Proceedings of the Canadian Society for Mechanical Engineering International Congress
32nd Annual Conference of the Computational Fluid Dynamics Society of Canada
Canadian Society of Rheology Symposium
CSME-CFDSC-CSR 2025
May 25–28, 2025, Montréal, Québec, Canada

Bearing Fault Diagnosis Using Traditional Machine Learning via ChatGPT

Niousha Khalilian[1], Mert Sehri[1], Zehui Hua[1], Patrick Dumond[1]
[1]Department of Mechanical Engineering, University of Ottawa, Ottawa, Canada
* nkhal007@uottawa.ca

Abstract—The increased use of large language models (LLMs) is facilitating a new era for machine fault diagnosis, where LLMs are now capable of conducting simple machine learning (ML) analysis for condition monitoring. This paper explores the potential of LLMs, specifically ChatGPT, for use in diagnosing rolling element bearing faults. Through structured prompts and automated model execution, ChatGPT was tested on the Case Western Reserve University (CWRU) bearing dataset using traditional ML algorithms—Support Vector Machine (SVM), Random Forest (RF), and k-Nearest Neighbors (KNN). The results demonstrate that while ChatGPT can effectively apply feature extraction techniques and execute ML models, its performance is highly dependent on structured guidance, dataset preprocessing, and feature selection. The findings highlight the strengths of ChatGPT in facilitating traditional ML-based fault diagnosis but also reveal its limitations in handling raw data and optimizing deep learning models. These insights pave the way for future research in integrating LLMs with industrial diagnostic frameworks.

Keywords-Large Language Models; ChatGPT; Rolling Element Bearing Fault Diagnosis; Machine Condition Monitoring;

I. INTRODUCTION

The emergence of large language models (LLMs) has opened a new era for fault diagnosis, particularly for machine condition monitoring [1], [2]. ChatGPT, a LLM, with its ability to process large amounts of data, is beginning to demonstrate potential for proposing and implementing traditional machine learning (ML) methods for diagnosing the health state of mechanical components [3]. Among the vast applications of ChatGPT [4], [5], [6], diagnosing rolling element bearing faults stands out due to the LLM's potential to provide insight into complex data driven challenges.

Bearing diagnosis is important in preventing machine failures [7]. Traditionally, bearing fault analysis has relied on statistical methods, signal processing, and ML algorithms to detect and classify faults [8]. However this reliance requires domain expertise and a good computational hardware, often limiting the transition from research labs to industrial applications [9]. The integration of ChatGPT [10] with fault diagnosis presents an opportunity to bridge some of these gaps by leveraging the LLM's pre-trained knowledge and natural language capabilities.

This paper delves into the potential of ChatGPT, a prominent LLM, to perform simple yet impactful ML analysis for bearing fault diagnosis. By evaluating its ability to interpret data, generate insights, and provide guidance for condition monitoring tasks, the aim is to explore the broader implications of LLMs in industrial fault diagnosis. A publicly available bearing dataset was used to assess the fault detection accuracy capabilities provided by ChatGPT. Appropriate prompts were developed to run the entire diagnostic process using only the LLM. Additionally, a custom Bearing Diagnosis GPT was created within ChatGPT so that ML researchers can upload their own datasets for testing fault detection accuracies. The findings demonstrate both the capabilities and the limitations of these models, paving the way for future research in integrating LLMs with traditional diagnostic frameworks.

II. LIMITATIONS OF FAULT DIAGNOSIS WITH CHATGPT

Although ChatGPT continues to develop at a fast rate, there are still limitations when it comes to conducting fault diagnosis. The following limitations are identified for ChatGPT v2 (released November 6th, 2023) [10]:

- Even though this LLM can run Python, PyTorch is not currently supported in ChatGPT's environment, preventing ML researchers from running deep learning algorithms. Moreover, PyTorch (or its equivalent) cannot currently be replicated by ChatGPT's resources and token/response limits.
- There is a data file storage limit, meaning that traditionally large datasets cannot currently be uploaded unless preprocessing is first conducted.
- Without proper prompts and guidance, in terms of explicit dataset and ML knowledge, ChatGPT won't be able to come up with a fault diagnosis methodology and detection accuracies will be poor.

III. ChatGPT Implementation Methods

In this section, methods for implementing traditional ML algorithms are discussed, including support vector machine (SVM), random forest (RF), and k-nearest neighbors (KNN). Also, by asking ChatGPT to try different hyperparameters, the model's performance can be further improved.

A. Traditional ML Analysis Using ChatGPT

ChatGPT, while a powerful language model, is inherently limited in its ability to perform advanced machine learning tasks beyond traditional supervised learning methods. This limitation stems from its architecture, which is designed primarily for natural language processing (NLP) tasks rather than complex data analysis or model training. Specifically, ChatGPT lacks the capability to perform deep learning model training or optimization directly. Instead, it is better suited for traditional ML techniques, such as classification, regression, and clustering, using pre-extracted features. In fact, bearing fault diagnosis can be considered as a classification problem, thus a classifier should be trained to help correctly predict the actual health states of each test sample.

As stated in Section II, ChatGPT cannot process raw data because it relies on preprocessed or extracted features. Although ChatGPT can provide code and guidance for training models, it cannot execute training or optimization processes itself. So, the input data should be well structured before applying different traditional ML algorithms.

B. Introduction to SVM, RF, and KNN

1) SVM

SVM is a supervised learning algorithm that finds the optimal hyperplane to separate data points of different classes in a high-dimensional space. It is particularly effective for binary classification tasks. Also, by modifying the different kernel functions, it can be further applied to multi-class classification problems.

SVM can be used to classify bearing health conditions by mapping the extracted features into a higher-dimensional space where the classes are separable.

2) RF

RF is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (for classification) or the mean prediction (for regression) of the individual trees. It is known for its robustness to overfitting and ability to handle high-dimensional data.

3) KNN

KNN is a simple, instance-based learning algorithm that classifies data points based on the majority class among their k-nearest neighbors in the feature space. KNN can classify bearing faults by comparing the extracted features of a test sample to those of labeled training samples.

It is important to note that these three traditional ML methods are all supervised learning methods, which means that both samples and labels should be used as inputs. As for testing the samples that are collected under different working conditions (i.e., varying speeds and loads), the model performance may degenerate as domain adaptation is not designed to deal with unknown domain shifts.

C. Optimization with ChatGPT

Once the algorithm is selected, and the data has been uploaded to the website, ChatGPT can then be used to provide classification results based on a preliminary set of selected parameters. To evaluate the model's performance, testing accuracies are used to provide a comparison.

Additionally, by leveraging ChatGPT's ability to fine-tune the hyperparameters of each algorithm—such as adjusting the number of trees and maximum depth in RF, selecting the optimal kernel and regularization parameter in SVM, or determining the ideal number of neighbors in KNN—it is possible to achieve a higher accuracy. This iterative optimization process allows the performance of these models to be further improved.

IV. Methodology

In this section, the methodology used in this study is outlined, detailing the steps taken to evaluate ChatGPT's ability to perform traditional ML tasks. This section also provides an overview of the dataset and domain selection criteria.

A. CWRU Dataset and Domain

The Case Western Reserve University (CWRU) bearing dataset is a widely used resource for analyzing bearing performance under various conditions. It contains vibration data collected from a 2-hp motor, including measurements from both healthy bearings and bearings with induced faults. The faults were introduced using electro-discharge machining (EDM) and vary in size, with diameters of 0.007, 0.014, and 0.021 inches. Vibration signals were recorded at sampling frequencies of 12,000 or 48,000 samples per second, depending on the experimental setup [11]. The CWRU dataset is selected for testing due to its well-documented and standardized structure, which provides a reliable benchmark for fault diagnosis in rotating machinery. Additionally, its inclusion of multiple working conditions and a range of fault severities allows for a comprehensive evaluation of the model's performance across varying scenarios.

In this study, a cross-domain analysis is conducted where models were trained on load 1 (L1) and load 2 (L2) files, and tested on load 3 (L3) files, as shown in TABLE I. For classification, three ML algorithms are tested: SVM, KNN, and RF. To enhance model performance, the vibration signals were divided into windows of 1024 and 2048 data points. Each window is applied using either raw data as direct input features or preprocessed data using the Fast Fourier Transform (FFT) and Root Mean Square (RMS) to extract frequency-domain and time-domain features, respectively. Additionally, training and testing were also conducted without preprocessing, using only 1024 and 2048 raw segment windows as inputs.

TABLE I. FILE NAMES USED FOR TRAINING AND TESTING

Domain	*Fault Type*	*Training*	*Testing*
7_L3	Ball	7_BA_l1 7_BA_l2	7_BA_l3
	Outer Race	7_OR1_l1 7_OR1_l2	7_OR1_l3
	Inner Race	7_IR_l1 7_IR_l2	7_IR_l3
14_L3	Ball	14_BA_l1 14_BA_l2	14_BA_l3
	Outer Race	14_OR1_l1 14_OR1_l2	14_OR1_l3
	Inner Race	14_IR_l1 14_IR_l2	14_IR_l3
21_L3	Ball	21_BA_l1 21_BA_l2	21_BA_l3
	Outer Race	21_OR1_l1 21_OR1_l2	21_OR1_l3
	Inner Race	21_IR_l1, 21_IR_l2	21_IR_l3

B. *Created GPT Description*

A GPT model is created to perform bearing fault diagnosis, as seen in Fig. 1. The GPT can analyze different data types, such as vibration, temperature, and acoustic signals, to detect and classify faults. To optimize the model's performance, detailed instructions and prompts are given for selecting and evaluating the best algorithms, such as RF, SVM, and KNN. Despite these optimizations, the model primarily relies on traditional ML techniques rather than deep learning. See the Appendix for the prompts given to the GPT.

Training was conducted using segmented data, where each file was divided into 1024 and 2048 segments to ensure consistent batch processing.

The proposed GPT is configured to display performance metrics, including test accuracy for each class and algorithm in a table format. The test results were based on different ML models, focusing on fault classification performance across the segmented data. The GPT used for this project can be accessed via the following link:

https://chatgpt.com/g/g-67a7ab766f90819192ce9e9ca3107a01-bearing-diagnosis

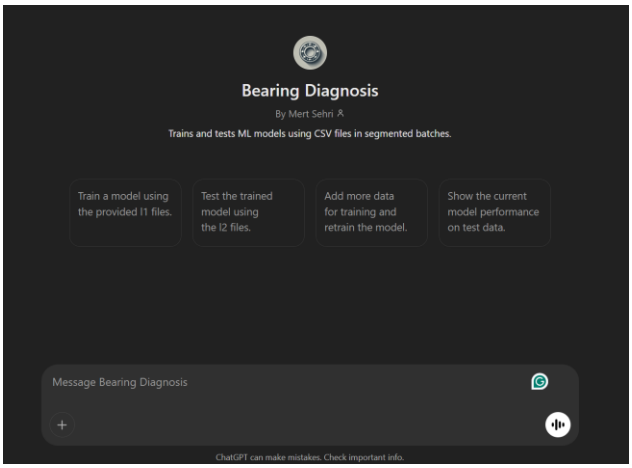

Figure 1. Bearing Fault Diagnosis GPT

C. *Prompts Given*

To ensure a clear and structured evaluation, specific prompts for training and testing the GPT are created, adjusting each prompt to define the necessary details for every cycle. The GPT was instructed to train models using L1 and L2 files and evaluate test accuracy on L3 files. Additionally, files are uploaded for domains 7, 14, and 21 separately, ensuring that each domain had its own dedicated prompt for training and testing. The ML algorithms used were SVM, RF, and KNN, with the GPT explicitly asked to report test accuracy as a percentage for each class. Since the GPT consistently defaulted to using RF, the authors made sure to specify the need to run SVM and KNN to ensure a balanced evaluation.

Preprocessing was applied in two distinct ways. One prompt instructed the GPT to preprocess 1024 samples of data, while another focused on 2048-segment data. Similarly, one prompt applied FFT, while another applied RMS, ensuring that each preprocessing method was tested independently for clear assessment. When handling raw data, no preprocessing instructions were provided, allowing GPT to process the data as-is.

By using these structured prompts, consistency is maintained in the training and testing process, ensuring reliable model performance, and enhanced overall evaluation accuracy.

V. RESULTS AND DISCUSSION

7_L3, 14_L3, and 21_L3 refer to the domains where the model was trained on L1 and L2 and tested on L3 for fault sizes of 0.007, 0.014, and 0.021 inches, respectively. Training and testing were conducted separately for ball, inner race, and outer race faults. The average accuracy for each testing condition, as summarized in TABLE II. , provides an overall assessment of model performance. Additionally, TABLE II. presents the average accuracy for 1024 and 2048 segments without any preprocessing, offering further insight into model behavior under different segment sizes.

TABLE II. provides the average results for each of the SVM, RF, and KNN models. The models were trained on bearing fault classes, including ball, inner race, and outer race. Regarding these results, it is evident that applying a FFT generally improves accuracy, but for the 7_L3 domain, RMS achieves higher accuracy. Additionally, the 2048 sample segment size consistently achieved better accuracy than the 1024 segment size.

TABLE II. TRAINED USING LOAD 1 AND LOAD 2, TESTED USING LOAD 3 WITH 1024/2048 SEGMENT SIZES USING FFT AND RMS PREPROCESSING

Domain Tested	*Segment Size and Preprocessing*	*SVM %*	*Random Forest %*	*KNN %*	*Computation Time (seconds)*
7_L3	1024 Raw	54.00	61.00	43.66	10
	1024 FFT	84.00	88.33	89.33	20
	1024 RMS	97.33	76.33	79.33	20
	2048 Raw	94.85	69.11	98.60	10
	2048 FFT	97.30	88.42	99.10	20
	2048 RMS	99.58	90.65	92.81	20
14_L3	1024 Raw	75.55	88.00	67.40	10
	1024 FFT	74.73	69.77	86.40	30
	1024 RMS	44.02	36.78	39.71	30
	2048 Raw	91.83	96.76	86.75	10
	2048 FFT	91.72	78.81	89.99	30
	2048 RMS	48.68	35.96	36.94	30
21_L3	1024 Raw	69.48	55.61	63.12	10
	1024 FFT	66.66	73.05	67.15	30
	1024 RMS	40.09	39.34	39.28	20
	2048 Raw	65.24	64.98	50.94	10
	2048 FFT	93.24	99.15	91.41	20
	2048 RMS	23.84	44.08	43.38	20

A. *Domain Accuracy Results*

Model performance exhibited variability based on both the segmentation size and the feature extraction method used. The key findings for each domain are summarized as follows:

Domain 7-L3:

- Accuracy was highest when using a KNN with 2048 FFT, achieving 99.10% accuracy.
- The RMS feature extraction for 1024 rows led to a lower performance for SVM (97.30%) compared to using a FFT.
- Even though raw data without preprocessing was tested, 2048-raw segments worked better than 1024-raw segments, but the application of preprocessing methods still yielded superior accuracy.
- All methods consistently completed their computations within 20 seconds.

Domain 14-L3:

- Accuracy dropped significantly when using RMS features, with the SVM achieving only 44.02% accuracy.
- FFT features yielded better results, with the SVM reaching 91.72% accuracy when using 2048-raw segments.
- Interestingly, raw data with 2048-raw segments provided results that were nearly as good as those achieved when using preprocessing methods.

Domain 21-L3:

- FFT features, with a 2048 segment size, delivered the best accuracy (99.15%) when using RF.
- RMS features with 1024 raw data resulted in poor performance across all algorithms, with the SVM at 40.09%.
- Raw data without preprocessing did not perform well in this domain. Higher accuracy was only achieved after applying preprocessing methods like the FFT and RMS.
- Computation times varied between 20 and 30 seconds depending on the size of the segment.

Fig. 2 compares the classification accuracies of SVM, Random Forest, and KNN across domains 7-L3, 14-L3, and 21-L3 using different segment sizes and preprocessing methods. The results highlight the superiority of the FFT-based preprocessing method, especially at a segment size of 2048, over RMS. RF and KNN consistently achieved the highest accuracy, making them the most reliable models, while SVM struggled, particularly with RMS. These findings emphasize the importance of frequency-domain feature extraction and selecting appropriate models for robust fault classification when using LLMs.

B. *Computation Time Observations*

The computation time remained almost the same across different configurations, with most methods completing in 20 seconds. However, there were instances where the GPT faced issues, causing the process to take up to 60 minutes. Issues like GPT losing track of the process and requiring file reuploads, encountering errors due to inconsistent column counts across

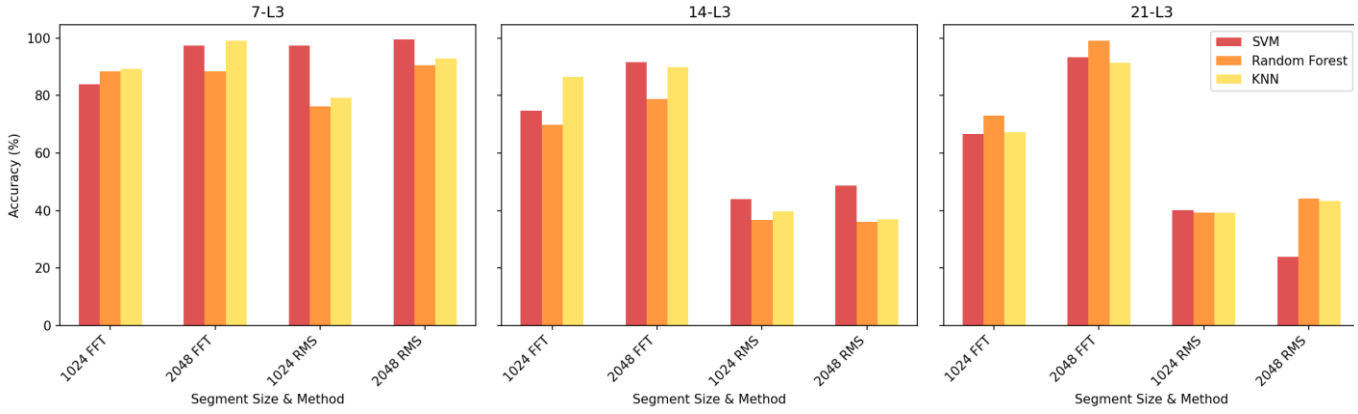

Figure 2. Classification Accuracies for Different Segment Sizes and Fault Sizes

data chunks, or detecting invalid numeric data that resulted in an empty dataset during feature extraction contributed to these delays. Analyzing the shape of the chunks and identifying the problem required additional time, sometimes extending beyond 20 seconds. Despite these challenges, the GPT consistently resolved the problems on its own without external intervention.

VI. Conclusion

This paper explored the application of ChatGPT, a LLM, in bearing fault diagnosis using traditional ML techniques. By using the CWRU dataset, this paper demonstrated that feature extraction plays a crucial role in fault classification accuracy when using LLMs. Among the techniques tested, FFT-based features consistently yielded the highest accuracy across different domains and segment sizes, emphasizing the importance of frequency-domain analysis for diagnosing mechanical faults. The performance was particularly notable with a segment size of 2048 data points, which provided improved results compared to smaller segments.

These findings also highlight the capability of ChatGPT to conduct fault diagnosis with minimal computational resources, overcoming the need for specialized hardware typically required by traditional ML methods. Despite these advantages, the approach showed sensitivity to feature selection and data segmentation, underscoring the need for further optimization in real-world applications. Additionally, ChatGPT's current limitations, such as the lack of support for deep learning frameworks and constraints on large data processing, restrict its ability to fully automate advanced diagnostic tasks.

References

[1] Y. Li et al., "Large Language Models for Manufacturing," Oct. 28, 2024, arXiv: arXiv:2410.21418. doi: 10.48550/arXiv.2410.21418.

[2] L. Tao, H. Liu, G. Ning, W. Cao, B. Huang, and C. Lu, "LLM-based framework for bearing fault diagnosis," Mechanical Systems and Signal Processing, vol. 224, p. 112127, Feb. 2025, doi: 10.1016/j.ymssp.2024.112127.

[3] A. Boonmee, K. Wongsuwan, and P. Sukjai, "Consultation on Industrial Machine Faults with Large language Models," Oct. 04, 2024, arXiv: arXiv:2410.03223. doi: 10.48550/arXiv.2410.03223.

[4] A. Bahrini et al., "ChatGPT: Applications, Opportunities, and Threats," in 2023 Systems and Information Engineering Design Symposium (SIEDS), Apr. 2023, pp. 274–279. doi: 10.1109/SIEDS58326.2023.10137850.

[5] I. Kostka and R. Toncelli, "Exploring Applications of ChatGPT to English Language Teaching: Opportunities, Challenges, and Recommendations," TESL-EJ, vol. 27, no. 3, 2023, Accessed: Jan. 23, 2025. [Online]. Available: https://eric.ed.gov/?id=EJ1409872

[6] S. S. Biswas, "Role of Chat GPT in Public Health," Ann Biomed Eng, vol. 51, no. 5, pp. 868–869, May 2023, doi: 10.1007/s10439-023-03172-7.

[7] M. Sehri, M. Ertarğın, A. Orhan, Ö. Yildirim, and P. Dumond, Deep Learning Approach to Bearing and Induction Motor Fault Diagnosis via Data Fusion. 2024.

[8] G. Vashishtha et al., "A roadmap to fault diagnosis of industrial machines via machine learning: A brief review," Measurement, vol. 242, p. 116216, Jan. 2025, doi: 10.1016/j.measurement.2024.116216.

[9] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "The Computational Limits of Deep Learning," Jul. 27, 2022, arXiv: arXiv:2007.05558. doi: 10.48550/arXiv.2007.05558.

[10] "ChatGPT." Accessed: Jan. 23, 2025. [Online]. Available: https://chatgpt.com

[11] "Welcome to the Case Western Reserve University Bearing Data Center Website | Case School of Engineering." Accessed: Feb. 10, 2025. [Online]. Available: https://engineering.case.edu/bearingdatacenter/welcome

Appendix

The following prompts were used with ChatGPT to create the Bearing Fault Diagnosis GPT:

- Can you make a Random Forest classifier, SVM and KNN and train using L1 and L2 files, and test using L3 files. I will be using this GPT to train my data, which will be provided. Additionally, allow more data to be added for training please. Make sure to segment each .csv file so that you can take 1024 rows at a time and ignore the first row as that is the title of each column. The first column is the accelerometer raw data. For each segment from the csv file, use 100 instances of 1024 to train and test. Only train using L1 and L2 files and test using L3 files. In the future I will add more files and give different commands for training. (For creating the GPT, only 7_L3 domain dataset files were attached, as shown in TABLE I.)
- Can you give me the test accuracy for L3 and train using L1 and L2 using SVM, Random Forest classifier and KNN for each class. I want the test accuracy as a % for each class please. Use 2048/1024 segments.
- Can you please give me the test accuracy for l3 and train using l1 and l2 using SVM, random forest classifier and KNN for each class. I want the test accuracy as a % for each class please. Preprocess the 2048/1024 segments taking an FFT (Fast Fourier Transform) / RMS (Root Mean Square) of the signal.