Towards Objectively Interpretable Fault Diagnosis for Time-Series Data in Grinding

Regular Paper

Tsan Tsai Chan[1*], Kai Lange[1], Ruoxuan Liu[1], Alessa Wein[1], Nina Keßler[1],
Maxx Richard Rahman[1,2], Wolfgang Maass[1,2]
[1]*Saarland University,* [2]*German Research Center for Artificial Intelligence (DFKI)*

*Corresponding author: tsch00001@uni-saarland.de

Introduction

Manually diagnosing mechanical faults is highly laborious, and a lack of strategies to intuitively explain model decisions has precluded effective automated diagnosis by unsupervised systems. To remedy this, we propose an objectively interpretable probabilistic framework for analysing time-series data in grinding, using a Gaussian mixture model (GMM) trained only on normal processes to assign likelihoods to each 50-ms segment of a grinding signal. A dashboard visualises these likelihoods in sequence, showing where during a process anomalies crop up and yielding diagrams that could expedite visual fault diagnosis. Anomalous grinding runs are objectively defined by deviations from normal training data, facilitating predictive maintenance. We show that our framework allows various simple GMM-based architectures to outperform a more complex recurrent one in stability of F2-scores on small training samples with simulated anomaly data.

Motivating Explainable Unsupervised Methods for Anomaly Detection

Grinding machines, or grinders, are widely used in various industries and rely on an abrasive wheel to bring about very precise changes to workpiece surfaces. Anomalies arise when a grinder behaves atypically, which could be manifested in defective workpieces, unusual patterns in energy consumption, or acoustic emissions (Lopes et al., 2018). Crucially, anomalies often indicate deeper mechanical faults that could, if left unattended to, lead to machine damage and production delays costing tens of thousands of euros (Kaufmann et al., 2020).

As the same type of anomaly may have different causes (Griffin et al., 2017), unguided manual inspection of machine parts to find the root cause of problems is generally laborious and time-consuming. This necessitates methods that clearly show what features in the grinding signal are deviant and may represent a fault in which parts of the machine.

In addition, real anomaly data is very scarce (Pang et al., 2021) and impractical to exhaustively simulate through costly methods such as nital etching. As unsupervised methods only need normal, i.e. non-anomalous, data for training, they are preferable to those that work with labelled data.

Related Work

Unsupervised anomaly detection methods for grinding and other time-series data commonly implicate various flavours of autoencoder and recurrent neural networks (RNNs), gated recurrent units (GRUs) in particular (Choi et al., 2021). Both learn meaningful representations of normal data in order to detect deviations in new data but have different strengths. Autoencoders excel at dimensionality reduction and anomaly detection using reconstruction distance (cf. Kieu et al., 2021), while RNNs crucially take into account temporal dependencies in their encodings.

Nevertheless, being black-box models, the decisions of both autoencoders and RNNs are not inherently interpretable. Existing strategies to improve the explainability of these models mainly concentrate on estimating feature contributions to decisions. The best known might be SHapley

Additive exPlanations (SHAP, Lundberg & Lee 2017) and Local Interpretable Model-agnostic Explanations (LIME, Ribeiro et al., 2016), which visualise feature weights in a model-agnostic fashion. However, their outputs are open to subjective interpretation (Li et al., 2023) and require extensive postprocessing to be actionable. This prevents current explainability methods from directly facilitating fault diagnosis (cf. Theissler et al., 2022).

What holds more promise for interpretable fault diagnosis is the fact that autoencoders and GRUs have been combined with GMMs to excellent effect in time-series anomaly detection (e.g. Zong et al., 2018, Serradilla et al., 2021, Zhu et al., 2023). The probabilistic properties of GMMs can be used to objectively define anomalies based on reconstruction distance. Unfortunately, the intrinsic interpretability of such an approach has not been sufficiently exploited for diagnostic purposes.

In addition, to our best knowledge, the complex architectures named above have not been evaluated on whether they are too computationally demanding for real-time use and how effective they would be on small training samples. Real-time fault detection is preferable because the lags involved in offline methods may allow underlying problems to worsen unnoticed. Also, modestly sized training samples enable the system to be retrained for use on different machines for scaling up, without first necessitating large amounts of training data from those machines being collected.

This Study

In the light of the above, we propose a GMM-based system that analyses grinding signals one segment at a time, capitalising on the explainability inherent to the probabilistic approach for visual fault diagnosis. Making our case using a convolutional autoencoder (CAE)-GMM, we show that this system can be coupled with various relatively simple architectures for real-time fault detection with modest training sample sizes, outperforming more complex state-of-the-art models.

Problem Statement

Each grinding run is represented by a sequence of unlabelled tensors containing three features each (x_1, x_2, x_3). Our anomaly detection system needs to be trained on tensors from normal grinding runs to learn a threshold separating such normal runs from deviant, i.e. anomalous, ones. The system will then be evaluated on labelled data consisting of tensors with the same three features, represented as (x_1, x_2, x_3, n), where n is the class label (0 for 'normal' and 1 for 'anomalous').

The ultimate goal is to minimise cross-entropy loss, i.e. to make as few wrong predictions as possible where the predicted ($\hat{y}$) and actual labels (y) match given the input features **x**:

$$\hat{y} = \arg\min_{y \in \{0,1\}} -[y \log \hat{y} - \ (1-y) \ \log(1-\hat{y})]$$

Background on Data

Our data comes courtesy of a large engineering company. It consisted of 258 pairs of acoustic emissions (AE; sampling rate 100 KHz, 1 channel) and electrical current (EC; 2000 KHz, 4 channels) recordings from a Tschudin-T25 grinder used to produce diesel injection valves. Each AE-EC pair pertained to the same grinding run, with all recordings in PARQUET format.

229 pairs of recordings reflected normal runs, while only 29 were of a simulated anomaly. The simulated anomaly data was recorded with the screws of the grinder's rollers adjusted in order to approximate roller failure. It was linearly separable from the normal class, which the literature suggests is not realistic (cf. Rameshkumar et al., 2021). Hence, a balanced validation set comprising 5 normal and 5 anomalous pairs of recordings was used to fine-tune only the number of input features. Other hyperparameters (anomaly thresholds, segment length, number of layers)

were decided independently to avoid overfitting on the simulated roller failure data. However, we still involve recall as a proxy for model performance later for want of more accurate data.

AE recordings were made using a Kistler Piezotron type 8152C1 sensor on the machine's tailstock. Three of the four channels in the EC recording correspond to input from a Chauvin Arnoux MN39AS current clamp, one for each cable of the three-phase grinder, while the fourth is the root mean square (I_{RMS}) of these currents. We only used the I_{RMS} channel of the EC recordings.

Example Implementation of Our Visual Fault Diagnosis System with a CAE-GMM

Data Preprocessing

In line with the structure of our data, our system assumes that each grinding run is represented by one pair of AE and EC recordings of the same length. Each AE-EC pair is aligned and split into an equal number of temporally matching 50-ms segments. We extract three features from all paired segments, AE variance and energy, and EC energy. The variance of a segment is the mean squared difference of all its data points from the segment mean, while energy is the sum of the squared data points in a segment. These features are normalised, then stored in one tensor representing a 50-ms timestep. Should paired recordings be of unequal length, segments beyond the last one of the shorter recording are discarded. Using only a small number of named features will contribute to interpretability of model decisions later.

Despite being correlated, these features were selected because AE RMS (derived from variance) and energy are both important to quantifying wear in grinders (Rameshkumar et al., 2021), and EC energy is less correlated with the AE features than is EC variance. To address the correlation, we add a CAE on top of a GMM for our example implementation, to which we now turn.

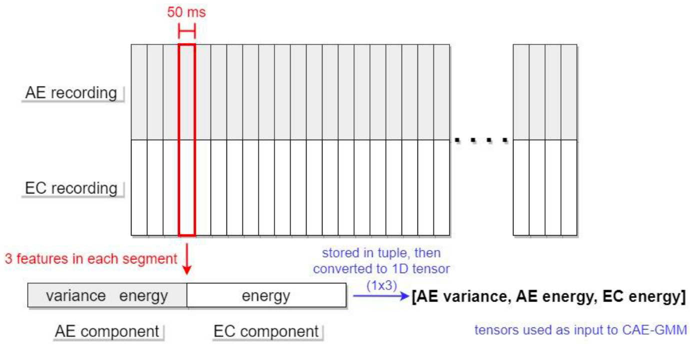
50 ms
AE recording
EC recording
3 features in each segment
stored in tuple, then converted to 1D tensor
(1x3)
variance energy
energy
[AE variance, AE energy, EC energy]
AE component
EC component
tensors used as input to CAE-GMM

Figure 1. How AE and EC recordings are segmented and 3 features extracted

Add-on Module – The Convolutional Autoencoder (CAE)

As mentioned, the CAE module disentangles correlated features in the input tensors. In the training phase, this allows normal training data to form a spherical cluster in 3D, optimising learning by the GMM in the next module. The decorrelation is achieved using an encoder-decoder architecture:

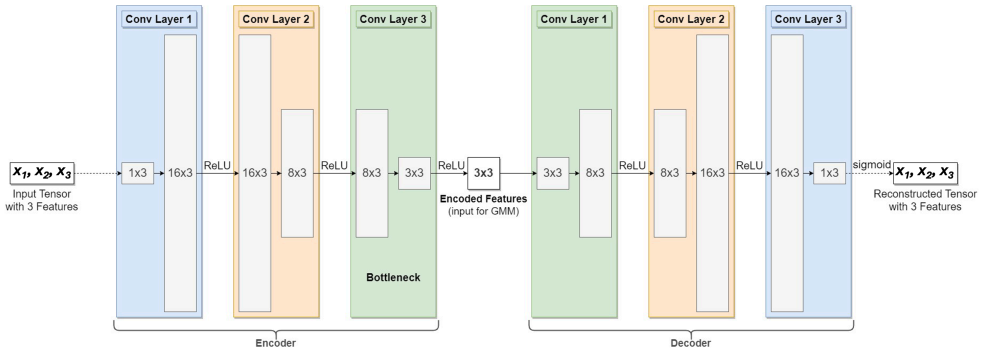

Figure 2. Simplified representation of our CAE module showing dimensions per layer

The encoder, consisting of three 1D convolutional layers with ReLU activations, transforms the 1D input into a 2D representation at the bottleneck layer. This process can be mathematically represented as follows, with h$^{(l)}$ standing for the lth hidden layer, W$^{(l)}$ the weights matrix of that layer and b$^{(l)}$ the bias being applied to h$^{(l-1)}$, the immediately preceding hidden layer:

$$h^{(l)} = \mathrm{ReLU}\left(\mathrm{Conv1D}\left(h^{(l-1)}, W^{(l)}, b^{(l)}\right)\right)$$

The decoder subsequently reconstructs the original input from the encoded representation using the same architecture as the encoder but in reverse, the reconstructions marked with a hat:

$$\widehat{h^{(l)}} = \mathrm{ReLU}\left(\mathrm{Conv1D}\left(\widehat{h^{(l-1)}}, \widehat{W^{(l)}}, \widehat{b^{(l)}}\right)\right)$$

The optimisation objective is to minimise the mean squared error (MSE) between the original (x_i) and reconstructed segments ($\hat{x}_i$), ensuring that the 2D encodings generated by the encoder are an accurate representation of the original inputs:

$$\min_{\theta} MSE\ , where\ MSE\ = \frac{1}{\mathrm{n}} \sum_{\mathrm{i=1}}^{\mathrm{n}} (x_i\ -\ \widehat{x_\iota})^2$$

With the Adam optimiser, the CAE trains for 100 epochs at a learning rate of 0.001.

Core Module – The Gaussian Mixture Model (GMM)

The GMM module processes the 2D encoded representation from the CAE. In the training phase, its chief purpose is to learn the distribution of the encoded normal segments, minimising the negative log-likelihood scores of those segments and classing those with scores in the 97.5th percentile as anomalous. For testing, it assigns scores to and labels as anomalous new segments above the threshold 97.5th-percentile score it learnt from the distribution of normal segments.

Our GMM has a full-rank covariance matrix, allowing it to capture the correlations of all combinations of encoded features. It only contains one mixture component, in line with our task to differentiate one type of normal data from all types of anomalous data. With only one component, the probability density function of the *d*-dimensional input vector ($\mathbf{x}$) coming from a Gaussian distribution with mean vector ($\boldsymbol{\mu}$) and covariance matrix ($\boldsymbol{\Sigma}$) is given by:

$$p(\mathbf{x}) = \mathcal{N}(\,\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}\,), \text{ where } \mathcal{N}(\,\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}\,) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right).$$

Postprocessing

After having been assigned negative log-likelihood scores by the GMM, segments representing the same grinding run are regrouped and the proportion of anomalous segments in each training sample is measured. During training, the proportions in the normal data are recorded and stored as an array. Grinding runs with a higher proportion of anomalous segments than 99.99% of the normal training data – our second threshold – are classed as anomalous runs. As a note, both this and the first threshold are inspired by standard p-value thresholds for statistical significance.

In the testing phase, new segments are passed through the CAE and GMM modules, assigned negative log-likelihood scores and labelled anomalous where they are more positive than 97.5% of the normal training segments, then regrouped by grinding run. The proportions of anomalous segments in each grinding run are compared against the array of proportions learnt during training – again, runs having a higher proportion than 99.99% of the normal training runs are flagged as anomalous runs based on this objective threshold. Finally, a dashboard visualises the negative log-likelihood scores output by the GMM, as the next section will demonstrate.

Visualisation of Anomaly Scores with a Dashboard

The negative log-likelihood scores returned by the GMM for each segment are visualised in real time as bars by a dashboard in sequential order. Non-anomalous scores below the 97.5th percentile of normal training data are concentrated below the horizontal axis and marked grey, while anomalous ones are invariably above that axis and coloured red. The length of each bar corresponds to the absolute value of the score. Normal grinding runs thus look much as in Figure 3, with most segments represented by grey bars extending downwards from the horizontal axis:

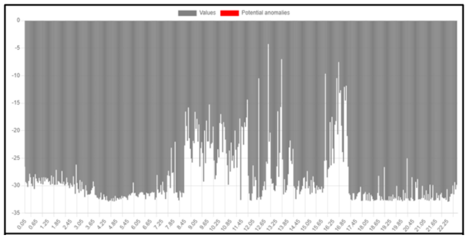

Figure 3. A normal grinding process with a preponderance of grey bars below the horizontal axis

For interpretability, two values are returned for each grinding sample used for testing – first, the proportion of anomalous segments – as defined during the training phase – in the sample and second, the percentage of normal training data that the test sample has a higher proportion of anomalous segments than – runs the system labels anomalous are stated to contain 'more anomalous segments than 100% of the normal training data'. The latter is designed to facilitate predictive maintenance, as steadily increasing percentiles could indicate wear warranting attention.

Potential for Visual Fault Diagnosis

In contrast to the normal grinding run pictured in Figure 3, anomalous runs featuring simulated roller failure consistently showed red bars, marking highly anomalous segments, extending far *upwards* from the horizontal axis in the final quarter of the run. Figure 4 shows two examples:

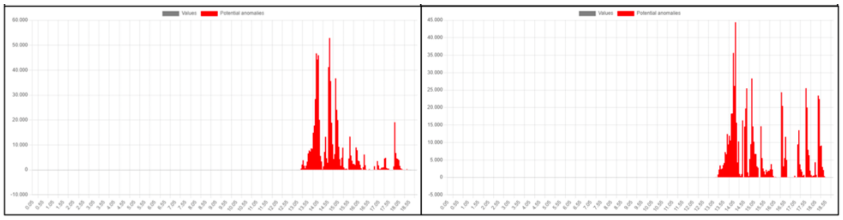

Figure 4. Two runs with simulated roller failure, showing long red bars in the last quarter

The red bars visible in Figure 4 were in fact of such magnitude as to render the foregoing grey bars, representing normal segments, barely visible. This consistent visual signature associated with all 29 simulated anomalies in our dataset may imply that other types of grinding anomaly implicating different parts of the machine could take on distinct shapes when visualised this way. Though speculative, this could massively expedite the diagnosis of mechanical faults if correct.

Experiment Comparing Explainable Architectures

Objectives

Having shown that a CAE-GMM is one possible implementation of our system, we now compare its performance against five other architectures whose outputs are amenable to the exact same type of visualisation. This is meant to show how compatible our objectively interpretable system is with different explainable architectures, and what level of complexity it requires of these models.

		Specifications				
Baseline model	**References consulted**	***Latent dimension***	***Hidden layers***	***Units/ layer (encoder)***	***Units/ layer (decoder)***	***Activation functions***
Standalone GMM	—	Full-rank covariance matrix, 1 component (applies to all other instances of GMMs used in this study)				
PCA-GMM	—	3 principal components				
DAE-GMM (noise = 0.3)	Zong et al., 2018	16	2	Linear (32, 16)	Linear (16, 32)	ReLU
VAE	Chettri et al., 2020	3	3	Conv1D (16, 8), Linear (128)	Linear (128, 24), ConvTranspose1D (16, 1)	ReLU, Sigmoid (final)
GRU-GMM	Zhu et al., 2023	—	3	GRU (256)		—

Table 2. Specifications of models tested with our system

These are namely – a standalone GMM; Principal Component Analysis (PCA) and a denoising autoencoder (DAE) each combined with a GMM; a variational autoencoder (VAE), which is a probabilistic autoencoder; and a GRU-GMM, whose recurrent architecture explicitly models temporal dependencies to cope with time-series data. They were specifically chosen to decide if our CAE module could be substituted with a more effective alternative, entirely left out (GMM), or the whole model replaced by something else altogether (VAE), the proviso being that the alternative model should produce output no less objectively interpretable than the CAE-GMM's.

Procedure and Metrics

The architectures were evaluated on the same test set that consisted of 24 normal and 24 anomalous grinding runs and did not overlap with the validation set referred to above. However, we varied the size of the training set, training each architecture in turn on 10, 20, 40, 100 and 200 normal grinding runs to yield five models per architecture.

We started by preprocessing the recordings as detailed in the 'Data Preprocessing' section above. Before being passed to the models, each feature was normalised relative to itself, the test data being normalised using the parameters of the training data to avoid leakage.

As our dataset is highly imbalanced, with only around 10% of data points representing the anomalous class, we use precision, recall and F2-score as our metrics. The last gives recall a weight four times that of precision, as we deemed false negatives more serious than false positives:

$$\mathrm{F}_2 = (1 + 2^2) \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{2^2 \cdot \mathrm{Precision} + \mathrm{Recall}}$$

These metrics were computed for each architecture on the small sample sizes stated above to find out how scalable each is. In addition, we measured the total time they took on an NVIDIA Tesla P100 PCIe 16 GB graphics processing unit (GPU) to complete training, and the mean inference time on all six sample sizes as a proxy for their computational efficiency and real-time capabilities.

Results and Interpretation

As Table 3 shows, all models are suited for real-time use, with inference times from 0.21 to 0.24s:

Model	Training Sample Size	Precision	Recall	F2-Score	Total Training Time	Mean Inference Time (s)
CAE-GMM	10	0.522	1.000	0.845	23 min 58 s	0.231
	20	0.522	1.000	0.845		
	40	0.827	1.000	0.960		
	100	**1.000**	**1.000**	**1.000**		
	200	**1.000**	**1.000**	**1.000**		
GMM	10	0.667	1.000	0.909	33 min 54 s	0.215
	20	0.649	1.000	0.902		
	40	0.632	1.000	0.896		
	100	**1.000**	**1.000**	**1.000**		
	200	**1.000**	**1.000**	**1.000**		
PCA-GMM	10	0.667	1.000	0.909	30 min 56 s	0.224
	20	0.649	1.000	0.902		
	40	0.632	1.000	0.896		
	100	**1.000**	**1.000**	**1.000**		
	200	**1.000**	**1.000**	**1.000**		
DAE-GMM	10	0.533	1.000	0.851	51 min 23 s	0.232
	20	0.706	1.000	0.923		
	40	0.686	1.000	0.916		
	100	1.000	0.833	0.862		
	200	1.000	0.000	0.000		
VAE	10	0.333	0.042	0.051	28 min 9 s	0.213
	20	0.333	0.042	0.051		
	40	1.000	0.042	0.051		
	100	1.000	0.042	0.051		
	200	1.000	0.042	0.051		
GRU-GMM	10	**1.000**	**1.000**	**1.000**	1 h 7 min 47 s	0.212
	20	**1.000**	**1.000**	**1.000**		
	40	1.000	0.667	0.714		
	100	**1.000**	**1.000**	**1.000**		
	200	1.000	0.000	0.000		

Table 3. Results of our model comparison

Of our six candidate models, the CAE-GMM has the shortest training time, 17.4% shorter than the next-best-performing model. This is probably testament to how effective the CAE module is in decorrelating the data for learning by the GMM. However, despite having longer training times, the standalone GMM and PCA-GMM had equally stable improvements in their F2-scores with increasing sample size in a low-data regime, and marginally lower inference times than the CAE-

GMM. This demonstrates that our objectively interpretable system performs optimally on the simulated anomaly data with these three relatively simple models, and which model is preferable would depend on user requirements, the specific dataset and practical constraints.

By contrast, the VAE consistently yielded low F2-scores. Also, the DAE-GMM and GRU-GMM, despite outperforming our CAE-GMM at sample sizes below 40, were unable to identify any simulated anomalies at 200 samples (F2 = 0), suggesting these architectures are inherently unstable. This last point makes clear that recurrent architectures considering temporal dependencies are for our purposes not inherently superior to simpler feedforward alternatives.

Conclusion

In the foregoing, we introduced an objectively interpretable Gaussian mixture model (GMM)-based framework for visualising anomalies in grinding processes, with significant implications for fault diagnosis. Splitting acoustic emissions (AE) and electrical current (EC) recordings into 50-ms segments and learning the distribution of three features (AE variance, AE energy, EC energy) from normal data, our system identifies significant deviations as anomalies using predefined thresholds. A dashboard visualises and colour-codes these anomalies, which makes them easily recognisable. As simulated roller failure yielded a consistent visual signature, our dashboard could also potentially expedite fault diagnosis. Requiring only modest numbers of normal samples for training and attaining optimal performance when coupled with relatively simple models like a convolutional autoencoder-GMM, our framework is easily adaptable to different applications.

We will however end with some of its limitations. First, less linearly separable data, preferably actual data representing more anomaly types, is needed to determine how far our visualisations could aid fault diagnosis. Next, although the small number of input features was intended to optimise explainability and efficiency, more features could be incorporated for greater robustness to novel anomalies, feature weights being estimated using existing techniques like SHAP. Additional desiderata include enhancing robustness to noise in the training data and using a dedicated channel for each input feature, all of which we will leave to future work. Despite these limitations, our system's objective visual explainability and implications for both diagnosis and predictive maintenance render it superior to existing alternatives for fault detection.

References

Chettri, B., Kinnunen, T., & Benetos, E. (2020). Deep Generative Variational Autoencoding for Replay Spoof Detection in Automatic Speaker Verification. *Computer Speech & Language*, 63, 101092.

Choi, K., Yi, J., Park, C., & Yoon, S. (2021). Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines. *IEEE Access*, 9, 120043-120065.

Griffin, J. M., Doberti, A. J., Hernández, V., Miranda, N. A. & Vélez, M. A. (2017). Multiple classification of the force and acceleration signals extracted during multiple machine processes: part 1 intelligent classification from an anomaly perspective. *The International Journal of Advanced Manufacturing Technology*, 93(1–4), 811–823.

Kaufmann, T., Sahay, S., Niemietz, P., Trauth, D., Maaß, W., & Bergs, T. (2020). AI-based framework for deep learning applications in grinding. In *18th IEEE World Symposium on Applied Machine Intelligence and Informatics (SAMI)* (pp. 195-200). IEEE.

Li, Z., Zhu, Y. & Van Leeuwen, M. (2023). A survey on explainable anomaly detection. *ACM Transactions on Knowledge Discovery from Data, 18*(1).

Lopes, W. N., Thomazella, R., Alexandre, F. A., De Aguiar, P. R., Bianchi, E. C., de Pontes, B. R., & Viera, M. A. A. (2018). Monitoring of self-excited vibration in grinding process using time-frequency analysis of acceleration signals. In *13th IEEE International Conference on Industry Applications (INDUSCON)* (pp. 659-663). IEEE.

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.

Pang, G., Shen, C., Cao, L., & Hengel, A. V. D. (2021). Deep learning for anomaly detection: A review. *ACM Computing Surveys (CSUR)*, 54(2), 1-38.

Serradilla, O., Zugasti, E., De Okariz, J. R., Rodriguez, J. & Zurutuza, U. (2021). Adaptable and Explainable Predictive Maintenance: Semi-Supervised Deep Learning for Anomaly Detection and Diagnosis in Press Machine Data. *Applied Sciences*, *11*(16), 7376.

Rameshkumar, K., Mouli, D. S. B., & Shivith, K. (2021). Machine learning models for predicting grinding wheel conditions using acoustic emission features. *SAE International Journal of Materials and Manufacturing*, 14(4), 387-406.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). "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).

Theissler, A., Spinnato, F., Schlegel, U., & Guidotti, R. (2022). Explainable AI for time series classification: a review, taxonomy and research directions. *IEEE Access*, 10, 100700.

Zhu, X., Yang, C., Yang, C., Gao, D., & Lou, S. (2023). An unsupervised fault monitoring framework for blast furnace: Information extraction enhanced GRU-GMM-autoencoder. *Journal of Process Control*, 130, 103087.

Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., & Chen, H. (2018, February). Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection. In *International Conference on Learning Representations*.