Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 184 (2021) 565–572

Procedia Computer Science

www.elsevier.com/locate/procedia

The 4th International Conference on Emerging Data and Industry 4.0 (EDI40), March 23 - 26, 2021, Warsaw, Poland

A micro-service-based machinery monitoring solution towards realizing the Industry 4.0 vision in a real environment

Athanasios Naskos[a,*], Nikodimos Nikolaidis[a], Vasileios Naskos[a], Anastasios Gounaris[b], Daniel Caljouw[c], Cosmas Vamvalis[a,d]

[a]*Atlantis Engineering, Thessaloniki, Greece*
[b]*Aristotle University of Thessaloniki, Greece*
[c]*Philips - The Netherlands*
[d]*European Federation of National Maintenance Societies*

Abstract

This work presents a modular Smart Maintenance Platform (SMP), which offers a toolbox of condition monitoring tools for the day-to-day inspection of the machinery equipment in a real industrial environment. SMP complements the manual inspection performed by the maintenance engineers with advanced data-analytics solutions that utilize both reactive and proactive techniques based on state-of-the-art machine learning proposals. Appropriate data acquisition mechanisms are built to acquire the sensorial data and transform them to suitable forms for data processing. Higher-level monitoring policies are enabled, through the fusion of multiple monitoring tasks increasing the precision of the results. The generated results are persisted for the continuous enhancement of the solution through the retraining of the proactive approach or the re-configuration of the reactive ones. The generated results are visualized in a user-friendly format, to help the maintenance engineers to assess the severity of the situation and to intervene in the machinery equipment if deemed necessary.

Keywords: predictive maintenance;timeseries analysis;multiple instance learning;streaming outlier detection;Industry 4.0

1. Introduction

The 4th Industrial revolution enables the utilization of advanced hardware virtualization techniques allowing the deployment of multiple services on the same hardware, which alleviates the cost for expensive hardware acquisition and maintenance. Service-based solutions take advantage of these technologies providing to the industries efficient software tools, deployed locally or to remote data centers. Such a solution is presented in this work, where a micro-

[*] Corresponding author. Tel.: +30-231-023-3266 ; fax: +30-231-080-4947.
E-mail address: naskos@abe.gr

10.1016/j.procs.2021.03.071

service-based Smart Maintenance Platform (SMP) that encapsulates state-of-the-art machine learning approaches is used to provide a day-to-day assisting tool for the detection and the prediction of the failures of the machinery equipment to the maintenance engineers of the Philips plant in the Netherlands.

While there is no precise definition of the term *"micro-service"*, [10] mentions "Micro-services are small, autonomous services that work together", while RedHat states "Microservices are distributed and loosely coupled, so one team's changes won't break the entire app"[1]; both these descriptions apply to our case. More specifically, this work complements our previous work in [9], which focused mostly on the algorithmic aspect of the data analytic tools and their performance. On the contrary, the present work sheds light on the development architecture for the software solution (termed SMP) that is used to satisfy the requirement of the same use case as in [9], presenting the web interfaces and the functionality offered to the data scientist and the maintenance engineers. A set of micro-services, which comprise the SMP, are presented to cope with the data acquisition, data analysis and results dissemination processes. The snapshot figures presented in the following sections are from a real industrial case study. Overall, the contribution of this work is the presentation of a complete end-to-end solution to data-driven predictive maintenance (PdM) in Industry 4.0.

In the following section, the concept of the proposed SMP is introduced, describing briefly the provided services. Section 3 presents the micro-services related to the Data Acquisition process with a focus on the handling of the data streams. The Sections 4 and 5 discuss about the Data Analytic micro-services of the platform, while Section 6 introduces the concept of the Results Fusion, which can increase the confidence of the results. Section 7 presents the reporting interfaces used in the real industrial case study of the paper. Finally, Sections 8 and 9 discuss the related work and conclude the paper, respectively.

2. Smart Maintenance Platform

The diverse nature of the machinery failures imposes the use of multiple condition monitoring approaches to achieve acceptable precision and recall rates. The maximization of the former metric would provide less false positive alarms increasing the confidence of the maintenance engineers towards the platform, while the maximization of the latter would increase the number of successfully detected incidents, making the solution appealing to the engineers to use it as a day-to-day tool. For that purpose, we propose the use of an ensemble of monitoring approaches providing both descriptive (i.e. reactive) and predictive (i.e. proactive) solutions, towards the realization of a prescriptive maintenance in order to proactively plan the required maintenance actions.

The different monitoring approaches are developed independently, as loosely-coupled micro-services. The combination and co-ordination of all the micro-services forms the proposed SMP, which is extensible by design. Figure 1 presents the architecture of the SMP, where all the micro-services (gray rectangles) communicate with each other through a central bridge (i.e. EventBus), which in our case can be a messaging queue like Mosquitto [2], RabbitMQ [3], or Kafka [4].

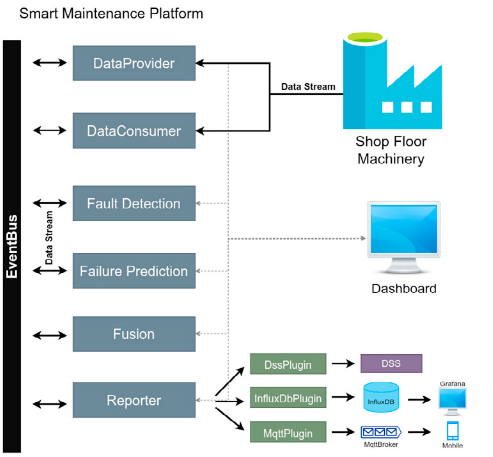


Fig. (1). Smart Maintenance Platform (SMP) - Micro-service-based Architecture.

The SMP offers the following micro-services:

[1] https://www.redhat.com/en/topics/microservices/what-are-microservices
[2] https://mosquitto.org/
[3] https://www.rabbitmq.com/
[4] https://kafka.apache.org/

- *DataProvider*, which is responsible for fetching data from various data sources. Its main role is to periodically request data from its customer (e.g., a database of an installed monitoring platform) and to publish the obtained data to the EventBus for processing by the data analytic micro-services.
- *DataConsumer* has a similar role to the DataProvider, with the difference that it receives data instead of requesting them periodically. The combination of these two micro-services enhances the agility of the SMP regarding the communication with third party software for data acquisition purposes.
- *Fault Detection*, which is one of the reactive data analytic micro-services offered by the SMP. It currently offers an ensemble of three monitoring approaches in order to accurately detect a fault combining unsupervised learning with domain expertise.
- *Failure Prediction*, which is based on a supervised machine learning approach in order to predict an upcoming failure in the machinery equipment (i.e., it is a proactive approach).
- *Fusion*, which combines the output of multiple either detection or prediction approaches based on pre-defined policies to increase the accuracy of the results.
- *Reporter*, which uses a plugin mechanism to communicate with different systems (in the figure, a Decision Support System (DSS), a timeseries database (InfluxDB) and a MQTT broker are shown) to propagate the analysis results.

The SMP offers a web interface (i.e. Dashboard) for the configuration of the micro-services and the visualization of the results. Figure 2 presents the home page of the SMP, which incorporates multiple Grafana dashboards.

The following sections elaborate on the offered functionalities of the SMP, presenting the Graphical User Interface of each micro-service.

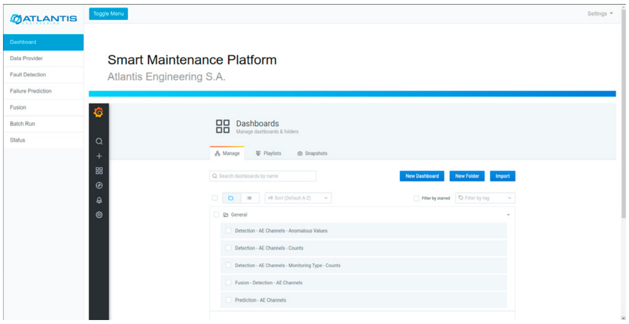

ATLANTIS ENGINEERING
Toggle Menu
Settings
Dashboard
Data Provider
Fault Detection
Failure Prediction
Fusion
Batch Run
Status
Smart Maintenance Platform
Atlantis Engineering S.A.
Dashboards
Manage dashboards & folders
Manage
Playlists
Snapshots
Search dashboards by name
New Dashboard
New Folder
Import
Sort (Default A-Z)
Filter by starred
Filter by tag
General
Detection - AE Channels - Anomalous Values
Detection - AE Channels - Counts
Detection - AE Channels - Monitoring Type - Counts
Fusion - Detection - AE Channels
Prediction - AE Channels

Fig. (2). Home page with Grafana system integrated.

3. Data Acquisition

The core functionality of the SMP is data processing, hence the agility of the data acquisition process is important in order to be able to obtain data from any relevant source. The SMP offers two micro-services for data acquisition, the DataProvider for fetching data to the platform upon request (e.g. HTTP GET requests to APIs) and the DataConsumer for receiving data from external sources (e.g. through subscription to MQTT topics).

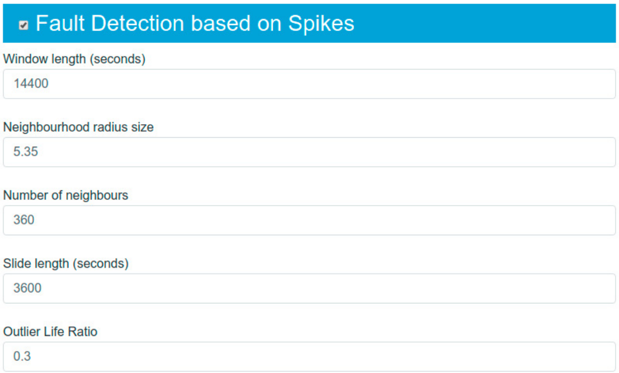
The DataProvider encapsulates a plugin mechanism to support the communication with any data source. Custom implementations are offered according to the use case requirements in order to build the required bridges to communicate with any system like Enterprise Resource Planning (ERP), Manufacturing Execution System (MES), Computerized Maintenance Management Systems (CMMS), Databases (DB), Application Programming Interface (API) or even parse text files to obtain the needed information. DataProvider is equipped with a scheduler, which is capable of handling multiple data fetching tasks for periodically acquiring data from multiple sources simultaneously.

4. Fault Detection

An ensemble of three monitoring approaches is offered by the Fault Detection micro-service. The rationale behind the utilization of an ensemble of techniques is that no single monitoring approach is capable of identifying all the possible faults in the machinery equipment due to the different footprint of each fault on the sensor data; in other words, the solutions can be seen as complementary to each other. The application of the Fault Detection component as an ensemble is optional. The user can select whether to use an ensemble of any combination of more than one technique or even a single monitoring technique, based on the nature of the fault that she wants to detect.

The first offered technique is called "Fault Detection based on Spikes", and it is presented in Figure 3. The given name is actually a simplification of what this technique offers.

More specifically, this monitoring approach is based on the Micro-Cluster-based Continuous Outlier Detection (MCOD) algorithm. MCOD is a state-of-the-art unsupervised distance-based outlier detection algorithm for streaming data [7], the advantages of which are highlighted in [16]. Distance-based outlier detection is a prominent example of proximity-based outliers, where an outlier is determined based on the distance from its neighbors [1]. This allows the identification of outlier values in the sensorial signal without the specification of hard thresholds on the actual values, but only thresholds for the neighborhood definition (i.e. neighborhood radius R and number of neighbors k) for a point to be designated as inlier, or not. Formally, an inlier is any point that has at least k other points within distance R, using in our case as the default distance metric the Euclidean distance. In a streaming setting, the solution adapts to evolving conditions through considering a sliding window keeping only the most recent values; the exact window size is the first parameter in Figure 3. The window slide size is user-defined. The last parameter in the figure aims to tune how aggressive outlier reporting is: for a window of length 60 secs with slide 10 secs, any point will be active for 6 slides. The outlier life ratio denotes the proportion of slides that the point needs to be designated as outlier before being reported; as such a ratio around 0.5 ensures that data points corresponding to normal changes in the behavior of the equipment are not treated as alarms. To account for the issue that distance-based outlier detection is notoriously sensitive to its input parameters R and k, there are variations of the algorithm for multi-parameter executions [14]. There are also distributed implementations [15], which utilize the Flink framework for handling streaming big data in a scalable manner in case the input data flow is too intense.


☑ Fault Detection based on Spikes

Window length (seconds)
14400

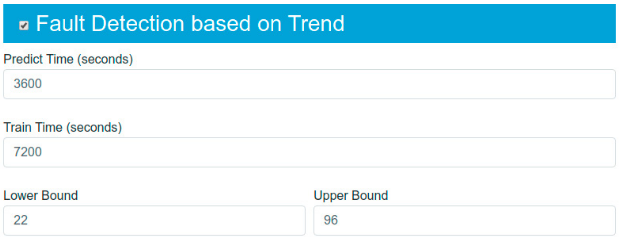
Neighbourhood radius size
5.35

Number of neighbours
360

Slide length (seconds)
3600

Outlier Life Ratio
0.3

Fig. (3). Parametrization of Fault Detection based on Spikes.


☑ Fault Detection based on Trend

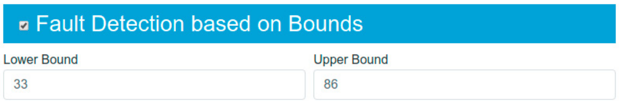
Predict Time (seconds)
3600

Train Time (seconds)
7200

Lower Bound
22

Upper Bound
96

Fig. (4). Parameterization of Fault Detection based on Trend.


☑ Fault Detection based on Bounds

Lower Bound
33

Upper Bound
86

Fig. (5). Parameterization of Fault Detection based on Bounds.

The previous monitoring approach is not able to detect slowly trending phenomena, as for example, the wear in the cutting edges of a die of a press. To address this issue, we propose the combination of the previous technique with a linear regression approach (see Figure 4). This technique leverages the fact that raw data are either locally stored in the timeseries database or can be retrieved multiple times from their source with the help of the Data Provider, but for small to moderate sizes, the most recent data can be held in main memory, exactly as in the previous case. The user specifies the training time period, where past data are collected in order to be fed to the linear regression function, and then to obtain the linear trend of the measurements for the next “prediction time” seconds. If the predicted value violates the specified lower of upper thresholds, an alert is generated. These thresholds are also guided with the help of historical data, from which the mean μ and standard deviation σ can be easily extracted. Then the thresholds are obtained from the formula $\mu \pm \alpha\sigma$, where typically, $2 \leq \alpha \leq 3$.

The last monitoring approach of the ensemble is a simple threshold-based technique, which is used as a fallback mechanism to detect any serious deviation from the normal behaviour that the previous two monitoring approaches failed to identify. As it is presented in Figure 5, the user again provides a lower and/or an upper threshold, from $\mu \pm \beta\sigma$, where $2 \leq \beta \leq 3$. When the linear regression approach is used in an ensemble with the other two approaches, it is advised to set relaxed thresholds in order to avoid generating multiple alarms in noisy environments, i.e. $2 \leq \beta < \alpha \leq 3$. Otherwise, the thresholds can be set in a stricter manner, as shown in the figure.

The discussion about how the results are fused is deferred to Section 6. Finally, the whole framework is extensible, i.e., additional techniques can be added.

5. Failure Prediction

The Failure Prediction encapsulates a proactive monitoring approach, which is a refinement of the procedure described in [9]. Our overall technique comprises two main modules: (i) an offline one, which yields a model for predicting failures; and (ii) an online one, which leverages the produced model for run-time predictions. Several machine learning techniques exist for various application domains [4] but, as stated in [8], *"each application exhibits it's own special characteristics that have great impact on the design of the corresponding algorithm"*. The key characteristics of a PdM task in our setting compared to traditional classification is that failure events are very rare and the feature set very sparse. In this setting, building upon the discrete-event based technique in [8], initially proposed for the aviation industry, is a promising approach. Then, the main challenge is to fill the gap from the initial measurements from sensors to event generation; and then to adapt [8] in our setting effectively.

Our novel rationale is that, instead of classifying time-series as a discrete set of events, we map time-series to a sequence of *artificial* events thus placing no burden to engineers to annotate the sensor measurements. We leverage the *Matrix Profile* (*MP*), which is a data structure that finds similarities between subsequences in a time-series [18, 17]. Each set of similar subsequences forms a discrete event type. This allows us to transform time-series to event sequences. Open source implementations of *MP* are publicly provided.[5], however, we have preferred to employ our own version, which relies on the Faiss library [6].[6]

During training, we split the data in episodes ending to a failure. We also group these events according to small time periods, e.g., 1 minute. Further, we associate each segment to a risk value based on its time distance from the failure event in the corresponding episode. To this end, a sigmoid function is used. Additional configurable pre-processing steps inlcude: (i) pruning of very rare and very frequent events in each group; (ii) pruning of complete groups very far away from the failure event; and (iii) applying classical feature selection, such as ReliefF [11]. Then, we cast our problem as a regression one, where we learn the weights of each event type in a group so that the sum of each group fits the sigmoid function. To this end, the Random Forests (RF) algorithm is used.

The derived model can be used online in a straightforward manner. During the online model usage, the user specifies a prediction threshold (i.e. the lowest threshold on the predicted risk), which is a value between 0 and 1 and defines the sensitivity of the approach, as lower values would generate more alarms. The model training process is not provided through the web interface. A data scientist uses the provided API in order to train RF models based on historical sensorial data and maintenance logs. The user selects the trained model that she wants to use from a drop down list. The selection depends on the sensors that need to be monitored and other parameters specific to the training of each model (the parametrization of each trained model is presented to the user upon every model selection).

6. Results Fusion

As it is already stated, no single monitoring approach is adequate to accurately detect or predict a fault, due to both their inherent limitations and the different nature of the faults. SMP offers an option to combine multiple detection or prediction approaches in order to output a single alarm with increased confidence.

For the Fusion of the detection monitoring approaches, the user selects from a list:

1. which running detection tasks would like to fuse;
2. what is the time frame ("Refresh Period" in the figure) to wait for reports from all the tasks before merging into a single result;
3. which is the fusion policy (two options are currently provided, namely logical "AND" and "OR"); and
4. whether to expect a report on the exact same point from all the detection tasks or any report on the given time frame ("Scope" in the Figure), when the "AND" policy is used.

Regarding the two provided policies, the "AND" policy expects all the selected detection tasks to report a single result, while the "OR" policy reports all the incoming results.

[5] https://www.cs.ucr.edu/~eamonn/MatrixProfile.html

[6] https://github.com/facebookresearch/faiss

For the Fusion of the prediction tasks, a more complex approach is used inspired by [3], which includes a training phase in order to assess the efficiency of each available predictor. More specifically, in order to fuse prediction results, the Fusion micro-service assigns specific weights to the predictors, which sum to 1 and are inversely proportional to the Mean Square Error (MSE) that each predictor achieved on the training phase. The result is the weighted average of all the results, treating explicitly the bias compensation. An alternative is to take diversity into account, e.g., as in [5]. Overall, through combining the results from multiple Fault Detection or Failure Prediction approaches, SMP is capable of decreasing the false positive reports and increasing the maintenance engineer's trust to our platform.

7. Results Report on a Real Industrial Environment

In this section, we present the failure reporting interfaces that are provided to the maintenance engineers of a Philips plant in the Netherlands. As in our previous work [9], the same case study is used, where acoustic emission signals are recorded from a cold forming press, in order to record the sound footprint of every punch of the stamper of the press. The process begins from a metal strip that comes from an input reel and passes through the first quality check. The strip then enters the press and is processed by each one of the six dies. After the strip takes its final form, it exits the press and passes a second quality check.

The aim is, through the reactive and proactive analysis of the acoustic emissions and the quality measurements, to detect or predict a failure in a timely manner. This can potentially shrink the time spent during maintenance through detecting a fault before it propagates to the other components of the press, and it also enables the development of a prescriptive maintenance policy based on the precise prediction of an upcoming failure.

SMP incorporates the Grafana[7] visualization platform, which, in combination with a timeseries database (in our case, InfluxDB), is used to visualize the results in an efficient and user-friendly manner for the maintenance engineers.

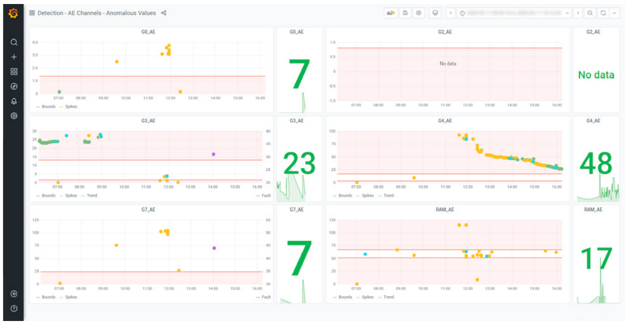


Fig. (6). Results visualization using Grafana dashboards. The red areas mark the specified upper and lower bounds per acoustic channel. The purple dot marks a maintenance action applied on the press.

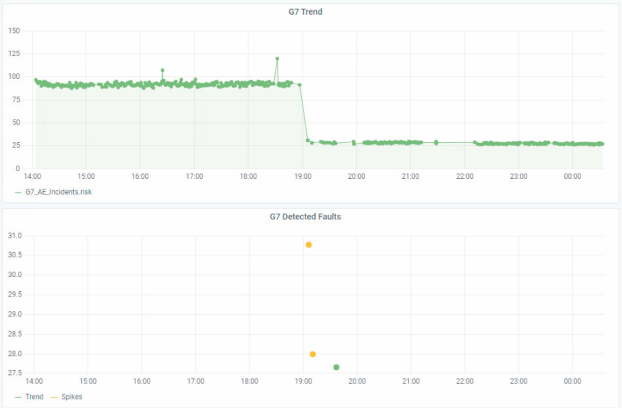


Fig. (7). Fault Detection on Acoustic Emissions - Different Modules Deviation. The change of the module is applied in the middle of the figure. The graph in the bottom depicts the points where anomalies are detected.

There are six acoustic emission sensors (i.e. RAM_AE, G0_AE, G2_AE, G3_AE, G4_AE, G7_AE) installed on the press as presented in the respective Grafana dashboard in Figure 6. The diagrams show the acoustic emission measurements that are deemed as possible faults (i.e. outliers or out of bounds) by any of the three fault detection monitoring options (listed as Spikes, Bounds and Trend in the legends of the diagrams, respectively). The red areas mark the specified upper and lower bounds per acoustic channel. In the G3_AE and the G7_AE channels, a purple dot is plotted, which marks a maintenance action applied on the press, without taking into consideration the results of the analysis. As we can observe, there are lots of detected incidents several hours before the maintenance action (i.e. 23 for G3_AE and 7 for G7_AE), indicating that our solution could potentially alert the maintenance engineers several hours before the actual incident.

[7] https://grafana.com/

There are cases where, for the same die of the press, there are more than one modules. Normally both the modules should have the same footprint in the acoustic emission signal, however as it is depicted in Figure 7, which focuses on the G7_AE channel, there is deviation from the expected behavior between the two modules. This deviation is not necessarily mapped to the quality of the end product, however it is something that needs to be taken into consideration by the maintenance engineers for further investigation. As it is presented in the figure, our detection techniques were able to spot the sudden signal drop and to depict the abnormal behavior in the lower graph. There are multiple other dashboards like the ones provided by the failure prediction or the fusion tasks, which are omitted due to lack of space.

A mobile application is also provided to the maintenance engineers for showing alerts regarding the reported faults. The engineers have the option to provide their feedback regarding a received alert. Their feedback is persisted in a database in order to be utilized for the fine-tuning of the detection approaches and the retraining of the prediction ones.

8. Related Work

Predictive maintenance is a multi-billion (i.e. $23.8B) worth market, according to the IoT Analytics industry report [13], that attracts the attention from big companies like IBM [8], SAP [9], SAS [10], Siemens [11], GE [12] and ABB [13] to small companies focused solely on that field, like Augury [14], Falkonry [15], Sight Machine [16] and others [12]. The commercial options from the big companies offer cloud-based solutions, which might not focus to specific industries, providing services to a wider range of use cases at the expense of possibly having limited domain expertise. On the other hand, solutions offered by smaller companies with expertise on specific industries may lack on the communication with existing complementary software like Computerized Maintenance Management System (CMMS), Decision Support System (DSS) or an Asset Management System (AMS).

Apart from the focus on specific industry and the communication with other systems, the deployment of the solutions is also of interest, as fog computing and edge devices help to bring the process of the analysis closer to the equipment where the data are produced. As an example, the authors in [2] identify four main requirements for industry applications (i.e. latency, reliability, security and privacy) that a smart maintenance solution should satisfy. They propose a platform for smart maintenance management, which utilizes both cloud and fog computing. To the contrary, our solution is light-weight and can easily be deployed on non-high-end devices close to the edge. The presented platform encapsulates the domain knowledge obtained by the Philips' maintenance engineers, provides the required extensible functionality for the communication with any existing system and the fact that the functionality is provided through decoupled micro-services, facilitates the transfer of the solution to the cloud and fog computing, if necessary.

Regarding the techniques used, summaries of the machine learning solutions that can be useful in PdM have appeared in works such as [4], where, apart from RF, various forms of neural networks (NNs) and SVMs are also commonly used. Our approach is shown to be superior to SVMs [8] and does not require high amounts of training data as in NNs; failure data is typically sparse in real industrial settings. Another distinctive feature is that we combine tailored machine learning models with intensive data preprocessing, streaming outlier detection and rule-based techniques to yield a complete ensemble solution that is practical. Also, we provide the full architectural details about how this solution can be built.

[8] https://www.ibm.com/services/technology-support/multivendor-it/predictive-maintenance

[9] https://www.sap.com/products/predictive-maintenance.html

[10] https://www.sas.com/en_us/software/asset-performance-analytics.html

[11] https://new.siemens.com/global/en/products/services/industry/predictive-services.html

[12] https://www.ge.com/digital/iiot-platform

[13] https://ability.abb.com/customer-segments/marine-ports/remote-diagnostics-predictive-maintenance/

[14] https://www.augury.com/category/predictive-maintenance/

[15] https://falkonry.com/

[16] https://sightmachine.com/product/ema/

9. Conclusions and Future Work

This work presents the Smart Maintenance Platform, which is an assisting tool for the maintenance engineers of the Philips plant in the Netherlands. The platform offers a toolbox of condition monitoring tools for the reactive and proactive inspection of the machinery equipment utilizing Industry 4.0 technologies. The functionality is offered through decoupled micro-services, which communicate with each other through a central event bus. Appropriate micro-services handle the data acquisition process from external sources, the data analysis for the detection or prediction of faults in the machinery equipment and the result fusion and dissemination, providing visualizations and mobile notifications for the maintenance engineers.

The selected micro-service architecture offers a modular implementation, which can be extended or adapted to each use case requirements. The platform will be extended both externally and internally. The external extensions consider the support for more data sources through the implementation of added data source plugins for the DataProvider or more data targets implementing additional plugins for the Reporter micro-service. The internal extensions consider the implementation of added data-analytics micro-services for providing functionalities like the estimation of the Remaining Useful Lifetime (RUL) of the equipment or the Health Profile-based monitoring that extends the Fault Detection approach with the capability to identify critical deviations from the normal behavior.

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., 2017. Outlier analysis (2nd ed.).

[2] Ashjaei, M., Bengtsson, M., 2017. Enhancing smart maintenance management using fog computing technology, in: 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), IEEE. pp. 1561–1565.

[3] Bonissone, P.P., Xue, F., Subbu, R., 2011. Fast meta-models for local fusion of multiple predictive models. Applied Soft Computing 11, 1529–1539.

[4] Carvalho, T.P., Soares, F.A.A.M.N., Vita, R., da Piedade Francisco, R., Basto, J.P., Alcalá, S.G.S., 2019. A systematic literature review of machine learning methods applied to predictive maintenance. Comput. Ind. Eng. 137.

[5] Hu, C., Youn, B.D., Wang, P., Yoon, J.T., 2012. Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life. Reliability Engineering & System Safety 103, 120–135.

[6] Johnson, J., Douze, M., Jégou, H., 2017. Billion-scale similarity search with gpus. arXiv preprint arXiv:1702.08734 .

[7] Kontaki, M., Gounaris, A., Papadopoulos, A.N., Tsichlas, K., Manolopoulos, Y., 2016. Efficient and flexible algorithms for monitoring distance-based outliers over data streams. Information systems 55, 37–53.

[8] Korvesis, P., Besseau, S., Vazirgiannis, M., 2018. Predictive maintenance in aviation: Failure prediction from post-flight reports, in: 2018 IEEE 34th International Conference on Data Engineering (ICDE), IEEE. pp. 1414–1422.

[9] Naskos, A., Kougka, G., Toliopoulos, T., Gounaris, A., Vamvalis, C., Caljouw, D., 2019. Event-based predictive maintenance on top of sensor data in a real industry 4.0 case study, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer. pp. 345–356.

[10] Newman, S., 2015. Building microservices: designing fine-grained systems. " O'Reilly Media, Inc.".

[11] Robnik-Šikonja, M., Kononenko, I., 1997. An adaptation of relief for attribute estimation in regression, in: Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97), pp. 296–304.

[12] Scully, P., 2019a. Predictive maintenance companies landscape 2019 iot analytics. Iot-analytics.com .

[13] Scully, P., 2019b. Predictive maintenance report 2019-2024 iot analytics. Iot-analytics.com .

[14] Toliopoulos, T., Gounaris, A., 2019. Multi-parameter streaming outlier detection, in: 2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI), IEEE. pp. 208–216.

[15] Toliopoulos, T., Gounaris, A., Tsichlas, K., Papadopoulos, A., Sampaio, S., 2018. Parallel continuous outlier mining in streaming data, in: 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA), IEEE. pp. 227–236.

[16] Tran, L., Fan, L., Shahabi, C., 2016. Distance-based outlier detection in data streams. Proceedings of the VLDB Endowment 9, 1089–1100.

[17] Yeh, C.C.M., Kavantzas, N., Keogh, E., 2017. Matrix profile iv: using weakly labeled time series to predict outcomes. Proceedings of the VLDB Endowment 10, 1802–1812.

[18] Yeh, C.M., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, H.A., Silva, D.F., Mueen, A., Keogh, E.J., 2016. 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.