A data-driven approach to fault diagnostics for industrial process plants based on feature extraction and inferential statistics

Simone Smeraldo
and Axel Busboom
Department of Engineering and Management
Munich University of Applied Sciences
Munich, Germany
{firstname.lastname}@hm.edu

Stefan Bendisch
and Hristo Hriskov
and Thomas Morgenstern
Equipment Health Center
Linde plc
Pullach, Germany
{firstname.lastname}@linde.com

Maria Prandini
Department of Electronics,
Information and Bioengineering
Politecnico di Milano
Milan, Italy
maria.prandini@polimi.it

***Abstract*—Accurate detection and diagnostics of faults in complex industrial plants are important for preventing unplanned downtime, optimizing operations and maintenance decisions, minimizing repair time, and optimizing spare part logistics. It is often infeasible to generate accurate physics-based models of complex equipment; therefore, and due to lower computational complexity, data-driven methods are frequently employed. We propose a novel method for data-driven fault diagnostics and validate it using the Tennessee Eastman process (TEP) benchmark. It is assumed that the time of the onset of the fault is known, such that time-series data from the process both before and after occurrence of the fault can be extracted. For each of the measured time-series, several statistical features are extracted. A statistical significance level is computed for each feature using inferential statistics measures. The matrix of significance levels serves as a "fingerprint" of each fault category and is used as input to a feedforward neural network. We show that the network can be trained to achieve high classification accuracy on data from the TEP benchmark model.**

***Keywords*—Fault Diagnosis, Feature Extraction, Predictive Maintenance**

I. Introduction

Complex industrial processes have challenging requirements in terms of their reliability, availability, maintainability, and safety (RAMS). Many companies are in the process of migrating from scheduled maintenance to condition-based or predictive maintenance schemes with the aim of optimizing the tradeoff between maintenance cost and planned downtime on the one hand vs. the cost of unplanned downtime and repair on the other hand [1]. The increasing use of connected Industrial Internet-of-Things (IIoT) devices facilitates the acquisition and communication of real-time data and thus is an important enabler of condition-based and predictive maintenance.

Condition-based maintenance (CBM) refers to measuring physical quantities related to the process or machinery condition in real-time or near real-time, inferring information about the current state-of-health of the system and basing maintenance decisions on the state-of-health. Predictive maintenance goes one step further in that it estimates the remaining useful lifetime (RUL) of the system from its past and current state-of-health. For the RUL estimation, different approaches can be pursued [2]: model-based approaches use first principle models of the physical system and its failure modes; data-based approaches purely rely on measured data and observed failures without using domain knowledge of the underlying system; knowledge-based approaches use explicit representations of human knowledge and experience, e.g., in the form of rules; and hybrid approaches use a combination of methods, such as fitting the parameters of physics-inspired models to measurement data.

Besides detecting and/or predicting that a fault has occurred or may occur, it is an important requirement to also determine the root cause of actual or imminent faults. Accurate fault diagnostics is a basis for operations and maintenance decisions, minimizing repair time and optimizing spare-part logistics. Fault diagnostics is also needed to train classifiers for on-line condition-based or predictive maintenance. A wide range of fault diagnostics approaches has been proposed in the literature, which again may be model-based, data-based, knowledge-base or hybrid (cf., e.g., [3], [4] for a survey).

Due to their complexity and stochasticity, industrial processes are often not amenable to first-principle modeling. Further, decisions need to be taken in or near real-time for a large fleet of asset, such that moderate computational complexity is required. For these reasons, many recently proposed approaches are based on the analysis of historical data as opposed to physics-based models [5].

In complex industrial plants, physical quantities are collected via local instrumentation and typically stored in a global database. Measured quantities (also referred to as "tags") are uniquely identifiable by their name. A time-series of data is acquired and archived for each tag. An actual or imminent fault may manifest itself in a single tag or – more typically

This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy (StMWi) under grant DIK0237/02.

– in a combination of multiple tags, where each measurement taken by itself may be inconspicuous or misleading. The fault may affect the magnitude of the measurement, but also more complex features of the time series, e.g., vibration modes.

Therefore, effective multivariate feature extraction from time-series data is pivotal in data-driven fault diagnostics. Various feature extraction techniques, both supervised and unsupervised, have been presented by Aldirch [6]. The author specifically proposes the use of co-occurrence matrices and local binary patterns as features for process monitoring [6]. In [7] and [8], deep learning architectures are used for automated feature extraction and combined with a classifier to diagnose the faults. In [9], the authors adopt heuristic feature selection and SVM recursive feature elimination to select the best set of features and enhance the classification performance of a Probabilistic Neural Network. He and Xu [10] adopt an adaptive manifold learning method to extract process statistics and project them onto a low-dimensional space, where deviations of the process statistics from the distribution during normal operations indicate an anomaly.

In this paper, we propose a novel, data-driven approach to fault diagnostics in complex industrial plants based on feature extraction and inferential statistics. The approach has the following advantages:

- We make no assumptions about the underlying physics of failures. Rather, we compute a wide range of statistical features of each tag and – for each fault type – generate a matrix of significance levels that highlights the most relevant combinations of tags and their statistical features for each fault type.
- The matrix of significance levels can be easily interpreted. This is essential in practice as failures not represented in the historic training data may need to be manually interpreted by experts. Further, from our experience, diagnostics and recommendation systems are more readily accepted by plant operators and maintenance managers if they provide a comprehensible explanation of the diagnosis provided.
- The matrix of significance levels is irrespective of the specific process equipment monitored. It can therefore be used to generate common, generic features from data from a fleet of assets that are similar in nature, yet distinct, e.g., in terms of their dimensioning. This mitigates the lack of a sufficient number of historical failures at a single asset level, a common challenge in industry (e.g., [11]).

The rest of the paper is structured as follows. Section II describes the proposed algorithm, providing a detailed analysis of each constitutive block, namely the feature extraction process in Section II-A, the inference statistics scores computation in Section II-B and the classification model in Section II-C. We introduce the Tennessee Eastman process (TEP) benchmark model in Section III-A and use it to validate our algorithm in Section III-B. Section IV concludes the paper and provides an outlook to future research.

II. Proposed Algorithm

The algorithm we propose makes no assumptions about the underlying physics of failures or any *a priori* knowledge of the failure characteristics in the tag data. It does require labelled data, though, from historic faults for training, i.e., recordings of time series data before and after fault events along with the respective fault types.

We assume that the time of the onset of a fault is known, such that we can extract time-series data from the process before vs. after the fault occurrence. It is therefore necessary to combine our algorithm with an anomaly detection scheme that can distinguish normal from abnormal operations without having to provide any information about the nature of the fault. For a survey of anomaly detection algorithms, the reader is referred, e.g., to [12]. In most process plants there is a central tag, such as the load or power, which is indicative of the operating point of the plant. We also assume that this "load tag" is known. Typically, most other tags vary together with the load tag as the plant traverses difference operating points. The algorithm scheme is depicted in Fig. 1. As soon as an anomalous condition is detected, data from before and after the detection point is extracted and partitioned into time windows, over which we compute time and frequency features for each of the tags.

To measure how strongly a feature differs between healthy and faulty states, we perform a statistical inferential test on the distributions over the features in normal and anomalous conditions. The outcome of this statistical analysis is a matrix of significance levels that assigns a score to each feature for each tag. A map of scores is considered a "signature", representative of the each type of fault.

In the final stage of the algorithm, the score maps are fed into a classifier for diagnosis of the fault. The classifier is trained using the score maps from historical events and the respective fault classes as labels.

We describe each algorithm block more in detail in the following sections.

A. Feature Extraction

Let us denote by $\hat{t}$ the onset of the fault as detected by the anomaly detection algorithm. If we denote by $X \in \mathbf{R}^{m \times T}$ the set of m measured process variables over the whole time span T of interest, we can partition X as $X = [X_h \mid X_f]$, where X_h and X_f are the measured tags in the portions of T respectively before (healthy) and after (faulty) the fault onset point $\hat{t}$. We move a sliding window over the time-series to extract sets of time sub-sequences X_{hk} and X_{fk} defined as:

$$X_{hk} = \begin{bmatrix} x_1(\hat{t}-kw+1) & x_1(\hat{t}-kw+2) & \dots & x_1(\hat{t}-(k-1)w) \\ x_2(\hat{t}-kw+1) & x_2(\hat{t}-kw+2) & \dots & x_2(\hat{t}-(k-1)w) \\ \vdots & \vdots & \ddots & \vdots \\ x_m(\hat{t}-kw+1) & x_m(\hat{t}-kw+2) & \dots & x_m(\hat{t}-(k-1)w) \end{bmatrix}$$

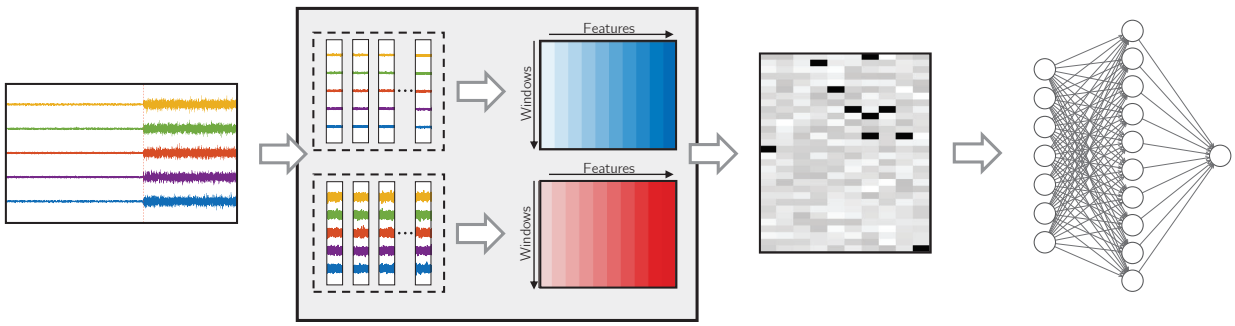

Fig. 1. Proposed algorithm scheme for feature extraction and fault diagnosis.

$X_{fk} =$

$$\begin{bmatrix} x_1(\hat{t}+(k-1)w+1) & x_1(\hat{t}+(k-1)w+2) & \dots & x_1(\hat{t}+kw) \\ x_2(\hat{t}+(k-1)w+1) & x_2(\hat{t}+(k-1)w+2) & \dots & x_2(\hat{t}+kw) \\ \vdots & \vdots & \ddots & \vdots \\ x_m(\hat{t}+(k-1)w+1) & x_m(\hat{t}+(k-1)w+2) & \dots & x_m(\hat{t}+kw) \end{bmatrix}$$

where w denotes the window length and $k = \{1, ..., N : 2Nw < T\}$ is the window index. We do not consider overlapping windows for the sake of simplicity of notation.

For each time-window, the following statistical features are computed:

- mean value
- standard deviation
- peak-to-peak value
- root-mean-square (RMS) value
- skewness of the distribution
- kurtosis of the distribution
- impulse factor: the ratio of the maximum absolute value to the mean value
- crest factor: the ratio of the maximum absolute value to the RMS value (useful to detect early faults evolution)
- clearance factor: the ratio of the maximum absolute value to the RMS value of the square roots of the absolute amplitudes (useful to detect bearing faults)
- high-frequency content: the root-sum-of-squares (RSS) of amplitudes in the frequency domain above a threshold frequency
- correlation with the load tag
- mean slope across the windows as determined by the first and last point

The output of this feature extraction step are two matrices $F_h, F_f \in \mathcal{R}^{N\times 12}$, in which the rows correspond to the different windows indices and the columns corresponds to the 12 features, as depicted in the second block in Fig. 1.

B. Inferential Statistics Scores: Computation and Pre-processing

In this step of the algorithm, for each of the features we perform an inferential test to determine whether the feature differs in a statistically significant way between before and after the onset of the fault, namely between their values in F_h and F_f. To compute such a difference, several metrics exist (cf. [13] for an overview). In our algorithm, we use a two sample t-test, a test statistic that measures the probability that the means of two groups differ from each other, taking into account the variability within the group:

$$t = \frac{\overline{x} - \overline{y}}{\sqrt{\frac{s_x^2}{n_x} + \frac{s_y^2}{n_y}}} \tag{1}$$

where $\overline{x}, \overline{y}$ are the sample means, s_x, s_y are the sample standard deviations and n_x, n_y are the sample sizes.

The resulting scores can be arranged in a matrix, the rows of which correspond to the individual tags from the process and the columns of which correspond to the different features.

The central idea of the proposed algorithm is to consider the resulting matrix of significance level a "fingerprint" of each type of fault. The assumption is that the same type of fault is likely to always affect the same combination of characteristics of the tag time-series while their magnitude may change with different operating conditions or assets in the fleet.

We found that the performance of the algorithm can be improved by suitably normalizing the score matrices to the interval $[0,\ 1]$ before using them for training and classification. The adopted normalization procedure consists in first computing the maximum score in absolute value, say m, within the matrix, then shifting all scores by adding m, and finally dividing the shifted scores by $2m$. By this normalization, we allow the classifier to exploit the information on the relative importance of each statistical feature within the map compared to the other features. This relative weighting appears to yield a more characteristic signature of each fault type, compared to using the absolute scores.

Note that after normalization, a value of 0 corresponds to a strong negative t-score, i.e., a feature that has significantly increased after the onset of the fault. A value of 1 corresponds to a strong positive t-score, i.e., a feature that is lowered by the respective fault.

As a further pre-processing step, we suppress features with a small absolute value of the t-score, i.e., features for which there is no evidence of statistical significance, to the neutral

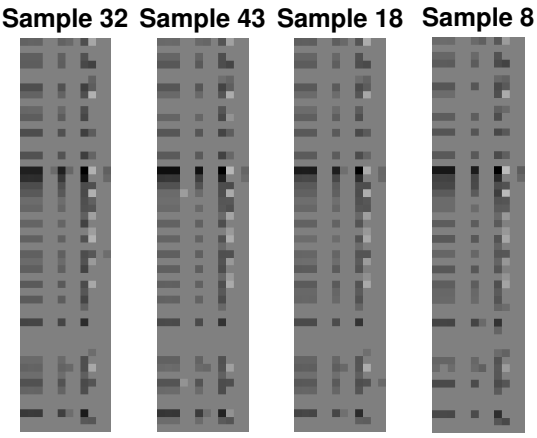

Fig. 2. Score matrices for four samples of simulation runs for fault type #10 in the TEP process validation dataset.

value of 0.5 (after normalization). The suppression is based on the comparison of two healthy datasets to determine a score map for normal process fluctuations. This is based on the premise that in industrial processes, a large base of data is typically available representing normal plant operation. From the obtained set of healthy significance levels, denoting their sample mean as $\overline{h}$ and their standard deviation as s_h, we define an interval $[\overline{h}-3s_h, \overline{h}+3s_h]$. Any scores obtained during fault diagnosis that fall within this interval are set to 0.5 as they are in the same range as scores obtained from healthy data only.

Fig. 2 shows an example of the resulting score matrices of a specific type of fault (fault #10 from the data set in [14]). The matrices are shown for four different samples of simulation runs. They are depicted as grayscale images where a neutral gray pixel corresponds to a non significant feature, a light pixel corresponds to a significant feature that is decreased by the respective fault, and a dark pixel corresponds to a significant feature that is increased by the respective fault.

It can be visually recognized that the four matrices look very similar, supporting the hypothesis that a score matrix can serve as a "fingerprint", specific to each fault type.

C. Classification Model

The labelled score matrices are used to train a classifier, that associates to a score matrix the corresponding fault type. Once trained, the classifier can be used to diagnose new faults during plant operation, by progressively collecting time windows of tags following a fault detection. The number of inputs of the classifier is the number of tags multiplied by the number of statistical features, and the number of outputs equals the number of different fault types.

During validation of our approach against the TEP model data (cf. Section III), we have found that score matrix is so characteristic of each fault type that a very simple classifier can be used. As also shown through numerical results, a feedforward neural network with a single, fully connected hidden layer with 50 neurons and a ReLU activation function was able to achieve remarkable classification accuracy. Note, however, that more complex problems from real industrial plants may require additional statistical features and/or more complex classifier architectures.

III. Results

A. Validation Setup

The so-called "Tennessee Eastman" process (TEP) model was first described by Downs and Vogel [15] in 1993. It consists of components typically found in the chemical industry, such as reactors, separators, mixers and compressors. The TEP model has since become a widely used benchmark for studies on plant-wide control, multivariate control, system identification, fault diagnostics and other problems.

For validating our algorithm, we used an open-source data set generated from the TEP model, comprising 500 simulation runs each for a healthy system state as well as for each type of fault [14]. The data set comprises 52 tags, each recorded at a sampling rate of three minutes. It consists of a training and a test set, the training set covering a period of 25 hours where the fault is introduced after one hour. The test set covers 48 hours, the fault being introduced after eight hours. All faults simulated in the TEP dataset are listed in Table I.

B. Performance Evaluation

1) Offline Classification: We first consider the training of the classifier against historical labelled events, i.e., the set of data $X = [X_h, X_f]$ is immediately available. The workflow is depicted in Fig. 3. We used each simulation as a whole time-sequence, over which the time and frequency features for each of the tags are computed. For each of the faults, we randomly drew (without replacement) 40 healthy and 40 faulty simulation runs from the training data sets, removing the first hour from the latter. We compute the statistical features and t-scores as described in Sections II-A and II-B, using the "*Total Feed Flow*" tag as the "load tag". The result of this step is a score matrix. The sampling and processing scheme was

TABLE I
Description of fault categories in TEP dataset [14]

NO.	Description	Type
1	A/C Feed ratio, B composition constant	Step
2	B composition, A/C ratio constant	Step
3	D feed temperature	Step
4	Reactor cooling water inlet temperature	Step
5	Condenser cooling water inlet temperature	Step
6	A feed loss	Step
7	C header pressure loss	Step
8	A,B,C feed composition	Random variation
9	D feed temperature	Random variation
10	C feed temperature	Random variation
11	Reactor cooling water inlet temperature	Random variation
12	Condenser cooling water inlet temperature	Random variation
13	Reaction kinetics	Slow drift
14	Reactor cooling water valve	Sticking
15	Condenser cooling water valve	Sticking
16-20	Unknown	Unknown

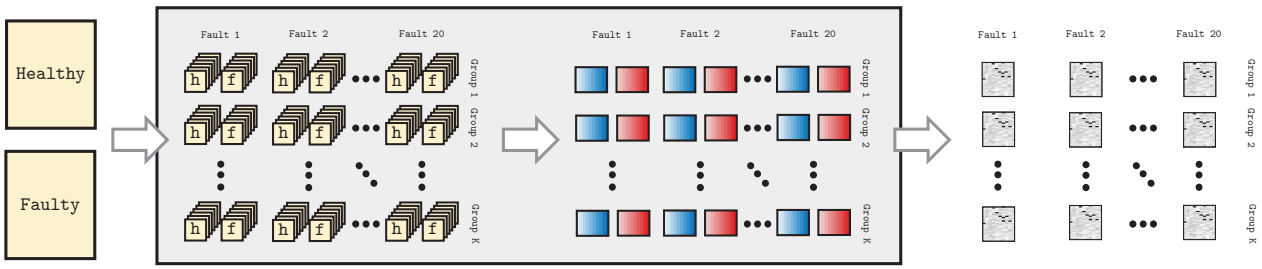

Fig. 3. Workflow for training ($K = 50$) and validation ($K = 25$) scores maps computation from TEP datasets.

repeated 50 times to generate 50 score matrices for each fault type. The resulting score matrices, together with the known fault types, were used to train a classifier, as described in Section II-C.

To validate the classifier, we built a second set of score matrices by repeating the extraction scheme for 25 groups of simulations for each fault over the TEP testing data. We again removed the first eight hours from the faulty simulation runs, and we cut the test simulations to match training length.
To measure the performance of the classification algorithm, we adopt the Precision (P) and Recall (R) metrics, defined as

$$P = \frac{TP}{(TP + FP)}, \tag{2}$$

$$R = \frac{TP}{(TP + FN)}. \tag{3}$$

For each fault class, TP is the count of correct predictions, FP is the count of the target class incorrectly predicted instead of other fault classes, and FN is the count of other faults incorrectly predicted instead of the target fault.

Table II summarizes the performance of the algorithm. The classifier returns outstanding performance, also for faults in the TEP process dataset commonly recognised as "difficult", e.g., fault 9. There was only a single mis-classification (fault 9 instead of 3) in the validation.

It is duly noted, though, that this classification performance relies (a) on the availability of a sufficiently rich database of labelled, historical faults and (b) on a sufficient number of post-fault data windows. The latter point will be discussed in more detail in the subsequent section.

2) Real-time Monitoring: During the actual operations of the diagnostic system, once a fault is detected in the process, windows of faulty data are not initially available, but are progressively collected as time progresses. To simulate this, we modify the last step in Fig. 3: instead of using all test data to compute a score map as described in Section III-A, we progressively compute multiple score maps by including faulty data windows one at a time (from a single one to all 40). In each step, precision and recall are updated using the updated score maps. The evolution of precision and recall for each fault type is shown in Fig. 4. As can be expected, the classifier performance tends to improve as time progresses after the onset of the fault and more time windows can be included in the computation of the score map. Some fault types (e.g., types 10 and 13) reach high precision much faster than high recall, other faults (e.g., type 4) exhibit an opposite behavior and reach high recall faster than high precision. Faults 9 and 15, in particular, need a longer time after the onset of the fault to reach satisfactory levels for both metrics.

IV. Conclusion and Future Work

We have proposed and validated a novel, data-driven approach for fault diagnosis that computes statistical features on multivariate time-series data. The algorithm performs a statistical significance test of healthy vs. faulty data to determine which statistical features of which tags are characteristic for each fault type. Based on an open-source TEP dataset, we have demonstrated that the features can be used to train a classifier to diagnose different fault types with very high accuracy.

The algorithm assumes that historical data of the different fault types is available for training purposes. Once trained, the classifier can be operationalized for diagnosing similar fault events with low computational complexity. The key

TABLE II
Offline performance of the proposed algorithm over the TEP dataset [14] faults for training and validation data. P: Precision; R: Recall.

Fault #	**Training Data**		**Validation Data**	
	P(%)	R(%)	P(%)	R(%)
1	100.0	100.0	100.0	100.0
2	100.0	100.0	100.0	100.0
3	100.0	100.0	100.0	96.2
4	100.0	100.0	100.0	100.0
5	100.0	100.0	100.0	100.0
6	100.0	100.0	100.0	100.0
7	100.0	100.0	100.0	100.0
8	100.0	100.0	100.0	100.0
9	100.0	100.0	96.0	100.0
10	100.0	100.0	100.0	100.0
11	100.0	100.0	100.0	100.0
12	100.0	100.0	100.0	100.0
13	100.0	100.0	100.0	100.0
14	100.0	100.0	100.0	100.0
15	100.0	100.0	100.0	100.0
16	100.0	100.0	100.0	100.0
17	100.0	100.0	100.0	100.0
18	100.0	100.0	100.0	100.0
19	100.0	100.0	100.0	100.0
20	100.0	100.0	100.0	100.0
All	100.0	100.0	99.8	99.8

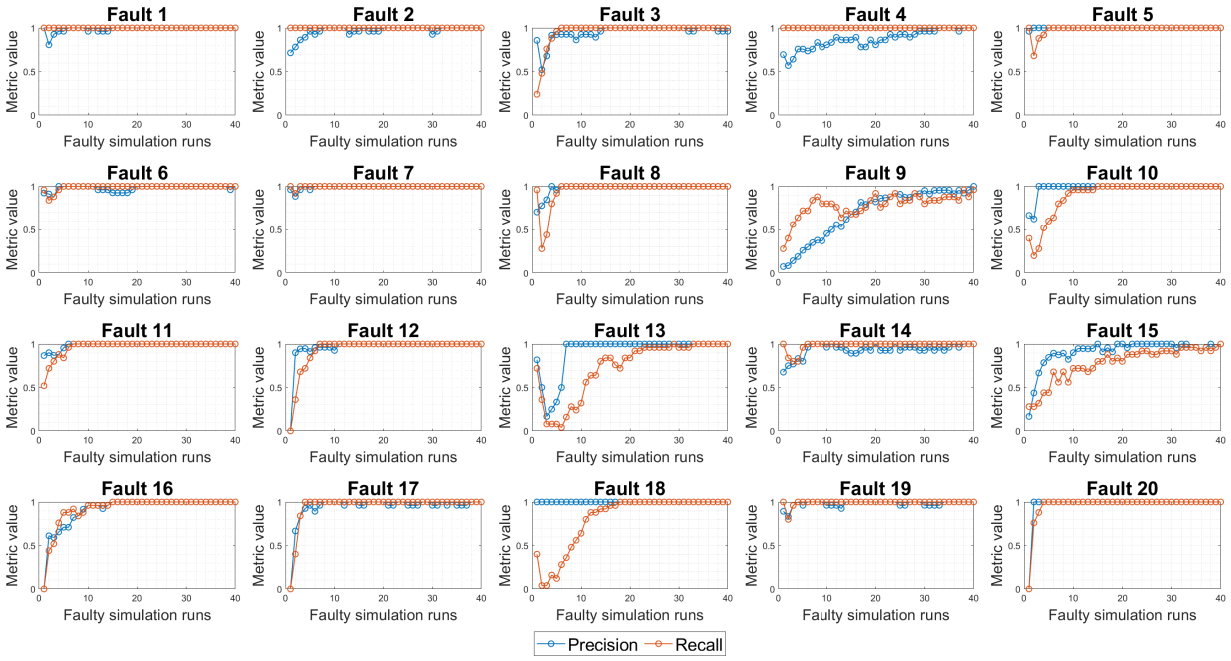

Fig. 4. Precision and recall for real-time classification of different fault types in the TEP dataset [14].

assumption, though, is that the onset of the fault event is known.

Future work will therefore focus on combining our algorithm with suitable anomaly detection schemes that are able to detect the presence and onset of a fault. We will then validate the combined anomaly detection and diagnosis approach against real datasets from Linde plants to confirm the ability to improve the reliability and availability of industrial assets.

REFERENCES

[1] A. Gillespie, "Condition based maintenance: Theory, methodology, & application," in *Reliability and Maintainability Symposium, Tarpon Springs, FL*, 2015.

[2] C. Okoh, R. Roy, J. Mehnen, and L. Redding, "Overview of remaining useful life prediction techniques in through-life engineering services," *Procedia CIRP*, vol. 16, pp. 158–163, 2014.

[3] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques — Part I: Fault diagnosis with model-based and signal-based approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, 2015.

[4] ——, "A survey of fault diagnosis and fault-tolerant techniques — Part II: Fault diagnosis with knowledge-based and hybrid/active approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3768–3774, 2015.

[5] T. Carvalho, F. Soares, R. Vita, R. Francisco, J. Basto, and S. G. Soares Alcalá, "A systematic literature review of machine learning methods applied to predictive maintenance," *Computers and Industrial Engineering*, vol. 137, p. 106024, 2019.

[6] C. Aldrich, "Process fault diagnosis for continuous dynamic systems over multivariate time series," in *Time Series Analysis*, C.-K. Ngan, Ed. Rijeka: IntechOpen, 2019, ch. 1. [Online]. Available: https://doi.org/10.5772/intechopen.85456

[7] S. Chen, J. Yu, and S. Wang, "One-dimensional convolutional neural network-based active feature extraction for fault detection and diagnosis of industrial processes and its understanding via visualization," *ISA Transactions*, vol. 122, 05 2021.

[8] F. Lv, C. Wen, Z. Bao, and M. Liu, "Fault diagnosis based on deep learning," in *2016 American Control Conference (ACC)*, 2016, pp. 6851–6856.

[9] H. Xu, T. Ren, Z. Mo, and X. Yang, "A fault diagnosis model for tennessee eastman processes based on feature selection and probabilistic neural network," *Applied Sciences*, vol. 12, no. 17, p. 8868, 2022.

[10] F. He and J. Xu, "A novel process monitoring and fault detection approach based on statistics locality preserving projections," *Journal of Process Control*, vol. 37, pp. 46–57, 2016.

[11] S. Sridhar and S. Sanagavarapu, "Handling data imbalance in predictive maintenance for machines using SMOTE-based oversampling," in *2021 13th International Conference on Computational Intelligence and Communication Networks (CICN)*. IEEE, 2021, pp. 44–49.

[12] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[13] S. Theodoridis and K. Koutroumbas, "Chapter 5 - feature selection," in *Pattern Recognition (Fourth Edition)*, 4th ed., S. Theodoridis and K. Koutroumbas, Eds. Boston: Academic Press, 2009, pp. 261–322. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9781597492720500074

[14] C. A. Rieth, B. D. Amsel, R. Tran, and M. B. Cook, "Additional Tennessee Eastman Process simulation data for anomaly detection evaluation," Harvard Dataverse, 2017. [Online]. Available: https://doi.org/10.7910/DVN/6C3JR1

[15] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Computers & chemical engineering*, vol. 17, no. 3, pp. 245–255, 1993.