Simpler Machine Learning Methods Outperform Deep Learning in Motor Fault Detection

Antonio C. Briza[1,2], Eduardo Jr Piedad[1,3], Elmer C. Peramo[1]
[1]*Advanced Science and Technology Institute, DOST, Quezon City 1101 Philippines*
[2]*Asian Institute of Management, Makati 1229 Philippines*
[3]*Universitat Politècnica de Catalunya, Barcelona 08028 Spain*

***Abstract*—In motor condition monitoring, deep learning techniques have been considered by using 2-dimensional plots as datasets instead of time-series signals. For example, a Convolutional Neural Network (CNN) can be trained using recurrence or frequency-occurrence plots. While previous studies demonstrated promising results using CNNs, the lack of discernible differences in the plots rendered the model's inner workings seem like a black box. This study applies ten traditional machine learning (ML) techniques and compares them with recent deep learning (DL) techniques used in motor fault diagnosis using the same dataset. The synthetically prepared motor current signal dataset with 3,750 samples has five classes – healthy and four faulty motor conditions, under five loading conditions – 0, 25, 50, 75, and 100%. After similar training and testing phases, the light gradient-boosting machine (LightGBM) showed the best overall classification accuracy of 93.20%, significantly outperforming by at least 10.4% the three CNN-based models, which obtained performances that range from 74.80 to 82.80%. LightGBM also has the best average performances in other metrics, such as F1 score, precision, and recall. Five out of ten ML models performed better than these three CNN-based models. Given the excellent performance of traditional ML models such as LightGBM, care and consideration has to be taken in the use of deep learning architectures especially since they are more computationally expensive and memory-intensive because there seems to be no guarantee that they will perform better than traditional models, especially on simpler problems, such as the motor fault classification using current signals that we have presented in this paper.**

***Index Terms*—motor fault, machine learning, convolutional neural network, deep learning, transfer learning, LightGBM**

I. Introduction

Machine fault diagnosis plays an important role in the industry. Accurate and timely identification of faulty machines can help minimize downtime in production and operations, allowing operators to react immediately and avoid delays and downtime. Due to its ability to recognize patterns and make informed predictions, artificial intelligence has been used for machine fault diagnosis. The studies of [1] and [2] reviewed the use of machine learning (ML), deep learning (DL), and transfer learning (TL) in detecting motor faults. The works of [3] explored the application of DL techniques in process monitoring. The current popularity of DL or TL methods may lead researchers to set aside traditional ML methods. This study compares traditional machine learning techniques with deep learning in motor fault detection.

A. Limitations of Image-Based Approaches Deep Learning

Two studies on motor fault classification that made use of deep learning techniques are [4] and [5]. Reference [4] obtained an overall performance accuracy of 80.25% in motor fault detection while considering loading conditions. The other study [5] used transfer learning or, specifically, pretrained Convolutional Neural Networks (CNNs) using the same dataset. This latter study improved the performance a little to 82.80%. The limitations of these two studies – the low overall accuracy for both, and the inclusion of loading information in the latter, led the authors of this paper to explore other methods to improve performance. In addition, the developed two-dimensional plots used by the earlier studies tend to be less interpretable than the typical one-dimensional Fourier transformed signals.

This study performs a comparative study of some traditional machine learning techniques and applies them to original motor signals in particular. It also considers developing a CNN model based on recurrence plots by [6] from the original current signals. In this way, a more detailed comparison will be performed.

B. Machine Learning Methods

The traditional machine learning classifiers used in this study are logistic regression, k-nearest neighbors (KNN), Naive Bayes, decision trees, random forest [7], support vector machines (SVM) [8], multilayer perceptrons (MLP), extra trees [9], LightGBM [10], and AdaBoost [11].

Logistic regression excels at interpretability due to its linear decision boundary. However, its performance may be better for complex, non-linear relationships between features.

KNN is non-parametric classifier that predicts a data point's class based on the majority class of its K nearest neighbors in the training data. While simple to implement and requiring minimal parameter tuning, its accuracy is sensitive to the choice of K and the presence of irrelevant features.

Naive Bayes's simplicity and efficiency make it ideal for large datasets with categorical features. However, its feature independence assumption can limit its accuracy when features are highly correlated. To classify a data point, it calculates the posterior probability of each class based on the prior probabilities and the likelihood of the data point's features belonging to each class.

Decision trees provide interpretability through their decision rules, built on a series of splits separating data points from different classes. They tend to be robust to irrelevant features but can be sensitive to the chosen splitting criteria and prone to overfitting without proper pruning.

Random Forest [7], an ensemble classifier, combines the strengths of multiple decision trees, boosting accuracy through diversity. It trains each tree on a different subset of the data and predicts the class by averaging their individual predictions. While inheriting interpretability from individual trees, its training can be computationally expensive compared to single decision trees.

SVMs [8] are known for their robustness to outliers and effectiveness for high-dimensional data. They search for a hyperplane that maximizes the margin between different classes in the training data. This hyperplane then forms the decision boundary for new data points. While robust and effective, SVMs have less interpretable decision boundaries, and their training can be computationally expensive for large datasets.

MLP consist of interconnected layers of neurons that process information through weighted connections, allowing them to learn intricate relationships. However, MLPs require significant training data and careful hyperparameter tuning to avoid overfitting and improve generalizability, making them less interpretable than other classifiers.

Extra trees [9], also known as extremely randomized trees, are an ensemble learning method that uses multiple decision trees for classification and regression tasks. They are similar to random forests but with some key differences.

LightGBM [10], short for Light Gradient Boosting Machine, is a high-performance, distributed gradient boosting framework for various machine learning tasks. It is fast and efficient due to its Gradient-Based One-Side Sampling (GOSS). GOSS weights data points with larger gradients higher while calculating the gain. In this method, instances that have not been used well for training contribute more. LightGBM also uses exclusive feature bundling.

AdaBoost [11], short for Adaptive Boosting, is an ensemble learning algorithm in machine learning. It is used for classification and regression tasks, but it's most commonly associated with classification. The main idea behind AdaBoost is to combine the predictions of weak learners (typically simple and not very accurate models) to create a strong learner that performs well on the overall task.

These traditional algorithms often perform better with small datasets. They also seem to require less computational power and are generally easier to interpret. When it comes to inference speed, they tend to be faster since their architecture is simpler than the deep learning (DL) models. Lastly, these methods can be less prone to overfitting, a common issue when using deep learning on small datasets. This study investigates their performance in motor fault diagnosis and compares them with previously developed DL models.

TABLE I: Test motor specifications

Label	Parameter/Value
Type	AEEF-90-4 Induction motor
Output	2 HP
Pole	4
Insulation	E
Volt	220/380 V
Amp	7 A
r.p.m.	1450/1720
Duty type	S1
Cycle (Hz)	50/60
Connection	delta low voltage/ wye high voltage
Manufacturer	TW-141221 Tai Wei Electric Factory, Ltd.

TABLE II: Motor Conditions and Labels

Motor Condition	Label
Bearing Axis Misalignment	Fault 1
Stator Inter-Turn Short Circuiting	Fault 2
Broken Rotor Strip Fault	Fault 3
Outer Ring Bearing Fault	Fault 4
No Fault	Healthy

II. Test Motor and Dataset

Current signal data is collected from motors with specifications described in Table I, which is reproduced here from [4] for easier reference. Concretely, data is collected from a total of five motors. One motor is healthy, while the other four motors have the following synthetically induced faults: bearing axis misalignment, stator inter-turn short circuiting, broken rotor strip fault and outer bearing fault. These 5 motor conditions serve as the labels of the dataset. A summary is given in Table II.

Moreover, each motor is sampled under five loading capacities: 0, 25%, 50%, 75% and 100%. For each motor condition and loading capacity, 150 samples of five-second current signals are collected, bringing the total dataset to 3750 items, having 750 samples for each of the five classes. A preprocessing step on this dataset was performed before using a CNN to classify them. A percentile clipping of 90% with log normalization was applied to the data before performing a Discrete Fourier Transform (DFT) on them. Finally, Frequency Occurrence Plots (FOPs) are produced which transformed these one-dimensional signal data into two-dimensional images. The FOP images are then fed to a CNN for classification. A visual comparison of sample FOP and recurrence plots with the DFT is shown in Fig. 2.

Frequency Occurrence Plots (FOPs) are visual representations of time series data, where each grid cell shows the number of times a specific combination of frequency and amplitude occurs. This visual format allows for easy identification of patterns, facilitates feature extraction for machine learning, and serves as a data transformation tool for algorithms like CNNs. FOPs are useful for analyzing motor current signals for fault diagnosis, extracting speech recognition and image processing features, and detecting anomalies in time series data. The FOP images of this dataset were published and made publicly available in [12]. The study of [5] used this published

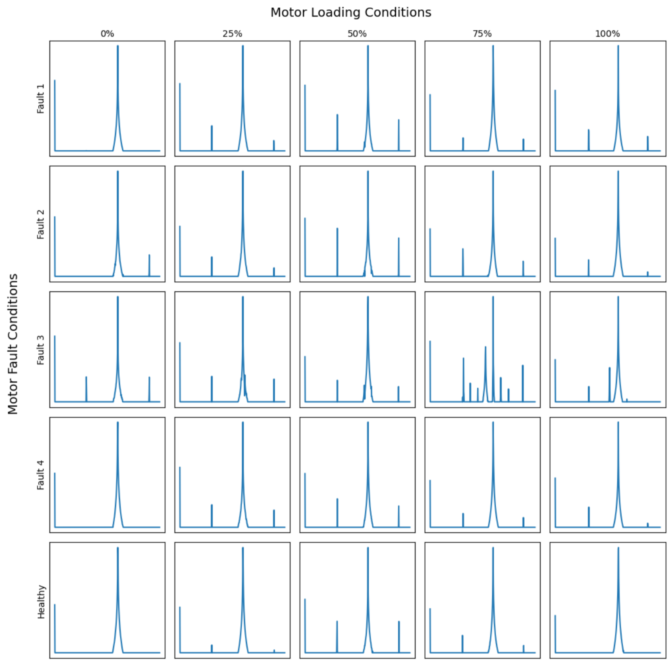


Fig. 1: Sample features used by the traditional machine learning methods in this study. The y-axis values range from 0 to 1 signifying signal strength while the x-values range from 0 to 500 Hz . Note that these images are only for illustration purposes. The actual data fed to the ML methods is a one-dimensional vector instead of an image.

dataset in their work wherein they used a transfer learning strategy to classify the motor faults.

This study used the raw current signals from the study in [4], and, similar to it, applied all the previous preprocessing steps except for the conversion to FOP images. All the traditional machine learning techniques in this study used that preprocessed data. Fig. 1 illustrates a sampling of the preprocessed data for all classes under all loading conditions.

III. Methodology

This study presents two separate analyses of the motor fault dataset: first, it applied a number of traditional machine learning techniques on the preprocessed data as described in the latter part of the previous section, and, second, it applied a CNN to the Recurrence Plots produced from the preprocessed data. For the ML methods, a grid search is done to find the best set of parameters. This set of best parameters is then evaluated using a stratified 5-fold cross-validation to obtain the accuracy performance of the model. Finally, the best model is run on the dataset on an 80% to 20% training and test split.

TABLE III: CNN network architecture used in classifying the Recurrence Plots of the dataset

Layer	Size/rate	Filters/Nodes	Activation
Convolution + Batch Norm	(3x3)	32	relu
Max Pooling	(2x2)	-	-
Convolution + Batch Norm	(3x3)	32	relu
Max Pooling	(2x2)	-	-
Dropout	0.25	-	-
Flatten	-	-	-
Dense	-	256	relu
Dropout	0.4	-	-
Dense	-	5	softmax

TABLE IV: Accuracy on entire data set for all loading conditions

Method	Code	Accuracy (%)
LightGBM	LGBM	93.20
K-Nearest Neighbors	KNN	89.90
Random Forest	RF	88.00
AdaBoost	ADA	84.50
Multi Layer Perceptron	MLP	84.10
Pretrained CNN on FOP [5][1]	PTCNN-FOP	82.80
Logistic Regression	LR	80.50
CNN on FOP [4]	CNN-FOP	80.25
CNN on RP	CNN-RP	74.80
Support Vector Machines	SVM	74.50
Decision Tree	DT	66.00
Extra Tree	XT	64.30
Gaussian Naïve Bayes	GNB	47.50

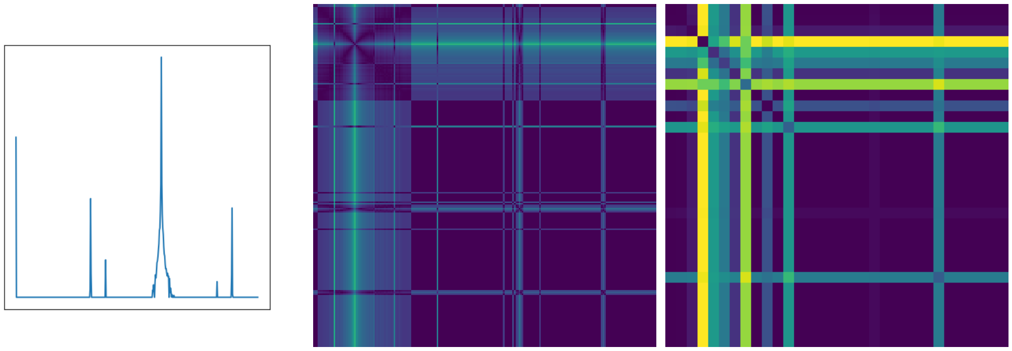
[1] Accuracy is taken as the average on all loads for Type 4 Classification reported in the paper.

For the second analysis, Recurrence Plots (RPs) [13] are produced from the preprocessed data and are fed to a CNN with the architecture shown in Table III. The architecture is based on [14], which used CNNs on RPs to predict rainfall severity. Similar to that study, the cosine loss function and Stochastic Gradient Descent with Warm Restarts are used in the training of the network. A sample RP image used in this study is provided in Fig. 2c.

A Recurrence Plot is a visual and quantitative tool for analyzing the dynamic behavior of systems by exploring how often they revisit states in their phase space. They reveal recurring patterns, assess system stability and complexity, and provide insights into synchronization and interaction between multiple systems. To create an RP, time series data is first embedded into a higher-dimensional phase space, representing the system's state at each time point. Then, recurrence calculation compares each pair of states in the phase space and places a point in the RP matrix if they are similar. Finally, the visualization of the RP matrix shows recurring states as points, while empty cells indicate states that never recur. This allows for easy identification of patterns and quantification of the system's behavior, making RPs valuable tools in various fields.

IV. Results and Discussion

Table IV and Fig. 3 show the performance of the analyses described above, together with the previous studies of [4] and Nandi et al. [5]. It can easily be seen that five traditional models (LightGBM, K-Nearest Neighbors, Random Forest, AdaBoost and Multilayer Perceptron) outperformed the two previous studies. It is worth noting that the best model, LightGBM, made a 16.13% improvement from the first study and a 12.62% improvement from the second study while using only one classifier and without using loading conditions. Compared with the performance of CNN on FOPs (shown in



(a) Discrete Fourier Transform   (b) Frequency Occurence Plot   (c) Recurrence Plot

Fig. 2: Illustration of the different preprocessed datasets used by the methods in this study. The images represent one sample of a healthy motor with a loading capacity of 75%.

TABLE V: Performance[1] of CNN and traditional ML Methods.

Metric	Class	CNN-RP	CNN-FOP [4]	LR	PTCNN-FOP [5][2]	MLP	ADA	RF	KNN	LGBM
F1	Fault 1	83.90	90.00	56.10	**91.70**	76.10	77.90	81.80	84.40	90.10
	Fault 2	60.40	76.60	64.20	91.85	81.40	82.60	87.40	91.50	**92.90**
	Fault 3	78.20	81.00	76.90	84.25	75.00	90.30	90.70	93.70	**94.70**
	Fault 4	67.40	68.40	51.60	**91.70**	66.90	76.40	80.50	83.10	91.00
	Healthy	83.80	84.51	69.90	70.23	84.30	85.00	87.70	94.30	**98.20**
	Average	74.74	80.25	63.74	88.47	76.74	82.44	85.62	89.40	**93.38**
Precision	Fault 1	84.50	93.40	54.90	**96.36**	73.00	71.60	76.80	78.50	87.00
	Fault 2	78.80	71.20	65.80	**97.10**	85.40	86.80	91.40	96.00	95.00
	Fault 3	89.07	78.00	**99.00**	86.60	76.60	94.20	94.30	94.00	95.30
	Fault 4	55.51	75.80	50.30	**96.36**	65.10	79.90	82.10	84.80	91.60
	Healthy	82.97	84.67	60.80	58.63	84.30	81.20	84.20	94.30	**97.90**
	Average	78.17	80.25	66.16	87.01	76.88	82.74	85.76	89.52	**93.36**
Recall	Fault 1	83.40	86.60	57.40	87.99	79.40	85.30	87.50	91.20	**93.40**
	Fault 2	48.90	83.00	62.70	87.33	77.70	78.90	83.70	87.30	**91.00**
	Fault 3	69.70	84.60	62.90	87.33	73.50	86.80	87.40	93.40	**94.00**
	Fault 4	86.00	62.60	52.90	87.99	68.80	73.20	79.00	81.50	**90.40**
	Healthy	84.78	88.50	82.10	91.33	84.30	89.30	91.40	94.30	**98.60**
	Average	74.56	81.00	63.60	88.40	76.74	82.70	85.80	89.54	**93.48**

[1] Best scores are in bold.
[2] It should be noted that [5] employed a different classification method. As the authors performed classification with loading information, the values we present in the table is the average of what they reported across the different loads. Moreover, the authors performed four types of classification: Type 1: Healthy vs. Fault 1 and Fault 4; Type 2: Healthy vs. Fault 2; Type 3: Healthy vs. Fault 3; and Type 4: Healthy vs. Faults 1 & 4, Fault 2 and Fault 3. For this reason, in the table, the values of the rows for Fault 1 and Fault 4 are the same. Finally, the row values for Healthy are taken Type 4 Classification.

Fig. 4), the confusion matrix of LightGBM (shown in Fig. 5) has excellent fault classification accuracies, achieving a value higher than 90% on all labels. This strongly suggests an excellent overall performance on the whole data set and on each of the individual classes. As regards the second analysis we employed in this paper — using CNN on RP — we can see from Table IV that it was able to approach the performance of both [4] and [5] but not surpass them. Six machine learning methods outperformed it.

A detailed performance report is given in Table V. It lists the two previous studies together with the traditional ML methods and CNN on RP method used in this study. From the table, it can be clearly seen that LightGBM obtained the highest average F1, precision, and recall values. Additionally, it almost obtained the highest values for all metrics and classes, except for 5 items where it was beaten by PTCNN-FOP [5] and 1 item where it was beaten by Logistic Regression. However, LightGBM's performance for these items was still very close, except for the precision score for Fault 1. It is worth emphasizing that the overall performance of LightGBM is more balanced than both PTCNN-FOP and LR, showing its dominance on both methods.

The underperformance of the CNN methods may not be attributed to a lack of optimization of the architecture of the CNNs. For example, the CNN-RP method presented in this study was the best among multiple runs that explored various hyperparameter values for the network architecture and recurrence plots. A visual inspection of FOPs and RPs used

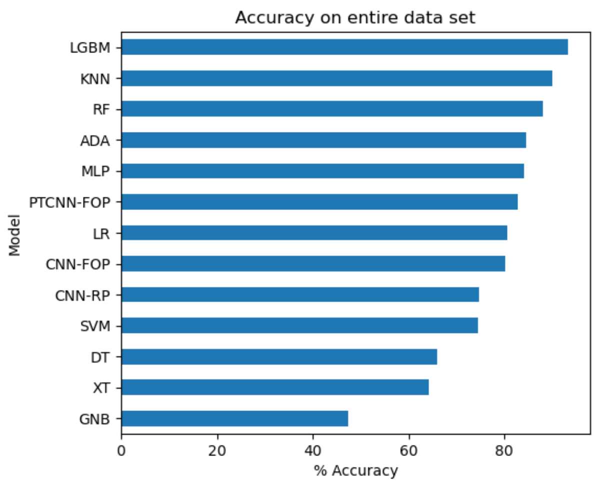


Fig. 3: Accuracy of the different methods

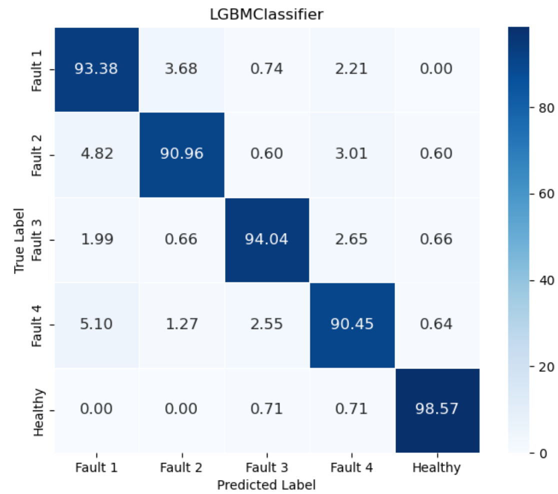


Fig. 5: Confusion matrix (in percentage) of the best performing ML model - LightGBM

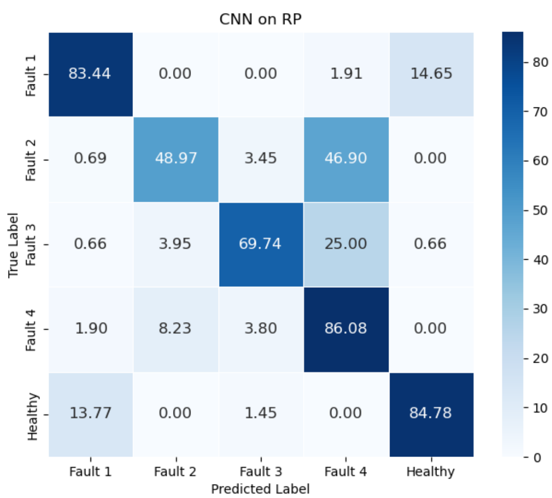


Fig. 4: Confusion matrix (in percentage) of the CNN applied to Recurrence Plots [4]

by the CNNs shows similarities across images that are difficult even for a human to distinguish. Moreover, the inductive bias of CNNs is meant to take advantage of spatial hierarchies and translation invariance. These are valuable for photographic images but may not necessarily be advantageous when applied to signals where peaks and valleys can easily be observed and identified at specific locations, such as in the case of frequency values in a Fourier Transform. CNNs can still be advantageous when dealing with large datasets or complex relationships within the data. However, for cases with limited data, like the motor fault diagnosis in this study, simpler models may offer a better option in terms of interpretability, efficiency, and even accuracy.

V. Conclusion

In this paper, we investigated the performance of traditional machine learning models, convolutional neural networks and transfer learning for the classification of motor faults. The results show that LightGBM had the best performance overall, obtaining an average accuracy of 93.2% on a stratified 5-fold cross-validation on all classes and loading conditions. This result outperformed all methods in terms of average F1, precision, and recall in all classes. This demonstrates that machine learning methods are still relevant in analyzing simple data and could provide a faster and more efficient solution to less complex problems. One limitation that should be mentioned is that feature engineering should be carefully done. If feature engineering is done well, it may not be necessary to employ more complex methods such as CNNs and transfer learning especially on simpler problems. As the advancement and popularity of deep learning and transfer learning continue, the consideration of traditional machine learning methods should not be overlooked, especially if faster and more explainable methods are sought.

Several related future research in this area can be mentioned. First, to conduct similar comprehensive comparison studies using other dataset on motor condition monitoring. Second, to consider other plot-based transformations such as short-term Fourier transform and wavelet transforms as inputs to deep learning methods such as [15] that used temporal Convolutional Neural Network with an attention mechanism. Third, to explore other deep learning techniques aside from plot-based techniques. Finally, to undertake a more thorough theoretical analysis on why traditional ML methods could outperform DL methods on specific applications or datasets.

VI. Acknowledgment

This study is supported by Project ASTI-ALaM of the Department of Science and Technology - Advanced Science and Technology Institute (DOST-ASTI). Mr. Piedad is supported by Project MaSense of DOST-ASTI and DOST S&T Fellows Program.

References

[1] 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.

[2] J. Cen, Z. Yang, X. Liu, J. Xiong, and H. Chen, "A review of data-driven machinery fault diagnosis using machine learning algorithms," *Journal of Vibration Engineering & Technologies*, vol. 10, no. 7, pp. 2481–2507, 2022.

[3] S. Heo and J. H. Lee, "Fault detection and classification using artificial neural networks," *IFAC-PapersOnLine*, vol. 51, pp. 470–475, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:53316652

[4] E. Piedad Jr, Y.-T. Chen, H.-C. Chang, C.-C. Kuo *et al.*, "Frequency occurrence plot-based convolutional neural network for motor fault diagnosis," *Electronics*, vol. 9, no. 10, p. 1711, 2020.

[5] A. Nandi, S. Biswas, K. Samanta, S. S. Roy, and S. Chatterjee, "Diagnosis of induction motor faults using frequency occurrence image plots—a deep learning approach," in *2019 International Conference on Electrical, Electronics and Computer Engineering (UPCON)*. IEEE, 2019, pp. 1–4.

[6] J.-P. Eckmann, S. O. Kamphorst, D. Ruelle *et al.*, "Recurrence plots of dynamical systems," *World Scientific Series on Nonlinear Science Series A*, vol. 16, pp. 441–446, 1995.

[7] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001. [Online]. Available: https://api.semanticscholar.org/CorpusID:89141

[8] C. Cortes and V. N. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995. [Online]. Available: https://api.semanticscholar.org/CorpusID:52874011

[9] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, pp. 3–42, 2006. [Online]. Available: https://api.semanticscholar.org/CorpusID:15137276

[10] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[11] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[12] E. J. Piedad, Y.-M. Hsueh, C.-C. Kuo, and H.-C. Chang, "Frequency occurrence plots for motor fault diagnosis based on image recognition," 2019. [Online]. Available: https://dx.doi.org/10.21227/77da-c563

[13] A. L. Moraes and R. A. S. Fernandes, "Recurrence plots: A novel feature engineering technique to analyze power quality disturbances," in *2020 IEEE Power Energy Society General Meeting (PESGM)*, 2020, pp. 1–5.

[14] S. A. A. Ojeda, G. A. Solano, and E. C. Peramo, "Multivariate time series imaging for short-term precipitation forecasting using convolutional neural networks," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE, 2020, pp. 33–38.

[15] Z. Xu, D. Yu, and Y. Hu, "Motor fault diagnosis method based on deep learning," in *2022 11th International Conference of Information and Communication Technology (ICTech))*, 2022, pp. 236–239.