2019 IEEE International Congress on Big Data (BigData Congress)

PREMISES, a scalable data-driven service to predict alarms in slowly-degrading multi-cycle industrial processes

Stefano Proto*, Francesco Ventura*, Daniele Apiletti*, Tania Cerquitelli*, Elena Baralis*, Enrico Macii†, Alberto Macii*
* Department of Control and Computer Engineering, Politecnico di Torino, Torino, Italy
† Interuniversity Department of Regional and Urban Studies and Planning, Politecnico di Torino, Torino, Italy
Email: * † name.surname@polito.it

Abstract*—In recent years, the number of industry-4.0-enabled manufacturing sites has been continuously growing, and both the quantity and variety of signals and data collected in plants are increasing at an unprecedented rate. At the same time, the demand of Big Data processing platforms and analytical tools tailored to manufacturing environments has become more and more prominent. Manufacturing companies are collecting huge amounts of information during the production process through a plethora of sensors and networks. To extract value and actionable knowledge from such precious repositories, suitable data-driven approaches are required. They are expected to improve the production processes by reducing maintenance costs, reliably predicting equipment failures, and avoiding quality degradation. To this aim, Machine Learning techniques tailored for predictive maintenance analysis have been adopted in** PREMISES (PREdictive Maintenance service for Industrial procesSES*****), an innovative framework providing a scalable Big Data service able to predict alarming conditions in slowly-degrading processes characterized by cyclic procedures.** PREMISES **has been experimentally tested and validated on a real industrial use case, resulting efficient and effective in predicting alarms. The framework has been designed to address the main Big Data and industrial requirements, by being developed on a solid and scalable processing framework, Apache Spark, and supporting the deployment on modularized containers, specifically upon the Docker technology stack.**

***Keywords*-Industry 4.0, Industrial Big Data, Industrial Machine Learning, Predictive Maintenance, Failure prognostics.**

I. Introduction

In recent days, the technological development applied to industrial contexts has led to a continuous growth of industry-4.0-enabled sites. As a consequence, from smart factories to more advanced IT industries (e.g., in the automotive scenario [1]), the amount of data produced by industrial sensors placed over the whole production chain is ever increasing. From the gathered data, modern connected industries aim to get scalable architectures able to support equipment maintenance and control within the production processes. By means of predictive maintenance, the costs of maintenance interventions can be considerably reduced, and production line interruptions can be avoided or scheduled to mitigate their impact.

In this paper we present PREMISES (*PREdictive Maintenance service for Industrial procesSES*), a scalable predictive maintenance service able to identify equipment critical conditions in multi-cycle industrial processes before their actual occurrences. The framework supports predictive diagnostic, allowing industrial stakeholders to easily and smartly plan maintenance operations and exploit the Machine Learning methodologies as a service. PREMISES is based on an intelligent data-driven approach to guide the prognostic of the smart industrial sensorized components and forecast the future evolution of the machine degradation from on-line data collected in factories. Innovatively, PREMISES provides predictions for slowly-degrading cyclic industrial processes over a user-defined time horizon, by exploiting both time series splitting and time-window aggregations.

The framework has been tested and validated on a real industrial use case of an international white-good company. Specifically, the production data have been gathered from industrial foaming machines, to monitor and predict the degradation of the equipment, and so promptly trigger the maintenance interventions. Several sensors measure different properties of the foaming process, cyclically applied to the chemicals involved, and their cycle operation timings: from an historical set of these measurement data, the aim of the analysis is to predict alarm conditions, which would bring to machine faults and production interruptions.

Furthermore, to comply with the common needs of manufacturing enterprises, PREMISES has been designed following specific principles: (i) built upon supported and well-known Big Data platforms; (ii) compatible with both on-premises and in-the-cloud environments; (iii) easily deployable, thanks to containerized software modules; (iv) virtually unlimited horizontal scalability; (v) fault-tolerance and ability to self-reconfigure; (vi) provisioning of self-tuning Machine Learning techniques.

The paper is organized as follows. Section II presents the current state-of-the-art in the industrial predictive maintenance context; Section III shows the main building blocks of PREMISES's architecture. Section IV discusses the PREMISES's performance over a real industrial use case. At the end, Section V draws the conclusions of the research work and provides future directions.

II. Related works

With the adoption of smart environments and platforms, industry 4.0 enables companies to reach even greater productivity and flexibility. In [2] the authors present a smart factory

978-1-7281-2772-9/19/$31.00 ©2019 IEEE
DOI 10.1109/BigDataCongress.2019.00032

framework incorporating industrial networks, the cloud, and supervisory control terminals with smart shop-floor objects such as machines, conveyors and products, resulting in a self-organized system leveraging the feedback and the coordination of the central control in order to achieve high efficiency. To this aim, the modern industrial context requires more flexible tools and platforms to collect and analyze the huge amount of data collected by sensorized machinery. This is both a challenge and an opportunity to extract even more value from the production processes. [3] addresses the trend of manufacturing transformations in industry 4.0 environments and study the readiness of IT tools in managing industrial Big Data and predicting maintenance operations. In [4] the authors propose a framework for structuring multi-source heterogeneous data, considering spatiotemporal properties and modeling invisible factors. This would make the production processes transparent and would allow implementing predictive maintenance and provide remaining life predictions of key components of machining equipment. Examples of architectures for real-time data processing are presented in [5] and [6]. Both are distributed architectures based on open source state-of-the-art frameworks (i.e. Apache Kafka, Apache Spark, Cassandra) providing reliability and scalability for IoT sensor networks. The former also provides the integration of a visualization tool, the latter presents a self-tuning engine for predictive maintenance, enabling manufacturing intelligence of which predictive maintenance is an expression. Furthermore, in [7] the authors explore a Big Data approach based on local learning with Support Vector Regression (SVR) to perform energy consumption predictions and compared this approach with traditional SVR and with Deep Neural Networks exploiting an H2O machine learning platform for Big Data. Many efforts to reduce the need of domain experts and the reduce the costs of running Machine Leaning algorithms have been done [8], [9], [10], [11], [12]. Among the works to make ML solutions exploitable as a service, [9] proposes an architecture to create a flexible and scalable Machine Learning service, developing an open source solution and validating it on a forecast analysis of electricity demand using real-world sensors and weather data. [10] provides an empirical comparison of MLaaS platforms, where the authors evaluate the effectiveness of fully-automated systems, turnkey systems and fully-customizable systems. To overcome the complexity of Big Data pipelines, in [13] the authors propose a new methodology based on Model Driven Engineering (MDE). Their work aims to lower the amount of skills required in the management of a Big Data pipeline and to support automation of Big Data analytics.

The proposed architecture, PREMISES, improves the state-of-the-art by introducing a new general approach in forecasting critical events generated by multi-cycles industrial processes and occurring in large time horizons with respect to the production cycle. It exploits a time-based aggregation approach and self-tuning strategies, still providing scalability, reliability and flexibility as required in modern industrial scenarios. The proposed solution has been validated on a real industrial use case concerning the prediction of the degradation of industrial foaming machines.

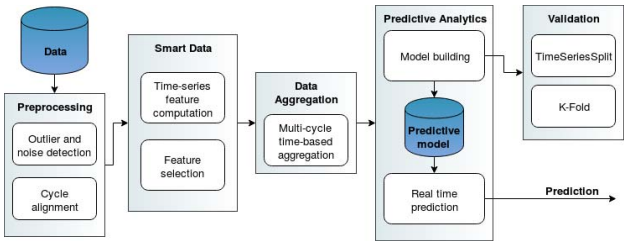

Fig. 1: PREMISES's architecture consisting of five building blocks.

III. The PREMISES's architecture

The architecture of PREMISES is presented in Figure 1. The analytics flow consists of five main building blocks, each executing a specific analytic step: *Preprocessing*, *Smart Data*, *Data Aggregation*, *Predictive Analytics*, and *Validation*. In the following, the five building blocks are described.

Preprocessing. This component performs the common steps required to clean the data collection under analysis, such as detecting and removing outliers (i.e., extreme values). Specifically, deciles of cycle lengths are exploited to remove cycles belonging to the first and the last deciles. Such approach was confirmed to be useful directly by domain experts, who knew that some cycles were not actual production measurements but test cycles, for instance, and indeed they were successfully discarded. Additionally, to address the cyclic nature of the industrial processes under exam, an alignment task is performed to make the data fit a fixed-time structure by means of padding (the last value of the cycle is repeated until the cycle time slot is filled).

Smart Data. This phase transforms the raw time series coming from sensors into a time-independent feature set able to characterize the corresponding portion of the original time series. Namely, the original time series is split into contiguous portions, with the split size being a parameter of PREMISES. Then, for each portion, statistical features able to summarize the time series trend are computed, such as mean, standard deviation, quartiles, Kurtosis, Skewness, sum of absolute values, number of elements over the mean, etc. It is worth noting that having same-size portions is a choice of simplicity that was proven to work in our use case, however the proposed approach can be successfully applied to splits of different sizes, since their purpose is to capture specific transient states and steady states in cyclic industrial processes, whose duration can vary.

Finally, a feature selection phase is performed over the full set of the many statistical features, to identify those which are more informative and discard the useless ones. To this aim, two techniques are exploited in PREMISES: (i) multicollinearity-based and (ii) correlation-based, both requiring a threshold to be defined as a parameter of PREMISES. The former removes attributes whose values can be trivially predicted by a multiple regression model of the other attributes. The latter computes

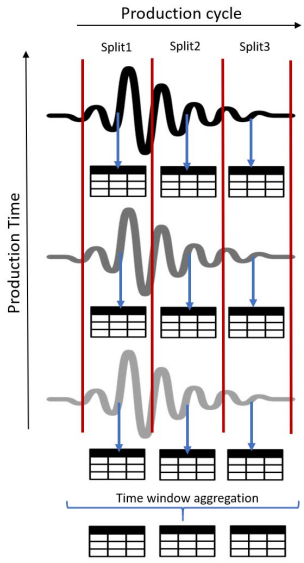

Fig. 2: Splitting of each cycle of the original time series, Smart Data computation of each split portion, and multi-cycle aggregation schema.

the correlation of each couple of attributes and removes those that are correlated the most, on average over all the (other) attributes.

Data aggregation. For many slowly-degrading cyclic industrial processes, such as ours, the single cycle is too short with respect to the target degradation phenomena and its prediction horizon. Often, in cyclic industrial processes, there is no interest in predicting the alarming-condition of a specific cycle, but that of a longer period, such as hours or days, which span over many cycles. Hence, this step aggregates the Smart Data cycle-related features over longer, multi-cycle, time windows. The aggregation is performed separately for each split. PREMISES captures the degradation of each Smart Data feature by computing a linear regression on the aggregated multi-cycle period, and records the slope and intercept coefficients. Furthermore, for each attribute, the min, max, mean and standard deviation of the values within the multi-cycle time window are recorded. A visual clue of the data processing executed from the raw cyclic time series to the data aggregation is presented in Figure 2. It is worth noting that both the feature selection and the feature aggregation preserve the meaning of the measurements in terms of human readability, hence keeping the approach transparent and its decisions easily accountable.

Predictive analytics. This building block consists of two steps: *model building* and *real-time prediction*. The *real-time prediction* step simply exploits a pre-built model to assign a label to new incoming data. The *model building* instead is a crucial step that performs the training on the historical data and extracts the latent relations among the data and the prediction labels (alarming conditions registered in the past). PREMISES exploits two ensemble learning classification algorithms: Gradient Boosted Tree Classifier [14] and Random Forest [15], as provided by the Spark MLlib library [16], automatically selecting the best performing one according to a score metric (defaulting to F-Score).

The validation is performed using the *Stratified K-Fold Cross Validation* technique. It equally divides the dataset into K folds (keeping the proportions of the original label distribution in each fold, hence the name stratified) and, for K times, alternatively uses a fold as test set and the other $K-1$ as training set. PREMISES evaluates the model by computing the precision, the recall and the F-Score for the class of interest (i.e. the alarming conditions or failures).

In conclusion, PREMISES provides a scalable Machine Learning-based predictive-maintenance service. It addresses slowly-degrading multi-cycle industrial processes to predict alarming conditions or failures by creating a data-driven model on historical data. Different strategies are exploited to describe cyclic time series data, aggregate them over multi-cycle horizons and self-tune the predictive model to offload data scientists and domain experts from manual interventions. While the multi-cycle aggregation time window and the split size within a cycle are parameters that can be set by domain experts, since they describe or capture features of the industrial process, the algorithmic parameters are self-tuned by PREMISES itself, thanks to a grid optimization search over the classification algorithm parameters and the feature selection thresholds (correlation and multicollinearity).

Finally, the whole architecture is designed and developed to provide a scalable service to address Big Data and Industry 4.0 requirements, able to manage rapidly increasing volumes of data. To this extent, the framework has been developed by exploiting the Apache Spark framework and MLlib library, containerized using Docker technology.

IV. Experimental results

The general-purpose PREMISES framework has been customized for a real industrial use case and validated on real data collected in a manufacturing plant of an international white-goods company.

The experimental results are obtained on an Intel Core i7 machine with 32 GB of main memory running Ubuntu 16.04 with Spark (and MLlib) 2.2.0 and Docker 17.09.

Real industrial data. The dataset was collected from sensors placed in a factory floor: a nozzle injecting a combination of two reacting chemicals has been sensorized to monitor the overall process. During each production cycle, the nozzle is used to inject an isolating foam. Different types of signals have been collected: temperature of the chemicals involved, pressure of the liquids before the injection, injection timing and quantity, ratio of the injected chemicals, etc. A set of different alarms, each describing a specific issue, has been collected and associated with the original production cycle that presented such issue. The final goal of the use case is to predict if a given set of monitored production cycles (e.g., hourly) will trigger an alarm condition in a given time horizon (e.g., the next few hours).

TABLE I: Cycles length distribution.

Deciles	min	10%	20%	30%	40%	50%	60%	70%	80%	90%	max
Values	67	122	164	165	166	166	167	168	169	171	568

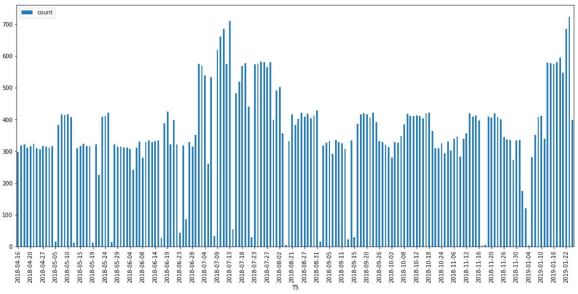

Fig. 3: Number of cycles per day

TABLE II: Alarms description.

Description	**Count**
Emergency	18
Chem1 tank: temperature out of range 1	250
Chem1 tank: temperature out of range 2	34
Chem1 tank: safety thermostat	30
Chem1 tank: minimum level	2
Chem2 tank: temperature out of range 1	154
Chem2 tank: temperature out of range 2	29
Chem2 tank: safety thermostat	30
Chem2 tank: minimum level	4
Chem2 loading stream distributor: not open	2
Fire safe: not open	48
Fire safe: not closed	20
High-pressure frame/piping: monitoring trouble	7
Chem1 high-pressure pump: motor fault	2
Chem1 recycle stream distributor 1: not closed	4
Chem1 pressure: maximum	1
Chem2 recycle stream distributor 1: not closed	30
Chem2 pressure: maximum	1
Ratio: out of range 1	4
Pouring weight: out of range 1	120
Pouring weight: out of range 2	72
Heads hydraulic unit: pressurization time long	2
Heads hydraulic unit: oil filter clogged	16
Head 1: air suction fault	2
Head 1: pouring piston not closed	2
Head 1: self-cleaning piston not open	68
Head 1: no enable to pour signal	10
Chem1 temperature head 1: out of range 1	6
Chem2 temperature head 1: out of range 1	8
Head 1 maintenance: maximum number of high-pressure hose cycles	1

The dataset collects measurements in the period ranging from April 2018 to January 2019. Data contains a total of 66,000 production cycles of the same industrial process, unevenly distributed over 183 actual days, due to holidays and work interruptions, as shown in Figure 3. It is worth noting that the non-contiguous data collection does not affect the experimental results: when no data were present, the equipment was not working, hence domain experts assumed that no degradation was happening.

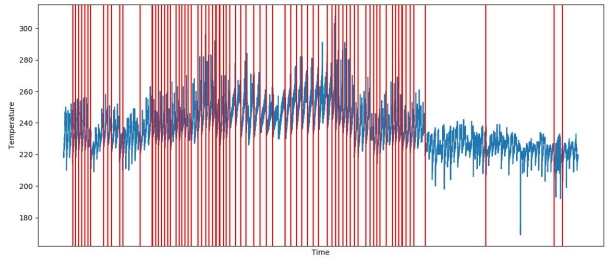

Fig. 4: Data labeling applied with 1-day aggregation window, red lines are alerts.

A. Data exploration and preprocessing

Table I reports the deciles of the cycle lengths in number of samples (sampling has a constant timing). The cycle lengths are reduced from 171 to 164 samples by removing the first and last deciles. Then, all cycles are padded to be as long as the longest ones.

Data labeling. Alerts have been divided into categories depending on their causes, as reported in Table II, along with their description and their count. Each alert is associated with the aggregation time window it belongs to, whose length can be 4-8-24 hours (as shown in Figure 4).

B. Smart Data computation and aggregation

Each cycle of the industrial process under exam consists of 8 changes of state, hence we set 8 splits as PREMISES parameter. Experimental results are reported for 5 and 10 as Variance Inflation Factor (VIF) in the multicollinearity test and using the values 0.3 and 0.5 as thresholds for the Pearson correlation test [17]. Generally, the former strategy selects fewer attributes with respect to the latter. Tables III reports the experimental results along with their configuration. Column *Time_W* describes the aggregation time window values (4 hours, 8 hours and 1 day), *Alg* refers to the algorithm used to perform the classification (Gradient Boosted classifier or Random Forest), *FS* describes the feature selection strategy adopted in the experiment, and *Thres* refers to the threshold value used for the feature selection strategy.

C. Predictive analytics results

To build the model we used two different algorithms (Gradient Boosted Tree Classifier and Random Forest), while for the testing phase we chose the Stratified K-Fold validation technique, configured with 3 splits ($K = 3$). For each configuration of the analytics flow, a set of evaluation indexes have been computed; specifically, we considered the precision, the recall and the F-Score (columns *Prec*, *Rec* and *F-Score* in Tables III) for the failure class, representing the occurrence of an alarm in the time window. The scores in Table III measure the ability of the different algorithms to predict the occurrence

TABLE III: Experimental results, Stratified K-Fold approach.

Time_W	Alg	FS	Thres	Prec	Rec	F-Score
1d	GB	all		0.817	0.763	0.776
		corr	0.3	0.627	0.662	0.624
			0.5	0.790	0.752	0.756
		m_coll	5	0.623	0.562	0.576
			10	0.683	0.685	0.665
	RF	all		0.843	0.818	0.812
		corr	0.3	0.665	0.629	0.627
			0.5	**0.842**	**0.830**	**0.822**
		m_coll	5	0.676	0.606	0.614
			10	0.667	0.607	0.619
8H	GB	all		0.449	0.314	0.334
		corr	0.3	0.318	0.229	0.247
			0.5	0.521	0.314	0.339
		m_coll	5	0.378	0.238	0.266
			10	0.315	0.210	0.237
	RF	**all**		**0.563**	**0.352**	**0.394**
		corr	0.3	0.272	0.133	0.178
			0.5	0.530	0.295	0.333
		m_coll	5	0.332	0.152	0.208
			10	0.242	0.114	0.155
4H	GB	all		0.466	0.301	0.344
		corr	0.3	0.243	0.146	0.173
			0.5	0.554	0.319	0.380
		m_coll	5	0.299	0.163	0.203
			10	0.219	0.137	0.163
	RF	**all**		**0.617**	**0.345**	**0.424**
		corr	0.3	0.310	0.112	0.162
			0.5	0.625	0.327	0.399
		m_coll	5	0.260	0.120	0.164
			10	0.306	0.069	0.111

of an alarm in a given time horizon: for each time window, the best performing configuration is highlighted in bold. From the evaluation of the performed experiments, it is possible to assess that the multicollinearity feature selection strategy is too restrictive, being *all* and the *correlation* strategies selected several times as best configurations. For the given configurations, the results show that the best time window length is the daily one.

V. Conclusions and future works

This paper presents PREMISES, a scalable analytic framework providing a prognostic service for predictive maintenance. The framework, customizable for different multi-cycle industrial processes, has been tested on a real use case, resulting efficient and effective in predicting alarm conditions within a certain user-defined time-horizon. PREMISES has been designed and developed to scale and support Big Data analysis.

Future directions of the research work include: (i) introducing the frequency-domain analysis, to be able to capture different aspects of the signals received from the sensorized factory floors, (ii) make the predictive model evolutive, introducing quality metrics to be able to self-assess the goodness of the results and trigger a new model-building phase to update the system, and (iii) consider and integrate in the model the ordinary maintenance operations scheduled on the machines.

Acknowledgment

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

References

[1] F. Giobergia, E. Baralis, M. Camuglia, T. Cerquitelli, M. Mellia, A. Neri, D. Tricarico, and A. Tuninetti, "Mining sensor data for predictive maintenance in the automotive industry," in *5th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2018, Turin, Italy, October 1-3, 2018*, 2018, pp. 351–360.

[2] S. Wang, J. Wan, D. Zhang, D. Li, and C. Zhang, "Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination," *Computer Networks*, vol. 101, 2016, industrial Technologies and Applications for the Internet of Things.

[3] 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, product Services Systems and Value Creation. Proceedings of the 6th CIRP Conference on Industrial Product-Service Systems.

[4] J. Yan, Y. Meng, L. Lu, and L. Li, "Industrial big data in an industry 4.0 environment: Challenges, schemes, and applications for predictive maintenance," *IEEE Access*, vol. 5, pp. 23 484–23 491, 2017.

[5] G. M. D'silva, A. Khan, Gaurav, and S. Bari, "Real-time processing of iot events with historic data using apache kafka and apache spark with dashing framework," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, May 2017.

[6] D. Apiletti, C. Barberis, T. Cerquitelli, A. Macii, E. Macii, M. Poncino, and F. Ventura, "istep, an integrated self-tuning engine for predictive maintenance in industry 4.0," in *IEEE International Conference on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications, ISPA/IUCC/BDCloud/SocialCom/SustainCom 2018, Melbourne, Australia, December 11-13, 2018*, 2018, pp. 924–931. [Online]. Available: https://doi.org/10.1109/BDCloud.2018.00136

[7] K. Grolinger, M. A. M. Capretz, and L. Seewald, "Energy consumption prediction with big data: Balancing prediction accuracy and computational resources," in *2016 IEEE International Congress on Big Data (BigData Congress)*, June 2016, pp. 157–164.

[8] T. Cerquitelli, E. Baralis, L. Morra, and S. Chiusano, "Data mining for better healthcare: A path towards automated data analysis?" in *32nd IEEE International Conference on Data Engineering Workshops, ICDE Workshops 2016, Helsinki, Finland, May 16-20, 2016*, 2016, pp. 60–63.

[9] M. Ribeiro, K. Grolinger, and M. A. M. Capretz, "Mlaas: Machine learning as a service," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, Dec 2015.

[10] Y. Yao, Z. Xiao, B. Wang, B. Viswanath, H. Zheng, and B. Y. Zhao, "Complexity vs. performance: Empirical analysis of machine learning as a service," in *Proceedings of the 2017 Internet Measurement Conference*, ser. IMC '17. New York, NY, USA: ACM, 2017, pp. 384–397.

[11] D. Apiletti, E. Baralis, T. Cerquitelli, P. Garza, D. Giordano, M. Mellia, and L. Venturini, "Selina: A self-learning insightful network analyzer," *IEEE Trans. Network and Service Management*, vol. 13, no. 3, 2016.

[12] T. Cerquitelli, E. D. Corso, F. Ventura, and S. Chiusano, "Data miners' little helper: data transformation activity cues for cluster analysis on document collections," in *Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics, WIMS 2017, Amantea, Italy, June 19-22, 2017*, 2017, pp. 27:1–27:6.

[13] C. A. Ardagna, V. Bellandi, P. Ceravolo, E. Damiani, M. Bezzi, and C. Hebert, "A model-driven methodology for big data analytics-as-a-service," in *2017 IEEE International Congress on Big Data (BigData Congress)*, June 2017.

[14] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[15] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, 2001.

[16] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen, D. Xin, R. Xin, M. J. Franklin, R. Zadeh, M. Zaharia, and A. Talwalkar, "Mllib: Machine learning in apache spark," *J. Mach. Learn. Res.*, vol. 17, no. 1, Jan. 2016.

[17] S. M. Ross, *Introduction to probability models*. Academic press, 2014.