Graph Neural Network-Based Motor Fault Classification Model

YUNGYEONG SHIN[1], JONGPIL JEONG[1], TAEGYUN KIM[2],
HYEONGSU NA[3], SEOCHAN LIM[3]

[1]Dept. of Smart Factory Convergence, Sungkyunkwan University, Suwon-si, Gyeonggi-do,
REPUBLIC OF KOREA
[2]Dept. of Intelligent Robotics, Sungkyunkwan University, Suwon-si, Gyeonggi-do,
REPUBLIC OF KOREA
[3]Dept. of Industrial Engineering, Sungkyunkwan University, Suwon-si, Gyeonggi-do,
REPUBLIC OF KOREA

Abstract: In this work, we propose a novel motor disorder diagnosis model based on graph neural networks (GNNs). This model maximizes model performance by incorporating advanced preprocessing techniques such as Fast Fourier Transform (FFT) and Wavelet Transform (WT). Conventional machine learning and deep learning models such as CNN and SVM find it difficult to handle nonlinear high-dimensional data in motor disorder diagnosis. On the other hand, GNN effectively handles these complex data structures, enabling more accurate and reliable defect classification. Experimental results show that the GNN-based model combining FFT and WT performed well in the diagnosis of motor disorder. Specifically, the FFT-based GNN showed high accuracy, accuracy, and reproducibility at an F1 score of 0.95. The GNN model has lower misclassification rate and higher reliability compared to other models, and ran consistently for various defect types. This is because GNNs can capture complex relationships within frequency domain function (FFT) and time frequency domain pattern (WT). For example, rotational imbalance defects are accurately classified thanks to the ability of GNNs to model harmonic frequency relationships, and bearing defects are accurately classified thanks to the model sensitivity to local frequency spikes that are effectively represented on nodes and edges of the graph. These results suggest that GNN-based motor defect diagnostic systems not only improve diagnostic accuracy, but also have significant potential for real-time applications in manufacturing environments. The system is expected to reduce maintenance costs and improve operational efficiency. The proposed GNN model makes an important contribution by providing practical solutions for the detection and prevention of motion disorders.

Key-Words: Motor failure diagnosis , Predictive maintenance , Failure prediction,Graph Neural Networks,FFT,Wavelet Transform.

Received: June 29, 2024. Revised: December 26, 2024. Accepted: February 21, 2025. Published: April 17, 2025.

1 Introduction

Graph neural networks (GNNs) show significant improvements in various fields. For example, GNNs are used in traffic prediction, weather forecasting, and recommended systems [1], [2]. This development highlights the scalability and efficiency of GNNs in handling complex and interconnected data and suggests they are also suitable for defect prediction in motors. In recent work, GNNs have been used to analyze motor current signals and improve defect detection precision [3], [4]. These models handle complex relationships in data better than traditional machine learning models. Enabling more accurate defect classification and incorporating advanced preprocessing techniques such as Fast Fourier Transform (FFT) and Wavelet Transform (WT) into GNN models can significantly improve defect diagnosis performance [5]. Since these preprocessing methods extract relevant features in the frequency and time-frequency domains, GNNs can achieve higher accuracy than traditional machine learning models.

Traditional motor failure prediction models mainly use machine learning classifiers to provide the benefits of simple interpretation and fast prediction time. However, these models have limited performance degradation due to nonlinear and high-dimensional data [6], [7]. CNN models can provide good performance from complex data, but they require large datasets and have the disadvantage of increasing training time. To address this problem, this study proposes a motion defect prediction model based on GNN. GNNs can effectively handle complex data structures, overcome the limitations of existing models, capture complex interactions within the data, and provide more accurate predictions. By integrating advanced preprocessing techniques such as FFT and Wavelet Transform, model performance can be maximized by extracting important features in the frequency and time frequency domains.

The process of proposing a motor defect classification model using GNN is as follows.

- Analyzing existing research: Analyzing existing motor disorder diagnostic studies to identify the advantages and disadvantages of each approach.
- GNN recognition: Based on GNN's data processing capabilities and successful applications in various fields, the GNN model assumes that it is suitable for predicting motion disorders. Item Configuration: Build a test bed similar to the actual motor drive environment to collect motor defect data.
- Integrating advanced preprocessing techniques: Integrating advanced preprocessing techniques such as FFT and wavelet transforms into GNN models to extract relevant features in the frequency and time frequency domains to improve the accuracy of the model.
- The model is designed and tested as follows. We design a model that combines GNN and preprocessing techniques, validate the performance of the model through various experiments, and ensure that it achieves higher accuracy than traditional machine learning models.

The purpose of this study is to propose a model that combines GNN (graph neural network) and advanced preprocessing technology to improve engine failure diagnosis performance and provide more accurate and reliable prediction. As a result, engine failure prediction becomes more efficient and applicable in industrial environments. The core idea of this study is to leverage GNN to improve the accuracy and reliability of engine failure diagnosis. Combined with GNNs, it extracts critical data in the frequency and time frequency domains and learns complex operating defect patterns. Since conventional machine learning and deep learning models have limitations in handling nonlinear or complex data structures, this study focuses on the ability of GNNs to effectively handle such data. Based on this idea, this study aims to propose a better behavioral anomaly classification model using GNNs. Research shows that GNN-based anomaly diagnosis models significantly outperform current machine learning and deep learning models (CNN, SVM, random forest, etc.). Specifically, the GNN model with FFT and WT for preprocessing achieved high precision, precision, reproducibility, and F1 scores in all tests. For example, the F1 score of the FFT-based GNN model outperformed the previous model with 0.95. This indicates that GNNs can effectively learn key functions in the frequency and time frequency domains, enabling more accurate classification of error patterns. The GNN model also showed stable performance across various engine failure types, and the misclassification rate of each category was minimal. This means that GNNs can capture and process complex relationships in the data and solve the constraints of existing models when processing nonlinear data. In addition, the GNN-based fault diagnosis system proposed in this study is expected to be commercialized in the actual industrial field. As it can be developed into a real-time monitoring and prediction system, it is expected to help reduce industrial site maintenance costs and improve operational efficiency. These results not only highlight the outstanding performance of GNN but also show its practical application to engine error diagnosis. This paper consists of five sections, each of which describes the following topics. The first Section 2 highlights the shortcomings of the current engine failure diagnostic method and explains why GNNs are used. The second section examines existing studies of engine failure diagnostics to clarify why GNNs are better than current methods. Section 3 introduces and describes a GNN-based diagnostic model for malfunctioning operations that integrates FFT and WT preprocessing techniques. Section **??** discusses experimental settings, datasets, and evaluation methods and evaluates GNN performance compared to existing models. Section 5 summarizes the results, highlights the usefulness of the GNN model, and discusses future research directions and potential for real-time systems.

2 Related Work

2.1 Motor Fault Diagnosis

Diagnosing motor failures in industrial environments is important in improving facility reliability and reducing maintenance costs. With the recent development of artificial intelligence and machine learning technologies, the accuracy and efficiency of fault diagnosis have been greatly improved [8]. Existing fault diagnostic methods are mainly based on physical signals such as vibration analysis, noise analysis, and temperature measurement. [9]. These methods are useful for analyzing simple physical properties, but are difficult to recognize complex patterns and have limitations in processing nonlinear or high-dimensional data. Machine learning models such as support vector machines (SVMs), decision trees (decision trees), and random forests (random forests) were introduced to recognize complex patterns based on data and enable rapid prediction. However, these models still have limited performance with high-dimensional data. Deep learning models such as Convolution Network (CNN) perform well to process complex data such as image data, but require large data sets and high computational resources. Combining old physical signal analysis techniques with the latest machine learning and deep learning methods is essential for effective fault diagnosis. This recognizes complex patterns, diagnoses and prevents failures in the early stages, increases facility stability, and reduces maintenance costs. Motor failures can be caused by a variety of causes, and the main types of failures are:

1. Bearing defects: Bearing is an important part of the engine's rotational motion. Defects can be caused by wear, lack of lubrication, physical damage, etc. A bearing defect may appear as an increase in the amplitude of a particular natural frequency in the frequency spectrum.

2. Rotational Imbalance: When the rotor of the motor is unbalanced, it can cause vibrations. This can be due to manufacturing imbalance, component wear or damage, or improper installation. Rotational imbalance results in large amplitudes at the rotational frequency and its harmonic frequencies.

3. Mechanical Defects: Defects in the motor' s mechanical components, such as gears, couplings, and shafts, caused by wear, deformation, or corrosion, may present as periodic patterns in the vibration signals.

4. Shaft Defects: Defects due to shaft cracks, wear, or misalignment. Such defects can cause significant changes in amplitude within the frequency spectrum of vibration signals.

To accurately diagnose and prevent motor fault types, data from various sensors measuring vibration, noise, temperature, etc., are collected. Initially, noise removal and normalization processes are conducted to transform the data into a form suitable for analysis. The collected data is then transformed from the time domain to the frequency domain using Fast Fourier Transform (FFT), enabling the analysis of the frequency spectrum to identify abnormal frequencies. In this course, various machine learning and deep learning models such as Support Vector Machine (SVM), Decision Tree, Random Forest, Convolutional Neural Network (CNN), etc. are used to recognise complex patterns and classify failures. Based on real-time data analysis, a warning system for early failure diagnosis and prevention can be built to increase equipment reliability and ensure proper maintenance before failure occurs. This holistic approach can improve the accuracy and efficiency of engine fault diagnosis, maximize facility reliability, and reduce maintenance costs.

2.2 Graph Neural Networks

Graph Neural Networks (GNNs) are deep learning models used to process and analyze graphically structured data. Unlike traditional grid-like data, GNNs can effectively handle data with irregular relationships [10].

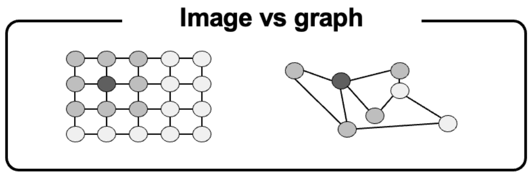

Figure 1: Traditional Grid Data vs. Graph Data

The left side of Figure 1 represents traditional grid data. A typical lattice structure is that each node is connected to the adjacent node according to a fixed pattern. Traditional neural network models such as CNN are mainly used for this type of data. On the other hand, the right side of Figure 1 shows graph data with irregular structures. In graph data, each node can have a variable number of adjacent nodes.

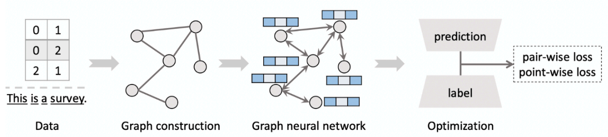

Figure 2: GNN Learning Steps

The four main steps of GNN are shown in Figure 2:

- Data and Graph Structure Preparation: Convert the original data to a graph structure consisting of nodes and edges. A node represents an individual element of data, and an edge represents a relationship between nodes.
- Graph Neural Network Learning: Using graphs as input, each node receives information from the neighboring node and updates its state. This process is repeated over and over again to reflect the overall structure information of the graph. This allows each node to learn the relationships in the graph.
- Prediction—Performs predictions based on the state of the trained node.

- Optimization: Compute the loss by comparing the predicted value with the actual label. To improve the prediction accuracy of the model, we optimize and update the model using per-pair and per-point losses.

GNNs can effectively handle complex graph structures that traditional neural network models cannot handle. This enables GNNs to efficiently analyze and predict complex data across a variety of industries.

3 Graph Neural Network-Based Motor Fault Classification Model

The framework of this paper presents learning results for two preprocessing stages of the GNN model, suggesting that the GNN model shows good performance in motor failure classification for each preprocessing method. A framework for each preprocessing method is shown in Figure 3.

3.1 Preprocessing and Data Feature Extraction

Motor fault diagnosis systems combine sensor vibration data with Fast Fourier Transform (FFT) and Wavelet Transform (WT) preprocessing to effectively classify faults by integrating them into graph neural networks (GNNs). This methodology quantitatively and qualitatively analyzes the frequency distribution to overcome the limitations of existing fault diagnosis methods. The motor fault classification procedure is as follows:

$$|\mathbf{x}|2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} = \sqrt{\sum i = 1^n x_i^2} \tag{1}$$

First, in a preprocessing step, the vibration data collected from the sensor is transformed into a single time series dimension using the Euclidean norm of acceleration across the x, y, and z axes [11]. The effect of gravity is then removed to obtain pure vibration data. To better capture low-frequency characteristics, low-pass filtering is used to remove high-frequency noise above 500 Hz. Min-max normalization is then applied to obtain data values within a specific range to facilitate analysis. Data processed in the time domain is then converted to the frequency domain for frequency analysis.

- FFT (Fast Fourier Transform) is used to transform time-domain data into the frequency domain to identify major vibration components. For example, bearing failure was identified by a sudden increase in amplitude at a specific natural frequency, such as 120 Hz, which corresponds to the frequency of bearing failure. For motors with rotational imbalance, significant peaks are observed at multiples of the rotational frequency, such as 60 Hz and 120 Hz, indicating a harmonic relationship. These frequency-domain features were extracted as nodes in the graph, and the relationship between the harmonics was indicated by the graph edges.

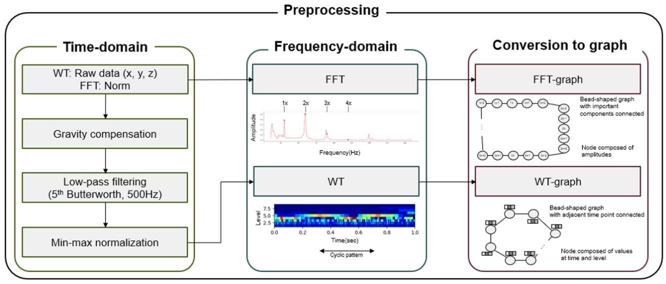

Figure 3: Preprocessing Framework

Based on FFT frequency components, key factors impacting motor fault prediction are extracted. For instance, during normal motor operation, consistent amplitude patterns appear at certain frequencies. However, when a fault occurs, abnormal amplitude changes are detected at specific frequencies. Bearing defects can cause a sharp increase in amplitude at specific natural frequencies, which vary depending on the bearing' s fault characteristics [12]. Rotational imbalance produces high amplitudes at the rotational frequency and its multiples, resulting in a periodic vibration pattern. Additionally, various mechanical issues within the motor may produce abnormal amplitudes at distinct frequency components.

From FFT-derived fault-related frequency components, elements such as the x-component, RMS value, and skewness that influence various fault types are extracted and structured into nodes [13]. Each node represents a significant frequency component, and edges represent the relationships between these components. This graph representation encapsulates both amplitude and frequency distribution, providing a comprehensive dataset for further analysis.

- WT (Wavelet Transform): Wavelet Transform enables simultaneous analysis of changes in both time and frequency domains, which is advantageous for identifying periodic patterns.

Applying the Wavelet Transform generates a Wavelet coefficient plot. From this, data at critical levels for fault detection is extracted. For example, components at levels 3 and 4 for two-pole motors, and levels 4 and 5 for four-pole motors, are selected, as normal motor operation shows consistent components per rotation cycle. Since the range of wavelet coefficients differs by level, min-max scaling is applied to each level to prevent skewed judgments toward particular levels. The extracted data is then transformed into a graph element structure. Each node comprises key-level wavelet coefficients in the x, y, and z dimensions at a specific time point, with edges connecting adjacent time periods. Additionally, since the motor is configured to repeat a consistent operation, nodes representing the beginning and end values of a cycle are connected considering the motor' s rotation speed. This process enables the visualization of vibration pattern information in the time-frequency domain as a graph.

3.2 GNN Model Training

Once the graph is constructed, the Graph Convolutional Network (GCN) is used to train the graph data. The GCN consists of the following structure: graph convolution layer, batch normalization layer, and ReLU activation. Hyperparameters such as learning rate (set to 0.001), dropout rate (0.2 for FFT-GNN and 0.3 for WT-GNN), and number of graph convolution layers (2 layers) were determined through grid search to optimize model performance. For example, a low dropout rate was chosen for the FFT-GNN case to preserve important frequency domain features, while a high dropout rate was applied for the WT-GNN case to prevent overfitting due to complex data structures.

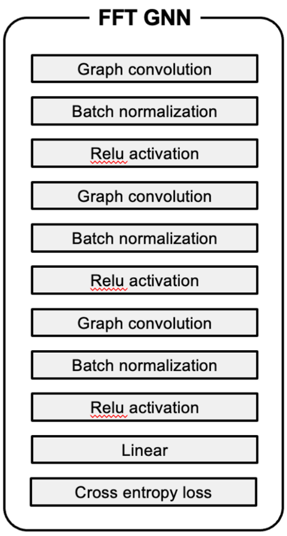

Figure 4: FFT GNN

Figure 4 shows the FFT GN that processes graph data through the Graph Synthesis (GC) hierarchy and extracts features. This hierarchy uses information from each node and its surrounding nodes to learn meaningful patterns. The batch normalization layer then stabilizes the learning process and increases the learning speed. The ReLU activation function provides nonlinearity and improves neural network performance by converting negative numbers to zero. This process is repeated twice with additional GC hierarchy, batch normalization, and ReLU activation sequentially applied. Finally, the linear hierarchy calculates the output and solves the classification task using cross entropy loss. Figure 5 shows the WT GNN, which also processes data through GC layers, with Batch Normalization and ReLU activation applied similarly. However, the WT GNN includes a Dropout layer to prevent overfitting by randomly excluding certain neurons during training, aiding model generalization [14].This Dropout layer is applied twice to enhance stable learning of WT data characteristics. Additionally, Dropout is included within the repeated GC, Batch Normalization, and ReLU layers, further strengthening overfitting prevention. The final Linear layer and Cross Entropy Loss function operate in the same way as in the FFT GNN. The main difference between FFT GNN and WT GNN is that the WT GNN includes a Dropout layer that provides an additional stabilization mechanism to prevent overfitting due to WT data complexity. The WTGN structure, which consists of multiple Dropout layers, can learn more complex model structures and capture more variability. FFT GNN, on the other hand, processes data in a simple structure without a dropout. In conclusion, each model is designed for data characteristics, and while WTGN uses a more complex learning process using Dropout to handle complex data structures, FFTGN has a simpler learning structure. Each preprocessing GCN performs synthetic operations from the

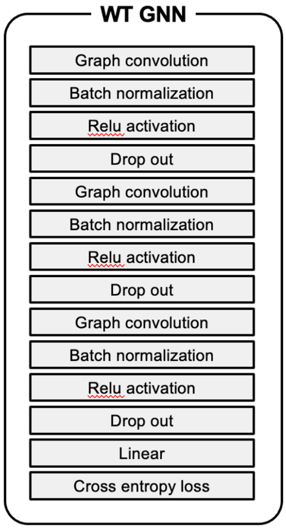

Figure 5: WT GNN

graph to effectively learn spatial dependencies and interactions between various frequency components. This process allows the GCN to extract higher-level features from graphs essential for accurate defect classification.

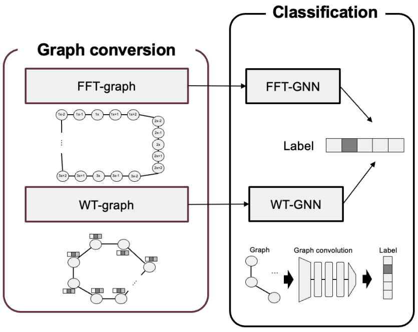

Figure 6: GNN Training

Then, as shown in Figure 6, the motor state is classified into various error categories using the trained GCN model. The classification process involves reducing the graph to a feature vector that best describes the motor's operating state. Then, when you enter this vector into a pre-trained classifier, you determine the error type based on the learned pattern. The effectiveness of this approach has been verified by extensive testing of data sets, including normal and error motor conditions. The experiment demonstrated the robustness and reliability of the proposed system in real-world environments with high accuracy and 0.94 F1 points. Innovative use of FFT and GNN not only improves defect detection accuracy, but also helps reduce computational complexity and data volume compared to traditional methods.

4 Experiment and Results

4.1 Experiment and Results

Table 1. System specification.

Hardware Environment	**Software Environment**
CPU: Intel Xeon Silver 4216 CPU @ 2.10GHz **GPU:** 4 x NVIDIA RTX A5000 **Memory:** 256GB DDR4 **Storage:** 2TB SSD	**OS:** Ubuntu 18.04.6 LTS **Framework:** TensorFlow 2.11.0 **Framework:** pytorch 2.0.1 **Programming Language:** Python 3.10.9

Table 1 shows the hardware and software experimental settings used in this study. The hardware configuration consists of an Intel Xeon Silver 4216 CPU @ 2.10GHz processor and four NVIDIA RTX A5000 graphics cards. The software configuration leveraged Python 3.10.9, a programming language commonly used for data analysis and machine learning, running on the Ubuntu 18.04.6 LTS operating system. In addition, TensorFlow 2.11.0, an open source machine learning framework, was used to generate and run machine learning models. Python is a multi-talented, user-friendly, and widely used framework in deep learning studies, and was also used as an essential tool in this study. Finally, we used the Cybit Learning Library, which provides various tools for data analysis and machine learning for model comparison. The data sets used in this study were collected from configurations that mimic various operating conditions. In the case of experimental setting, 30T wooden blocks were used for vibration measurement and 25T rubber pads were used to stop the motor shaking. In addition, a flat iron plate was used to stabilize the motor, and a clamp was implemented to secure the sensor to the iron plate to prevent movement. In the load environment, the timing belt was connected to the motor to apply the load, and the sensor was fixed to the motor under load conditions to collect vibration data.

4.2 Dataset

This study utilizes big data vibration data beyond industrial machinery. This dataset contains vibration signals for both normal and various abnormal operating conditions to effectively analyze potential mechanical anomalies that may occur in an industrial environment. Future work plans to further expand the dataset to include additional types of anomalies (e.g., electrical defects, thermal imbalances), broader mechanical types (e.g., induction motors, synchronous motors), and various load conditions. This improves the generalization of the model and verifies its applicability in a variety of industrial environments. The vibration data were collected using sensors attached to the machinery, measuring signals along the X, Y, and Z axes. The magnitude and characteristics of the measured vibrations can vary depending on the sensor' s installation orientation and the machine' s structural configuration. Such multi-axial vibration signals provide essential information for accurately understanding the machine' s operating conditions. Furthermore, by employing a consistent sampling rate (e.g., 1000 Hz) and sufficient recording periods (e.g., on the order of milliseconds to seconds), we obtained high-resolution time-domain vibration data. This approach allows us to clearly differentiate between normal states and various abnormal conditions.

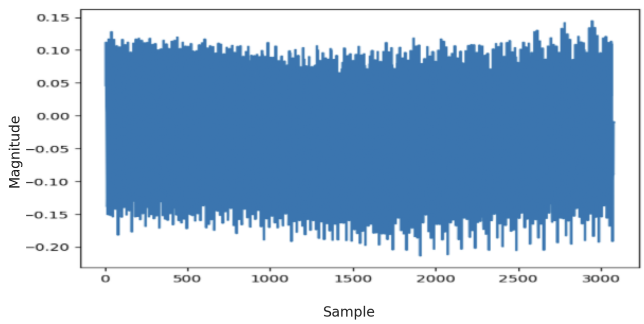

Figure 7: x axis data

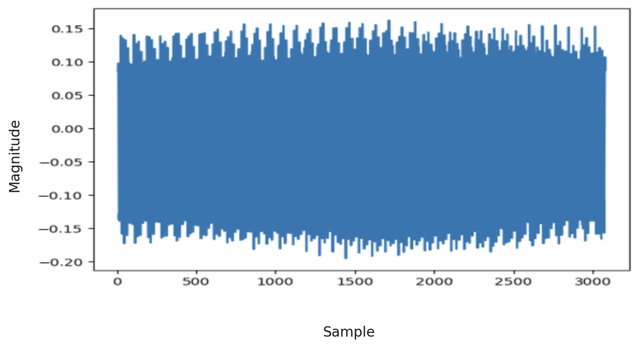

Figure 8: y axis data

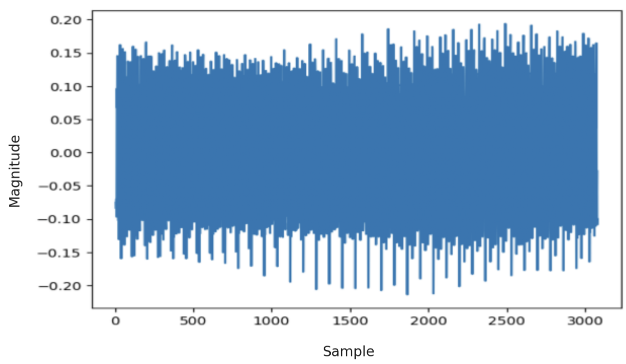

Figure 9: z axis data

Figures Figure 7,Figure 8, and Figure 9 present examples of time-series vibration data collected along the X, Y, and Z axes, respectively. By examining these multi-axial vibration signals, it is possible to identify the distinct characteristics of each axis and analyze machine anomalies from multiple perspectives. For instance, the X-axis data (Figure 7) can reveal how vibrations change along a specific direction, while the Y-axis data (Figure 8) can complement the analysis by highlighting differences in another direction. The Z-axis data (Figure 9) further contributes to understanding the three-dimensional vibration pattern, thus supporting the development and refinement of anomaly detection models based on multi-axial analysis. Overall, this comprehensive multi-axis vibration dataset encompasses a wide range of conditions—from normal operation to various types of abnormalities (e.g., bearing damage, rotor imbalance, and other mechanical faults). As a result, it provides a rich environment for training and validating predictive models. By leveraging such data, it is possible to develop and improve predictive models that can address a variety of real-world industrial scenarios, ultimately enhancing the robustness and applicability of these models across different industrial sectors.

4.3 Evaluation Metrics

Several evaluation metrics are commonly used to assess the performance of classification machine learning models. These metrics evaluate various aspects of the model, with the most popular being Accuracy, Precision, Recall, and F1 Score. These metrics are calculated based on the Confusion Matrix.

- TP (True Positive): Correctly predicting positive data as positive.
- FP (False Positive): Incorrectly predicting negative data as positive.
- TN (True Negative): Correctly predicting negative data as negative.
- FN (False Negative): Incorrectly identifies positive data as negative values.

Accuracy gauges the proportion of correctly predicted instances among all instances and gives a summary of model performance. Accuracy is computed as follows:

$$\text{Accuracy} = \frac{TP}{TP + FP + TN + FN} \tag{2}$$

Recall shows the portion of accurately predicted positive cases compared to all true positive cases. This matters when the expense of false positives is significant and shows how effectively the model identifies real positive cases. Recall is computed in this way:

$$\text{Recall/Sensitivity} = \frac{TP}{TP + FN} \tag{3}$$

Precision assesses the percentage of positive cases that are accurately predicted from all cases forecasted as positive. This metric is important when the expense of false positives is significant and shows how effectively the model ensures the correctness of positive predictions. Precision is computed in this way:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

The F1 score is the average of precision and recall and gives a balanced measure that considers both false positives and false negatives. It gives a complete evaluation of how well the model is doing on both areas; the F1 score is computed as follows:

$$\text{F1-Score} = \frac{2 \cdot \text{Sensitivity} \cdot \text{Precision}}{\text{Precision} + \text{Sensitivity}} \tag{5}$$

4.4 Results

In this study, the performance of the GNN (Graph Neural Network) model for the motor defect classification task was evaluated in combination with various preprocessing techniques. In particular, wavelet transform and FFT (Fast Fourier Transform) were applied to transform the data features and then input into the GNN model to measure its performance. Experimental results are summarized as follows

Table 2. Performance Metrics Comparison

Dataset	***Accuracy***	***Precision***	***Recall***	***F1-Score***
Validation	0.92	0.92	0.92	0.92
Test	0.95	0.95	0.95	0.95

[a]Performance metrics based on model evaluation.

Table 2 confirms that the GNN model achieved an accuracy of 0.92, precision of 0.92, recall of 0.92, and F1 Score of 0.92 on the validation dataset. On the test dataset, it reached 0.95 across all metrics, demonstrating high performance.

In the case of the gradient boosting model in Figure 10, there was a high misclassification rate in Class 2 and Class 3. In particular, Class 2 was often misclassified as Class 1, which is interpreted as the gradient boosting model not being able to clearly distinguish the boundaries between classes. The gradient boosting model works by learning the data sequentially to correct errors, but in this case, the performance was degraded due to the occurrence of misclassification among certain classes. Overall, the model recorded low accuracy, especially in Class 3. This suggests that the model had difficulty handling complex data structures. In the case

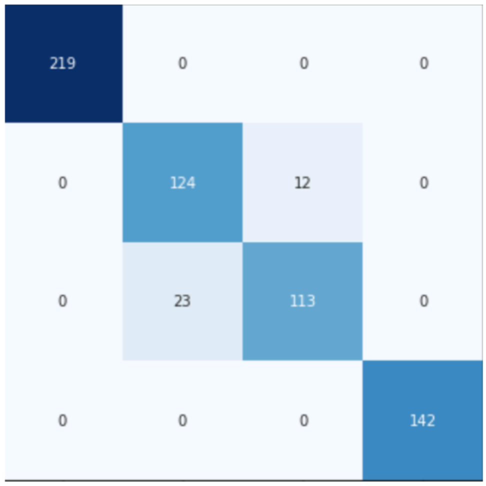

Figure 10: Gradient

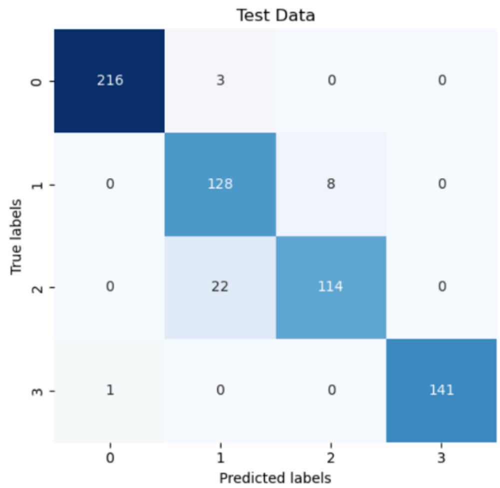

Figure 11: Randomforest

of the random forest model in Figure 11, a high misclassification rate was observed in class 1 and class 2. In particular, samples from class 2 were often incorrectly classified as class 1. This can be seen as a result of the random forest model's poor distinction between these two classes. Although the random forest makes predictions using a number of decision trees, it had difficulty in distinguishing the complex boundaries between classes. Overall, the performance for class 3 was relatively good, but the boundary between classes was not clear, resulting in misclassification. In the case of the SVM model in Figure 12, there was a high misclassification rate between class 1 and class 2. In particular, the case where class 2 was incorrectly classified as class 1 was noticeable. SVMs work by finding the optimal

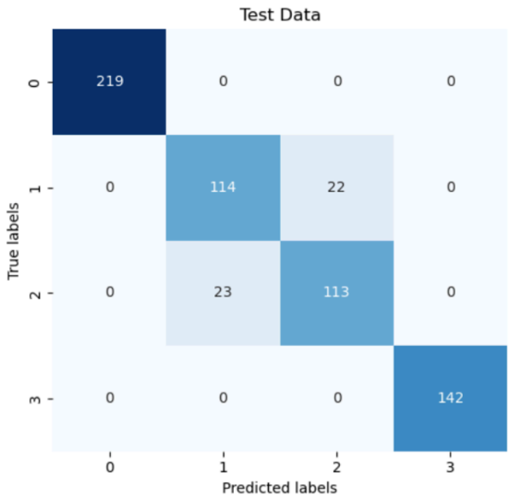

Figure 12: SVM

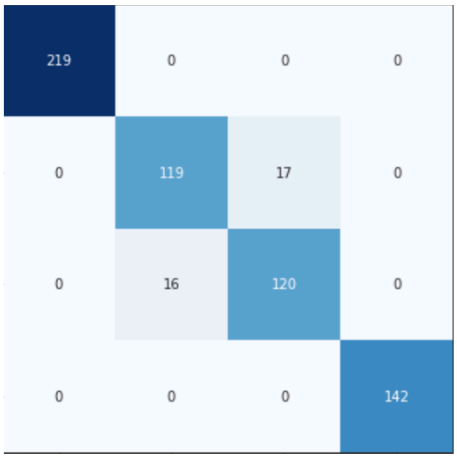

Figure 13: KNN

boundary in high-dimensional space, but performance can be degraded if the distinction between classes is difficult. This model showed relatively stable performance for class 3, but there was a problem with the distinction between classes 1 and 2. In the case of the KNN model in Figure 13, there was a high misclassification rate between class 2 and class 3. In particular, the sample of class 2 tended to be misclassified into class 3. The KNN model had difficulty in distinguishing the boundaries between classes in FFT pre-processing data, and since it is classified based on nearby data points, it seems that the result did not reflect the complex boundaries well. Overall, it showed lower performance than the GNN model, and the misclassification rate was high. In the case of the GNN model in Figure 14, high

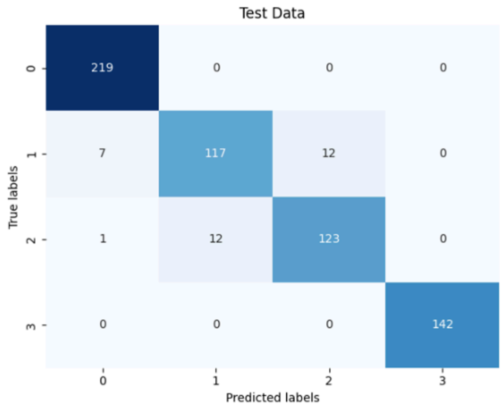

Figure 14: GNN

accuracy and low misclassification rate were shown in all classes. In particular, the GNN model showed a clear distinction between classes, high performance in FFT pre-processing data, and excellent performance in data using Wavelet Transform. GNN is a model that learns the relationships between data well and performs well in complex data structures, and showed consistent high performance compared to other models. The GNN model maintained high accuracy in all classes compared to other models and had the lowest misclassification rate. On the other hand, gradient boosting, random forest, SVM, and KNN models showed high misclassification rate in some classes. In particular, gradient boosting models performed poorly in classes 2 and 3, random forest and SVM models in classes 1 and 2, and KNN models recorded high misclassification rates in classes 2 and 3. The GNN model effectively learned the relationship between data and showed superior performance compared to other models.

Table 3. Performance Metrics Comparison

Dataset	***Accuracy***	***Precision***	***Recall***	***F1-Score***
Validation	0.96	0.95	0.95	0.95
Test	0.96	0.94	0.94	0.94

[a]Performance metrics based on model evaluation.

Table 3 confirms that in the Confusion Matrix for the training dataset, the GNN model recorded high accuracy across all classes, with minimal misclassification, particularly in major classes. The GNN model maintained high accuracy (Validation: 0.964, Test: 0.963) in the validation and test datasets, with high values in precision, recall, and F1 Score.

The GNN model in Figure 15 showed high accuracy in all classes and very low misclassification

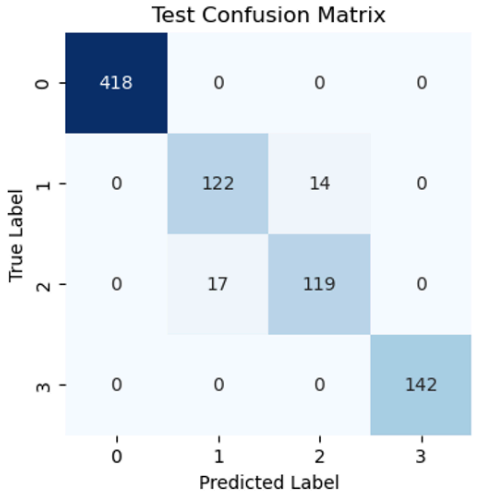

Figure 15: GNN

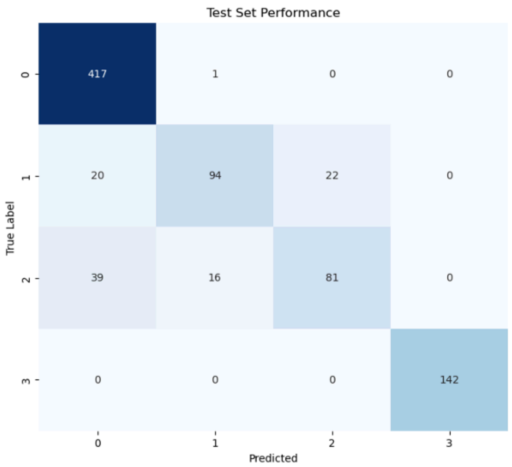

Figure 16: Randomforest

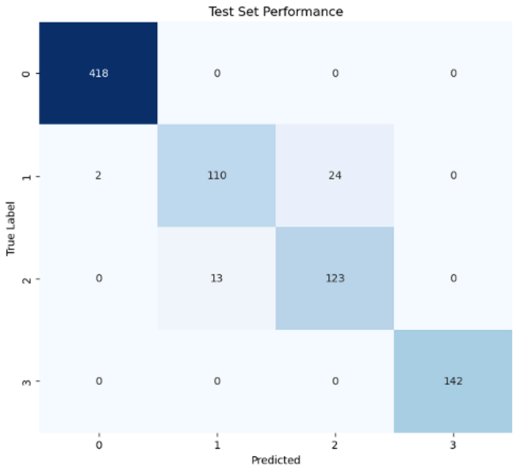

Figure 17: SVM

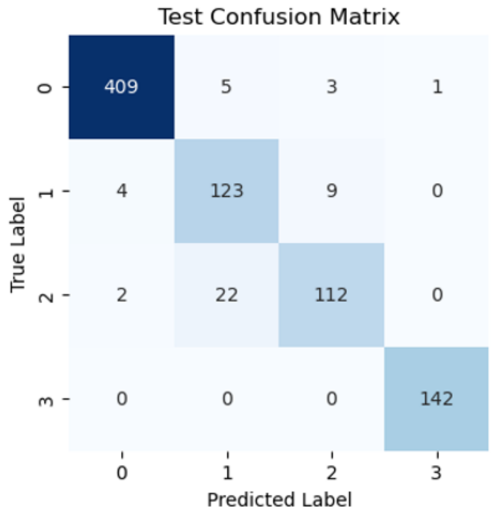

Figure 18: CNN

rate performance. In particular, the prediction performance in classes 0, 1, 2, and 3 was very consistently high, and the boundaries between classes were clearly separated. GNNs performed well on wavelet transformed data, and the accuracy was evenly maintained in all classes. This shows that the GNN model can learn the relationship between data well and recognize complex patterns effectively. The random forest model in Figure 16 showed a high misclassification rate in classes 1 and 2. In particular, samples of class 1 were often incorrectly classified into class 2, which resulted in poor performance of the model. Random forest models tended to fail to distinguish complex inter-class boundaries, resulting in inaccurate predictions between classes. The overall performance showed lower accuracy than that of the GNN model, and there were many cases where the boundaries between classes were not clear. The SVM model in Figure 17 showed a high misclassification rate in Class 1 and 2. Class 1 tended to be misclassified into Class 2, which is interpreted as the fact that the SVM had difficulty finding the optimal boundary in the high-dimensional space. The SVM model showed constant performance on the wavelet-transformed data, but the boundary between Class 1 and 2 was not well distinguished, resulting in misclassification. As a result, there was a problem that the ability to distinguish between classes was inferior to that of the GNN model, recording lower accuracy. The CNN model in Figure 18 showed relatively high accuracy, but the performance was lower than that of the GNN model due to the occurrence of class 1 to class 2 misclassification. The CNN model is particularly strong in feature extraction and can learn the spatial relationship of data well, but misclassification can occur if the boundaries between classes are not clear. The prediction discrepancy between classes 1 and 2 caused the performance degradation of the CNN model, and the GNN model showed consistent high performance without these problems. The GNN model showed high precision and low misclassification rate in both wavelet conversion data and FFT pre-processing data, and showed consistent performance in all classes. Meanwhile, random forest, SVM, and CNN models showed relatively high misclassification rates in major classes, while random forest and SVM models recorded high misclassification rates, especially in classes 1 and 2. The CNN model showed relatively high precision, but the performance was degraded due to classification mistakes in classes 1 and 2. The GNN model showed good performance in accuracy, repeatability, and F1 score, and consistent performance across all classes. These results suggest that the GNN model is highly effective for motor failure classification tasks, and that its real-time processing power and low misclassification rate greatly help to reduce maintenance costs and improve operational efficiency. Furthermore, the GNN model has shown strong performance that is useful in various industries.

5 Conclusion

In this study, a motor disorder diagnosis model using a graph neural network (GNN) was introduced and its effect was verified with various preprocessing methods. The strong data processing power of GNN is integrated into Fast Fourier Transform (FFT) and Wavelet Transform (WT), which significantly improves the accuracy and reliability of motor failure diagnosis compared to previous models. The test results show that the GNN-based model achieves excellent accuracy, repeatability, and F1 score on datasets treated with FFT and WT, which is very effective in diagnosing motor failures. Specifically, the GNN model combined with FFT achieved an F1 score of 0.95, and the WT-based model also showed consistent results across all classes. This shows that the GNN can learn more accurately about complex motor failure patterns, extract key features in the frequency and time frequency domains, and predict failures effectively. The proposed GNN-based model outperformed traditional machine learning and deep learning models (CNN, SVM, random forest, etc.). Existing models had limitations in dealing with high-dimensional data and nonlinear properties, but GNN overcame these limitations and achieved higher accuracy in defect diagnosis. The GNN model provides consistent predictive performance for all defect types, allowing more accurate detection of defect patterns through FFT and WT pretreatment. These results mean that GNN-based systems are very useful in real-world industrial environments. The model's high accuracy and low motor defect diagnostic misclassification rates have the potential to reduce maintenance costs and significantly improve operational efficiency. It also has the potential to develop into a real-time defect prediction system to increase the potential applicable in the industrial environment and to demonstrate important technological advances. Future research directions include greater data diversity, model optimization, and real-time applications. In particular, the proposed GNN-based motor defect diagnosis model can be integrated into industrial monitoring systems such as manufacturing plant forecast maintenance platforms. By using edge computing devices to process motor sensor data in the field, real-time systems can be developed to ensure low-latency defect detection. For example, real-time analysis of vibration data collected from an industrial motor in operation enables immediate intervention and reduces shutdown time.

- Expanding data diversity: expanding data collection across different motor types and failure types to validate model generalizability and evaluate model performance in a wider range of environments.

- Model Optimization: We plan to optimize the hyperparameters of the GNN model and apply various GNN structures to further improve model performance.

- Real-time application: We would like to build a real-time motor fault diagnosis system, test its applicability in a real industrial environment, and validate its real-time data processing and prediction accuracy.

Acknowledgment:

This work was supported by ICT Creative Consilience Program through the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (RS-2020-II201821). Corresponding author: Prof. Jongpil Jeong.

This research (paper) used datasets from 'The Open AI Dataset Project (AI-Hub, S. Korea)'. All data information can be accessed through 'AI-Hub (www.aihub.or.kr)'.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work, the authors used ChatGPT (OpenAI) for the purpose of improving the clarity and readability of the manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References:

[1] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," AI open, vol. 1, pp. 57–81, 2020

[2] S. Qiu, X. Cui, Z. Ping, N. Shan, Z. Li, X. Bao, and X. Xu, "Deep learning techniques in intelligent fault diagnosis and prognosis for industrial systems: a review," Sensors, vol. 23, no. 3, p. 1305, 2023.

[3] D. Hou, B. Zhang, J. Chen, and P. Shi, "Improved gnn based on graph-transformer: A new framework for rolling mill bearing fault diagnosis," Transactions of the Institute of Measurement and Control, p. 01423312241265774, 2024.

[4] H. Chen, X.-B. Wang, and Z.-X. Yang, "Semi-supervised self-correcting graph neural network for intelligent fault diagnosis of rotating machinery," IEEE Transactions on Instrumentation and Measurement, 2023.

[5] L. Wang, F. Xie, X. Zhang, L. Jiang, and B. Huang, "Spatial-temporal graph feature learning driven by time–frequency similarity assessment for robust fault diagnosis of rotating machinery," Advanced Engineering Informatics, vol. 62, p. 102711, 2024.

[6] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," Electronic Markets, vol. 31, no. 3, pp. 685–695, 2021.

[7] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamar´ıa, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," Journal of big Data, vol. 8, pp. 1–74, 2021.

[8] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, "Applications of machine learning to machine fault diagnosis: A review and roadmap," Mechanical systems and signal processing, vol. 138, p. 106587, 2020.

[9] H. Henao, G.-A. Capolino, M. Fernandez-Cabanas, F. Filippetti, C. Bruzzese, E. Strangas, R. Pusca, J. Estima, M. Riera-Guasp, and S. Hedayati-Kia, "Trends in fault diagnosis for electrical machines: A review of diagnostic techniques," IEEE industrial electronics magazine, vol. 8, no. 2, pp. 31–42, 2014

[10] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," IEEE transactions on neural networks and learning systems, vol. 32, no. 1, pp. 4–24, 2020.

[11] A. Kafeel, S. Aziz, M. Awais, M. A. Khan, K. Afaq, S. A. Idris, H. Alshazly, and S. M. Mostafa, "An expert system for rotating machine fault detection using vibration signal analysis," Sensors, vol. 21, no. 22, p. 7587, 2021.

[12] S. A. McInerny and Y. Dai, "Basic vibration signal processing for bearing fault detection," IEEE Transactions on education, vol. 46, no. 1, pp. 149–156, 2003.

[13] M. Altaf, T. Akram, M. A. Khan, M. Iqbal, M. M. I. Ch, and C.-H. Hsu, "A new statistical features based approach for bearing fault diagnosis using vibration signals," Sensors, vol. 22, no. 5, p. 2012, 2022.

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhut- dinov, "Dropout: a simple way to prevent neural networks from overfitting," The journal of machine learning research, vol. 15, no. 1, pp. 1929–1958, 2014.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

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