REDTag: A Predictive Maintenance Framework for Parcel Delivery Services

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 4, 2019, accepted January 6, 2020, date of publication January 14, 2020, date of current version January 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2966568

REDTag: A Predictive Maintenance Framework for Parcel Delivery Services

STEFANO PROTO[1], EVELINA DI CORSO[1], DANIELE APILETTI[1,2], LUCA CAGLIERO[1], (Member, IEEE), TANIA CERQUITELLI[1], (Member, IEEE), GIOVANNI MALNATI[1], AND DAVIDE MAZZUCCHI[3]

[1]Dipartimento di Automatica e Informatica, Politecnico di Torino, 10129 Torino, Italy
[2]SmartData Center, Politecnico di Torino, 10129 Torino, Italy
[3]Zirak srl, 12084 Mondovì, Italy

Corresponding author: Luca Cagliero (luca.cagliero@polito.it)

This work was supported by the SmartData@PoliTO Center of Politecnico di Torino, the Regional Funds for Regional development, the Italian Ministry of Research (MIUR), and in part by the Piedmont Region under Grant POR FESR 2014-2020.

ABSTRACT The overwhelming increase of parcel transports has prompted the need for effective and scalable intelligent logistics systems. In parallel, with the advent of Industry 4.0, a tight integration of Internet of Things technologies and Big Data analytics solution has become necessary to effectively manage industrial processes and to early predict product faults or service disruptions. In the context of good transports, the development of smart monitoring tools is particularly useful for couriers to ensure effective and efficient parcel deliveries. However, the existing predictive maintenance frameworks are not tailored to parcel delivery services. We present REDTag Service, an integrated framework to track and monitor the shipped packages. It relies on a network of IoT-enabled devices, called REDTags, allowing courier employees to easily collect the status of the package at each delivery step. The framework provides back-end functionalities for smart data transmission, management, storage, and analytics. A machine-learning process is included to promptly analyze the features describing event-related data to predict potential breaks of the goods in the packages. The framework provides also a dynamic view on the integrated data tailored to the different stakeholders, as well as on the prediction outcomes, enabling immediate feedback and model improvements. We analyze a real-world dataset including event-related data about parcel transports. To validate the hypothesis that the acquired data contains information relevant to predict the package status (i.e., broken or safe), we empirically analyze the performance of different, scalable classifiers. The experimental results confirm, in good approximation, the predictive power of the models extracted from the event-related features. To the best of the authors' knowledge, this work is the first attempt to address predictive maintenance in smart good transport logistics to predict package breaks from real-world data.

INDEX TERMS Big data analytics, Industry 4.0, intelligent transports and logistics, Internet of Things, machine learning, predictive maintenance.

I. INTRODUCTION

The emergence of Industry 4.0 factories has fostered the diffusion of Internet of Things (IoT) technologies and big data analytics tools in the industrial sector [1]. The so called Logistics 4.0 has deeply increased the needs for transparency in the supply chain and integrity control in good selling and delivery (i.e., sell the right product at the right cost and deliver it at the right time and place). As pointed out by [2], a smart way to handle logistics entails relying on the following technological applications: (i) Resource planning, (ii) Warehouse Management Systems, (iii) Transportation Management Systems, (iv) Intelligent Transportation Systems, and (v) Information Security. Since the scope of this work is to push advanced Information Technology solutions into good transports, it falls into category (iii).

In recent years, parcel transports have grown at a surprisingly high rate. The globalization and the spread of online shops are the key factors that caused the significant growth in the number of commodities delivered by specialized couriers. To design intelligent logistics systems that are able to address today's challenges, couriers need to adopt advanced

The associate editor coordinating the review of this manuscript and approving it for publication was Najah Abuali.

sensor and information technologies to avoid damages to goods in packages [3]. The key requirements for a successful customer-oriented intelligent logistics system have been summarized in [4]. However, unsolved problems still raise despite the use of modern intelligent logistic technologies. Among the others, promptly identifying damaged packages is still an open research issue. In particular, this is the main challenge addressed by the present study. Predicting potential package breaks is particularly useful in transport logistics, because it may help decrease the waste of unusable merchandise, reduce the transport means causing the breaks, promote accountability for the transportation processes, provide early warning at different stages of a delivery trip, yield better end-user satisfaction, and promote high-quality couriers.

To predict critical situations as soon as possible, a relevant research effort in the research community of industrial data mining and machine learning has been devoted to predictive maintenance [5], with a special focus on smart factories. It entails automatically predicting critical events by training data-driven classification or regression models. The analyzed data consists of past events, product characteristics, geo-spatial object tracking, and service usage monitoring. Predictive maintenance has the twofold aim to (i) reduce maintenance frequency to lowest possible state leading to a huge cost saving in keeping resources in normal working condition [6], and (ii) avoid catastrophic situations (e.g., product breaks, faults, service disruptions) by detecting anomalies a-priori from historical data. In the latter case, which is the most relevant one to our purposes, maintenance should be undertaken on time to prevent failure occurrence. The opportunities of using IoT, data analytics, and machine learning approaches to predict critical situations have already been investigated in several industrial contexts, among which vehicle fleet management [7], oil and gas mining [8], and power plant design [9]. This work is the first attempt to address predictive maintenance in smart good transport logistics to predict package breaks.

This paper proposes an IoT-based, big data analytics framework, namely REDTag Service, to monitor the status of the shipped packages and to early predict their accidental break. It relies on an IoT-enabled device, which consists of a smart ad-hoc hardware tag (hereafter denoted as REDTag). The tag is applied on each shipped package. Notice that such a small, cheap, and low-energy tag can be equipped with many different sensors, in order to measure and record different events based on the kind of item to monitor. Indeed, the REDTag can track fall events, impacts, temperature, humidity, and position of the parcels on which it is applied. A logical schema of the REDTag components is provided in Figure 1.

The proposed framework provides back- and front-end services for storing, managing, and analyzing REDTag data. It relies on a Big Data architecture to store data in a scalable and effective way and on a machine learning component, which learns prediction models from the input data. A classification model is trained on a subset of discriminating features describing the tracked data in order to predict the status of

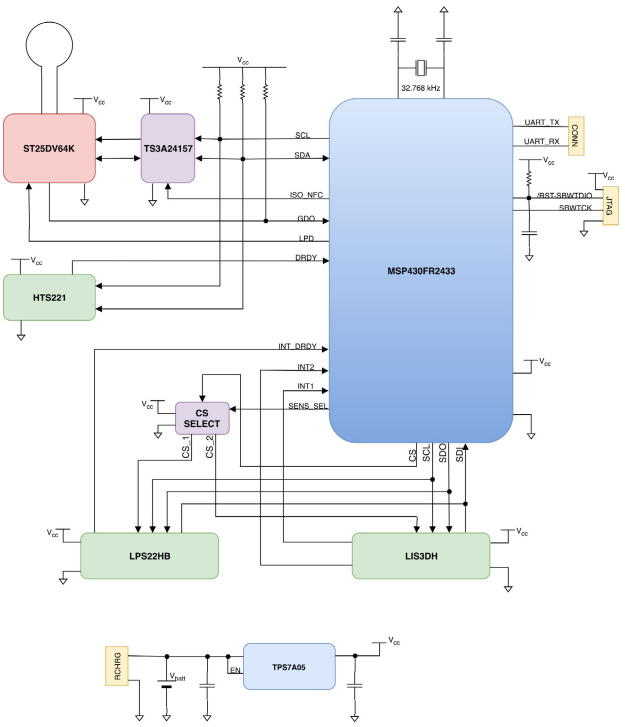


FIGURE 1. Logical schema of the REDTag component.

each package. The acquired data and the prediction outcomes are displayed through dynamic and configurable informative dashboards to allow courier employees to monitor good transports and promptly react against potentially critical situations. Customers can also get notified about where, when, and who caused a damage to the transported packages, encouraging the carriers to pay more attention to the transported parcels.

The effectiveness of the proposed approach has been empirically validated in a real-world case study. The experiments show that the descriptors used to describe the tagged events are fairly correlated with package status. Thus their predictive power can be exploited to train classification models with desirable performance.

The contributions of the paper can be summarized as follows:

- This work is, to the best of the authors' knowledge, the first attempt to address machine-learning-based good transport logistics to predict package breaks from historical data.
- It provides a data-driven solution with tight integration with novel IoT-enabled technologies (i.e., the REDTag) and with established Big Data scalable frameworks.
- A preliminary performance analysis has been conducted on real-world data.

This paper is organized as follows. Section II discusses the position of the paper with respect to the related literature. Sections III and IV describe the use case scenario addressed by the work and the framework enabling the proposed service, respectively. Section V thoroughly describes the data analytics and mining steps. Section VI shows the experimental

results achieved in a real scenario. Finally, Sections VII draws conclusions and presents the future research agenda related to this research activity, respectively.

II. LITERATURE REVIEW

Several previous studies have addressed predictive maintenance problems in the field of industrial big data analytics. For example, a pioneering work has been presented by [10], where the authors highlight the need for a change of paradigm from the traditional *fail and fix* maintenance practices to the *predict and prevent* e-maintenance methodology. They introduce one among the first examples of performance assessment and prediction tools, which performs proactive maintenance to prevent machines from breakdowns. As recently pointed out by [11], [12], limiting the risks due to unexpected faults/issues not only improves the management activities but also reduces the economic, environmental, and social costs.

In the context of Industry 4.0 factories, two main challenges need to be addressed in order to effectively accomplish predictive maintenance tasks: (i) the tight integration of Internet of Things (IoT) technologies in the smart factory context, which enables smart sensor networks and smart machines to share data with the system components [13], [14], and (ii) the Variety, Velocity, and Volume of the acquired data, which need to be analyzed through big data and machine learning technologies [15]. Integrating IoT and data mining technologies allows us to overcome the strong dependence on human experts, which is unacceptable when the analyzed data scales towards huge datasets.

A literature review of the state-of-the-art big data analytics solutions in manufacturing has recently been proposed by [16]. The authors in [17] have presented a new prognostics model based on neural networks to support industrial maintenance decisions. It predicts both failure likelihood and the remaining equipment lifetime. Authors in [18] have proposed a data-driven risk management framework based on time series data analyses, while the frameworks presented by [3] and [19] respectively address energy saving and optimization and product life-cycle management and maintenance. With the goal of detecting the health of a system, [20] has applied anomaly detection algorithms to detect failure or a pending failure from the system measurements. A particular attention has also been paid to the application of Deep Learning models in prognostics (e.g., [19], [21]). Despite Deep Neural Network models are potentially more accurate than traditional ones, they are inherently not explainable. In fact, they are usually applied as a black-box, without giving any valuable insight into the reasons behind the generated predictions. Furthermore, they require accurate settings of the model hyperparameters in order to avoid data over- and under-fitting. To address the above-mentioned issues, parallel research efforts have addressed the visualization of the input data to perform quality assessment and prognostics [22] and the self-tuning of the machine learning algorithms to minimize human intervention in the data analytics process [23].

This paper presents a predictive maintenance framework tailored to the context of intelligent good transports and logistics. It proposes a machine learning approach to predict package breaks based on the analysis of historical data acquired by IoT-enabled devices. The proposed solution is designed to be flexible, scalable, and incremental, thus allowing carriers to monitor the status of good transports and promptly react against potential package damages. To the best of our knowledge, this work is the first attempt to analyze good transports data in this way.

III. USE-CASE SCENARIO

The scenario of our use case can be summarized as follows. Customers submit orders for goods. Orders are managed by the employees of the courier. For the sake of simplicity, each order is associated with a single parcel, which is going to travel from a sender to a receiver. During good transports, the courier employees track the status of the package by using the REDTag technology. In case an event causes a package break, it is unlikely to be detected immediately unless a prediction system would be able to detect changes in the package characteristics and forecast a potentially critical situation. This is the main purpose of the REDTag Service, which is in charge of providing courier workers and customers with ad hoc alerts along with a detailed description of the current situation.

The framework keeps track of the following information.

- **Customer**: The sender and receiver of the package.
- **Package**: The package of the parcel carried by the transport services.
- **Tag**: The IoT-enabled hardware tag (i.e., REDTag) applied on each package.
- **Worker**: A employee of the courier, moving the package at different stages towards its destination.
- **Tag_event**: The reading of the REDTag collected data by a worker.
- **Order**: The shipping order made by a customer (sender) to another (receiver).
- **Segment**: The progress level of a parcel towards the destination. It is tagged by the worker who is currently in charge of managing the parcel.
- **GPS_worker**: The geographical position of the worker in charge of managing the parcel.

Based on the tracked information, the framework predicts whether the current status of a parcel is *broken* or *safe*. The underlying assumption is that the collected data about the previous events of package deliveries are likely to be correlated with the current-delivery package status. An empirical evaluation of such assumption is given in Section VI.

IV. PROPOSED FRAMEWORK

This section describes the framework proposed to effectively and efficiently monitor the status of the packages of a courier during their transports.

The REDTag Service framework allows stakeholders (e.g., customers, deliverers, retailers) to monitor the package

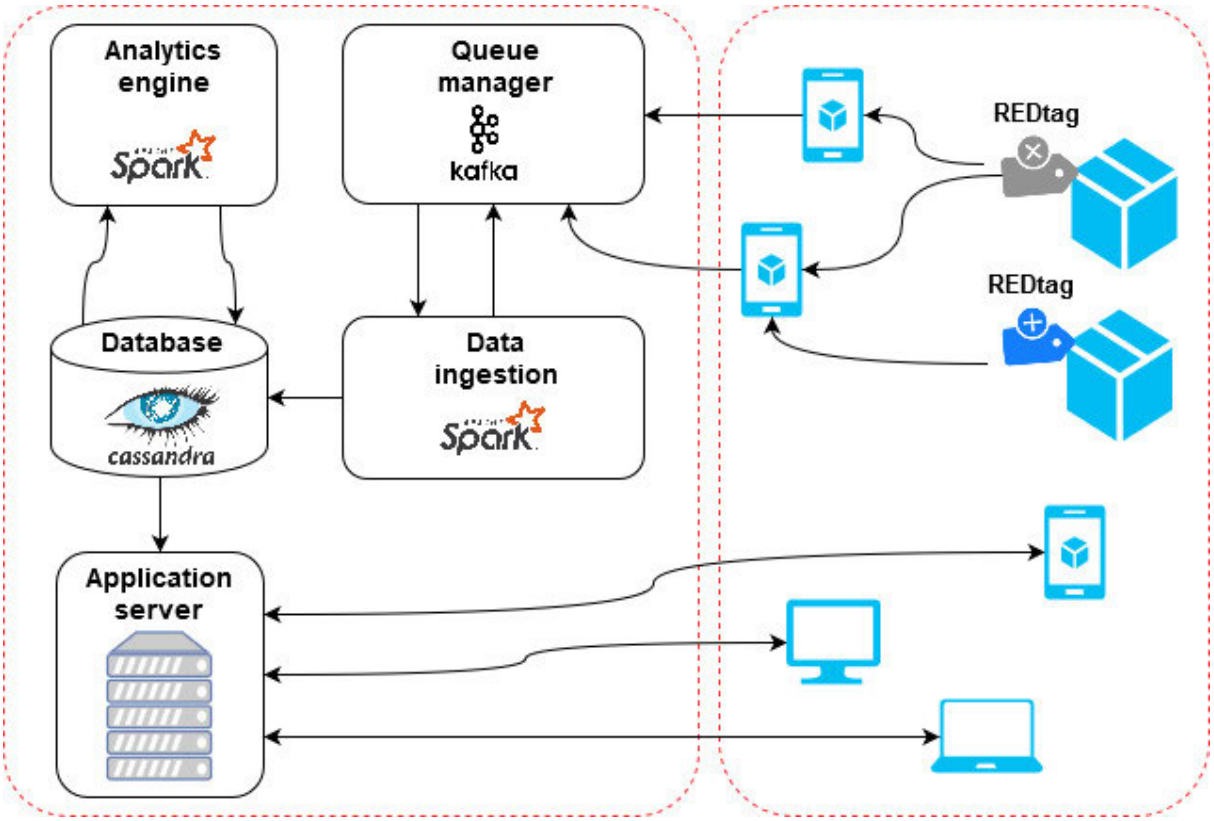


FIGURE 2. The framework architecture.

status in real time. Figure 2 shows the architecture of the proposed REDTag Service framework.

It provides a variety of back- and front-end services aimed to support predictive maintenance activities. The back-end services include:

- **REDTag**: The hardware technology enabling event detection and package-data recording and collection.
- **Queue Manager**: The IoT service adopted to coordinate data exchange from the sensors to the back-end.
- **Data ingestion**: The Big Data technologies used to manage the input data flow and to store them in a non-relational database (Cassandra).
- **Application server**: The server running back- and front-end services.
- **Analytics engine**: The machine learning modules in charge of predicting package breaks based on historical data.

The front-end services offer data visualization, aggregation, and reporting functionalities over the acquired data as well as visualization of the prediction outcomes (i.e., the predicted status of the package). It allows stakeholders to deeply analyze the real-time predictions in order to react against unexpected events, limit inefficiencies, and improve the quality of the offered service. A more detailed description of the provided back-end services is given in Section IV-A, whereas front-end services are described in Section IV-B.

A. BACK-END SERVICES

This section details the most relevant characteristics of the back-end services.

1) REDTag: EVENT DETECTION AND RECORDING

REDTag is a state-of-the-art event recording technology.[1] It is designed to be attached to the packages to efficiently and effectively record all the events occurring during parcel transport. The tag consists of a red box (for which the name comes from) equipped with sensors, batteries, memories, and a processor. A networking module is also embedded in the tag to allow the processor to exchange data with external mobile devices. REDTag has been adopted as event detection and recording systems because it is fairly cheap, robust, small but well visible. Notably, it does not require any configuration setting before plugging it in the system. Furthermore, it has been designed to be eco-friendly, having the possibility to use just eco-sustainable electronic components. When it detects an event, e.g. the acceleration is above a given threshold, the REDTag stores it into its memory. Data downloads from tags to smartphone are performed via NFC (Near Field Communication). A specific smartphone application, operated by courier workers, is in charge of sending data to the back-end services. The key interactions between the architectural elements of the REDTag Service are depicted in Figure 3.

[1] http://www.zirak.com/engineering_embedded_rfid/

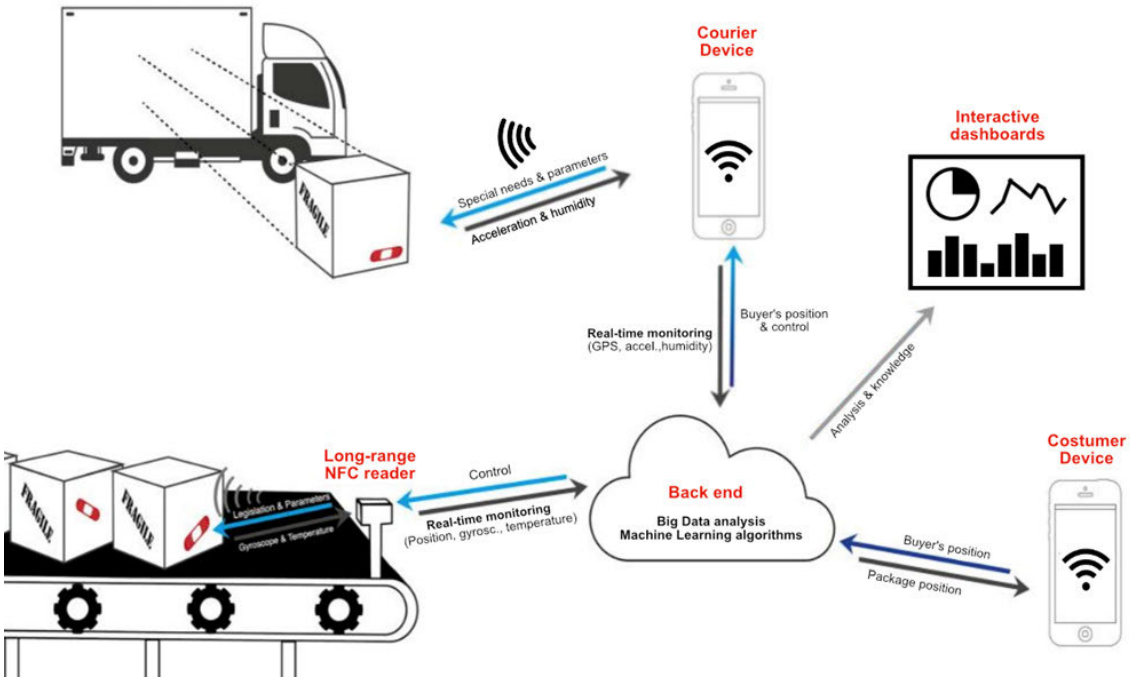


FIGURE 3. Relevant interactions among the architecture elements.

2) QUEUE MANAGER

When workers scan the REDTags, data acquired by the sensors are sent to the back-end and gathered by a queue manager. Apache Kakfa [24] is the technological solution adopted in the REDTag Service framework for the data management and distributed data processing. The platform is designed to manage high-rate data streams and ensures reliability and scalability by distributing jobs over multiple workers. In this framework it is adopted to collect data from IoT devices and manage data ingestion with the necessary reliability, effectiveness, and scalability levels.

A Kafka REST Proxy Server exposes the Application Programming Interfaces through which the event messages are exchanged. Kafka has a dedicated channel per event type. The event types are classified as follows: (i) package break, (ii) new association tag/package, (iii) worker commitment for a parcel, (iv) new tag event, (v) new delivery, (vi) new GPS event. For event (i), the worker in charge can confirm the break by manual inspection. When a worker is committed for a given package, i.e., event (iii), a new session starts. Notice that all the past events corresponding to the package (if any) are downloaded from the tag and associated with the current package segment.

3) DATA INGESTION

Once the data arrives to the back-end of the service, the data ingestion component is in charge of the Extraction, Transformation, and Loading (ETL) process [25]. Sensor data are integrated with the carrier-provided information about packages, workers, and customers. The database employed to manage and store data is Apache Cassandra [26]: it is a largely used NoSQL column-based database. It ensures scalability, availability, and fault-tolerance capabilities.

4) ANALYTICS ENGINE

The analytics engine applies machine learning techniques to predict the current status of the parcel package (i.e., broken or not). To ensure scalability towards Big collections of data, it is designed on top of Apache Spark [27]. A thorough description of the data transformation and analytics steps are given in Section V.

B. FRONT-END SERVICES

The application front-end provides stakeholders with a graphical interface to visualize acquired data and prediction outcomes under various viewpoints. Two main functionalities are provided.

- *Multi-dimensional view of the historical data.* Views aggregate data at different abstraction levels and according to different facets.
- *Real-time updates.* Data and prediction outcomes are dynamically updated as soon as new entries are stored.

A set of predefined dashboards offer different data views according to the type of user interacting with the framework. The front-end service relies on various standards for visual analytics, among which Freeboard,[2] Grafana,[3] Kibana,[4]

[2]https://freeboard.io/
[3]https://grafana.com/
[4]https://www.elastic.co/products/kibana

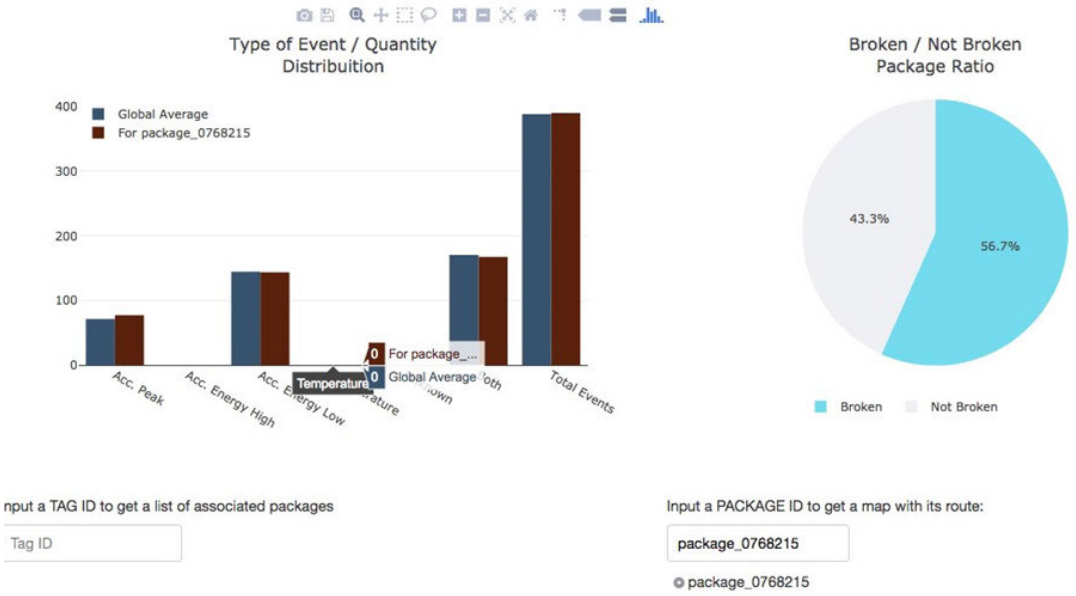


FIGURE 4. Example dashboards: event frequency and package status distributions.

ngx-admin.[5], and Plotly[6] All the visualization models require human interactions, because domain experts interact with the system and specify which kind of data they are really interested in.

For example, the developer dashboards show the key information about shipped packages and related events. Figure 4 shows a representative dashboard reporting the frequency distributions of specific events and the percentages of broken/safe packages according to the prediction outcomes. The statistics are computed separately for each package or averaged over all the packages associated with specific properties.

Users can also explore the routes of specific tags or packages (e.g., see Figure 5). Per-tag maps show the path of the package associated with the selected tag, whereas per-package maps draw the recorded package-route points along with the associated events and update the event distribution graph with information regarding the events. Figure 5 shows the tag routes associated with a couple of packages. Notice that segment ends may not exactly match a node due to missing annotations or approximated geo-location data. Segment colours indicate the likelihood of package breaks during that segment of shipment, according to the data analytics outcomes (green = safe, blue = broken). Different point colours are associated with different types of workers annotating the events.

V. DATA ANALYTICS

The data analytics process, depicted in Figure 6, consists of the following steps: (i) *Feature extraction*, which defines the features used to describe the events related to the packages. (ii) *Feature transformation*, which transforms the extracted features in order to make them more relevant to predict the package status. (iii) *Classifier training*, which learns classification models on the selected data. (iv) *Model application and tuning*, which apply the models trained at the previous step to predict the unknown package status. (v) *Classifier tuning*, which entails refining the previously trained model by exploiting the newly labeled data.

A. FEATURE EXTRACTION

The package-related events are collected through Apache Kafka (https://kafka.apache.org/), i.e., a distributed platform that builds real-time data pipelines and supports fault-tolerant streaming applications. An Apache Spark application [27] consumes the event messages of the Kafka streaming. The well-known big data platform Apache Spark allows the system back-end to accomplish many data preparation steps in a scalable way. This allows a large number of carriers to send data to the service at the same time.

At the service back-end, many different messages are collected by the Kafka queue manager. Specifically, the messages related to the position of the workers, those related the new takeovers from the couriers, and the messages reporting the events recorded by the tags. In the pre-processing phase, the messages are transformed to fit the logic schema of the Cassandra database where they will be stored. In some cases, the already available information is exploited to adapt the structure for new input data and ease the information retrieval process. For example, the events recorded from the tags are not associated to any position. Hence, to simplify the next data visualization and information extraction steps, the missing positions are filled up with the corresponding worker positions. For each recorded package event, its GPS position is set to the value of the time-weighted distance computed

[5]http://akveo.com/
[6]https://plot.ly/

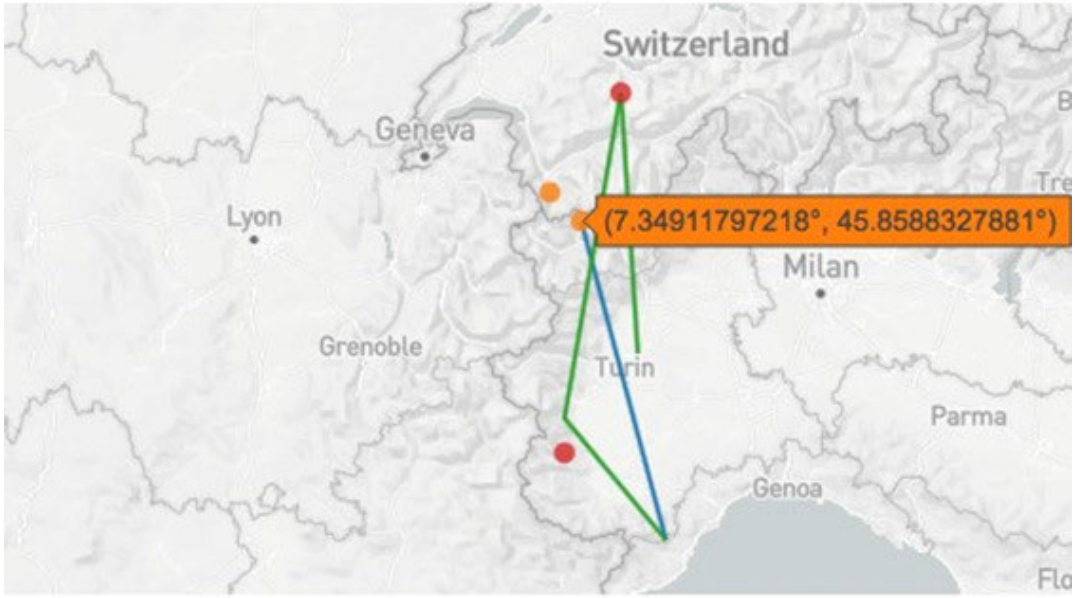


FIGURE 5. Example dashboard: tag routes over a map with predicted package status.

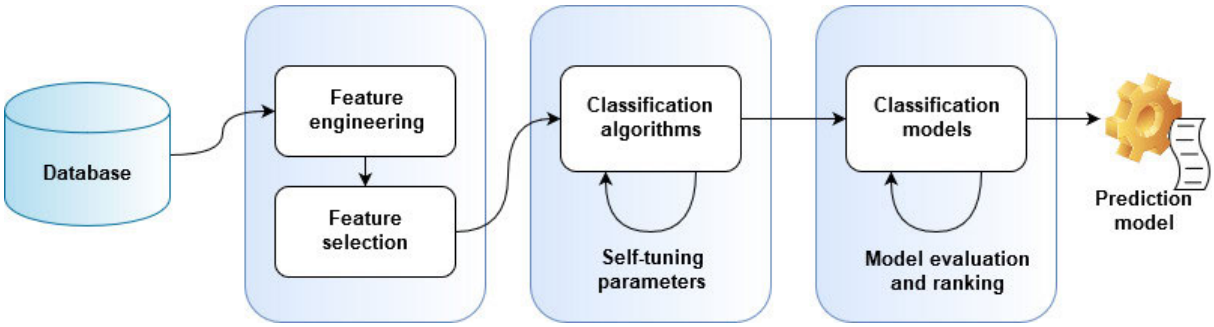


FIGURE 6. The analytics pipeline.

between the two time-closest GPS positions of the worker responsible for a given package transport.

In the transportation process, while downloading the event data, a new logical transportation segment is created every time a scan of the REDTag is performed. A segment indicates that the worker is now in charge of managing the package. Once processed and stored in the database, data will be processed by the analytics engine.

B. FEATURES ENGINEERING AND SELECTION

Event-related data are transformed and aggregated in order to make them suitable for the next prediction phase. Each event, related to an arbitrary package, can be modelled as a set of multidimensional time series $\mathcal{T}(p)$ associated with the package p. For example, a GPS event (see Section IV-A) is stored as a set of GPS positions related to the package. Since, in our context, all the spatial dimensions are deemed as equally relevant, each series of points is summarized by its L^2 norm. To synthetically describe the time series under multiple aspects, we first extract various descriptive features according to the methodology presented in [28]. Then, we filter the generated features according to their discriminating power to predict the target class by applying the feature selection approach presented by [29]. At the end of feature engineering and selection phase, package-related events are transformed from time series data to a set of discrete attributes that are fairly correlated with the class. The subset of selected features is enumerated in Table 1.

To further enrich the discrete data model, we group event-based data by package (i.e., by tagID), event type (regardless of the corresponding sub-events), and package status (i.e., the class attribute). Next, the aggregation functions enumerated in Table 2 are computed to describe the event-status relationships.

C. CLASSIFIER TRAINING, APPLICATION, AND TUNING

Classification is an established technique to learn predictive models from a set of labeled data (i.e., the training phase). The generated models are then applied to a set of unlabeled test records [30].

In our context, training and test data are collected from the non-relational dataset and temporarily stored in a table

TABLE 1. Feature definition.

Feature	Description
file	File name
tagID	Tag ID
userID	User ID
eventType	Event type
eventSubType	Sub-Event type
timestamp	Event timestamp
windowLength	Window Length
max	Maximum value of each time series
sumAbsValues	Sum of the absolute values of each time series
numElemOverMean	Number of data points, which are larger than the average value of each time series
var	Variance value of each time series
absEnergy	Absolute energy of each time series computed as the sum of the squared values
longestStrikeOverMean	Length of the longest consecutive subsequence in each time series, which is larger or equal to the mean
hasLargeStd	Boolean feature indicating that the standard deviation of each time series is bigger than half the difference between the maximal and minimal value
meanAbsChange	Arithmetic mean of absolute differences between subsequent time series values
numElemOverMeanStd	Number of data points, which are higher than the average value of each time series plus the standard deviation
Status	Classification Label

TABLE 2. Aggregated features.

Attribute	Description
countType1	Count of events of type 5.1 occurred for the package
countType2	Count of events of type 5.2 occurred for the package
countType3	Count of events of type 5.3 occurred for the package
countType4	Count of events of type 5.4 occurred for the package
min_ts	Smallest timestamp of the events occurred for the package
max_ts	Largest timestamp of the events occurred for the package
max_max	The maximum between the maximum values of the time series
mean_max	The mean between the maximum values of the time series
sum_sumAbsValue	Sum of the sum of the absolute values of time series
max_numOverMean	Maximum value between the number of points greater than the mean value of the time series, for each time series
mean_numOverMean	Mean value between the number of points greater than the average value of the time series, for each time series
sum_numOverMean	Sum of values between the number of points greater than the average value of the time series, for each time series
max_numOverMeanStd	Maximum number of data points, which are higher than the average value of each time series plus the standard deviation, for each time series
mean_numOverMeanStd	Mean number of data points, which are higher than the average value of each time series plus the standard deviation, for each time series
sum_numOverMeanStd	Sum of the data point values, which are higher than the average value of each time series plus the standard deviation, for each time series
max_longestStrikeOverMean	Maximum length of the longest consecutive subsequence in each time series, which is larger or equal to the mean
mean_longestStrikeOverMean	Mean length of the longest consecutive subsequence in each time series, which is larger or equal to the mean
sum_longestStrikeOverMean	Sum of consecutive subsequence points in each time series, which are larger or equal to the mean
max_var	The maximum between the variance values of the time series
mean_var	The mean between the variance values of the time series
sum_absEnerg	Sum of absolute energy of each time series computed as the sum of the squared values
max_largeStd	Boolean feature indicating that the standard deviation is bigger than half the difference between the maximal and minimal value
mean_meanAbsChange	Mean value of the arithmetic means of absolute differences between subsequent time series values
broken	Classification label, characterising the status of the package after the events occurred in the time series

with the structure reported in Table 2. Each data entry corresponds to the occurrence of a distinct event for a given package. The goal is to predict the status of the package (broken or safe) that has been tagged when the event occurred.

To tackle the classification problem, many different classification approaches have been proposed in literature (e.g., Bayesian classifiers [31], [32], decision trees [33], Support Vector Machines (SVMs) [34], [35], Neural Networks [36], ensemble methods [37], [38], and associative classifiers [39], [40]). Among the variety of possible solutions, we select those satisfying the following constraints tailored to the analyzed application context.

- **Scalability:** Since the number of package-related events is potentially very large, we focus on the algorithms for which a parallel and distributed version is available in Apache Spark MLlib library [41].
- **Applicability to heterogeneous data:** Since the event descriptors are partly numerical and partly categorical, we consider the scalable algorithms that support both attribute types.

In light of the constraints mentioned above, we have considered the MLlib implementations of the following classifiers: Gradient Boosting Classifier, Logistic Regression, and Support Vector Machines [41].

VI. EXPERIMENTAL RESULTS

In this section we summarize the results of the empirical evaluation conducted on a real-world dataset. The dataset stores information about the shipping of fragile items,

namely glasses. To collect data, packaged glasses have been made fall from different heights and directions with different speeds. The tags, placed all over the boxes, recorded all the accelerations and impacts experienced by the packages. For each fall, the condition of the glass inside the package has been checked, in order to assign a status label (i.e., safe or broken) to each sequence of signals gathered from the REDTag IoT-enabled device.

The dataset consists of 22,700 entries (7,840 of which are labeled as broken, whereas the remaining ones are labeled as safe). The experiments are aimed to empirically demonstrate that event-related information is actually correlated with the package status in a real case study.

We run the experiments on an Intel(R) Core(TM) i7-8550U CPU with 16 GB of RAM running Ubuntu 18.04 server. For most of the executed experiments, the execution time is in the order of tens of seconds for training the classification models, whereas it is negligible for label assignment.

This section is organized as follows. Section VI-A briefly describes the experimental design. Section VI-B shows a preliminary data visualization to qualitatively show the complexity of the addressed task. Section VI-C shows the results of the pairwise feature correlation analysis, while Section VI-D reports the results of classification process.

A. EXPERIMENTAL DESIGN

To quantitatively evaluate classifier performance in predicting package status, we apply a stratified 5-fold cross-validation strategy and compute the following performance metrics [30]:

- *Accuracy:* It is the percentage of events whose package status (*broken* or *safe*) has been correctly assigned.
- *Precision of class label broken:* It is the ratio of the number of events that have been correctly labeled as *broken* with respect to the total number of events labeled as *broken*.
- *Recall of class label broken:* It is the ratio of the number of packages that have been correctly labeled as *broken* to the total number of events that actually belong to class *broken*.
- *F1-score of class label broken:* It is the harmonic mean of precision and recall of class *broken*.

Since the main goal of predictive maintenance is to early detect package breaks, the event counts for accuracy computations are weighted by the relative class frequencies in order to properly handle imbalances among the two classes. Furthermore, for the class-specific metrics (i.e., precision, recall, and F1-score) we specifically focus on the evaluation metrics referred to the label *broken*.

B. PRELIMINARY DATA EXPLORATION

The scatter plot in Figure 7 shows the distribution of the training data points in a 3-dimensional space after Singular Value Decomposition (SVD). SVD is an established process for dimensionality reduction. It relies on a matrix factorization, which transforms the input data into a latent space where the key components (represented by the most relevant singular values) are preserved [30].

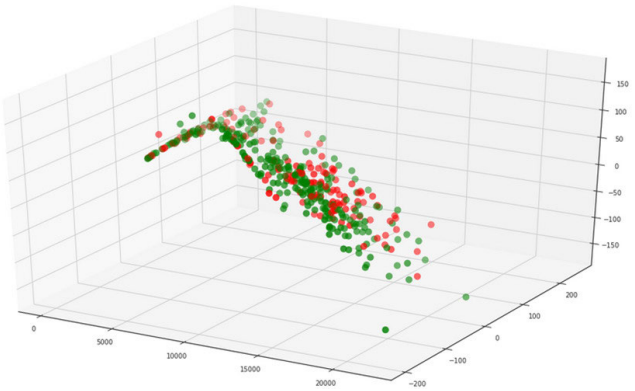


FIGURE 7. Scatter plot of the training data (class labels: broken, safe) in a Singular Value Decomposition (number of dimensions: 3).

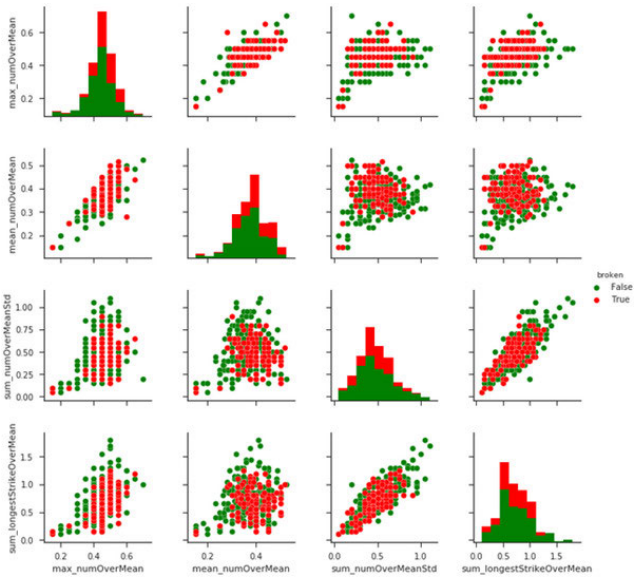


FIGURE 8. Scatter matrix of the most significant pairs of feature categories. Class value broken in red, class value safe in green.

To visualize the relative package positions in the latent space, points in Figure 7 are coloured according to their class membership. Specifically, points in red correspond to broken packages, whereas green ones are mapped to safe packages. The plot shows a mixed coloured point cloud, indicating the absence of a clear separation between broken and safe packages. This prompts the use of more advanced machine learning solutions to tackle the prediction task.

C. ANALYSIS OF THE PAIRWISE FEATURE CORRELATION

The scatter matrix in Figure 8 shows the dependencies among pairs of feature categories in the training dataset. The presence of strongly correlated pairs of descriptive categories (e.g., categories sum_numOverMeanStd and sum_longestStrikeOverMean) is deemed as redundant in the training data, because they provide roughly the same information. To tailor the model to the features that are most

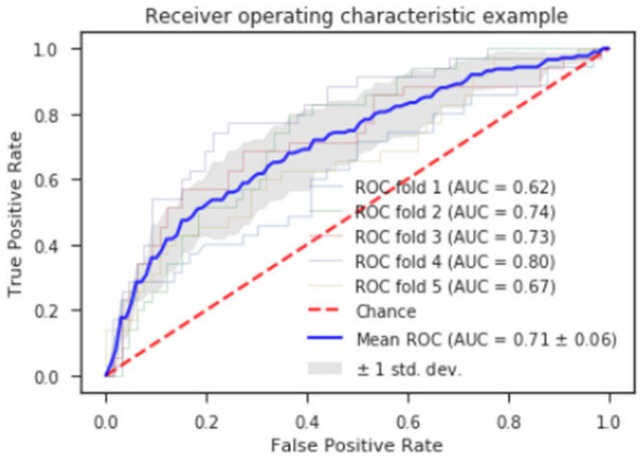


FIGURE 9. Receiver operating characteristic curve. Stratified 5-fold CV. gradient boosting classifier.

correlated to the class feature while avoiding data redundancy, we take one representative for each pair of categories with high pairwise correlation and, out of all the remaining features, we keep only those that are significantly correlated with the class according to the approach described in Section V-B.

D. CLASSIFIER PERFORMANCE

The best weighted accuracy values achieved by the Gradient Boosting (GBC), Support Vector Machines (SVM), and Logistic Regression (LR) Classifiers over all the performed runs are 74.2%, 72.8%, and 61.1%, respectively. The preliminary results show that the Gradient Boosting Classifier is the most accurate model to predict package breaks.

The recall values associated with label *broken*, i.e., GBC 78%, SVM 85%, 67%, indicate that the SVM model forecasts a larger number of actual glass breaks (+7%), but it generates also a higher number of false positive outcomes, i.e., the precision gap between GBC and SVM is 6%. It turns out that relying on SVM model predictions yields a more sensitive alerting system, which generates approximately 25% extra alerts compared to GBC. The optimal trade-off between model precision and sensitivity strongly depends on the impact of the generated alerts on the operational costs needed to apply package quality checks in the real scenario.

Figure 8 shows the Receiver Operating Characteristic (ROC) curves corresponding to each Cross-Validation fold of GBC, the mean ROC curve, and the baseline curve achieved by random choice. The ROC graph [30] plots the True Positive Rate (i.e., the percentage of entries labeled as broken and actually broken) vs. the False Positive Rate (i.e., the percentage of entries labeled as broken but actually safe). As expected, the achieved curves are all above the baseline for the performed runs.

VII. CONCLUSIONS AND FUTURE WORKS

In this paper, we investigate the application of machine learning models to address the data-driven prediction of courier package breaks in smart good transportation systems. We analyze data acquired by an IoT-enabled device monitoring the events collected by the courier employees during package shipping. It presents also a Big Data analytics framework to efficiently store and manage analyzed data as well as to generate and visualize the prediction outcomes. In the experimental evaluation, we explore the effectiveness of scalable classification models predicting package breaks in real time scenarios. The results show that Gradient Boosting Classifiers achieve desirable accuracy and recall performance.

The presented results are mainly focused on classifier performance. In our future research agenda, we aim to investigate the scalability of the framework in terms of memory used and communication costs in a Big Data scenario. Furthermore, we plan to further investigate the applicability of self-tuning approaches able to capture relevant changes in the predictive patterns as soon as they emerge in the acquired data.

REFERENCES

[1] L. Barreto, A. Amaral, and T. Pereira, "Industry 4.0 implications in logistics: An overview," *Procedia Manuf.*, vol. 13, pp. 1245–1252, Jan. 2017.
[2] D. Uckelmann, "A definition approach to smart logistics," in *Next Generation Teletraffic and Wired/Wireless Advanced Networking*, S. Balandin, D. Moltchanov, and Y. Koucheryavy, Eds. Berlin, Germany: Springer, 2008, pp. 273–284.
[3] 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. 23484–23491, 2017.
[4] D. Mcfarlane, V. Giannikas, and W. Lu, "Intelligent logistics: Involving the customer," *Comput. Ind.*, vol. 81, pp. 105–115, Sep. 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166361515300488
[5] M. Mabkhot, A. Al-Ahmari, B. Salah, and H. Alkhalefah, "Requirements of the smart factory system: A survey and perspective," *Machines*, vol. 6, no. 2, p. 23, Jun. 2018. [Online]. Available: https://www.mdpi.com/2075-1702/6/2/23
[6] K.-S. Wang, Z. Li, J. Braaten, and Q. Yu, "Interpretation and compensation of backlash error data in machine centers for intelligent predictive maintenance using ANNs," *Adv. Manuf.*, vol. 3, no. 2, pp. 97–104, Jun. 2015.
[7] D. Markudova, E. Baralis, L. Cagliero, M. Mellia, L. Vassio, E. Amparore, R. Loti, and L. Salvatori, "Heterogeneous industrial vehicle usage predictions: A real case," in *Proc. Joint Conf. Workshops EDBT/ICDT*, Lisbon, Portugal, Mar. 2019. [Online]. Available: http://ceur-ws.org/Vol-2322/DARLIAP_13.pdf
[8] S. Martinelli, "Interpretation and compensation of backlash error data in machine centers for intelligent predictive maintenance using ANNs," in *Proc. Abu Dhabi Int. Petroleum Exhib. Conf.*, vol. 3, Nov. 2016.
[9] F. Civerchia, S. Bocchino, C. Salvadori, E. Rossi, L. Maggiani, and M. Petracca, "Industrial Internet of Things monitoring solution for advanced predictive maintenance applications," *J. Ind. Inf. Integr.*, vol. 7, pp. 4–12, Sep. 2017.
[10] J. Lee, J. Ni, D. Djurdjanovic, H. Qiu, and H. Liao, "Intelligent prognostics tools and e-maintenance," *Comput. Ind.*, vol. 57, no. 6, pp. 476–489, Aug. 2006.
[11] E. Fadda, P. Plebani, and M. Vitali, "Monitoring-aware optimal deployment for applications based on microservices," *IEEE Trans. Services Comput.*, to be published.
[12] E. Fadda, G. Perboli, and R. Tadei, "A progressive hedging method for the optimization of social engagement and opportunistic IoT problems," *Eur. J. Oper. Res.*, vol. 277, no. 2, pp. 643–652, Sep. 2019.
[13] J. Lee, E. Lapira, B. Bagheri, and H.-A. Kao, "Recent advances and trends in predictive manufacturing systems in big data environment," *Manuf. Lett.*, vol. 1, no. 1, pp. 38–41, Oct. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2213846313000114
[14] J. Lee, E. Lapira, S. Yang, and A. Kao, "Predictive manufacturing system–trends of next-generation production systems," *IFAC Proc. Volumes*, vol. 46, no. 7, pp. 150–156, May 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667015356664

[15] 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 *Proc. IEEE Int. Conf. Parallel Distrib. Process. Appl., Ubiquitous Comput. Commun., Big Data Cloud Comput., Social Comput. Netw., Sustain. Comput. Commun. (ISPA/IUCC/BDCloud/ SocialCom/SustainCom)*, Dec. 2018, pp. 924–931.

[16] Y. Cui, S. Kara, and K. C. Chan, "Manufacturing big data ecosystem: A systematic literature review," *Robot. Comput.-Integr. Manuf.*, vol. 62, Apr. 2020, Art. no. 101861. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0736584519300559

[17] S. A. Asmai, A. S. H. Basari, A. S. Shibghatullah, N. K. Ibrahim, and B. Hussin, "Neural network prognostics model for industrial equipment maintenance," in *Proc. 11th Int. Conf. Hybrid Intell. Syst. (HIS)*, Dec. 2011, pp. 635–640.

[18] T. Niesen, C. Houy, P. Fettke, and P. Loos, "Towards an integrative big data analysis framework for data-driven risk management in industry 4.0," in *Proc. 49th Hawaii Int. Conf. Syst. Sci. (HICSS)*, Jan. 2016, pp. 5065–5074.

[19] Y. Zhang, S. Ren, Y. Liu, and S. Si, "A big data analytics architecture for cleaner manufacturing and maintenance processes of complex products," *J. Cleaner Prod.*, vol. 142, pp. 626–641, Jan. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959652616310198

[20] J. Murphree, "Machine learning anomaly detection in large systems," in *Proc. IEEE AUTOTESTCON*, Sep. 2016, pp. 1–9.

[21] M. Miskuf and I. Zolotova, "Comparison between multi-class classifiers and deep learning with focus on industry 4.0," in *Proc. Informat. (K&I)*, Feb. 2016, pp. 1–5.

[22] Y. Chen, F. Zhu, and J. Lee, "Data quality evaluation and improvement for prognostic modeling using visual assessment based data partitioning method," *Comput. Ind.*, vol. 64, no. 3, pp. 214–225, Apr. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166361512001753

[23] F. Ventura, S. Proto, D. Apiletti, T. Cerquitelli, S. Panicucci, E. Baralis, E. Macii, and A. Macii, "A new unsupervised predictive-model self-assessment approach that SCALEs," in *Proc. IEEE Int. Congr. Big Data (BigDataCongress)*, Milan, Italy, Jul. 2019, pp. 144–148, doi: 10.1109/bigdatacongress.2019.00033.

[24] N. Garg, *Apache Kafka*. Packt Publishing, Birmingham, U.K., 2013.

[25] R. Kimball and M. Ross, *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 3rd ed. Hoboken, NJ, USA: Wiley, 2013.

[26] A. Lakshman and P. Malik, "Cassandra: A decentralized structured storage system," *SIGOPS Oper. Syst. Rev.*, vol. 44, no. 2, pp. 35–40, Apr. 2010, doi: 10.1145/1773912.1773922.

[27] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, and I. Stoica, "Apache spark: A unified engine for big data processing," *Commun. ACM*, vol. 59, no. 11, pp. 56–65, Oct. 2016. [Online]. Available: http://doi.acm.org/10.1145/2934664

[28] M. Christ, A. W. Kempa-Liehr, and M. Feindt, "Distributed and parallel time series feature extraction for industrial big data applications," 2016, *arXiv:1610.07717*. [Online]. Available: https://arxiv.org/abs/1610.07717

[29] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, "Time series feature extraction on basis of scalable hypothesis tests (tsfresh—A Python package)," *Neurocomputing*, vol. 307, pp. 72–77, Sep. 2018.

[30] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to Data Mining*, 2nd ed. London, U.K.: Pearson, 2018.

[31] D. D. Lewis, "Naïve (Bayes) at forty: The independence assumption in information retrieval," in *Proc. ECML*, 1998, pp. 4–15.

[32] E. Baralis, L. Cagliero, and P. Garza, "EnBay: A novel pattern-based Bayesian classifier," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 12, pp. 2780–2795, Dec. 2013.

[33] R. J. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 1993.

[34] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, Apr. 2011.

[35] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proc. ECML*, 1998, pp. 137–142.

[36] G. P. Zhang, "Neural networks for classification: A survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 30, no. 4, pp. 451–462, Nov. 2000.

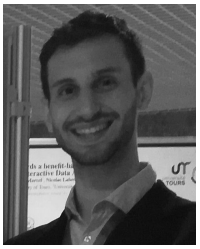
[37] T. Kam Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998, doi: 10.1109/34.709601.

[38] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[39] L. Cagliero and P. Garza, "Improving classification models with taxonomy information," *Data Knowl. Eng.*, vol. 86, pp. 85–101, Jul. 2013, doi: 10.1016/j.datak.2013.01.005.

[40] E. Baralis, S. Chiusano, and P. Garza, "A lazy approach to associative classification," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 2, pp. 156–171, Feb. 2008, doi: 10.1109/tkde.2007.190677.

[41] 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, pp. 1235–1241, Jan. 2016. [Online]. Available: http://dl.acm.org/citation.cfm?id=2946645.2946679



STEFANO PROTO received the master's degree in computer engineering from the Politecnico di Torino, Italy, in 2018.

His research interests are the improvement and the tuning of machine learning algorithms, with specific focus to industry 4.0, the IoT, and NLP fields.

EVELINA DI CORSO received the master's degree in mathematical engineering and the Ph.D. degree in computer engineering from the Politecnico di Torino, Italy, in 2015 and 2019, respectively.

Her research interest includes the design and the development of self-learning methodologies in applications of both structured and unstructured data.

DANIELE APILETTI received the master's and Ph.D. degrees in computer engineering from the Politecnico di Torino, Italy, in 2005 and 2008, respectively.

He has been an Assistant Researcher with the Database and Data Mining Group, Politecnico di Torino, since 2009. His research interests are in the field of NoSQL databases, large-scale data mining techniques, and big data machine learning, with specific focus on network-traffic, sensor-data, and industrial applications.

LUCA CAGLIERO (Member, IEEE) received the master's degree in computer and communication networks and the Ph.D. degree in computer engineering from the Politecnico di Torino. He has been an Assistant Professor with the Dipartimento di Automatica e Informatica, Politecnico di Torino, since January 2016. His current research interests are in the fields of machine learning, textual mining, and deep NLP. Specifically, he has worked on text summarization, classification, and association rule mining.

TANIA CERQUITELLI (Member, IEEE) received the master's degree (Hons.) in computer engineering from the Politecnico di Torino, Italy, in 2003, the master's degree (Hons.) in computer science from the Universidad De Las Américas Puebla, Mexico, in 2003, and the Ph.D. degree from the Politecnico di Torino, in 2007. She has been an Associate Professor with the Department of Control and Computer Engineering, Politecnico di Torino, since March 2018. Her research activities have been mainly devoted to fostering and sharing research and innovation on automated data science and machine learning in different real-life settings. She has been involved in many European and Italian research projects addressing different research issues related to machine learning and data analytics for energy-related data and Industry 4.0.

GIOVANNI MALNATI received the degree in electronic engineering and the Ph.D. degree in computer and system engineering from Politecnico di Torino. He is currently a Researcher with the Dipartimento di Automatica e Informatica, Politecnico di Torino. His research activities include software and network technologies for distributed and pervasive system support, vehicular network applications, indoor positioning systems, and multimedia technologies supporting e-learning environments.

DAVIDE MAZZUCCHI received the master's degree in computer and communication network engineering from the Politecnico di Torino. He is currently a system Architect and the Technical Leader of embedded software and the IoT projects with Zirak Srl. He also leads the overall production cycle of innovative self-designed the IoT platforms for predictive logistics and maintenance, with strong focus on innovation and research in cutting-edge machine learning and big data management technologies.