Bosch's Industry 4.0 Advanced Data Analytics: Historical and Predictive Data Integration for Decision Support

João Galvão[1[0000-0003-4263-8726]], Diogo Ribeiro[1 [0000-0002-6270-3962]], Inês Machado[1[0000-0003-2332-166X]], Filipa Ferreira[1[0000-0002-0527-046X]], Júlio Gonçalves[1[0000-0001-8265-3600]], Rui Faria[1[0000-0002-7181-543X]], Guilherme Moreira[2 [0000-0001-6139-0071]], Carlos Costa[1[0000-0003-0011-6030]], Paulo Cortez[1[0000-0002-7991-2090]], Maribel Yasmina Santos[1[0000-0002-3249-6229]]

[1]ALGORITMI Research Centre, University of Minho, Guimarães, Portugal
[2]Bosch Car Multimedia, Braga, Portugal
{joao.galvao, carlos.costa, pcortez, maribel}@dsi.uminho.pt
{a80365, a78447, a79136, a78977}@alunos.uminho.pt
diogo.ribeiro08@gmail.com, guilherme.moreira2@pt.bosch.com

Abstract. Industry 4.0, characterized by the development of automation and data exchanging technologies, has contributed to an increase in the volume of data, generated from various data sources, with great speed and variety. Organizations need to collect, store, process, and analyse this data in order to extract meaningful insights from these vast amounts of data. By overcoming these challenges imposed by what is currently known as Big Data, organizations take a step towards optimizing business processes. This paper proposes a Big Data Analytics architecture as an artefact for the integration of historical data - from the organizational business processes - and predictive data - obtained by the use of Machine Learning models -, providing an advanced data analytics environment for decision support. To support data integration in a Big Data Warehouse, a data modelling method is also proposed. These proposals were implemented and validated with a demonstration case in a multinational organization, Bosch Car Multimedia in Braga. The obtained results highlight the ability to take advantage of large amounts of historical data enhanced with predictions that support complex decision support scenarios.

Keywords: Big Data Warehousing, Advanced Analytics, Machine Learning, Industry 4.0.

1 Introduction

There is a huge growth in the data that is generated, being a challenge to deal with this rapid growth, as well as with the complexity of the data and its interconnection [1]. Big Data involves a large set of data in which its size and structure are not properly handled by traditional database systems, such as relational databases [2]. These large datasets usually integrate richer data for decision support, with more details about behaviours,

activities, and events, providing huge diversity of data and requiring shorter response time [3]. To take full advantage of the strategic potential of the large volume of data provided by organizations, Big Data Analytics is necessary [4]. It includes procedures to extract relevant information from a large volume of data. Its main purpose is to enable organizations to make better decisions according to their mission and objectives. It also helps in solving problems quickly, providing relevant and valuable insights that can bring a competitive advantage to the organization [2].

Machine Learning is a technique for the analysis of Big Data, which consists in detecting relationships and predicting future behaviours through the modelling process based on a large set of data [2]. The use of Machine Learning algorithms with predictive capabilities in Big Data can promote the discovery of new knowledge and bring additional value to organizations [1]. The effective use of data has been increasing competitiveness and economic growth in various industries, including manufacturing [5].

This work is developed under a partnership between Bosch Car Multimedia in Braga and Academia, which aims to create new scientific and technological knowledge to achieve the company's competitiveness goals by improving its main industrialization processes. This will allow a fast adaptation of the company to new market demands. Due to the complexity of both areas, Big Data Analytics and Machine Learning, and all the challenges that need to be faced to combine the analysis of huge amounts of historical data with predictions, this paper has as main goal the design, implementation, and evaluation of an advanced data analytics platform for supporting complex decision support environments. A central component in this architecture is its Big Data Warehouse (BDW), the supporting data system, ensuring the integration of all the data (historical and predictive) in a consolidated and coherent way. This architecture and the data modelling method here presented are the main contributions of this work.

This paper is structured as follows. Section 2 summarizes related work in the field, namely the development of BDWs, the adoption of Machine Learning techniques, and their integration in complex decision processes. Section 3 presents the proposed architecture for advanced data analytics, describing its various components and the supporting technologies, and the data modelling method. Section 4 describes the demonstration case, the screwing case, outlining its motivation and purpose. It also presents the supporting data model and the Machine Learning model, how the data integration and data flows were implemented, and some data visualizations for advanced data analytics. Section 5 concludes with some remarks and guidelines for future work.

2 Related Work

Since 1956, with greater emphasis on the last decade, the quantity of data has been growing exponentially and, as such, some challenges have appeared relatively to the storage and analysis of that data [6]. Important contributions in the past years had changed the databases field. Many organizations, such as Facebook, Google, and others, have had a really hard task to analyse an unprecedented amount of data that is not necessarily in a format or structure that makes it easy to analyse [7].

Big Data is mainly related with enormous amounts of unstructured data produced by high-performance applications that can range from computing applications to medical information systems. The data that is stored in this fashion, and its processing, has some specific characteristics and needs, such as: i) large-scale data, which refers to the size of the data repositories; ii) scalability issues, due to the vast amount of data and the performance concerns in its processing; iii) supporting Extraction-Transformation-Loading (ETL) processes, handling the input raw data in order to reach the required structured data; and, iv) analytical environment, designing and developing user-friendly analytical interfaces in order to extract useful knowledge from data [8].

Data-intensive systems are built to consolidate and make available relevant information for decision support. In them, data from different sources require a complex process of data integration that ensures a unified and coherent view of the organizational or application domain data [9]. As challenges such as volume, variety, or velocity emerge, Data Warehouses or other data storage systems require performant solutions able to deal with these data characteristics [10]. Big Data techniques and technologies support mixed and complex analytical workloads (e.g., streaming analysis, ad hoc querying, data visualization, data mining, simulations) in several emerging contexts [11].

Research in Big Data Warehousing [9] has proposed a structured approach for the design and implementation of BDWs, mainly focused in modelling highly autonomous objects, addressing performance issues, that integrate the relevant data to answer a specific analytical question. These objects are named Analytical Objects and can include both factual and predictive attributes. They can be integrated with Complementary Analytical Objects (to share data between several Analytical Objects), Special Objects (to normalize common Date, Time, or Spatial attributes), and Materialized Objects (to physically implement views that enhance performance).

Data Science techniques must be able to extract unknown features from data, to improve the value of the data itself, making it easier to understand behaviours, optimize processes, and improve scientific discovery. Big Data takes advantage of data analytics and Machine Learning, both being key steps for enhancing the value of data [12]. The integration of Machine Learning-based predictive applications in Big Data contexts has been proposed to address the challenges that emerge in complex decision-support environments characterized by a vast amount of data. The work of [13] aims to prevent losses caused by faults in assembly lines with a real-time monitoring system that uses data from IoT-based sensors, Big Data processing, and a hybrid prediction model.

An architecture that can automate and centralize data processing, health assessment, and prognostics is present in [14]. This architecture covers all necessary steps from acquiring data, its processing and presenting it to the users, supporting decision making. The work of [15] presents an architecture that facilitates the task of analysing and extracting value from Big Data using Hadoop-based tools for Machine Learning. This architecture supports batch and streaming processing modules, with Machine Learning tools and algorithms, so that developers take advantage of them to carry out tasks such as prediction, clustering, recommendation, or classification. Also, analytical dashboards present the results of the batch analysis and display them to the users. In [16], processing tools available in the Hadoop and Spark ecosystems, as well as optimization techniques, are combined in wind energy resource assessment and management. The

work of [17] presents an architecture that includes the dimensions of data capture, processing, storage, visualization and decision-aid through Machine Learning, leaving for further research, the implementation of the proposed architecture.

In summary, several works combine Big Data with Machine Learning. However, the work presented in this paper addresses this challenge by proposing a data modelling method and an architecture that has a BDW as its central data system, integrating historical and predictive data. This integration directly supports an advanced decision-support environment that can process large datasets combining historical data with predictions obtained from models that learned from that historical data.

3 Proposed Architecture and Data Modelling Method

Research in Big Data Analytics and Machine Learning is usually done by different teams, who independently work in providing analytical means to analyse the available data. This work aims to integrate the scientific and technological contributions from both fields, supporting the integration of predictions to enrich a BDW that is used in an advanced data analytics environment that assists decision-makers for better decisions.

The proposed architecture (Fig. 1) creates a unified environment between Machine Learning processes and the BDW, as the main storage component, in Big Data contexts. Besides the storage itself, this architecture allows the monitoring of the Machine Learning models and the BDW, expanding the analytical scope beyond the decision-making at the business level - it is now possible to establish performance metrics for the BDW and the Machine Learning models and monitor them over time.

3.1 Components and Supporting Technologies

The advanced data analytics architecture (Fig. 1) is composed of three main components: *Data Sources*, *Big Data Cluster*, and *Visualization Tools*.

The Data Sources can be of different types, depending on the data they produce/handle: data can be structured, semi-structured, or unstructured. Besides this classification, the data sources may present data that is produced at different speeds, with different sizes and formats, thus justifying the context of Big Data [18].

The *Big Data Cluster* component integrates two areas, which are *Data Lake* and *Big Data Warehouse*. A distinction in data storage was made to accommodate analytical data and non-analytical data. The *Data Lake* is used to support the storage of any kind of data/processes/models such as Raw Data, Data Pipelines, Machine Learning Models, among others. The *Big Data Warehouse* is a storage ecosystem supporting the storage of data modeled as Analytical Objects, representing highly independent and autonomous entities with focus on analytical subjects in terms of decision support [9].

In the proposed architecture, the *Data Lake* has three subareas, the *Standard Raw Data*, the *Data Pipelines Repository*, and *the Machine Learning Models Repository and its Interfaces*. The purpose of the *Standard Raw Data* subarea is to standardize the data and its access, so that throughout the system data follows the same naming structure, making it easier for all users to understand and use. To be efficient and coherent, this

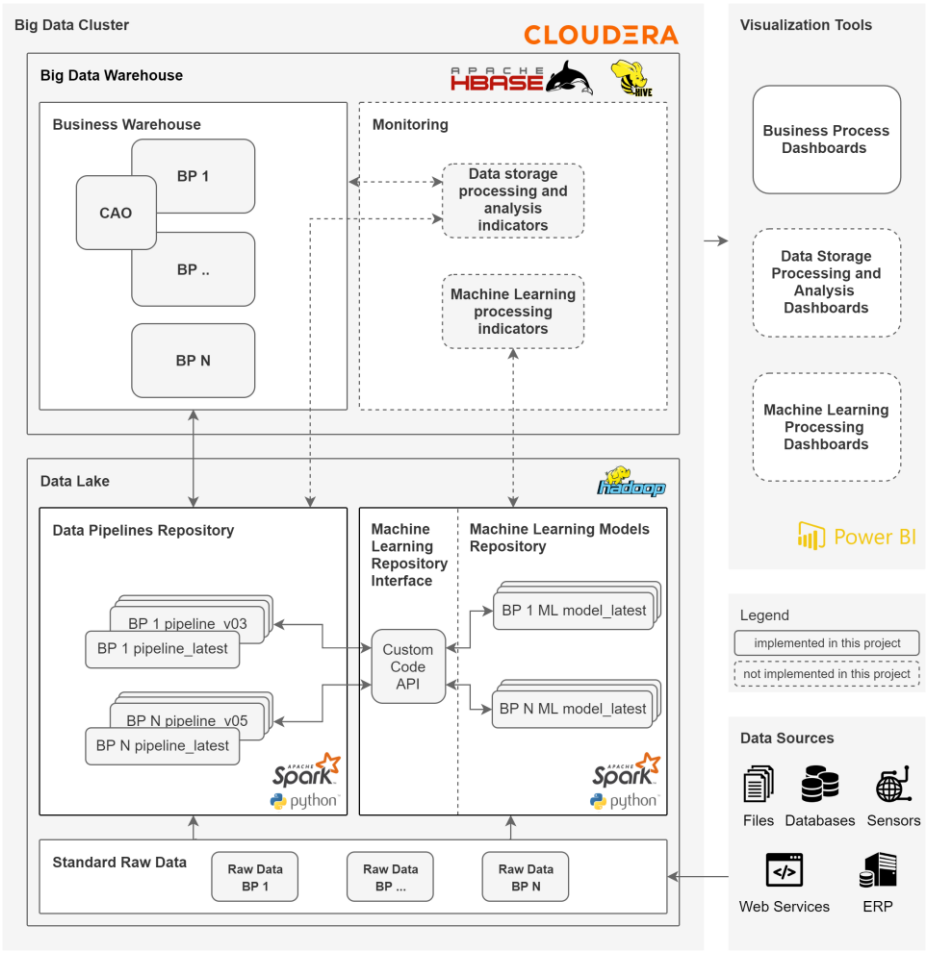

Fig. 1. Advanced Data Analytics Architecture

standardization needs the definition of a set of basic rules that must be applied to all the data that is here stored. These rules should follow the DATSIS principles [19], which enable data to become Discoverable, Addressable, Trustworthy, Self-describing, Interoperable, and Secure. For example, this proposal includes rules for standardization of the attributes' name, sharing the information about the BDW and Data Lake, and ownership in the Organization Wiki, among others included in organization directives.

The *Standard Raw Data* is used to feed two distinct, interrelated subareas, namely the *Data Pipelines Repository* and the *Machine Learning Models Repository and its Interfaces*. More specifically, data is stored in different sets linked to their Business Processes (BPs), where can be accessed in a Jupyter Notebook [20], which allows it to be read in the form of a Spark Dataframe [21]. These technologies are examples and

were the ones used in this implementation. Nevertheless, other technologies with similar purposes can be used, as the technological landscape is quite diverse in this field. The same applies for the other technologies mentioned throughout this paper.

Data Pipelines in the *Data Pipelines Repository* prepare, transform and enrich data according to the defined data model, creating one or several tables in the BDW. These tables are the physical implementation of the modeled objects. BPs have their data, but they also use data that can be shared between them. This data that can be shared by several BPs is stored inside the CAO (Complementary Analytical Objects) folder of each BP. The Machine Learning models in the *Machine Learning Models Repository* provide predictions of events in the data, which brings a competitive advantage for the decision-making process. The modeled Analytical Objects integrate historical attributes with predictive attributes obtained from trained Machine Learning models. To access these predictions, an interface needs to be established between the Data Pipelines and the Machine Learning models. This interface is based on a class, developed in PySpark [22], which encodes a series of functions that allow a Machine Learning model to run on the Spark Dataframe. Thus, in the Data Pipelines development environment (in this case in Jupyter Notebooks), another notebook containing the Machine Learning Class is invoked. After invoking and correctly importing it, the functions in it are applied to the Spark Dataframe that contains the *Standard Raw Data*. The output is a Spark Dataframe, which contains the predictions for the processed events. Once the output of the Machine Learning model is obtained, it is integrated into the Data Pipeline. After integrating the predictive outputs, the Analytical Object including the historical and predictive attributes is stored in the BDW as a Hive table.

The BDW is organized according to the purpose of the data and integrates two distinct subareas: *Business Warehouse* and *Monitoring*. This division is important to efficiently store data regarding the BPs and the performance of the BDW and the Machine Learning models. Besides training Machine Learning models and using them, or creating Analytical Objects and storing them in the BDW, their evolution must be monitored over time so that these components can be improved, updated, and maintained. Otherwise, the system can become obsolete or not address performance requirements in an industrial context. For the Machine Learning models, for example, due to the volatile nature of the data in a business activity, the data that is used for training the models can change and the models need to evolve to meet the new data needs, thus obtaining more accurate predictions.

Once all the data has been integrated and properly stored into the BDW, it is possible to analyse it in the *Visualization Tools* component. This includes data visualizations that support analytical tasks with associate indicators regarding Machine Learning (such as accuracy, for instance), Data Processing (such as processing time, for instance), and Analytical Objects (such as the number of records, for instance). This component foresees analytical dashboards for the analysis of the different BPs, and for the monitoring of the Data Storage Processing and Analysis and the Machine Learning models. In the work here presented, the visualizations were implemented in PowerBI. Although the architecture presents the *Monitoring* subarea and the related visualizations, they are considered future work and for this reason are not described in this paper.

3.2 Data Modelling Method

For the design and implementation of a BDW, a data modelling approach was pursued, by following specific steps. This modelling approach extends the one presented in [9] and addresses the evolution of the BDW by integrating new BPs or domains of analysis as needed. When new domains need to be added for analysis, a set of steps must be performed. Fig. 2 summarizes, with a simple example, the proposed steps (ST).

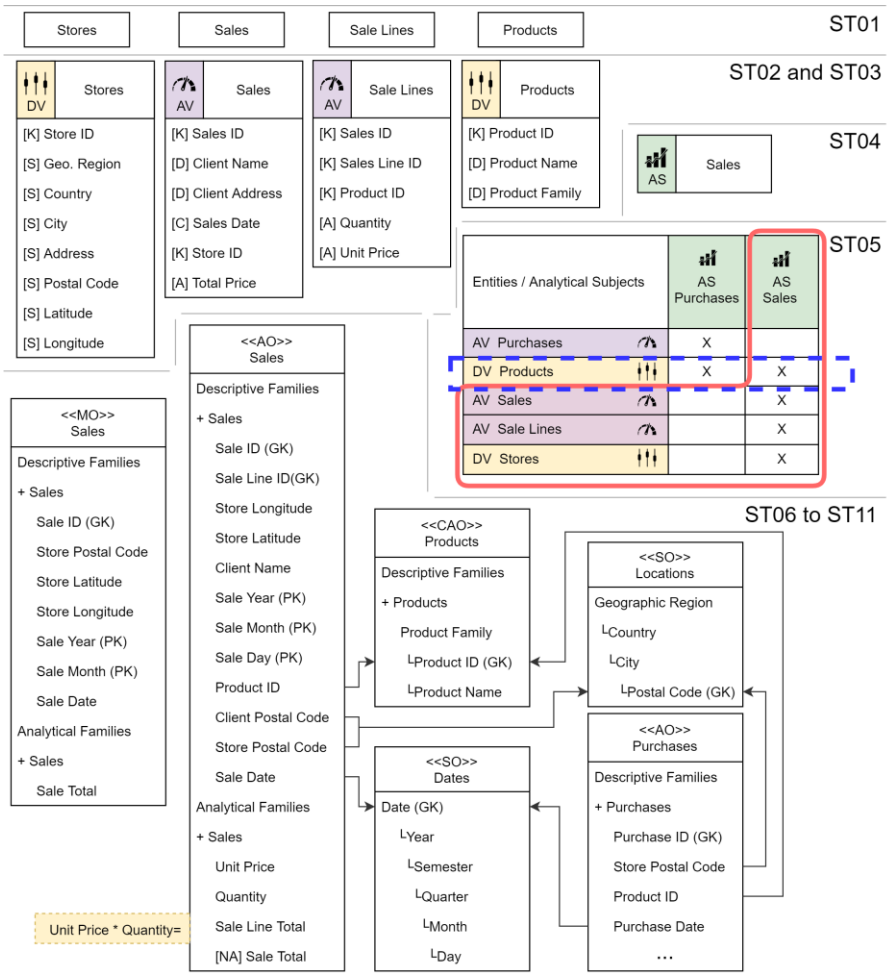

Fig. 2. Summary of the data modelling steps

ST01 – Identify the relevant entities of the domain under analysis. These entities can be identified by domain specialists through their empirical knowledge about the organization, using existing conceptual models (e.g. ERDs), logical models (e.g. star-schemas) or others, or even directly from data sources.

ST02 – Classify entities with Analytical Value (AV) or Descriptive Value (DV). Typically, entities with AV have two characteristics: (1) provide the main business or analytical indicators that are needed for decision support; (2) have high cardinality, as the growth rate of rows is considerably higher than the one verified in entities with DV. Entities with DV are used to provide analytical context and enable different levels of detail in the analysis. Nevertheless, it is important to mention that this method is based on a goal-driven approach, so the result of this classification depends on the domain in analysis. Thus, if the domain changes, the classification can also change.

ST03 – Identify and classify the attributes of the entities. Associate the relevant attributes of each entity classified in ST02. Attributes can be classified as: Descriptive (D); Analytical (A); Date (C); Time (T); Spatial (S); Key (K). Take into consideration that some operational systems have attributes that do not add value to analytical systems. Examples of that are attributes that always have the same value, are completely null or that are not useful for analysis. Those attributes should not be considered.

ST04 – Identify the Analytical Subjects (ASs). Based on the analytical requirements of the application domain given by the decision-makers, the analytical subjects emerge from the entities with AV. They can be a subset of them, all, or a relationship between two entities with AV. For example, in the sales domain, sales and purchases (from previous iteration) were classified as entities with AV and products as an entity with DV. Fig. 3 presents three different results for step ST04 according to the analytical requirements. If the focus of analysis, for now, is only about sales, then the AS will be Sales (case 1). If the focus will be the analysis of sales and purchases separately, then the result of ST04 will be the AS Sales and the AS Purchases (case 2). Another possible result is Case 3, where the focus is the sales and purchases in an integrated way. The result is an AS Sales with Purchases that, in future steps, will denormalize purchases to sales.

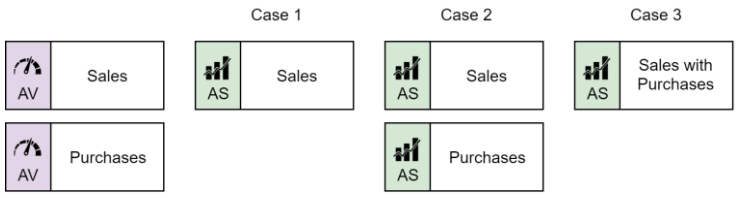

Fig. 3. Types of Analytical subjects

ST05 – Define the relationships matrix. Based on the domain knowledge, the relationships matrix needs to be defined or updated, mapping each AS with the available entities (AV or DV). In this mapping, the granularity of the AS cannot be changed by the denormalization of the mapped entities.

ST06 – Identify the Special Objects (SOs). Temporal and/or spatial attributes point to the need for SOs that include the calendar, temporal or spatial descriptive attributes that are relevant in the application domain. Special care is needed in Temporal and Spatial objects. These objects are used to normalize this kind of information in the BDW. It is not recommended to use attributes with high granularity (i.e. seconds, latitude or longitude) as they will highly increase the number of records in these tables.

ST07 – Define the Analytical Objects (AOs) and Complementary Analytical Objects (CAOs). Through the relationships matrix obtained in ST05 and the mapping of attributes in ST03, AS are classified as AOs and the entities that are related to the AS are denormalized to the AOs or included in the SOs. AOs are characterized by being autonomous objects in terms of processing and by answering specific domain questions, based on a subject of interest for analytical purposes. They are highly denormalized structures that can answer queries without the constant need of joins with other data sources. The logic representation of AOs is divided into families, namely Descriptive, Analytical and Predictive Families. The attributes of the entities with AV classified as descriptive or analytical are placed inside the respective family. CAOs emerge from the relationships matrix. Without existing a strict threshold, if an entity is shared by multiple AS and that number tends to grow, then it should be considered as CAO.

ST08 – Identify the Granularity Keys (GKs). The GKs represent the level of detail of the records to be stored in an AO and integrate one or more descriptive attributes that can uniquely identify a record. Each object in the BDW needs to have a GK.

ST09 – Identify the Partition Keys (PKs). The physical partitioning scheme applied to the data is normally made through date, time or geospatial attributes, that fragment the AO into lower size files, that can be accessed individually, enabling the loading and filtering in hourly/daily batches for specific regions or countries. As an example, Hive does not properly deal with a large number of small files, so it is important to choose the appropriate PK, in other to avoid unnecessary fragmentation. Although analytical attributes can be used to form a PK, that is not recommended. These keys are relevant to increase the system performance.

ST10 – Identify the Non-Additive (NA) Analytical or Predictive Attributes. As AOs are highly denormalized structures, they can have Analytical or Predictive Attributes that do not depend of the global GK. When numerical, those attributes are classified as NA as they cannot be aggregated with a SUM in a query that uses a GROUP BY, for example.

ST11 – [Optional] Define Materialized Objects (MOs). To improve the response querying time of the BDW, sometimes is useful to create MOs. MOs are usually created to answer specific needs of the user, joining the data of one or several objects and aggregating that data by a set of attributes.

4 Industrial Demonstration Case: The Screwing Case

4.1 Motivation and Goal

The industrial facility in which this work took place is used for the development and assembly of automotive instrument clusters with the help of specialized tools and personnel. This plant focuses on optimizing assembly and testing procedures due to the critical nature of these processes for the business goals. The assembly of an instrument cluster is an extensive procedure with many checkpoints that are not the object of study in this paper. Instead, we will focus on one of the final phases where the plastic housings are combined with either printed circuit boards (PCB) or plastic parts. Bonding plastics or electronics to plastics can be achieved via a multitude of techniques that involve adhesives, welding, or the use of fasteners. Validating a fastening procedure is a difficult task as many variables are at play at the same time. The process experts develop screw tightening programs with the help of the handheld driver manufacturers to perform a fastening process and overcome most of the problems inherent to the use of this bonding technique. The settings specified on this program are used as a baseline to which we compare the actual values and assess the success of the fastening procedure. The process starts when the operator guides a handheld driver to a feeder which is always on, not controlled by the developed program. Once a screw is loaded on the screwdriver bit, the operator is guided through the tightening sequence with the aid of instructions carefully illustrated on a monitor above the station. Each inserted screw results in a Good or Fail (GoF) message on the screen which indicates whether the fastening succeeded or not. Depending on the result, two different actions are triggered: on the success, the display instructs the operator to transfer the part to the next station; on failure, the operator is instructed to stop the procedure and the process data is uploaded to a remote server where it will be thoroughly analysed by an expert tool which compares the actual results against a defect catalog. This catalog is developed and maintained by experts who constantly add new rules to accommodate new products and fault modes. One major drawback of this piece of software is its lack of scalability and its constant need for updates. With the use of Machine Learning techniques, supported by a BDW with vast amounts of historical data, we identified two models that can correctly identify good and bad screw tightening curves and provide various insights on the motives for such results.

4.2 Supporting Data and Machine Learning Models

To support the identified methods, data must be prepared and fed the models in a structured manner. The structure of data has the characteristics detailed next. For each part number *p* (represented by a unique identifier) we have multiple distinct serial numbers (*sn*). Each serial number contains a set of records regarding control procedures conducted on the shop floor, spread across multiple machines. One of these control procedures is the screw tightening procedure validation. In the data *granularity*, Fig. 4, *i* represents a screw fastening procedure where $i \in \{1,2,...,N\}$ and *N* represent the total

number of screws for a specific *p* and *sn* pair. Each *i* is composed of $k \in \{1,2,3,...,K\}$ observations categorized by *a* attributes ($a = 44$).

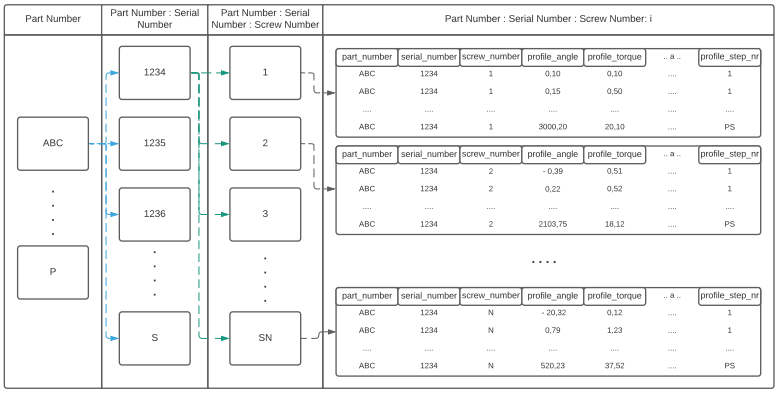

Fig. 4. Data Outlook – The screwing case

The focus of the analysis is on the *profile_angle* and *profile_torque* attributes as they allow us to visualize a fastening process in a 2D space (Fig. 5). Although this dataset is comprised of time-series data, we are using the angle variable ($\alpha_{i,k}$) as our sequential or temporal measure of the fastening as in most cases its values increase with the process duration.

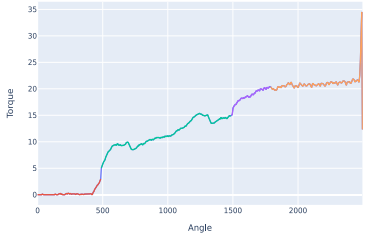

Fig. 5. Example Screw Tightening Curve

Several machine learning models were trained for this demonstration case [23] but only the two best performing unsupervised models were selected and the predictions included in the BDW, namely the ones that used the Isolation Forest and the Autoencoder. The Isolation Forest (iForest) leverages a clear distinction of characteristics of anomalous points which are present in fewer quantities and numerically different to normal instances and isolates them from normal points. Based on this principle, this anomaly detection algorithm is built upon a tree structure that attempts to isolate instances and then evaluate their normality. Anomalous instances tend to be isolated more easily as fewer features can describe them, forcing them to be closer to the root of the tree. At runtime, multiple trees are generated for a given dataset which forms an ensemble model - the iForest [24]. Normal points are isolated from anomalies that will, on average, have shorter path lengths.

Autoencoder (AE) is a type of unsupervised learning technique widely used for anomaly detection, image denoising, and feature extraction. AEs [25] are particularly strong in compressing and encoding high-dimensional data into a lower-dimensional space. This is achieved by imposing a bottleneck in its architecture which forces the neural network to create a compact representation (latent space) of the original input. Aside from this intermediate step, the AE is composed of two main stages: an encoding stage, where the input data is compressed using a specific number of features that describe the dataset, and a decoding stage where the model tries to recreate the original input with the smaller number of features present in the latent space. In this demonstration case, normal pairs of angle and torque values $(\alpha_{i,k}, \tau_{i,k})$ are provided as inputs and reconstructed $(\hat{\alpha}_{i,k}, \hat{\tau}_{i,k})$ pairs are generated by the model. Evaluating each pair (i,k) is calculated by computing the Mean Absolute Error (MAE) [26]:

$$MAE_{i,k} = (\left|\alpha_{i,k} - \hat{\alpha}_{i,k}\right| + \left|\tau_{i,k} - \hat{\tau}_{i,k}\right|)/2 \quad (1)$$

This reconstruction error $(d_{i,k} = MAE_{i,k})$ between the output and the input is then used as a decision score where greater reconstruction errors denote a higher anomaly probability.

4.3 Data Model

In this demonstration case, the Screw data model was identified following the steps presented in subsection 3.2 and integrates one Analytical Object (AO Screws), two Special Objects (Dates and Locations), and one materialized object (MO Screws). Due to confidentiality reasons, Fig. 6 only presents MO Screws as only this object is used to feed the dashboards here presented.

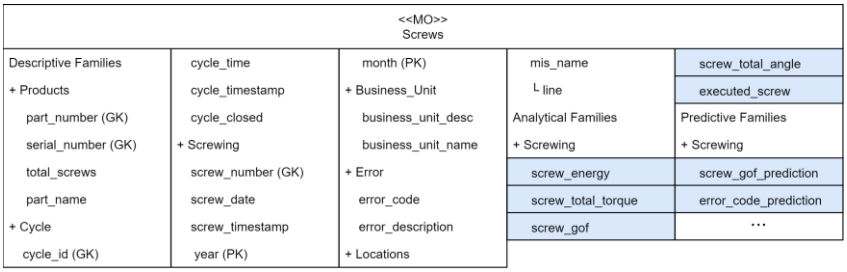

Fig. 6. MO Screws

AO Screws will provide the necessary analytical information to MO Screws so that predictions can be made in the backend, and these predictions are stored in MO Screws along with other relevant attributes. It is important to mention that AO Screws has, for each screw, more than 400 records, and the MO Screws only has one record for each screw with the aggregated data. In the proposed model, the attributes highlighted in blue are considered NA in the AO Screws, but not in the MO Screws.

4.4 Integration and Flows

All the available data sources were integrated in the Data Pipelines that load the historical data of the screws. Once the necessary standardization has been made to the data, it is stored in the *Standard Raw Data Screw* folder. The screw data stored in the *Standard Raw Data* was used to train and optimize the prediction models. Those prediction models are stored in the *Machine Learning Models Repository* and mapped in the corresponding Application Program Interface (API).

To feed the Business Warehouse another Data Pipeline is created. This pipeline loads the screw data from the *Standard Raw Data*, and for each set of Part Number, Serial Number, Cycle ID, and Screw Number, the Machine Learning API is called to predict the result of the GoF test. With the prediction results, a Dataframe is created containing the screw data from the *Standard Raw Data* grouped by Part Number, Serial Number, Cycle ID, and Screw Number, along with the GoF prediction. After that, the Dataframe is stored in a Hive Table that matches the *Analytical Object Screw* (AO Screws), an Analytical Object modeled for this BP and that integrates all the data relevant to support the decision support needs in this industrial plant. The decision process is supported by several analytical dashboards available in the *Visualization Tools*. As proposed in the architecture, all pipelines are stored in the *Data Pipelines Repository*.

4.5 Decision Support Dashboards

Dashboards are key elements in the daily work of the decision-makers, so their design was achieved with the engagement and validation of the final users.

As a requirement for this demonstration case, decision-makers must have a set of dashboards that present macro visualizations of the screwing process, as well as more detailed ones capable to show the results of the GoF test of each screw. All the dashboards developed for this demonstration case have two main areas, an L shape bar along top and left side is dedicated to filters and a more central area with all the graphical/table elements that integrate the dashboard. In the filters area, the user can select from a wide range of options, such as temporal options, production line, or equipment, among others. Regarding all the examples presented in this paper, it is worth mentioning that all data was anonymized for confidentiality reasons.

The first dashboard example (Fig. 7) is a general dashboard, with a holistic view of the screwing process. The purpose of this dashboard is to allow the user to consult potentially important data of the business process in a fast and effective way. Starting with the first element (Fig. 7, part 1), it is possible to analyse data regarding the quantities by equipment. These quantities are related to the successful or unsuccessful production of each equipment (Screw GoF 1 and Screw GoF 0, respectively). The available data is presented in a descending order considering the produced quantity. It is possible to apply top and side filters to this same visualization, allowing, for example, a visualization of the equipment with an unsuccessful production, with the Screw GoF at 0 (Fig. 7, part 6), or filter this data by a specific equipment or production line (Fig. 7, part 4). It is important to see in the dashboards the temporal attributes, year, month, week, and day filters (Fig. 7, part 5), allowing the user to filter the data by a specific date, thus

increasing the level of detail and specificity of these visualizations. It is important to see in the dashboards the temporal attributes, year, month, week, and day filters (Fig. 7, part 5), allowing the user to filter the data by a specific date, thus increasing the level of detail and specificity of these visualizations.

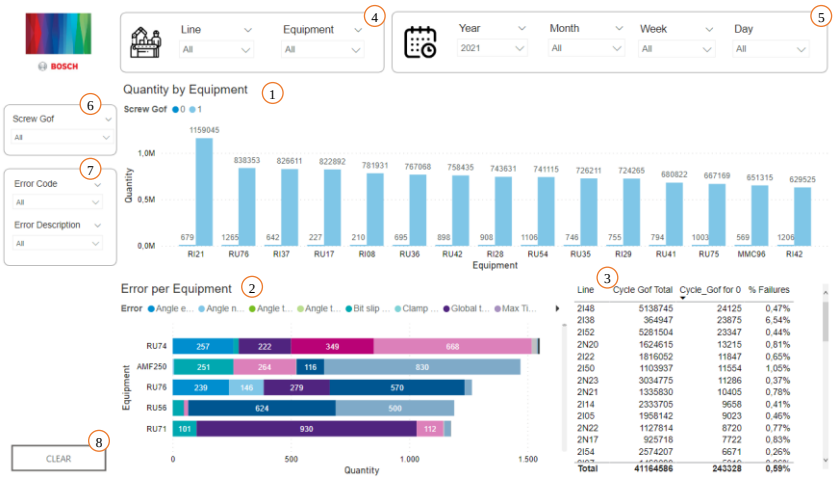

Fig. 7. Macro Screw Tightening dashboard.

Fig. 7, part 2, provides information about the most common errors that cause problems in a production equipment. It is possible to apply again the filters to a specific equipment, to a specific production line, to detail a specific date, search by error code and error description (Fig. 7, part 7), resulting in information regarding the equipment that most tend to suffer that specific error during the production process. In the last element (Fig. 7, part 3), a table calculates the percentage of failures relative to the production cycles, Cycle GoF, for each production line. This table shows the count of the total Cycle GoF values per production line, the count of the lines with Cycle GoF at 0 and calculates the failure percentage. Again, the user can filter a specific production line to return the failure percentage for that same line in the chart. It is also possible to quickly clear all the filters selected by pressing the clear button (Fig. 7, part 8) which resets all the previous settings.

The dashboard with more detailed data (Fig. 8) takes as input the part number and the serial number of a product (Fig. 8, part 1). For that part and serial number, the user has an overview of the GoF test results in a bar graphic (Fig. 8, part 5). Also, the user can see the stations where the product pass considering the screws GoF test results in each station (Fig. 8, part 6). If the user selects a station, the bar graphic in Fig. 8, part 7, will highlight the screw cycles of that station and, for each Cycle ID, the number of GoF tests OK vs NOK (Not OK) is presented. Fig. 8, part 8, shows in detail what are the results of the GoF test for each screw id based on a previously selected cycle id. Also, the prediction of the GoF test result is presented to the user, since in this phase decision-makers want to see the GoF test results and their prediction to evaluate if they can stop doing GoF tests or if they decide to do it by sampling. In this dashboard, a set

of temporal filters can be used, Fig. 8 parts 2 and 3, and it is also possible to filter by GoF test result (part 4). Additionally, the dashboard has two cards to show the values about the percentage of failures (Fig. 8, part 9) and the total of screws (Fig. 8, part 10).

Fig. 8. Detailed Screw Tightening dashboard

Fig. 9 presents a different serial number of the same part number presented in the dashboard of Fig. 8. For this product, in the selected cycles, 7 screws were tight (OK) and 1 is not ok (NOK), ending the process with 14,3 % of failures. Also, it is possible to see that the tightening fails in the first screw and then the product changes to a different station, starting a new screw cycle that also starts the screwing process. Moreover, it is important to highlight that the prediction was capable to detect the failure in the first screw.

Fig. 9. Detailed Screw Tightening dashboard with GoF 0

5 Conclusions

This paper presented the design and implementation of an advanced data analytics environment, taking advantage of Big Data and Machine Learning techniques and technologies. The proposed architecture handles the integration of data from multiple *Data Sources* and for different business processes, providing a way to use that data to train Machine Learning models and store the predicted data. The proposed data modelling method guides practitioners through a set of steps so they can create and evolve the Big Data Warehouse physical implementation based on a logical data model.

In terms of design, the *Big Data Cluster* is divided into the Data Lake and the Big Data Warehouse areas. The Data Lake area stores the Raw Data, the Data Pipelines, and the Machine Learning models. The data in the Big Data Warehouse, the core element for analytical data storage, includes historical data and predictive data obtained using Machine Learning models. This Big Data Warehouse uses Hive tables to store objects that support all the analytical capabilities needed in the *Visualization Tools*.

The Screw Tightening demonstration case was presented, providing historical and predictive data made available throughout a set of dashboards fed by the Big Data Warehouse. In this demonstration case, the data is processed and stored on a daily basis. As future work, data needs to be processed in real-time to avoid rejection before further production steps. This presents several challenges such as applying prediction models to real-time events and the huge volume of data handled by this industrial process.

Acknowledgements. This work has been supported by FCT – *Fundação para a Ciência e Tecnologia* within the Project Scope: UIDB/00319/2020, the Doctoral scholarships PD/BDE/135100/2017 and PD/BDE/135105/2017, and European Structural and Investment Funds in the FEDER component, through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) [Project nº 039479; Funding Reference: POCI-01-0247-FEDER-039479]. The authors also wish to thank the automotive electronics company staff involved with this project for providing the data and valuable domain feedback. This paper uses icons made by Freepik, from www.flaticon.com.

References

1. Wang, L., Alexander, C.A.: Machine learning in big data. International Journal of Mathematical, Engineering and Management Sciences. (2016).
2. Alswedani, S., Saleh, M.: Big data analytics: Importance, challenges, categories, techniques, and tools. Journal of Advanced Trends in Computer Science and Engineering. (2020).
3. Alsghaier, H.: The Importance of Big Data Analytics in Business: A Case Study. American Journal of Software Engineering and Applications. 6, 111 (2017).
4. Rialti, R., Marzi, G., Caputo, A., Mayah, K.A.: Achieving strategic flexibility in the era of big data: The importance of knowledge management and ambidexterity. Management Decision. (2020).

5. Gao, R.X., Wang, L., Helu, M., Teti, R.: Big data analytics for smart factories of the future. CIRP Annals. 69, 668–692 (2020).
6. Papageorgiou, L., Eleni, P., Raftopoulou, S., Mantaiou, M., Megalooikonomou, V., Vlachakis, D.: Genomic big data hitting the storage bottleneck. EMBnet J. 24, e910 (2018).
7. Chavalier, M., El Malki, M., Kopliku, A., Teste, O., Tournier, R.: Document-oriented data warehouses: Models and extended cuboids, extended cuboids in oriented document. Proceedings - Conference on Research Challenges in Information Science. 2016-Augus, (2016).
8. Cuzzocrea, A., Song, I.Y., Davis, K.C.: Analytics over large-scale multidimensional data: The big data revolution! Conference on Information and Knowledge Management (2011).
9. Santos, M.Y., Costa, C.: Big Data: Concepts, Warehousing and Analytics. River (2020).
10. Vaisman, A., Zimányi, E.: Data Warehouses: Next Challenges. In: Business Intelligence: First European Summer School, eBISS 2011, Tutorial Lectures.
11. Costa, C., Santos, M.Y.: Evaluating Several Design Patterns and Trends in Big Data Warehousing Systems. In: Advanced Information Systems Engineering (2018).
12. Elshawi, R., Sakr, S., Talia, D., Trunfio, P.: Big Data Systems Meet Machine Learning Challenges: Towards Big Data Science as a Service. Big Data Research. 14, 1–11 (2018).
13. Syafrudin, M., Alfian, G., Fitriyani, N.L., Rhee, J.: Performance Analysis of IoT-Based Sensor, Big Data Processing, and Machine Learning Model for Real-Time Monitoring System in Automotive Manufacturing. Sensors. 18, 2946 (2018).
14. Lee, J., Ardakani, H.D., Yang, S., Bagheri, B.: Industrial Big Data Analytics and Cyber-physical Systems for Future Maintenance & Service Innovation. Procedia CIRP. (2015).
15. Baldominos, A., Albacete, E., Saez, Y., Isasi, P.: A scalable machine learning online service for big data real-time analysis. In: 2014 IEEE Computational Intelligence in Big Data.
16. Krishnamoorthy, R., Udhayakumar, K.: Futuristic Bigdata Framework with Optimization Techniques for Wind Energy Resource Assessment and Management in Smart Grid. In: 2021 7th International Conference on Electrical Energy Systems (ICEES). pp. 507–514 (2021).
17. Montoya-Torres, J.R., Moreno, S., Guerrero, W.J., Mejía, G.: Big Data Analytics and Intelligent Transportation Systems. IFAC-PapersOnLine. 54, 216–220 (2021).
18. Cai, L., Zhu, Y.: The challenges of data quality and data quality assessment in the big data era. Data Science Journal. 14, (2015).
19. Dehghani, Z.: How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh, (2019).
20. Project Jupyter: Project Jupyter | Home, https://jupyter.org/, last accessed 2021/07/19.
21. Spark.apache.org: Spark SQL and DataFrames - Spark 1.5.2 Documentation, https://spark.apache.org/docs/latest/sql-programming-guide.html, last accessed 2021/07/19.
22. PySpark Documentation — PySpark 3.1.2 documentation, https://spark.apache.org/docs/latest/api/python/, last accessed 2021/07/19.
23. Ribeiro, D., Matos, L.M., Cortez, P., Moreira, G., Pilastri, A.: A Comparison of Anomaly Detection Methods for Industrial Screw Tightening. Computational Science and Its Applications – ICCSA 2021.
24. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: Proceedings - IEEE International Conference on Data Mining, ICDM. pp. 413–422 (2008).
25. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science. 313, 504–507 (2006).
26. Alla, S., Adari, S.K.: Traditional Methods of Anomaly Detection. Apress, CA (2019).