Interpreting Fault Prediction in Induction Motors Using Explainable Artificial Intelligence

SERKAN AKIN, University of Twente, The Netherlands

This research explores the effectiveness of machine learning (ML) and deep learning (DL) models in fault prediction of induction motors, focusing on fault detection through the analyzes of sensor data. The study examines the performance of gradient boosting and feedforward neural networks. Both models are evaluated on their ability to classify the health status of induction motors, using data provided by Fraunhofer Innovation Platform at the University of Twente.

Key to this research is the integration of Explainable Artificial Intelligence (XAI) methods, specifically SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations), to get insights on the decision-making process of the models. These XAI techniques reveal how specific features influence model predictions, making them transperent for the end-user in industrial settings.

Additional Key Words and Phrases: Explainable AI, Induction Motors, Classification, Fault Detection, Machine Learning, Deep Learning

1 INTRODUCTION

Induction motors play a crucial role in industrial operations due to their efficiency and reliability in industrial operations[3, 6] However, like any critical machinery, they are prone to faults, which may lead to significant down times and economic losses. This has increased the interest in fault detection techniques. Developments in machine learning (ML) and deep learning (DL) are particularly promising, as they offer powerful tools for analyzing complex data and identifying patterns.

Despite the potential of ML and DL in real-time fault classification, their adoption in industrial settings faces considerable challenges. The primary issue is their "black-box" nature, which makes it difficult to understand how decisions are made. This lack of transparency is a significant barrier to trust and usability in environments where understanding and trust in automated processes are crucial.[8]

Explainable Artificial Intelligence (XAI) provides methodologies to reveal the reasoning behind the decision-making process of advanced models. By making the underlaying mechanics of these models more accessible and understandable, XAI can enhance the trust in these tools.[2, 9] This research aims to integrate XAI techniques into fault classification systems, making them transparent and building trust among users. To clarify, the following research question will be the main focus:

RQ: How can Explainable Artificial Intelligence (XAI) techniques be integrated into predictive maintenance models that are targeted at classifying the health status of induction motors using sensor measurements to make decisions given by the models transparent?

TScIT 41, July 5, 2024, Enschede, The Netherlands

This study will utilize raw sensor readings from an experimental setup of an induction motor at Fraunhofer Innovation Platform at the University of Twente to train and evaluate the effectiveness of the applied ML and DL methods. Analysis of the performance metrics of these models and application of popular XAI techniques, such as Shapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) will be made to interpret and understand their decision-making processes.

2 LITERATURE REVIEW

Induction motors are pivotal components in various industrial applications due to their robustness, efficiency, and cost-effectiveness.[6] These motors operate on the principle of electromagnetic induction, wherein an electric current generates torque in the motor's rotating element, known as the rotor. Despite their advantages, induction motors are susceptible to some operational issues, with misalignment being a particularly significant fault that can severely impact their performance.[10] Misalignment in induction motors refers to the scenario where the motor shaft does not properly align with the axis of the driven load[1].

For instance, misalignment increases mechanical stress on the motor's bearings and shafts, accelerating wear and tear and potentially leading to failures. Moreover, misaligned components often lead to vibrations and noise, decreasing overall operational efficiency.[10]

Gradient Boosting is a powerful ensemble learning technique widely used in machine learning in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting algorithms, but it generalizes them by allowing optimization of an arbitrary differentiable loss function This method is widely known for its ability to deal with non-linear data and its robustness against over-fitting.[1]

Neural networks have revolutionized the field with their ability to model complex nonlinear relationships that exist in real-world data. A neural network typically consist of several layers through which data is processed, allowing the model to learn complex features at each layer. This architecture makes them suitable for high-dimensional data, as they can do no require any feature selection or extraction methods for improved performance.[11]

Traditional fault detection methods such as Motor Current Signature Analysis (MCSA) have long been standard in monitoring the state of induction motors. MCSA analyzes a motor's signal at a given instance to detect any irregularities that signify faults. While effective for most cases, MCSA can sometimes miss subtleties in signal anomalies, especially in environments with noisy data where fault indicators are less pronounced.[10]

Recent studies have shown that machine learning and deep learning methods promise improvements over traditional techniques like MCSA. Techniques such as Support Vector Machines (SVM),

[1]The term "driven load" refers to the machinery or equipment that is powered by the motor.

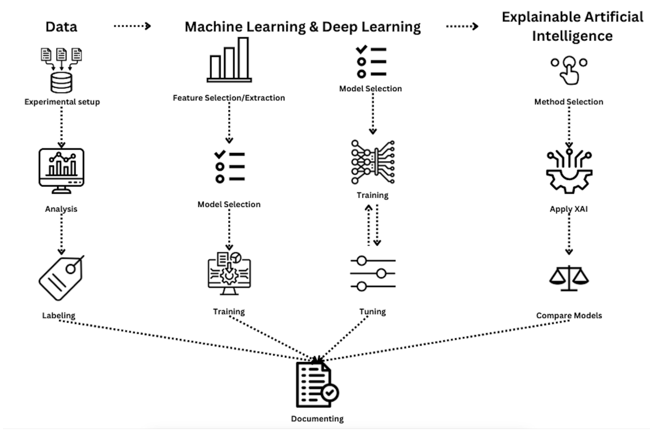

Fig. 1. Methodology Pipeline

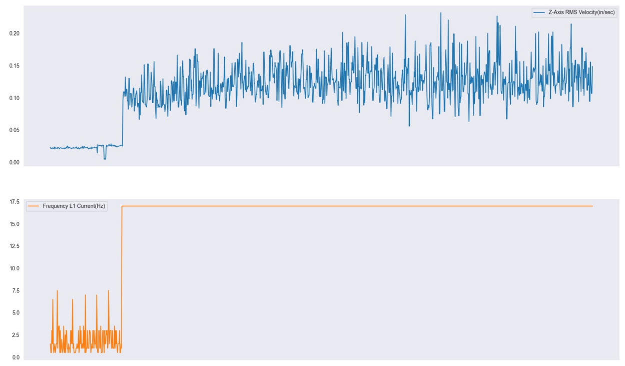

Fig. 2. Example data samples showing downtime and how they effect sensor readings

Artificial Neural Networks (ANN), and Decision Trees have been employed to improve fault detection's accuracy and efficiency. These models can handle large, noisy datasets as well as distinguish between complex fault patterns more effectively than traditional methods.[4, 5]

In a study, SVMs were praised for their ability to classify nonlinear data and manage the challenges posed by non-linear fault dynamics in induction motors.[5] Deep learning models , have also improved precision in these tasks by learning vast amount of data.

The adoption of ML and DL models in fault detection, despite their effectiveness, introduces complexity in understanding model decisions, which is critical for trust and accountability in industrial operations. Explainable Artificial Intelligence (XAI) addresses this challenge by making the outcomes of AI systems transparent and understandable to human operators.[8]

XAI provides deeper insights into classification models giving explanations for why certain decisions or predictions are made. For instance, by applying **SHAP**, stakeholders can understand which features of the data most influence the outcome of a gradient boosting or neural network model.[2] **LIME**, another popular, XAI technique, contributes by approximating the locally behavior of the complex models, providing with explanations for individual predictions made.[7]

3 METHODOLOGY

The steps taken during this research were similar to any AI/XAI pipeline. A structured approach was taken during the research (See Figure 1) which resulted in a smooth and effective research environment.

3.1 Data Processing

3.1.1 Data Collection. As in any ML or DL data collection and analysis was the foundation of the research. Data was sourced from an experimental setup at the Fraunhofer Innovation Platform at the University of Twente, designed to simulate operational scenarios for induction motors under different conditions. At various phases of the data collection, misallignment was intentionally introduced to produce data that can be interpreted as faulty state.

3.1.2 Data Preparation. Raw data collected from these sensors was initially in a format unsuitable for direct analysis. This data underwent a pre-processing phase to convert it into a structured format suitable for ML and DL applications. This process included labeling and combining all data readings from different dates. Exploratory data analysis was conducted using line graphs and histograms. These visual tools were crucial in identifying underlying data patterns, distributions, and potential anomalies, aiding the steps for feature selection.

Outliers can significantly skew with the results of ML and DL models, potentially leading to poor generalizations on unseen data. After initial analysis on the data using the processed graphs data samples with a frequency lower then 10 were removed. The experimental setup was never set to work on a frequency lower then 10 and after analyzing data samples with the graphs (See Figure 2) it was concluded that such areas in the dataset do not represent real world data and might introduce bias in the models. To address this further, Isolation Forests were used to identify and remove outliers. The Isolation Forest method is particularly effective for high-dimensional datasets as it isolates anomalies instead of profiling normal data points. The "contamination" parameter was set to 0.1, indicating an expected proportion of outliers in the dataset. This value was chosen based on preliminary data exploration which suggested that an estimate of %10 of the could be anomalous. Finally, from

In the datasets used for training the models, a significant class imbalance was observed. The amount of faulty condition data (majority class) far exceeded the instances of normal operational data, specifically 67% of the data was faulty . This imbalance can lead models in being biased towards predicting the majority class, meaning that even high accuracies might mask poor performance in detecting less frequent conditions. To observe any changes in results, each model is trained on original unbalanced data and balanced data.

Data balancing is achieved using the Synthetic Minority Oversampling Technique (SMOTE). SMOTE creates synthetic samples from the minority class instead of creating random copies. This is done by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point in that line. This approach helps to create a more diverse and representative data, which could potentially improve generalization capabilities of the models to be trained.

Based on the data analysis, and limitations in scope of this studies, frequency component of the data was ignored from this stage of the project and onwards.

3.1.3 Feature Preparation. Unlike traditional feature engineering for most ML approached, where new features are derived or constructed, this study focused on feature selection. The reason being, that the sensors on the experimental setup providing with extracted features directly during the data collection phase. Some of such features are: "X-Axis Kurtosis" and "X-Axis Crest Factor". Based on insights gained from data visualizations some features were selected for the ML model. Additionally, an ANOVA was conducted to statistically evaluate the impact of different features. This helped in identifying the most significant features that contribute to accurate classifications.

3.2 Model Development

This section outlines the development and configuration of the ML and DL models used in this study: a feed forward neural network and gradient boosting classifier. Each model was chosen for its ability to handle different aspect of the datasets and to explore how various approached impact the classification of fault conditions in induction motors.

3.2.1 Gradient Boosting Classifier. The gradient boosting classifier was selection for its strength in handling tabular data and its effectiveness in improving accuracy by reducing bias and variance through ensemble methods. The configuration of the gradient boosting model included:

· **Number of Estimators:** A total of 100 trees were used, allowing the model to learn from errors of the previous trees and adjust accordingly.

· **Learning Rate:** Set at 0.1, this parameter controls the contribution of each tree to the final outcome, helping to avoid any over-fitting over the training set by making the learning process more gradual.

· **Max Depth of Trees:** Each tree had a maximum depth of 3, which was sufficient to capture the necessary interaction between features.

3.2.2 Feedforward Neural Network . The feedforward neural network implementation in this study is designed to capture and analyze complex patterns in the sensor data from the experimental setup. The network architecture consists of three main layers: input, hidden, and output layers, along with ReLU activation functions. Below is the breakdown the final network's architecture:

· **Input Layer:** The input layer receives the data directly from the features processed during the data preparation stage. It connects to a hidden layer of size as specified by a hidden_dim variable.

· **Hidden Layer:** The network includes two hidden layers. This first hidden layer consists of 32 neurons, followed by a ReLU activation to introduce non-linearity. The second hidden layer contains 16 neurons, also followed by ReLU activation.

· **Output Layer:** The final layer of the network is the output layer, which uses a sigmoid function to output a probability, indicating the likelihood of a fault condition in the induction motor. This is layer is crucial for the binary classification at hand.

The number of neurons in the hidden layer were tuned to get the best results possible. The initial approach was to start with a random number ranging from the size of the input layer and the output layer size. After tuning and training with several combinations the final architecture of the network was decided on. To optimize the training process and avoid overfitting, an early stopping mechanism was implemented. This mechanism monitors the validation loss during training and will end the training process if the validation loss fails to improve over a defined number of epochs.

3.3 Model Training and Evaluation

3.3.1 Training. The training of the gradient boosting classifier and feedforward neural network was conducted in a structured approach to ensure consistent development and analysis. The dataset was split into the training and testing sets. The training set was used to train the models and the test set was used to evaluate the model's performance.

The modes were then trained using the algorithms with specified hyper parameters (See 3.2). For the neural network, training involved forward propagation of inputs, back propagation of errors, and adjustments of weights. Specifically, the Stochastic Gradient Descent (SGD) optimizer was used, optimized after tuning to ensure optimal performance. The final configuration of SGD included a learning rate of 0.01, momentum of 0.9, and a weight decay of 0.01.

3.3.2 Evaluation. Evaluating the performance of the models is critical to understand their effectiveness. The metrics used to evaluate the performance of the models were: Accuracy, Precision, Recall (Sensitivity), F1-Score and Confusion Matrices.

Each metric offers insights into to the model performance and together provide a well understanding of how the models predict unseen data. The use of different metrics in essential to capture different aspects of model performance beyond simple accuracy, addressing both the model's strength and weaknesses in predicting fault.

3.4 Explainable Artificial Intelligence (XAI)

In efforts of making the models developed transparent and trustworthy, XAI techniques, specifically SHAP and LIME, were implemented. These techniques were applied after the models were trained to provide insights into the decision-making processes of both the gradient boosting classifier and feedforward neural network.

SHAP is utilized to quantify the contribution of each feature to the prediction. This method provides information about how much each feature is pushing the model output towards or away from a particular prediction.

LIME, on the other hand, offers local interpretability by approximating the underlaying model locally around a prediction. This algorithm perturbs the input data by tweaking the feature values and observes any changed in the predictions. Using these observations, LIME generates an interpretable model around the prediction,

	precision	recall	f1-score
0	0.88	0.85	0.86
1	0.93	0.94	0.94
Accuracy			0.91

Table 1. Results of Gradient Boosting Classifier model trained on unbalanced data (0 and 1 representing normal and faulty state respectively).

	precision	recall	f1-score
0	0.83	0.90	0.86
1	0.95	0.91	0.93
Accuracy			0.91

Table 2. Results of Gradient Boosting Classifier model trained on balanced data (0 and 1 representing normal and faulty state respectively).

	precision	recall	f1-score
0	0.97	0.90	0.93
1	0.95	0.99	97
Accuracy			0.96

Table 3. Results of Feedforward Neural Network model trained on unbalanced data (0 and 1 representing normal and faulty state respectively).

	precision	recall	f1-score
0	0.93	0.94	0.93
1	0.97	0.96	97
Accuracy			0.96

Table 4. Results of Feedforward Neural Network model trained on balanced data (0 and 1 representing normal and faulty state respectively).

which helps in understanding what changes to the input features lead to changes in the output.

4 RESULTS

4.1 Model Metrics

The performance of the gradient boosting algorithm on unbalanced and balanced datasets reveals several insights into the model's capabilities in fault detection. High accuracy is achieved when training on unbalanced data (See Table 1), specifically an accuracy of 91%. The precision and recall on the normal conditions were 88% and 85% respectively, and even higher for faulty data predictions with precision of 93% and recall 94%. These results indicate that the model is slightly better at identifying faulty conditions than normal conditions in terms of both precision and recall, resulting in an F1-score of 86% and 95% for classes 0 and 1, respectively.

When the data is balanced using the SMOTE techniques, the overall accuracy remains consistent at 91%, suggesting that balancing the classes does not have any negative nor positive impact on the model's overall accuracy (See Table 2). However, the internal dynamics of the model performance show some changes. The precision for normal conditions decreases to 83% but sees an increase in recall to 90%, indicating a trade-ff where the model can now capture higher proportion of actual normal conditions at the expense of slightly more false positives. An improvement for the faulty condition's precision is seen as it increases to 95% with a slight reduction in recall to 91%. This change suggests that while the model becomes more reliable in confirming fault conditions when it predicts them, it slightly reduces its sensitivity in detection all potential faults. The performance of the feedforward neural network also provides with some valuable insights into the model's behavior under different data conditions. Initially, on the unbalanced datasets, the model achieves an accuracy of 96% (See Table 3). It shows a precision of 97% on normal condition and 95% for the faulty condition, reflecting on the model's ability on distinguishing between normal and faulty states. The recall rates are 90% for the normal conditions and nearly perfect with 99% for the faulty conditions, meaning that while the model is slightly conservative in labelling conditions as normal, it excels at identifying most of the faulty conditions, corroborated by a high F1-score of 97% for normal condition predictions.

Upon applying the SMOTE technique to balance the dataset, the model's performance demonstrates some shift (See Table 4). For the normal class, precision drops to 93%, while the recall increases to 94%. This shift suggests that the model, while the model became less precise, it is now more inclusive when classifying conditions as normal, catching more true normal states but at the cost of increase false positives. Conversely, for the faulty class, precision rises to 97% indicating that a high number of faulty conditions were made right. However, the recall for faulty conditions decreases to 96%, pointing to a slight drop in the model's ability to detect all faulty conditions compares to the unbalanced scenario.

Particularly in settings with class imbalance, balancing the datasets typically leads to improved overall accuracy and model robustness by ensuring that the model does not favor the majority class. However, in the case of the feedforward network analyzed here, the expected increase was not observed here, instead, there was a slight decrease.

While balancing of classes though techniques like SMOTE ensures that the minority class is better represented, this can sometimes lead to overfitting. The model may adapt too well at the synthetic characteristics of the generated samples, which might not perfectly represent the true distribution of the minority class.

As observed, balancing improved recall at the expense of precision in certain classes. This trade-off can affect overall accuracy, which is critical in fault detection scenarios where false positives can be as detrimental as false negatives.

4.2 Explainable Artificial Intelligence (XAI)

To better understand these unexpected outcomes, XAI techniques, specifically SHAP and LIME, may be crucial.

By applying SHAP, we can identify how each feature contributes to the decisions behind the models. SHAP values can reveal how certain features may behave differently when the data is balanced, potentially leading to insights on whether the model's unexpected performance is due to certain features in the data.

LIME may be useful to explore local probabilities around individual predictions across both balanced and unbalanced datasets. It can help understand on a case-by-case basis why certain instances are misclassified after balancing proving insights into specific conditions or feature ranges where the model's performance weakens.

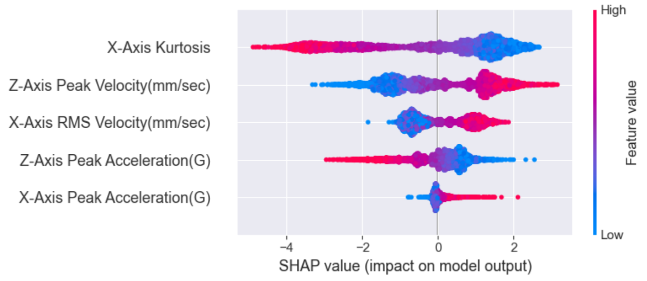

Fig. 3. SHAP values for the Gradient Boosting model trained on unbalanced data. Plot shows how much each feature has an impact on the output.

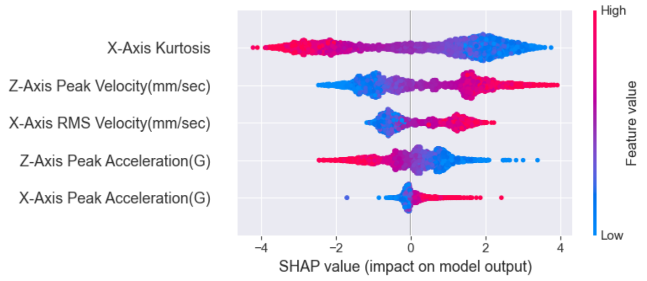

Fig. 4. SHAP values for the Gradient Boosting model trained on balanced data. Plot shows how much each feature has an impact on the output.

The SHAP value plots for the gradient boosting model applied to both unbalanced (See Figure 3) and balanced (See Figure 4) data provide visual evidence of how certain features impact model predictions. The SHAP analysis indicate that features "X-Axis Kurtosis" and "Z-Axis Peak Velocity" have a significant influence on the prediction of the models. This suggests a strong dependency on these features, meaning that small changes in these readings may lead to shifts in the model's predictions.

In our analysis using LIME on the gradient boosting model, particular attention was given to instances that were incorrectly classified. These instances generally demonstrated a distinct pattern where the "X-axis" feature influenced towards a faulty state, while most of the other features would contribute to pushing the prediction back towards a non-faulty state.

As seen in Figure 5 even though the true label was normal state, only "X-Axis Kurtosis" contributed the prediction towards faulty, making it a wrong classification. However, an interesting observation was that for wrong prediction of faulty (See Figure 6) "X-Axis Kurtosis" would again push towards the prediction being faulty, however, other features would this time push the prediction in being normal state.

The SHAP plot for unbalanced data (See Figure7) shows significant positive and negative impact from various features. Notably, features like "X-Axis Kurtosis" and "Z-Axis Peak Acceleration(G)" have

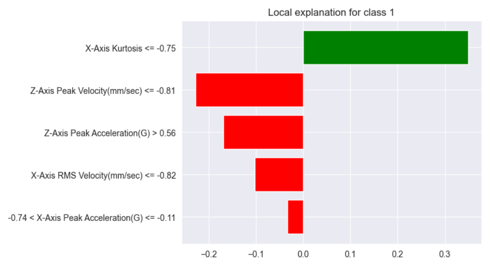

Fig. 5. LIME Gradient Boosting False Negative: Case when the actual state of the motor is normal, but the prediction is made as faulty state.

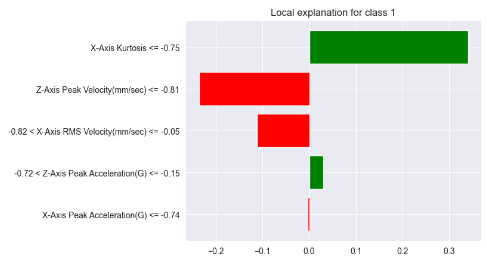

Fig. 6. LIME Gradient Boosting False Positive: Case when the actual state of the motor is faulty, but the prediction is made as normal state.

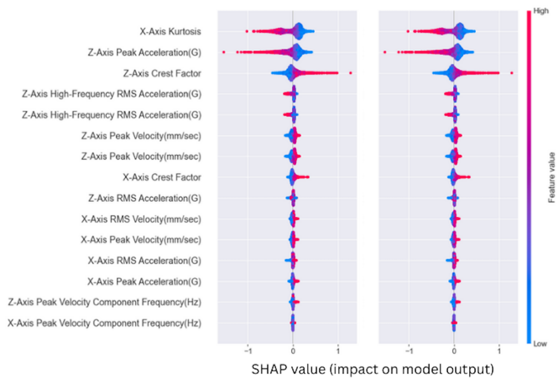

Fig. 7. SHAP values for the Feedforward Neural Network model trained on unbalance and balanced data, left and right plots respectively. Plot shows how much each feature has an impact on the output.

a considerable negative effect, suggesting these features strongly predict that there is no fault. On the other hand, "Z-Axis Crest Factor" and "X-Axis Crest Factor" show a positive influence, which might indicate their strong associations with fault conditions. Analyzing the LIME output in Figure 8, depicting a false positive scenario, where the model predicts the state as fault it is a normal condition. It is easily seen that "Z-Axis Crest Factor" is influencing these results by pushing the prediction towards the faulty prediction. After analyzing most of the false positive prediction this pattern was seen in predictions.

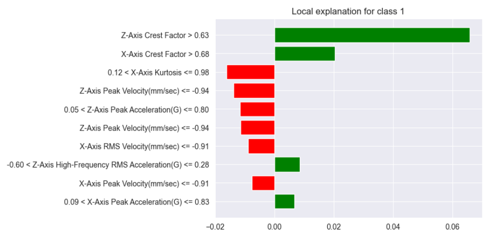

Fig. 8. LIME Feedforward Neural Network False Positive: Case when the actual state of the motor is faulty, but the prediction is made as normal state.

5 DISCUSSION

The comparative analysis of the gradient boosting and feedforward neural network models reveals that while both models perform well under unbalanced conditions, the introduction of SMOTE balancing does not enhance the performance, however, has varied effects on precision and recall. For all the cases it was evident that the precision for normal state would decrease, while for faulty state it would increase. Moreover, while an increase in recall for normal state was seen, it was also observed that there is decrease in recall for faulty states. These shifts can be attributed to how SMOTE changes the decision boundary by introducing synthetic characteristics or under fitting the data patterns, potentially effecting models' ability to generalize.

SHAP analysis provided a deeper understanding of feature influences, showing a strong dependency on features like 'X-Axis Kurtosis' and 'Z-Axis Peak Velocity". These insights were complemented by LIME, which illustrated the local decision-making process, highlighting instances where feature interactions led to misclassifications. These insights not only make the models created (See 3.2) transparent, but also guide targeted improvements though refined feature engineering and model tuning. By utilizing these interpretable tools, it might be possible to address specific missclassificatinos, adapt features for better accuracy, thereby strengthening decision-making processes.

Future studies should consider the inclusion of frequency-based features that capture the dynamic behaviors of induction motors under various operational conditions. By integrating these frequency-based features into the models, it is anticipated that the models may perform better on unseen data.

Also, an important field for future research is the development of time-series models to predict faults in induction motors before they occur. Current models developed focus on classification tasks, determining if a fault exists at a given time point. However, integrating time-series forecasting can extend this analysis by predicting when faults would likely occur.

6 CONCLUSION

This research, has investigated the effectiveness of machine learning and deep learning models, specifically gradient boosting and feedforward neural networks, in detecting faults in induction motors. The study leveraged XAI techniques, notably SHAP and LIME, to make created models (See 3.2) transparent, providing insights into how various features influence model predictions. The study effectively demonstrates that XAI techniques can be integrated into classification models for faults in induction motors (See 4.2). By applying SHAP, we were able to identify how each feature influenced models globally, revealing what were the features that influenced the predictions the most. LIME provided local interpretation that offered insight into specific instances, explaining why the models made certain decisions.

The integration of XAI techniques addresses the opaque nature of "black-box" models by providing clear explanations of why faults are predicted, which is crucial for end-users in operational settings.

ACKNOWLEDGMENTS

I would like to extend my gratitude to Hari Subramani Palanisamy for his valuable technical assistance and data provision, and to both Hari and Ahmed Faizan for their patient guidance and supervision throughout this research. Their expertise and support has been crucial to the finalization of this research.

REFERENCES

[1] Rui An, Zhaomin Tong, Yimei Ding, Bo Tan, Zihao Wu, Qiangqiang Xiong, and Yaolin Liu. 2022. Examining non-linear built environment effects on injurious traffic collisions: A gradient boosting decision tree analysis. *Journal of Transport and Health* 24 (3 2022). https://doi.org/10.1016/j.jth.2021.101296
[2] Anna Bogdanova, Akira Imakura, and Tetsuya Sakurai. 2023. DC-SHAP Method for Consistent Explainability in Privacy-Preserving Distributed Machine Learning. *Human-Centric Intelligent Systems* 3, 3 (7 2023), 197–210. https://doi.org/10.1007/s44230-023-00032-4
[3] Mohamed El and Hachemi Benbouzid. 2000. *A Review of Induction Motors Signature Analysis as a Medium for Faults Detection.* Technical Report 5.
[4] Czeslaw T. Kowalski and Teresa Orlowska-Kowalska. 2003. Neural networks application for induction motor faults diagnosis. In *Mathematics and Computers in Simulation*, Vol. 63. 435–448. https://doi.org/10.1016/S0378-4754(03)00087-9
[5] Prashant Kumar and Ananda Shankar Hati. 2021. Review on Machine Learning Algorithm Based Fault Detection in Induction Motors. *Archives of Computational Methods in Engineering* 28, 3 (5 2021), 1929–1940. https://doi.org/10.1007/s11831-020-09446-w
[6] Yuri Merizalde, Luis Hernández-Callejo, and Oscar Duque-Perez. 2017. State of the art and trends in the monitoring, detection and diagnosis of failures in electric induction motors. https://doi.org/10.3390/en10071056
[7] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Vol. 13-17-August-2016. Association for Computing Machinery, 1135–1144. https://doi.org/10.1145/2939672.2939778
[8] Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J. Anders, and Klaus Robert Müller. 2021. Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications. *Proc. IEEE* 109, 3 (3 2021), 247–278. https://doi.org/10.1109/JPROC.2021.3060483
[9] Rudra Tiwari. 2023. Explainable AI (XAI) and its Applications in Building Trust and Understanding in AI Decision Making. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT* 07, 01 (1 2023). https://doi.org/10.55041/ijsrem17592
[10] Carlos Verucchi, José Bossio, Guillermo Bossio, and Gerardo Acosta. 2016. Misalignment detection in induction motors with flexible coupling by means of estimated torque analysis and MCSA. *Mechanical Systems and Signal Processing* 80 (12 2016), 570–581. https://doi.org/10.1016/j.ymssp.2016.04.035
[11] Piotr Iwo Wójcik and Marcin Kurdziel. 2019. Training neural networks on high-dimensional data using random projection. *Pattern Analysis and Applications* 22, 3 (8 2019), 1221–1231. https://doi.org/10.1007/s10044-018-0697-0

A USE OF AI TOOLS

During the process of this research AI was used to help with code completion and support on grammar fixes for the paper. After using these tools, everything was reviewed and edited. The author takes full responsibility for everything.