Bearing fault diagnosis by machine learning and deep learning-based models: A comparative study applying for HUST bearing dataset

Nguyen Thi Hoai Thu*, Pham Nang Van, Hoang Quoc Hung

School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, 1 Dai Co Viet, Hai Ba Trung, Hanoi, Vietnam.
*Corresponding author: thu.nguyenthihoai@hust.edu.vn
Received 10 Dec. 2024; Revised 14 Mar. 2025; Accepted 09 May 2025; Published 25 May 2025.
DOI: https://doi.org/10.54939/1859-1043.j.mst.103.2025.31-39

ABSTRACT

Diagnosing bearing faults is essential for ensuring the reliability and operational safety of mechanical and electronic systems. This paper presents a comparative analysis of different machine learning-based models for classifying bearing fault conditions, including Support Vector Machines (SVM), Long Short-Term Memory (LSTM) networks, One-Dimensional Convolutional Neural Networks (1D-CNN), Two-Dimensional Convolutional Neural Networks (2D-CNN), and Transformer model. These models are applied to the HUST bearing dataset and evaluated based on their ability to accurately classify defects from vibration signal data. The results indicate that 1D-CNN, 2D-CNN, and Transformer model exhibit superior performance in bearing fault diagnosis. 1D-CNN attained 99.8% accuracy on the training set and 99.83% on the test set, followed by 2D-CNN with 99.1% and 99.3%, respectively. The Transformer model also performed well, reaching 99.7% accuracy within 1 hour of training, similar to 1D-CNN (1 hour) and 2D-CNN (0.8 hours). In contrast, LSTM and SVM exhibited lower accuracy and significantly longer training times, with LSTM requiring 11.5 hours and SVM 8 hours. These findings suggest that 1D-CNN, 2D-CNN, and the Transformer model are highly effective approaches for bearing fault diagnosis, with the Transformer model achieving performance and training efficiency comparable to CNN-based models.

Keywords: Bearing fault diagnosis; Machine learning; Deep learning; Convolutional neural network; Long short-term memory; Support vector machine; Transformer model.

1. INTRODUCTION

The growing demand for continuous industrial operations increases machine stress, leading to higher maintenance costs and safety risks [1]. Bearings, essential for reducing friction and supporting loads, are especially vulnerable, causing nearly one-third of machinery failures [2]. Harsh conditions like high humidity, temperature, and contamination accelerate wear. Early fault detection using vibration analysis, temperature monitoring, and ultrasound testing enables timely intervention, reducing downtime and repair costs while improving efficiency. This proactive approach enhances machine reliability and promotes sustainable industrial operations.

While traditional methods have significantly contributed to bearing fault diagnosis, they still face several critical limitations that must be addressed to achieve higher accuracy and robustness in practical applications [3]. One major challenge is that many existing techniques struggle to process complex vibration signals and are highly susceptible to noise, leading to unstable diagnostic performance. To overcome these issues, various artificial intelligence (AI) techniques, such as artificial neural networks (ANN) [4] and support vector machines (SVM) [5], have been employed in machine fault diagnosis. However, traditional machine learning models are limited by their shallow architectures, making it difficult to effectively learn discriminative features from raw, high-dimensional inputs [6]. In recent years, deep learning has gained increasing attention in rolling bearing fault diagnosis, with commonly used architectures including Convolutional Neural Networks (CNN) [7], Recurrent Neural Networks (RNN) [8], Long Short-Term Memory (LSTM)

[9]. Furthermore, Transformer model has recently demonstrated its potential in bearing fault diagnosis due to its ability to capture long-range dependencies in time-series data, achieving competitive performance in both accuracy and computational efficiency [10].

This study investigates the effectiveness of various machine learning and deep learning models in identifying and classifying bearing faults, including SVM, LSTM, 1D-CNN, 2D-CNN, and Transformer. The objective is to evaluate each model's accuracy, computational efficiency, and generalization ability in ball bearing fault detection. A comparative analysis was conducted using the HUST BEARING dataset, which provides a diverse set of bearing fault data suitable for benchmarking fault diagnosis models.

2. DATASET DESCRIPTION

The HUST bearing dataset was developed by Nguyen Duc Thuan and Hoang Si Hong at Hanoi University of Science and Technology [11]. It contains 99 raw vibration signals corresponding to six defect types (internal cracks, external cracks, ball cracks, and two combinations thereof) across five bearing types (6204, 6205, 6206, 6207, and 6208) under three operating conditions (0 W, 200 W, and 400 W). Each vibration signal was sampled at a rate of 51,200 samples per second over a duration of 10 seconds.

With a shaft rotating at 1400 rpm and a sampling frequency of 51.2 kHz, approximately 2200 data points are captured per rotation. For data preparation, the 510,400 data points representing each defect type are divided into 232 samples, each containing 2200 data points. To enhance the number of samples and better capture key features, sampling is performed in steps of 200 data points. The first sample includes data within the range [0, 2200], while the second sample covers the range [200, 2400], creating overlapping data segments between adjacent samples. Furthermore, the samples are labeled and organized into separate lists for efficient processing.

Table 1. *List of bearing error types of data.*

Health Conditions	Class labels	Fault size (mm)	Total Dataset
Normal Fault	0	-	12740
Inner Fault	1	0.2	38220
Inner and Outer Fault	2	0.2	38217
Outer Fault	3	0.2	38220
Outer and Ball Fault	4	0.2	33610
Ball Fault	5	0.2	25969
Inner and Ball Fault	6	0.2	30826

3. METHOD

3.1. Support vector machine (SVM)

Support vector machine (SVM) [12] is mainly used for binary classification problems, but it can also be extended to solve multiclass classification and regression problems. The basic principle of SVM is to find an optimal hyperplane to separate data classes in the feature space. Distance from data point to hyperplane is the distance between the classification hyperplane and the nearest data point of each class. SVM seeks to maximize this margin to ensure that data points of different classes are separated as clearly as possible. The distance from a data point x_i to the hyperplane is given by:

$$d = \frac{|\mathrm{w} \cdot \mathrm{x_i} + \mathrm{b}|}{|\mathrm{w}|} \quad (1)$$

The goal of SVM is to optimize the boundary [13], that is, to maximize the distance from the hyperplane to the support points. This is expressed by the optimization problem:

$$\min_{w,b} \frac{1}{2}|w|^2 \quad (2)$$

To solve this problem, SVM uses the hinge loss function and adds a regularization algorithm to balance between large margins and classification errors:

$$\min_{w,b} \frac{1}{2}|w|^2 + C\sum_{i=1}^{N} \max\left(0,1 - y_i(w \cdot x_i + b)\right) \quad (3)$$

where *C* is a parameter that adjusts the level for misclassified data points.

3.2. Convolutional Neural Network (CNN)

A convolutional neural network [14] (CNN) is a special type of deep neural network designed to process and analyze data in the form of grids, such as images and time series. CNNs are notable for their ability to automatically extract features from input data, making them a powerful tool in many applications, including image recognition, natural language processing, and incident detection. CNNs typically consist of three main types of layers: convolutional layers, pooling layers, and fully connected layers, as illustrated by figure 1. Each of these layers has a specific role and function in processing and extracting features from the input data

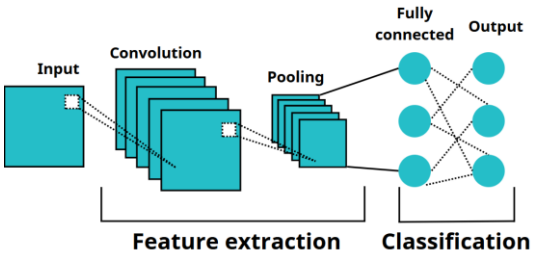


Figure 1. *Basic CNN network structure.*

The convolutional layer is the main layer of the CNN, responsible for extracting features from the input data. This layer uses filters or kernels to perform convolution operations on the input data, creating feature maps. The layer can be represented as follows:

$$(f * g)(t) = \sum_{a=-\infty}^{\infty} f(a)g(t-a) \quad (4)$$

where *f* is the input data and *g* is the filter.

3.3. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a special type of recurrent neural network introduced by Hochreiter and Schmidhuber [15] in 1997 to solve the problems of traditional RNNs in processing and remembering long-term information. LSTM uses a special structure called "cell" to store information and three control gates to regulate the flow of information into and out of the cell.

An LSTM unit was described in figure 2 which consists of four main components: cell state, input gate, forget gate, and output gate. The calculation formula of the input gate is as follows:

$$\mathrm{i_t} = \sigma(\mathrm{W_i} \cdot [\mathrm{h_{t-1}}, x_t] + \mathrm{b_i}) \quad (5)$$

where σ is the sigmoid function, W_i is the weight matrix of the input gate, h_{t-1} is the output from the previous time step, x_t is the input at the current time step and b_i is the bias vector of the input gate.

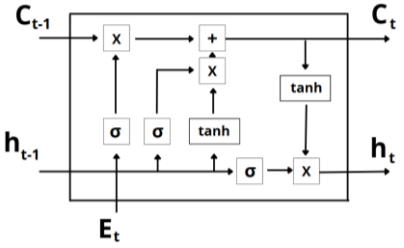


Figure 2. *Basic LSTM cell model.*

The calculation formula for the forget gate is as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{6}$$

where W_f is the weight matrix of the forget gate, b_f is the bias vector of the forget gate.

The calculation formula of output gate is as follows:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{7}$$

where W_o is the weight matrix of the output gate, b_o is the bias vector of the output gate.

3.4. Transformer model with an attention mechanism

Transformer model, introduced by Vaswani et al [16] has revolutionized many fields, including Natural Language Processing (NLP), machine translation, and other deep learning tasks. Transformer architecture consists of two main components: the Encoder and Decoder. However, in many applications, only the Encoder is used for tasks like fault diagnosis. The Encoder processes the input sequence into a context-aware representation, capturing meaningful patterns and dependencies. Meanwhile, the Decoder, when used, generates the target sequence based on the encoder's output. Each layer in the Transformer Encoder and Decoder is composed of:

- Multi-Head Self-Attention Layer: Enables the model to attend to all positions in the sequence.
- Position-wise Feedforward Network (FFN): Applies non-linear transformations using fully connected layers.
- Layer Normalization and Residual Connections: Improve training stability and gradient flow.

A crucial aspect of the Transformer is its Self-Attention mechanism, which enables the model to weigh the importance of each element in the input sequence relative to the others. The core mathematical formulation of the Scaled Dot-Product Attention mechanism in the Transformer is given by:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{8}$$

where: *Q (Query):* The input matrix containing the query information.

K (Key): The matrix containing the key information used for matching.

V (Value): The matrix containing the actual information that will be passed forward.

d_k: The dimension of the key vectors, used for scaling.

To enhance the ability to capture diverse relationships within input sequences, the Transformer utilizes Multi-Head Attention. The Multi-Head Attention mechanism is defined as follows:

$$MultiHead(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Concat}(head_1, \dots, head_h)\boldsymbol{W^O} \tag{9}$$

where each attention *head* is computed as:

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \quad (10)$$

where W_i^Q, W_i^K, W_i^V are learned weight matrices that transform the inputs before computing attention. By employing multiple attention heads, the model captures different aspects of the input representation, leading to improved learning capacity.

Mathematically, the output of the Feedforward Network is computed as:

$$\text{FFN}(\boldsymbol{x}) = \mathbf{max}(\mathbf{0}, \boldsymbol{xW_1} + \boldsymbol{b_1})\,\boldsymbol{W_2} + \boldsymbol{b_2} \quad (11)$$

where W_1, W_2 are weight matrices and b_1, b_2 are bias terms.

3.5. Evaluation metrics

In fault detection problems, model performance is commonly evaluated using accuracy, which measures the proportion of correctly classified instances out of the total number of diagnoses. However, accuracy alone does not provide detailed insights into the model's classification performance for each fault category. To better understand how errors are distributed, a confusion matrix is used. The confusion matrix consists of four key indicators for each classification category:

True Positive (TP): The number of correctly predicted positive instances.

True Negative (TN): The number of correctly predicted negative instances.

False Positive (FP): The number of incorrect positive diagnosis.

False Negative (FN): The number of incorrect negative diagnosis.

Using these values, accuracy is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

By analyzing both accuracy and the confusion matrix, we can assess not only the overall classification performance but also identify specific error patterns, such as the frequency of false alarms or missed detections. This enables a more informed evaluation of the model's effectiveness in fault diagnosis.

4. RESULTS AND DISCUSSION

4.1. Model parameters

The parameters for each model are detailed in tables 2, table 3 and table 4, corresponding to the configurations of the 1D-CNN, 2D-CNN, LSTM and Transformer models, respectively. These tables provide a comprehensive overview of the architectures and associated parameters, highlighting both similarities and differences across the models.

Table 2. *1D-CNN and 2D-CNN model parameters.*

Layer	1D-CNN		2D-CNN	
	Data Dimensions	**Weight (N)**	**Data Dimensions**	**Weight (N)**
Input	(2194, 1)	0	(47, 47, 1)	0
ConvXD_relu_1	(1096, 16)	64	(24, 24, 16)	160
MaxPooling_XD_1	(548, 16)	0	(12, 12, 16)	0
ConvXD_relu_2	(273, 32)	1568	(6, 6, 32)	4640
MaxPooling_XD_2	(136, 32)	0	(3, 3, 32)	0
ConvXD_relu_3	(67, 64)	6208	(2, 2, 64)	18496
MaxPooling_XD_3	(33, 64)	0	(1, 1, 64)	0

ConvXD_relu_4	(16, 128)	24704	(1, 1, 128)	73856
MaxPooling_XD_4	(8, 128)	0	(1, 1, 128)	0
Flatten	(1024)	0	(128)	0
Dense relu	(100)	102500	(100)	12900
Dense relu	(50)	5050	(50)	5050
Dense softmax	(7)	357	(7)	357

Table 3. *LSTM model parameters.*

Layer	Data Dimensions	Weight (N)
Input	(2209, 1)	0
LSTM	(2209, 32)	4352
Flatten	(2209 x 32)	0
Dense	(7)	495
Softmax	(7)	0

Table 4. *Transformer model parameters.*

Layer	Data Dimensions	Weight (N)
Input	(2209, 1)	0
Embedding Layer	(2209, 32)	70688
Positional Encoding	(2209 x 32)	70688
Multi-Head Attention (MHA)	(2209 x 32)	24,576
Feedforward Network (FFN)	(2209 x 32)	8,192
Layer Normalization	(2209 x 32)	64
Output Dense Layer	(2209, 7)	224
Softmax Layer	(2209, 7)	224

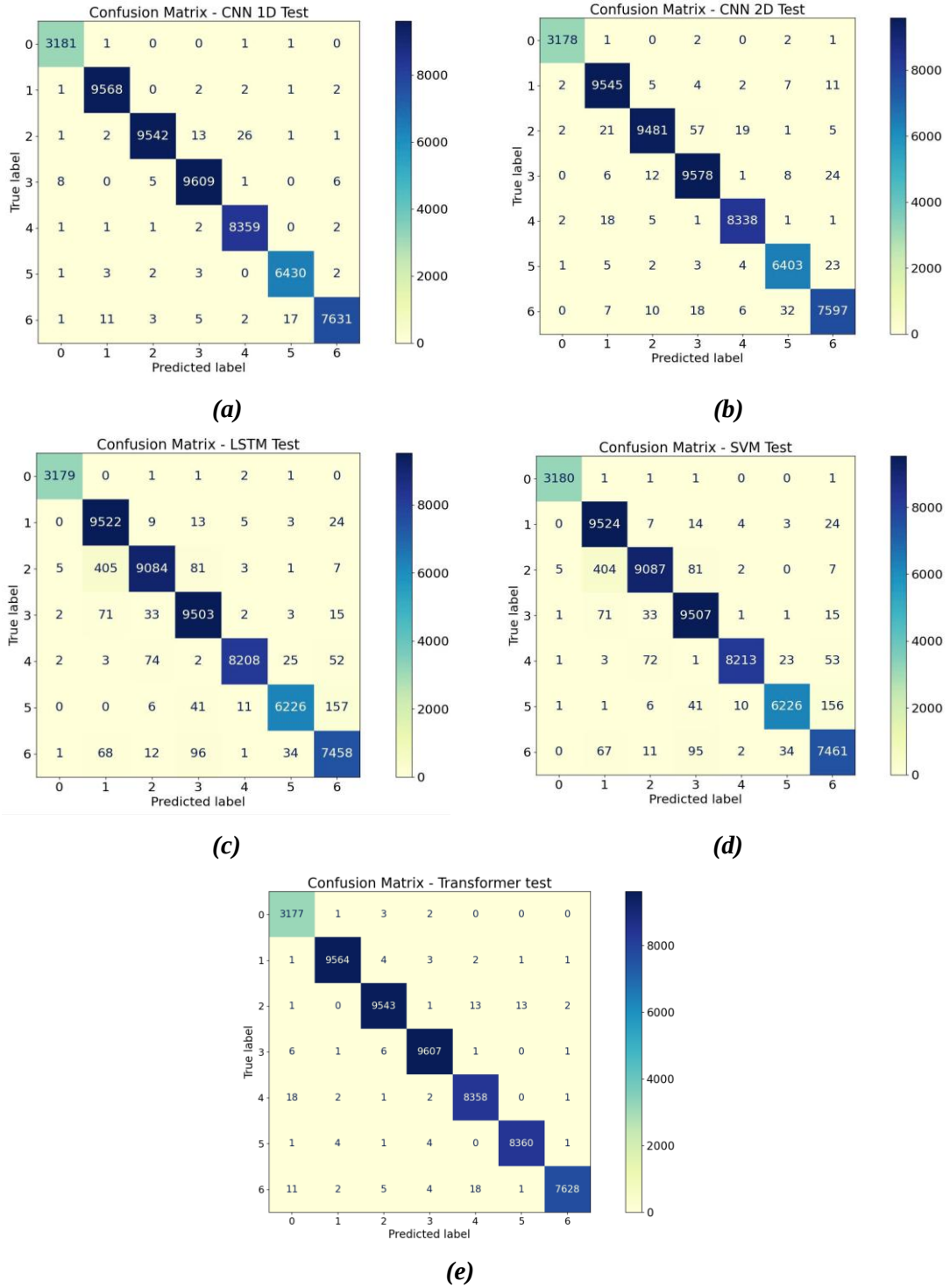
4.2. Fault diagnosis performance

After running the collected data set through different methodological models, the models predict which of the 7 different states that were listed in table 1 to determine the state that have such vibration data.

Figure 3 illustrates the confusion matrices for SVM, LSTM, 1D-CNN, 2D-CNN and Transformer model. The vertical and horizontal axes represent the seven bearing fault categories, as mentioned earlier. The diagonal elements in each matrix indicate the correctly classified instances for each class, while the off-diagonal elements represent misclassifications.

To further quantify model performance, table 5 reports the accuracy and training time for each model. The 1D-CNN model demonstrates the highest performance, achieving an accuracy of 99.8% on the training set and 99.83% on the test set. The 2D-CNN model follows closely with accuracy values of 99.1% for the training set and 99.3% for the test set. The LSTM model, with training and test accuracy of 98.5% and 98.7%, respectively. The SVM model shows the lowest performance, with a training accuracy of 97.5% and a testing accuracy of 97.8%. However, when considering training time, the 2D-CNN model is the fastest, requiring only 0.8 hours, followed by 1D-CNN, which takes 1 hour. Both the LSTM and SVM models require significantly longer training times, at 11.5 hours and 8 hours, respectively. These results suggest that CNN-based models, particularly the 1D-CNN, are not only more effective in terms of classification accuracy but also more efficient in terms of training time, demonstrating a strong ability to rapidly learn and accurately classify key features from vibration signals in bearing fault diagnosis. Notably, the Transformer model, achieving an accuracy of 99.71% on the training set and 99.75% on the test

set, shows competitive performance, close to the top performers. With a training time of just 1 hour, it aligns well with the 1D-CNN model in terms of computational efficiency, presenting itself as a strong alternative for tasks requiring both high accuracy and low training cost.



(a) ***(b)*** ***(c)*** ***(d)*** ***(e)***

Figure 3. *Test set confusion matrix of (a) 1D-CNN; (b) 2D-CNN; (c) LSTM; (d) SVM; (e)Transformer.*

Table 5. *Evaluation results of the models on the training set and test set.*

Model	Accuracy on training set (%)	Accuracy on test set (%)	Training time (h)
1D-CNN	99.80	99.83	1
2D-CNN	99.1	99.3	0.8
LSTM	98.50	98.7	11.5
SVM	97.5	97.8	8
Transformer	99.71	99.75	1

5. CONCLUSIONS

In conclusion, this study presents an effective approach to bearing fault diagnosis using vibration data, evaluated on the HUST bearing dataset. The proposed deep learning models achieved high accuracy, with 1D-CNN excelling at 99.8% accuracy while maintaining minimal training time. This result highlights its suitability for real-time applications, particularly in scenarios with limited computational resources. The 2D-CNN method demonstrated superior efficiency for stride lengths exceeding 200, suggesting that its ability to extract spatial features from vibration signals contributes significantly to its performance. Similarly, the Transformer model, with a competitive accuracy of 99.71% on the training set and 99.75% on the test set, offers a promising balance between performance and training efficiency, making it another viable option for resource-constrained environments.

Furthermore, the comparison of models reveals a trade-off between accuracy and computational efficiency. While LSTM and SVM also achieved high accuracy, their longer training times may limit their practicality in time-sensitive industrial applications. The study also found that reducing stride length improved accuracy up to a certain threshold, with 200 identified as the optimal balance between performance and computational cost. This finding indicates that stride length is a critical hyperparameter influencing model generalization and robustness.

The results confirm that both 1D-CNN and 2D-CNN are reliable for fault diagnosis, even under resource-constrained conditions, due to their efficient feature extraction capabilities. Moreover, the adaptability of this approach to similar datasets suggests its potential applicability across various industrial diagnostic scenarios. The addition of Transformer-based models further strengthens the approach, offering an alternative that delivers high accuracy with reasonable training times. In future studies, we will focus on improving the model by using a hybrid model, integrating advanced signal processing and decomposition techniques to enhance the accuracy of bearing fault diagnosis.

REFERENCES

[1]. Wang Y., Li D., Li L., et al. *"A novel deep learning framework for rolling bearing fault diagnosis enhancement using VAE-augmented CNN model"*. Heliyon, 10(15), e35407, (2024).

[2]. Hakim M., Omran A.A.B., Ahmed A.N., et al. "*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 Eng J, 14(4), 101945, (2023).

[3]. Tian A., Zhang Y., Ma C., et al. *"Noise-robust machinery fault diagnosis based on self-attention mechanism in wavelet domain"*. Measurement, 207, 112327, (2023).

[4]. Vyas N.S. and Satishkumar D., *"Artificial neural network design for fault identification in a rotor-bearing system"*. Mech Mach Theory, 36(2), 157–175, (2001).

[5]. Dong S., Xu X., Liu J., et al. *"Rotating machine fault diagnosis based on locality preserving projection and back propagation neural network–support vector machine model"*. Meas Control, 48(7), 211–216, (2015).

[6]. Hakim M., Omran A.A.B., Ahmed A.N., et al. *"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 Eng J, 14(4), 101945, (2023).
[7]. Han S. and Jeong J., *"An weighted CNN ensemble model with small amount of data for bearing fault diagnosis"*. Procedia Comput Sci, 175, 88–95, (2020).
[8]. Jiang H., Li X., Shao H., et al. *"Intelligent fault diagnosis of rolling bearings using an improved deep recurrent neural network"*. Meas Sci Technol, 29(6), 065107, (2018).
[9]. R. Sabir, D. Rosato, S. Hartmann and C. Guehmann. *"LSTM based bearing fault diagnosis of electrical machines using motor current signal"*, 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Boca Raton, FL, USA, pp. 613-618, (2019), doi: 10.1109/ICMLA.2019.00113.
[10]. H. Wu, M. J. Triebe, J. W. Sutherland. *"A transformer-based approach for novel fault detection and fault classification/diagnosis in manufacturing: A rotary system application"*. Journal of Manufacturing Systems, 67, 439-452, (2023).
[11]. Thuan N.D. and Hong H.S. *"HUST bearing: a practical dataset for ball bearing fault diagnosis"*. *BMC Res Notes*, 16(1), 138, (2023).
[12]. Cortes C. and Vapnik V. *"Support-vector networks"*. Mach Learn, 20(3), 273–297, (1995).
[13]. Kankar P.K., Sharma S.C., and Harsha S.P. *"Fault diagnosis of ball bearings using machine learning methods"*. Expert Syst Appl, 38(3), 1876–1886, (2011).
[14]. LeCun Y., Bengio Y., and Hinton G. *"Deep learning"*. Nature, 521(7553), 436–444, (2015).
[15]. Yang J., Chen F., Long A., et al. *"Runoff simulation of the Kaidu River Basin based on the GR4J-6 and GR4J-6-LSTM models"*. J Hydrol Reg Stud, 56, 102034, (2024).
[16]. A. Vaswani, N. Shazeer, N. Parmar., et al. *"Attention is all you need. NIPS'17"*: Proceedings of the 31st International Conference on Neural Information Processing Systems, 6000-6010, (2017).

TÓM TẮT

Chẩn đoán lỗi ổ bi sử dụng mô hình máy học và học sâu: một nghiên cứu so sánh ứng dụng cho bộ dữ liệu HUST bearing

Chẩn đoán lỗi ổ bi đóng vai trò quan trọng trong việc đảm bảo độ tin cậy và an toàn vận hành của các hệ thống cơ khí và điện tử. Nghiên cứu này thực hiện phân tích so sánh giữa các mô hình máy học trong nhận dạng lỗi ổ bi, bao gồm Máy vector hỗ trợ (SVM), Mạng Long Short-Term Memory (LSTM), Mạng nơ-ron tích chập một chiều (1D-CNN), Mạng nơ-ron tích chập hai chiều (2D-CNN) và Mô hình Transformer. Các mô hình này được áp dụng trên tập dữ liệu HUST bearing và đánh giá dựa trên khả năng phân loại chính xác lỗi từ tín hiệu rung động. Kết quả cho thấy 1D-CNN, 2D-CNN và Transformer có hiệu suất vượt trội trong chẩn đoán lỗi ổ bi. Cụ thể, 1D-CNN đạt độ chính xác 99.8% trên tập huấn luyện và 99.83% trên tập kiểm tra, tiếp theo là 2D-CNN với 99.1% và 99.3%. Mô hình Transformer cũng đạt kết quả tốt với độ chính xác 99.7% sau 1 giờ huấn luyện, tương đương với 1D-CNN (1 giờ) và 2D-CNN (0.8 giờ). Ngược lại, LSTM và SVM có độ chính xác thấp hơn và thời gian huấn luyện dài hơn đáng kể, trong đó, LSTM mất 11.5 giờ và SVM mất 8 giờ. Những kết quả này cho thấy 1D-CNN, 2D-CNN và Transformer là các phương pháp hiệu quả cao trong chẩn đoán lỗi ổ bi, với mô hình Transformer đạt hiệu suất và tốc độ huấn luyện tương đương các mô hình CNN.

Từ khoá: Chẩn đoán lỗi ổ bi; Học máy; Học sâu; Mạng nơ-ron tích chập; Mạng có bộ nhớ dài ngắn hạn; Máy vector hỗ trợ; Mô hình Transformer.