Enabling of Predictive Maintenance in the Brownfield through Low-Cost Sensors, an IIoT-Architecture and Machine Learning

Patrick Strauß, Markus Schmitz
Innovation, Digitalization, Data Analytics
BMW Group
Munich, Germany
e-mail: patrick.strauss@bmwgroup.com,
markus.ms.schmitz@bmwgroup.com

René Wöstmann, Jochen Deuse
Institute of Production Systems (IPS)
Technical University of Dortmund
Dortmund, Germany
e-mail: rene.woestmann@ips.tu-dortmund.de,
jochen.deuse@ips.tu-dortmund.de

Abstract— **Predictive maintenance is one of the major drivers of Industry 4.0 as it can significantly reduce costs by improving overall equipment effectiveness and extending the remaining useful life of production machines. Most of the potential lies in the brownfield with old equipment where no sensors or connectivity are available. This paper shows how these production machines can be enabled for predictive maintenance by retrofitting with low-cost sensors, an Industrial-Internet-of-Things-architecture and machine learning. An industrial implementation on a heavy lift Electric Monorail System at the BMW Group will be shown.**

predictive maintenance; sensor; IIoT-architecture; machine learning

I. INTRODUCTION

As digitalization is advancing at rapid pace, it opens up new opportunities in the design and support of business and production processes. New technologies in the field of data acquisition, sensor technology, Industrial Internet of Things (IIoT), storage, cloud computing and machine learning enable cyber-physical production systems [1–4], which are often referred to as Industry 4.0 [1]. According to [5], one of the greatest potentials for increasing productivity lies in maintenance. Here, the focus switches from improved business reporting and condition monitoring to predictive maintenance [6]. Compared to the still existing reactive and preventive maintenance approaches with fix maintenance intervals, predictive maintenance enables new opportunities for predicting upcoming failures. This is done by calculating the remaining useful life (RUL) or by anomaly detection based on mathematical algorithms and machine learning models [7, 8]. For this purpose, production machines must be digitized with sensors that can measure the mechanical condition [9]. This information can be used to avoid unplanned downtime, increase availability and efficiency, optimize maintenance strategies and thus save costs [5, 9].

Even though the potentials of predictive maintenance have been recognized, there is a lack of implementation in practice [10]. Barriers are high due to the implementation and hardware costs, uncertain outcomes, and the lack of access to a fully extended predictive maintenance solution [11]. Also, [12] states that methods for extracting, processing and analyzing large data sets were developed within the big data analytics environment [13], but were only used to a limited extent in the area of predictive maintenance. Furthermore, today's production sites are often equipped with old machines without sensors or connectivity [14]. According to [12, 15], the greatest potential lies in the brownfield with old systems. Production is often characterized by those old machines, since only minor technical production adjustments are necessary for the implementation of new products. To enable those and even new machines for predictive maintenance, retrofitting (installation of additional sensors on the production machines) can be carried out. It can solve the so-called brownfield problem [16], which is the challenge of embedding predictive maintenance solutions into existing framework conditions [17]. To overcome the challenges, a cyber-physical system (CPS) for predictive maintenance by retrofitting is presented. As shown in Fig. 1, the solution is based on low-cost sensors, an IIoT-architecture and machine learning.

The remainder of this paper is as follows: In Section II, related work is presented. Section III proposes the need and requirements of low-cost sensors in production. Section IV provides an overview of existing IIoT-architectures and the requirements for implementing a scalable predictive maintenance solution by retrofitting. Section V presents an industrial implementation of the system including the application of various machine learning methods, while Section VI summarizes the results and experiences of the industrial implementation.

II. RELATED WORK

Retrofitting approaches are not fundamentally new. For example, in the aerospace industry various investigations have been carried out to upgrade systems with wireless sensor networks to CPS for enabling applications such as predictive maintenance [18, 19]. Lee coined the term CPS, which is characterized by a combination of real (physical) objects and processes that are monitored and controlled by embedded computers within information networks [20]. Industry 4.0 is considered to be a form of CPS [21] and many studies have been carried out to this subject. Lanfer and Trautmann for example described a retrofit approach for integrating RFID in existing material flow systems

978-1-5386-5035-6/18/$31.00 ©2018 IEEE

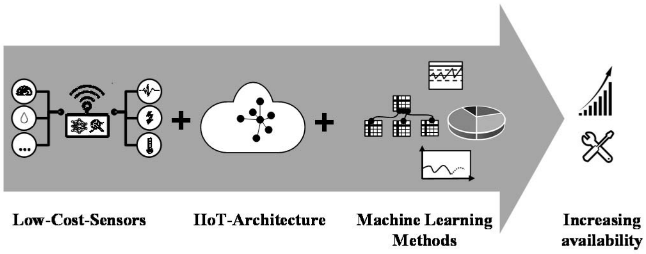

Figure 1. Components of CPS for predictive maintenance

[22]. The research project Retronet also has been concerned with the question of how an IIoT-architecture and corresponding embedded systems and gateways for retrofitting should be designed [23]. In the Cyber System Connector research project, the development of a connector for dynamic plant documentation created the basis for continuously updated virtual twin and real-time maintenance management [24]. Numerous research projects deal with retrofit approaches in specific domains, such as increasing productivity in mechatronics production [25], intelligent quality control systems (IQ4.0) [26], the human-machine interface through the development of assistance systems and dashboards (CPPS Process Assist) [27] or assistance systems for existing plants in the process industry (CyProAssist) [28].

Furthermore, [29] presented an IIoT monitoring solution for enabling monitoring and predictive maintenance application of industrial machines. Their system is based on battery powered sensors installed at a power plant. Another system which is designed for low energy vibration sensors through a new wireless sensing and network management was presented by [30].

Even though several studies and applications for retrofitting based on low-cost embedded systems, such as RaspberryPi or Arduino, and low-cost sensors based on Microelectromechanical systems (MEMS) have been carried out [4, 29–38], the requirements for an industrial rollout were not considered. In summary, there is still no system on the market that meets industrial requirements and can be used by companies with little effort for condition monitoring and for predictive maintenance. The following section presents an in-house retrofit approach of the BMW Group in cooperation with the IPS, which meets these requirements [39].

III. Low-Cost IIoT-Sensor-Device

Most common reasons of machine failures are mechanical defects of a component due to wear [40]. This is often indicated by signs, which can be measured by sensors. In conjunction with mathematical and statistical models, failures can be avoided and lead to an increase of the overall equipment efficiency (OEE) of industrial plants [9]. Although the data availability of new systems is increasing due to integrated sensor technology, the use of it is difficult due to the lack of standardized interfaces. The sensors must collect process, machine and quality data in large quantities and at high speeds. The sensors and measurements required for predictive maintenance are generally not taken into account when designing machines [12]. New possibilities in sensor and embedded technologies as well as IIoT-architectures enable a paradigm shift to achieve the vision of a digital factory – also in the brownfield [17]. Another challenge is that according to a present study 75 % of the industrial companies would not be willing to invest more than 500 EUR to digitize their machines [41]. [36] also states that the installation of industrial sensors is complex and cost-intensive. Furthermore, condition monitoring solutions available on the market are expensive and only designed for specific applications [11]. For a comprehensive rollout, a low-cost retrofit solution applicable in the industry is needed.

For implementing IoT devices within a production plant, industrial requirements of the following fields have to be considered: Sensors & hardware, communication, software & application and management. While within the field of sensor & hardware needed IP-requirements are specified, the field of communication determines permitted wired and wireless communication technologies. One example is the use of only 5 GHz Wi-Fi with WPA2-Enterprise encryption to avoid interference with other wireless transmissions and ensure security and safety. Software & application defines e.g. required JSON-structure, needed IoT protocols like AMPQ or MQTT and access and security policies. For a plant-wide rollout, management functions to configure the firmware, the operating system and the applications are needed and have to be specified.

Since there were no solutions available on the market that fulfilled all requirements or supported a low-cost approach, an own solution was developed. Therefore, a BananaPi was used as the central embedded device. According to a previous conducted study, the most common sensors for predictive maintenance are acceleration, current and temperature sensors [42]. In this project, the temperature sensor MCP9808 and the vibration sensor ADXL345 from Adafruit were used. To fulfill the requirement of a non-invasive installation, inductive folding current transformers were implemented as well. Furthermore, the BananaPi was equipped with a 5 GHz Wi-Fi dongle in order to enable a wireless communication. The application for reading the sensor values, data preparation and communication with the IIoT cloud was realized with the Microsoft Azure IoT Device SDK. Using the SDK, management functions like cloud-to-device messages were also implemented. This enables changing the measuring frequency of the sensors or configuring the IoT connection at the IoT device over the cloud. Further requirements and capabilities of an IIoT architecture are described in the next section.

IV. IIoT-Architecture

The data collection of the above-mentioned sensor values as well as their integration confronted many users with a great challenge in the past. New possibilities of the IIoT offer a strong simplification to combine internal machine data, external sensors and adjacent maintenance systems. The integration of internal and external data is of high importance, as not all relevant information can be covered externally. Internal data from a PLC (Programmable Logic Controller) is often required to get information of the current status of the machine and process. Especially within the field of maintenance, integrating incident reporting and maintenance management systems is needed to connect cause (e.g. increasing vibrations at a specific section of the process) and effects (e.g. breakdowns). A lot of so-called "reference architecture models" like RAMI 4.0 [43], ARM [44] or IIRA [3] for the construction of (Industrial) IoT environments that integrate assets up to applications and services in the cloud have been created in the recent past. Tab. 1 gives an overview of existing reference architecture models, complementing the above by the approaches of WSO2 [45], IAAS [46], Cisco [47] and Microsoft [48].

Even if the names of the individual levels of technical design (see lower half of Tab. 1) vary slightly, the described architectures are similar. At the lowest level, measured variables, frequencies and the sensors suitable for monitoring must first be determined. The chosen sensors should follow the components and error patterns to be monitored and discovered. The selection of the interfaces for connecting the sensors are addressed in the Connectivity or so-called Gateway or Communication Level. Those Industrial interfaces and bus systems such as IO-Link, PROFINET or PROFIBUS are widely used. However, open interfaces from the consumer area such as I²C or SPI are also being used increasingly in the development of low-cost embedded systems. The selection of a suitable interface from the sensor to the embedded system is of high relevance since in the following step it has to be examined how a data transformation and decentralized data preprocessing can be carried out. This is needed in order to reduce the data volume, to pass only the selected and extracted characteristics (features) and not all measured data to a central cloud. Layers describing those functionalities are middleware, event processing, aggregation, data abstraction & accumulation, transformation, edge computing etc. Concepts of using edge computing reduce data volume, save computing power in the cloud and allow real-time data analysis. In the central cloud, data from various sources, e.g. sensor data from various measuring points or fault messages, is stored, aggregated and abstracted with one another to enable multivariate data analysis. Finally, the top layers of the reference architectures describe an integration into the business processes, e.g. the integration of machine learning models within the process organization of the maintenance division.

While the described layers of the architectures are very similar, their focus and intention vary Tab. 1 gives an overview on how the product and life cycle, data security, industry neutrality and the theoretical and implementation-oriented level of detail are addressed. Furthermore, an implementation recommendation has been evaluated. However, despite a high theoretical level of detail, most of the architecture models lack implementation recommendations since they have no user-oriented hard- and software solutions. Due to the implementation orientation of the reference architecture of Microsoft and their available IoT-stack, the Azure Cloud was used to set up an own architecture for predictive maintenance.

With Azure Internet of Things services, Microsoft offers a platform for implementing IoT projects. The platform offers a scalable connection, storage, analysis and operationalization of device data. An overview of the overall architecture can be found in [48]. The architecture is very similar to the three-tier IIoT system architecture shown and described in the IIRA [3]. The layers can be found in Fig. 2. The first layer is "Device Connectivity" where IP capable devices can communicate with the cloud

<table>
<tr><th>Model</th><th>RAMI 4.0</th><th>IIRA</th><th>ARM</th><th>Reference Archi-tecture for IoT</th><th>IoT Reference Architecture</th><th>IoT Reference Model</th><th>Azure IoT Reference Architecture</th></tr>
<tr><td>Institution</td><td>Plattform Indus-trie 4.0 & ZVEI</td><td>Industrial Internet Consortium (IIC)</td><td>Research Project IoT-A</td><td>WSO2</td><td>IAAS</td><td>Cisco Systems</td><td>Microsoft</td></tr>
<tr><td>Product and life cycle</td><td>●</td><td>◕</td><td>◑</td><td>○</td><td>○</td><td>◑</td><td>◔</td></tr>
<tr><td>Data security</td><td>●</td><td>●</td><td>●</td><td>◑</td><td>○</td><td>◑</td><td>●</td></tr>
<tr><td>Industry neutrality</td><td>◑</td><td>●</td><td>◔</td><td>◕</td><td>●</td><td>◕</td><td>◑</td></tr>
<tr><td>Theoretical level of detail</td><td>◑</td><td>◑</td><td>●</td><td>◔</td><td>◕</td><td>◑</td><td>◑</td></tr>
<tr><td>Implementation-oriented level of detail</td><td>○</td><td>◔</td><td>○</td><td>◔</td><td>◔</td><td>◕</td><td>●</td></tr>
<tr><td>Implementation recommendations</td><td>◔</td><td>◑</td><td>◕</td><td>◔</td><td>◔</td><td>◑</td><td>●</td></tr>
<tr><td rowspan="10">Layers</td><td rowspan="2">Business</td><td rowspan="2">Business</td><td rowspan="2">Application</td><td rowspan="2">Web/Portal, Dashboad, API Management</td><td rowspan="2">Application</td><td>Collaboration & Processes</td><td>Business Integration</td></tr>
<tr><td rowspan="2">Application</td><td>UI & Reporting</td></tr>
<tr><td rowspan="3">Functional</td><td rowspan="3">Operations, Applications</td><td rowspan="4">Service Organisation, IoT Process Management, Virtual Entity, IoT Service</td><td rowspan="3">Event Processing and Analytics</td><td rowspan="4">IoT Integration Middleware (IoTIM)</td><td>Machine Learning</td></tr>
<tr><td>Data Abstraction</td><td>Stream Processing</td></tr>
<tr><td>Data Accumulation</td><td>Storage</td></tr>
<tr><td>Information</td><td>Information</td><td>Aggregation/ Bus Layer</td><td>Edge (Fog) Computing</td><td>Data Transformation</td></tr>
<tr><td>Communication</td><td>Connectivity</td><td rowspan="3">Communictaion</td><td rowspan="3">Communications</td><td>Gateway</td><td>Connectivity</td><td>Cloud Gateway (IoT Hub)</td></tr>
<tr><td rowspan="2">Integration</td><td>Control</td><td>Device</td><td rowspan="3">Physical Devices & Controllers</td><td rowspan="3">IoT (Edge) Devices</td></tr>
<tr><td>Sense, Actuation</td><td rowspan="2">Sensor & Actuator</td></tr>
<tr><td>Asset</td><td>Physical Systems</td><td>Device</td><td>Devices</td></tr>
</table>

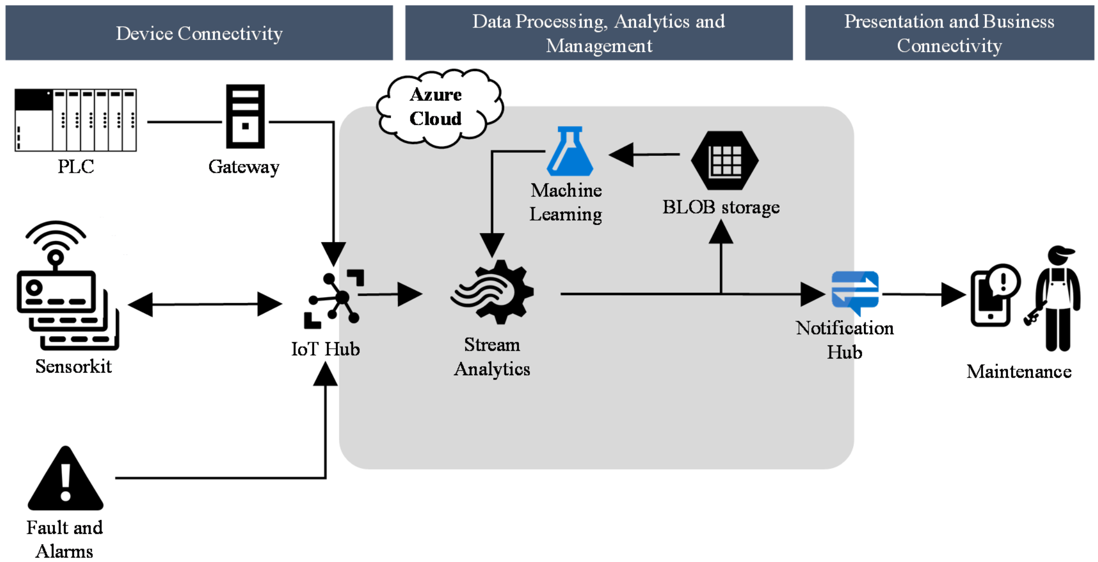

Figure 2. Implemented IIoT-architecture for predictive maintenance on Microsoft Azure.

gateway via an IoT client. The cloud gateway is the entry point of the "Data Processing, Analytics and Management" layer containing a Storage, Stream Processor, Analytics & Machine Learning, Business Integration Connectors and Gateway components. The third layer is "Presentation & Business Connectivity" for integrating personal mobile devices and business systems. The components set up and used of the Microsoft Azure Cloud are the IoT Hub, the Stream Analytics Job, a BLOB storage, Microsoft Machine Learning and a Notification Hub (see Fig. 2). The IoT Hub is a service for managing large data streams of numerous IoT devices. It is the entry point towards the Azure Cloud and makes these data available for other services. Within the project, 60 Sensorkits, the PLC and the IPS-T (system of the BMW Group for recording faults and interruptions in production) were connected. The IoT Hub sends the data to the Stream Analytics Job that collects and transforms all transmitted raw data. Here, the three different data points were joined together, and the JSON-Raw format was transferred into a wide table in csv. The Stream Analytics Job then puts the data into several sinks such as a BLOB-container (Binary Large Objects) for storing and a Microsoft Machine Learning function for analyzing. Once a model is trained, it can be deployed via a web host in the Stream Analytics. Thus, a real time, hot path analysis was implemented. Furthermore, when the model is detecting a fault, a notification via the Notification Hub can be sent to the maintenance. The following section gives an overview of the ML models that were used.

V. Machine Learning

Machine Learning (ML) is a subset field of Artificial Intelligence (AI). Currently, it is regarded as the most effective way of implementing AI [49, 50]. In general, Machine Learning allows computers to learn and identify patterns inside data and use this learned knowledge to perform different operations such as clustering, classification, prediction and description of known or unknown datasets [51]. ML is not a new concept, as there were surges in popularity and research in the past through breakthroughs and invention of Support Vector Machines (SVM), Decision Trees (DT) and Neural Networks (NN) [52]. Most recently, more complex and demanding methods of machine learning, such as Deep Learning and Reinforcement Learning have been in the forefront of research [50, 53].

Predictive maintenance is one application of machine learning that is popular in the environment of production. Its strategies focus on predicting both part degradation and equipment failure through sophisticated models, trained on historical data [54, 55].

Modern implementations include IoT devices that quantify the machines status by collecting sensor data as described above. Common sensors measure temperature, pressure, acceleration, positional data and more. With the help of those sensors, a valid and complete information of the machine health and status is given. This can lead to sophisticated models [56, 57]. Research and the use of big data or smart data have greatly improved the performance of machine learning models [58]. In the next section, it will be shown how IIoT and machine learning for predictive maintenance can be applied.

VI. Case Study

The case study was conducted using the CRISP-DM framework [59], as it fits well on the industrial environment and integrates the deployment into a structural project procedure (see Fig. 3). It provides a framework and structure for the Data-Mining process by organizing it in six phases (Business-Understanding, Data-Understanding, Data Preparation, Modeling, Evaluation and Deployment) [59]. The case study is organized and described accordingly and is concluded with a recommendation.

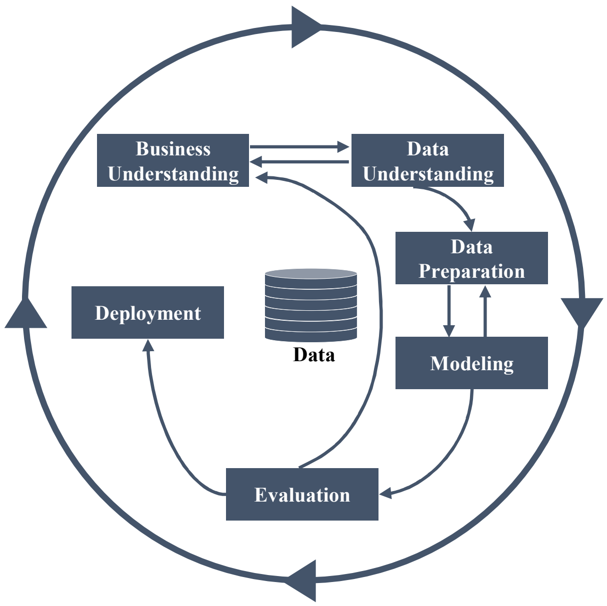

Figure 3. Cross-instustry standard process for data minig (CRISP-DM) [53]

A. *Business- and Data-Understanding*

The proposed cyber-physical system has been applied to a heavy lift EMS at the BMW Group. The conveyor was put into operation in 2003 and consists of a rail with 60 identical cranes which are chained up in a cycle. One section can be seen in Fig. 4. The cranes transport vehicle chassis through the assembly line in which workers carry out assembly operations. Due to the interlinking of the assembly line, the conveyor operates on a critical path since a machine failure would lead to a breakdown of the entire production. The current maintenance strategy is reactive and preventive by visual and auditory inspections. In addition, fixed maintenance intervals for certain components like the vulkollan wheels and electric motors are carried out. Since the conveyor provides neither connectivity nor sensors, a retrofit solution was needed. The main goal was to implement a health / condition monitoring system first, and then extending it to a predictive maintenance strategy. Therefore, the proposed CPS including an IIoT-architecture and the Sensorkit based on a BananaPi were installed at all 60 cranes. After analyzing the most critical components with the Failure Mode and Effects Analysis (FMEA), each Sensorkit was equipped with two acceleration sensors for the wheels, three inductive current clamps for the electric motors (two drive motors and one lift motor) and three temperature sensors (two for the drive motors and one for the local control cabinet). The average data streamed per hour was about 1.2 GB resulting from a combined output of 480 sensors at a frequency of up to 10 Hz. The streaming service had to handle ~2.4 million messages per hour and, furthermore, had to join the sensor-data from the IIoT-device with the data from the PLC-controllers in soft real-time. The sensor values alone do not give much insight. Therefore, the sensor values have to be interlinked with process information like the current location of the crane.

Figure 4. Heavy lift EMS at BMW Group

Then, a comparison between the cranes and a first anomaly detection can be carried out. This was done by collecting the interchanging messages between the main PLC and each local PLC (at each crane) and sending them to the IIoT-Cloud. These messages contain valuable information like the current location of each crane and the lifted car model. The current location of each crane is measured by the local PLC via a closed-circuit barcode, which is separated into approximately 60 blocks. At each block, the crane operates differently like accelerating, decelerating or lifting the car up or down.

B. *Data Preparation*

The data originates from two major, distinct sources which are forwarded to the IIoT cloud: first, the described Sensorkits and second, the PLC of the EMS. The Sensorkits provide the sensor readings while the PLC holds the contextual data about the production process. As described in section IV. IIoT-architecture, the data is sent to a cloud storage. As the data is not in an optimized state, it needs to be prepared for being used in machine learning applications. The complete preparation is performed using the Pandas Python library and a Jupyter Notebook environment on a virtual Machine. While the PLC data is already in a wide format, the sensor data is not and must therefore be pivoted before being used in most machine learning algorithms. After converting the sensor data into a wide format, both datasets are merged.

Next, the data is cleaned of outliers. This is a difficult task as it is important to avoid cleaning anomaly data. Through statistical thresholds, anomalies were determined and evaluated in cooperation with domain experts. Outliers that could clearly be identified as not being anomalies were then eliminated. After cleaning the data from outliers, a feature extraction was performed to generate powerful characteristics. The data was aggregated over short periods of time to extract statistical features such as minima, maxima, mean, standard deviation, variance and gradients. This results in 130 extracted features. The included features, such as the variance of the power, have already been used successfully for anomaly detection and predictive maintenance in production machines [60].

In live prediction, this extraction is accomplished by using a windowing utility, aggregating the data within the specified window. As not all features provide valuable information about the state of the machine and contain a large amount of redundancy, several feature selection methods were applied. Besides the recursive feature elimination provided by Decision Trees and Support Vector Machines, one ensemble feature selection method and two unsupervised methods were employed. The ensemble feature elimination was conducted using Scikit-learn's RFECV module. It used Decision Tree, Random Forest, Linear Regression and Support Vector Machines to rank all features with their respective RFECV. The feature rankings were averaged, and a new ranking based on the accumulated average feature importance score was generated. Especially when using algorithms that do not have an own feature selection function, using a single ranking from a recursive feature elimination with e.g. a decision tree would introduce a bias toward its selection technique. With this ensemble feature selection, the ranking has less bias and at the same time is more robust to errors in the selection process of a single algorithm [61]. The number of acknowledged unsupervised feature selection algorithms is very sparse. Most algorithms use a kind of similarity measure to select features by correlation and redundancy. The aforementioned Laplace score and spectral embedded analysis are the best algorithms researched and tested of the category [62, 63]. The unsupervised feature selection methods were based on the spectral embedded analysis and the Laplace Score [64] and implemented using the scikit-feature utility [65]. The resulting feature rankings from all selection techniques had in common that they prioritized both acceleration and energy current data. Laplace score and spectral embedded analysis both returned almost identical rankings. This may be contributed to the fact that the spectral embedded analysis is based on the Laplace score [64].

For the live deployment, the data preparation steps were copied and implemented in an Azure Stream Analytics Job in order to reach the required performance (Fig. 5). The architecture was scalable and could handle even more data for this use case, although the performance implied that the machine learning applications would possibly create a bottleneck in the pipeline.

C. *Modeling*

Through transforming the data enables, it can be used in analytics and modeling. The model creation can be accomplished by using statistical or machine learning methods. While statistical methods have a longer history, the recent up rise in machine learning has led to new and powerful algorithms for industrial use cases [66, 67]. In order to determine the best suitable algorithms for the task of anomaly detection and fault classification, a subset of the available algorithms from the Scikit-Learn library was chosen to be applied to the use case [68].

For testing, two datasets were sampled from historical data. The first dataset only contains normal data and is randomly sampled over time (~70.000 instances). The

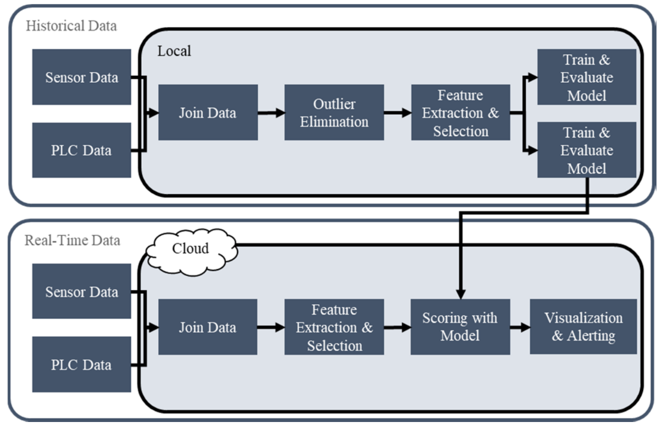

Figure 5. Data flow and processing

second dataset contains only fault data that was collected during tests and from real-world defects (~17.000 instances). Failure data is very scarce and costly to acquire on electric conveyors. Therefore, although there may be access to labeled data in testing, a real-life implementation of predictive maintenance would initially lead to an unsupervised or semi-supervised method in order to be effective.

As semi-supervised algorithms mainly address anomaly detection, they were evaluated on their performance on dividing normal data from fault data [69, 70]. The selected semi-supervised algorithms are:

- One-Class Support Vector Machine [66]
- Isolation Forest [71]
- Local Outlier Factor [55]

These algorithms were chosen for their use in either anomaly detection or outlier detection [66, 67]. The semi-supervised models were created using only data from the normal dataset. They were then tested by using a combined dataset of normal and fault data to calculate accuracy, precision and F1-Score. The normal data used for training and for testing is distinct. No differentiation was made between fault types and all faults were labeled as "1" (anomaly), while all normal data was labeled as "0" (normal). After training, the models were evaluated using the test dataset and are scored on accuracy, AUC and F1-Score for predicting anomalous data.

For the evaluation, the F1-Score played a major role as it represents the harmonic average of the precision and recall. The *recall* is calculated by the proportion of true positive detected anomalies and the number of real anomalies, including false negative predicted values. The *precision* is calculated by the proportion of true positives and all true predicted values, also including false positive predicted values [72]. In order to acquire an unbiased, optimal parametrization, all tested algorithms were constructed using a grid-search method over the parameters offered by the scikit-learn API [68]. Considering all semi-supervised models, the Isolation Forest performed best reaching an F1-Score of up to 89 % when predicting anomalies (Fig. 6.). The One-class SVM also performs very well, scoring only slightly worse than the Isolation Forest.

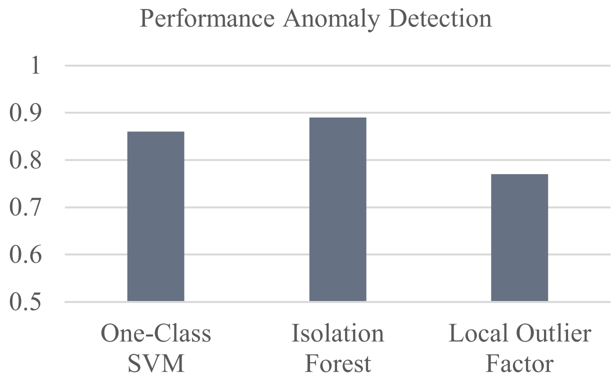

Figure 6. Performance of semi-supervised algorithms by F1-Score

The Local Outlier Factor also returns fair results but does not reach the performance of either the Isolation Forest and the One-class SVM. In terms of computation time, the Isolation Forest was the fastest in both training and prediction. The One-class SVM was slightly slower, having an 86% higher training time and a 38 % higher prediction time. The Local Outlier Factor cannot directly handle prediction from a stored model. Therefore, it predicts outcomes directly without a prescribed training phase. The prediction time was 3 % higher than the time achieved by the Isolation Forest.

As the semi-supervised models can only determine whether an instance was fault or normal data, unsupervised models were deployed to cluster the faults into groups. The selected unsupervised algorithms were:

- K-Means
- Mean shift clustering
- Spectral clustering
- Agglomerative clustering
- DBSCAN

All unsupervised algorithms employed use clustering. Unsupervised models were used directly to cluster the test dataset and were able to reach an F1-Score of up to 70 % for anomaly detection, thereby performing much worse than the semi-supervised models. Differentiation between faults was possible when excluding normal data from the training phase. K-means clustering and Agglomerative clustering both performed similarly well. Mean shift clustering and spectral embedded clustering has a lower performance, especially when differentiating faults. DBSCAN performed badly and did not create useful results.

In combination with a semi-supervised anomaly detection, it was possible to create datasets of clustered faults. These datasets enabled the use of supervised methods. The selected supervised machine learning algorithms were:

- Decision Tree (with variations)
- Random Forest
- Support Vector Machine (SVM)
- Logistic Regression
- k-Nearest Neighbors
- Naïve Bayes
- Quadratic Discriminant Analysis (QDA)
- Neural Nets (Multilayer perceptron, MLP)

Supervised models were created using a normal dataset and a fault data set. The fault dataset contained three types of discreetly labeled fault data. All models were trained and tested with cross-validation and balanced data.

A selection of relevant parameters and according parameter values was created for each algorithm. Many algorithms were able to achieve F1-Scores over 90 %, as seen in Fig. 7. In the category of supervised algorithms, the random forest performed best, reaching F1-scores of 98.1 %. Decision Trees reached 97.3 % and the k-Nearest Neighbors algorithm reached F1-Scores of up to 97.4 %. Decision Trees, Random Forests and SVMs processed the scoring very fast, while the Logistic Regression and k-Nearest Neighbors Algorithm were relatively slow. The parameters used for the final models can be seen in table II.

D. Evaluation

The results of the modeling phase show clear advantages and disadvantages of the applied algorithms. Naturally, the different categories of algorithms are geared towards performing different tasks. Nevertheless, the study aimed to evaluate which algorithms were applicable in predictive maintenance scenarios based on different environments and preconditions. Overall, the supervised algorithms performed best when it comes to classification, which also includes anomaly detection. They are not only able to distinguish between normal and fault data, but also to correctly classify the single faults. However, they require labeled fault data, which is rare in the industrial context. Also, previously unknown faults would be misclassified. The Isolation Forest performs better than the One-class SVM in anomaly detection, however, the SVM does prove to be more robust. Both perform well for live anomaly detection. On the other hand, the Local Outlier Factor is infeasible for live detection due to the

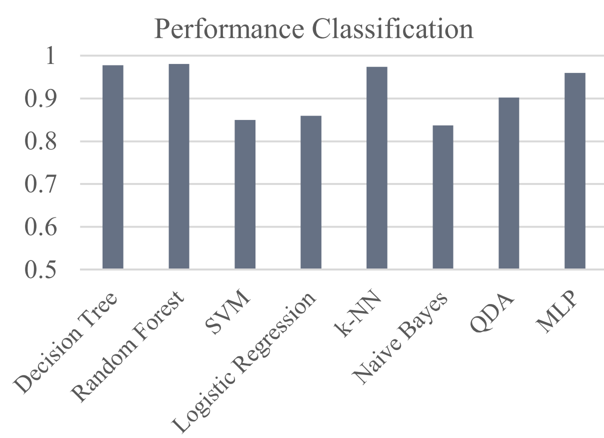

Figure 7. Performance of supervised algorithms by F1-Score

TABLE II. BEST PARAMETERS FOR ALGORITHMS

OSVM	K-Means	RF
Nu = 0.1	Clusters = 3	Estimators = 10
Gamma = 0.1	Init = random	Criterion = gini
Kernel = polynomious	Tolerance = 0.1	Max_depth = None
Shrinking = True	N_init = 300	Min_samples_split = 2
Coef0 = 0	-	Min_samples_leaf = 1
F. selection = Spectral	F. selection = Spectral	F. selection = ensemble
Training Data: 20.000	Training Data: 20.000	Balanced Training Data: 5 x 1.320
Testing Data: 13:000	Testing Data: 13:000	Balanced Testing Data: 5 x 330

computational requirements. The unsupervised models perform worse on anomaly detection than the semi-supervised models but, to a certain degree, are able to cluster different types of faults. The results lead to the indication that detecting anomalies with a semi-supervised model such as the One-class SVM or the Isolation Forest is feasible, as they perform well. K-means clustering, and agglomerative clustering only provide relative value in fault classification on purely fault data. In contrast, supervised models are the best choice under the condition that labeled fault data is available.

E. Deployment

The machine learning models were implemented with the help of the IIoT-architecture set up for the application at the BMW Group. In the course of this, data preprocessing was automated and scoring processes of the various models were introduced based on the modeling. This enables a real-time comparison of the supervised, unsupervised and semi-supervised machine learning models regarding their performance for fault classification, anomaly detection and real-time prediction capability. The results are shown in Fig. 8. The maintenance operators use dashboards on smart devices and personal computers. This allows the user to have access to the current state of the EMS with the aid of condition monitoring, fault classification and predictions of the machine learning models. The deployment made it possible to detect several motors in an advanced state of wear and several defective installations within the control cabinet. Both components are very critical since a defect during operation would cause a downtime of approximately one hour per motor on the EMS and the whole assembly. Through the implementation of condition monitoring and anomaly detection, maintenance operators are now able to detect defective monorails and components and reject them prematurely.

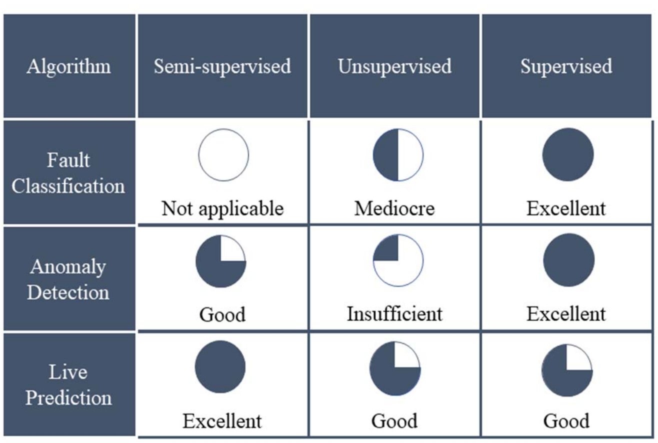

Figure 8. Comparison-Matrix of different machine learning categories

VII. CONCLUSION

This paper shows an approach to enable predictive maintenance for production machines by retrofitting using low-cost sensors, an IIoT-architecture and machine learning. It describes how low cost hard- and software can be used to build up a CPS for implementing condition monitoring as well as machine learning. Furthermore, the CPS presented is flexible, scalable and universal and can therefore be applied to all kinds of production machines. It contributes to a significant increase of the OEE, especially within the brownfield. This was successfully shown by an industrial implementation on an EMS at the BMW Group. In part of machine learning, the recommendation of this case study is to start by using a semi-supervised model to detect and store anomalous data. Once enough fault data has been gathered, a clustering method can be trained and inserted into the predictive maintenance process after the anomaly detection model. Thereby only clustering the anomalous data marked by the anomaly detection model. The results, with input from domain experts, would lead to a dataset of detected anomalies that can be classified and optimally used as a basis for supervised classification of faults. Further research focuses are the standardization of the IIoT-architectures and use cases as well as the integration into existing maintenance systems and upstream- and downstream processes, such as the system ramp-up [73]. Another future challenge will be maintaining the models and architecture and expanding their utility for further applications and the production system's life cycle.

VIII. REFERENCES

[1] H. Kagermann, J. Helbig, A. Hellinger, and W. Wahlster, *Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0: Deutschlands Zukunft als Produktionsstandort sichern; Abschlussbericht des Arbeitskreises Industrie 4.0*: Forschungsunion, 2013.

[2] M. Eickelmann, M. Wiegand, B. Konrad, and J. Deuse, "Die Bedeutung von Data-Mining im Kontext von Industrie 4.0," *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*, vol. 110, no. 11, pp. 738–743, 2015.

[3] S.-W. Lin *et al.,* "The Industrial Internet of Things Volume G1: Reference Architecture," *Industrial Internet Consortium (IIC), Tech. Rep*, 2017.

[4] L. Monostori *et al.,* "Cyber-physical systems in manufacturing," *CIRP Annals - Manufacturing Technology*, vol. 65, no. 2, pp. 621–641, 2016.

[5] acatech, Ed., “acatech POSITION: Smart Maintenance für Smart Factories,” Mit intelligenter Instandhaltung die Industrie 4.0 vorantreiben, 2015.

[6] T. Erwin, P. Heidekamp, and A. Pols, “Mit Daten Werte schaffen,” 2016.

[7] E. Maleki, F. Belkadi, M. Ritou, and A. Bernard, “A Tailored Ontology Supporting Sensor Implementation for the Maintenance of Industrial Machines,” (English), *Sensors*, vol. 17, no. 9, 2017.

[8] D. Kwon, M. R. Hodkiewicz, J. Fans, T. Shibutani, and M. G. Pecht, “IoT-Based Prognostics and Systems Health Management for Industrial Applications,” (English), *IEEE Access*, vol. 4, pp. 3659–3670, 2016.

[9] H. M. Hashemian and W. C. Bean, “State-of-the-Art Predictive Maintenance Techniques*,” *IEEE Trans. Instrum. Meas.*, vol. 60, no. 10, pp. 3480–3492, 2011.

[10] G. Güntner, R. Eckhoff, and M. Mark, “Bedürfnisse, Anforderungen und Trends in der Instandhaltung 4.0,” 2014.

[11] E. Uhlmann, A. Laghmouchi, C. Geisert, and E. Hohwieler, “Decentralized Data Analytics for Maintenance in Industrie 4.0,” *Procedia Manufacturing*, vol. 11, pp. 1120–1126, 2017.

[12] B. Klimm, C. Berger, and D. Kapec, “Offene und intelligente Dienste für die Instandhaltung und Qualitätssicherung,” *productivity*, no. 22, pp. 17–20, 2017.

[13] C. Snijders, U. Matzat, and U.-D. Reips, “"Big Data": Big gaps of knowledge in the field of internet science,” *International Journal of Internet Science*, vol. 7, no. 1, pp. 1–5, 2012.

[14] B. Lydon, “Production line and machine upgrades: Upgrades improve productivity and efficiency,” *InTech Magazine*, no. Jul - Aug, pp. 22–25, 2014.

[15] K. Lorentzen, “Herausforderungen in der Ausgestaltung von Industrie-4.0-Lösungen: Interface-Design zwischen Kunde und Provider,” in *Erfolg durch Lean Smart Maintenance: Bausteine und Wege des Wandels, 31. Instandhaltungs-Forum*, H. Biedermann, Ed., Köln: TÜV Media GmbH TÜV Rheinland Group, 2017, pp. 65–82.

[16] R. Hopkins and K. Jenkins, *Eating the IT elephant: Moving from greenfield development to brownfield*: Addison-Wesley Professional, 2008.

[17] S. Michel and S. Käfer, *Mit Retrofit zu Industrie 4.0.* [Online] Available: https://www.maschinenmarkt.vogel.de/mit-retrofit-zu-industrie-40-a-663312/index2.html. Accessed on: Nov. 25 2017.

[18] W. Wilson and G. Atkinson, “Wireless sensing opportunities for aerospace applications,” in *Sensor Networks and Wireless Sensor Networks*, International Frequency Sensor Association Publishing, Ed., 2008, pp. 83–90.

[19] P. Wright, D. Dornfeld, and N. Ota, “Condition monitoring in end-milling using wireless sensor networks (WSNs),” 2008.

[20] E. A. Lee, “Cyber physical systems: Design challenges,” 2008.

[21] R. Heinze, C. Manzei, and L. Schleupner, *Industrie 4.0 im internationalen Kontext: Kernkonzepte, Ergebnisse, Trends*: Beuth Verlag, 2017.

[22] A. Lanfer and A. Trautmann, “Der Lebenszyklus eines Internet der Dinge Materialflusssystems: Umbau und Modernisierung,” in *Internet der Dinge in der Intralogistik*: Springer, 2010, pp. 237–247.

[23] Horn, Christian, and Jörg Krüger, “A retrofitting concept for integration of machinery with legacy interfaces into cloud manufacturing architectures,” *IEEE Control, Automation and Systems (ICCAS), 2016 16th International Conference on*, pp. 350–352, 2016.

[24] A. Barthelmey, D. Störkle, B. Kuhlenkötter, and J. Deuse, “Cyber physical systems for life cycle continuous technical documentation of manufacturing facilities,” *Procedia CIRP*, vol. 17, pp. 207–211, 2014.

[25] A. Beu, A. Menschig, A. Miclaus, C. Neuy, and C. Rupp, “Industrielle Apps für den Mittelstand erleichtern Industrie-4.0-Anwendungen,” 2018.

[26] A. Albers, B. Gladysz, T. Pinner, V. Butenko, and T. Stürmlinger, “Procedure for defining the system of objectives in the initial phase of an industry 4.0 project focusing on intelligent quality control systems,” *Procedia CIRP*, vol. 52, pp. 262–267, 2016.

[27] K. Wenzel, A. Singer, S. Struwe, and T. Lutze, “Modulare Assistenzsysteme für heterogene Produktionsumgebungen–CyProAssist,” *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*, vol. 112, no. 3, pp. 177–181, 2017.

[28] Fraunhofer IFF Magdeburg, “Tagungsband: Anlagenbau der Zukunft 2016,”

[29] F. Civerchia *et al.,* “Industrial internet of things monitoring solution for advanced predictive maintenance applications,” *Journal of Industrial Information Integration*, vol. 7, pp. 4–12, 2017.

[30] K. Das, P. Zand, and P. Havinga, “Industrial wireless monitoring with energy-harvesting devices,” *IEEE internet computing*, vol. 21, no. 1, pp. 12–20, 2017.

[31] L. Swathy and L. Abraham, “Vibration Monitoring Using MEMS Digital Accelerometer with ATmega and LabVIEW Interface for Space Application,” *IJISET Y International Journal of Innovative Science, Engineering & Technology*, vol. 1, no. 5, pp. 1–5, 2014.

[32] S. B. Chaudhury, M. Sengupta, and K. Mukherjee, “Vibration monitoring of rotating machines using MEMS accelerometer,” *International Journal of Scientific Engineering and Research*, vol. 2, no. 9, pp. 4–11, 2014.

[33] A. Laghmouchi, E. Hohwieler, C. Geisert, and E. Uhlmann, “Intelligent configuration of condition monitoring algorithms,” *Proceedings of 5th WGP*, 1–8, 2015.

[34] N. Suresh, E. Balaji, K. J. Anto, and J. Jenith, “Raspberry PI based liquid flow monitoring and control,” *IJRET*, vol. 3, no. 07, pp. 122–125, 2014.

[35] M. V. García, F. Pérez, and I. Calvo, “Building Industrial CPS with the IEC 61499 Standard on Low-cost Hardware Platforms,” (eng), *IEEE Emerging Technology and Factory Automation (ETFA)*, pp. 1–4, http://ieeexplore.ieee.org/servlet/opac?punumber=6994138, 2014.

[36] E. Uhlmann, A. Laghmouchi, E. Hohwieler, and C. Geisert, “Condition monitoring in the cloud,” *Procedia CIRP*, pp. 53–57, 2015.

[37] S. Tedeschi, J. Mehnen, N. Tapoglou, and R. Roy, “Secure IoT Devices for the Maintenance of Machine Tools,” *Procedia CIRP*, vol. 59, pp. 150–155, 2017.

[38] J. Wang, L. Zhang, L. Duan, and R. X. Gao, “A new paradigm of cloud-based predictive maintenance for intelligent manufacturing,” *J Intell Manuf*, vol. 28, no. 5, pp. 1125–1137, 2017.

[39] P. Strauß, R. Wöstmann, and J. Deuse, “Predictive Maintenance: Entwicklung eines cyber-physischen Predictive-Maintenance-Systems basierend auf einem Low-Cost-Sensorkit und Data Analytics,” in *Erfolg durch Lean Smart Maintenance: Bausteine und Wege des Wandels, 31. Instandhaltungs-Forum*, H. Biedermann, Ed., Köln: TÜV Media GmbH TÜV Rheinland Group, 2017, pp. 121–142.

[40] M. Stockinger, *Untersuchung von Methoden zur Zustandsüberwachung von Werkzeugmaschinenachsen mit Kugelgewindetrieb*: Shaker, 2010.

[41] Fraunhofer IOSB, *Auswertung der Nutzeranforderungen zu IoT-Adaptern.* Accessed on: Aug. 22 2017.

[42] R. Wöstmann, P. Strauß, and J. Deuse, “Predictive Maintenance in der Produktion: Anwendungsfälle und Einführungsvoraussetzungen zur Erschließung ungenutzter Potentiale,” *wt Werkstattstechnik online*, vol. 107, no. 7/8, pp. 524–529, 2017.

[43] E. V. BITKOM, E. V. VDMA, and E. V. ZVEI, “Umsetzungsstrategie Industrie 4.0-Ergebnisbericht der Plattform Industrie 4.0,” *Berlin, Frankfurt am Main*, pp. 1–98, 2015.

[44] A. Pastor, “Internet of Things Architecture IoT-A: Project Deliverable D6. 2-Updated Requirements List,” 2012.

[45] P. Fremantle, “A reference architecture for the internet of things,” *WSO2 White paper*, 2014.

[46] J. Guth *et al.,* "A Detailed Analysis of IoT Platform Architectures: Concepts, Similarities, and Differences," in *Internet of Everything*: Springer, 2018, pp. 81–101.
[47] J. Green, "The Internet of Things Reference Model," 2014.
[48] Microsoft, *Microsoft Azure IoT Reference Architecture.* [Online] Available: https://azure.microsoft.com/de-de/updates/microsoft-azure-iot-reference-architecture-available/. Accessed on: Oct. 29 2017.
[49] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, 436 EP -, 2015.
[50] I. Goodfellow *et al.,* "Generative Adversarial Nets," 2014.
[51] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*: Morgan Kaufmann, 2016.
[52] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[53] D. Silver *et al.,* "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, 484 EP -, 2016.
[54] H. Faria, J. G. S. Costa, and J. L. M. Olivas, "A review of monitoring methods for predictive maintenance of electric power transformers based on dissolved gas analysis," *Renewable and Sustainable Energy Reviews*, vol. 46, pp. 201–209, 2015.
[55] B. Kroll, D. Schaffranek, S. Schriegel, and O. Niggemann, Eds., *System modeling based on machine learning for anomaly detection and predictive maintenance in industrial plants*. Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA), 2014.
[56] J. Lee, H.-A. Kao, and S. Yang, "Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment," *Procedia CIRP*, vol. 16, pp. 3–8, 2014.
[57] P. K. Behera and B. S. Sahoo, "Leverage of Multiple Predictive Maintenance Technologies in Root Cause Failure Analysis of Critical Machineries," *Procedia Engineering*, vol. 144, pp. 351–359, 2016.
[58] E. Ruschel, E. A. P. Santos, and E. d. F. R. Loures, "Mining Shop-Floor Data for Preventive Maintenance Management: Integrating Probabilistic and Predictive Models," *Procedia Manufacturing*, vol. 11, pp. 1127–1134, 2017.
[59] C. Shearer, "The CRISP-DM model: The new blueprint for data mining," *Journal of data warehousing*, vol. 5, no. 4, pp. 13–22, 2000.
[60] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality reduction for fast similarity search in large time series databases," *Knowledge and information Systems*, vol. 3, no. 3, pp. 263–286, 2001.
[61] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.
[62] A. K. Farahat, A. Ghodsi, and M. S. Kamel, Eds., *An efficient greedy method for unsupervised feature selection*: IEEE, 2011.
[63] M. Mirza, A. Courville, and Y. Bengio, "Generalizable features from unsupervised learning," *arXiv preprint arXiv:1612.03809*, 2016.
[64] U. Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.
[65] J. Li *et al.,* "Feature Selection: A Data Perspective," *arXiv preprint arXiv:1601.07996*, 2016.
[66] M. Amer, M. Goldstein, and S. Abdennadher, "Enhancing One-class Support Vector Machines for Unsupervised Anomaly Detection," in New York, NY, USA, 2013, pp. 8–15.
[67] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.
[68] F. Pedregosa *et al.,* "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.
[69] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, 2009.
[70] K. P. Bennett and A. Demiriz, "Semi-Supervised Support Vector Machines,"
[71] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 413–422.
[72] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *2229-3981*, 2011.
[73] C. H. Glock and E. H. Grosse, "Decision support models for production ramp-up: A systematic literature review," *International Journal of Production Research*, vol. 53, no. 21, pp. 6637–6651, 2014.