "Bearing fault diagnosis employing Gabor and augmented architecture of convolutional neural network," *Journal of Mechanical Engineering and Sciences*, vol. 13, no. 3, pp. 5689–5702, 2019.

[29] H. M and S. M.N, "A Review on evaluation metrics for data classification evaluations," *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, pp. 1–11, 2015.