Explainable AI to Improve Machine Learning Reliability for Industrial Cyber-Physical Systems

Annemarie Jutte[1,2*][0009−0004−9322−0674] and Uraz Odyurt[3*][0000−0003−1094−0234]

[1] Ambient Intelligence Group, Saxion University of Applied Sciences, Enschede, The Netherlands
[2] Department of Computer Science, University of Twente, Enschede, The Netherlands a.m.p.jutte@saxion.nl
[3] Faculty of Engineering Technology, University of Twente, Enschede, The Netherlands u.odyurt@utwente.nl

Abstract. Industrial Cyber-Physical Systems (CPS) are sensitive infrastructure from both safety and economics perspectives, making their reliability critically important. Machine Learning (ML), specifically deep learning, is increasingly integrated in industrial CPS, but the inherent complexity of ML models results in non-transparent operation. Rigorous evaluation is needed to prevent models from exhibiting unexpected behaviour on future, unseen data. Explainable AI (XAI) can be used to uncover model reasoning, allowing a more extensive analysis of behaviour. We apply XAI to improve predictive performance of ML models intended for an industrial CPS use-case. We analyse the effects of components from time-series data decomposition on model predictions using SHAP values. Through this method, we observe evidence on the lack of sufficient contextual information during model training. By increasing the window size of data instances, informed by the XAI findings for this use-case, we are able to improve model performance.

Keywords: Cyber-physical systems, Industry, Explainable AI, Time-series data, ML model development

1 Introduction

Cyber-Physical Systems (CPS), particularly industrial CPS, are considered as sensitive infrastructure, both in terms of safety and economic security. Solutions incorporating Machine Learning (ML) are often utilised in the lifecycle of industrial CPS, e.g., for health monitoring, and therefore must be highly reliable.

Considering the ML model component of CPS, evaluation methods beyond common ML model performance metrics, i.e., prediction accuracy, F1 score, etc., are in demand. While these metrics indicate how well a model performs on the test data, they do not reveal *why* it performs well or demonstrate its ability to generalise to *unseen data*. Consequentially, spurious correlations used by the

* These authors contributed equally to this work.

model, which are present in both training and test data, but are not actual causal relationships, cannot be detected. Understanding the relations a model relies on is important for robust evaluation and alignment with domain knowledge.

Deep neural networks, often employed as ML solutions, are considered black boxes due to their complexity [1]. Evaluating such models' decision rationale is non-trivial. Explainable AI (XAI) techniques such as SHAP [16] and LIME [20] can be used to provide insights into the reasoning of AI-based systems [1].

XAI can be used during model development to inform the design process, and during model deployment to support human decision making. By understanding model behaviour, training hyperparameters can be adjusted based on empirical deductions, rather than design-space search methodologies. *We focus on the use of XAI for informed model development.* It must be mentioned that the high computational cost is the downside. Considering the context of our use-case, ensuring reliability of CPS is far more important than the computational cost.

Application of XAI methods to practical CPS use-cases remains limited. The context of industrial CPS is especially interesting as time-series data, commonly associated with such systems, encapsulates complex behaviour. As a consequence, explanations regarding model-reasoning based on time-series might be less intuitive to understand than reasoning, for example, image data. One option is to explain model reasoning in terms of high-level human-interpretable concepts, specifically components from time-series decomposition [10].

Proposed solution We generate and incorporate SHAP values for the fine-tuning of the ML model module, a Convolutional Neural Network (CNN), in a Fault Detection and Identification (FDI) solution. We specifically leverage decomposition of the time-series pseudo-signals[†] and well-argued adjustments to data formatting based on model's response to signal components. Given the component scoring, upon consistent reliance on a particular component, better representation of that component per data instance could have a positive effect. For instance, a wider data instance could better reflect certain component states, or a narrower data instance could result in better focus on abrupt variations.

Contribution We provide:

- The application of SHAP to an industrial CPS use-case. Specifically, the application of concept-based SHAP utilising a custom decomposition of time-series signals.
- A demonstration of how model performance can be improved during model development through XAI-informed decision making.
- Publicly available data [18] and code [11] for reproduction.

The use-case serves as a demonstration of achievable improvements. The approach can be extended to further improve a targeted ML model. Following this introduction, this paper is organised in Background, Model improvement,

[†] A time-series is the collection of data points ordered in time, which can be considered as signal sampling.

Implementation, and Outcome, Sections 2 to 5, respectively. After a concise related work in Section 6, concluding remarks are given in Section 7.

2 Background

We discuss the relevance of *execution phases* in industrial CPS to ML models and provide an overview of XAI methods.

2.1 Industrial CPS data structures

Execution phases are segments of time-series monitoring data collected from industrial CPS, reflecting the behaviour of the system under scrutiny per individual task during an execution [19]. As industrial CPS operates primarily in sequential and repetitive machine cycles, the concept of phases is an effective method of data compartmentalisation. As such, data related to each task or sub-task can be analysed individually and in isolation. Phases covering tasks and respective sub-tasks form a hierarchical relation. Compartmentalised and repetitive phase data are most suitable for ML dataset instances.

2.2 Explainable AI

XAI approaches generally fall into two categories: explaining black box models or developing inherently interpretable 'white box' models. While white box models offer transparency, due to the restrictions on their architecture, these often provide lower performance than black box models [1]. Post-hoc and model-agnostic explainability methods can be applied to any predictive model.

SHAP (SHapley Additive exPlanations) [16] is a popular model-agnostic method for explaining ML predictions across diverse data types, such as tabular, image, and time-series data. SHAP determines the contribution of input features to a model's output across all possible feature combinations, known as coalitions. This is done by comparing the model output when a feature is included or excluded across all possible coalitions. Coalitions are constructed by masking, i.e., removing, features which are not included. The removal of features is performed by replacing them with non-informative replacement data.

SHAP's computational cost grows exponentially with the number of input features. Therefore, approximation methods such as KernelSHAP were proposed [16]. However, KernelSHAP does not inherently account for dependencies in time-series data, posing a problem for CPS use-cases, which commonly include time-series data. To overcome this issue, specialised methods such as TimeSHAP [2] and WindowSHAP [17] were introduced to both accelerate SHAP calculations and incorporate temporal dependencies.

C-SHAP [10] offers another approach, whereas TimeSHAP and WindowSHAP determine the contribution of data points and segments, C-SHAP considers the contribution of global concepts. Concepts can be defined as high-level (as opposed to low-level) input features [5]. For example, Sun *et al.* [21] define

image segments as concepts and Kim *et al.* [15] allow any kind of visual observation such as 'stripes' or 'female'. C-SHAP [10] proposes to use components from signal decomposition as concepts for time-series data.

2.3 Signal decomposition

Signal decomposition is a well-established technique, applicable to time-series data, with examples such as: Discrete Wavelet Transform (DWT) [7], Empirical Mode Decomposition (EMD) [9], and Short-Time Fourier Transform (STFT) [6]. While these techniques yield mathematically meaningful components, their practical interpretation can be challenging.

The custom decomposition approach by [10] is intended for human interpretability. Instead of considering how a signal can be mathematically decomposed, the emphasis is on considering which concepts within the data are most relevant in understanding its behaviour. Examples of such concepts include classical signal components such as *Bias* (the mean level), *Trend* (the global change), and *Variance*, representing changes in amplitude over time.

3 Model improvement

3.1 Use-case

We apply the SHAP-based explainability approach to an anomaly detection and identification solution [19], with the aim of improving its performance. For the use-case, time-series traces collected from an experimental platform are processed to identify operational anomalies. This data-centric solution relies on Convolutional Neural Networks (CNNs), trained in a supervised fashion. The considered platform execution conditions (classes) are *Normal*, *NoFan*, and *UnderVolt*, with the last two being anomalous. Electrical metrics such as current, voltage, and energy are monitored, with *current* being the most effective. The data collection and processing steps are given in Figure 1.

Experimental setup The experimental platform consists of the ODROID-XU4[‡] computing device, monitored by a high-frequency power data logger. To achieve highest isolation, the computing device and the power logger are attached to individual Power Supply Units (PSUs). The processor design follows the ARM big.LITTLE architecture. The computational workload, executed per input, is an object detection task, processing images provided through a camera or a storage medium. Power metric readings are collected under the aforementioned execution conditions. The NoFan execution represents operating with a faulty cooling fan, while the UnderVolt describes operating with inconsistent and under-provisioned voltage supply, both affecting the performance and stability of the platform. With different combinations of workload, processor core, and execution condition, 24 scenarios are defined. Experiments are not synthetic, nor simulated.

[‡] `https://www.odroid.nl/odroid-xu4`

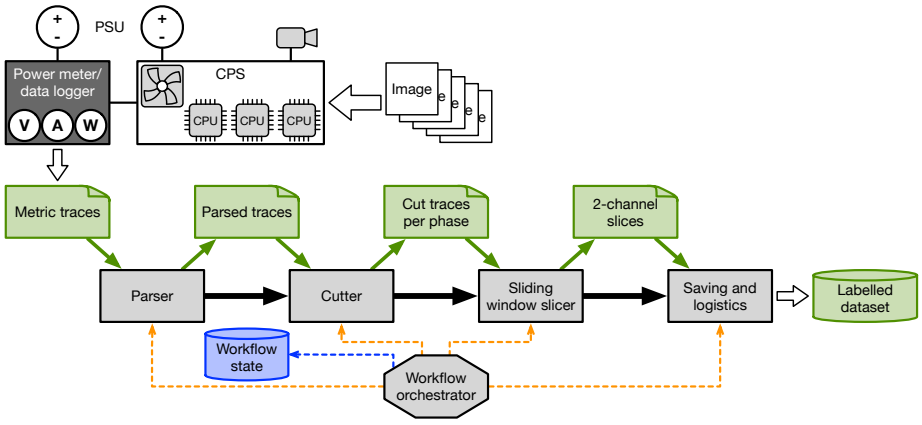


Fig. 1: The experimental setup for machine trace collection and the data processing steps leading to a labelled dataset, suitable for training a CNN model.

ML workflow There are two ML workflows for this use-case, one to prepare a labelled dataset (Figure 1) and another to train a CNN model. The workflow to train the CNN model is rather straightforward. Dataset formation in the former workflow relies on the compartmentalisation of data per execution phases, as explained in Section 2. The raw monitoring data collected in a continuous fashion during any particular execution is parsed and then cut to system-specific phase structures. The phase type intended to be used for training and inference shall be kept. We consider the full processing activity per image, i.e., `cycle-op` phases. As the full image processing phase contains more diverse behaviour, it results in a better demonstration of generating per concept scoring. A windowing algorithm is used to generate slices of data (instances) from compartmentalised phase data, alongside the relevant label.

Dataset The resulting dataset in this case is well-balanced. Machine trace data is collected per scenario followed by a sliding window algorithm. Resulting data instances cover two dimensions, the temporal dimension (timestamps) and one metric dimension, i.e., electrical current values. Brief statistics for variations of this dataset with different sliding window parameters are listed in Table 1. Instance count variation is the result of image processing time differences, i.e., execution time, which in turn is driven by the employed CPU core type.

ML model A CNN model is used for the scoring method. This classifier CNN network takes two data channels as input, the time channel and the metric channel. There are five sequential convolution layers with different output channel sizes of $[64, 64, 128, 128, 256, 256]$, all having kernel sizes of 5. The convolution block is followed by a single fully-connected layer of the size 4096. The predicted class is one of three aforementioned execution conditions.

Table 1: Instance counts per sliding window profile for the *training dataset*.

Statistic	**Profile 100**	**Profile 200**	**Profile 400**
Window size	100	200	400
Window shift	10	10	10
Per scenario instance (Max)	365 136	362 136	356 136
Per scenario instance (Min)	12 749	12 449	11 849
Per scenario instance (Mean)	136 620	134 973	131 678
Total instances	3 278 899	3 239 359	3 160 279

3.2 High-level approach

The C-SHAP workflow for post-hoc explanations consists of three steps: model training, concept construction and calculation of the SHAP values, as depicted in Figure 2. We already covered the model training. What remains is the concept construction using time-series decomposition, as will be discussed next, and the calculation of the SHAP values.

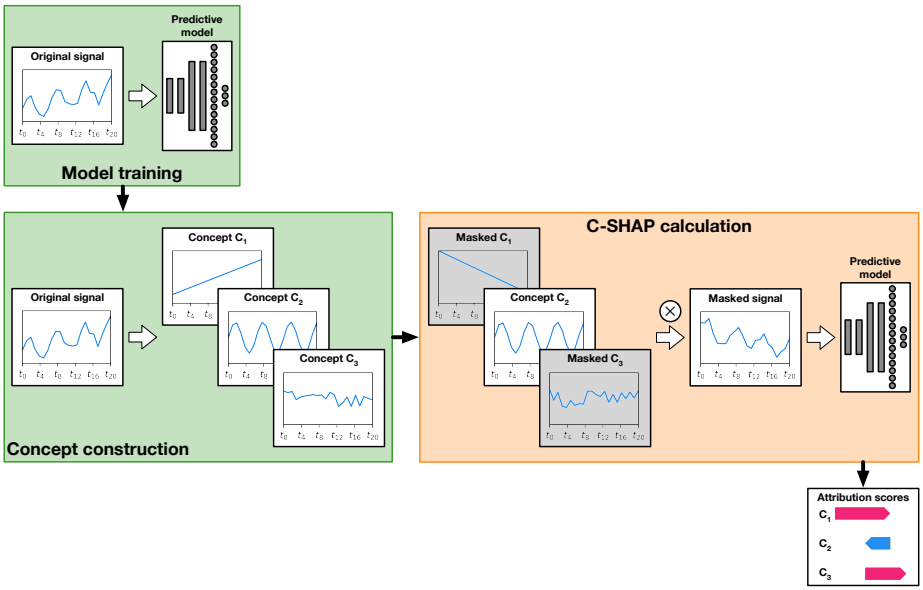


Fig. 2: C-SHAP workflow, including concept construction from signal data, masking and calculation of C-SHAP, i.e., attribution scores.

For the concept construction, time-series decomposition is used. Existing decomposition methods come with limitations, hindering their applicability to our use-case data. For example, EMD cannot represent the discrete jumps present in

our data. Initial experiments with the DWT revealed its capability for dealing with jumps using the Haar wavelet. However, the interpretability of the resulting components is questionable. For C-SHAP to be useful, domain experts need to be able to understand the meaning of considered components. We use a custom decomposition designed specifically for interpretability, as discussed in Section 2. The defined concepts are: 'Levels', 'Peaks', 'Scale', 'Low Frequency (LF)' and 'High Frequency (HF)'. We model the 'Scale' component as multiplicative, while the other components are additive. This results in the decomposition:

$$\mathbf{y}(t) = \mathbf{y}_{\text{Levels}}(t) + \mathbf{y}_{\text{Peaks}}(t) + \mathbf{y}_{\text{Scale}}(t) \times (\mathbf{y}_{\text{LF}}(t) + \mathbf{y}_{\text{HF}}(t)). \tag{1}$$

4 Implementation

4.1 Concept construction

To generate the C-SHAP, i.e., SHAP scores, to determine concept contribution in model prediction, concepts are constructed from the input data. The concept construction uses the decomposition defined in Equation (1). For the implementation of this decomposition, components are extracted from left to right.

First, the 'Levels' component is constructed, modelling the local mean of the signal. We achieve this using Change Point Detection (CPD), which detects changes in the data distribution, yielding segments with similar behaviour. The 'Levels' component is modelled as the mean of the signal on these segments. Change points are detected using the Pruned Exact Linear Time (PELT) algorithm [14]. For the detection, a Radial Basis Function (RBF) cost function is used, with a subsampling of 40 points and a penalty of 50. Residuals resulting from removing the level changes are filtered from the signal. Specifically, points within 20 time-steps of a level change are resampled from a normal distribution with the mean and standard deviation of the preceding segment. The resampled values are smoothened using a moving average with window size 20.

Next, the 'Peaks' component is modelled as statistical outliers, i.e., points with values below the first quantile or above the third quantile. To maintain plausibility of the filtered signal, peaks are not replaced by zero, but replaced by the *median of the signal*. Noise is added to the replacement values with zero mean and a standard deviation equal, matching that of the peak-filtered signal. To maintain the smoothness of the signal, a moving average is applied to the noise before adding it to the signal.

The 'Scale' component is modelled as the maximum absolute value of the remaining signal. The 'Low frequency' component is calculated as the moving average of the signal with a window of 75 points. Finally, the 'High frequency' component represents the residual of the signal.

4.2 C-SHAP calculation

The SHAP values are calculated exactly, rather than using approximation methods such as KernelSHAP [16]. To mask features, i.e., concepts, with uninforma-

tive data they are replaced by concepts sampled from the training data, a common practice in literature. We use training data, rather than zero values, which are unrepresentative of our data distribution (a sample with no high-frequency behaviour is unrealistic) and can lead to out-of-distribution effects inherent in SHAP-based methods.

This training data is selected from scenarios corresponding to the same CPU core type and number of execution rounds. This results in a selection of six scenarios (out of 24). From each scenario we randomly select five cycles. The choice of five was made to capture diversity while limiting computational costs. To allow for the substitution of components, the selected samples need to match the length of the original components. Longer samples are truncated by removing data from the end, shorter samples are padded by repeating the final value.

4.3 Test data

C-SHAP is applied to data instances kept aside during training, to be used for final inference. We select two samples (`cycle-op` phases) from 3 scenarios, each belonging to one of the execution classes, i.e., Normal, NoFan, and UnderVolt. We select the samples from early and late points in the execution timeline to reflect the variations and effects of anomalies. The instances generated by running the windowing algorithm over these samples forms our test data.

5 Outcome

The CNN model trained on windows of size 100 achieved an accuracy of 83.78% on the test data. Figure 3 shows the mean absolute C-SHAP values over all test data. The 'Levels' concept is the most influential on the model's predictions, followed by 'High frequency' and 'Peaks'. Hence, the model mainly relies on the local level of the current, while also considering rapid fluctuations. The scale of the fluctuations and slow changes over time are considered less important.

These results are illustrated in Figures 5a and 5b, visualising two segments of test samples with the SHAP values for the concepts of the associated windows. 'Levels' generally displays higher SHAP values than the other concepts. Importantly, drops in the 'Levels' SHAP values coincide with misclassifications, demonstrating the influence of the 'Levels' concept on these errors.

The histogram of 'Levels' values in the training data (Figure 6a) shows there is a class separation based on level, but also an overlap between classes. Analysis of misclassified samples of the classes Normal and NoFan (Figure 6b) reveals that many errors occur in the overlapping 'Levels' value range. We focus on these two classes, since we found that most errors occur due to the confusion between them. The confusion is specifically apparent between 0.680–0.685, where Normal is prevalent, and 0.685–0.700, where NoFan is prevalent.

We hypothesised that increasing the window size would provide more contextual information and reduce the ambiguity related to the 'Levels' values, since more windows may include multiple levels. Training models with window sizes

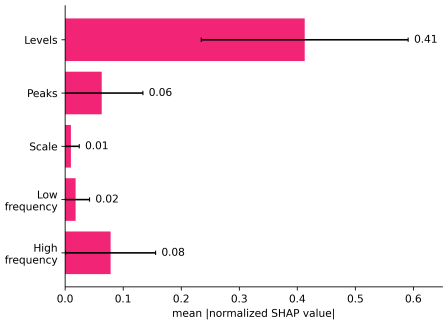


Fig. 3: The mean absolute SHAP values (with respect to the ground truth classes) over the test data with respect to the concepts.

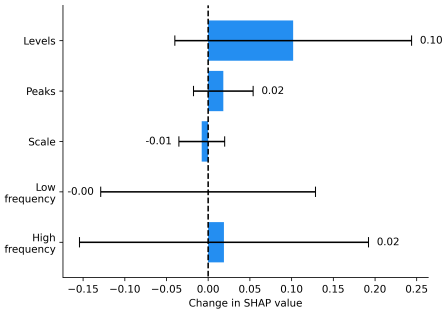


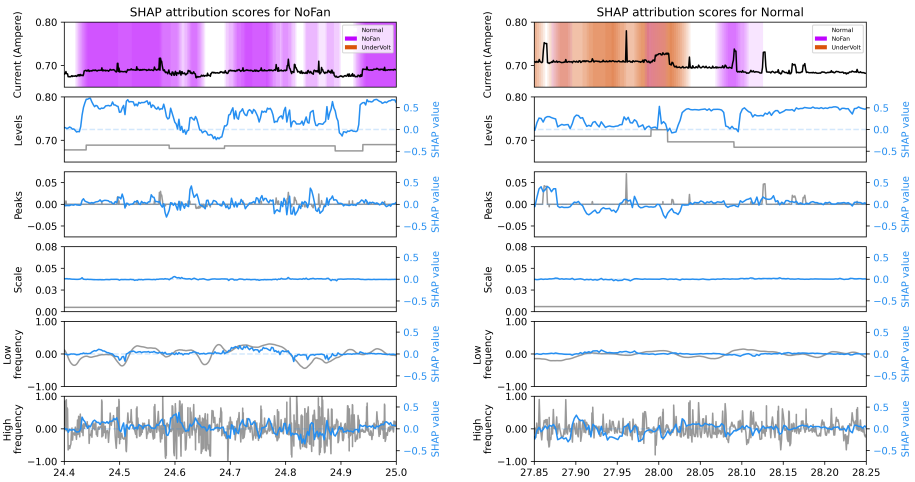
Fig. 4: Change in SHAP values (mean and standard deviation) for the test data when increasing the window size from 100 to 400.

of 200 and 400 data points, improved test accuracies to 87.9% and 92.3% respectively. Comparing SHAP values between the models with window sizes 100 and 400 (Figure 4), illustrates that this improvement is matched by an increase in the SHAP values for the 'Levels' concept, as hypothesized. Figure 5 shows that the new models seem to display more consistent SHAP values for 'Levels'. This increased stability is confirmed by the decrease in standard deviation of the mean absolute SHAP value of the 'Levels' concept from 0.178 for window size 100, to 0.170 for window size 200, and to 0.155 for window size 400.

6 Related work

The use of SHAP values for feature selection has been covered in the literature, e.g., [16,22]. While for traditional models, the effect of high-level features is direct [13], the control over features for deep learning models such as CNNs is generally indirect. These models extract suitable high-level features themselves, which can only be influenced through hyperparameter or data manipulation. Common examples of XAI through the SHAP scoring of high-level features revolve around image analysis use-cases [12,3]. As per context of our use-case, we focus on solutions for industrial CPS and apply SHAP scoring to features extracted from time-series data, which portray complex behaviour.

Within the literature, while claiming application of XAI in solutions intended for CPS domain, shortcomings are apparent. Not every considered use-case can be designated as a CPS in its industrial sense, e.g., a hard disk drive [4]. Such loose categorisations are seen in CPS-focused surveys [8]. The main cause is perhaps the variable definition of the term 'CPS' itself.



(a) Example NoFan, window size 100.

(b) Example Normal, window size 100.

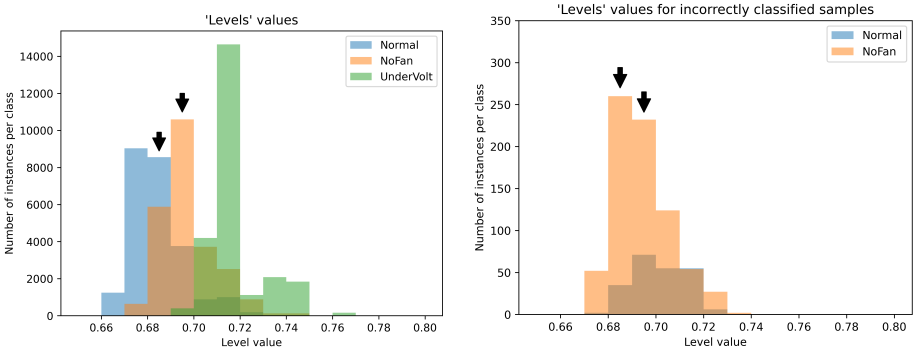
(c) Example NoFan, window size 200.

(d) Example Normal, window size 200.

(e) Example NoFan, window size 400.

(f) Example Normal, window size 400.

Fig. 5: The predictions and SHAP values for two selected example segments from the inference data. a), c) and e) correspond to segments of the same NoFan sample, while b), d) and f) correspond to segments of the same Normal sample. a) and b) correspond to the results for window size 100, while c) and d) correspond to size 200, and e) and f) correspond to 400. In the top rows of the images, the signal is shown along with the predicted classes as coloured overlays, marking the associated windows. For the NoFan sample, the purple windows are correct, and for the Normal sample, the white windows are correct. For each component, the component itself is plotted, combined with the SHAP values for each window. For c)-f) only the 'Levels' component is shown for clarity.



(a) Training data. (b) Misclassified inference data.

Fig. 6: Value histograms for component 'Levels', per class label. The arrows in both figures mark the bins with most misclassifications for NoFan.

7 Conclusion and future work

We described an approach to arrive at effective ML model hyperparameter adjustments, relying on XAI techniques. We argued the advantage over design-space search methodologies, as XAI provides the designer with reasoning behind the adjustment, making the model superior in terms of reliability and dealing with future, unseen data. We have shown how this approach is implemented for time-series data collected from industrial CPS machine operations, by considering custom, human-interpretable signal decompositions. The improvements in prediction performance is demonstrated using our experimental platform. It must be noted that while our method of applying XAI to uncover model reasoning is generalisable, both the model reasoning itself and the potential improvements presented in this paper are dataset and model specific.

Given the component scoring, upon consistent reliance on a particular component, a less complex architecture could be considered, or the opposite could be argued. Plus, depending on component importance, a lower sample rate could be argued in favour of. This means that the monitoring subsystem can be adjusted accordingly, leading to lower data sampling overhead and data size reduction. Additionally, the measures taken based on the XAI findings should be compared with the other techniques, such as automated hyperparameter tuning and alterative XAI approaches, e.g., WindowSHAP [17]. These can be interesting future work avenues to explore.

Acknowledgments. This publication is part of the project ZORRO with project number KICH1.ST02.21.003 of the research programme Key Enabling Technologies (KIC), which is (partly) financed by the Dutch Research Council (NWO).

References

1. Adadi, A., Berrada, M.: Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). IEEE Access (2018). https://doi.org/10.1109/ACCESS.2018.2870052
2. Bento, J.a., Saleiro, P., Cruz, A.F., Figueiredo, M.A., Bizarro, P.: TimeSHAP: Explaining Recurrent Models through Sequence Perturbations. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (2021). https://doi.org/10.1145/3447548.3467166
3. Dardouillet, P., Benoit, A., Amri, E., Bolon, P., Dubucq, D., Credoz, A.: Explainability of Image Semantic Segmentation Through SHAP Values. In: Pattern Recognition, Computer Vision, and Image Processing. ICPR 2022 International Workshops and Challenges (2023). https://doi.org/10.1007/978-3-031-37731-0_19
4. Ferraro, A., Galli, A., Moscato, V., Sperlì, G.: Evaluating eXplainable artificial intelligence tools for hard disk drive predictive maintenance. Artificial Intelligence Review (2023). https://doi.org/10.1007/s10462-022-10354-7
5. Goyal, Y., Feder, A., Shalit, U., Kim, B.: Explaining Classifiers with Causal Concept Effect (CaCE) (2020). https://doi.org/10.48550/arXiv.1907.07165
6. Griffin, D., Lim, J.: Signal estimation from modified short-time Fourier transform. IEEE Transactions on Acoustics, Speech, and Signal Processing (1984). https://doi.org/10.1109/TASSP.1984.1164317
7. Heil, C.E., Walnut, D.F.: Continuous and Discrete Wavelet Transforms. SIAM Review (1989). https://doi.org/10.1137/1031129
8. Hoenig, A., Roy, K., Acquaah, Y.T., Yi, S., Desai, S.S.: Explainable AI for Cyber-Physical Systems: Issues and Challenges. IEEE Access (2024). https://doi.org/10.1109/ACCESS.2024.3395444
9. Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., Yen, N.C., Tung, C.C., Liu, H.H.: The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences (1998). https://doi.org/10.1098/rspa.1998.0193
10. Jutte, A., Ahmed, F., Linssen, J., van Keulen, M.: C-SHAP for time series: An approach to high-level temporal explanations (2025). https://doi.org/10.48550/arXiv.2504.11159
11. Jutte, A., Odyurt, U.: XAI for CPS - Code and Results (apr 2026). https://doi.org/10.5281/zenodo.19257179
12. Kawauchi, H., Fuse, T.: SHAP-Based Interpretable Object Detection Method for Satellite Imagery. Remote Sensing (2022). https://doi.org/10.3390/rs14091970
13. Khan, T., Ahmad, K., Khan, J., Khan, I., Ahmad, N.: An Explainable Regression Framework for Predicting Remaining Useful Life of Machines. In: 2022 27th International Conference on Automation and Computing (ICAC) (2022). https://doi.org/10.1109/ICAC55051.2022.9911162
14. Killick, R., Fearnhead, P., Eckley, I.A.: Optimal Detection of Changepoints With a Linear Computational Cost. Journal of the American Statistical Association (2012). https://doi.org/10.1080/01621459.2012.737745
15. Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., sayres, R.: Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In: Proceedings of the 35th International Conference on Machine Learning (2018)

16. Lundberg, S.M., Lee, S.I.: A Unified Approach to Interpreting Model Predictions. In: Advances in Neural Information Processing Systems (2017)
17. Nayebi, A., Tipirneni, S., Reddy, C.K., Foreman, B., Subbian, V.: WindowSHAP: An efficient framework for explaining time-series classifiers based on Shapley values. Journal of Biomedical Informatics (2023). https://doi.org/10.1016/j.jbi.2023.104438
18. Odyurt, U.: XAI for CPS - Data and Trained Models (Apr 2026). https://doi.org/10.5281/zenodo.19135488
19. Odyurt, U., Roeder, J., Pimentel, A.D., Alonso, I.G., de Laat, C.: Power passports for fault tolerance: Anomaly detection in industrial cps using electrical efb. In: 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS) (2021). https://doi.org/10.1109/ICPS49255.2021.9468262
20. Ribeiro, M.T., Singh, S., Guestrin, C.: "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 (2016). https://doi.org/10.1145/2939672.2939778
21. Sun, A., Ma, P., Yuan, Y., Wang, S.: Explain Any Concept: Segment Anything Meets Concept-Based Explanation. Advances in Neural Information Processing Systems (2023)
22. Wang, H., Liang, Q., Hancock, J.T., Khoshgoftaar, T.M.: Feature selection strategies: a comparative analysis of SHAP-value and importance-based methods. Journal of Big Data (2024). https://doi.org/10.1186/s40537-024-00905-w