A cloud-to-edge architecture for predictive analytics

David Bowden*, Angelo Marguglio†, Lucrezia Morabito‡, Chiara Napione‡, Simone Panicucci‡, Nikolaos Nikolakis§, Sotiris Makris§, Guido Coppo**, Salvatore Andolina**, Alberto Macii††, Enrico Macii‡‡, Niamh O'Mahony*, Paul Becker§§, Sven Jung§§

* DELL EMC, Cork, IRELAND, name.surname@dell.com
† Engineering Ingegneria Informatica S.p.A., Palermo, ITALY, name.surname@eng.it
‡ COMAU S.p.A., Turin, ITALY, name.surname@comau.com
§ Laboratory for Manufacturing Systems & Automation, Department of Mechanical Engineering and Aeronautics, University of Patras, Patras, GREECE, name.surname@lms.mech.upatras.gr
** SynArea Consultants S.r.l., Turin, ITALY, name.surname@synarea.com
†† Department of Control and Computer engineering, Politecnico di Torino, Turin, ITALY, alberto.macii@polito.it
‡‡ Interuniversity Department of Regional and Urban Studies and Planning, Politecnico di Torino, Turin, ITALY, enrico.macii@polito.it
§§ Fraunhofer Gesellschaft zur Förderung der angewandten Forschung, Aachen, Germany, name.surname@ipt.fraunhofer.de

ABSTRACT

Data management and processing to enable predictive analytics in cyber physical systems, holds the promise of creating insight into the underlying processes, discovering criticalities and predicting imminent problems. Hence, proactive strategies can be adopted, with respect to predictive analytics. This paper discusses the design and prototype implementation of a plug-n-play end-to-end cloud architecture, enabling predictive maintenance of industrial equipment. This is enabled by integrating edge gateways, data stores at both the edge and the cloud, and various applications, such as predictive analytics, visualization and scheduling, integrated as services in the cloud system. The proposed approach has been implemented into a prototype and tested in an industrial use case related to the maintenance of a robotic arm.

1 INTRODUCTION

The advent of Industry 4.0 trend in automation and data exchange, leads to a constant evolution towards smart environments, including an intensive utilization of Cyber-Physical System (CPS). This promotes a full integration of manufacturing IT and control systems with physical objects embedded with electronics, software and sensors. This new industrial model leads to a pervasive integration of information and communication technology into productive components, generating massive amounts of data. Powerful and reliable cyber-physical architectures are becoming prominent to effectively analyze such large amounts of data, creating insight into the production process, and, thus, enabling its improvement, as well as competitive business advantages.

This paper presents a cloud architecture designed for the Industry 4.0 vision, bridging the gap between the physical world, which provides raw data, and the cyber space, which processes those data and creates insight, aiming to enable predictive analytics at the edge. With the goal of anticipating failures and estimating the remaining useful life (RUL) of physical equipment, a two-tier data analytics architecture has been developed. This architecture, the "SERENA" system, can identify the symptoms of imminent machine failure, through the characterization of the current dynamics of the process/machine (at any given time) using on-line data collected in the factory. A scalable and modular approach has been taken in the design of the architecture, decoupling the overall design from any specific set of technologies. For testing and validating the proposed approach, a prototype has been implemented, integrating services such as visualization, scheduling and predictive analytics. This prototype was validated in a real-world scenario involving anomaly detection on a robotic axis and concerning the maintenance requirements caused by the backlash effect. The visualization service enables a real-time data stream and machine visualization, while the predictive analytics services generate the estimated RUL value, which is consumed by the scheduling service to proactively schedule the maintenance activities.

This paper is organized as follows. Section 2 discusses the state-of-the-art analytics approaches targeting to Industry 4.0. Section 3 describes the proposed cloud-to-edge architecture for predictive analytics, while Section 4 introduces the industrial use case. Section 5 presents the preliminary results achieved by exploiting the proposed architecture in a real use-case. Finally, Section 6 draws conclusions and discusses future developments of the proposed cloud-to-edge architecture for predictive analytics in Industry 4.0.

2 RELATED WORK

A large variety of studies have been carried out to develop efficient data management systems, data analytics engines, business processes and risk assessment in Industry 4.0. The authors in [15] presented a case study exploiting Big Data analytics to improve production processes. It exploited a methodology, called Cross-Industry Standard Process for Data Mining, to present and organize results for better understanding businesses. The work in [7] presented an integrated self-tuning engine for predictive maintenance in Industry 4.0. Specifically, a distributed architecture, based on Apache Kafka, Spark Streaming, MLlib, and Cassandra was proposed and discussed. The proposed approach integrated the monitoring and prediction tasks, along with a self-tuning approach for the dynamic selection of the best predictive algorithm, and specific attention to providing interpretable knowledge to end users. Manufacturing computerization is another crucial issue to be addressed in the Industry 4.0 ecosystem. The study presented in [11] proposed a semantic reduction of heterogeneous sources, based on Semantic Web approaches, to foster better analytics implementations.

Another interesting issue to address is the increasing amount of data to be managed by machine learning techniques. In this context, an interesting comparison between multi-class classifiers and deep learning techniques is discussed in [12]. Furthermore, a comparative experimental analysis of exploratory techniques for Big Data is provided in [3]. The authors in [2] present a Big-Data scalable predictive approach in the energy domain Industry. The study presented in [9] proposed a framework for on-demand remote sensing data analysis to speed up the execution of models by reducing data transfers through the network. This allows for classical remote data service systems to evolve into remote sensing data processing infrastructures.

Advanced Internet of Things (IoT) and ICT technologies allow linking physical manufacturing facilities and machines in integrated applications. The authors in [5] provided a review of virtualized and cloud-based services in the context of manufacturing systems. A predictive maintenance approach involving cyber-physical systems with wide IoT capabilities along with complex event processing features was discussed.

Among the most widespread maintenance approaches, condition based maintenance (CBM) is usually considered the most effective. Efficiently determining the health status of a monitored device, in such context, is of major importance. Prognostics and diagnostics applied to raw data collected from sensors aim to determine the health of the monitored system or equipment. To this end, detecting and analyzing underlying data trends allow anomalies to be discovered. An overview of data analytics techniques for anomaly detection is provided in [13]. The authors exploited artificial neural networks in large systems to effectively predict their health. Prognostics or predictive analytics are usually associated with the computation of a key performance indicator, such as the RUL. The authors in [17] presented a deep-belief network ensemble method with multiple objectives to estimate RUL. Similarly, the authors in [4] exploited a neural-network prognostics model to support industrial maintenance scheduling. The failure probabilities were computed from real-world equipment measurements through a logistic regression approach. Such measurements were then routed to a prognostics model to forecast failure conditions and, finally, to estimate RUL. In this scenario, predictive analytics are affected by the quality of data used for prediction. The authors in [6] proposed a method for improving data quality in diagnosing the health of devices and production equipment. First, a visualization-based grouping, based on the dissimilarity spectrum, was performed on critical measurements, which were then clustered and evaluated, in terms of their fitness and separation with each other. An outlier-detection visual assessment was also presented to identify outliers in the data.

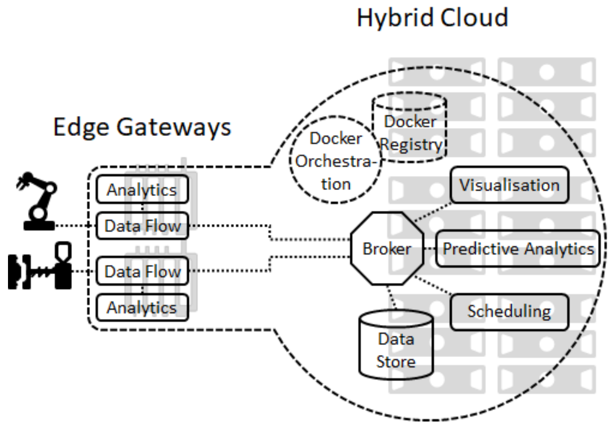

Figure 1: SERENA Architecture

3 THE SERENA APPROACH

The SERENA system comprises a number of services, which collectively provide predictive analytics functionality, enabling predictive maintenance policies to be applied. It is implemented using a light-weight micro-services architecture, which utilizes Docker containers to wrap the services into deployable units. The services can then be distributed across the SERENA hybrid cloud, extending their functionality out to edge gateways on the factory floor. The distribution of services and dynamic communications channels is implemented using a Docker orchestration manager. Wrapping services in containers abstracts them from the underlying host infrastructure. As Docker is a commonly supported open source technology, the SERENA system can be deployed on a wide variety of infrastructures, from hardware servers and gateways, through virtual machines, to hosted environments on public and hybrid clouds. Using the same Docker solution across the SERENA hybrid cloud and gateways, gives the system a unified architecture, which can be operated and managed as a single unit. The services represent logical elements that provide defined functionality in the SERENA system. Whilst the SERENA reference implementation uses specific technology to realize each service, the common interface allows technology to be swapped, depending on the specific implementation requirements. This technology transparency is an important concept in SERENA's plug-n-play architecture. Figure 1 illustrates the main components of the SERENA system and their interactions, which are further described in the following subsections.

3.1 Services

The SERENA system is designed to integrate external applications using a service-oriented approach. In this context, the following services have been designed, implemented and integrated in the above mentioned system:

A predictive analytics service, based on machine learning techniques, to forecast future failures of machinery/equipment. The aim of this service, whose functional building blocks are shown in Figure 2, is two-fold: (i) Building a prediction model, based on historical data, by means of machine learning algorithms; and (ii) applying such a model in real time to new incoming data streams, to identify possible failures. A two-tier architecture exploiting both edge and cloud computing has been proposed to address phase (i) in the cloud, exploiting (theoretically) unlimited resources, while phase (ii), which requires less

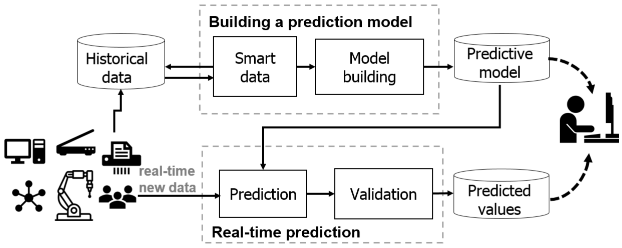

Figure 2: The predictive analytics service: main building blocks

computation resources, runs at the edge due to the limited resources and to increase responsiveness.

The *smart data* block derives relevant static features from the raw data (in many cases raw data are time series), supporting the predictive maintenance goal. Smart data represents the key characteristics of the raw data, as well as context information about how the data was collected and the operating conditions of the equipment it was collected from. In the current implementation, the block computes a large variety of statistical indices, including maximum, minimum, mean, peak to peak distance, variance, inter-quartile ranges, standard deviation, root mean square, kurtosis and skewness.

The *model building* block is executed on a batch schedule on historical data. These data include the smart data computed over the original time series and their corresponding class labels (e.g., failure presence or absence, category of failures). All data are related to an industrial device/robot/piece of equipment of interest that can fail and for which a predictive maintenance strategy should be addressed.

Many classifiers do not manage time series data by design but, since the original time-series of measurements are not considered for training the model, a wide range of classifiers could be used. In the current implementation to train a predictive model, the proposed methodology exploits one of the following machine learning algorithms: Neural networks (NN) [16], Random Forest (RF) [10], Logistic Regression (LR) [10], Support Vector Machines (SVM) [10], and Gradient-Boosted Tree (GBT) [8].

In the *validation* block, performance of the *prediction* block, is evaluated by exploiting either a k-fold stratified cross-validation or a hold-out strategy based on the cardinality of the training set. In addition, the training dataset defaults to the available historical data, even if shorter and more specific periods can be selected to address ad-hoc predictive maintenance issues. The prediction performance is evaluated through quality indices, such as accuracy, to evaluate the overall efficacy of the classifier, whilst f-measure, precision, and recall offer important insights on the performance of the classifier with respect to a given class.

A forecasted failure time horizon is generated as the output of the predictive analytics service and consumed by **a scheduling service** [14]. The aim of the service is to prevent the predicted failure, by assigning the required maintenance activities to operators within the given timeframe. This service can be extended to consider the current production plan, hence fitting the maintenance activities within a given time slot to optimise production outputs.

The **visulization** block provides a 3D view of the relevant machinery/equipment, using the data collected in the field, along with the results of the predictive maintenance algorithms. This service allows for the presentation of the data and maintenance operations to the local technicians or remote experts in an effective and intuitive way.

3.2 Edge gateway

As illustrated on the left of Figure 1 the edge gateways are located on the factory floor and collect sensor data from industrial equipment and channel it, through the **data flow** engine, to the communications **broker** running in the SERENA hybrid cloud. The gateways also host **analytics** models, which are used to process data at the edge, converting raw data into *smart data*. Both the data flow engine and the analytics model are deployed to the gateway as Docker containers, under the control of the Docker orchestration manager. The gateway can host multiple types of data flow engines and analytics models, depending on the types of equipment that are being monitored . The majority of the raw sensor data is transformed into smart data, by the analytics model running on the gateway, but when specific criteria are met, a sample of the raw sensor data is sent to the SERENA cloud for more in-depth analysis and to train the analytical models. Typically, gateways have enough computing power to run analytical models, but not to train them. The training is handled by the predictive analytics service running in the SERENA cloud. The service uses the raw data to train the prediction algorithms and package the resultant models up in a Docker container, which are deployed to the edge gateways.

3.3 Broker

As shown in the middle of Figure 1, the communications broker acts as the central communication hub between the Data Stores, SERENA services and the edge gateways. The broker primarily handles HTTPS traffic by exposing a number of REST endpoints. The broker also supports a number of other protocols, such as MQTT and Web Services, for real-time data transfer. In addition to receiving sensor data directly from the factory floor, the broker acts as the access point for external facilities, such as enterpise resourse planning systems (ERP). Security is a critical part of the SERENA system, and the broker, as the communications hub, provides secure channels to and from the gateways and other SERENA services. It will also validate the authenticity of incoming messages, and whether the requester is authorized to use the requested service.

3.4 Cloud data storage

SERENA supports a number of different data stores, depending on the type of data and its function within the system, including raw sensor data, smart data, metadata, equipment manuals, 3D objects for virtual reality applications, etc. The data stores and data repositories are also implemented as containerized SERENA services, which gives them the same flexibility as other services on the system.

3.5 Docker orchestration

The orchestration manager is responsible for deploying the service containers to the host infrastructure and managing their lifecycle. It also defines and manages the communications channels between services. The core SERENA cloud services are deployed as resilient clusters of Docker containers. If a container fails, the orchestrator automatically starts a new container to replace it. Additionally, the orchestrator can be used to increase or decrease the number of containers in a service, thus scaling the operation of the service.

As the SERENA cloud servers and gateways are registered within the same Docker domain, the orchestrator can manage the deployment of new services (e.g. data flow engines and analytics models) from the SERENA cloud, all the way to the edge gateways. Docker uses labels to specify which containers are deployed to which hosts. If a new data flow engine is required to support a piece of equipment, the appropriate Docker label is defined on the host gateway; the Docker orchestrator will then ensure that the appropriate data flow container is automatically deployed to the gateway. In this way, thousands of containers can be deployed to hundreds of gateways, simply by defining the appropriate Docker labels.

3.6 Docker registry

The SERENA system also implements its own local Docker image registry. Docker containers are deployed from images in the local registry, rather than using a public registry, which ensures that the required images are always available locally, and from a trusted source.

4 USE CASE AND EXPERIMENTAL SETTING

COMAU (https://www.comau.com/) deploys industrial robots around the world and it has an increasing requirement to collect data that monitors the health status of all its machines, in order to avoid sudden failure. To cope with this complexity, further studies on predictive maintenance approaches are needed. For this reason a test-bed has been built, which is called *RobotBox* and consists of a motor from a Comau medium size robot, with its associated controller. Then it is constituted by an adaptor, a belt and a 5 kilos weight, which simulates an end effector. The choice of using a single axis rather than an entire robot is due to the fact that manipulating a robot is very expensive. In addition, in a complete robot there are many factors having impact on the physical conditions of the robot behaviour (e.g. frictions, temperature, vibrations, humidity, etc.). It is difficult to isolate single effects and decouple environment phenomena, especially because the only two monitored signals are the axis position and the current required from the motor to perform the expected cycle. So even noise signals impact on these two time series. Nevertheless, it is possible to extend the knowledge acquired from the single axis to robots with more axes, in order to derive a comprehensive knowledge of the asset health status.

This initial experiment only takes into account position and current, but in the future more parameters will be collected and analyzed.

In a predictive maintenance perspective two possible motor failures have been defined, namely backlash and incorrect belt tensioning. In this study we focus on the belt tensioning issue.

Real data. COMAU collects real data by monitoring a motor from a Comau medium size robot, with its associated controller. The collection phase started on September 2018 up to December 2018, during which a cycle has been collected every 120 seconds. The sampling rate to collect raw data is 2 milliseconds, i.e. the sampling frequency is 500 Hz. Therefore, the total number of monitored cycles is 87,840. A cycle is the sequence of moves which the motor has been coded to perform in loop; in this case the cycle lasts for 24 seconds.

In order to study the belt tensioning phenomenon with a machine learning approach, six levels of tensioning have been defined with the domain expert's help. The dataset collected is not balanced, which means the number of samples for each tensiosing level is different; which slightly complicates the machine learning approach but allowed us to collect more data about the levels most complicate to analyze. Each datapoint consists of all the information provided from the *RobotBox* controller and from the user setting connected to the choice of the belt tensioning degree:

- header information: machine id, program number, cycle start time, cycle time;
- time series data: position and current data, collected with a sampling time of 2 milliseconds for a duration of 24 seconds;
- label: level of belt tensioning.

Smart data. From the current raw data, 12 statistical features have been calculated and used to classify each cycle independently. Smart data include: maximum, minimum, mean, peak to peak distance, variance, standard deviation, root mean square (rms) of raw data, kurtosis, skewness and rms of three types of filter on current data (low pass, band-pass and high pass filter). In previous internal studies these features have proved to be effective to model a current cycle.

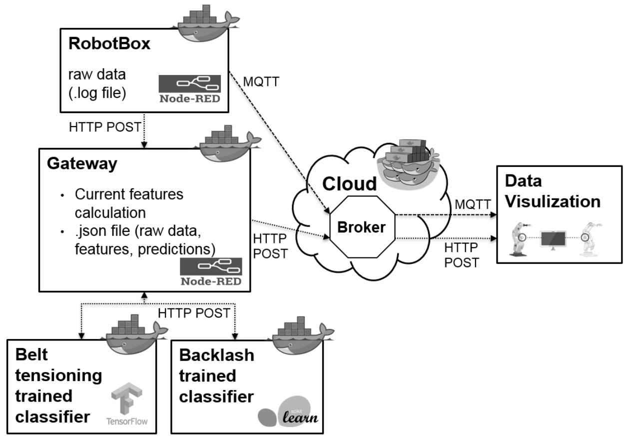

Figure 3: Experimental setting

Experimental setting. In order to implement the first prototype of the proposed architecture, position and current data were acquired by the *RobotBox* controller and transmitted to the Gateway in a JSON format (the .log file), as presented in Figure 3. The Gateway then calculates some statistical features (i.e., smart data) of the current time series. Then, it communicates with two other services, listed below, to obtain classification information about the *RobotBox* cycle:

- a neural network classifier able to recognize the belt tensioning level;
- a classifier capable of giving a qualitative backlash status and an estimation of the remaining useful life expressed in days.

At the end, all the raw data, the current features and the classifiers' outputs are sent to the broker ingestion service running on the cloud.

The use of Node-RED (https://nodered.org/) makes it possible to program each block which implements the required functionalities remotely, since the operator has only to connect or launch the flow. The Node-RED service in the *RobotBox* controller, the two classifiers and the Node-Red flow in the Gateway are all located in Docker containers and the relative images have been created and added to the SERENA Docker Registry.

Big data framework. As a first attempt, the proposed architecture exploits a NoSQL database as a cloud storage layer. However, the current solution is planned to be replaced by Big data framework, exploiting the Cloudera stack. A MongoDB solution has been adopted in the experiment, in order to ease the data management services delivered upon flexible message formats, guaranteeing fast performance on both write/read directions. MongoDB collections and contents have been exposed through HTTP REST endpoints to the SERENA Ingestion Service running on the cloud implemented using Apache NiFi.

HTTP and real time feed (MQTT). A service in a Docker container for almost real time data streaming has been deployed: this is situated in the *RobotBox* controller and it publishes data to a MQTT broker in the cloud, so as to make data available to the visualization application which subscribes to the same topic. The data stream position was used to update a virtual representation of the *RobotBox* and a sample period of 50 milliseconds was chosen as a good trade off between the visualization quality and the bandwidth requested.

5 PRELIMINARY RESULTS

In this section, some preliminary results obtained through the exploitation of the proposed architecture and its provided services are discussed.

5.1 The predictive analytics service

The predictive analytics has been tailored to forecast the belt tensioning level. To this end, a machine learning algorithm has been applied to smart data, in order to recognize a tensioning level, given a new cycle of data. In the current implementation, we exploited the TensorFlow library [1] to implement a Neural Network algorithm. After an in-depth sensitivity analysis, the specific algorithm parameters were set to the following:

- two hidden layers with, respectively, 50 and 25 neurons;
- Adam optimiser with default values;
- cross-entropy loss;
- 100 thousand of epochs.

Given the amount of data available, an hold-out approach was used to divide the dataset into train and test sets, with 75% of the data used for training and the remainder used for test. In both datasets, shuffling has been performed and a batch of size 300 samples for the training set and 100 for the test set have been chosen.

Since the belt tensioning is changed by moving the motor with respect to the adaptor and in order to make experiments reproducible, six washers have been used to discretize the six levels of belt tensionsing taken into account (each washer is 0.2 mm thick). The lower the number of washers, the higher both the belt tensioning and the current cunsumption.

The accuracy of the final model was found to be approximately 94%. Table 1 shows the confusion matrix; both 0 and 1 washers are almost perfectly recognized by the model and it is due to the fact that those classes are easily divisible since the belt is extremely tense and thus the current consumption is different from the other classes. Regarding the other classes, even though the model has good performance, there are more incorrect classifications due to an environment factor: the temperature. In fact we have noticed that the higher the environment temperature, the lower the current consumption of the motor due to lower friction between the motor components. The shift introduced by this phenomenon is quite similar to the one caused by adding or removing a washer; which is why our model has difficulty in identifying the correct class. Future work might consider the temperature as another feature to be considered by the classifier.

		Predicted					
		0	1	2	3	4	5
Actual	0	**1373**	0	0	0	0	0
	1	0	**2145**	0	5	0	0
	2	0	0	**6673**	97	67	32
	3	0	0	36	**3718**	40	219
	4	0	0	51	230	**3302**	293
	5	0	0	0	119	30	**3530**

Table 1: Confusion matrix

5.2 Visualization application

In order to setup the first experiment in the Comau test-bed, SynArea (http://www.synarea.com/) has developed an HTML5 Unity 3D interactive prototype application, to show in a web browser the 3D model of the RobotBox. Furthermore, some interface methods have been implemented to manage the information coming from the SERENA platform. In particular:

- display, in *near real-time* warnings, errors and RUL with different colors highlighting the involved part, to immediately capture the operator's attention, and provide an intuitive indication of the main information to check;
- preventive and predictive textual information to be displayed by selecting the involved part of the *RobotBox*;
- 3D virtual procedure to guide the operator while performing the replacement of the involved part (i.e. an example of operator support);
- subscribing to the defined topic of the MQTT broker in the cloud, used for the data stream, visualize the real-time position on the *RobotBox* 3D model, to enable a remote monitoring of the physical behaviour observed.

Figure 4 shows a screenshot of the HTML5 Unity 3D interactive application showing the Comau *RobotBox* without the associated controller. The central (yellow) element is a 5 kilos weight simulating an end effector, and the highlighted element is a medium size motor of a Comau robot, connected with an adaptor and a belt.

The application is connected to the SERENA cloud platform in order to provide intuitive and real-time information to the maintenance operator, as a result of the analytics and predictive algorithms, and to enable remote monitoring using the position.

The highlighted color on the motor shows its failure status (green = correct; yellow = warning; red = failure) and, by clicking on it, an information box (on the left side) is displayed with some important prognostic or predictive values, such as the label (level of belt tensioning) and RUL (Remaining Useful Life).

By clicking on the "Maintenance Procedure" button, a virtual procedure of the belt replacement and tensioning is also displayed.

5.3 Scheduling application

The scheduling service has been implemented in Java, following a client-server architecture. The service inputs include the monitored equipment, RUL value, maintenance tasks, including

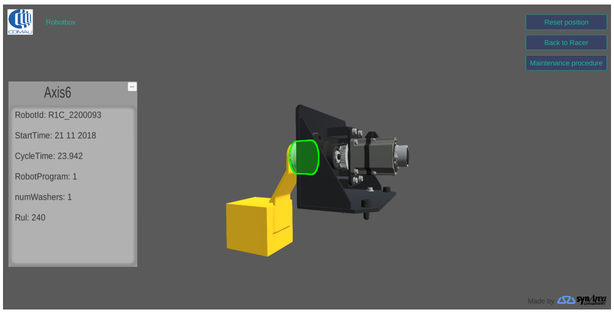

Figure 4: 3D visualization of the RobotBox

Resource Name	Task	Duration (minutes)	Cost (Euros/minute)
Newcomer, Middle	machine inspection	20	0.25
Newcomer, Expert	machine inspection	120	0.25
Expert	machine inspection	15	0.4
Newcomer, Middle	replacement of the gearbox	100	0.4
Newcomer, Expert	replacement of the gearbox	10	0.5
Expert	replacement of the gearbox	80	0.5

Table 2: Information used by the scheduling service for the experiment

precedence relations and default duration per operation experience, and a number of potential operators with their characteristics, such as experience level. The server side includes a multi-criterion decision making framework, evaluating the alternative scheduling configurations, ranking them and selecting the highest ranked one. The client side communicates with the server side via restful APIs, supporting the following functionalities:

- editing of tasks, resources, equipment;
- time series visualisation;
- process plan Gantt visualisation.

The process time required to create a new schedule depends on the complexity of the schedule, referring to the number of tasks, resources, and their dependencies, along with the evaluated criteria. In the current experiment, the schedule was generated in approximately 11 msec, and included the execution of two tasks; machine inspection and replacement of the gearbox, along with three potential resources; (1) a team of one newcomer and one of middle experience, (2) one newcomer and one expert and (3) a team of one expert. The difference in task completion time as well as cost is presented in the Table 2, per task.

6 CONCLUSIONS AND FUTURE APPLICATIONS

This work presents a flexible and scalable architecture merging cloud based and edge deployed components. Through the proposed unified integration and deployment concept, different applications can be enabled with various applications, considering underlying CPS features and under the vision of Industry 4.0 and connected factories. In this regard, the proposed architecture has been designed with the goal of addressing some common needs of industrial enterprises such as:

- compatibility with both the on-premise and the in-the-cloud environments;
- exploitation of reliable and largely supported Big Data platforms;
- virtually unlimited horizontal scalability;
- easy deployment through containerized software modules.

To test the proposed approach, a prototype has been created and validated in an industrial use case on the predictive maintenance of a robotic manipulator, in particular the *RobotBox* device. To enable the evaluation on the basis of predictive analytics, visualization and consequent maintenance planning, three applications have been integrated as services. As a result of the validation of the early prototype, the integrated solution achieved to bridge the gap between machine data acquisition and generation of predictive maintenance policies based on the analysis of the acquired data. Additionally, dynamic allocation of docker containers at the edge was achieved, enabling a dynamic way of allocating functionalities to shop floor equipment, as long as they are connected to the cloud platform and properly labelled. Existing frameworks, such as Arrowhead (http://www.arrowhead.eu/), provide a high-level representation of the underlying architecture without any specification on an end-to-end implementation with a certain set of components addressing some application. This paper provides a reference implementation for a predictive maintenance system using a certain set of components and a specific interaction mechanism. Moreover, the presented implementation is not coupled to any specific technology or technique, thus making it suitable for overlaying other reference architectures, such as Fiware (https://www.fiware.org/). The architecture presented in this work has been focused on flexibility and ease of implementation, extension and deployment. A set of technologies have been used without restricting any user to adopt the same set of technologies, for example, for data storage, or final user services, such as scheduling. As a result, they can be easily substituted, following the proposed integration approach. This will allow the proposed architecture to fit a variety of applications and domains. Hence, the main contribution of this work is providing a set of required end-to-end functionalities for creating a cloud platform for Industry 4.0, not limited to the maintenance domain.

Future activities will focus on integrating additional functionalities to the overall architecture, such as data security features, increasing the robustness of the integrated solution, and evaluating it in versatile use cases with the aim of improving its efficiency and user-friendliness. Moreover, with regards to the data analytics, further investigation and research is required to identify the most appropriate algorithms for enabling data driven predictive analytics and validating their outcome.

ACKNOWLEDGMENT

The research leading to these results has received funding from European Commission under the H2020-IND-CE-2016-17 program, FOF-09-2017, Grant agreement no. 767561 "SERENA" project, VerSatilE plug-and-play platform enabling REmote predictive mainteNAnce.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. (2015). https://www.tensorflow.org/ Software available from tensorflow.org.

[2] A. Acquaviva, D. Apiletti, A. Attanasio, E. Baralis, L. Bottaccioli, F. B. Castagnetti, T. Cerquitelli, S. Chiusano, E. Macii, D. Martellacci, and E. Patti. 2015. Energy Signature Analysis: Knowledge at Your Fingertips. In *2015 IEEE International Congress on Big Data*. 543–550. https://doi.org/10.1109/BigDataCongress.2015.85

[3] Daniele Apiletti, Elena Baralis, Tania Cerquitelli, Paolo Garza, Fabio Pulvirenti, and Luca Venturini. 2017. Frequent itemsets mining for Big Data: a comparative analysis. *Big Data Research* 9 (2017), 67–83.

[4] S. A. Asmai, A. S. H. Basari, A. S. Shibghatullah, N. K. Ibrahim, and B. Hussin. 2011. Neural network prognostics model for industrial equipment maintenance. In *2011 11th International Conference on Hybrid Intelligent Systems (HIS)*. 635–640. https://doi.org/10.1109/HIS.2011.6122180

[5] Radu F. Babiceanu and Remzi Seker. 2016. Big Data and virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook. *Computers in Industry* 81 (2016), 128 – 137. https://doi.org/10.1016/j.compind.2016.02.004 Emerging {ICT} concepts for smart, safe and sustainable industrial systems.

[6] Yan Chen, Feibai Zhu, and Jay Lee. 2013. Data quality evaluation and improvement for prognostic modeling using visual assessment based data partitioning method. *Computers in Industry* 64, 3 (2013), 214 – 225. https://doi.org/10.1016/j.compind.2012.10.005

[7] Tania Cerquitelli Alberto Macii Enrico Macii Massimo Poncino Daniele Apiletti, Claudia Barberis and Francesco Ventura. 2018. iS-TEP: an integrated Self-Tuning Engine for Predictive maintenance in Industry 4.0. In *16th IEEE International Symposium on Parallel and Distributed Processin with Applications, ISPA-18 Melbourne, Australia, December 11-13, 2018*. 8.

[8] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. [n. d.]. *The elements of statistical learning: data mining, inference and prediction* (2 ed.). Springer.

[9] Z. Huang, A. Zhong, and G. Li. 2017. On-Demand Processing for Remote Sensing Big Data Analysis. In *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*. 1241–1245. https://doi.org/10.1109/ISPA/IUCC.2017.00187

[10] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An introduction to statistical learning*. Vol. 112. Springer.

[11] V. Jirkovsky, M. Obitko, and V. Marik. 2016. Understanding Data Heterogeneity in the Context of Cyber-Physical Systems Integration. *IEEE Transactions on Industrial Informatics* PP, 99 (2016), 1–1. https://doi.org/10.1109/TII.2016.2596101

[12] M. Miškuf and I. Zolotová. 2016. Comparison between multi-class classifiers and deep learning with focus on industry 4.0. In *2016 Cybernetics Informatics (K I)*. 1–5. https://doi.org/10.1109/CYBERI.2016.7438633

[13] J. Murphree. 2016. Machine learning anomaly detection in large systems. In *2016 IEEE AUTOTESTCON*. 1–9. https://doi.org/10.1109/AUTEST.2016.7589589

[14] Nikolaos Nikolakis, Apostolos Papavasileiou, Konstantinos Dimoulas, Kiriakos Bourmpouchakis, and Sotirios Makris. 2018. On a versatile scheduling concept of maintenance activities for increased availability of production resources. *Procedia CIRP* 78 (2018), 172 – 177. https://doi.org/10.1016/j.procir.2018.09.065 6th CIRP Global Web Conference âĂŞ Envisaging the future manufacturing, design, technologies and systems in innovation era (CIRPe 2018).

[15] M. Niño, J. M. Blanco, and A. Illarramendi. 2015. Business understanding, challenges and issues of Big Data Analytics for the servitization of a capital equipment manufacturer. In *2015 IEEE International Conference on Big Data (Big Data)*. 1368–1377. https://doi.org/10.1109/BigData.2015.7363897

[16] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural Networks* 61 (2015), 85–117.

[17] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan. 2016. Multiobjective Deep Belief Networks Ensemble for Remaining Useful Life Estimation in Prognostics. *IEEE Transactions on Neural Networks and Learning Systems* PP, 99 (2016), 1–13. https://doi.org/10.1109/TNNLS.2016.2582798