DiffLIME: Enhancing Explainability with a Diffusion-Based LIME Algorithm for Fault Diagnosis

David Solis-Martin[1], Juan Galan-Paez[1], Joaquin Borrego-Diaz[1]

[1,2] *Computer Science and Artificial Intelligence Department, Seville University, Seville, Spain*
dsolis,juangalan,jborrego@us.es

ABSTRACT

The aim of predictive maintenance within the field of Prognostics and Health Management (PHM) is to identify and anticipate potential issues in equipment before they become serious. Deep learning models, such as deep convolutional neural networks (CNNs), long short-term memory (LSTM) networks, and transformers, have been widely adopted for this task, achieving significant success. However, these models are often considered "black boxes" due to their opaque decision-making processes, making it challenging to explain their outputs to stakeholders, such as industrial equipment experts. The complexity and large number of parameters in these models further complicate the understanding of their predictions.

This paper presents a novel explainable AI algorithm that extends the well-known Local Interpretable Model-agnostic Explanations (LIME). Our approach utilizes a conditioned probabilistic diffusion model to generate altered samples in the neighborhood of the source sample. We validate our method using various rotating machinery diagnosis datasets. Additionally, we compare our method against LIME, employing a set of metrics to quantify desirable properties of any explainable AI approach. The results highlight that DiffLIME consistently outperforms LIME in terms of coherence and stability while maintaining comparable performance in the selectivity metric. Moreover, the ability of DiffLIME to incorporate domain-specific meta-attributes, such as frequency components and signal envelopes, significantly enhances its explainability in the context of fault diagnosis. This approach provides more precise and meaningful insights into the predictions made by the model.

1. INTRODUCTION

Early artificial intelligence models, such as simple decision trees, were transparent and easy to interpret, but their capabilities were limited. However, in recent years, there has been a significant surge in the performance of predictive models used for tasks such as classification and regression. This improvement in performance has often come at the cost of increased complexity, reducing the interpretability of these models. These complex models, often referred to as "black-box" models (Rudin & Radin, 2019), are difficult to understand and explain due to their opaque decision-making processes.

This lack of transparency becomes particularly problematic in high-stakes fields where model predictions can have significant consequences, such as medicine, law, criminal profiling, autonomous driving, and defense (Goodman & Flaxman, 2017). Furthermore, black-box models are more difficult to debug, making it challenging to identify the root causes of errors or biases. In contrast, interpretable models facilitate diagnosing issues and implementing corrective measures (Gilpin et al., 2018).

1.1. Explainable Artificial Intelligence

Explainability and interpretability are related but distinct concepts in AI and machine learning. While no universally standardized definition exists, a commonly accepted distinction is as follows. Interpretability refers to the extent to which a human can understand how a model generates its predictions. An interpretable model is inherently understandable without requiring additional techniques. For example, decision trees and linear regression models are considered interpretable because their decision-making process is transparent. Explainability refers to the ability to describe or justify how a model makes decisions, often using additional techniques. Explainability is particularly important for complex models like deep learning, where the decision process is not inherently interpretable.

Explainable Artificial Intelligence (XAI) aims to address the challenges associated with the opacity of black-box machine learning (ML) models by developing methods that either produce explanations for these models or design inherently interpretable models, particularly in the context of post hoc ex-

https://doi.org/10.36001/IJPHM.2025.v16i3.4166

plainability (Arrieta et al., 2020). Post hoc explainability approaches can be categorized into model-agnostic and model-specific techniques. Model-agnostic techniques are versatile and can be applied to a wide range of ML models. Examples include LIME (Local Interpretable Model-Agnostic Explanations) (Ribeiro, Singh, & Guestrin, 2016) and SHAP (SHapley Additive ExPlanations) (Scott, Su-In, et al., 2017). Conversely, model-specific techniques are tailored to specific types of ML models. Notable examples include Grad-CAM (Gradient-weighted Class Activation Mapping) (Selvaraju et al., 2017), saliency maps (Simonyan, Vedaldi, & Zisserman, 2013), and layer-wise relevance propagation (LRP) (Bach et al., 2015), which are predominantly used for interpreting deep learning (DL) models.

XAI techniques are widely applied across various domains, with a particular focus on tasks involving tabular and image data. These domains have seen significant advancements in explainability methodologies, leveraging the structured nature of tabular data and the visual explainability of image-based tasks. However, signal processing tasks have historically received comparatively less attention in the XAI literature, resulting in a gap in the development of explainability methods for these applications. Similarly, the application of XAI techniques to time-series models remains largely underexplored (Schlegel, Arnout, El-Assady, Oelke, & Keim, 2019).

Unlike tabular or image data, time-series data often exhibit a non-intuitive nature characterized by complex temporal dependencies, seasonality, and variability. These characteristics make time-series data harder to interpret, both for humans and algorithms, complicating the development of effective explainability techniques (Siddiqui, Mercier, Munir, Dengel, & Ahmed, 2019). Addressing this challenge requires tailored approaches that consider the temporal and sequential structure of time-series data, potentially opening new avenues for advancing XAI in this domain.

Attribution methods, widely used in computer vision, generate heatmaps to highlight the most relevant regions of an input, helping to understand model predictions. These techniques have also been adapted for time-series data, aiming to identify the most influential time steps that contribute to the decision of a model.

Schlegel *et al.* (Schlegel et al., 2019) explored the use of several attribution interpretability techniques, including LIME, SHAP, and saliency maps, to explain deep learning models for time-series classification. Their work demonstrated that these methods, originally designed for images, could be effectively applied to sequential data by treating temporal steps as features and generating importance scores accordingly.

Class Activation Mapping (CAM), a popular technique in image interpretability, has been adapted for time series. Wang *et al.* (Wang, Yan, & Oates, 2017) and Selvaraju *et al.* (Selvaraju et al., 2020) proposed to use CAM-based approaches for time-series classification. These methods rely on the activations of convolutional layers to identify the most discriminative time steps, effectively highlighting key regions in the input sequence that influence the prediction of the model.

Another relevant contribution is TSViz (Siddiqui et al., 2019), which leverages saliency maps for analyzing time series. TSViz provides a visualization framework that applies gradient-based saliency techniques, such as Guided Backpropagation and Integrated Gradients, to reveal which temporal segments contribute most to a classification outcome.

Schlegel *et al.* (Schlegel, Vo, Keim, & Seebacher, 2021) proposed TS-MULE, an extension of LIME for time series forecasting models that utilizes local surrogates with a time-aware neighborhood generation strategy. It segments the time series into subcomponents and evaluates their contributions to predictions. Differently, Meng *et al.* (Meng, Wagner, & Triguero, 2023) identified key time series segments by optimizing perturbations that maximize the change in prediction probability. Their method determines the most influential regions of the signal based on their contribution to classification decisions. This approach is focused on network architecture, as it relies on gradients for optimization. It is noteworthy that these methods are subsequence-based, as their explanations refer to specific subparts of the time series.

Despite their effectiveness, these attribution-based methods have limitations. Many gradient-based approaches, including saliency maps, suffer from noise and instability, producing inconsistent explanations across similar inputs. Additionally, LIME and SHAP rely on perturbation-based sampling, which may not capture temporal dependencies accurately. These challenges highlight the need for more robust interpretability methods tailored specifically to time-series data which incorporate domain-specific constraints to improve explanation.

1.2. XAI in Predictive Maintenance

XAI methods have seen limited application in the field of PHM (Vollert, Atzmueller, & Theissler, 2021). Despite the growing recognition of the importance of explainability in industrial settings, few studies have focused on developing new XAI techniques or adapting existing ones to address the unique challenges of PHM.

Notable progress has been made in specific applications. For instance, (Decker, Lebacher, & Tresp, 2023) proposed a method that transforms input data from the time domain to the frequency domain, allowing feature attribution techniques to be applied directly in the frequency domain. This approach avoids retraining or modifying the weights of pre-trained models in the time domain, simplifying the explainability process.

Similarly, (Zereen, Das, & Uddin, 2024) and (Santos, Guedes,

& Sanchez-Gendriz, 2024) demonstrated the use of LIME and SHAP techniques to improve machine fault diagnostics. These methods enable targeted feature selection from time and frequency-domain attributes, achieving accuracy comparable to traditional approaches while significantly reducing feature complexity.

The FaultD-XAI framework (Brito, Susto, Brito, & Duarte, 2023) provides an explainable and scalable solution for fault diagnosis in rotating machinery. By leveraging transfer learning with synthetic vibration signals and applying Grad-CAM to 1D CNNs, it enhances both diagnostic performance and user confidence by offering post hoc explanations of model predictions.

While these efforts represent significant advancements, the integration of XAI into PHM is still in its early stages. Further research is required to address the complexity of industrial environments and develop more robust solutions tailored to the specific challenges of this domain.

1.3. Aim and Structure of the Paper

The primary objective of this work is to propose a novel method that extends the well-known LIME algorithm. This extension incorporates additional features specifically designed to enhance its applicability to time-series data, with a particular focus on fault diagnosis signals. Traditional LIME, while effective in many domains, encounters challenges when applied to time-series data due to its inherent temporal dependencies and non-stationary characteristics. Our proposed method addresses these limitations by incorporating domain-specific adaptations, such as signal envelopes, frequency components, and other meta-features, to improve the explainability of predictions in fault diagnosis tasks.

DiffLIME introduces a novel approach to enhancing the explainability of black-box models by addressing key limitations of existing methods like LIME, SHAP, and CAM-based techniques. Unlike traditional perturbation-based methods such as LIME, which randomly alters input features and may disrupt temporal dependencies, DiffLIME leverages a diffusion probabilistic model (DPM) to generate more realistic perturbations, preserving the underlying structure of time-series data. This leads to more coherent and stable explanations. Additionally, the incorporation of meta-attributes enhances the robustness of explanations by providing additional context for model explainability in concrete for its application in fault diagnosis. Compared to CAM-based methods, which are limited to convolutional architectures and struggle with sequential dependencies in time series, DiffLIME remains model-agnostic, making it applicable to a broader range of architectures and other machine learning models. Moreover, our experiments demonstrate that DiffLIME significantly improves explanation stability while maintaining competitive selectivity and coherence, offering a more reliable alternative.

The main contributions of this work are as follows:

- Diffusion-based Perturbation Mechanism: Unlike LIME, which relies on arbitrary perturbations, DiffLIME employs a diffusion probabilistic model (DPM) to generate more structured perturbations, preserving the temporal consistency of time-series data.
- Enhanced Stability and Coherence: Our method significantly improves explanation stability across similar instances, reducing variability in feature attributions while maintaining high coherence.
- Meta-Attribute Integration: DiffLIME introduces meta-attributes to enhance interpretability, providing additional contextual insights into model predictions.
- Comprehensive Empirical Validation: We compare DiffLIME against LIME and TS-MULE on benchmark datasets, demonstrating its competitive performance in terms of coherence, stability, and selectivity.
- Application: We applied the proposed method to fault diagnosis, an area where there is a significant gap in the application of XAI techniques and where more attention is needed.

This paper is structured as follows. Section 2 introduces the proposed method, detailing the algorithmic modifications and additional features designed to enhance its performance for time-series data. Section 3 presents experimental results on benchmark fault diagnosis datasets, demonstrating the efficacy of the proposed approach compared to traditional methods. Section 4 discusses the implications of the findings, the limitations of the current work, and potential avenues for future research. Finally, Section 5 summarizes the contributions and highlights the importance of explainability in fault diagnosis applications.

2. METHODS

In this section, we outline the methodology used to explain the predictions of a predictive model through a novel diffusion-based adaptation of LIME, referred to as DiffLIME. First, we introduce the foundational concepts of the LIME algorithm. Next, we describe the training process of the DPMs and how they are used to generate the neighborhood of source signals. Finally, we detail the training procedure for the explainable surrogate model employed in DiffLIME.

Figure 1 provides a summary of the complete methodology, which is detailed in the following sections. First, features such as envelopes, frequencies, slopes, and noise ratios are extracted from the original data to capture its key characteristics. Using these extracted features, a synthetic dataset is generated and used to train a DPM. Once the black-box model is

trained, DiffLIME leverages the DPM to generate explanations for each of its predictions, ensuring more coherent and stable interpretability.

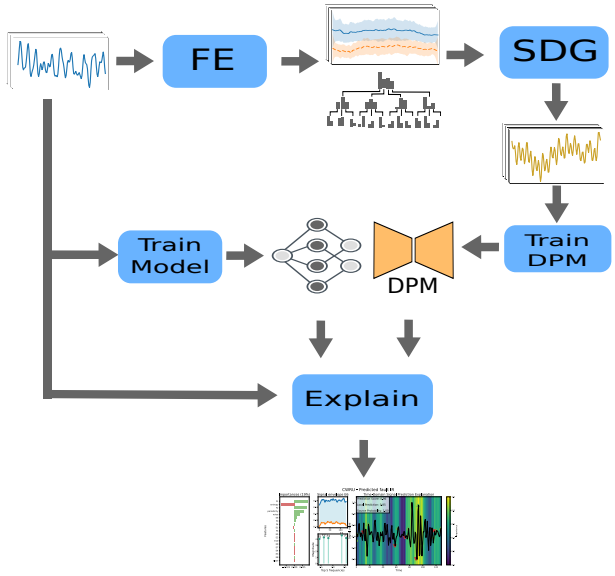

Figure 1. Summary graph of the complete methodology that define DiffLIME. FE: Feature Extraction, SDG: Synthethic Data Generation, DPM: Denoising Probabilistic Model

2.1. Interpretable Model-Agnostic Explanations (LIME)

LIME is a prominent technique within XAI, designed to provide explainability for complex machine learning models. LIME approximates the behavior of a black-box model locally around a specific instance by constructing a surrogate model that is inherently interpretable. This surrogate model is trained on a perturbed neighborhood of the original data instance, enabling the extraction of localized feature importance.

Formally, given a machine learning model $f : \mathcal{X} \to \mathbf{R}$, where $\mathcal{X}$ represents the input space, LIME aims to explain the prediction $f(x)$ for a specific instance $x \in \mathcal{X}$. To achieve this, LIME constructs a local neighborhood $\mathcal{N}(x)$ by generating perturbed versions of x. The perturbed instances $\{\tilde{x}^{(j)}\}$ are sampled from a distribution $\mathcal{D}(x)$, which preserves the proximity of x while introducing variations. Typically, the neighborhood distribution $\mathcal{D}(x)$ is defined as a set of perturbed samples derived from the original input $x = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n]$ by applying Gaussian noise to each feature independently:

$$\mathcal{D}(x) = \{\tilde{x} \mid \tilde{x}_i = \mathbf{s}_i + \epsilon_i,\ \epsilon_i \sim \mathcal{N}(0, \sigma^2),\ i = 1, 2, \ldots, n\}. \tag{1}$$

Here, $\tilde{x} = [\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n]$ is a perturbed sample, where each feature $\tilde{x}_i$ is drawn from a normal distribution centered at $\mathbf{s}_i$ with variance σ^2. For each perturbed instance $\tilde{x}^{(j)}$, the output of the model $f(\tilde{x}^{(j)})$ is computed.

To approximate the local behavior of f, LIME trains an interpretable surrogate model $g : \mathcal{X} \to \mathbf{R}$, such as a linear regression or decision tree, using the dataset $\{(\tilde{x}^{(j)}, f(\tilde{x}^{(j)}))\}$. The surrogate model is optimized to minimize the following objective function:

$$\mathcal{L}(f, g, \pi_x) = \sum_{\tilde{x}^{(j)} \in \mathcal{N}(x)} \pi_x(\tilde{x}^{(j)}) \cdot \left(f(\tilde{x}^{(j)}) - g(\tilde{x}^{(j)})\right)^2, \tag{2}$$

where $\pi_x(\tilde{x}^{(j)})$ is a proximity kernel that assigns higher weights to instances closer to x in the input space. A common choice for $\pi_x(\tilde{x}^{(j)})$ is an exponential kernel:

$$\pi_x(\tilde{x}^{(j)}) = \exp\left(-\frac{dist(x, \tilde{x}^{(j)})}{\sigma}\right) \tag{3}$$

where dist$(\cdot, \cdot)$ measures the distance between x and $\tilde{x}^{(j)}$, and σ controls the spread of the neighborhood. The interpretable surrogate model g provides an explanation for $f(x)$ by revealing the contributions of input features to the prediction within the local neighborhood.

The flexibility of LIME lies in its model-agnostic nature, as it does not require access to the internal workings of f and is applicable to any black-box model. However, the method is inherently local and focuses on the behavior of f near x. While this provides valuable insights, it does not capture the global decision boundary of f. Additionally, the quality of the explanation depends on the representativeness of the neighborhood $\mathcal{N}(x)$ and the simplicity of the surrogate model g.

In the following section, we will use the symbol s instead of x to denote a time-series signal, aligning the notation with the context of signal processing.

2.2. Local Neighborhood in DiffLIME

In the DiffLIME algorithm, a DPM is used to generate the local neighborhood of an input signal $\mathbf{s} \in \mathbf{R}^n$. The DPM, trained on a synthetic dataset, produces diverse and representative samples that capture the characteristics of the underlying data distribution. Given an input signal $\mathbf{s}$, neighborhood signals are generated by conditioning the DPM on meta-features extracted from $\mathbf{s}$, ensuring that the local neighborhood is well-sampled while preserving key signal characteristics.

The meta-features, $\mathbf{m} \in \mathbf{R}^k$, are computed using:

$$\mathbf{m} = h(\mathbf{s}), \tag{4}$$

where $h(\cdot)$ represents the meta-feature extraction function.

Synthetic samples $\mathbf{s}'$ are generated by perturbing $\mathbf{s}$, $\mathbf{m}$ and

sampling from the DPM:

$$\mathbf{s}' \sim DPM(\tilde{\mathbf{s}}|\tilde{\mathbf{m}}), \tag{5}$$

where $\tilde{\mathbf{m}}$ and $\tilde{\mathbf{s}}$ are generate using the process defined in Equation 1.

Diffusion Probabilistic Models (DPMs). DPMs (Ho, Jain, & Abbeel, 2020) are generative frameworks that systematically remove noise from data through an iterative denoising process. These models are trained to estimate the noise introduced during the forward diffusion process, enabling the reconstruction of the original data by reversing the diffusion dynamics. The forward diffusion process incrementally corrupts signal $\mathbf{s}_0$ with Gaussian noise across T time steps. The posterior distribution of this process, denoted as $q(\mathbf{s}_{1:T}|\mathbf{s}_0)$, is formally expressed as:

$$q(\mathbf{s}_{1:T}|\mathbf{s}_0) = q(\mathbf{s}_0)\prod_{t=1}^{T}\mathcal{N}(\mathbf{s}_t|\sqrt{1-\beta_t}\mathbf{s}_{t-1}, \beta_t I), \tag{6}$$

where I is the identity matrix, $\beta_t \in (0,1)$ represents a variance schedule that increases with t, and $\mathcal{N}(\mathbf{s}|\mu, \sigma^2 I)$ denotes a Gaussian distribution with mean μ and variance $\sigma^2 I$. As $T \to \infty$, the forward process transforms the data distribution into an isotropic Gaussian distribution.

To efficiently sample intermediate states without iterating through all T steps, the forward process can be reformulated. Defining $\alpha_t = 1 - \beta_t$ and $\overline{\alpha}_t = \prod_{i=1}^{t}\alpha_i$, the marginal distribution at step t is given by:

$$q(\mathbf{s}_t|\mathbf{s}_0) = \mathcal{N}(\mathbf{s}_t|\sqrt{\overline{\alpha}_t}\mathbf{s}_0, (1-\overline{\alpha}_t)I). \tag{7}$$

The generation of new samples requires the reverse diffusion process, characterized by the posterior distribution $q(\mathbf{s}_{t-1}|\mathbf{s}_t)$. However, directly computing this distribution is intractable due to its dependence on the true data distribution. To address this challenge, a parameterized model $p_\phi(\mathbf{s}_{t-1}|\mathbf{s}_t)$ is introduced to approximate $q(\mathbf{s}_{t-1}|\mathbf{s}_t)$. Assuming a Gaussian form, this model is expressed as:

$$p_\phi(\mathbf{s}_{t-1}|\mathbf{s}_t) = \mathcal{N}(\mathbf{s}t-1|\mu_\phi(\mathbf{s}_t, t), \sigma_\phi^2(\mathbf{s}_t, t)I), \tag{8}$$

where μ_ϕ and σ_ϕ^2 represent the predicted mean and variance, respectively, and ϕ denotes the trainable parameters of the model.

The parameters ϕ are optimized by minimizing the negative log-likelihood of the training data. An effective alternative to the Evidence Lower Bound (ELBO) was proposed in (Ho et al., 2020), leveraging a simpler objective that directly optimizes the noise estimation task. The loss function is defined as:

$$\underset{\phi}{argmin}\ \frac{1}{M}\sum_{i=1}^{M}\mathcal{L}\left(\epsilon - \epsilon_\phi\left(\sqrt{\overline{\alpha}_t}\mathbf{s}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, t\right)\right), \tag{9}$$

where $\epsilon \sim \mathcal{N}(0, I)$ represents the noise sampled from a Gaussian distribution, and $\mathcal{L}$ is a loss function measuring the discrepancy between the true noise ϵ and the predicted noise ϵ_ϕ.

The reverse diffusion step is thus formulated as:

$$\mu_\phi(\mathbf{s}_t, t) = \frac{\sqrt{\overline{\alpha}_t}\mathbf{s}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon - \sqrt{1-\overline{\alpha}_t}\epsilon_\phi\left(\sqrt{\overline{\alpha}_t}\mathbf{s}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, t\right)}{\sqrt{\overline{\alpha}_t}} \tag{10}$$

$$p_\phi(\mathbf{s}_{t-1}|\mathbf{s}) = \mathcal{N}\left(\mathbf{s}_{t-1}\middle|\mu_\phi(\mathbf{s}_t, t), I\right). \tag{11}$$

This framework enables the generation of high-quality signals by iteratively denoising the data.

Synthetic Data Generation Process. DPMs typically require large amounts of training data to learn complex distributions effectively. However, in many real-world scenarios, especially in domains like fault diagnosis or time-series analysis, the available dataset is often limited or imbalanced. The synthetic data generation process overcomes this limitation by producing diverse and representative samples that reflect the characteristics of the underlying data distribution:

$$\mathcal{D}_{synthetic} = \bigcup_{i=1}^{N}\{\mathbf{s}^{(j)}, \tilde{\mathbf{s}}^{(j)}, \mathbf{m}^{(j)}, \mathbf{e}^{(j)}\},$$

where $\mathbf{s}^{(j)}$ is a clean synthetic signal, $\tilde{\mathbf{s}}^{(j)}$ is the corresponding noisy signal, and $\mathbf{m}^{(j)}$ contains metadata describing the characteristics of the signal $\mathbf{s}^{(j)}$, and $\mathbf{e}^{(j)}$ represent the upper and lower envelopes of the signal $\mathbf{s}^{(j)}$.

The process generates synthetic signals by varying critical domain-specific parameters, such as frequencies, slope, noise, and envelope structures. These variations ensure that the DPM learns to model not only the observed data but also potential unseen variations, improving generalization.

Algorithm 1 shows the complete process of synthetic data generation. The goal of the $GenerateDistributions$ function is to analyze a dataset $\mathbf{X}$ of time-series signals and compute a hierarchical representation of their statistical properties across different frequency bands. This representation captures information such as frequency distributions, noise ratios, signal slopes, and probabilities of cluster assignments for signal envelopes. It provides a structured probabilistic summary of the dataset, useful to generate synthetic time-series while respecting the original statistical characteristics,

Algorithm 1: Synthetic Data Generation Algorithm

Inputs: Dataset $\mathbf{X}$, total samples N to generate

$\mathbf{D} \leftarrow GenerateDistributions(\mathbf{X})$;
$\mathbf{E} \leftarrow ExtractEnvelopes(Detrend(\mathbf{X}))$;
$\mathbf{C} \leftarrow KMeans(\mathbf{E}, N_{cluster})$;
$N_s \leftarrow$number of signals to generate for each feature distribution in $\mathbf{D}$;

Initialize arrays $\mathcal{D}_{synthetic} \in \mathbf{R}^{2N \times T}$, ;

foreach $d \in \mathbf{D}$ **do**
  for $n \in N_s[d]$ **do**
    Sample frequencies $\mathbf{f}$, slope m, noise η, and envelope index i_e from d;
    Generate envelope $(\mathbf{e}_u, \mathbf{e}_l)$ sampled from $\mathbf{C}[i_e]$ and its deviations;
    Generate clean signal;
    $\mathbf{s} \leftarrow \sum_{f \in \mathbf{f}} \sin(\mathbf{t} \cdot f)$;
    $\mathbf{s} \leftarrow AdjustToEnvelopes(\mathbf{s}, \mathbf{e}_u, \mathbf{e}_l)$;
    $\mathbf{s} \leftarrow AddSlope(\mathbf{s}, m)$;
    $\mathcal{D}_{synthetic}[i], \mathbf{M}[i] \leftarrow \mathbf{s}, \{\mathbf{f}, m, 0, i_e\}$;
    $\mathcal{D}_{synthetic}[i+1], \mathbf{M}[i+1] \leftarrow AddNoise(\mathbf{s}, \eta), \{\mathbf{f}, m, \eta, i_e\}$;
    $i \leftarrow i + 2$;
  end
end
return $\mathcal{D}_{synthetic}$;

including frequency dynamics, noise levels, and envelope distributions.

For each time-series signal $\mathbf{s}^{(j)} \in \mathbf{X}$, the function extracts the top n dominant frequencies using spectral decomposition or Fourier transform. This produces a matrix $\mathbf{F} \in \mathbb{R}^{N \times k}$, where N is the number of signals and k is the number of dominant frequencies for each signal:

$$\mathbf{F}_{jk} = f_k(\mathbf{s}^{(j)}), \quad i = 1, \ldots, N, \quad k = 1, \ldots, K,$$

where $f_k(\mathbf{s}^{(j)})$ denotes the k-th dominant frequency of the signal $\mathbf{s}^{(j)}$.

The most prominent frequencies $\mathbf{F}_{:,1}$ are used to define an initial coarse segmentation of the frequency range. The range $[\min(\mathbf{F}_{:,1}), \max(\mathbf{F}_{:,1})]$ is divided into C equal chunks:

$$\Delta f = \frac{\max(\mathbf{F}_{:,1}) - \min(\mathbf{F}_{:,1})}{c},$$
$$R_c = \left[\min(\mathbf{F}_{:,1}) + (c-1)\Delta f, \min(\mathbf{F}_{:,1}) + c\Delta f\right],$$
$$c = 1, \ldots, C$$

For each frequency range R_c, the function identifies signals whose most prominent frequencies fall within R_c. The process is recursively applied to their remaining $k-1$ frequencies. At each step, the range of frequencies is further partitioned, creating a hierarchical structure of increasingly finer resolutions.

At each level of the hierarchy, the mean and standard deviation of the frequencies are computed. Additionally, at the final level of the hierarchy, the mean and standard deviation of the noise and slope are included, along with the most likely envelope of the signals in the group.

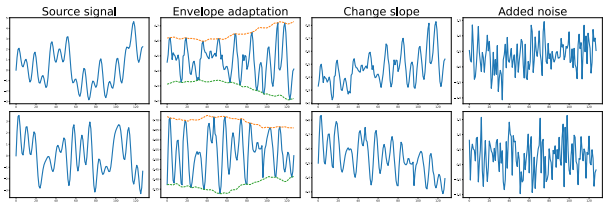

Figure 2. Two examples of synthethic signal generation process

To ensure that the generation process accurately captures the full distribution of the data, meta-features are extracted from all categories present in each dataset, including both healthy and faulty conditions.

The signal is detrended before extracting the envelopes, ensuring they are centered around the zero y-axis. We hypothesize that this facilitates the clustering of signal envelopes and improves their interpretability during the visualization of the explanations. Additionally, the original trend is treated as a meta-feature, which may be useful depending on the dataset and task. For instance, in Remaining Useful Life (RUL) prediction, the trend can provide valuable insights into the degradation process, making it a crucial feature to consider.

To ensure that the synthetic dataset contains signals representing the same feature distribution, the number of samples to generate for each set, N_s, is determined based on the ratio of the number of samples exhibiting those features in the original dataset. This approach preserves the proportional representation of features, maintaining consistency between the synthetic and real data distributions.

From the predefined ranges of meta-features, a random value is selected for each specific feature, including frequencies, slope, envelope, and noise level. Using the frecuency values, an initial synthetic signal is generated as a combination of sinusoidal components. This intermediate signal is then adjusted to match the selected envelope, and finally, the specified noise level is added. Figure 2 illustrates two examples of this synthetic signal generation process.

Residual U-Net Model. The DPM model is implemented as a Residual U-Net, a neural network architecture specifically designed for denoising time-series data (Solis-Martin, Galan-Paez, & Borrego-Diaz, 2023). This model utilizes skip connections, residual blocks, and metadata to improve its performance on complex temporal signals. The input to the model consists of a noisy time-series signal, concatenated with both upper and lower envelopes, meta-attributes, and the denoising step index.

It is composed of down-sampling blocks, residual blocks, and up-sampling blocks. The down-sampling blocks progressively

reduce the temporal resolution while increasing the feature dimensions, enabling the model to capture high-level patterns. On the other hand, the up-sampling blocks reconstruct the time-series signal, incorporating information from earlier stages through skip connections. Figure 3 shows the schematic of the down-up blocks.

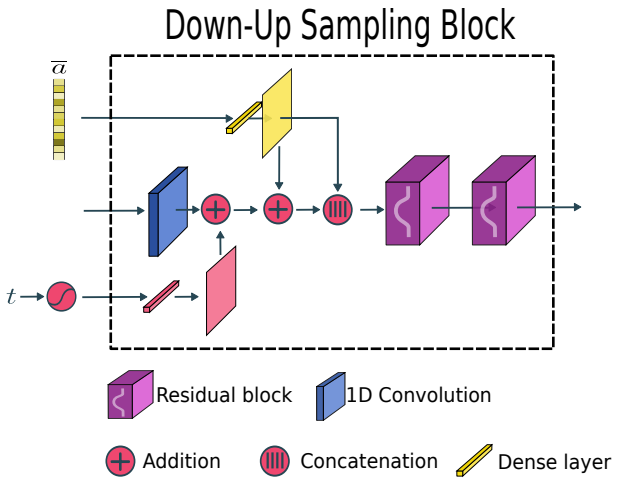

Figure 3. Diagram of the down-sampling and up-sampling blocks.

Meta-attributes and timestep information are integrated into the model as conditional inputs. For timestep embedding, the model employs an embedding mechanism where the timestep index is projected into a high-dimensional space. This embedding is further processed through fully connected layers (see Figure 3) with non-linear activations, such as the Swish activation function (Ramachandran, Zoph, & Le, 2017). Meta-attributes are incorporated as a conditioning mechanism through layers that integrate these features into the temporal representation. Figure 4 shows the full architecture of the DPM network.

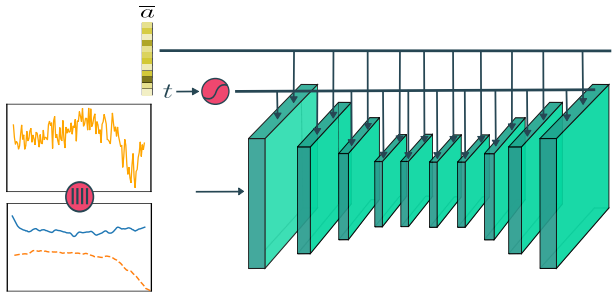

Figure 4. A schematic representation of the DPM neural network.

The model architecture is structured with six down-sampling levels and six up-sampling levels, each with a depth of three. This depth refers to the number of down-sampling and up-sampling blocks present at each level. The channel sizes vary across levels, with the number of convolutional kernels defined by the following sequence: 32, 64, 64, 32, 32, 32. These values specify the number of filters used in the convolutional layers at each respective level.

Additionally, the DPM network is trained to perform over 20 denoising timesteps, effectively removing noise from the input signal progressively through the reverse diffusion process.

The Adam optimizer is used with a learning rate of 1×10^{-5}. The loss function is set to the mean absolute error (MAE), which quantifies the difference between the predicted and actual values. The model is trained for a total of 2000 epochs, with a batch size of 32. Each epoch processes 100 steps.

2.2.1. Surrogate Model

Given a signal $\mathbf{s}$ and a predictive model f, DiffLIME generates M local neighborhood samples using the trained DPM. From the signal $\mathbf{s}$, the vector of meta-attributes $\mathbf{m}$, and the envelope index e are obtained. A new dataset is then created to feed the surrogate model:

$$k = \text{argmax} f(\mathbf{s})$$

$$\mathbf{s}'^{(j)} = DPM(\tilde{\mathbf{s}}^{(j)}, \tilde{\mathbf{m}}^{(j)}, C_{\tilde{e}^{(j)}})$$

$$\mathcal{D}_{explanaible} = \{\mathbf{s}, \mathbf{m}, \mathbf{e}, f_k(\mathbf{s})\} \bigcup_{j=1}^{M} \{\mathbf{s}'^{(j)}, \mathbf{m}'^{(j)}, \mathbf{e}'^{(j)}, f_k(\mathbf{s}'^{(j)})\},$$

where $\tilde{\mathbf{s}}^{(j)}$ and $\tilde{\mathbf{m}}^{(j)}$ represent the perturbed signals and perturbed meta-attribute vectors, respectively, using the process defined in Equation 1, and $C_{\tilde{e}^{(j)}}$ is the envelope associated with the perturbed signal $\tilde{\mathbf{s}}^{(j)}$. Once the signal has been reconstructed by the DPM, the meta-attributes and the envelopes are computed over $\mathbf{s}'^{(j)}$, yielding $\mathbf{m}'^{(j)}$ and $e'^{(j)}$.

After all samples are generated, the meta-attributes are scaled using a standard scaler to ensure proper normalization. Finally, the envelope indices $e'^{(j)}$ are encoded using one-hot encoding, resulting in the vectors $\mathbf{e}'^{(j)}$, which are included in the explainable dataset along with the predicted probability for the category selected by f over the source sample $\mathbf{s}$.

2.3. Explanations

The surrogate model $f_{\text{surrogate}}$ is trained on the dataset $\mathcal{D}_{\text{explainable}}$ to approximate the predictions of the black-box model $f_k(\mathbf{s}'^{(j)})$. In this work, we use a Ridge regression model as the surrogate, which is categorized as an explainable model. Since Ridge regression is a linear model, it provides an easily interpretable explanation in the form of a weighted sum of input features. This approximation remains valid within the neighborhood of the original instance, ensuring that the explanation

reflects the local behavior of f_k rather than a global summary.

Once the Ridge regression model is trained as the surrogate, its learned coefficients play a central role in interpreting the predictions of the black-box model f_k. Each coefficient represents the estimated contribution of a particular feature in the perturbed dataset $\mathcal{D}_{\text{explainable}}$ to the prediction of the model.

A higher absolute value of a coefficient indicates a stronger influence of the corresponding feature on the prediction. Positive coefficients suggest that increasing the feature value leads to a higher predicted output, while negative coefficients indicate an inverse relationship. Furthermore, the coefficients serve as a mechanism to rank feature importance, helping to identify which aspects of the input signal most significantly affect the decision of the model.

The coefficients aligned with the raw signal data points provide insights into time-domain feature importance. Additionally, the coefficients aligned with the meta-attributes extend the explanations to other domains, such as frequency components, global characteristics like envelope shapes, or statistical properties such as slopes and noise ratios.

3. EXPERIMENTS

This section details the experiments conducted to validate the proposed DiffLIME methodology. The datasets and model architectures used are described in Sections 3.1 and 3.2, respectively. Section 3.3 presents an example of a visual explanation generated by the method. Section 3.4 summarizes the results obtained using various XAI metrics to validate the approach. Section 3.6 describes an experiment designed to enforce specific feature importance and verify whether the method correctly highlights them. Section 3.7 explains how to generate global explanations with DiffLIME, and finally, Section 3.8 highlights how this method can be practically beneficial for engineers.

3.1. Datasets

To conduct the experimental phase, two well-known datasets were utilized: the CWRU dataset and the JNUB dataset. The CWRU dataset (*Bearing Data Cente Case School of Engineering; Case Western Reserve University*, n.d.), provided by Case Western Reserve University, is a widely used benchmark for fault diagnosis in rotating machinery. It contains vibration data collected from an experimental setup consisting of a motor, a torque transducer, and a dynamometer. Faults were artificially introduced in the drive-end and fan-end bearings with varying severity levels and fault types, including inner race, outer race, and ball defects. The data was recorded under different operating conditions, such as varying loads and speeds, providing a diverse set of scenarios for evaluating diagnostic algorithms. The signals were sampled at high frequencies, ensuring sufficient resolution for both time and frequency domain analyses.

The JNUB dataset (Li, Ping, Wang, Chen, & Cao, 2013) includes vibration data collected under various fault conditions, including inner race, outer race, and ball defects, across different severity levels. The data was recorded using a motor-driven test rig equipped with sensors placed on the bearings to monitor vibration. The signals were captured at different operating speeds and loads, offering a comprehensive representation of real-world fault scenarios in rotating machinery.

For the experiments, a signal length of 128 points was used. Figure 5 illustrates examples of signals categorized by fault type, highlighting the differences in signal characteristics between the two datasets. These variations are further emphasized in Figure 6, which presents the 10 envelope centroids extracted from both datasets using KMeans.

The number of clusters was set to 10, aligning with the number of categories in the datasets. However, this choice may need to be re-evaluated depending on the specific characteristics of each dataset. A more thorough analysis would be required to determine the optimal number of clusters for different cases. Although both datasets focus on rotating machinery and vibration signals, they exhibit significant differences in their signal patterns, particularly in the shape of their envelope representations. These variations arise from factors such as differences in operating conditions, fault severities, sensor placements, and data acquisition setups, which can influence the clustering process and the interpretability of the results.

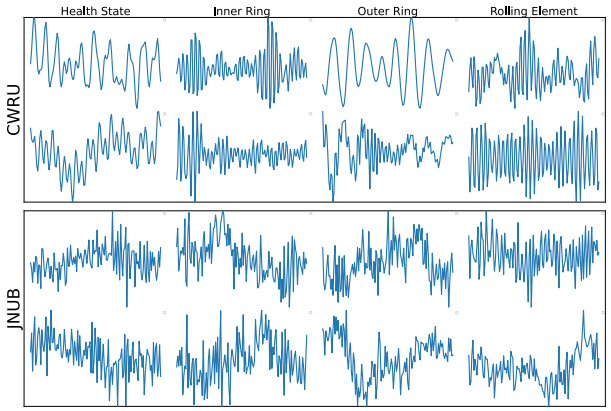

Figure 5. A few samples of dataset signals. (Top) CWRU signals (Bottom) JNUB signals.

3.2. Models to analyze

For each dataset, a 1D-CNN and an LSTM network were trained to predict faults.

The 1D-CNN architecture consists of three convolutional blocks, each containing three consecutive convolutional layers fol-

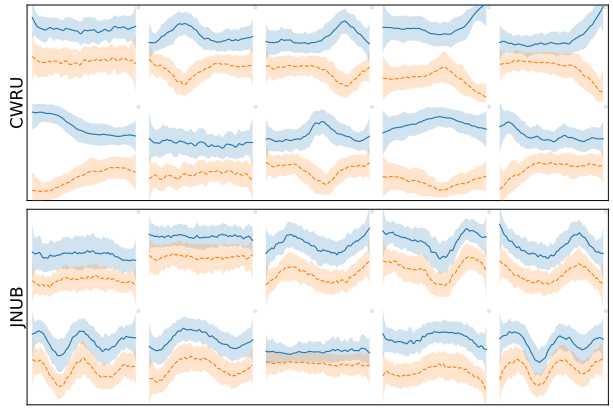

Figure 6. Top) CWRU centroid envelopes. (Bottom) JNUB centroid envelopes. Solid lines represent the upper envelope, while dashed lines represent the lower envelope.

lowed by a max-pooling layer. After the convolutional blocks, a flattening layer is applied, followed by two fully connected (dense) layers before the output layer. The network is trained using the Adam optimizer for up to 200 epochs, with the same configuration applied to both datasets.

The LSTM network consists of two stacked LSTM layers, each with 64 units. The output of the second LSTM layer is flattened and then follows the same fully connected structure as the 1D-CNN.

Figure 7 displays the confusion matrices for the fault prediction models applied to the CWRU and JNUB datasets. It is important to note that the models were not extensively optimized, as the primary focus of this work lies in the evaluation of the XAI method rather than achieving the highest possible predictive performance.

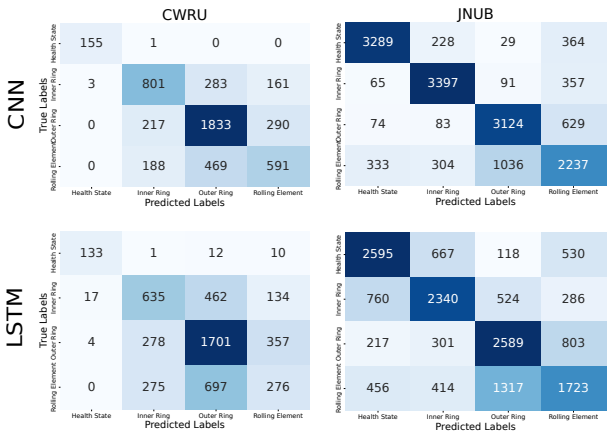

Figure 7. Confussion Matrices of the CWRU and JNUB fault prediction models

3.3. Local Explanation

By incorporating meta-features into the local prediction analysis, the explainability graph can be extended to include these meta-features alongside the signal data points. Figure 8 illustrates an example of a local explanation, highlighting the signal region that DiffLIME identifies as the most relevant for the prediction of the model.

Additionally, the graph displays the importance of the meta-features, which contribute 19% of the total importance in this example. The five frequencies with the highest magnitudes are presented in a frequency graph, complementing the frequency importances. Similarly, the envelope cluster assigned to the signal is shown, allowing for a comparison of its importance relative to the other nine envelope clusters.

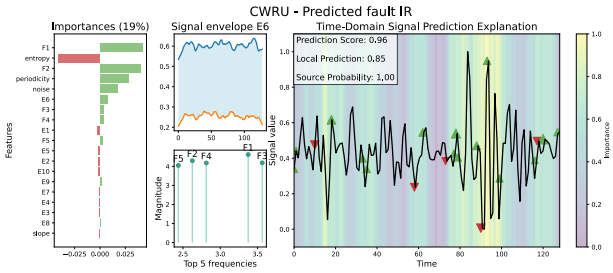

Figure 8. Local explanation of the CWRU prediction for an inner ring fault (correctly classified). E_x corresponds to the envelope centroids, while F_x represents one of the top five frequencies.

3.4. DiffLIME validation

Tables 1 and 2 summarize the results of the validation metrics for the proposed method compared to the baseline LIME (Schlegel et al., 2019) and TS-MULE (Schlegel et al., 2021), applied to both the CWRU and JNUB datasets. The metrics evaluated include coherence, selectivity, and stability, which collectively assess the quality and reliability of the explanations generated (Solís-Martín, Galán-Páez, & Borrego-Díaz, 2023). The experiments were executed 10 times to ensure statistical robustness, reducing the impact of variability in the results. The mean and standard deviation of each metric are reported to provide a comprehensive evaluation of performance consistency.

For the coherence metric, which quantifies the decrease in prediction confidence when relevant features are excluded, DiffLIME outperformed both LIME and TS-MULE across all datasets and network architectures. The improvement, although slight, was consistent for both datasets and model types.

Regarding selectivity, which measures the impact of removing non-important features on the prediction, TS-MULE outperformed both LIME and DiffLIME. This is likely due to the optimization-based approach of TS-MULE, which systemat-

Model	Method	Coherence	Selectivity	Stability
1D-CNN	LIME	0.365 ± 0.010	0.702 ± 0.013	1.035 ± 0.004
1D-CNN	TS-M	0.331 ± 0.019	0.669 ± 0.011	0.727 ± 0.007
1D-CNN	Ours	$\mathbf{0.380 \pm 0.018}$	$\mathbf{0.707 \pm 0.010}$	$\mathbf{0.507 \pm 0.009}$
LSTM	LIME	0.209 ± 0.005	0.674 ± 0.003	0.652 ± 0.005
LSTM	TS-M	0.185 ± 0.004	$\mathbf{0.712 \pm 0.007}$	0.350 ± 0.102
LSTM	Ours	$\mathbf{0.213 \pm 0.005}$	0.677 ± 0.007	$\mathbf{0.307 \pm 0.004}$

Table 1. Comparison of the proposed DiffLIME method with standard LIME and TS-MULE (TS-M) across multiple metrics for the CWRU dataset. The evaluated metrics include coherence, selectivity, and stability, which assess the quality and reliability of the explanations. Bold values indicate the best performance for each metric.

Model	Method	Coherence	Selectivity	Stability
1D-CNN	LIME	0.254 ± 0.005	0.835 ± 0.009	0.755 ± 0.006
1D-CNN	TS-M	0.263 ± 0.006	$\mathbf{0.848 \pm 0.007}$	0.410 ± 0.007
1D-CNN	Ours	$\mathbf{0.282 \pm 0.006}$	0.842 ± 0.009	$\mathbf{0.383 \pm 0.004}$
LSTM	LIME	0.175 ± 0.002	0.750 ± 0.007	0.572 ± 0.004
LSTM	TS-M	0.181 ± 0.004	$\mathbf{0.762 \pm 0.006}$	0.450 ± 0.010
LSTM	Ours	$\mathbf{0.183 \pm 0.002}$	0.755 ± 0.005	$\mathbf{0.309 \pm 0.002}$

Table 2. Comparison of the proposed DiffLIME method with standard LIME and TS-MULE (TS-M) across multiple metrics for the JNUB dataset. The evaluated metrics include coherence, selectivity, and stability, which assess the quality and reliability of the explanations. Bold values indicate the best performance for each metric.

ically perturbs the input signal to identify the most influential regions while minimizing the effect of irrelevant features. By leveraging meaningful perturbations and optimization strategies, TS-MULE enhances the ability of the model to focus on truly significant signal components, leading to superior selectivity performance.

The most significant difference between the methods was observed in the stability metric, which evaluates the consistency of explanations across similar instances. DiffLIME substantially outperformed LIME and TS-MULE in this regard, with lower stability values indicating greater robustness. This improvement is probably due to the use of the DPM model, which generates synthetic perturbations in a more controlled manner, ensuring that explanations remain consistent even when small variations are introduced in the input data.

Overall, the results highlight that DiffLIME provides more coherent and stable explanations compared to the baseline LIME and TS-MULE methods while maintaining a satisfactory level of selectivity.

3.5. Execution Time Analysis

To assess the computational efficiency of DiffLIME in comparison to LIME and TS-MULE, we measured the execution time required for each method across different datasets and models. Table 3 presents the results obtained for the CWRU and JNUB datasets using both CNN and LSTM models.

The results indicate that TS-MULE is the most computationally efficient method, consistently achieving the lowest execution times across all scenarios. LIME follows closely, with slightly higher execution times. In contrast, DiffLIME exhibits significantly longer execution times, being approximately 5 to 9 times slower than TS-MULE. This increase in computational cost is expected due to the incorporation of a diffusion-based generative process, which enhances the quality and stability of explanations at the expense of speed.

Additionally, it is important to highlight that the synthetic data generation and the training of the diffusion probabilistic model (DPM) require an initial computational overhead of approximately 1000 seconds. This one-time cost is incurred before DiffLIME can be applied to generate explanations. While this preprocessing step increases the overall computational burden, it enables DiffLIME to produce more coherent and stable explanations by leveraging the synthetic data generated by the DPM.

Dataset	Model	LIME (s)	TS-MULE (s)	DiffLIME (s)
CWRU	CNN	0.1600	0.1086	0.9538
CWRU	LSTM	0.1809	0.1163	0.9904
JNUB	CNN	0.1707	0.1105	0.9810
JNUB	LSTM	0.1691	0.1112	0.9919

Table 3. Comparison of execution times (in seconds) for LIME, TS-MULE, and DiffLIME across different datasets and models.

3.6. Testing the Method with Synthetic Forced Importances

To validate the proposed method, it is possible to create a synthetic dataset designed to enforce specific feature importances and evaluate whether DiffLIME can accurately detect these predetermined importances. For this purpose, 10 artificial target responses were generated using the CWRU dataset, each correlated with the envelope associated with a particular signal. The predictive model used for this evaluation was the same as the one described in later sections, and the resulting confusion matrix is shown in Figure 9.

DiffLIME was applied to the test set to generate explanation vectors, and the ranking of envelopes associated with the model predictions was computed. Figure 10 illustrates the ranking distribution for each envelope. In 40% of the cases, the most important envelope identified by DiffLIME matched the envelope directly correlated with the prediction, demonstrating the ability of the method to effectively highlight relevant features. Nearly 60% of the cases fell within the top 3 ranked envelopes, indicating strong alignment. Only envelope $E4$ appears to be consistently misranked, suggesting potential challenges in detecting its relevance.

3.7. Global explanation

While the proposed method focuses on local predictions, it is also possible to perform a global analysis by ranking the importance of meta-attributes per state. Figures 11 and 12 present the average ranking of meta-attributes for both datasets,

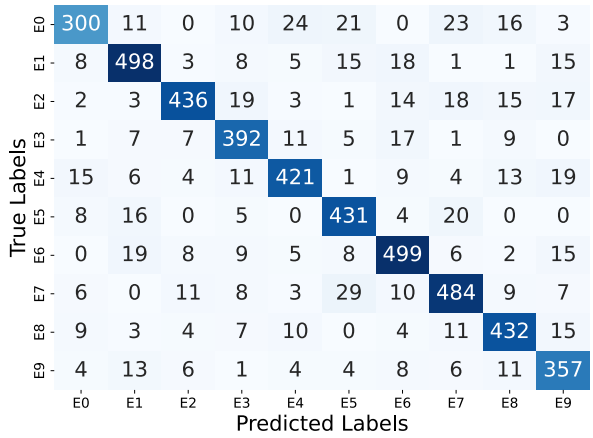

Figure 9. Confusion matrix for the model trained on the synthetic dataset with forced feature importances. The dataset was generated by correlating artificial target responses with specific signal envelopes from the CWRU dataset.

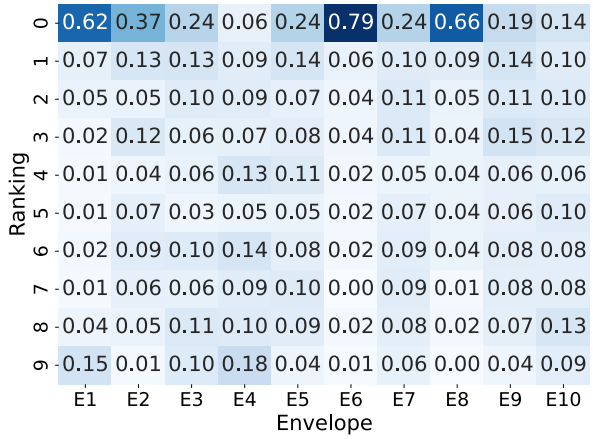

Figure 10. Envelope ranking results for the synthetic dataset using DiffLIME.

categorized by the health state and the rolling element fault state.

The results indicate that, for the CWRU dataset, frequencies with higher magnitudes are more associated with the fault state. However, this behavior is not observed in the JNUB dataset, where no such bias is apparent. Regarding the meta-attributes, features such as slope, entropy, noise, and periodicity show differing levels of influence on the predictions. Among these, slope exhibits minimal influence, whereas entropy and periodicity are significant contributors. Specifically, entropy dominates in the CWRU dataset for the health state, while periodicity is more influential in the JNUB dataset for the fault state.

The envelope clusters generally demonstrate medium-to-low importance with a balanced contribution across states. An exception is observed for $E1$, which, particularly in the JNUB dataset, emerges as a key characteristic associated with the fault state.

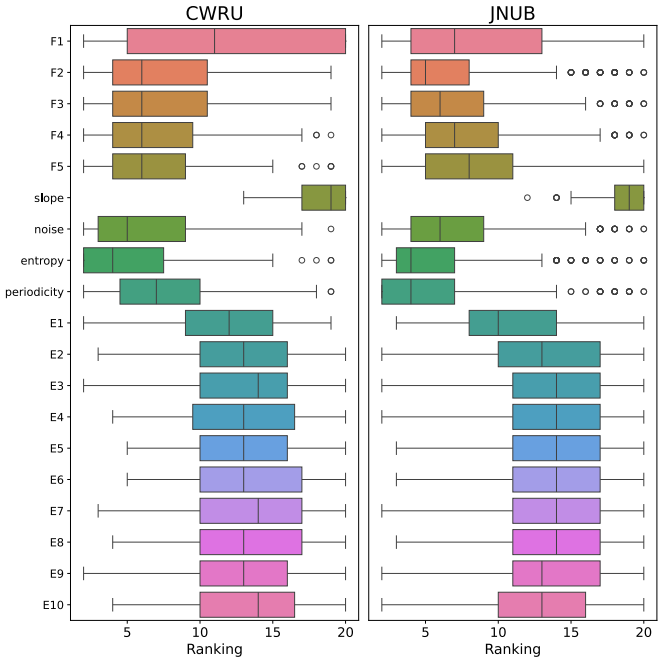

Figure 11. Meta-attributes ranking for health state in CWRU and JNUB datasets.

3.8. Engineer-Focused Explanations with DiffLIME

DiffLIME enhances fault detection in rotating machinery by providing explainability to deep learning models. When a model predicts a bearing fault, engineers need to understand the reasoning behind this decision. The goal of DiffLIME is to generate insights so engineers can verify whether the model relies on physically relevant features or if biases and artifacts influence the results.

For example, an engineer diagnosing a bearing fault in an industrial motor uses a deep learning model that classifies the machine as faulty. To confirm the reasoning, DiffLIME identifies high energy in a specific frequency band and an anomalous envelope pattern as key contributing factors. Since these align with known characteristics of inner race defects, the engineer can trust that the decision of the model is based on meaningful physical indicators. If DiffLIME instead highlighted irrelevant features, such as random noise, the engineer might suspect model bias or sensor anomalies, prompting further investigation. This transparency improves trust in AI-driven diagnostics, facilitates preventive maintenance, and helps refine predictive models for more reliable fault detection.

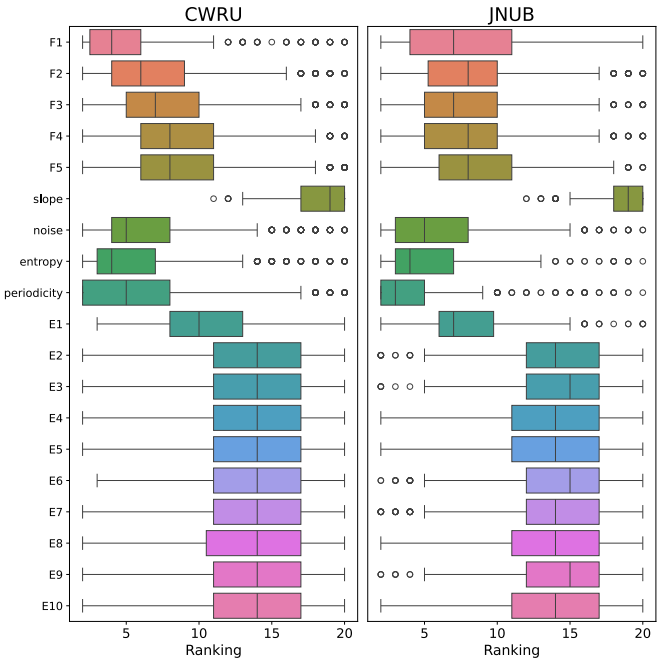

Figure 12. Meta-attributes ranking for rolling element fault in CWRU and JNUB datasets.

4. DISCUSSION

This study presents a novel approach to enhancing the explainability of machine learning models for fault diagnosis tasks, with a particular emphasis on time-series data. By extending the LIME algorithm with meta-attributes and leveraging the DPM for feature generation, we introduce a framework capable of providing robust and insightful explanations. Below, we discuss the implications of our results, the role of DPM in the analysis, limitations, and directions for future research.

The integration of the DPM played a critical role in this study. DPM was used to model the underlying data distribution and generate synthetic samples, which were instrumental in estimating meta-attributes such as noise, slope, and envelope probabilities. This probabilistic approach enabled a more comprehensive representation of the latent structure of the signal, bridging the gap between raw signal analysis and higher-level feature explainability.

One of the key contributions of this work is the incorporation of meta-attributes, such as slope, noise, entropy, periodicity, and envelope clustering, into the explanation process. These attributes, derived in part through the DPM, provide a structured understanding of how different signal characteristics contribute to model predictions. The analysis revealed that frequencies with higher magnitudes in the CWRU dataset dominate the importance rankings in the fault state, which aligns with the fundamental role of these frequencies in fault diagnosis signals.

The envelope clustering analysis, enhanced by the probabilistic insights from DPM, reveals that the earlier clusters ($E1$) hold significant importance, particularly in the JNUB dataset for fault states. This suggests that $E1$ captures key fault-related features critical for accurate predictions.

Our proposed approach builds upon LIME by introducing domain-specific enhancements for time-series data. The use of DPM differentiates this method from traditional LIME implementations and other model-agnostic methods like SHAP. While these methods primarily focus on static data features, the DPM-enhanced framework enables dynamic analysis of signal properties, making it better suited for fault diagnosis tasks.

Compared to deep learning-specific explanation methods, such as Grad-CAM or saliency maps, our approach provides a complementary perspective. It bridges the gap between high-level feature importance and detailed signal-level insights, ensuring that both the meta-attributes and signal structure are accounted for in the explanation.

Despite its advantages, this approach has several limitations. First, the computational overhead introduced by DPM and meta-attribute calculations can be substantial, especially for large datasets or real-time applications. While the probabilistic framework enhances explainability, it also increases the complexity of the explanation process. Nevertheless, the improved coherence and stability provided by DiffLIME justify its use in scenarios where interpretability quality is prioritized over execution speed. However, for applications requiring real-time or low-latency interpretability, TS-MULE or LIME may be more suitable due to their faster response times.

Additionally, the clustering of envelopes and the selection of meta-attributes require careful tuning to ensure generalizability across different datasets. The current approach assumes a static clustering structure, which may not adapt optimally to varying signal types or fault scenarios.

Future research could focus on optimizing the integration of DPM by reducing computational complexity or exploring lightweight generative models. Dynamic clustering methods could also be investigated to improve the adaptability of envelope analysis across diverse datasets.

Another promising direction involves extending the meta-attributes to incorporate temporal dependencies, allowing the framework to capture evolving fault characteristics over time. Hybrid methods combining DPM-enhanced LIME with other explanation techniques based on deep learning, such as Grad-CAM, could provide multi-level insights, blending global and local interpretability.

Finally, the application of this approach to broader domains, including predictive maintenance, anomaly detection, and health-

care time-series analysis, would further validate its versatility and robustness.

5. CONCLUSION

This work demonstrates the potential of combining a DPM with an extended LIME algorithm to improve explainability in fault diagnosis tasks. By incorporating meta-attributes derived from probabilistic modeling, the proposed approach generates more comprehensive visualizations that integrate time-domain, frequency-domain, and other relevant features, providing deeper insights into model predictions. The method has been evaluated against baselines LIME and TS-MULE using three explainability metrics across two different datasets, demonstrating superior performance in two cases and comparable behavior in the other. While some challenges remain, this framework represents a significant advancement in enhancing the interpretability of machine learning models for fault diagnosis.

DATA AVAILABILITY STATEMENT

The datasets used in this work are publicly available. To access the datasets, the tool *phmd* (Solís-Martín, Galán-Páez, & Borrego-Díaz, 2025) was used. All the source code necessary to reproduce the experiments and results presented in this paper can be found in the GitHub repository[1].

ACKNOWLEDGMENT

Grant PID2023-147198NB-I00 funded by MICIU/AEI/10.13039/501100011033 (Agencia Estatal de Investigación) and by FEDER, UE, and by the Ministry of Science and Education of Spain through the national program "Ayudas para contratos para la formación de investigadores en empresas (DIN2019-010887 / AEI / 10.13039/50110001103)", of State Programme of Science Research and Innovations 2017-2020.

REFERENCES

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., ... others (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, *58*, 82–115.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, *10*(7), e0130140.

Bearing Data Cente Case School of Engineering; Case Western Reserve University. (n.d.). `https://engineering.case.edu/bearingdatacenter`. (Accessed 08-04-2024)

Brito, L. C., Susto, G. A., Brito, J. N., & Duarte, M. A. V. (2023). Fault diagnosis using explainable ai: A transfer learning-based approach for rotating machinery exploiting augmented synthetic data. *Expert Systems with Applications*, *232*, 120860.

Decker, T., Lebacher, M., & Tresp, V. (2023). Does your model think like an engineer? explainable ai for bearing fault detection with deep learning. In *Icassp 2023-2023 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 1–5).

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. In *2018 ieee 5th international conference on data science and advanced analytics (dsaa)* (pp. 80–89).

Goodman, B., & Flaxman, S. (2017). European union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, *38*(3), 50–57.

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, *33*, 6840–6851.

Li, K., Ping, X., Wang, H., Chen, P., & Cao, Y. (2013). Sequential fuzzy diagnosis method for motor roller bearing in variable operating conditions based on vibration analysis. *Sensors*, *13*(6), 8013–8041.

Meng, H., Wagner, C., & Triguero, I. (2023). Explaining time series classifiers through meaningful perturbation and optimisation. *Information Sciences*, *645*, 119334.

Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for activation functions. *arXiv preprint arXiv:1710.05941*.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 1135–1144).

Rudin, C., & Radin, J. (2019). Why are we using black box models in ai when we don't need to? a lesson from an explainable ai competition. *Harvard Data Science Review*, *1*(2), 1–9.

Santos, M. R., Guedes, A., & Sanchez-Gendriz, I. (2024). Shapley additive explanations (shap) for efficient feature selection in rolling bearing fault diagnosis. *Machine Learning and Knowledge Extraction*, *6*(1), 316–341.

Schlegel, U., Arnout, H., El-Assady, M., Oelke, D., & Keim, D. A. (2019). Towards a rigorous evaluation of xai methods on time series. In *2019 ieee/cvf international conference on computer vision workshop (iccvw)* (pp. 4197–4201).

Schlegel, U., Vo, D. L., Keim, D. A., & Seebacher, D. (2021). Ts-mule: Local interpretable model-agnostic explanations for time series forecast models. In *Joint european conference on machine learning and knowledge discovery in databases* (pp. 5–14).

[1] DiffLIME GitHub source code: https://github.com/dasolma/diffLIME

Scott, M., Su-In, L., et al. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, *30*, 4765–4774.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the ieee international conference on computer vision* (pp. 618–626).

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2020). Grad-cam: visual explanations from deep networks via gradient-based localization. *International journal of computer vision*, *128*, 336–359.

Siddiqui, S. A., Mercier, D., Munir, M., Dengel, A., & Ahmed, S. (2019). Tsviz: Demystification of deep learning models for time-series analysis. *IEEE Access*, *7*, 67027–67040.

Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Solis-Martin, D., Galan-Paez, J., & Borrego-Diaz, J. (2023). D3a-ts: Denoising-driven data augmentation in time series. *arXiv preprint arXiv:2312.05550*.

Solís-Martín, D., Galán-Páez, J., & Borrego-Díaz, J. (2023). On the soundness of xai in prognostics and health management (phm). *Information*, *14*(5), 256.

Solís-Martín, D., Galán-Páez, J., & Borrego-Díaz, J. (2025). Phmd: An easy data access tool for prognosis and health management datasets. *SoftwareX*, *29*, 102039. doi: https://doi.org/10.1016/j.softx.2025.102039

Vollert, S., Atzmueller, M., & Theissler, A. (2021). Interpretable machine learning: A brief survey from the predictive maintenance perspective. In *2021 26th ieee international conference on emerging technologies and factory automation (etfa)* (pp. 01–08).

Wang, Z., Yan, W., & Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. In *2017 international joint conference on neural networks (ijcnn)* (pp. 1578–1585).

Zereen, A. N., Das, A., & Uddin, J. (2024). Machine fault diagnosis using audio sensors data and explainable ai techniques-lime and shap. *Computers, Materials and Continua*, *80*(3), 3463–3484.