

# Event-based predictive maintenance on top of sensor data in a real Industry 4.0 case study

Athanasios Naskos<sup>1</sup>, Georgia Kougka<sup>1,2</sup>, Theodoros Toliopoulos<sup>1,2</sup>,  
Anastasios Gounaris<sup>2</sup>, Cosmas Vamvalis<sup>1,3</sup>, and Daniel Caljouw<sup>4</sup>

<sup>1</sup> Atlantis Engineering, Thessaloniki, Greece  
{naskos,kougka,vamvalis}@abe.gr

<sup>2</sup> Aristotle University of Thessaloniki, Greece  
{georkoug,tatoliop,gounaria}@csd.auth.gr

<sup>3</sup> European Federation of National Maintenance Societies  
chairman@efnms.org

<sup>4</sup> Philips Consumer Lifestyle - The Netherlands  
Daniel.Caljouw@philips.com

**Abstract.** We deal with a real predictive maintenance case study encountered in modern Industry 4.0 settings: based on logs of past failures, we train a model to predict critical failures of equipment without any other domain expert knowledge well in advance. We propose a novel methodology that combines and extends the state-of-the-art in event-based predictions with advanced time-series analytics. This renders our technique applicable directly onto the sensor data, as it is produced in a modern factory setting. Further, we show that employing unsupervised learning techniques, such as continuous outlier monitoring, is a competitive approach. Although our techniques are developed and tested in a specific case study, they can be transferred to arbitrary settings.

## 1 Introduction

One of the key characteristics of the 4th Industrial Evolution, broadly known as Industry 4.0, is the wide-spread of Predictive Maintenance (PdM), which is reported to be capable of yielding “*tremendous*” benefits.<sup>5</sup> The goal of PdM is to eliminate machinery downtime and operational costs. Broadly, PdM capitalizes on the benefits stemming from mature techniques in the fields of data analytics, machine learning and big data management in distributed and IoT/edge-computing settings[11].

In this work, we aim to devise PdM techniques, when there is no prior domain expert knowledge, e.g., rules or models that can be used to predict events. Event-based PdM approaches train models with logged events to recognize patterns which precede a failure incident (called *target events*). We explore both supervised and unsupervised learning techniques. For the former, a novel technique for timeseries discretization, which is based on [16], is utilized in order to

---

<sup>5</sup> <https://www.pwc.nl/nl/assets/documents/pwc-predictive-maintenance-beyond-the-hype-40.pdf>

leverage event-based prediction approaches, especially in cases where the provided maintenance logs are not sufficient to be used for predictions (which is a commonplace situation). In this way, the timeseries is transformed to a sequence of *artificial* events, which may not have meaning in the physical world but are shown capable of capturing preceding hidden patterns before failures. Then, an event-based PdM approach borrowed from the field of aviation industry is adapted to process such *artificial* events and to feed a regression analysis model that predicts prominent machinery failures.

In addition, we apply anomaly detection on streaming data [1] to detect early signs of failure. Anomaly detection naturally lends itself to failure detection in Industry 4.0; here we provide concrete evidence that it can also be used for prediction per se and thus become part of an advanced PdM solution. The key strength of this approach is that it does not rely on model training.

The approaches are tested in a real case study of a cold forming press in the Philips Consumer Lifestyle plant in the Netherlands, but we ensure that the approach is applicable to arbitrary settings, provided that sensor measurements are available and, for the supervised learning approach, there exists information about the time a failure occurred. We have provided the implementation of the core parts of both the supervised and unsupervised learning techniques as open source<sup>6</sup>, so that third parties can easily adapt our solution to their problems. In summary, our contribution is threefold:

1. We present a novel methodology that is directly applicable to sensor measurements. Our key rationale is to transform timeseries to a series of events, which allow for state-of-the-art event-based PdM techniques to apply.
2. We present how streaming outlier detection can be leveraged to predict critical equipment failures.
3. We evaluate our solutions to a real setting, and the results are particularly encouraging. The supervised learning solution managed to achieve 61% precision and 61% recall (0.61 F1-score) when predicting 1-8 hours ahead. The unsupervised learning one managed to achieve different but equally interesting trade-offs. The performance is further improved with combinations of predictors, e.g., exhibiting 59% recall and 82% precision (0.67 F1-score).

*Paper structure:* the next section presents the case study we consider. In Section 3, we present our novel methodology, which is based on combination of time-series and event-based PdM techniques. Section 4 discusses the application of outlier detection directly on streaming data for PdM purposes. In Section 5, the obtained results are discussed. We conclude with the related work and the final remarks in Sections 6 and 7, respectively.

## 2 The Case Study

*High-level description.* The focus of the current use case is the production line of a Philips factory and more specifically, the cold forming press that is a part of

<sup>6</sup> <http://interlab.csd.auth.gr/anaskos/pdm-solution>

the line. During a production run the input material passes through many form-shaping and quality check stages to reach its final form. The cold forming press is one such form-shaping stage and a complex and expensive piece of equipment, where a metal strip goes in and cold formed products come out. The press contains various modules that cut, bend and flatten the input strip. The main problem of the specific machinery is that it is acting as a black box, which means that there is no option to monitor its status or the status of each individual module that is contained within it, without completely stopping the machine. The modules within the press are arranged in a specific order to produce the correct output shape. The different modules form a pipeline, where the output of one module upstream forms the input of the next module downstream. The metal strip that comes into the press passes through all of the modules in the specified order with its shape constantly changing. The metal strip comes from an input reel and passes through the first quality check. The strip then enters the press and is processed by each of the six modules. After the strip takes its final form, it exits the press and passes a second quality check.

The modules are subject to breakage induced by age and other reasons, which cascade from one module to the next in line and so forth. Hence, planned preventive maintenance checks are applied removing the whole press from the production line. The cost for stopping the production run and especially the cost for repairing the faulty parts of the machinery can be very high for the company. This means that there is a need for some type of prediction or early detection of failures in the cold forming press.

*Failure Types and Sensors Available.* There are multiple data sources in place. The first one considers the profile of the raw material (i.e. metal strip), such as part number, while the second one considers the thickness and the temperature of the metal strip entering the press. The quality check at the end product of the press contain measurements on the thickness, the shape and other proprietary quality measurements. The final data source comes from the press itself. To monitor the opaqueness of the cold forming process, an acoustic emission sensor is placed in the press; overall, the different acoustic sensors are comprise 6 channels, and at each step, a measurement for each angle and channel is monitored (the dimensionality of each acoustic channel is at the orders of several hundreds). The sensor detects and emits the acoustic waves that radiate from the material when it is processed by the press. The difference on the emission can detect possible faults on the parts of the press that cannot be detected through other means. In the following sections, acoustic emission measurements are used to predict and/or detect prominent machine stops due to machinery failures.

### 3 A supervised learning technique based on a timeseries discretization methodology

Our rationale is summarized as follows. We take for granted that event-based machine learning for PdM is a rather mature field, in the sense that PdM techniques tailored to Industry 4.0 setting have been developed, e.g., [12,8,14]. The

key characteristics of a PdM task compared to traditional classification is that events are very rare and the feature set very sparse. Then, the main challenge is to fill the gap from the initial measurements from sensors to event generation. Our key novelty is that, instead of classifying timeseries as a discrete set of events, we map timeseries to a sequence of *artificial* events thus placing no burden to engineers to annotate the sensor measurements. We only require information about the time of several critical failures in order to train our methods.

The key part is the discretization of the timeseries to a series of events. To this end, we employ the *Matrix Profile (MP)*, which is a data structure that annotates a time-series [16]<sup>7</sup>.

### 3.1 Data pre-processing and MP-based Timeseries Analytics

For the purpose of the Philips’s case study, we focus on subsequences of a pre-defined pattern-length of size  $PL$ . The subsequences’s length corresponds to individual acoustic measurements. Acoustic measurements are processed per channel after taking the maximum value among all angles for each time point; therefore, all acoustic channels are transformed to 1-dimensional series. For a timeseries  $T$  of length  $n$ , we estimate  $MP$ , which is a vector of length  $n - PL + 1$ .  $MP(i)$  denotes the distance of the sub-sequence starting at the  $i^{th}$  position in  $T$  to its nearest neighbor. Any distance metric can be used, but as explained in [16], the default option is the *z-normalized Euclidean Distance*. The  $MP$  vector is accompanied by the *Matrix Profile Index (MPI)*, which is of same size as  $MP$ .  $MPI(i)$  keeps the pointer to the position of the closest neighbor of the sub-sequence of length  $PL$  starting at  $T(i)$ . The lower the values in the  $MP$ , the higher the similarity of the  $PL$ -size pattern beginning at the corresponding point to its closest neighbor.

### 3.2 The MP-based algorithm to extract artificial events

We introduce an algorithm that is based on the estimation of  $MP$  in order to create the artificial, yet significant, events through similarity estimates. Then, on top of this, a complete technique for extracting hidden patterns to predicting or early detect failures is developed. The technique for artificial event extraction is summarized in Algorithm 1.

The proposed technique does not require any logs apart from raw measurements, not even information about past failures. Given the pattern-length parameter ( $PL$ ), we apply the  $MP$  methodology in order to compute the  $MP$  based on subsequences of length  $PL$  and thus also generate the  $MPI$ .  $MPI$  is essentially a directed graph, where each edge points to the most similar subsequence. We consider the  $MPI$  as a graph  $G = (V, E)$ , where the  $V = v_1, \dots, v_n$  denotes a set of nodes and  $E = e_1, \dots, e_z$  defines the edges of the graph  $G$  weighted by the values in  $MP$ . Some edges have either globally or locally very high weights.

<sup>7</sup> Open source implementations are provided from <https://www.cs.ucr.edu/~eamonn/MatrixProfile.html>

---

**Algorithm 1** Mapping timeseries values to artificial events

---

**Require:** A time-series of press incidents  $T = t_1, \dots, t_n$ , the pattern-length  $PL$ .

- 1:  $MP, MPI \leftarrow \text{computeMatrixProfile}(T, pl)$
- 2:  $X \leftarrow 10$  //default value for the  $X$  threshold
- 3:  $Y \leftarrow 2$  //default value for the  $Y$  threshold
- 4:  $Z \leftarrow 2$  //default value for the  $Z$  threshold
- 5: **for**  $i = 1$  to  $n - PL + 1$  **do**
- 6:   **if**  $MP(i) > X * MP(MPI(i))$  **then**
- 7:     Remove the corresponding edge from  $MPI$
- 8:   **end if**
- 9:   **if**  $MP(i) > Y * \text{mean}(MP)$  **then**
- 10:     Remove the corresponding edge from  $MPI$
- 11:   **end if**
- 12: **end for**
- 13:  $conComp \leftarrow$  Calculate the weakly connected components of  $MPI$
- 14: Filter the distinct components with less than  $Z$  members
- 15: **for**  $i = 1$  to  $n - PL + 1$  **do**
- 16:   **if**  $MPI(i)$  belongs to a connected component **then**
- 17:     map  $T(i)$  to the connected component id
- 18:   **end if**
- 19: **end for**

---

Therefore, we apply a set of thresholds in order to eliminate the nodes that probably are noise and are connected to other nodes with low similarity. More specifically, we filter out the edges of the graph that connect two nodes when their distance is  $X$  times greater than the distance of the edge connecting the sink node to its nearest neighbor (local rule). A global rule is that we prune all edges with a weight more than  $Y$  times the mean  $MP$  value. As a following step, we estimate the weakly connected components (sub-graphs) of the  $MPI$  graph and map each such component to a distinct artificial event, i.e., in this step we disregard edge directions. We prune small components with less than  $Z$  members. Finally, every point of the timeseries that is part of a connected component is labeled by the id of that component. In Algorithm 1, we provide default values for the three thresholds employed, based on our experiments in the real dataset (we omit sensitivity analysis due to space constraints).

**Variants.** Similarly to the version of the algorithm that we described above, we have also implemented and tested a community detection algorithm, proposed in [4], for finding communities of graph nodes instead of connected components. Then all the members of a detected community are labeled with the community id.

### 3.3 Supervised event-based PdM approach using artificial events

Utilizing the events generation technique presented in Section 3.2, we are able to use any data-driven event-based prediction approach to tackle the PdM problem.

In this work, we showcase the usage of a PdM approach applied on aviation data [8], adapted to the Philips use case. Adapting the selected PdM approach to the press use case, the *artificial* events obtained by the MP-based algorithm are subject to intensive preprocessing in order to expose the patterns of machine failure and then leverage such patterns to train a model to predict prominent machine failures. The proposed approach penalizes both rare and frequent events (implicitly performing feature selection) and amplifies the strength of the events closer to machine failure incidents, applying a Multi-Instance Learning (MIL) technique to over-sample the aforementioned events. Such preprocessed log data form the training set, which is then fed into a regression analysis algorithm for the prediction of the machine failures. Next, we further elaborate on this approach as a key representative of the state-of-the-art.

### 3.4 Event-based PdM solution details

The artificial events are mapped to actual timestamps based on the origin time-series and partitioned in ranges defined by the occurrences of the fault that PdM targets. These ranges are further partitioned into time segments, the size of which (i.e. minutes, hours, days) correspond to the time granularity of the analysis. In the press use case, hourly segments are used, based on the knowledge acquired from the maintenance engineers. More specifically, engineers want to be warned at least one hour and at most several hours before the occurrence of a machine failure. The rationale behind the time segmentation is that the segments that are closer to the end of the range may contain fault events that are potentially indicative of the main event. The goal is to learn a function that quantifies the risk of the targeted failure occurring in the near future, given the events that precede it. Hence, a sigmoid function is proposed, which maps higher values to the segments that are closer to a machine failure. The steepness and shift of the sigmoid function are configured to better map the expectation of the time before the failure at which correlated events will start occurring. The segmented data in combination with the risk quantification values are fed into a Random Forests algorithm as a training set to form a regression problem.

In practice, the event types are hundreds if not thousands. Each event type is essentially a dimension. Therefore, to increase the effectiveness of the approach standard preprocessing techniques can be applied: (i) Multiple occurrences (*MO*) of the same event in the same segment can either be noise or may not provide useful information. Hence, multiple occurrences can be collapsed into a single one. (ii) Standard feature selection (*FS*) techniques (like [5]) can also be used in order to further reduce the dimensionality of the data.

Finally, to deal with the imbalance of the labels (given that the fault events are rare) and as several events appear shortly before the occurrence of the fault events, but only a small subset of them is related to them, Multiple Instance Learning (*MIL*) can be used for bagging the events and automatically detecting the events that can act as predictors. A single bag contains events of a single hour. Also, the data closer to the fault events (according to a specified threshold) are over-sampled, so that training is improved.

### 3.5 Experiments and results

The experiments were done using historical timeseries and converting them to a continuous stream. The ground truth used for the measurements is the information of the timestamps that the machine stopped working due to technical reasons, e.g. damage of a module on the press; this information has been provided by the engineers responsible for the examined machinery. Each machine stop represents a failure mode, each prediction represents an alarm and the detected stops are the ones that have at least one preceding alarm within a fixed period before the fault. We assess the efficiency of the technique using the *recall* and *precision* metrics adapted to the PdM context, measured according to the following definitions: *precision* is the ratio of the successfully predicted stops to the number of total alarms, and *recall* is the ratio of the predicted stops to the number of total stops, where a stop is considered as successfully predicted if there is any prediction made in a specified time gap before a machine stop. Multiple alarms inside the specified time gap for the same machine stop are counted as a single alarm, while the false alarms (i.e. before the time gap) are counted individually. The rationale is that the maintenance engineers are prompted to respond to the first alarm for a specific machine stop, while in the case of the false alarms, they are called to respond to every one of them.

The data used for the assessment of the supervised learning approach are the acoustic emission measurements. The acoustic emission sensors are placed in 6 different spatial positions on the cold forming press, generating data in 6 distinct channels providing measurements in hundreds of different angles per channel. We perform dimensionality reduction through maintaining only the maximum value across all angles per channel per time point. I.e., finally, for each timestamp, we consider a single measurement per acoustic channel.

The experiments share a common parametrization and fine-tuning is beyond the scope of this work (due to space limitations). Three values of pattern-length (*PL*) for the Matrix Profile are used (i.e. 5, 10 and 50). The number of distinct artificial events are depicted in Table 1. *MO* is enabled in all the experiments and over-sampling is applied. The steepness and the shift parameters of the sigmoid function are set to 0.8 and 4, respectively; the threshold for the value of the sigmoid function to set an alarm is set to 0.3, while the time gap for true alarm consideration is set between 1 and 8 hours before a machine stop incident. All measurements refer to 10-fold cross validation. As there are lots of event types, *FS* preprocessing step is also tested. For partitioning the dataset into 10 folds, we use the number of incidents and not the number of time segments.

Table 2 presents the recall and precision values, of the results that achieved the best F1-score per channel. The second column depicts the pattern-length used in the Matrix Profile algorithm, while the third one indicates the usage of the *FS* preprocessing step. The Table also presents the results in two of the channels (i.e. 1st and 2nd) where the community detection (*CD*) algorithm is used in place of the connected component (*CC*) algorithm utilized in the MP-based *artificial* event generation approach. As we observe, Channel 1 and Channel 4 achieved the highest F1-score (0.61 and 0.59 resp.). There is no clear

Source	PL-5	PL-10	PL-50
Channel 1	6411	5615	1794
Channel 2	6401	5800	5234
Channel 3	6374	5722	3591
Channel 4	6334	5148	1305
Channel 5	6353	5837	3445
Channel 6	6396	5724	4216

**Table 1.** Number of distinct artificial event types generated per pattern length (PL).

Source	PL	FS	Recall	Precision	F1-score
Channel 1	5	x	0.61	0.61	0.61
Channel 1 <i>CD</i>	5	x	0.57	0.63	0.59
Channel 2	10		0.63	0.53	0.55
Channel 2 <i>CD</i>	5		0.49	0.45	0.47
Channel 3	10	x	0.55	0.61	0.57
Channel 4	50		0.49	0.76	0.59
Channel 5	50	x	0.45	0.80	0.54
Channel 6	50	x	0.48	0.68	0.50

**Table 2.** Experimental results on all the acoustic channels using the supervised learning technique (CD: community detection).

Source	Strategy	Recall	Precision	F1-score
Ch.4 - Ch.1	<i>AND</i>	0.62	0.5	0.55
Ch.1 <i>CD</i> - Ch.1	<i>OR</i>	0.59	0.82	0.67

**Table 3.** Experimental results on all the acoustic channels using an ensemble of the supervised learning technique.

winner between the different pattern-lengths and whether feature selection has been applied or not. Regarding the application of the *CD*, the results are inferior to those achieved by *CC*, despite the fact that the number of the generated artificial event types is almost the same in both the cases.

Next, we employ two simple ensemble strategies with two predictors each: the *AND* strategy, where two predictors need to raise an alarm, and *OR* strategy, where an alarm is raised whenever at least one of the predictors votes for it. We have computed the precision, recall and F1-score of all the possible pairs between all the previous experiments. The results with the highest F1-score per strategy are shown in Table 3. As we observe, the *OR* strategy was able to enhance the previous results, achieving 0.67 F1-score combining two cases with low recall but high precision. Note that in this scenario, a random predictor achieved F1-score of 0.31; moreover, a dummy predictor with recall 1 through raising an alarm every 7 hours cannot exceed F1-score of 0.58.

Source	Recall	Precision	F1 score
Channel 1	0.57	0.41	0.48
Channel 2	0.83	0.48	0.61
Channel 3	0.74	0.49	0.59
Channel 4	0.38	0.82	0.52
Channel 5	0.73	0.48	0.58
Channel 6	0.84	0.43	0.58
Ch.5 - Ch.6 <i>AND</i>	0.72	0.75	0.74
Ch.2 - Ch.4 <i>OR</i>	0.84	0.57	0.68

**Table 4.** Best results in terms of F1-score of the outlier detection technique for each of the 6 acoustic channels. *AND* and *OR* refer to the ensemble experiments.

## 4 An unsupervised learning technique

In this section, we present the streaming distance-based outlier detection algorithm, namely MCODE [7], that was used for early detection of failure on the dataset. We employ sliding windows. Given a set of objects  $\mathbb{O}$  and the threshold parameters  $R$  and  $k$ , we report all the objects  $o_i \in \mathbb{O}$  for which the number of objects  $o_j$ ,  $j \neq i$  for which  $dist(o_i, o_j) \leq R$  is less than  $k$ . The report should be updated after each window slide. Note that according to this definition, outliers may be reported during any time they belong to the window and not necessarily when they are first inserted into it.

The experimental setting is the same as the one used in the supervised approach with the difference that the *precision* is the ratio of all the true alarms to the number of total alarms, where true alarm is any prediction (in the form of warning outlier detection) made in a specified time gap before a machine stop. To avoid the problem of fine-tuning, we experimented with the combinations of 3 values of  $R$  and 3 values of  $k$ , i.e., 9 combinations of parameters. The window contains the last 3600 measurements and the window slide is fixed to 10% of the window size. As previously, we aim to predict faults 1 to 8 hours ahead.

In almost all of the experiments, there is a trade-off between recall and precision. Based on the algorithm parameters chosen, this trade-off can be configured in favor of either of the measures depending on the output needed or preferred. Table 4 shows the best results per channel in terms of the F1 score. An observation that can be drawn from the table is the difference in the results among the data sources, which means that each source can have a different impact on the PdM technique. It seems that especially regarding the acoustic emission sensors, specific positions in the press can yield better understanding of the faults in the press. In general, the unsupervised learning one managed to achieve different but equally interesting trade-offs compared to the supervised one, e.g., 83% recall and 48% precision (0.61 F1-score). We also tested an ensemble of the different parameterizations of the acoustic emission channels. Running more than one parameterization helps to improve the results and either exacerbate or mitigate the trade-off. Certain combinations can greatly increase one measure while slightly

decreasing the other one. The last two lines of Table 4 show indicative results from this experiment.

## 5 Discussion

In the previous sections, we have presented a supervised predictive approach based on a novel timeseries discretization approach and an unsupervised detection approach used as predictive mechanism. Both the approaches are applied on the same use case and dataset. The results of the two approaches suggest that there actually exist preceding hidden patterns or indications for most of the machine stops and prove that the novel MP-based timeseries discretization approach has managed to successfully reveal those previously unknown patterns. In addition, outlier detection can also enhance PdM.

Some further comments on the logs are as follows. The logs with the machine stops contained 82 records of 8 machine stop categories. These categories are quite different in their nature. We did not preprocess the machine stop logs before the execution of the experiments in order to assess the robustness of the approaches. Cleaning the logs is expected to decrease the precision and affect the recall of the unsupervised approach, however it will potentially increase the metrics of the supervised approach. Also, to stress-test the supervised learning approach, we have considered all the different machine stop categories as a single failure category, thus training a single model. The results of the supervised approach will potentially improve if separate models are used for each individual machine stop category.

Finally, we clarify that the purpose of this research work is not to compare the supervised against the unsupervised learning approach and promote the most efficient or appropriate one. On the contrary, the goal is to promote the strengths of each approach and to provide the basis for an heterogeneous ensemble solution utilizing multiple instances of both the supervised and unsupervised approaches. Overall, we envisage a multi-layer ensemble solution, where different instances of the same predictor type are combined at a lower layer, while, at a higher-level, different types of predictors form an ensemble.

## 6 Related Work

Data-driven techniques, where the data refer to past events, commonly in the form of log entries, are widely used in PdM. [8] is a key representative of the state-of-the-art. Another event-based approach is presented in [12], where historical and service data from a ticketing system are combined with domain knowledge to train a binary classifier for the prediction of a failure. As in the previous work, a feature selection [3] and an event amplification technique is used to enhance the effectiveness of the SVM-based classifier. Event-based analysis, based on event and failure logs, is also performed in [14], where it is assumed that the system is capable of generating exceptions and error log entries that are inherently relevant to significant failures. This work relies on pattern extraction

and similarity between patterns preceding a failure, while emphasis is posed on feature selection.

The work in [17] proposes a correlation-driven approach between different sensor signals and fault events to guide the PdM process. This approach tries to identify correlations between detected anomalies in different sensor signals, which are mapped to specific faults. Here, we focus on event processing, where events are generated from the sensors artificially.

Data-driven PdM is also related to online frequent episodes mining; research works [2] and [10] propose techniques in this topic. The key strength of [2] is that it can apply to an online (i.e. streaming) scenario. [10] further improves upon it through providing solutions for the case where the event types is unbounded. Complex-event processing (CEP) [6] is also a technology that enables PdM enforcement after warning sequential patterns have been extracted. A good overview of the the data-driven PdM is presented in [9].

Motif-detection in timeseries can also be used in prediction scenarios. The authors in [15] propose a tool that is able to predict outcomes based on weakly labeled time series of millions of data points. Finally, outlier detection is a vivid research field that has developed broad and multifaceted algorithmic solutions. The comparative study [13] presents a wide range of distance-based outlier detection algorithms and suggests that the MCO algorithm, which is employed in this work, is considered as a state-of-the-art solution in the streaming data processing for distance-based outlier detection.

## 7 Conclusions and Future Work

In this work, three state-of-the-art techniques in timeseries analysis, event-based PdM and streaming outlier detection are leveraged in order to provide effective PdM solutions operating directly on the output of sensors. The key strength is that no domain expert knowledge is required and one of the techniques does not require model building at all. More specifically, a novel timeseries discretization approach is proposed for the generation of *artificial* events, in order to enable the utilization of event-based predictive approaches. In parallel, distance-based outlier detection is shown to be effective in capturing early signs of abnormal equipment behavior. The solution is evaluated in a real setting, and the results are particularly encouraging achieving high F1 scores and useful trade-offs between the recall and precision metrics. As future work, we intend to work towards an ensemble solution and derive an efficient manner to tune the various parameters involved automatically. However, the most important next step is to build on these early insights into the benefits of our proposal and proceed to a more thorough experimentation and testing.

*Acknowledgement.* ■ This research work is funded by the Z-BRE4K project (funded by European Union’s Horizon 2020 research and innovation program under grant agreement No 768869).

## References

1. Aggarwal, C.C.: Outlier analysis. In: Data mining, pp. 237–263. Springer (2015)
2. Ao, X., Luo, P., Li, C., Zhuang, F., He, Q.: Online frequent episode mining. In: IEEE 31st Int. Conf. on Data Engineering (ICDE). pp. 891–902 (2015)
3. Bach, F.R.: Bolasso: model consistent lasso estimation through the bootstrap. In: Proc. of the 25th Int. Conf. on Machine learning. pp. 33–40. ACM (2008)
4. Ghosh, S., Halappanavar, M., Tumeo, A., Kalyanaraman, A., Lu, H., Chavarrià-Miranda, D., Khan, A., Gebremedhin, A.: Distributed louvain algorithm for graph community detection. In: IPDPS. pp. 885–895 (2018)
5. Kira, K., Rendell, L.A.: The feature selection problem: Traditional methods and a new algorithm. In: Aaii. vol. 2, pp. 129–134 (1992)
6. Kolchinsky, I., Schuster, A.: Efficient adaptive detection of complex event patterns. PVLDB **11**(11), 1346–1359 (2018)
7. Kontaki, M., Gounaris, A., Papadopoulos, A.N., Tsihlias, K., Manolopoulos, Y.: Efficient and flexible algorithms for monitoring distance-based outliers over data streams. Information systems **55**, 37–53 (2016)
8. Korvesis, P., Besseau, S., Vazirgiannis, M.: Predictive maintenance in aviation: Failure prediction from post flight reports. In: IEEE Int. Conf. on Data Engineering (ICDE). pp. 1414–1422 (2018)
9. Kovalev, D., Shanin, I., Stupnikov, S., Zakharov, V.: Data mining methods and techniques for fault detection and predictive maintenance in housing and utility infrastructure. In: 2018 International Conference on Engineering Technologies and Computer Science (EnT). pp. 47–52 (2018)
10. Li, H., Peng, S., Li, J., Li, J., Cui, J., Ma, J.: Once and once+: Counting the frequency of time-constrained serial episodes in a streaming sequence. arXiv preprint arXiv:1801.09639 (2018)
11. Lu, Y.: Industry 4.0: A survey on technologies, applications and open research issues. Journal of Industrial Information Integration **6**, 1 – 10 (2017)
12. Sipos, R., Fradkin, D., Moerchen, F., Wang, Z.: Log-based predictive maintenance. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1867–1876. ACM (2014)
13. Tran, L., Fan, L., Shahabi, C.: Distance-based outlier detection in data streams. Proceedings of the VLDB Endowment **9**(12), 1089–1100 (2016)
14. Wang, J., Li, C., Han, S., Sarkar, S., Zhou, X.: Predictive maintenance based on event-log analysis: A case study. IBM Journal of Research and Development **61**(1), 11–121 (2017)
15. Yeh, C.C.M., Kavantzias, N., Keogh, E.: Matrix profile IV: using weakly labeled time series to predict outcomes. Proceedings of the VLDB Endowment **10**(12), 1802–1812 (2017)
16. Yeh, C.M., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, H.A., Silva, D.F., Mueen, A., Keogh, E.J.: Matrix profile I: all pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. In: IEEE ICDM. pp. 1317–1322 (2016)
17. Zhu, M., Liu, C.: A correlation driven approach with edge services for predictive industrial maintenance. Sensors (Basel, Switzerland) **18**(6) (2018)