Explaining Anomalies in Industrial Multivariate Time-series Data with the help of eXplainable AI

Sarthak Manas Tripathy
Industrial Data Analytics
ABB Corporate Research Center
Ladenburg, Germany

Ashish Chouhan
Big Data and Business Analytics
SRH University
Heidelberg, Germany

Marcel Dix
Industrial Data Analytics
ABB Corporate Research Center
Ladenburg, Germany

Arzam Kotriwala
Industrial Data Analytics
ABB Corporate Research Center
Ladenburg, Germany

Benjamin Klöpper
Industrial Data Analytics
ABB Corporate Research Center
Ladenburg, Germany

Ajinkya Prabhune
Big Data and Business Analytics
SRH University
Heidelberg, Germany

2022 IEEE International Conference on Big Data and Smart Computing (BigComp) | 978-1-6654-2197-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/BIGCOMP54360.2022.00051

***Abstract*—The large amount of data generated by industrial plants provides an excellent opportunity to use Machine Learning (ML) for a better understanding of plant behaviour. Thus, supporting plants operators in running their plants efficiently. An example of an anomaly detection method is to help plant operators detect if their plant is still running normally or if curative actions are needed. Users of ML-based solutions often complain of a challenge called lack of interpretability, i.e., the degree to which one can understand the model's outcome. To address the need for better interpretability of ML models, the research field eXplainable Artificial Intelligence (XAI) has recently received increasing attention in the industrial domain. This paper performs a survey and investigates different XAI techniques that can be applied for detecting anomalies in industrial plant assets. The paper focuses on multivariate time-series data because a) it is the most predominant type of data in industrial systems and b) all the available XAI techniques have applications for various data types such as images, tabular, or textual data. However, these techniques are generally not well suited to explain anomalies in multivariate time-series. To solve this problem, we build an anomaly detection method for the multivariate time-series data generated by the industrial simulators using auto-encoders. Based on feature attribution, examples, and trees, seven different XAI techniques are ideated, developed, and discussed. Out of the seven XAI techniques, a SHAP based explainer, called DTFS, correctly identified the root cause of the anomaly with an accuracy of 86% and took 1.53 seconds to explain our benchmark system.**

***Index Terms*—eXplainable Artificial Intelligence (XAI), industrial process plants, anomaly detection, assets failures, multivariate time-series data**

I. Introduction

Modern industrial process plants such as chemical plants or oil-refineries collect a vast amount of data [1], which provides a great opportunity for Artificial Intelligence (AI) and Machine Learning (ML) [2]. Various possible ML use cases may apply, however, one of the most important use cases is anomaly detection [3]–[6]. Due to advances in automation technology, the trend today for operators is to observe the automation, ensure safety, and "normal" running of the process plant, instead of manually operating the process. Monitoring an industrial process plant is challenging and depends on the operator's experience. Handling a critical situation requires quick analysis and reaction that can be challenging even for experienced operators. Suppose the production is no longer within normal bounds. In that case, the operator needs to analyse the problem and take appropriate actions. Anomaly detection provides this kind of information to the operator, however, for two reasons, anomaly detection is not an entirely satisfying solution for industrial practice. First, the lack of variability in the process manufacturing system provides only training data sets within a narrow operational range, i.e., most situations are labelled as acceptable, and a few rare situations will be labelled anomalous. Second, the anomaly detection methods usually only point the operator to the anomaly. However, due to the complexity of plant equipment, there is often the need for a better explanation to help the operator in the situation analysis [7].

The given use case considered in this paper is a dataset taken from a study by Dix et al. [3] that contains various fault situations that occurred in an industrial separator. Such an asset is typically found in oil production fields, and it separates the fluids coming from the ground well into three output components: oil, gas, and water. Several sensors and actuators such as valves are attached to the asset to control and measure the separation process. These components can have technical issues; for example, a sensor could send biased values, or a valve may be blocked. The resulting high complexity of the asset faults can make manual detection and troubleshooting difficult. Anomaly detection models can help uncover the hidden signs for asset performance issues in the asset signal data, and present any anomalous situation that was round to the user. Here, it is important to present ML outcomes to the plant operator so that the presented information is comprehensible and helps the operator make decisions. However, a challenge of ML is the lack of interpretability of the model predictions

2375-9356/22/$31.00 ©2022 IEEE
DOI 10.1109/BigComp54360.2022.00051

[8], i.e., the degree to which the humans can perceive the model's prediction outcome [9], [10].

The progress in the ML domain has further driven a body of increasingly complex models (ensemble models, Deep Neural Networks (DNN)). Better performance of such complex models comes at the cost of a lack of transparency into their internal behaviour. Thus, limiting the understanding of how the model reaches a particular decision or prediction [11]. Especially in the industrial domain, having high technical complexity, explainable ML outcomes are needed to help the user make decisions. For example, pointing the user to the component of the asset which seems to cause a given anomaly could narrow down the search space for the user in the troubleshooting process.

To address the need for better explainability of ML, the research field eXplainable Artificial Intelligence (XAI) has recently been considered in the industrial domain [7]. The goal of XAI is to produce reasons or details for ML models to make their functioning clear and easy to comprehend [12]. LIME [13] and SHAP [14] are two popular open-source XAI techniques aimed at providing explanations for the predictions made by ML models. The available XAI techniques perform quite well for data types commonly used in ML research, such as image, textual, and tabular data. However, for the industrial case where IoT sensor data (time-series data) is typically the predominant form of data collected, these XAI techniques have their limitations. [15].

This paper, therefore, seeks to contribute to the XAI debate by highlighting the high relevance and need for XAI solutions for time-series data in the industrial domain. The key contribution is to propose and compare seven techniques that try to explain the root cause of the anomalies found in the separator use case by analysing the multivariate sensors of the asset. Note that the paper does not focus on techniques to detect the anomalies; instead, it focuses on explaining them. Techniques to detect the anomalies were addressed by Dix et al. [3]. The remainder of this paper is structured as follows. Section II discusses the related work in XAI. Section III introduces the proposed XAI techniques for explaining the root causes of various given anomalies detected in the time-series data from the industrial separator use case. Section IV presents the evaluation and discussion of the results obtained from the proposed XAI approaches. Section V concludes with the findings and the indication of potential further research.

II. Related Work

Different XAI techniques are discussed in this section, along with a classification of different explanations provided to a user. XAI techniques are classified into white-box or post-hoc techniques and model-specific or mode-agnostic techniques. Suppose an ML model is intrinsically interpretable or explainability is incorporated into the model's architecture. In that case, it is known as a white-box or ante-hoc technique. Whereas if a second model is built to explain the prediction results of the original ML model by analysing the input and outputs, it is termed the post-hoc technique. The ante-hoc or intrinsic interpretable are model-specific, and the post-hoc technique used for specific models is model-agnostic. Unlike model-specific techniques, model-agnostic do not require any access to the internal logic of the ML model.

XAI post-hoc techniques are analysed in this paper for the anomaly detection model. Thus models are not altered, and an explanation technique is built to add to the already well-functioning anomaly detection model. This paper also focuses on model-agnostic XAI techniques that can identify the cause of anomalies in multivariate time-series data irrespective of the anomaly detection model.

Feature attribution methods: These methods provide insights on how important certain features are for the entire model (global explanations) or on the level of individual predictions. SHAP values provide a unified framework for additive feature attributions (feature attribution that considers the interactions between features) with a formal game-theoretic foundation [14]. In their introduction of SHAP values, Lundberg et al. [14] provide one model-agnostic technique (based on the LIME algorithm [13]) and an ANN specific technique with lower computation effort (based on DeepLift [16]).

Explanation by examples: Examples provide a foundation or baseline for humans to interpret or understand any given topic. According to Byrne [17], the generation of example-based text explanations for complex ML models in financial text classification were significantly more understandable to humans. The following are some of the effective example-based explanation XAI techniques [18]:

- *Counterfactual explanations* describe how an instance has to change its prediction significantly. One can learn how the model makes its predictions and explains specific predictions by generating counterfactual instances. Mothilal et al. [19] provide the explanations through counterfactuals or hypothetical examples that explain to users how to obtain a different prediction. The intuition behind this approach is that knowing the reasons for an adverse outcome is not enough. It is also important to know what is needed for a better outcome in the future, considering that the algorithm remains relatively static.
- *Prototypes* are selected representatives or instances from the data, and criticisms are the instances that are not represented by those prototypes. Li et al. [20] proposed a deep learning network with a combination of an auto-encoder for self-supervised learning of relevant features, and a prototype classification layer for determining the sample class of MNIST [21] handwritten digit images. The auto-encoder network independently could be used to reconstruct these images. However, this model is ex-

tended by adding a prototype layer to the encoded latent space. The encoder of the auto-encoder helps to perform comparisons within the latent space. At the same time, the decoder allows visualising the learned prototypes. Thus, it produces the class prototypes and classifies data samples based on the proximity to the prototypes.

- *Influential instances* are the training data points that are the most influential or significant for the parameters of a prediction model or the predictions themselves. Identification and analysis of such influential instances help find problems with the data, debug the model, and better understand the model's behaviour.

Causal mechanism: A causal mechanism presents an explanation in a form similar to a logical expression. A specific example of the causal mechanism used for explanations is decision trees and rules. A decision tree presents a sequence of criteria applied to the data sample that leads to the final output of the decision tree. Anchors [22], on the other hand, is a method that generates post-hoc rule-based explanations and is model-agnostic.

Time-series based XAI: Monitoring of industrial processes deals almost only with signal data (flows, temperatures, pressures, vibrations) that for ML are considered as time-series data. However, only a few XAI techniques have already been applied to time-series data. LIME developed by Ribeiro et al. [13] is applied to time-series classification called Lime-for-Time [23]. LIME used is a modified version of LIME Text Explainer and is applied to explain univariate time-series classification algorithms by highlighting the areas of the time-series data used by the classifier in its prediction. The time-series data is divided into several slices (a subsequence of time-series) of fixed size (like static windows). LIME-for-Time tries to recreate a new dataset by randomly replacing these slices. Another instance of feature attribution technique or SHAP [14] in particular, is used to explain univariate time-series classification [24].

Giurgiu and Schumann [23] highlight that a lot of research is carried out around explainable models for images and text, and little attention is given to explaining models based on temporal data, namely time or event series. It discusses building an interpretable Recurrent Neural Network (RNN) based classifier for the time-series data. The Long short-term memory (LSTM) model has explainability incorporated directly into the classifier, thus being both ante-hoc and model-specific. Another technique of analysing time-series data is to work on them in the image domain. Assaf et al. [25] provided a visual explanation by employing Convolutional Neural Networks (CNNs) as both a forecasting model as well as an explainer model. The success factor for the CNNs is dependent on the type of data they worked upon, i.e., the data used in the paper was having unusual events such as sudden rise or fall, which is missing in the industrial data.

In summary, the related work does not provide a direct solution to the problem of explaining anomalies in multivariate time-series data. At the same time, the XAI techniques for time-series data has provided solutions for univariate and spiky data. In the given case, the time-series data from process plant assets are typically multivariate from multiple asset sensors and do not change instantaneously but quite slowly, because such processes are slow. The existing XAI techniques may provide a reasonable basis for developing solutions for explaining such multivariate time-series data. Thus, this paper presents a survey of different XAI techniques developed from existing XAI techniques for multivariate time-series data.

III. Proposed Pipeline

A. Solution Pipeline Overview

The proposed pipeline, as shown in Fig. 1, comprises anomaly detection and XAI techniques for multivariate time-series data. The input data used in this paper consists of two types of datasets: the first type represents the time-series data of the separator during normal operation, and the second dataset includes failure cases from the same separator that includes valve blockage, valve leaking, or valve dead-band.

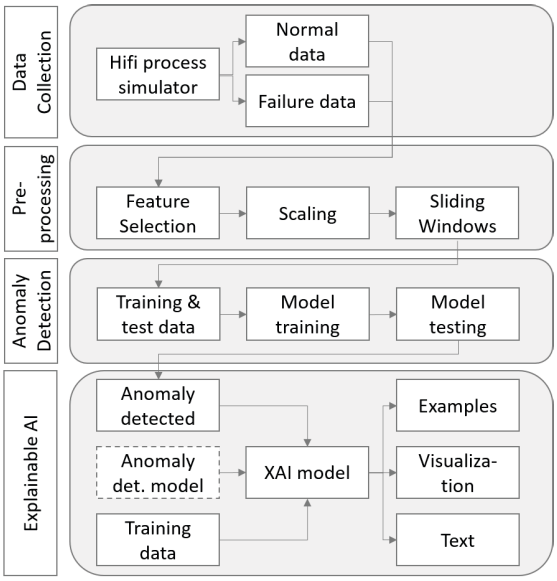

Fig. 1. Overview of the Proposed Pipeline

The datasets first undergo pre-processing, such as resampling the time-series and "windowing" the time-series using a sliding window approach. The dataset representing normal data is then used to train an anomaly detection model based on auto-encoder neural network architecture. The failure data is used to test this model to detect the given failure cases as anomalies. The anomalies detected are then passed through an XAI technique that explains through examples, visualisations, or plain text.

A common challenge in industrial AI is the lack of failure data to train reliable ML models. Industrial systems are typically very robust, so real failure cases are rare. To overcome this limitation, we made use of simulated plant data that was generated using process simulation technology [26]. Using simulated rather than historical data allowed us to simulate various plant equipment failures to evaluate XAI techniques.

Measurements and setpoints are the two types of time-series data that explain anomalies. Measurements are readings from sensors (for example, a level sensor), and a setpoint is a target value that operators can set for different measures (for example, setting the target level). The control system automatically increases or lowers the valve positions until the target value is reached. The dataset considered in this paper consists of nine variables from three different valves: gas (identified by tag number 1037), oil (identified by tag number 1031), and water (identified by tag number 1034) for three different types: setpoint, process value, and valve-opening. The failure scenario simulated is a leak in the oil valve that affects the signal "*20-LV-1031_Z_Y_Value*", i.e., the oil valve-opening.

B. Explaining Anomalies in Industrial Time-Series

After detecting the anomalies in the time-series data for the industrial asset, the next step is to explain why those particular instances are predicted as anomalies and the possible root cause of the anomalies. This paper focuses on analysing and evaluating the XAI techniques for predicting the possible root cause of the anomalies for the industrial multivariate time-series data. The anomaly detection model identifies anomalies in the simulated dataset, and the XAI techniques explain the predicted anomalies. In particular, seven XAI techniques are discussed as follows: (1) DiscreteTimeForLime (DTFL), (2) DiscreteTimeForSHAP (DTFS), (3) Custom Local Explainer (CLE), (4) SimEx, (5) Setpoint Clustering, (6) ProtoGen, (7) Shapelet Identification. All the seven XAI techniques are post-hoc, i.e., techniques are applied after the original ML model has predicted the outcome, and model-agnostic, i.e., techniques do not refer to the internal working of the original ML model.

DiscreteTimeForLime (DTFL): LIME is one of the open-source post-hoc, model-agnostic XAI technique. There are three types of explainers for three kinds of data, i.e., text, image, and tabular. Since there is no application for time-series data, except LIME-for-time, that applies to univariate data. An XAI technique called DiscreteTimeForLIME (DTFL) is developed for the multivariate time-series with the data and model transformations. The inputs to DTFL are the anomalous instance (local), transformed normal windows (training set), and auto-encoder (model-agnostic) trained on transformed normal windows. The LIME tabular is used as an explainer, as shown in Fig. 2, as the transformed windows are in the tabular form.

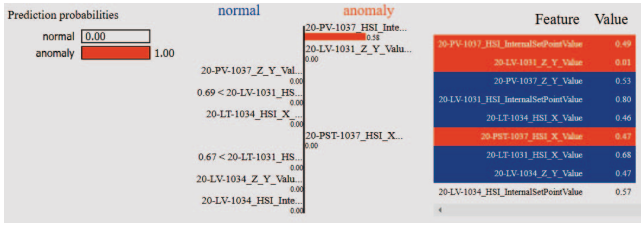

Fig. 2. Explanation using DTFL

LIME creates new samples by perturbating each numerical feature, i.e., sampling from a normal distribution and performing an inverse operation of mean and standard deviation taken from the training data. The result shown in Fig. 2 is for 5000 perturbations. The left part of the explanation shows the prediction outcome of the test instance (anomalous window) that is an apparent anomaly. The middle part shows the features contributing to either of the predictions. A value, i.e., *0.58* is mentioned below for the feature "*20-PV-1037_HSI_InternalSetPointValue*" (gas pressure setpoint). The value indicates that the feature has a 58% contribution to the test data as anomalous from all the perturbations. The right part shows all the features sorted according to their importance to the prediction outcome, and the colour indicates the influenced outcome. The experiment with DTFL is carried out with a different number of samples ranging from 100-10,000. In most of those cases, the failure-causing signal was not having a high value of the contribution to the anomaly.

DiscreteTimeForSHAP (DTFS): SHAP is a game-theoretic XAI technique to explain the output of any ML model. It connects optimal credit allocation with local explanations using the classical Shapley values from game theory [14]. From the available SHAP explainers, KernelExplainer is used to develop DiscreteTimeForSHAP (DTFS). DTFS explains the anomaly in the time-series data as the perturbation-based characteristics, and the Shapley values are needed to identify the signal(s) causing the anomaly. KernelSHAP uses a specially-weighted local linear regression to estimate SHAP values for any model, thus, keeping the explanation model-agnostic.

Fig. 3. Force plot for the anomalous window

The Force plot explanation as shown in Fig. 3 shows three features, each contributing to pushing the model output from the base Shapley value (the average model output over the training dataset has been passed) towards 1.94. The features pushing the class label higher are shown in red. The greater the value of the Shapley value, the more significant is the importance of that feature in the contribution of the class, i.e., anomaly. DTFS, unlike DTFL, helps provide the global

explanation for the model's prediction. The signal "*20-LV-1031_Z_Y_Value*" (oil valve-opening) is found to be having high importance in deciding the anomaly detection to be normal or abnormal. The high Shapley value confirms this, and it has a global explanation of the anomaly detection model. Therefore, it is safe to conclude that DTFS applied to the modified time-series data is correctly isolating the cause of the anomaly.

Custom Local Explainer (CLE): In this approach, a custom local explainer (CLE) is developed based on the idea of LIME and Anchors. The approach is to perturb the data points of the transformed anomalous window for several iterations, and the new perturbed or permuted time-series window is checked with the original anomaly detection model for the prediction outcome. The data points where the prediction changes from anomaly to normal are analysed, and the features contributing to such change are observed. A bar chart for feature importance is prepared to find which feature has the highest contribution in making the anomalous window normal.

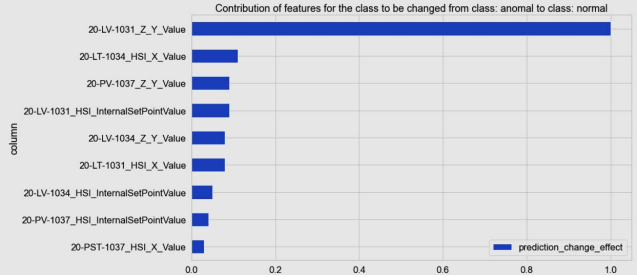

Fig. 4. Custom Local Explainer (CLE): Feature Importance Plot

Fig. 4 shows a bar chart sorted in descending order of feature importances. The order is derived from the permuted time-series windows (variations of the anomalous window) that changed prediction from initially abnormal to normal.

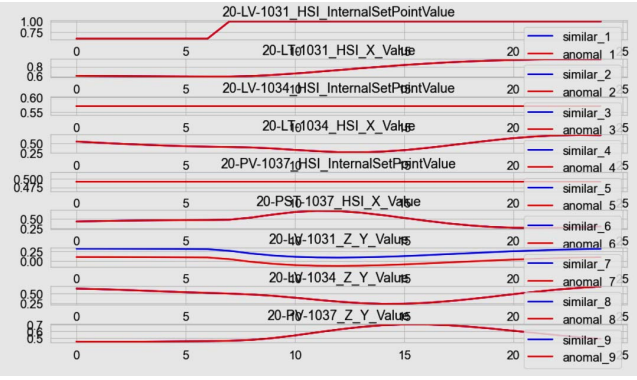

Fig. 5. Perturbed sample resulting in the normal prediction

Fig. 5 shows an example drawn from the permuted data that went back to being normal. The figure compares the signals or features from the anomalous window (red) to the signals from a varied "normal" time-series window. It is observed that the data points for feature "*20-LV-1031_Z_Y_Value*" in the permuted data sample goes over 0.25 higher than the original 0.0. This feature change in the anomalous window is detected as a normal scenario that identifies the anomaly signal and suggests an "alternate" value that results in the window being considered "normal". CLE overcomes the shortcomings of DTFL by identifying the signal that led to the anomaly and providing adversarial examples to support the cause.

SimEx: SimilarityExplainer (SimEx) aims at comparing the anomalous window with all the normal training windows and finding the most similar match for the same. After the match is found, the feature level comparison is made to determine the difference with the similar (normal) window. The comparison is made using the visualisation and the statistics mode of central tendencies, such as the median or mean.

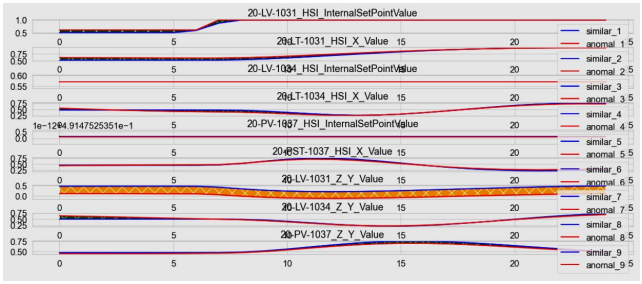

Fig. 6. SimEx: Signal comparison plot

The least similar features are identified as the probable cause of the anomaly. The plot in Fig. 6 is a line chart that compares features of both the anomalous window (in red) and the similar-looking example window (in blue).

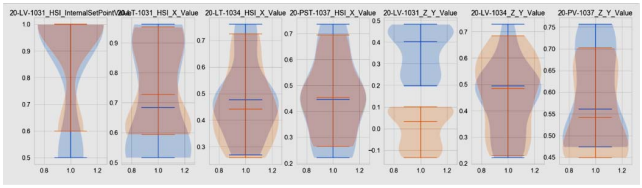

Fig. 7. SimEx: Comparison of median and distribution plot

Fig. 7 uses the violin plot [27] to compare the signals or the features of both the anomalous window and the most similar-looking example window statistically. For the feature "*20-LV-1031_Z_Y_Value*" or the oil valve-opening, the medians and the extremes are far apart, suggesting this is the odd-looking signal of all the signals in the anomalous window. While the explanation provided by SimEx is beneficial to the maintenance engineers in diagnosing the abnormal or failure scenarios, they are still challenging for the plant operators to look into and react accordingly.

SimEx faces a challenge despite solving the problem: as the size of the data used comprised only a few days of plant operation, the search and comparison for the most similar-looking window are faster. However, the training data could comprise several weeks or even months in the real world, and comparing the anomalous time window to every training window would be computationally and time expensive. The explanation result would take several minutes to be computed. This would prove to be ineffective and detrimental for the actual motivation of providing the plant operators quick and efficient explanations for the anomalies. To solve this challenge, two extensions Setpoint Clustering and ProtoGen, are proposed where a subset within training data is identified, and an explanation is then provided using SimEx.

Setpoint Clustering: This approach tries to address the comparison challenge overhead faced in the SimEx by finding a smaller subset with the most similar window. In order to find a smaller subset, the windows are assigned to a cluster using the setpoints. Setpoints are the guiding features the operator defines for processes to work, and these are not prone to failures. However, the process values act accordingly as their values drop or rise concerning the changes in the setpoint value.

Three setpoints are considered in this paper for oil level, water level, and gas pressure. The approach defines a process state by combining all the process setpoints. Suppose ten changes are made to each variable, then the total number of clusters defined is equal to the combinations of plant conditions, i.e., $10^3 = 1000$. Clustering helps identify the cluster that belongs to the anomalous window and then considers that cluster's training data, i.e., the required subset, as an input to the SimEx. The comparison is fast as there would be significantly fewer comparisons.

The artifacts obtained here are similar to SimEx, as SimEx is used to generate explanations. The challenge of finding a smaller subset is overcome in Setpoint Clustering as the size of the training set is reduced by using only the data from one cluster. The time windows comparison gets $\sim$15 times faster (as per the simulation data considered in this paper). Furthermore, the quality of the explanation does not get affected after this enhancement. The causing signal can still be singled out, and the similarity scores are comparable with the scores from the original SimEx results.

ProtoGen: ProtoGen is a challenge-specific enhancement for the SimEx. The intuition behind the ProtoGen is inspired by Li et al. [20], where 15 prototypes were defined for the handwritten digits, and the classification of any new data was only performed with these 15 prototypes. The distance is measured, and the highest similarity to the prototype is selected as the correct prediction. Prototype Generator (ProtoGen) is a low-key implementation that does not involve the auto-encoders latent space.

The comparison search for ProtoGen gets faster than SimEx as the training dataset shrinks by $\sim$ 250 - 300 times and can still identify the causing signal. However, the quality is not optimal. The distribution in the violin plots shows that the prototype does not precisely match the previous approaches. However, in a real-time plant scenario, the operator has to decide or act fast in an anomaly alert. ProtoGen is helpful for the plant operator to decide whether the anomaly is credible or just a hit.

Shapelets Identification: The tree-based explainers are generally used for the ensemble models like the random forest. However, the idea of classifying data based on the information gained is applied in this paper. The intuition is that the above methods have detected the anomalous valve signal. However, what is unique in those signals (a signal window consisting of 25 data points) that classified such signal as an anomaly is not answered. Identifying a few subsequences in a time-series window is interesting in capturing a unique pattern to a particular class (normal or abnormal). These patterns in time-series are Shapelets [28]. The state-of-art Shapelets is the Shapelet Transform (ST) [29]. The transform improves because it separates the shapelet extraction from the classification algorithm. Thus, allowing interpretable classification of time-series with any standard classification algorithm.

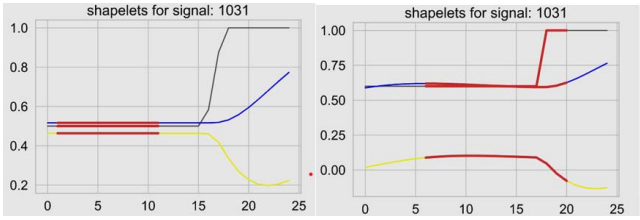

Fig. 8. Shapelets for an normal class (left) and abnormal class (right)

In Fig. 8, the Shapelets are marked in red, and grey, blue and yellow signals are setpoint, the valve opening percentage, and process value for the device 1031, i.e., the oil outlet valve. The left plot in Fig. 8 is an example of a normal time-series window where the Shapelets are flat. In the right plot, it can be seen that the Shapelets mark the transition, like setpoint change. Thus, affecting process value and valve closing percentage. This transition is observed in other anomalous signals that can help the maintenance engineers prevent device failure, identify the anomaly's cause, or save maintenance troubleshooting/diagnostics time. This method cannot be used alone in explaining anomalies (thus, exempted from evaluation). The shapelet transforms in conjunction with SimEx can provide additional information that might explain the anomaly better. This approach is shown as a proof-of-concept, not a complete implementation. The promising results could be extended to make it a better XAI technique for domain experts/plant maintenance engineers.

TABLE I
COMPARISON TABLE FOR XAI TECHNIQUES

Approach	Accuracy (%)	Time taken (sec)	Ease of use	Explanation type
DTFL	23	1.8	Requires data & model transformation	Static visualization of feature attribution
DTFS	86	1.53	Requires data & model transformation	Static & interactive visualization of feature attribution
CLE	82.5	3.6	Requires data & model transformation	Static visualization of feature importances & adversarial example
SimEx	42	6.93	Easily accommodates time-series windows	Static visualization, examples, text
Setpoint Clustering	42	4.51	Easily accommodates time-series windows	Static visualization, examples, text
ProtoGen	47	4.22	Easily accommodates time-series windows	Static visualization, prototypes, text

IV. EVALUATION & DISCUSSION

In this paper, we surveyed seven XAI techniques to explain anomalies in multivariate industrial time-series data. Some techniques worked better as compared to others concerning model accuracy or computation time. We compared all the six techniques (except Shapelet Identification) both quantitatively and qualitatively, and TABLE I summarizes our results. The following evaluation metrics are considered for the evaluation of the XAI techniques:

- *Accuracy*: Accuracy (see equation 1) is calculated for the XAI technique, and not for the anomaly detection model. The higher the accuracy, the better the trust in the explanation of the XAI technique.

$$Accuracy = \frac{\text{Number of times XAI identified the correct cause}}{\text{Total number of anomalies (true positives only)}} * 100 \quad (1)$$

- *Performance (in terms of time)*: This is the time taken by the XAI technique to explain a critical factor to the plant operators. As a result, this evaluation metric guides the user to decide the XAI technique that is best suited to their problem.
- *Ease-of-use or reuse*: The XAI technique should be readily applicable to the multivariate time-series data and should not require much effort in getting the explanation. This is a qualitative criterion.
- *Explanation type*: The type(s) of explanation like static/interactive, visualization-based, example-based, counterfactuals, or text is/are mentioned.

The XAI techniques, when compared based on the accuracy of identifying the actual cause (the anomalous signal or feature), show that perturbation-based approaches like DTFS and CLE have the highest values, i.e., 86% and 82.5%. The accuracy is low for example-based approaches, like 42%, 42%, and 47% for SimEx, Setpoint Clustering, and ProtoGen, respectively. The low accuracy for the example-based approaches can be attributed to the threshold (default value 85) chosen to categorise the anomalous instances, and tuning the threshold might increase the accuracy. When looking at the performance in terms of time taken to explain, DTFS stands out again while the SimEx lags the most. The ease-of-use criterion safely accommodates the example-based explainer better without transforming the data and model. These retain the temporal characteristics of time-series data. The last two comparison criteria, explanation type, and properties have a qualitative impact on providing explanations. All the XAI techniques explain the anomalies visually. The perturbation-based like DTFL and CLE show the static visualisation of feature importances. At the same time, SHAP also has an interactive interface for global explanations. The example-based explainers have the advantage of going further by having comparison plots for both the anomalous instance and the example. Besides, they have an explainer text to transfer the knowledge gained from the comparison plots to non-technical users such as operators.

V. CONCLUSION

This paper aimed to find techniques to explain anomalies in industrial equipment based on their time-series data. To the best of our knowledge, this is the first study exploring XAI techniques for anomaly detection on time-series data. The feature attribution-based approaches stand out in terms of accuracy and performance in isolating the anomaly root causes. In contrast, the example-based approaches consider the temporal characteristics to explain the cause through explainer text and provide a visual and statistical comparison of the anomaly and the example features. As there are insufficient XAI techniques for the time-series data in the state-of-art today, the paper's objective is to highlight the relevance and need for such solutions for the industrial domain. The paper further proposes and analyses seven possible techniques at the example use case of explaining anomalies found in an industrial separator. A limitation of this study is the need to validate the results against additional failure data sets, ideally from other industrial domains. Besides the attempt to provide an explanation that would be useful to the end-users, its efficacy needs to be verified through usability testing with end-users from the same domain. Further research is being carried out to develop Shapelet Identification and find alternate approaches like counterfactuals to explain the predictions better. The current version of Shapelet Identification works in conjunction with SimEx, where additional data is required for SimEx training.

REFERENCES

[1] S. Yin and O. Kaynak, "Big data for modern industry: challenges and trends," *Proceedings of the IEEE*, vol. 103, no. 2, pp. 143–146, 2015.

[2] B. Klöpper, M. Dix, L. Schorer, A. Ampofo, M. Atzmueller, D. Arnu, and R. Klinkenberg, "Defining software architectures for big data enabled operator support systems," in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, pp. 1288–1292, 2016.

[3] M. Dix, A. Chouhan, S. Ganguly, S. Pradhan, D. Saraswat, S. Agrawal, and A. Prabhune, "Anomaly detection in the time-series data of industrial plants using neural network architectures," in *2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)*, pp. 222–228, IEEE, 2021.

[4] R. Borrison, B. Klöpper, M. Chioua, M. Dix, and B. Sprick, "Reusable big data system for industrial data mining-a case study on anomaly detection in chemical plants," in *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 611–622, 2018.

[5] C. Yiakopoulos, K. Gryllias, M. Chioua, M. Hollender, and I. Antoniadis, "An on-line sax and hmm-based anomaly detection and visualization tool for early disturbance discovery in a dynamic industrial process," *Journal of Process Control*, vol. 44, pp. 134–159, 2016.

[6] I. M. Cecílio, J. R. Ottewill, J. Pretlove, and N. F. Thornhill, "Nearest neighbors method for detecting transient disturbances in process and electromechanical systems," *Journal of Process Control*, vol. 24, no. 9, pp. 1382–1393, 2014.

[7] A. Kotriwala, B. Kloepper, M. Dix, G. Gopalakrishnan, D. Ziobro, and A. Potschka, "XAI for operations in the process industry – applications, theses, and research directions," in *Proceedings of the Spring Symposium on Combining Machine Learning and Knowledge Engineering in Practice, AAAI-MAKE*, 2021.

[8] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017.

[9] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial Intelligence*, vol. 267, pp. 1–38, 2019.

[10] B. Kim, R. Khanna, and O. Koyejo, "Examples are not enough, learn to criticize! criticism for interpretability," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 2288–2296, ACM, 2016.

[11] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Communications of the ACM*, vol. 63, no. 1, pp. 68–77, 2019.

[12] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, *et al.*, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[13] M. T. Ribeiro, S. Singh, and C. Guestrin, ""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*, KDD '16, pp. 1135–1144, 2016.

[14] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, pp. 4765–4774, ACM, 2017.

[15] U. Schlegel, H. Arnout, M. El-Assady, D. Oelke, and D. A. Keim, "Towards a rigorous evaluation of XAI methods on time series," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 4197–4201, 2019.

[16] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 3145–3153, PMLR, 2017.

[17] R. M. Byrne, "Counterfactuals in explainable artificial intelligence (XAI): Evidence from human reasoning.," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 6276–6282, 2019.

[18] C. Molnar, *Interpretable machine learning*. Leanpub, 2020.

[19] R. K. Mothilal, A. Sharma, and C. Tan, "Explaining machine learning classifiers through diverse counterfactual explanations," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 607–617, 2020.

[20] O. Li, H. Liu, C. Chen, and C. Rudin, "Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 3530–3537, 2018.

[21] Y. LeCun, C. Cortes, and C. Burges, "MNIST handwritten digit database," 1998.

[22] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," *in Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[23] I. Giurgiu and A. Schumann, "Explainable failure predictions with RNN classifiers based on time series data," in *AAAI-19 Workshop on Network Interpretability for Deep Learning*, 2019.

[24] F. Mujkanovic, V. Doskoč, M. Schirneck, P. Schäfer, and T. Friedrich, "timeXplain - a framework for explaining the predictions of time series classifiers," *arXiv preprint arXiv:2007.07606*, 2020.

[25] R. Assaf and A. Schumann, "Explainable deep neural networks for multivariate time series predictions," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 6488–6490, 7 2019.

[26] M. Dix, B. Kloepper, J. C. Blanchon, and E. Thorud, "A formula for accelerating autonomous anomaly detection," *ABB Review 02/2021*, 2021.

[27] J. L. Hintze and R. D. Nelson, "Violin plots: A box plot-density trace synergism," *The American Statistician*, vol. 52, no. 2, pp. 181–184, 1998.

[28] L. Ye and E. Keogh, "Time series shapelets: a novel technique that allows accurate, interpretable and fast classification," *Data Mining and Knowledge Discovery*, vol. 22, no. 1, pp. 149–182, 2011.

[29] J. Lines, L. M. Davis, J. Hills, and A. Bagnall, "A shapelet transform for time series classification," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 289–297, 2012.