



INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

D03 State of the Art

InValue

Industrial Enterprise Asset Value Enablers

Work Package: WP1

Created by: GECAD

Date: 24-11-2016

Version: 1.0.0

Apart from the deliverables which are defined as public information in the Project Cooperation Agreement (PCA), unless otherwise specified by the consortium, this document will be treated as strictly confidential.

Document History

Version	Author(s)	Date	Remarks
1.0.0	GECAD-ISEP	24-11-2016	Initial Release
	ACD		

Table of Contents

1. Introduction	8
2. Data Acquisition	9
2.1. Contemporary, external State-Of-The-Art	9
2.2. State-Of-The-Practice at the Belgian Consortium	10
2.3. State-Of-The-Practice at the Portuguese Consortium	12
2.4. State-Of-The-Practice at the Spanish Consortium	12
2.5. State-Of-The-Practice at the Turkish Consortium	13
3. Data Processing	16
3.1. State of the Art	16
3.1.1. Big Data Technologies	16
3.1.1.1. Apache Cassandra	17
3.1.1.2. Apache Hadoop	17
3.1.1.3. Apache Storm	17
3.1.1.4. Apache Spark	18
3.1.2. Data Mining Tools	18
3.1.2.1. Python	19
3.1.2.2. R Programming Language	19
3.1.2.3. Graphic Tools	20
3.1.3. Data Pre-Processing	21
3.1.4. Feature Engineering	21
3.1.4.1. Feature Selection and Extraction	22
3.1.4.2. Feature Construction	24
3.1.5. Analytical Models	25
3.1.5.1. Supervised Learning	25
3.1.5.2. Unsupervised Learning	30
3.1. State-Of-The-Practice at the Belgian Consortium	33
3.2. State-Of-The-Practice at the Portuguese Consortium	34
3.1. State-Of-The-Practice at the Spanish Consortium	36
3.2. State-Of-The-Practice at the Turkish Consortium	37
4. Information Delivery	39
4.1. State-Of-The-Practice at the Portuguese Consortium	41
4.2. State-Of-The-Practice at the Spanish Consortium	44



INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

Invalue
State of The Art

4.3. State-Of-The-Practice at the Turkish Consortium	45
5. Conclusion	47
References	48

1. Introduction

The Industrial Internet promises to change our world. The current global industrial system, made possible by the Industrial Revolution, is adapting to the ever-evolving computing and communication systems that are part of the Internet Revolution. New frontiers have been opened, with the ability to accelerate productivity, reduce inefficiency and waste, and enhancement of the human work experience. Companies have been applying internet-based technologies to industrial applications from the moment they became available over the last decade. However, we are still standing far below of the real possibilities: the full potential of internet-based digital technology has yet to be fully realised across the global industry system.

In fact, one of the main problems of the different industrial sectors is related to the productive availability of their processes: a process that is not working or producing at its maximum capacity represents a loss for the company. This is an increasingly important issue, considering the current concerns with optimization of the production processes. Finding answers to these questions would enable a company to reduce productions costs, increase its eco-efficiency and, simultaneously, provide competitive products to the already competitive the global market.

Industrial Internet can be defined as the exchange of information between sensors, industrial machines and end users through the use of Big Data analysis techniques, data visualization and physical and human networks. Through these, new IoT services relying on Enterprise Asset Management (EAM), Conditions-Based Maintenance (CBM) and fleet capacity optimization will emerge. In a manufacturing scenario, this means that all information generated by the machining processes could be fed into the back-office systems of the companies, such that it can be used to create new business opportunities and inform decision-making. Big Data technologies fulfil the need to store and process these substantial amounts of heterogenous data in a useful period of time.

In this document, three main topics are considered: (1) Data Acquisition, approached in the following section; (2) Data Processing, in the third section; and (3) Information Delivery, presented in the fourth section. These sections are organized in the following fashion: for each topic, a state of the art is presented, following by a short state of the practice provided by each country consortium. Finally, some conclusions are presented in the fifth section.

2. Data Acquisition

Already for a number of years, more and more appliances are designed in such a way that they can have a (permanent) connection to the internet. This allows devices from various origins to interconnect through the internet and exchange data with each other and other computation nodes leading to the IoT (i.e. Internet-of-Things) paradigm, which based on a distributed network of sensors, actuators and devices allows to define new systems and services hitherto assumed impossible. When those IoT systems are composed of computation, communication and control elements that are tightly intertwined with the physical world in different domains such as the mechanical, optical or electrical domain the term Cyber-Physical-Systems (CPS) is also used. It is exactly this kind of systems that are the subject of the INVALUE project.

In the context of the INVALUE project, data acquisition refers to: a) the collection of information from sensors obtained from appliances or manufacturing equipment in the field and b) the transfer of this information to a centralised or cloud based repository in such a way that it is amenable for Data Processing as elaborated in section 3. In the exposé below, an overview of the external State-Of-The-Art (SOTA) pertaining Data Acquisition is given first. In subsequent section the internal State-Of-The-Practice of the different industrial partners in the INVALUE project with respect to Data Acquisition are given.

2.1. Contemporary, external State-Of-The-Art

The emergent Industrial Internet is based upon the Internet of Things (IoT). There are already some IoT / M2M Data Management platforms available on the market (as well as collaborative research projects dealing with IoT reference architectures – <http://www.iiot-a.eu/public>).

Many of these platforms only focus on the management of IoT and M2M configurations and applications, with often a minimal support for in depth analysis (i.e. only basic dashboard). In the following we give a brief overview of such systems:

- **Axeda:** The device management platform from Axeda (<https://www.ptc.com/en/axeda>) provides a cloud- based platform to manage devices and their configuration, retrieve data and build basic applications (mostly dashboard-like).
- **NanoService:** The ARM (<https://www.arm.com/>) Sensinode NanoService (https://community.arm.com/cfs-file/_key/telligent-evolution-components-attachments/01-1996-00-00-00-00-53-29/WhitePaper_5F00_NanoServiceTM-Solution.pdf) platform offers comprehensive M2M features, such as events, semantic naming of smart objects, data security and distributed discovery mechanisms. It also allows customers to use cloud infrastructures as a backbone, integrate other software components via web service APIs and define Apps as a way of user interface. The NanoService solution on its own does not contain any data analytics functionality.
- **LORD MicroStrain** from SensorCloud (<http://www.microstrain.com/>) is another quite “low-level” platform. It again allows for quite comprehensive sensor data handling (alerting, data storage etc.) and adds mathematical analysis and technical visualization capabilities (building on Octave and Python components). This interface can also be used to support more complex analytics (a connection towards R is planned).
- **Xiveley** (<https://www.xively.com/>) provides a cloud-based IoT platform that offers a real-time message bus and a series of client- and server-side libraries allowing for rather complex

applications. Following the Platform-as-a-Service paradigm it offers not only the platform itself, but also a development environment for their customers.

- **ThingSpeak** (<https://thingspeak.com/>) is another promising IoT base platform that to collect and store sensor data in the cloud. It facilitates the development of IoT solutions by providing elaborate data management, analysis and visualisation connectors to external tools like for example MATLAB. Next to extensive importing, data processing and privacy management features, it offers an open source framework for fully localized data retrieval and storage. Out of the Box support for Arduino, Raspberry Pi, and BeagleBone Black based end-node hardware is available.
- **Google Cloud IoT** (<https://cloud.google.com/solutions/iot/>) is a solution from Google that enables end-to-end secure communication from an appliance in the field towards their Google Cloud services allowing for easy bidirectional information exchange between the Cloud and the end nodes. The connection to the Google Cloud seamlessly allows this information to be made available to the different data analytics products from Google.
- The **Azure IoT** (<https://www.microsoft.com/en-us/internet-of-things/azure-iot-suite>) suite from Microsoft is a competing solution to Google Cloud IoT with roughly similar functionality. It spans the full set of required functionalities from secure communication, deployment facilities, cloud connectivity and dash-boarding/data-analytics backend functionality. They provide an SDK for the most common embedded operating systems (Linux, Windows ...). It is built around the OPC Unified Architecture (<https://opcfoundation.org/about/opc-technologies/opc-ua/>).
- Oracle's **Arkessa**¹ platform can be seen as quite widely used in the area of machine-to-machine (M2M) communication, cloud-based IoT integration (also with public data-feeds) and mobile access to those data streams. Given Oracle's vast offer in platform hosting as well as integration customers can combine the purchasing of Arkessa with pre-identified system integrators to add data analysis features.
- Exosite's **ONE PLATFORM**² aims at a cloud-based integration of smart objects and respective end users via the offered web portal framework. Although this platform already offers many mechanisms InValue also deems necessary, it completely lacks any functionality for data analytics and thus data management on top of the pure integration of smart objects.
- Another, quite "low-level" platform is offered by MicroStrain in form of their **SensorCloud**³ system. It again allows for quite comprehensive sensor data handling (alerting, data storage etc.), but again lacks any noteworthy data analytics mechanism apart from very technical visualization features (similarly to high-level, interpreter like MATLAB⁴ API commands).
- Bosch's Software Suite Internet of Things and Services Edition⁵ combines a powerful Business Process Management engine (building on Bosch's Business Rules system) with M2M components.

2.2. State-Of-The-Practice at the Belgian Consortium

Both digital cinema projectors and medical imaging devices are currently monitored by different device management platforms – respectively CineCare and QAWeb – that allow inspecting the devices in the

¹ <http://www.arkessa.com/>

² <http://exosite.com/products/onep>

³ <http://www.sensorcloud.com/>

⁴ <http://www.mathworks.de/>

⁵ <http://www.bosch-si.com/products/iots-edition/new-business-models.html>

field and supporting limited interaction with them. As such, current and historic information about healthcare displays and digital cinema projectors is available in different systems.

Digital cinema projectors – field operation parameters:

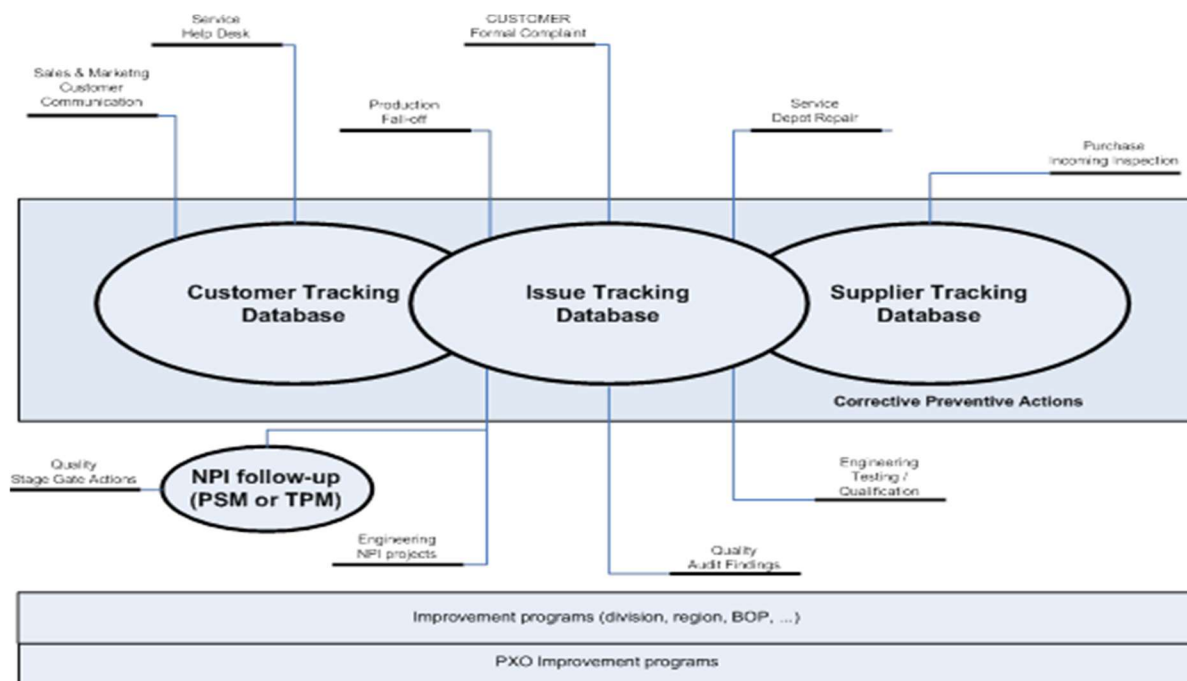
Barco currently has developed as system to acquire, transfer, coalescence and unify the data collected via the following information sources:

CineCare Web: Via this remote asset management system based on Axeda (<https://www.ptc.com/en/axeda>), standard device management information and a limited set of operational parameters are retrieved from the digital cinema projectors through SNMP (yielding tables of key-value pairs) on a continuous basis. A cinema projector typically collects a few hundred parameters (exact number depends on the model). There are a few dozen different Barco Digital cinema projector models in the field. Barco has an install base totalling over 50.000 digital cinema projectors (<https://www.barco.com/en/News/Press-releases/Barco-extends-global-leadership-and-reaches-50000-digital-cinema-projector-milestone-for-worldwide-d.aspx>).

Onboard device Diagnostics: A cinema projector has a complex federated onboard diagnostics system that continuously stores information in a complex hierarchical log file structure. Currently Barco only has access to this data if it explicitly requests the customer to upload the resulting log-file information into the issue tracking database (SalesForce.com)

Network Operations Centers: Regional third party management systems used by the different service providers with which Barco collaborates (i.e. NOCs) do use their own data formats and logging systems.

Other information sources, like the Customer and the Supplier tracking databases, use also their own format. These different sources are shown below:



Healthcare displays – field operation parameters:

Barco has developed an inhouse IoT platform to constantly collect field data from a worldwide installation base of over 50000 healthcare displays. The number of parameters that are permanently monitored via QaWeb this way is rather small (less than 100 parameters). There exists a separate tool called “MFD Control” that can be used to request more detailed information from a larger set of monitored parameters in a healthcare display but that is only used in ad hoc circumstances by engineering, customer service and repair facilities and is in fact never installed at the customer site. As such, the degree of variation in both the format and number of information sources healthcare displays is lower than is the case for digital cinema projectors.

The current technology on which the QaWeb facility is built consists of a pc based java agent on the display side, IP relay proxy as network technology and a combination of the Java Spring framework and Microsoft SQL Server interconnected via the JDBC-API at the hosting site.

2.3. State-Of-The-Practice at the Portuguese Consortium

The Portuguese consortium has implemented a system to digitalize existent industrial machines with different ages and technologies (Machining Center and CNC Turning Center).

Evoleo has developed a system to acquire, transfer and unify the data collected from different sources:

- Data available on machine BUS
- External sensors (machines where this BUS is not available)

With this hybrid approach, all machines, from new to legacy, can be monitored in order to become part of the InValue Platform. The data to be monitored is, among other:

- Temperature on Tool and Temperature on Product;
- Noise and Vibrations;
- Load and Pressures;
- Job info, like: n° of cycles, Operator ID, Tool in use, hours tool used, etc..
- Emulsion Levels and Quality;
- Lubricant Levels;
- Energy Consumption;
- Movements done;

For the acquired data publication, this gateway implements a Publish/Subscribe secure communication pattern using the Apache Kafka open-source platform for the management of messages. This tool focuses on high throughput, low latency, horizontal scalability and fault tolerance in communications, features required for this project. In this way, the sending layer present in the gateway implements the message publishing interface, while the receiving layer present in the server implements the message consumption interface. All data is compressed with the gzip algorithm, in order to reduce the size of the messages and, consequently, to increase the speed of the communication.

Since Apache Kafka works on a Publish / Subscribe system, new data consumers could be created. This functionality is used to create a real-time stream to provide data to the eco-efficiency and maintenance algorithms.

2.4. State-Of-The-Practice at the Spanish Consortium

Cyber-Physical Systems

The Spanish consortium has implemented an automatized quality control system, composed by an optical scanner and a CMM, which has been integrated with the ERP and MES systems that control the

manufacturing process through the M3 software and its analytical tools. Hence, the software is capable of automatically controlling the CMM + scanner depending on the part to be analysed, which has to be indicated by the operator in advance and determine the trajectories of the scanner, speed, etc. The aim is to achieve an automatic process of data acquisition in form of point clouds, capable of analysing 100% of the manufactured parts. This implementation, in addition to the integration between manufacturing systems (ERP and MES), ensures the data exchange and interoperability, giving an added value to the quality control and increasing remarkably the traceability of the parts.

Data Acquisition

Through the Spanish use case an integration between a metrological system (M3 + MMC), an Enterprise Resource Planner (Izaro i68) and a Manufacturing Execution System (Olanet) has been developed.

The acquired data, by means of an optical scan working conjointly with a CMM, is a point-cloud, which represents a high volume of data (Big-Data). This point-cloud needs to be stored, analysed and transferred to the other systems. By means of big-data analysis tools, this point-clouds are processed by the M3 Software, resulting in reports or statistical analyses, which are stored at the M3 Servers and available for ERP or MES to use. This big-data files are stored in a way that can be easily exchanged by the different systems and used for any purpose.

The novelty of this implementation is that the metrological data, which is usually stored independently from the other manufacturing information, integrates the manufacturing order data, so it is possible to correlate manufacturing variables with the obtained dimensions. The interchangeability of the data is essential in order to allow M3 to acquire the data from the process (time, operator, machines, etc.) in order to provide more information in the reports and analyses, and also allows ERP and MES systems to collect the data created by M3 in order to have the information always available through a unique platform (ERP). This allows EPC to have the information (i.e. point-clouds, reports, analyses, etc.) available for any user that should use them for any purpose, by securing the access and restricting it to chosen users.

Self-optimizing production systems

The InValue project implementation in the Spanish consortium integrates metrological quality systems in the manufacturing line. Thus, the quality control becomes a part of the manufacturing process and provides valuable information to EPC. This information (i.e. reports, statistical analyses, etc.) can be used to implement preventive maintenance, avoid tool wear and, consequently, avoid non-conformant parts, etc.

Hence, this implementation integrates process and quality control, enhancing the quality of the production system by means of better and eased decision making, a better-quality process that leads to less defective parts and, in general, lowers the amount of waste material and enables non-conformance related costs reduction.

2.5. State-Of-The-Practice at the Turkish Consortium

Recently, the term big data was used to label data with different attributes. Moreover, different data processing architectures for big data have been proposed to address the different characteristics of big data. As a general, data acquisition has been understood as the process of gathering, filtering, and cleaning data before the data has been reached to target such as data warehouse or any other storage solution. The Figure 1 shows us, position of the data acquisition in the overall big data value chain.

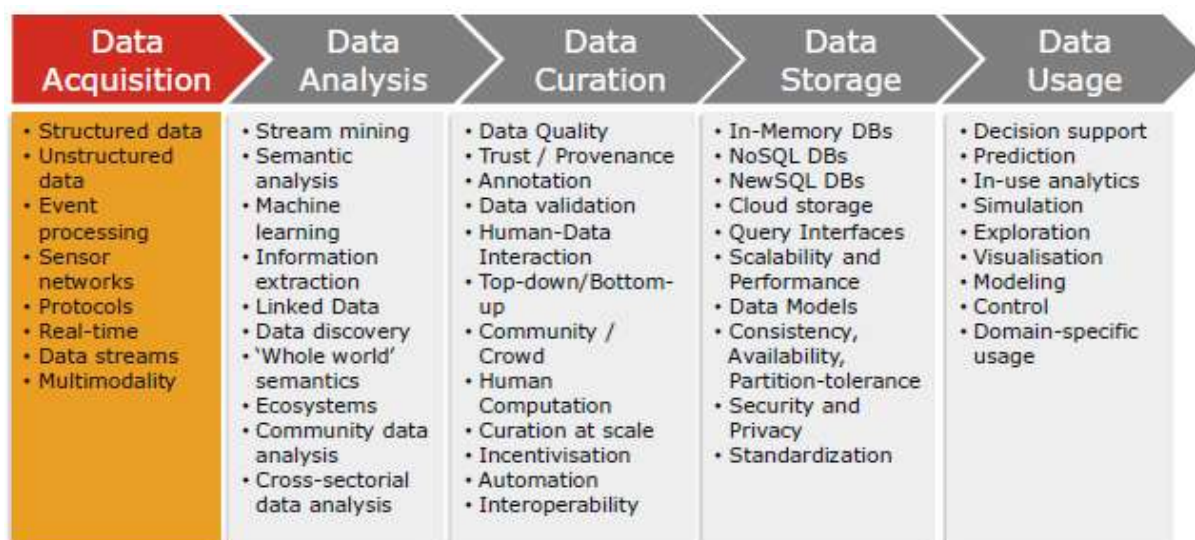


Figure 1 - Data acquisition in the big data value chain (Depicted from Lyko et al. 2016)

There are different architectures to big data processing. The core of data acquisition is to gathering data from distributed information sources to storing them in scalable, big data-capable data storage.

To achieve this goal, three main components are required:

1. Protocols that allow the gathering of information for distributed data sources of any type (unstructured, semi-structured, structured).
2. Frameworks with which the data is collected from the distributed sources by using different protocols.
3. Technologies that allow the persistent storage of the data retrieved by the frameworks.

The bulk of big data acquisition is carried out within the message queuing paradigm, sometimes also called the streaming paradigm, publish/subscribe paradigm (Carzaniga et al. 2000), or event processing paradigm (Cugola and Margara 2012; Luckham 2002). Here, the basic assumption is that manifold volatile data sources generate information that needs to be captured, stored, and analysed by a big data processing platform. The new information generated by the data source is forwarded to the data storage by means of a data acquisition framework that implements a predefined protocol. This section describes the two core technologies for acquiring big data (Protocols and Softwares).

Protocols - Several of the organizations that are interested on big data processing have invented enterprise-specific protocols. Most of these protocols have not been publicly released. You can also find commonly used open protocols for data acquisition in the field of big data such as AMQP (Advanced Message Queuing Protocol), JMS (Java Message Service).

Software Tools - With respect to software tools for data acquisition, many of them are well known and many use cases are available all over the web so it is feasible to have a first approach to them. Despite this, the correct use of each tool requires a deep knowledge on the internal working and the implementation of the software. To determine which software tools should be used ? You have to know your data structure, data model and what you want to get from analytics of your data ?. You should locate your project in the Big Data environment that shown in Figure 2.

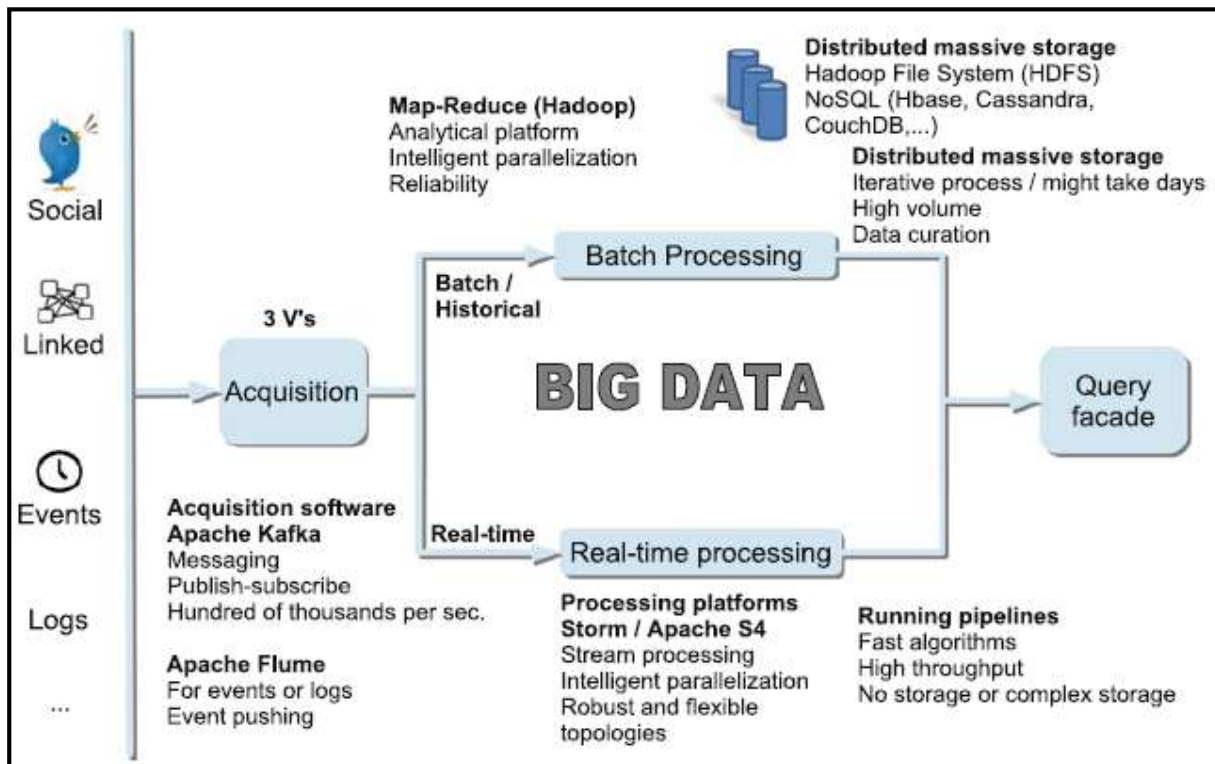


Figure 2 - Big Data Workflow (Depicted from Lyko et al. 2016)

The most commonly used software tools in the data acquisition are Storm, Spark Streaming, S4, Kafka, Flume, Oracle GoldenGate and Java MapReduce programming model.

Machine data like heat, maintenance, production, setup, failover etc., are collected by multi-protocol communication software in Turkish use case. This is the initial step of the introduction to Invalue Platform. Collected data allows production line monitoring and analytics depending on platform defined KPIs based on efficiency, utilization.

Beside production devices in manufacturing there are also human resources which are tracked by ERP systems as well. This resource information also combined with ERP integration and related data should be collected to correlate with production environment.

3. Data Processing

3.1. State of the Art

In this section, the concepts of Big Data and Data Mining are introduced and, for each of them, a survey of popular technologies and approaches are presented.

Most manufacturing organizations operate in data-rich environments, but decision-making is still largely dependent on human experience. However, driven by rapid technological changes, modern manufacturing is increasingly moving towards flexible, intelligent production systems. Productivity, performance and product quality are determined by the conditions of machines, manufacturing processes and manufacturing decision making [1, 2].

Manufacturing operations generate vast amounts of data, but in many cases organizations aren't collecting, storing and using this data to improve process performance. The advent of Big Data technology, coupled with efficient data storage mechanisms and parallel processing frameworks, has found new use for the petabytes of data generated by manufacturing operations. Machine Learning and Data Mining techniques can be applied to the shop floor to gain insights and accurately predict outcomes to support decision-making and help organizations improve their operations and competitiveness [1, 2].

3.1.1. Big Data Technologies

While there isn't currently a single definition for the term Big Data, it is generally accepted that it usually includes huge volumes of data, which is available at different levels of complexity and generated at different levels of speed, and often inconsistent, incomplete and even noisy. The volume of generated data is such that traditional data analysis techniques become inadequate to process and extract value from it. Therefore, new technologies and processes are being developed in order to capture, manage and process these volumes of information in a useful amount of time [3].

Big Data is commonly described by the three dimensions proposed by Douglas Laney: Volume, Velocity and Variety [4]:

- **Volume:** data isn't sampled, it is merely stored. Every action in an environment is possible of generating data, therefore producing enormous volumes of data which are left mostly unprocessed.
- **Velocity:** data is often generated in real-time and it may be very important for it to be analyzed in a matter of milliseconds from its arrival so that actions can be taken, and even compare it with previously stored data.
- **Variety:** data can be gathered from several different sources, meaning that it will be unstructured, possibly incomplete and very heterogeneous in nature.

These dimensions require processing and storage ability beyond that of current database management systems. Relational models are slow and unable to deal with the variety of data. NoSQL (Not only Structured Query Language) solutions aim to go beyond the abilities provided by separating data storage from data management. NoSQL databases can be schema-free, thus enabling applications to modify the structure of data without the need to rewrite tables in the storage phase. This provides them a much greater flexibility to deal with heterogeneous data. The data management layer then enforces the validity and integrity of the information [5].

The most popular NoSQL database is Apache Cassandra. Another popular tool, though providing functionalities far beyond simple NoSQL, is Apache Hadoop.

3.1.1.1. Apache Cassandra

Apache Cassandra [6] is a NoSQL database that manages large volumes of structured data on a series of commodity servers. Cassandra is highly distributed, which allows for a high tolerance to node failure, as well as processing large datasets while remaining available to thousands of concurrent users, making it ideal for processing real-time transactional data. However, Cassandra does require its data to be at least partially structured.

3.1.1.2. Apache Hadoop

Apache Hadoop [7] is a Big Data analytics framework focused on batch-oriented analytics of historical data. It allows analytics of high volumes of data to be performed on commodity hardware at a very high scale. It uses open source software, provides its own distributed file system (the HDFS: Hadoop Distributed File System) and a programming framework (MapReduce) for storing, managing and analysing large unstructured datasets. Hadoop can schedule computation tasks on the same nodes the data is stored in, thus being best suited for running near real-time and batch-oriented processing [8, 9]. This makes it more reliable when it comes to processing historical data on a reliable and fault-tolerant way.

MapReduce is a programming model for processing large datasets. A Map function is specified to go from a key/value pair into an intermediate set of key/value pairs. These are then the input of a Reduce function, which will merge all the intermediate values with the same intermediate key. This means that, in a network, each node needs to know only a portion of the data, thus making the workload highly distributable. Hadoop focuses deeply in the partitioning and scattering of data through many hosts, while running the processes in parallel in order to keep them as close as possible to the data they need [10, 11].

The Hadoop Distributed File System (HDFS) introduces two types of nodes: the NameNodes and the DataNodes [9].

The HDFS is a hierarchy of files and directories, which are represented on the NameNode through iNodes, recording attributes such as permission, modification and access times. The files are split into large blocks and each part is then scattered and replicated through multiple DataNodes. When a file is requested, the HDFS client will contact the NameNode in order to find the closest DataNode with the file so it can provide its contents. When writing a file, the NameNode is queried about which DataNodes nearby have available writing space, and then the file is written on those DataNodes in a pipeline fashion [9].

While NoSQL databases solve the question of storing huge amounts of data and batch processing, there remains the issue of processing data being provided in real-time.

3.1.1.3. Apache Storm

Apache Storm [12] is written primarily in the Clojure programming language and is focused on processing streams of data, making it appropriate for complex event processing tasks and incremental computation. It is massively scalable, fault-tolerant, and gives a strong guarantee that every tuple is

processed. It introduces the concept of Topology, which replaces MapReduce jobs on the Storm environment [12, 13].

Topologies are directed acyclic graphs, that define the flow of data between nodes and will run until killed, much unlike a MapReduce job (which must eventually end) [13]. The Topology acts as a data transformation pipeline. It can be executed as a whole or partially and its execution is distributed among the network nodes.

The most predominant aspects of a Storm Topology are the Spouts and the Bolts, which act as the graph's vertices. The Spouts bring data into the system, therefore representing a streaming data source, and hand off the data to the Bolts – where the bulk of the computation work is done. Bolts can then deliver their output to other Bolts, thus being capable of being both consumers and providers of data. Bolts can execute computational tasks, write data to a database or filesystem, send messages to external systems or make results available to the users [13-15].

A rich variety of Spouts is currently available, specializing in receiving data from many different types of data sources – e.g. the Twitter API, Apache Kafka and JMS brokers. Nevertheless, it is always possible to write new Spouts in case of extremely specialized applications.

Furthermore, there's also a range of Adapters for integration with filesystems – including HDFS – and several programming languages. Spouts and Bolts can be written in almost every programming language, with the communication being made with JSON [16].

3.1.1.4. Apache Spark

Apache Spark is an open source cluster computing framework [15]. It runs on top of an HDFS and complements it with enhanced and additional functionality. Being an independent tool Apache Spark does not necessarily require a Hadoop environment to run and can interface with a variety of distributed storage systems, including not only Hadoop, but also MapR-FS, Cassandra, OpenStack Swift, Amazon S3, among others. As for cluster management, it supports a native format – Spark Cluster – and both Hadoop YARN and Apache Mesos [17].

Spark provides a set of tools for data processing and relies on a data structure known as RDD – resilient distributed dataset – which is a read-only multiset of items distributed over a cluster of machines. The RDD data structure facilitates the implementation of both iterative algorithms – visiting datasets multiple times in a loop – and interactive/exploratory data analysis – repeated database-style querying of the data [18].

As a framework, it provides programmers with an API giving access to several tools to perform actions over data, such as MapReduce, SQL queries, machine learning and graph processing. These tools can be used by themselves or combined through a pipeline, with a much smaller latency than that of Hadoop's regular MapReduce [15]. Furthermore, Spark can work over cold, historical data or streaming data, enabling Hadoop clusters to run several times faster, either on disk or on memory.

3.1.2. Data Mining Tools

Data Mining, as a term, is often used to describe the computational processes associated with the discovery of patterns in large volumes of data [19]. Its goal is to extract information from data sets and provide it in an understandable, useful way, e.g., to discover previously unknown trends, anomalies and correlations in the data to answer specific answers and generate predictions. This can be achieved using graphical tools, which specify which algorithms and mechanisms can be used on the data, or

through programming tools, which have a larger learning curve, but give the user total freedom. When it comes to choosing an approach (graphical tools vs programming languages), the decision relies on what is intended with the data analysis and what's the use for the results. While graphical tools do have the advantage of not requiring code to be written, code is much more flexible and gives the user more control to define exactly what he/she wants to achieve.

When it comes to programming languages adequate for data mining, two are particularly popular: Python and R.

3.1.2.1. Python

Python [20], as a programming tool, is largely adopted in scientific computing. It's known for being easy to learn and very readable, with immense documentation available online [20, 21]. Functionalities are very easily added to it through the installation of packages. The most common packages for data mining are: NumPy, SciPy, Pandas, matplotlib and scikit-learn [21]. A command shell for interactive computing is provided by IPython and its successor, Jupyter.

- NumPy: NumPy [22] is a scientific computing package which supports multi-dimensional arrays and matrices and a number of high-level mathematical functions to operate on said arrays [21, 22].
- SciPy: SciPy [23] is dependent on NumPy and adds a vast number of efficient numerical routines to operate on NumPy arrays [21, 23].
- Pandas: Pandas [24] provides flexible and expressive data structures to work with relational or labeled data and time series. It provides a simple way to deal with heterogeneous, unstructured data [21, 24].
- Matplotlib: a 2D plotting library. It's a data visualization tool that provides an easy way to generate several kinds of charts, plots, histograms, power spectra, among others. [21, 25].
- Scikit-learn: scikit-learn provides a simple, reusable API for several (supervised and unsupervised) state-of-the-art algorithms for modeling data, including but not limited to: Clustering, Cross Validation, Dimensionality Reduction, Feature Extraction & Selection [26].
- IPython: is a browser-based command shell for interactive computing in Python. It features introspection, rich media, shell syntax, completion, parallel computing and history [27].

3.1.2.2. R Programming Language

R [28] is a programming language for statistical computing which is widely used for data analysis. It has been developed from the S programming language and most R programs will run on S unaltered [28]. It provides a large variety of statistical techniques, including modeling, time-series analysis, classification, and clustering, among others. Furthermore, it is easily extended and several user-created packages are available, making it extremely adequate for prototyping. Data manipulation, calculation and graphic display facilities include [28]:

- Effective data storage and handling
- Operators for calculations with arrays and matrices
- Graphical facilities for data analysis

As things stand today, if a statistical method exists, it's very likely that it can be found already implemented through a R package. Common packages for R include ggplot2, shiny and dplyr:

- ggplot2: is a plotting system for R. It provides a simple way to generate complex, multi-layered graphics. It has a consistent underlying grammar of graphics, with the plot specification having a high level of abstraction. With the existence of themes that can be manipulated by the user, it becomes a very simple and flexible choice [29].
- shiny: provides a simple yet powerful framework for web applications based on R. By using Shiny, no knowledge of web programming is required to generate flexible and interactive data visualizations [30].
- dplyr: aims to make data manipulation simple by constraining the options it lends to the user. It provides data manipulation functions in the form of “verbs”, including the most common data manipulation tasks in this form. Furthermore, it works not only with in-memory data-frames, but is also capable of connection to out-of-memory, remote databases, translating the R code into the appropriate SQL statements, thus allowing for working with both types of data with the same toolset [31].

While R is very reliable when it comes to data analysis, it does not offer the robustness required for large-scale, real-world systems. It is much more academic-oriented and adequate to prototyping [32]. On the other hand, Python is much more flexible and almost as good as R for data modeling and analysis, while also being practical and adequate to build strong and robust software solutions [32, 33]. Furthermore, R and Python can be combined in many ways to obtain the best of both in a single software solution [33, 34].

For calling R from Python, one can use sub processes and run R script files, as if through the command line. However, there's the rpy package, currently named rpy2 [35], which provides simple and robust access to R from within Python. As for calling Python from R, the rPython [36] package is available, which allows running Python code, making function calls, and assigning and retrieving variables.

3.1.2.3. Graphic Tools

It is relevant, however, to focus a little on some of the most popular graphic tools used for data mining. While not exactly simple or straight-forward, graphical tools are much easier to learn and to get used to when doing data analysis than a programming language. Graphic tools such as RapidMiner [37], Weka, Orange and KNIME are also very commonly used, with RapidMiner being the favourite [38]:

- RapidMiner: Java-based tool for advanced analytics through template-based frameworks. RapidMiner provides utilities such as preprocessing, visualization, predictive analysis and statistical modeling, among others. It's very popular and considered a very complete tool, even though it's provided as a service instead of a piece of local software [37, 38].
- WEKA: a sophisticated Java-based tool commonly used for data visualization, data analysis and predictive modeling. It's open-source under the GNU Public License, thus allowing it to be customized by the users as they see fit. It supports several data mining tasks, including data preprocessing, clustering, visualization, feature selection, among others [38, 39].
- Orange: open-sourced and based on Python, it is a very simple yet powerful tool with visual programming modes and machine learning components. It is known for being very simple to learn and adequate for Python developers [38, 40].
- KMINE: does all three of the main components of data processing – extraction, transformation and loading. It integrates several components for machine learning and data mining through modular data pipelining. It is open-source, written in Java and based on Eclipse, which allows it to be easily extended through plugins [38, 41].

3.1.3. Data Pre-Processing

Data pre-processing is an important step in the data mining process that involves transforming raw data into an understandable format. The phrase "garbage in, garbage out" is commonly used in data mining and machine learning projects to refer to the poor results that are obtained when low quality data is used. Data in the real world is "dirty" and "messy" and needs to be transformed and improved before it can be used for analysis. Quality decisions must be based on quality data [42].

Real world data is usually incomplete, it might be missing feature values, contain only aggregate data or lack certain features of interest. Raw data can also be noisy, containing errors or outliers (e.g., income: -100), and inconsistent, meaning it contains discrepancies in codes or names [42].

The major tasks in data pre-processing are:

- Data Cleaning
- Data Integration
- Data Transformation
- Data Reduction

These tasks comprise most the work performed in a data mining or machine learning project [42].

Data cleaning consists in filling in missing values, identifying outliers and smoothing out noisy data, as well as correcting inconsistencies. A missing value may be filled in various ways, such as by using the feature's mean value, or by predicting the missing value using a learning algorithm. Ignoring the data row is also possible (usually done when the class label is missing), but doing so will result in a loss of information. Outlier identification can be done with clustering techniques or distance-based methods, among several other techniques. Methods like binning and regression can be employed to smooth out noisy data, and correcting inconsistent data requires domain or expert knowledge [42].

Real world data often comes from heterogeneous sources and needs to be consolidated before it can be used. This process involves combining data from multiple sources (databases, files), performing schema integration, detecting and resolving data value conflicts – e.g. for the same real world entity, feature values from different sources have different scales –, and removing duplicates and redundant data [42].

Data transformation includes normalization, namely scaling feature values to fall within a specified range, data aggregation (summarization) and feature construction [42]. Data transformation and data reduction are related to feature engineering, a complex and important topic that will be discussed in more detail in Section 3.1.4.

3.1.4. Feature Engineering

In machine learning, feature engineering is the step between data preparation and data modelling. It's the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data. It is a crucial step, because the right features can make the job of modelling much easier, and therefore the whole process has a higher chance of success. Feature engineering involves feature construction, the process of deriving new features from the original input by incorporating domain knowledge into the data, as well as dimensionality reduction through feature selection and extraction.

3.1.4.1. Feature Selection and Extraction

Two decades ago most domains dealt with only a few dozen features. However, in recent years there's been a substantial increase in the number of features used, with many domains using hundreds to tens of thousands of features [43].

As the number of features increases, the volume of the state space increases so fast that the available data becomes sparse. To obtain a statistically significant result, the amount of data necessary grows exponentially with the dimensionality (number of features). Considering that the complexity of most learning algorithms depends on the number of input features and on the size of the data sample, reducing the dimensionality of the problem is imperative to reduce memory and computational requirements. In addition, feature selection can also be beneficial by: improving the prediction performance of the predictors (reduces overfitting and improves accuracy); facilitating data visualization and exploration; making the underlying process that generated the data easier to understand (a simpler model is simpler to understand and explain) [43].

A feature selection algorithm works by first presenting a possible subset of features using a search technique and then scoring it with an evaluation measure. The simplest approach consists in finding and evaluating all possible subsets, subsequently choosing the one with the best score (or lowest error rate). The issue with this method is that such an exhaustive search is, for all but the smallest feature sets, so computationally expensive that it becomes infeasible [43].

Feature selection methods typically belong to one of three classes, determined by the evaluation metric of choice: filter methods, wrapper methods and embedded methods.

Another means of performing dimensionality reduction is feature extraction. Like feature selection, feature extraction seeks to reduce the number of attributes in the data, but whereas feature selection does so by including and suppressing features present in the data, feature extraction uses the existing features to create new ones [44].

Feature extraction techniques include Principal Component Analysis (PCA) and Kernel PCA, among others.

Filter Methods

Filters select features during pre-processing of the data by using statistical measures such as the Pearson product-moment correlation coefficient, significance tests, pointwise mutual information, or mutual information (information gain). These measures are faster to compute and less computationally demanding than wrappers or embedded methods. However, the subset of chosen features isn't tuned to a specific type of predictive model resulting in lower predictive performance. On the other hand, the feature set selected by a filter is more general than the one selected by a wrapper and is, therefore, more useful for exposing the relationships between features [43, 45].

Many filter methods work by ranking features individually and suppressing the least interesting ones; they don't select a subset of features per se. Consequently, these methods have the disadvantage of not considering the interactions between features. This results in the possible selection of redundant features and in the suppression of features that might be unimportant individually, but useful when grouped with others. It is also possible to use a filter method as a pre-processing step, followed by a wrapper or an embedded method to reduce space dimensionality and overcome overfitting [43, 45].

Wrapper Methods

With wrapper methods, the selection of a subset of features is done using the predictive model as a black box (no internal knowledge of the algorithm is necessary). The model's predictive performance is

used to assess the value of a subset of features. This works by searching the state space for all possible subsets and evaluating each one based on model accuracy, which is usually done by cross-validation or using a validation set. The chosen subset of features is the one that produces the best accuracy [43, 45].

As mentioned in subsection **Error! Reference source not found.**, searching the entire state space is infeasible unless the number of features is small. Several search strategies can be used such as best-first search, branch-and-bound, random hill-climbing, simulated annealing, or genetic algorithms. While computationally intensive, wrappers optimize prediction performance by selecting the best performing feature set for a particular type of model. Wrappers are considered “brute force” methods that are computationally demanding, but efficient search strategies that are robust to overfitting and less (but still) computationally demanding may be devised. Two such (greedy) strategies, also known as sequential methods, are forward selection and backward elimination [43, 45].

Forward selection begins with an empty set of features and adds new features one by one, according to the one that decreases the error the most, until no improvement can be obtained. Alternatively, backward elimination starts with a full set of features and progressively removes the feature that decrease the error the most until the error rate stops decreasing. In both cases, to test the generalization accuracy, the model must be evaluated using a (validation) data set distinct from the training set, since a greater number of features usually results in lower training error, but not necessarily in lower validation error [43, 45].

Embedded Methods

With wrapper methods, the selection of a subset of features is done using the predictive model as a black box (no internal knowledge of the algorithm is necessary). The model’s predictive performance is used to assess the value of a subset of features. This works by searching the state space for all possible subsets and evaluating each one based on model accuracy, which is usually done by cross-validation or using a validation set. The chosen subset of features is the one that produces the best accuracy [43, 45].

As mentioned in subsection **Error! Reference source not found.**, searching the entire state space is infeasible unless the number of features is small. Several search strategies can be used such as best-first search, branch-and-bound, random hill-climbing, simulated annealing, or genetic algorithms. While computationally intensive, wrappers optimize prediction performance by selecting the best performing feature set for a particular type of model. Wrappers are considered “brute force” methods that are computationally demanding, but efficient search strategies that are robust to overfitting and less (but still) computationally demanding may be devised. Two such (greedy) strategies, also known as sequential methods, are forward selection and backward elimination [43, 45].

Forward selection begins with an empty set of features and adds new features one by one, according to the one that decreases the error the most, until no improvement can be obtained. Alternatively, backward elimination starts with a full set of features and progressively removes the feature that decrease the error the most until the error rate stops decreasing. In both cases, to test the generalization accuracy, the model must be evaluated using a (validation) data set distinct from the training set, since a greater number of features usually results in lower training error, but not necessarily in lower validation error [43, 45].

Principal Component Analysis (PCA)

Principal Component Analysis is a statistical procedure that orthogonally transforms the original n coordinates of a data set into a new set of n coordinates called principal components. Because of the

transformation, the first principal component has the largest possible variance, with each succeeding component having the highest possible variance under the constraint that it is orthogonal to (uncorrelated with) the preceding components. The principal components are orthogonal because they are the eigenvectors of the covariance matrix, which is symmetric [46, 47].

PCA reduces the dimensionality of the data by finding a new, smaller set of features that retains most of the data information (variation in the data). This is possible because the principal components obtained with PCA are sorted by variance, therefore, keeping the first m principal components should also retain most of the data information, while reducing the dimensionality. The number of principal components to retain is decided beforehand, which is a strong assumption since there could be interesting information associated with the principal components of lesser variance. However, the principal components with low variance usually represent uninteresting noise symmetric [46, 47].

Principal Component Analysis is sensitive to the relative scaling of the original variables. Data column ranges need to be normalized before applying PCA. It must also be taken into consideration that the new coordinates (principal components) are not real system-produced variables anymore. Applying PCA to the data loses its interpretability. PCA is not an adequate method of dimensionality reduction if interpretability of the results is important [46, 47].

Principal Component Analysis is a linear projection technique that works well if the data is linearly separable. However, in the case of linearly inseparable data, a nonlinear technique is required to reduce its dimensionality. Kernel PCA is an extension of principal component analysis that uses techniques of kernel methods to achieve this purpose. The original data is projected onto a higher dimensional space where it becomes linearly separable. The nonlinear mapping function used for this purpose is called "kernel function". The kernel function maps the original d -dimensional features into a larger k -dimensional feature space by creating nonlinear combinations of the original features. It does so without explicitly calculating the coordinates of the data in the higher-dimensional space, but rather by calculating the dot products of each point of the transformed data with respect to all the transformed points. This approach is called the "kernel trick" and is usually computationally cheaper than the explicit computation of the coordinates. Because it never works directly in the feature space, kernel PCA is restricted in that it computes not the principal components themselves, but the projections of the data onto those components [48, 49].

3.1.4.2. Feature Construction

Machine learning starts with the design of appropriate data representations. Better performance is often achieved using features derived from the original input. Building a feature representation is an opportunity to incorporate domain knowledge into the data and can be very application specific. This implies manually creating new features, a process that requires spending long periods of time analysing the data and thinking about the underlying problem. Two distinct goals may be pursued for feature construction: achieving best reconstruction of the data or being most efficient for making predictions [43].

For tabular data, feature construction might involve creating new features by aggregating or combining features, and decomposing or splitting features. With textual data, it often means devising document or context specific indicators relevant to the problem. When it comes to image data, it might mean spending vast amounts of time prescribing automatic filters to pick out relevant structures [44].

This is the part of feature engineering that is often talked the most about as an art form, the part that is attributed the most importance and signalled as the differentiator in competitive machine learning. It is manual, slow, requires lots of human brain power, and makes a big difference [44].

3.1.5. Analytical Models

Analytical models are statistical algorithms that discover patterns and relationships from data and express them as mathematical equations. They include a variety of statistical techniques from machine learning and data mining that analyse current and historical data to discover and interpret meaningful patterns, and make predictions about future or otherwise unknown events.

Machine Learning and Data Mining are similar in many ways, with both employing the same methods and overlapping significantly. The difference lies in their purpose. While Data Mining seeks to discover previously unknown properties in the data, Machine Learning focuses on prediction based on known properties learned from the training data. Moreover, Machine Learning uses Data Mining methods to perform unsupervised learning, or as a pre-processing step to improve learner accuracy [50].

Machine Learning tasks are typically classified into three different categories [50]:

- **Supervised Learning:** Algorithms are trained using example inputs and their desired outputs (labels) with the intent of learning a general rule that maps inputs to outputs. The resulting models can then be applied to new data to make predictions.
- **Unsupervised Learning:** Unlike the case of supervised learning, there are no labels available. The learning algorithm must find patterns in the data by itself. This type of learning can also be used for other purposes, such as feature engineering.
- **Reinforcement Learning:** A software agent interacts with a dynamic environment in which it must autonomously fulfil a certain goal by automatically determining the ideal behaviour. Simple reward feedback is required for the agent to learn its behaviour; this is known as the reinforcement signal.

Sections 3.1.5.1 and 3.1.5.2 present some Machine Learning and Data Mining techniques, including some that have been successfully employed in manufacturing environments [1, 51].

3.1.5.1. Supervised Learning

Random Forests

Random forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Each decision tree is built following a strict set of criteria meant to optimize the predictive capability on unseen data. Figure 3 shows three such trees that might be learned from training data, in making a yes/no decision about whether someone would eat at a new restaurant. Depending on the features of the new restaurant, such as whether they serve salads or whether it has more than 3 star reviews, a decision is made by majority voting the results in the leaf nodes of each of the trees [49, 52].

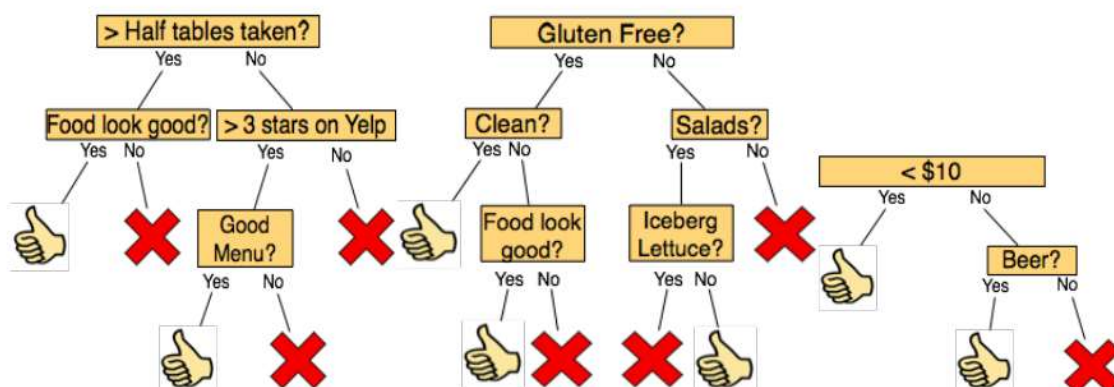


Figure 3 - Decision Trees

Each tree is generally different because they are learned on a random subset of the training data and the split-point decision at each node is also decided on a random subset of the data. This randomization is called bagging and serves to decorrelate the trees, thus lowering the variance on the predicted outcomes while controlling for bias. Individual decision trees are not very accurate and trees that are grown very deep tend to overfit the training data, having low bias, but very high variance. By averaging multiple deep decision trees, trained on different parts of the same training data, Random Forests greatly improve the performance of the final model [49, 52].

The type of model built is determined by the types of outcomes predicted by each tree. For example, categorical leaf node predictions are used for classification trees, numerical or functions for regression trees, and Gaussian functions for density trees. One of the attractive properties of Random Forests is that predictions are inherently probabilistic, and such probabilities are a natural outcome of the model itself. In a yes-no decision if 835 trees vote yes and 165 trees vote no on a new example, then the learned probability from training data that this answer is yes is 83.5% [49, 52].

Random forests have several advantages. They are applicable to both regression and classification problems. They handle categorical predictors naturally and can deal with highly nonlinear interactions and classification boundaries. They are non-parametric, making no assumptions about the probability distributions of the data, and they're computationally simple and quick to fit, even for large problems. Moreover, random forests can compete with the best-known machine learning methods in terms of accuracy and are relatively stable, unlike individual decision trees. However, random forests are a lot less interpretable than individual decision trees [49, 52].

Artificial Neural Networks

A neural network is a powerful computational data model that can capture and represent complex input/output relationships. The motivation for the development of neural network technology stemmed from the desire to develop an artificial system that could perform "intelligent" tasks similar to those performed by the human brain [53].

A neural network usually involves many processors operating in parallel and arranged in tiers. The first tier receives the raw input information. Each successive tier receives the output from the tier preceding it, rather than from the raw input. The last tier produces the output of the system. Each processing node has its own small sphere of knowledge, including an activation function, what it has seen and any rules it was originally programmed with or developed for itself. The tiers are highly interconnected, which

means each node in tier n will receive its inputs from the nodes in tier $n-1$ to which it is connected, and its output will be the input to the nodes in tier $n+1$. The output layer may contain one or multiple nodes (see Figure 4) [54].

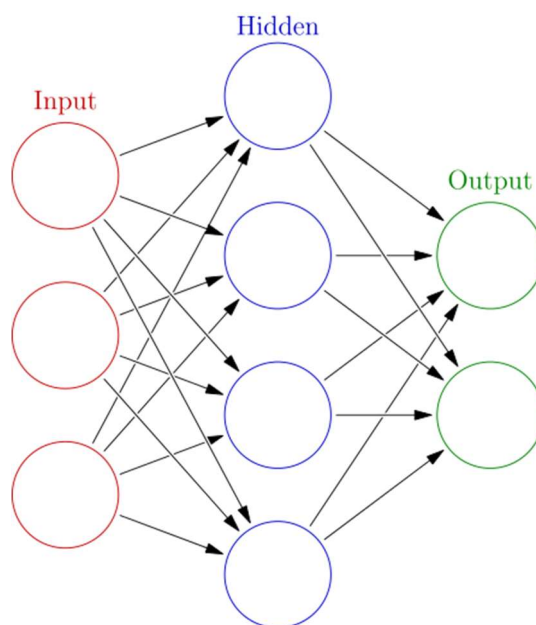


Figure 4 - Artificial Neural Network

Neural networks are notable for being adaptive, which means they modify themselves as they learn from initial training and subsequent runs provide more information about the world. The most basic learning model is centred on weighting the input streams, which is how each node weights the importance of the input from each of its predecessors. Inputs that contribute to getting right answers are weighted higher. The most common class of neural networks are called backpropagational neural networks. Backpropagation refers to the backwards propagation of the error. With backpropagation, learning is a supervised process that occurs each time the network is presented with a new input through a forward activation flow of outputs, and the backwards error propagation of weight adjustments. What this means is that when a neural network is initially presented with an input it makes a random guess as to what it might be. It then sees how far its answer was from the correct one and makes an appropriate adjustment to its connection weights [54, 55].

Several techniques may be used to decide what each node should send on to the next tier based on the inputs from the previous tier. These include gradient-based training, fuzzy logic, genetic algorithms and Bayesian methods. The network may be given some basic rules about object relationships in the space being modelled. For example, a facial recognition system might be instructed that “eyebrows are found above eyes”, or “moustaches are below a nose; moustaches are above a mouth”. Preloading rules can make training faster and make the model more powerful. However, it also builds in assumptions about the nature of the problem space, which may prove to be either irrelevant and unhelpful, or incorrect and counterproductive, making the decision about what, if any, rules to build in very important [54].

After a neural network has been trained, it will work on forward propagation mode only. New inputs are presented to the network and processed by the middle layers as though training were taking place, but

at this point the output is retained and no backpropagation occurs. The output of a forward propagation run is the predicted model for the data which can then be used for further analysis and interpretation [55].

Neural networks are sometimes described in terms of their depth, including how many layers they have between input and output, or the model's so-called hidden layers. They can also be described by the number of hidden nodes the model has or in terms of how many inputs and outputs each node has. Variations on the classic neural-network design allow various forms of forward and backward propagation of information among tiers [54].

Neural networks are well suited to problems where the relationships may be quite dynamic or non-linear. They provide an analytical alternative to conventional techniques which are often limited by strict assumptions of normality, linearity, or variable independence, among others. Because neural networks can capture many kinds of relationships they can quickly and relatively easily model phenomena which may have been very difficult or impossible to explain otherwise. The learning procedure described above is a form of supervised learning, but neural networks can be applied to unsupervised and reinforcement learning tasks as well [55].

Neural networks aren't without limitations, however. Many types of neural networks, including backpropagational ones, work like "black boxes" in the sense that the user has no role other than defining the general architecture of the network and providing it with inputs for training. The final product of this activity is a trained network that provides no equations or coefficients defining a relationship beyond its own internal mathematics. In addition, neural networks are universal approximators that work best if the system being modelled has a high tolerance to error. They excel at pattern discovery and when the relationships between variables are vaguely understood or difficult to describe with conventional approaches, but shouldn't be used to model critical systems [54, 55].

Support Vector Machines

Support Vector Machines (SVMs) are supervised learning models that can be used for classification and regression problems [56, 57].

Given a set of training data, with each instance labelled as belonging to one of two classes, a SVM training algorithm builds a model that assigns new instances to one class or another. A SVM divides the feature space into two separate parts by means of a hyperplane. The distance between the hyperplane and the closest data points is referred to as the margin. The data might be separable by several different hyperplanes, so the optimal hyperplane that can separate the two classes is the one that has the largest margin. This is called the maximum-margin hyperplane (see Figure 5). The margin is calculated as the perpendicular distance from the hyperplane to only the closest points. Only these points are relevant in defining the hyperplane and in the construction of the classifier. These points are called the support vectors. They support or define the hyperplane. The hyperplane is learned from training data using an optimization procedure that maximizes the margin [56, 57].

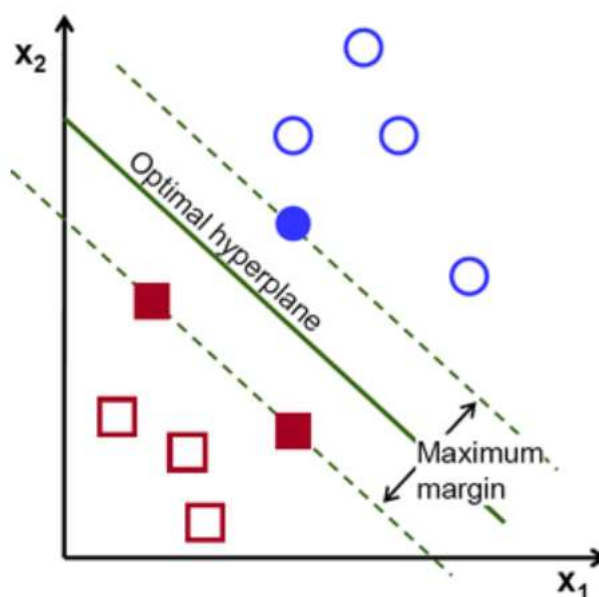


Figure 5 - Maximum-margin hyperplane

In its basic form, a SVM is a non-probabilistic binary linear classifier but extensions have been developed for multiclass classification. The dominant strategy for doing so is to reduce a single multiclass problem into multiple binary classification problems. This can be performed by either using a “one-versus-all” approach, which consists of building one SVM per class, trained to distinguish the instances in a single class from the instances in all remaining classes, or a “one-versus-one” strategy, whereby one SVM is built for each pair of classes [58]. SVMs can also perform classification on non-linear problems using the kernel trick (described in Section **Error! Reference source not found.**) to implicitly map the inputs into high-dimensional feature spaces [56].

SVMs have several advantages. They are effective in high dimensional spaces, being effective even in cases where the number of dimensions is greater than the number of instances. They are also memory efficient because SVMs only use the support vectors to maximize the margin and not all the data. In addition, SVMs are versatile by working with a number of different kernel functions. SVMs also have some disadvantages though. They perform poorly if the number of features is much greater than the number of instances, and they do not provide probability estimates naturally [59].

Regression Analysis

Regression is a statistical technique to determine the linear relationship between two or more variables. Regression is primarily used for prediction and causal inference. In its simplest (bivariate) form, regression shows the relationship between one independent variable X and a dependent variable Y , as in the formula below [60, 61]:

$$Y = \beta_0 + \beta_1 X$$

The magnitude and direction of that relation are given by the slope parameter (β_1) and the status of the dependent variable when the independent variable is absent is given by the intercept parameter (β_0). Regression thus shows how variation in one variable co-occurs with variation in another [60, 61].

Regression analysis is commonly used for modelling complex relationships among data points, such as estimating the impact of a treatment on an outcome, and extrapolating into the future. Regression

methods are also used for hypothesis testing, which involves determining whether data indicate that a presupposition is more likely to be true or false. The regression model's estimates of the strength and consistency of a relationship provide information that can be used to assess whether the findings are due to chance alone [60].

Regression analysis is not synonymous with a single algorithm. Rather, it includes a large number of methods that can be adapted to nearly any machine learning task. The most basic type of regression is called linear regression. If there is only a single independent variable, this is known as simple linear regression, otherwise it is known as multiple regression. Both of these assume that the dependent variable is continuous [60-62].

It is possible to use regression for other types of dependent variables and even for classification tasks. For instance, logistic regression can be used to model a binary categorical outcome, while Poisson regression models integer count data. Linear regression, logistic regression, Poisson regression, and many others fall in a class of models known as generalized linear models (GLM), which allow regression to be applied to many types of data. Linear models are generalized via the use of a link function, which specifies the mathematical relationship between X and Y [60-62].

3.1.5.2. Unsupervised Learning

Clustering

Clustering is an unsupervised learning method that deals with finding a structure in a collection of unlabelled data. Clustering analysis finds clusters of data objects that are similar in some sense to one another. The members of a cluster are more like each other than they are like members of other clusters. The goal of clustering is to find high-quality clusters such that the inter-cluster similarity is low and the intra-cluster similarity is high. In distance-based clustering the similarity criterion is a distance metric. Two or more objects belong to the same cluster if they are close according to a given distance. The choice of criterion depends on the final aim of the clustering [63, 64].

Cluster algorithms can be categorized based on how the underlying models operate. No type of algorithm is better than the others and the choice of algorithm is very specific to the problem at hand. The most appropriate clustering algorithm for a given problem often needs to be chosen experimentally, unless there is a reason to prefer one cluster model over another. Clustering methods can be divided into two basic types: hierarchical and partitional clustering. Within each of these types there are several different algorithms for finding clusters [63, 64].

Hierarchical clustering proceeds successively by either merging smaller clusters into larger ones, or by splitting larger clusters. Hierarchical clustering algorithms differ in the way in which they choose which two small clusters to merge, or which large cluster to split. Given the nature of the algorithms, there is no single partitioning provided, but rather a hierarchy of clusters which expand or decrease in number based on distance, the choice of distance function and linkage criterion [64].

Partitional clustering, on the other hand, attempts to directly decompose the dataset into a set of separate clusters. The choice of criterion function might highlight the local structure of the data, by assigning clusters to peaks in the probability density function for example, or it might highlight the global structure. Typically, partitional clustering involves minimizing some measure of dissimilarity between objects in each cluster, while maximizing the dissimilarity of different clusters. The most commonly used partitional clustering method is the K-Means algorithm [64].

Clustering is useful for exploring data. If there are many cases and no obvious groupings, clustering algorithms can be used to find natural groupings. Clustering can also serve as a useful data pre-processing step to identify homogeneous groups on which to build supervised models. Moreover, clustering can be used for anomaly detection. Once the data has been segmented into clusters, there might be some cases that do not fit well into any clusters. These cases are anomalies or outliers [63, 64].

Association Rule Learning

Associated rule-based learning is an unsupervised ruled-based machine learning method which aims to discover relationships in large datasets.

Rules are composed by two sets of items: the antecedent and the consequent, which compose an if-then relationship [65]. The antecedent (the if clause) is a set of attributes which are found in the data. The consequent (the then clause) is a set of items which are found in combination with the antecedent.

Furthermore, there are four relevant concepts when it comes to generating and approving rules: Support, Confidence, Lift and Conviction:

- **Support** is a measure of how frequent the antecedent appears in the dataset [66];
- **Confidence** measures how often the rule happens to be true. A rule is considered to have been followed if both the antecedents and the consequents appear together [67];
- **Lift** measures the probability of the antecedent and the consequent being independent from one another. If two events are independent between one another, no rule can be drawn to establish a relationship between them. This can be concluded when there's a Lift of 1. The more the Lift is higher than 1, the stronger the correlation between the events. Higher Lifts are particularly useful for predicting future consequents [66];
- **Conviction** is a measure of how the antecedent and consequent appear together more often than by random chance [68].

Both Lift and Conviction are considered measures of interestingness [69], i.e., they intend to select and rank itemsets according to how potentially useful they may be to the user.

In order to find possible rules and establish their validity, user-supplied thresholds of both support and confidence usually come into play, in a process composed of two steps:

- Minimum **support** threshold is applied to the dataset in order to find all frequent itemsets. This involves finding all item combinations possible, thus possibly resulting in a very large search space. The exploration of the entire search space can be avoided by exploiting the *downward-closure* property of support: for a given frequent itemset, all its subsets are also frequent, meaning that no infrequent subset can be found in a frequent itemset [66, 70];
- Minimum **confidence** threshold is then applied to these itemsets to generate rules.

Furthermore, a statistical analysis may be applied to the discovered rules in order to establish if these are statistically sound. The larger the dataset, the bigger the possibilities for rule-making, which can result in an absurd number of rules. Statistical tests for significance and independence can thus be applied in order to invalidate rules that represent frequent coincidences.

Anomaly Detection

Anomaly Detection, also known as Outlier Detection, is the process through which “non-conformant” instances can be found within datasets. An anomaly, by definition, is an instance that deviates markedly from the norm of the dataset it is a part of, and can constitute an incorrect measurement, a structural defect, or indication of suspicious activities. Furthermore, anomalies have two important characteristics:

- Their features deviate from the dataset norm;
- They are rare when compared to the normal instances [71].

Anomaly Detection can be used in a multitude of scenarios, including fraud detection, finding structural defects, identifying medical problems, etc. It be used in both supervised and unsupervised training scenarios. If used for supervised training, it will require the labelling of the dataset, while if unsupervised, it will try to score the data based only on the intrinsic properties of the dataset. With the use of a score metric, anomalies can be ranked and thresholds can be applied for further categorization.

Deviation from the norm can mean different things according to the context. Considering the two-featured dataset in Figure 6, X_1 and X_2 are easily identifiable as outliers; they are therefore considered to be **global anomalies**. On the other hand, when focusing on the C_2 cluster, the X_3 point can be considered an outlier: X_3 is what is called a **local anomaly**, since it only becomes anomalous when analysed in the context of the C_2 cluster.

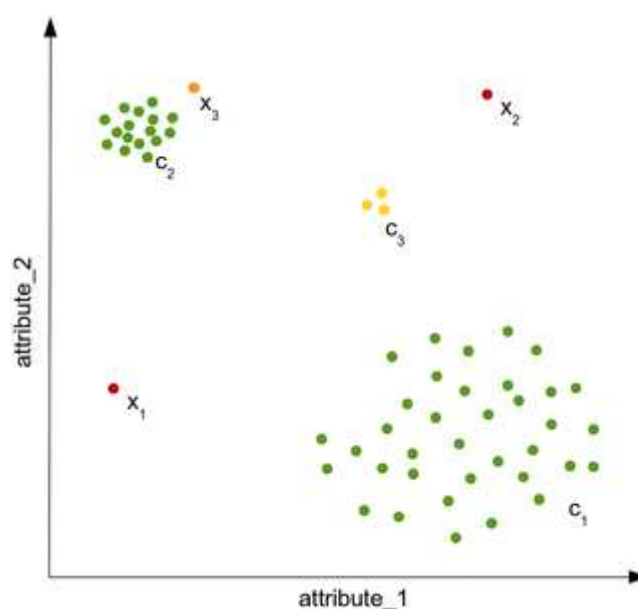


Figure 6 - Two-dimensional dataset

The cluster C_3 itself either be considered a **micro cluster** or three singular anomalies. Scoring algorithms should give the members of a micro cluster a score higher than that of the other regular instances, but lower than that of the more obvious outliers. This example serves to show that different kinds of anomalies can be discovered by applying different criteria. Considering that, three kinds of anomalies can be described:

- A single point anomaly: a single instance that is clearly identifiable as an outlier, easily identifiable through the use of *point anomaly detection* [72] algorithms;
- Collective anomalies: a collection of instances that, when together, form an outlier micro cluster. It is possible that, individually, the instances of the cluster may not be outliers, but together they may represent a larger-scoped anomalous behaviour;
- Contextual anomalies: an instance that may be an outlier and fit into the expected behaviour unless a contextual clue is added. The most common scenario is when considering the

effect of time, as certain behaviours can be normal at a certain time of the day but unexpected at others;

Collective and contextual anomaly detection tasks can be converted into point anomaly detection tasks if context or further features can be derived from the original feature set – the new dataset is commonly referred to as *data view* [73] and can differ widely from the original. It can be achieved through the use of correlation, aggregation and grouping algorithms, and requires an extensive knowledge of the dataset [74].

Expectation-Maximization Algorithm

Expectation-Maximization (EM) is a statistical iterative method to find the maximum likelihood (or maximum a posteriori – MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. The unobserved data can either be a variable that, had it been measured, would make the data objects more easily categorizable, or missing measurements.

The Expectation-Maximization algorithm is a refinement of the Maximum-Likelihood Estimation [75] for incomplete or unknown data. The optimization problem on the EM is generally more complex than that of the Maximum-Likelihood Estimation, given that the latter has a single global optimum, while the former has multiple local maxima and no closed form solution [76].

Each iteration is comprised of two steps: an expectation step (E), which computes an expected probability through the use of the latent variable as if they had been observed, and a maximization step (M), which computes parameters that maximize the expectation found in the previous step. These values are then fed to the next iteration and thus the algorithm iterates until its eventual convergence.

The probabilities for each possible completion (or hidden value) are computed, given the parameters that are already known. The resulting dataset is a weighted training dataset which includes all the possible completions, which are then used as input to compute new parameter estimates. In short, the E-step guesses at a probability distribution and uses it to fill the incomplete data and the M-step re-estimates the model parameters by using the generated data, usually by finding the maximum log-likelihood of the data [77].

3.1. State-Of-The-Practice at the Belgian Consortium

Data pre-processing

The data used in Light Lease use case is mainly sensor data that is collected by an IoT system and is saved to Barco's analytical environment in the cloud. Because serial numbers of lamps that are used in a digital cinema projectors need to be entered manually, this results in a lot of noise and poor data quality and makes it difficult to identify when a specific lamp was used in a specific projector. Therefore, a lot of energy was devoted to determine correct lamp identification based on sensor data like runtime counters.

The raw data does not directly contain data that indicates if, and when, a setting of a projector is changed. Because this is crucial information for remaining lamp lifetime estimation, a hierarchical cluster algorithm with single linkage has been evaluated on the obtained sensor data to see if the different operating modes of the projector could be detected. This proved to be the case.

Further data cleansing: outlier detection is used to remove unusual values in time series that might be caused by a sensor measurement error. This is needed to get clean light evolutions that reflect reality.

Feature engineering: The collected sensor data is re-organized and divided in time buckets of 30 minutes. Those time buckets are also the basis of the SAS analytical base table: One row in the base table represents one time bucket of 30 minutes of lamp usage.

Domain and statistical features have been inferred in order to enrich the analytical base table with powerful predictors. Domain features are based on R&D knowledge and experience while statistical features are statistics like a standard deviation that are calculated from the sensor data in a time bucket.

Analytical models

Barco has built & trained two models: a light output model and a survival model.

When the brightness of a lamp drops below the minimum allowed brightness, the lamp can no longer be used during a movie show and must be replaced. Usable lifetime of lamps has a big spread and typically lies in the range of 1000 till 4000 lamp hours. The light output model predicts the remaining useful life of a Xenon lamp that is used in a cinema projector. The model is based on similarity matching of the brightness evolution of the lamp. The brightness evolution of the lamp that needs to be scored is compared with a library of brightness evolutions of historical lamps. The brightness evolution of the historical that matches the best the brightness evolution of the lamp to be predicted is considered as the best prediction for the future brightness evolution. Before the matching can be done, the brightness evolution of each lamp is fitted by a cubic spline.

The survival model predicts the occurrence of short-term failures like a lamp shutdown/explosion failure and a strike failure. This model used is multinomial discrete time logistic hazard regression. This is a classification algorithm that outputs a probability for every failure category.

3.2. State-Of-The-Practice at the Portuguese Consortium

ISEP-GECAD began by performing a preliminary study of anomaly detection using freely available artificial datasets. This study allowed a better understanding of the issues underlying the problem of Predictive Maintenance. It was later extended to an initial sample of data collected directly from the factory machines, as well as from sensors monitoring them. The data was explored to better understand the context of the problem and discern the meaning of each variable. This information was complemented with domain knowledge provided by FACORT with the purpose of discovering meaningful relationships between variables. In addition, Feature Selection was performed by applying the mRMR algorithm to the data to detect the most redundant feature pairs and reduce its dimensionality.

Several variables of interest have been identified, such as variables concerning the spindle and the machine's operating coordinates.

Information regarding the spindle is very important, since anomalies related with this component might signal problems in a machine. The spindle has a maximum rating of 22.4 kW and can work at this rate indefinitely. Additionally, it can bear a load of 150% the maximum capacity for 30 minutes and of 200% for 3 minutes. As can be observed in Figure 7, the spindle worked below its maximum capacity for most of the day, but, as both plots show, it surpassed the 22.4 kW rate once. The event lasted for a moment, but this information can be quite useful for purposes of predictive maintenance.

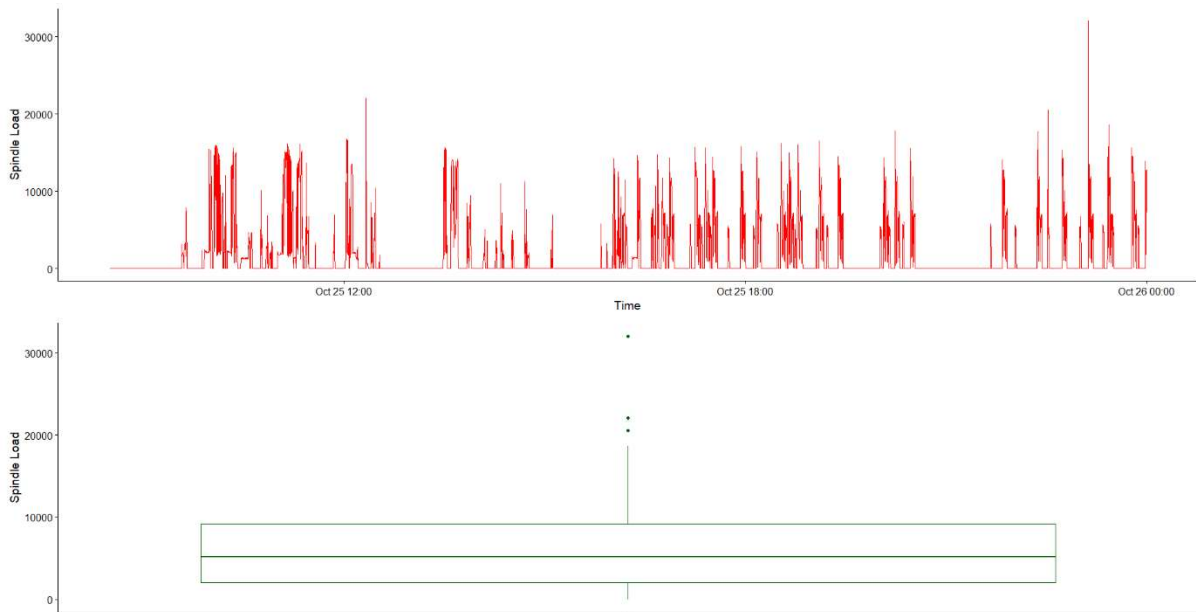


Figure 7 – Spindle Load.

Variables related to the axes' coordinates of the machine provide meaningful information regarding the production of parts. The production of different units of the same part should exhibit similar coordinate patterns, since there's a well-defined design for every part. However, because the machine operator can intervene in the production of a part and perform manual operations, these patterns can present some variation, as can be observed in Figure 8. The continuous line traces the coordinate position for axis X over time, while the dashed line shows when the production of each part began and ended. While some variation can be observed, a coordinate pattern is still obvious. When related with other variables such as the Spindle Load, this information can be useful in the detection of anomalies. Moreover, by allowing production managers to discern when and for how long a machine is stopped, this information can help them optimize the production of parts.

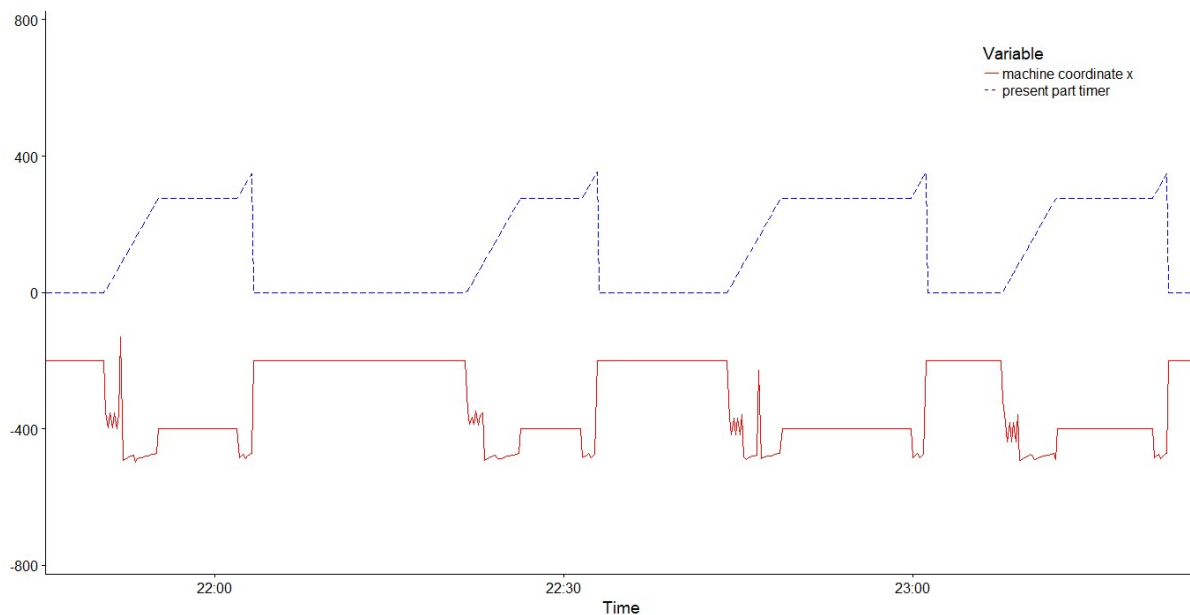


Figure 8 - Coordinate position for axis X during four production cycles.

Since processing the full data sample collected by the acquisition module requires a specific set of technologies capable of extracting knowledge from the data, a Cloudera distribution of Hadoop was installed in four virtual machines. The Hadoop multi-node cluster includes tools such as Spark, Mahout and Sqoop, and is capable of processing large quantities of data in a quick and efficient way, whether in real-time or in batch.

The historic data was transferred from a SQL Server to the Big Data platform. Considering the data was acquired from different sources (external sensors, machine protocol, etc), it was necessary to fuse the data based on its acquisition date/time to obtain a consolidated dataset. This dataset underwent subsequent cleaning and transformation processes.

Considering FACORT's equipment fails infrequently, ISEP-GECAD is now exploring one-class classification methods, such as one-class SVM and autoencoder, as a means of detecting anomalies in the machines.

3.1. State-Of-The-Practice at the Spanish Consortium

Data Analysis

The Spanish use case provides a tool for Data Analysis, which enhances the control and maintenance operations in the manufacturing line. The developed solution, M3 Analytics, is a tool that will be integrated with the M3 Software, providing analytical capabilities that enhances the traceability of the manufactured parts, which is increased by means of its integration with the ERP and MES systems in EPC.

Not only the point-clouds and virtual metrology typical analyses are provided, but also the possibility to compare each of the manufactured parts. M3 analytics is capable of comparing different features, which have to be specified in advance, such as dimensions, geometries or shapes, and provides feedback to the manufacturing and maintenance engineering departments, with high-value information. Then, this information can be used for reverse engineering purposes, and will permit the correlation between machine's performance and the parts' shape. Thus, it allows to control in a more accurate way

parameters such as tool wear, which increases the capability of the maintenance engineers to precisely schedule maintenance operations, allowing them to reduce costs directly related to tool wear and the related material wasted, such as scrapped parts. The statistical analysis performed by M3 Analytics increases traceability and allow the manufacturers to establish a better correlation between the machines and the produced parts, enhancing the global manufacturing process and easing the root cause and trend analyses.

Semantic Data Representation

As it has been explained previously, the objective of M3 Analytics is to provide statistical analyses based on virtual metrology and point cloud analysis. Due to the nature of the solution, it is essential to define the ontology in such a way that allows the different actors in the quality process to easily access to the information, and more concretely the M3 Software to be able to store the information in such a way that data can be compared and analysed, i.e. to allow the comparison between certain geometries or dimensions in an easy way. Hence, it has been necessary to implement algorithms capable of analysing the stored data of the parts and to extract the necessary information for the analyses.

3.2. State-Of-The-Practice at the Turkish Consortium

Data comes in many forms and one dimension to consider and compare differing data formats is the amount of structure contained therein. Big data analysis is the sub-area of big data concerned with adding structure to data to support decision-making as well as supporting domain-specific usage scenarios. The position of big data analysis within the overall big data value chain can be seen in figure shown below.

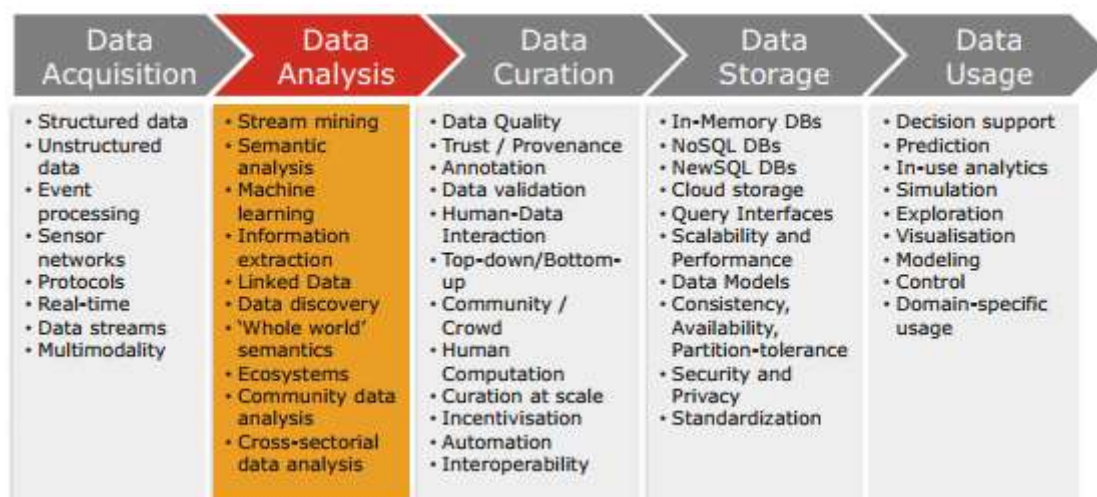


Figure 9 - Data analysis in the big data value chain

In general, data analysis includes many techniques such that semantic processing, data mining, machine learning, information extraction, and data discovery. Industry is today applying large-scale machine learning and other algorithms for the analysis of huge datasets, in combination with complex event processing and stream processing for real-time analytics. There are also calculations and aggregation processes to find statistical results from big data.

Turkish use case aims asset management and monitoring on industrial data analytics that is provided on the cloud side. To realize the use case the data is gathered from the automobile production band to cloud analytics platform that is host of several big data analytics technologies such that NoSQL Database Cassandra, Spark, REST Web Service, Scala programming language etc. Security is also enabled between the cloud side web service and the local data source. When the data is born from its source (sensor data or machine related data), it is streamed to cloud data analytics platform within the secure pipeline. InValue REST API catch the incoming data and write into Cassandra NoSQL database. ERP data is also taken with the same way from local to cloud to find new patterns from correlation of industrial and ERP data. Spark is used to find some aggregations, calculations, and patterns from correlation. These findings are written to new Cassandra tables that are read by dashboards. InValue cloud data analytics platform has the ability of realize many different data processing techniques. To realize them you just define your data structure and decide what you expect from analytics as a result.

4. Information Delivery

The amount of information generated by organizations today is bigger than ever, with the fast-paced nature of business environments and increasing competitiveness. More often than not, these large amounts of information are delivered through a number of different channels, including Enterprise Resource Planning (ERP) systems, performance scorecards, Business Intelligence reports, among others, potentially resulting in an information overload for managers and hindering their ability to make informed decisions. The relevant information must be delivered in a consistent way that is easy to read and analyse for it to be truly useful.

In order to proactively deliver information to the concerning parties, several business process oriented knowledge management approaches have been developed [78]. Traditionally, static workflow/process specifications are more typically adopted, although these are generally considered inadequate for knowledge-intensive office work processes. They support mainly routine activities for office work and are difficult to adapt to new situations. An insufficient understanding of the importance of knowledge-intensive work and adequate integration of information support and work activities results in a lack of improvement in productivity [79].

Dashboards are defined in [80] as “[...] a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance.” They present a way to provide, in a single interface, information from several sources. A dashboard aims to collect, summarize and present information so that the user can analyse several variables simultaneously. Dashboards can, therefore, be seen as data-driven decision support systems, providing information in a given format to the decision maker.

In [81], O’Donnel and David Identify three relevant features in information systems’ design that can be applied to dashboard design, namely:

- possibility of interaction and feedback;
- type of presentation format and
- differences in information load.

The level of interaction is likely to depend on the purpose and features provided by the dashboard, with feedback being appropriate for alerts and notifications and to inform a possible course of action. As for data visualization, trends and metrics can be displayed in a number of ways, such as tabular or graphical representation. Visualization is efficient when the maximum amount of data is interpreted in the minimum amount of time – to this end, several visual attributes can be used, including shapes, colours, textual information, text placement, etc. It is important to keep a good balance between visual complexity and information utility [82].

Dashboards support the decision-making process by allowing the users to monitor the variables that are important to their end-goals. These are often classified according to three categories:

- Operational dashboards, which show information concerning real-time events;
- Strategic dashboards, which report on key performance indicators and
- Analytical dashboards, showing processed data and identified trends.

Operational dashboards are concerned with rapidly-changing, real-time information. With very frequent data updates, they are designed to be consulted several times a day and are suitable for tracking events or monitor the progress towards a goal. An example of such a dashboard is shown in Figure 10, below:

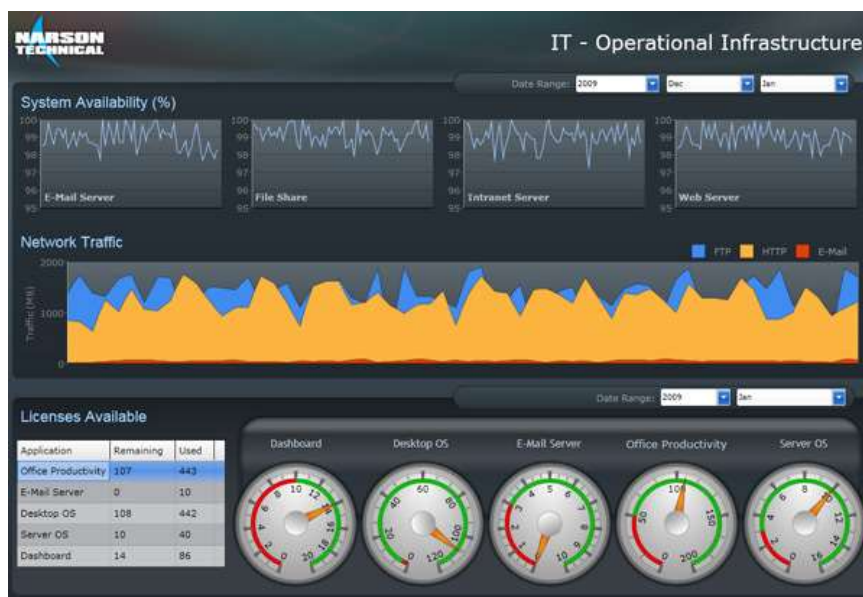


Figure 10 - Dundas operational dashboard [83]

Strategic dashboards are particularly suited for the monitorization of key performance indicators, which are typically involved in strategic decision-making. These require less frequent updates and are also focused on displaying the current status of a variable against the goals the user intends to achieve. An example of such a dashboard can be seen in Figure 11 below:

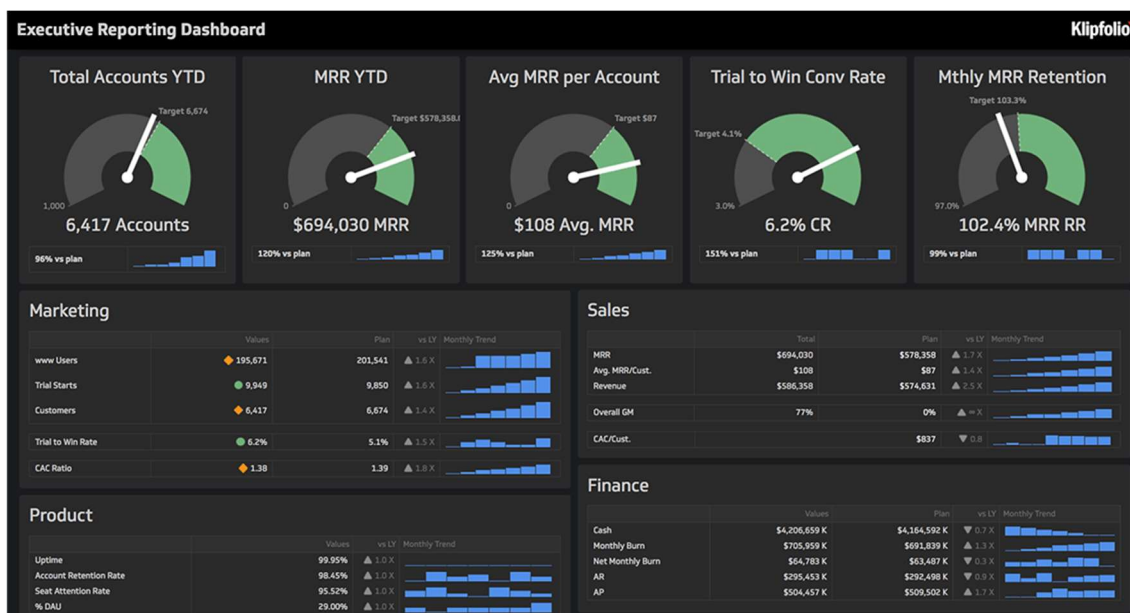


Figure 11 - Strategic Dashboard [84]

Analytical dashboards go beyond a single screen in their display of information: effectively becoming a console where one can navigate, drill down, visualize and interpret several layers of interconnected information, with exploration from summary to detail in a single visualization system [85]. These

dashboards double as cognitive tools, impacting the user's control span over heterogenous, rapidly-changing data in a digestible, easy to use fashion.



Figure 12 - Analytic Dashboard [86]

Dashboards bring benefits over traditional reporting mechanisms by supplying easy-to-read, tailored information with at-a-glance views of key performance indicators more relevant to their users' goals. More and more, they are becoming an important part of Business Intelligence and Business Performance Management. They provide an effective solution for the overwhelming amount of data generated every day in many environments, by showing dense information in an immediate, easily readable way and improving the decision-making processes and workplace productivity.

4.1. State-Of-The-Practice at the Portuguese Consortium

The goal of Information Delivery Component of INVALUE Platform is to deliver the correct information to the correct people or systems at the right moment in time.

In order to provide the FACORT Company with more proactive industrial management practices, the INVALUE platform, after taking on the tasks of acquisition, integration, merging and analysis of data from machines, sensors and documents, will make the information available to the different recipients through a Web Based Interface (WBI).

The information to be delivered to each of the recipients, may have provenance in multiple sources, being initially guaranteed the flexibility and autonomy of the platform regarding to external systems, sources of the data that gave rise to the information processed (Data acquisition systems, ERP systems, MES systems, ...).

This communication platform (WBI) with the user will be able to support:

- A fusion of service oriented and event driven architectures
- Integration of distributed and heterogeneous data sources and services
- Data transformations over information, so that data sources give their data "as-is" and
- Data processing applications receive them as they need for processing/storing

- Reliable and secure communication channels.

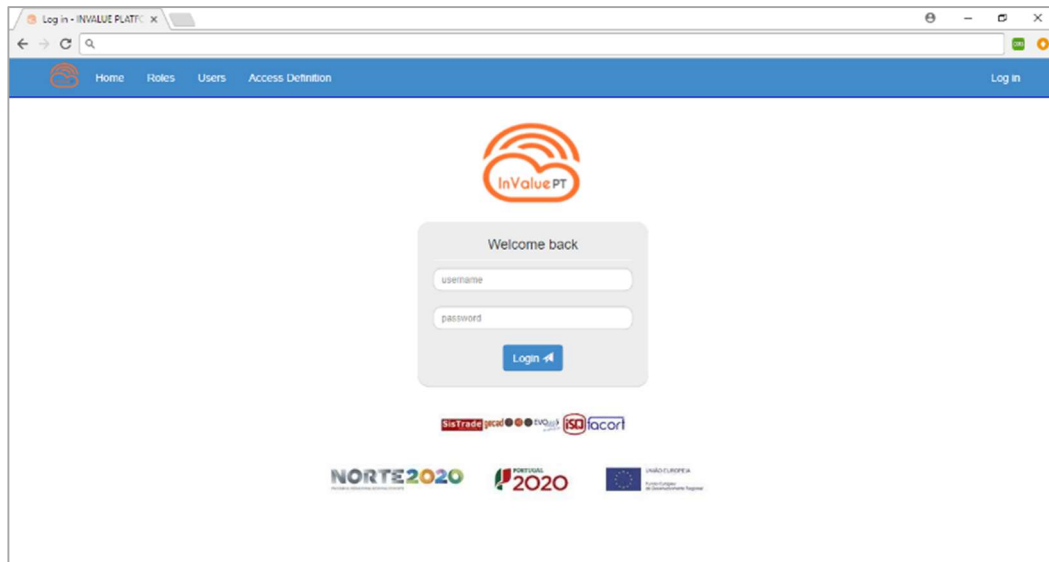


Figure 13 - Platform Access Web Interface

The system will provide a web-based interface (Figure 13) that allows the User / Machine Operator to visualize in real time, information on the operation status of the Manufacturing Equipment, process parameters (Figure 14), graphical information of values considered critical for the process (Figure 15), and at the same time provide trends graphics and predictive notifications.

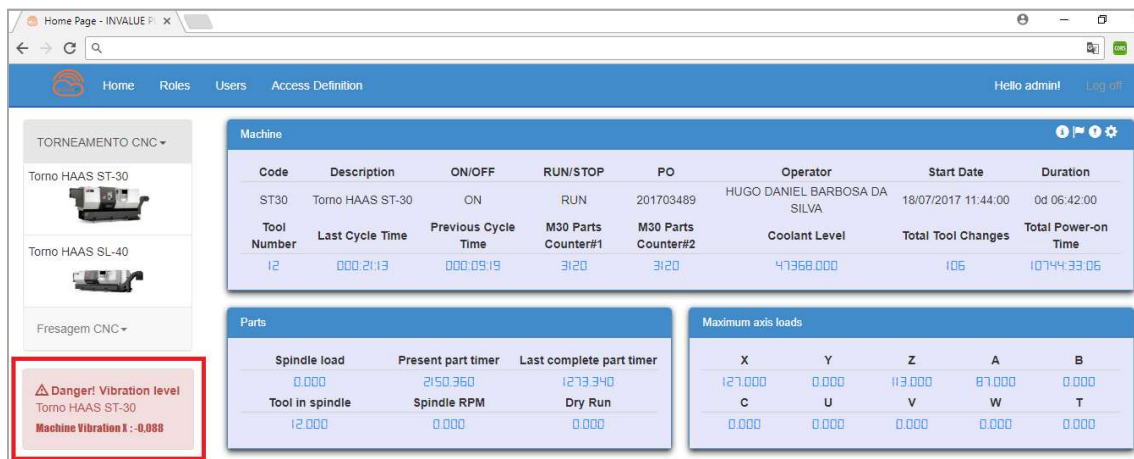


Figure 14 - Process Parameters and Machine Vibration (X Axis) Alert to the screen

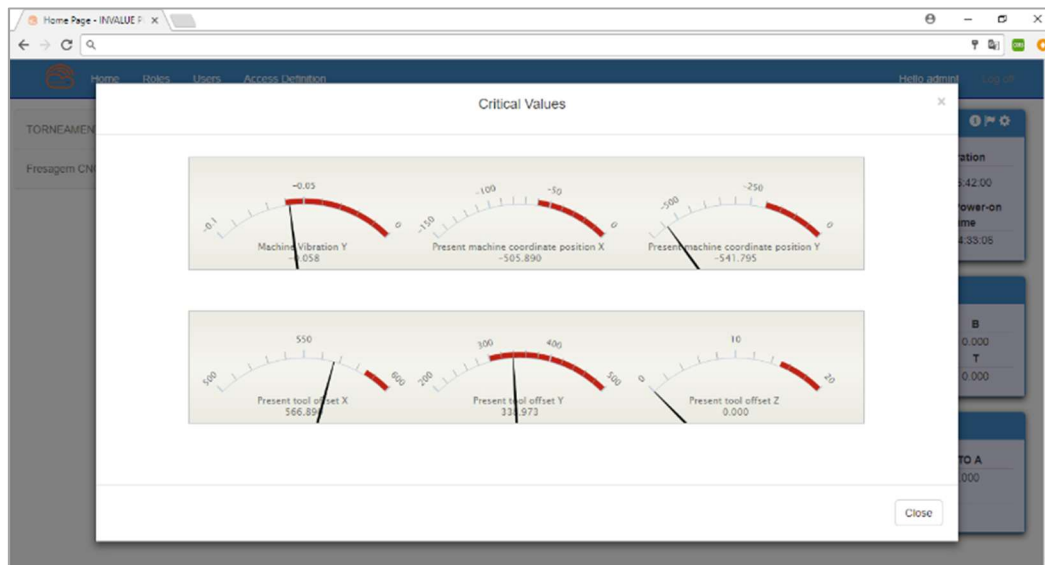


Figure 15 - Critical values

In terms of event-oriented architecture, the system will provide a set of notifications and alerts, triggered by events, some of them, configurable by the user and other pre-configured on the Platform (Figure 15).

From a functional perspective, the Platform will guarantee authorized access by User (Figure 17), different access levels by User Group according to their tasks performed in the Organization (Figure 16) so that the contents made available by the Platform are adapted according to the specific needs of each type of Recipient / User of information.

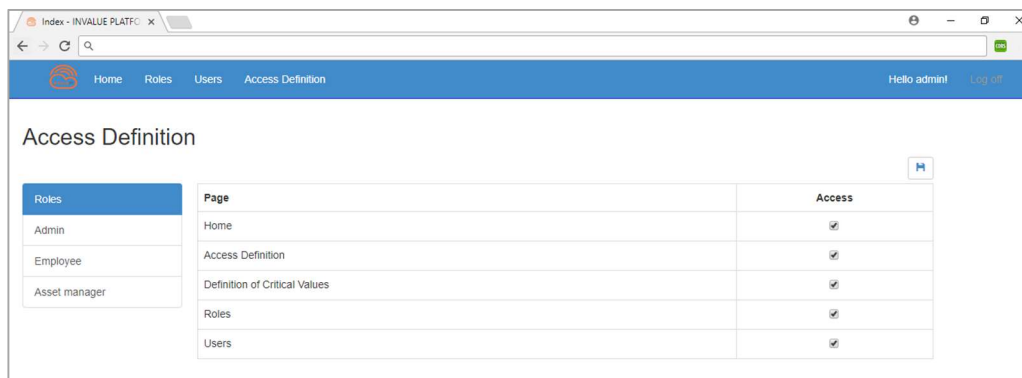
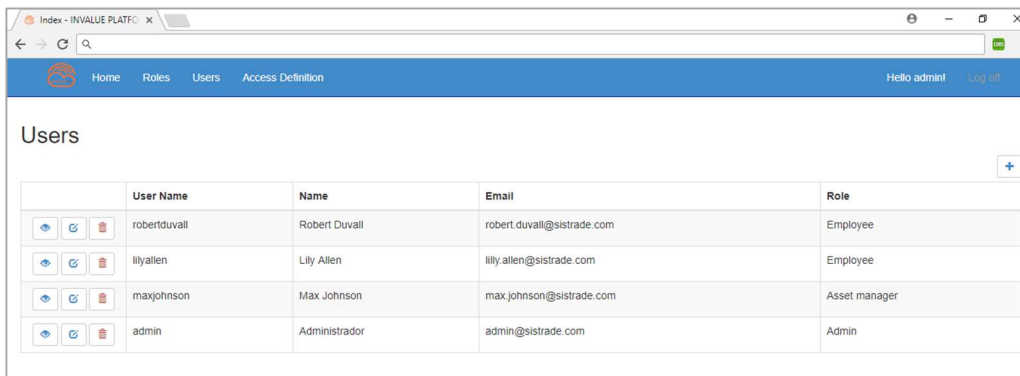


Figure 16 - Access definition per Role















	User Name	Name	Email	Role
  	robertduvall	Robert Duvall	robert.duvall@sistrade.com	Employee
  	lilyallen	Lily Allen	lily.allen@sistrade.com	Employee
  	maxjohnson	Max Johnson	max.johnson@sistrade.com	Asset manager
  	admin	Administrador	admin@sistrade.com	Admin

Figure 17 - User Roles

SISTRADE took care to ensure that the technical and functional specification of this component (Information Delivery), obeys to the Market best practices and the most advanced techniques for the purpose for which it is intended.

4.2. State-Of-The-Practice at the Spanish Consortium

The goal is to deliver the correct information to the correct people or systems at the right moment in time. The objectives have been:

- ✓ Integration of distributed and heterogeneous data sources and services
- ✓ A fusion of service oriented and event driven architectures
- ✓ Data transformations over information, so that data sources give their data “as-is” and data processing applications receive them as they need for processing/storing
- ✓ Reliable and secure communication channels

The Spanish partners defined information delivery/exchange/interoperability:

- The reports and statistical analyses will be available for all the systems. Data interchangeability is granted.
- QIF data information of the parts will be accessible.
- Reports & Statistical Analyses always available.
- Process (Manufacturing Order) information can be exchanged among systems.
- CAD models readable by M3.

Spanish partners worked on the information and data sharing following actual industrial standards. So QIF, promoted by the NIST, has been selected as the most suitable to share all quality related information with other solutions. It is based on XML coded tables, where nominal and measured values, tolerances, deviations and so on are written. This work has ensured interoperability and data integration.

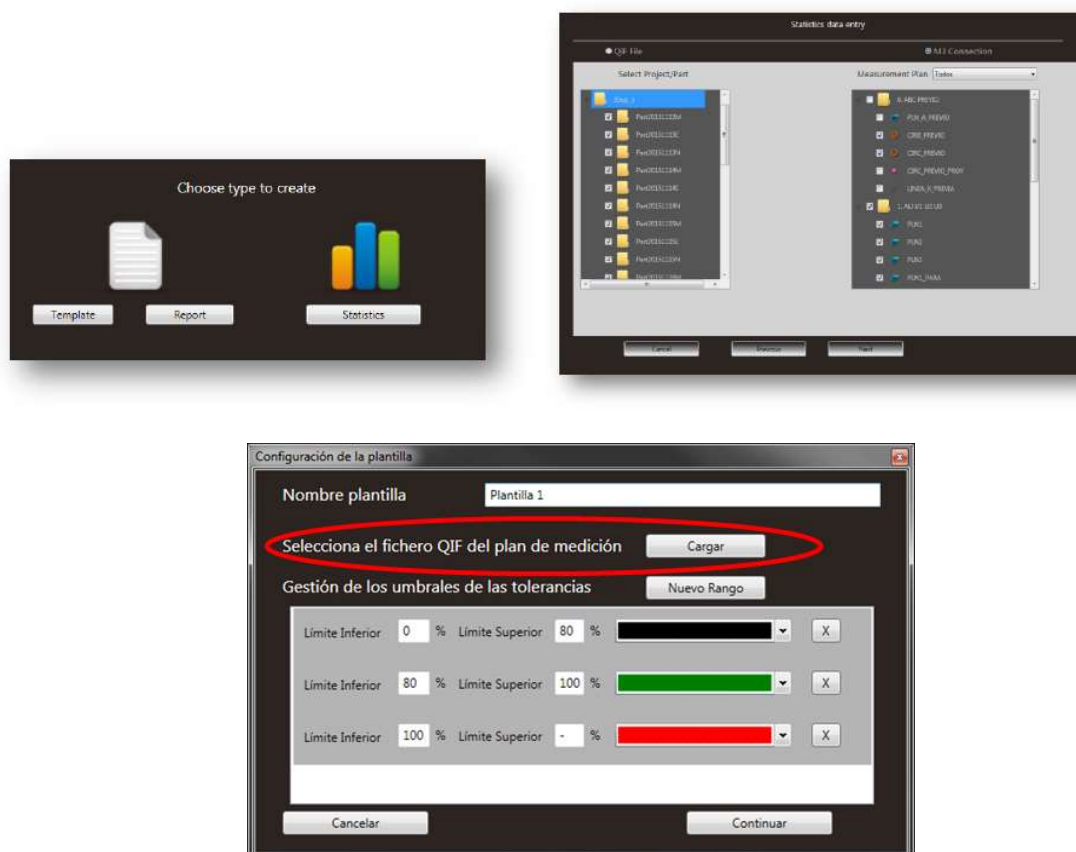


Figure 18 - Demo Screens

4.3. State-Of-The-Practice at the Turkish Consortium

The findings of analytics platform are written to new Cassandra tables that are read by dashboards in Turkey InValue use case. Information delivery includes (1) how the results should be viewed to customers? and (2) the data visualization. There are some designated dashboards for reporting and visualization of the information. You can see the dashboard previews as follows. The dashboards can be reached out according to analytics techniques and also according to data.

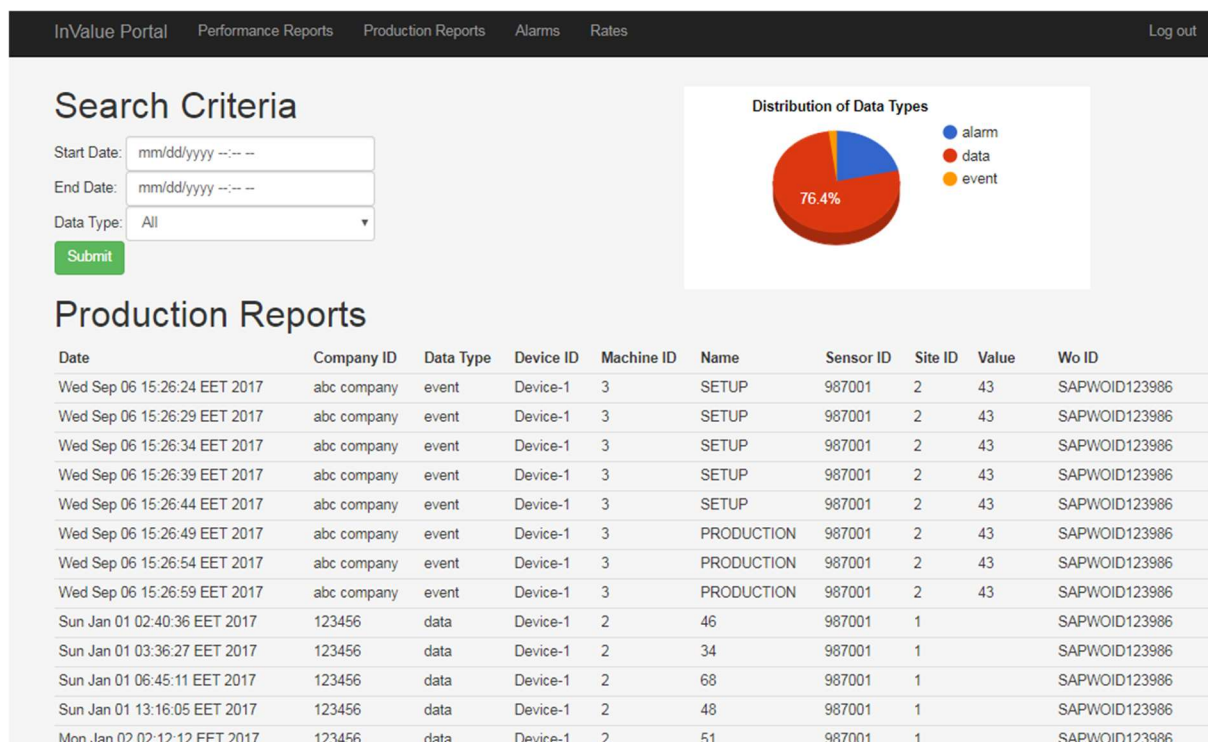


Figure 19 - InValue Platform reports

The data collected on the platform can be filtered by the users depending on desired parameters. Graphical reports can be seen on the screen to analyse historical data. There are predefined reports already exists on platform for manufacturing business and it can be enhanced with other type of reports for different business types also.

The InValue Platform not only collects the data from the sensors on the devices of production line but also collects the data from the ERP system of the manufacturing company which allows combination of process, operator and the sensor data on the same data scheme. Each user can access only to his company data related with his user name. Company configuration on the platform holds the data type and manufacturing specific information and can relate them with operators and production line. With this integration, users can follow the operator performance, product faults and product line efficiency. Furthermore, this infrastructure and data model easily can be remodelled for other business areas outside of automotive production.

5. Conclusion

This document detailed the two major steps of Big Data management: Data Acquisition and Data Processing. For each one of these steps, the approach taken by each country consortium was described.

Section 2 described some existing Data Acquisition platforms, followed by a brief description of the implementation addressed by each country consortium.

Section 3 presented some of the technologies available for processing and analysing the vast amounts of data generated by the manufacturing industry. In particular, technical options for working with Big Data, including data storage, integration and analysis, with a focus on the Hadoop framework and associated tools were evaluated. Furthermore, different algorithms and methods for data processing and intelligent data analysis were also analysed. These technologies and methods are in line with the requirements defined in Work Package 1 (WP1 – D1.2), such as “data fusion and homogenization”, which is part of the pre-processing phase, and “storage and analysis of large volumes of data”, which requires the use of Big Data technologies and analytical models.

Information delivery methods were described in section 4, as well as the implementations done by each country consortium.

References

1. Wang, K.-S., *Towards zero-defect manufacturing (ZDM)—a data mining approach*. *Advances in Manufacturing*, 2013. **1**(1): p. 62-74.
2. Joseph, J., O. Sharif, and A. Kumar, *Using Big Data for Machine Learning Analytics in Manufacturing*. 2014, TATA CONSULTANCY SERVICES.
3. Beyer, M., *Gartner Says Solving 'Big Data' Challenge Involves More Than Just Managing Volumes of Data*. Gartner. Archived from the original on, 2011. **10**.
4. Laney, D., *3D data management: Controlling data volume, velocity and variety*. META Group Research Note, 2001. **6**: p. 70.
5. EDLICH, P. *NOSQL Databases*. 2016 [cited 2016 06/12/2016]; Available from: <http://nosql-database.org/>.
6. Foundation, T.A.S. *Apache Cassandra*. 2016 06/12/2016]; Available from: <http://cassandra.apache.org/>.
7. Foundation, T.A.S. *Welcome to Apache Hadoop*. 2016 [cited 2016 06/12/2016]; Available from: <http://hadoop.apache.org/>.
8. IBM. *What is the Hadoop Distributed File System (HDFS)?* 2016 06/12/2016]; Available from: <https://www-01.ibm.com/software/data/infosphere/hadoop/hdfs/>.
9. Borthakur, D. *HDFS Architecture Guide*. 2013 [cited 2016 06/12/2016]; Available from: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html.
10. Foundation, T.A.S. *Hadoop MapReduce Tutorial* 2013 06/12/2016]; Available from: https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html.
11. Works, H. *MapReduce*. 2016 06/12/2016]; Available from: <http://hortonworks.com/apache/mapreduce/>.
12. Foundation, T.A.S. *Apache Storm*. 2016 06/12/2016]; Available from: <http://storm.apache.org/>.
13. Foundation, T.A.S. *Apache Storm Tutorial*. 2016 [cited 2016 06/12/2016]; Available from: <http://storm.apache.org/releases/current/Tutorial.html>.
14. Foundation, T.A.S. *Apache Storm Concepts*. 2016 [cited 2016 06/12/2016]; Available from: <http://storm.apache.org/releases/current/Concepts.html>.
15. Foundation, T.A.S. *Apache Spark - Lightning-Fast Cluster Computing*. 2016 [cited 2016 12/12/2016]; Available from: <http://spark.apache.org/>.
16. Foundation, T.A.S. *Apache Storm Multi-Language*. 2016 [cited 2016 06/12/2016]; Available from: <https://storm.apache.org/about/multi-language.html>.
17. Noyes, K. *Five things you need to know about Hadoop v. Apache Spark* | *InfoWorld*. 2015 2015-12-11T14:07:05:00 [cited 2016 12/12/2016]; Available from: <http://www.infoworld.com/article/3014440/big-data/five-things-you-need-to-know-about-hadoop-v-apache-spark.html>.
18. Zaharia, M., et al., *Spark: cluster computing with working sets*. HotCloud, 2010. **10**: p. 10-10.
19. Sigkdd. *Data Mining Curriculum: A Proposal*. 2016 [cited 2016 12/12/2016]; Available from: <http://www.kdd.org/curriculum/index.html>.
20. Foundation, P.S. *Python.org*. 2016 [cited 2016 12/12/2016]; Available from: <https://www.python.org/>.
21. Media, D. *Python Packages For Data Mining*. 2015 2015-01-05 [cited 2016 12/12/2016]; Available from: <http://dataconomy.com/2015/01/python-packages-for-data-mining/>.
22. NumPy. *NumPy*. 2016 [cited 2016 12/12/2016]; Available from: <http://www.numpy.org/>.
23. SciPy. *SciPy.org*. 2016 [cited 2016 13/12/2016]; Available from: <http://scipy.org/>.
24. PyData. *Pandas: Python Data Analysis Library*. 2016 [cited 2016 13/12/2016]; Available from: <http://pandas.pydata.org/>.
25. Hunter, J., et al. *Matplotlib 1.5.3 documentation*. 2016 [cited 2016 13/12/2016]; Available from: <http://matplotlib.org/>.
26. scikit. *scikit-learn: machine learning in Python*. 2016 [cited 2016 13/12/2016]; Available from: <http://scikit-learn.org/stable/>.
27. Ipython. *IPython*. 2016 [cited 2016 13/12/2016]; Available from: <https://ipython.org/>.

28. Foundation, T.R. *R: The R Project for Statistical Computing*. 2016 [cited 2016 13/12/2016]; Available from: <https://www.r-project.org/>.
29. Wickham, H. *ggplot2*. 2013 [cited 2016 13/12/2016]; Available from: <http://ggplot2.org/>.
30. Rstudio. *Shiny*. 2016 [cited 2016 07/12/2016]; Available from: <https://shiny.rstudio.com/>.
31. Rstudio. *Introducing dplyr*. 2014 2014-01-17 [cited 2016 07/12/2016]; Available from: <https://blog.rstudio.org/2014/01/17/introducing-dplyr/>.
32. DataCamp. *Choosing R or Python for data analysis?* 2016 [cited 2016 07/12/2016]; Available from: <http://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis>.
33. Lee, C.H. *How to Choose Between Learning Python or R First*. 2015 2015-01-12 [cited 2016 06/12/2016]; Available from: <http://blog.udacity.com/2015/01/python-vs-r-learn-first.html>.
34. Warwick. *Combining R and Python*. 2016 [cited 2016 06/12/2016]; Available from: http://www2.warwick.ac.uk/fac/sci/moac/people/students/peter_cock/r/rpy/.
35. Gautier, L. *rpy2: R in Python*. 2016 [cited 2016 06/12/2016]; Available from: <http://rpy2.bitbucket.org/>.
36. Bellosta, C.J.G. *R Package rPython*. 2016 [cited 2016 06/12/2016]; Available from: <http://rpython.r-forge.r-project.org/>.
37. Rapidminer. *RapidMiner*. 2016 [cited 2016 06/12/2016]; Available from: <https://rapidminer.com/>.
38. Goopta, C. *Six of the Best Open Source Data Mining Tools - The New Stack*. 2014 2014-10-07 [cited 2016 12/12/2016]; Available from: <http://thenewstack.io/six-of-the-best-open-source-data-mining-tools/>.
39. Waikato, U.o. *Weka 3 - Data Mining with Open Source Machine Learning Software in Java*. 2016 [cited 2016 12/12/2016]; Available from: <http://www.cs.waikato.ac.nz/ml/weka/>.
40. Bioinformatics Laboratory, U.o.L. *Orange Data Mining*. 2016 [cited 2016 12/12/2016]; Available from: <http://orange.biolab.si/>.
41. Knime. *KNIME | Open for Innovation*. 2016 [cited 2016 06/12/2016]; Available from: <https://www.knime.org/>.
42. Zhang, W. *Data Preprocessing*. 2011 [cited 2016 13/12/2016]; Available from: <http://www.cs.wustl.edu/~zhang/teaching/cs514/Spring11/Data-prep.pdf>.
43. Guyon, I. and A. Elisseeff, *An introduction to variable and feature selection*. Journal of machine learning research, 2003. **3**(Mar): p. 1157-1182.
44. Brownlee, J. *Discover Feature Engineering, How to Engineer Features and How to Get Good at It*. 2014 6/12/2016]; Available from: <http://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/>.
45. Bouchard-Côté, A. *Feature Engineering and Selection*. 2009 [cited 2016 6/12/2016]; Available from: <https://people.eecs.berkeley.edu/~jordan/courses/294-fall09/lectures/feature/slides.pdf>.
46. Shlens, J., *A tutorial on principal component analysis*. arXiv preprint arXiv:1404.1100, 2014.
47. Filter, L.V. and P. Filter, *Seven Techniques for Dimensionality Reduction*. 2014.
48. Kim, E., *Everything you wanted to know about the kernel trick*. 2013, jan.
49. Wise.io, I. *Practical Machine Learning in Production: Using Decision Forests*. 2014 12/12/2016]; Available from: <http://info.wise.io/machine-learning-white-paper>.
50. Russell, S.J., et al., *Artificial intelligence: a modern approach*. Vol. 2. 2003: Prentice hall Upper Saddle River.
51. Pham, D. and A. Afify, *Machine-learning techniques and their applications in manufacturing*. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 2005. **219**(5): p. 395-412.
52. Cutler, A. *Random Forests for Regression and Classification* 2010; Available from: <http://www.math.usu.edu/adele/RandomForests/Ovronnaz.pdf>.
53. Shiffman, D. *The Nature of Code*. 2012 [cited 2016 12/12/2016]; Available from: <http://natureofcode.com/book/chapter-10-neural-networks/>.
54. Rouse, M. *Neural Network*. 2016 [cited 2016 12/12/2016]; Available from: <http://searchnetworking.techtarget.com/definition/neural-network>.
55. Burguer, J. *A Basic Introduction to Neural Networks*. 1996 12/12/2016]; Available from: <http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html>.
56. Burges, C.J., *A tutorial on support vector machines for pattern recognition*. Data mining and knowledge discovery, 1998. **2**(2): p. 121-167.

57. Hansson, K., et al., *Machine learning algorithms in heavy process manufacturing*. American Journal of Intelligent Systems, 2016. **6**(1): p. 1-13.
58. Hsu, C.-W. and C.-J. Lin, *A comparison of methods for multiclass support vector machines*. IEEE transactions on Neural Networks, 2002. **13**(2): p. 415-425.
59. scikit, I. *Support Vector Machines*. 2017 [27/01/2017]; Available from: <http://scikit-learn.org/stable/modules/svm.html>.
60. Lantz, B., *Machine learning with R*. 2013: Packt Publishing Ltd.
61. Campbell, D. and S. Campbell. *Introduction to regression and data analysis*. in *Stat Lab Workshop*. 2008.
62. Ray, S., *7 Types of Regression Techniques you should know*. 2015.
63. Hartigan, J.A., *Clustering algorithms*. 1975.
64. Kaski, S. *Data exploration using self-organizing maps*. in *ACTA POLYTECHNICA SCANDINAVICA: MATHEMATICS, COMPUTING AND MANAGEMENT IN ENGINEERING SERIES NO. 82*. 1997. Citeseer.
65. Agrawal, R., T. Imieliński, and A. Swami. *Mining association rules between sets of items in large databases*. in *Acm sigmod record*. 1993. ACM.
66. Hasher, M., et al., *Introduction to arules—A computational environment for mining association rules and frequent item sets, 2007*.
67. Hahsler, M., *A probabilistic comparison of commonly used interest measures for association rules*. Available online: http://michael.hahsler.net/research/association_rules/measures.html, 2015.
68. Brin, S., et al. *Dynamic itemset counting and implication rules for market basket data*. in *ACM SIGMOD Record*. 1997. ACM.
69. Geng, L. and H.J. Hamilton, *Interestingness measures for data mining: A survey*. ACM Computing Surveys (CSUR), 2006. **38**(3): p. 9.
70. Tan, P.-N., M. Steinbach, and V. Kumar, *Chapter 6. Association analysis: basic concepts and algorithms*. *Introduction to Data Mining*. 2005, Addison-Wesley Boston.
71. Grubbs, F.E., *Procedures for detecting outlying observations in samples*. Technometrics, 1969. **11**(1): p. 1-21.
72. Chandola, V., A. Banerjee, and V. Kumar, *Anomaly detection: A survey*. ACM computing surveys (CSUR), 2009. **41**(3): p. 15.
73. Goldstein, M., *Anomaly Detection*, in *RapidMiner: Use Cases and Business Analytics Applications*, M. Hofmann and R. Klinkenberg, Editors. 2013, Chapman and Hall/CRC. p. 367-394.
74. Goldstein, M. and S. Uchida. *Behavior analysis using unsupervised anomaly detection*. in *The 10th Joint Workshop on Machine Perception and Robotics (MPR 2014)*. Online. 2014.
75. Dempster, A.P., N.M. Laird, and D.B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*. Journal of the royal statistical society. Series B (methodological), 1977: p. 1-38.
76. Borman, S., *The expectation maximization algorithm—a short tutorial*. Submitted for publication, 2004: p. 1-9.
77. Do, C.B. and S. Batzoglou, *What is the expectation maximization algorithm?* Nat Biotech, 2008. **26**(8): p. 897-899.
78. Abecker, A., Bernardi, A., Maus, H., Sintek, M., and Wenzel, C., *Information supply for business processes – coupling workflow with document analysis and information retrieval in Knowledge-Based Systems, Special Issue on AI in Knowledge Management*, 2000, 13(5):271–284.
79. Schütt, P., *The post-Nonaka Knowledge Management in Journal of Universal Computer Science*, 2003, 9(6):451–462.
80. Few S., *Information dashboard design, the effective visual communication of data*. O'Reilly Media, Inc., 2006.
81. O'Donnell E, David JS., *How information systems influence user decisions: a research framework and literature review*. in *Int J Account Inf Syst*, 2000;1:178–203
82. Yigitbasioglu, O. M., & Velcu, O., *A review of dashboards in performance management: Implications for design and research*, in *International Journal of Accounting Information Systems*, 2012, 13(1), 41-59.
83. Dundas, *Types of Dashboards: The Operational Dashboard*. Available online: <http://www.dundas.com/support/blog/types-of-dashboards-the-operational-dashboard>

84. Klipfolio, *Dashboard Reporting. Choosing the right type of dashboard for your business*, Available online: <https://www.klipfolio.com/resources/articles/operational-analytical-bi-dashboards>
85. Dubriwny, D. and Rivards, K., *Are you drowning in BI reports? Using analytical dashboards to cut through the clutter*. April 2004.
86. InetSoft, *Dashboards Reporting for Enterprise Asset Management*, Available online: https://www.inetsoft.com/solutions/eam_enterprise_asset_management_dashboard_reporting/