PROTEUS

Scalable online machine learning for predictive analytics and real-time interactive visualization

687691

D2.5 Catalogue of scientific and technical requirements

Lead Author: Rubén Casado
With contributions from: Álvaro Agea, Asterios Katsifodimos
Reviewers: Hamid Bouchachia

Deliverable nature:	Report (R)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	31/05/16
Actual delivery date:	31/05/16 (first submission); 31/07/2017 (re-submission)
Version:	1.2
Total number of pages:	79
Keywords:	Software requirements, non-functional requirements

Abstract

The main goal of this deliverable is to define the software requirements that PROTEUS project must fulfil during its lifecycle. The software requirements are derived from the functional needs of the ArcelorMittal real scenario. This document reviews the scenario from both functional and technical point of views. Specific requirements of each area (predictive analytics, visual analytics and data processing) are analysed taking into account functional needs, current technologies and scientific gaps. As conclusion, the Catalogue of Software requirements that will drive all software developments tasks is presented.

Executive summary

In steelmaking industry, the life-cycle of steel is long: from the raw material extraction to the manufacturing of final products. A key phase of the steel production is performed in the Hot Strip Mill and this facility will be essential to carry on the early detection of defects. All processes there are monitored using real-time sensors that produce extremely large and diverse structured and unstructured data streams. The main objectives are to determine the principal process variables that originate these defects and to detect future defective coils while they are still in production in the Hot Strip Mill (HSM).

From a technical point of view, three main areas requirements:

- **Predictive analytics**: The goal is to formalize a multivariate model or choicely two univariate models able to predict accurately the symmetry ad asymmetry indexes with respect to other process variables. So the model/s will help to decide if the coils can be considered defective or not. A key need is that the model has to deal with massive streaming data, not only with historical information.
- **Visual analytics**: Low latency of visual interaction is a must, as they will allow the end-users to have a general initial analysis of the data with a visual inspection. And all the visualization tasks should deal with the characteristics of streaming data.
- **Data processing**: It is necessary to process both data-at-rest and data-in-motion to provide the results. Data-at-rest includes both quantitative and qualitative data. Data-at-rest data comes from the system that measures flatness and from different sensors installed in the hot strip mill.

In terms of predictive analytics, the proposed solution should be able to provide near-real time analysis of the HSM process, discover unexpected situations, anomalies in the process, and existing patterns. The proposed machine learning pipeline should be parallelizable, scalable, and provide incremental processing of the data. The previous requirements are not fulfilled for existing solutions.

With regard to the visual analytics, the requirements can be summarized as ability to load real-time streaming data, interactive visualisations, support for massive volume of data points, suitability for scientific data analytics and machine learning, usability and enough extensibility to make custom visualisations. While some of the studied libraries meet one or more of the requirements, there is none that meets all of them at the same time.

There are several streaming and batch data processing systems including Apache Flink and Apache Spark. That is, they support streaming and batch processing in their engine. However, combining batch and stream processing in one system or running them in one in one system as a whole is still a scientific gap.

Based on the scenario requirements and analysis of the current solutions, this deliverable presents the catalogue of software requirements that will drive all software developments tasks

Document Information

IST Project Number	687691	**Acronym**	PROTEUS
Full Title	Scalable online machine learning for predictive analytics and real-time interactive visualization		
Project URL	http://www.proteus-bigdata.com/		
EU Project Officer	Martina EYDNER		

Deliverable	**Number**	D2.5	**Title**	Catalogue of scientific and technical requirements
Work Package	**Number**	WP2	**Title**	Industrial case: requirements, challenges, validation and demonstration

Date of Delivery	**Contractual**	M06	**Actual**	M06
Status	version 1.1		final x	
Nature	report x demonstrator □ other □			
Dissemination level	public x restricted □			

Authors (Partner)	Rubén Casado (TREE), Álvaro Agea (LMBDP), Asterios Katsifodimos (DFKI)			
Responsible Author	**Name**	Rubén Casado	**E-mail**	ruben.casado@treelogic.com
	Partner	Treelogic	**Phone**	+34 902 286 386

Abstract (for dissemination)	The main goal of this deliverable is to clearly define the software requirements that the PROTEUS project must fulfil during its lifecycle. The software requirements are derived from the functional needs of the ArcelorMittal real scenario. This document reviews the scenario from both functional and technical point of views. Specific requirements of each area (predictive analytics, visual analytics and data processing) are analysed taking into account functional needs, current technologies and scientific gaps. As conclusion, the Catalogue of Software requirements that will drive all software developments tasks is presented.
Keywords	Software requirements, non-functional requirements

Version Log			
Issue Date	**Rev. No.**	**Author**	**Change**
14/01/2016	0.0.1	Ignacio García (TREE)	Initial ToC
20/01/2016	0.0.2	Rubén Casado (TREE)	Initial ToC v2
25/04/2016	0.1	Asterios Katsifodimos (DFKI)	DFKI contributions
26/04/2016	0.2	Alvaro Agea (LMBDP)	LMBDP contributions
23/05/2016	1.0	Rubén Casado (TREE)	Ready to review
30/05/2013	1.05	Asterios Katsifodimos (DFKI)	DFKI contributions after internal review
31/05/2016	1.1	Rubén Casado (TREE)	Final version
31/07/2017	1.2	Marcos Sacristán (TREE)	Re-submission version: revision after rejection and feedback for improvements

Table of Contents

1 Introduction

The main goal of this deliverable is to clearly define the software requirements that the PROTEUS project must fulfil during its lifecycle. The software requires are derived from the functional needs of the ArcelorMittal real scenario.

Firstly, this document reviews the scenario from both functional and technical point of view. Then, Sections 2 to 4 analyse the specific requirements of each area (predictive analytics, visual analytics and data processing) in order to provide, in Section 5, the target software requirements of this project. Section 5, therefore, presents the Catalogue of Software requirements that will drive all software developments tasks. This catalogue will be used to evaluate the advances of the project as well as a mechanism of tradability to ensure the achievement of the PROTEUS goals.

1.1 Scenario: functional review

In steelmaking industry, the life-cycle of steel is long: from the raw material extraction to the manufacturing of final products, such as screws, cans and vehicles, several weeks can pass in between. Nevertheless, the detection of defects in early stages of the process of steel production is a key point as they have a great economic impact due to the costs of producing posterior transformations from the defective products which will not serve for their purposes (will be finally rejected). Thus, the sooner the defects are detected, the sooner the process can be modified/stopped in order to save these expenses.

A key phase of the steel production is performed in the Hot Strip Mill and this facility will be essential to carry on the early detection of defects. Hot Strip Mill is a facility where steel is transformed from slabs to coils after heating and rolling the material through rolls at high pressure and temperature, while keeping a controlled tension over the material, and finally cooling by using water showers in a continuous process. All processes are monitored using real-time sensors that produce extremely large and diverse structured and unstructured data streams.

There are three main parameters to take into account regarding the final product. A key benefit for industrial managers would be the prediction of thickness, width and flatness measurement of the coils. The later one is the hardest to predict due to its variability. ArcelorMittal has developed a sensor system capable to detect a defect on a coil and to assign several indexes measuring the degree of flatness of the coils.

It is required to implement an early defect detection system able to predict the distribution of the defects over the coil and also their severity, as not all the defects will imply that the coil has to be rejected. Moreover, apart from the early detection, it seems important to identify which variables are relevant for the appearance of flatness defects.

To predict the flatness measurement, it is necessary to deal with a continuous learning process as steel composition varies with time, and so does its mechanical behaviour. Another problem that should be faced is the lack of data due to sensor malfunction. Visualization methods for understanding the process are also essential. By relying on visual interactive interfaces, a better perception of the data and the analysis outcome will be obtained.

1.2 Scenario: technical review

The control and tracking of defects in coils is a key problem for ArcelorMittal. The reason is that it derives in an important economic impact due to the time and resources invested to manufacture a coil which will be rejected afterwards. Thus, the main objectives are to determine the principal process variables that originate these defects and to detect future defective coils while they are still in production in the Hot Strip Mill, a facility which plays a key role in the steel production process and where many of the defects arise.

From a technical point of view, three main areas are affected:

- **Predictive analytics**: The main goal is the prediction of the flatness in the coils. In addition, it is also needed to obtain a predictive model able to provide the amount and the regions of the coils which have a higher probability of having flatness problems, or in other words, a prediction of the flatness distribution. Thus, the goal is to formalize a multivariate model or choicely two univariate models able to predict accurately the symmetry ad asymmetry indexes with respect to other process variables. So

the model/s will help to decide if the coils can be considered defective or not. Another important goal is that the model to be formalized has to be dynamic. In fact, due to the type of data considered, the model has to adapt and overcome the possible inclusion of incorrect data, the lack of partial data or the fact that some delay can be included before obtaining some data. A key need is that the model has to deal with massive streaming data, not only with historical information.

- **Visualization tools**: Sometimes predictive models are difficult to understand by operators, so it is necessary to join these models with powerful visualization tools. A key requirement is the ability to visualize metrics about the impact of different process parameters in the presence of flatness defects in coils. With this visualization of the metrics, an alarm system may be constructed where the operator can see and easily decide if the on-going coil is a defective candidate or not. To this extent, the visualization system should be able to relate the output of the application with historical process data of past defective coils. Moreover, it should be possible to provide a visual representation of the important variables, so that some hidden information can be extracted or discovered. Finally, it would be advantageous that the visualization tool is interactive and can be customized by the user. Low latency of visual interaction is a must, as they will allow the end-users to have a general initial analysis of the data with a visual inspection. For instance, by allowing us to zoom certain regions, to include more data so that the accuracy and the details of the visualization are richer, or to colour the regions depending on certain goals or variables. And all the visualization tasks should deal with the characteristics of streaming data.
- **Data processing**: It is necessary to process both data-at-rest and data-in-motion to provide the results. Data-at-rest includes both quantitative and qualitative data, currently stored in 42 different tables. The tables in the hot strip mill database store a total of 7475 variables related to the coil production process since 2010 with ~840000 records for each variable. It contains mostly numerical and categorical values and its size increase as new coils are produced. The size of each of the 42 tables is around 300-700 MB. Data-at-rest data comes from the system that measures flatness and from different sensors installed in the hot strip mill. It consists of a series of time series variables that are measured continuously along the coil length. It includes variables such as temperatures, flatness, etc. The interval in which each variable is measured varies depending of the system capabilities to acquire and store the data and the speed of the coil being processed, but in the less than 1 second. Thus, the analysis of this data requires online capabilities in order to obtain the results as soon as possible. Additionally, flatness maps are also generated once the entire coil is processed and measured by the flatness measurement system. These maps are unstructured data (images) that contains useful information about the flatness of the coil. One flatness map is available for each coil. This real-time data includes 29 numeric variables and images (maps) from the processed coils.

2 Predictive Analytics

This section overviews the current state of the analytics being used by ArcelorMittal in the Hot Strip Mill process, and defines a set of requirements related with the use of predictive analytics in that process.

2.1 Current state

ArcelorMittal collects a significant amount of data of the Hot Strip Mill (HSM) process for quality monitoring purposes. The information about the production parameters of each coil is gathered at each stage of the HSM. All information is analysed in order to obtain a global score about the quality of the coil. The main problem of the current approach is that the quality analysis is only run after the coil has been produced, with a delay varying from hours to weeks from the production date. Given this time difference, it is not possible to use the global score as an indicator for the operator in charge of the HSM to modify production parameters. Additionally, in case a problem is detected in the analysis stage, it is likely that a significant number of coils produced afterwards show similar problems.

2.2 Needs and gaps

In terms of predictive analytics considering the current state of the process, the following aspects may be addressed by ArcelorMittal to improve the HSM process.

1. **Streaming analytics.** The ability to obtain real time analytics will provide ArcelorMittal with actionable information about the HSM process while the coils are being produced. By removing the delay between data collection and analysis, it will be possible to inform the operator about the current quality allowing her to modify the HSM configuration if needed.
2. **Data reduction.** The current approach to quality assessment collects a significant amount of variables. This number increases the complexity of the analysis requiring more computing resources (e.g., CPU, memory, bandwidth, etc.), and increases the latency between data and actionable information. In order to address this problem while maintaining the quality of analysis, we propose to introduce a feature selection pre-processing stage, so only the relevant variables are considered in the analysis. Notice that this study can be run once in the project based on the historic dataset provided by ArcelorMittal.
3. **Context.** The global score is a value that depends on the time and the context (e.g., current coil, regions, etc.). The analysis of the streaming data should take into account this information in order to help detecting strange situations, time patterns or different global events.
4. **Detect relevant situations.** An operator of a complex process such as the HSM can be easily overwhelmed by the large amount of monitoring information being produced on top of the analytics results. The predictive analytics method must be able to alert the user when relevant situations appear (e.g., coil quality drop, unexpected measurements, or hidden problems).
5. **Prediction.** The system should be able to predict the global score of a coil based on real time and historical data abstracted in the machine learning model. As such, the pipeline requires a fast and low latency storage that allow access to the existing models.
6. **Autonomous.** The HSM process is complex and is subject of continuous evolution. For these reasons the selected predictive method must be able to adapt itself to the current configuration of the process maintaining the quality of the information being produced.
7. **Get the feedback.** The ArcelorMittal operators have an intensive knowledge of the HSM process and are able to asset the relevancies of different situations. The proposed algorithm should be able to include part of this knowledge by asking operators to provide feedback in relevant situations. The supervision layer on top of the algorithms will increase the accuracy of the models.

Considering the aforementioned aspects, the proposed solution should be able to provide near-real time analysis of the HSM process, discover unexpected situations, anomalies in the process, and existing patterns. The proposed machine learning pipeline should be parallelizable, scalable, and provide incremental processing of the data.

2.3 Existing solutions analysis

The current market offers some solutions to analyse streaming data using online machine learning techniques. The main problem of current solutions is that they are focused not on system operators, but on developers, data scientist and other expert profiles.

- Machine learning libraries for streaming processing frameworks: Most streaming processing frameworks have integrated an online machine learning library to build analytics solutions. For example, Storm [1] offers Samoa , Spark Streaming [2] offers MLlib [3], and Flink [4] offers FlinkML. These libraries contain general purpose machine learning algorithms and allow users to write their own using basic abstractions. This type of solutions is oriented to developers and requires a high technical level.
- Machine learning integrated services: Another possible solution is use machine learning services such as SQL stream [5], DataTorrent [6], and Treasure. The service analyses the stream data information but it remains a responsability of the final user to build the required machine learning pipeline for its particular problem. This kind of solution is oriented to data scientist as it removes the infrastructure layer from the problem.
- All-included services. There are complete solutions for specific problems that may be used to address the aforementioned requirements, e,g,. Adonot, Novelti, or Predix. These products are focused on single problems or domains and integrate the required machine learning stack. This type of solutions is dedicated for operators or system experts.

3 Visual Analytics

This section will analyse the functional scenario from the visual analytics point of view. The objective is to identify the functional and technical gaps that will be addressed in the scope of the project. Based on this discussion, Section 5.2 will list the software requirements for visual analytics in the PROTEUS project.

3.1 Current state

One of the steps in the steel manufacturing process is the rolling operation on the hot strip mill, where the steel labs are reheated and rolled to hot rolled coil. During this operation, a number of real-time sensors monitor every aspect of the process, generating extremely large and diverse data streams, both structured and unstructured. This data is mainly formed by parameters to measure thickness, width and flatness of the steel slabs, and it is useful to predict the result of the final product and improve the process in the future. Data produced by the sensors is stored in databases for later consumption, typically by SQL queries and applications connected via JDBC.

At the present time, there is no system in place to manage this extremely large streams of data in real-time, therefore, the data cannot be queried nor visualised in real time, only historically.

3.2 Needs and gaps

After analysing current state of technological client side environment and the processing used until now, it's easy to conclude that actual processing doesn't allow managing in real-time huge amount of data generated by Arcelor's sensors.

In order to improve this processing method and be able to analyse and visualize all the data generated in real time, Proteus project visualization attacks this new scenario that implements streaming process.

First of all, there are several points detected and explained to get a suitable visualization tool adapted to Proteus:

- Currently, relevant information must be discarded in order to obtain a system manageable analysis and reactive, in the way that information and stored data is available only after the production process has finished, denying from real time error detection and decision making. These facts prevent the metal process benefit from high valuable techniques, only available through tools that empower real time monitoring, real time decision making, error prediction, optimization processes and intuitive and interactive visualization.
- New visualization techniques will improve the actual analysis and decision making process, allowing real time response to executing tasks and enabling production behaviour prediction, based in historical and recently acquired data both. Tools implementing these techniques will allow operators to visualize and interpret different results and variables of the production process in an incremental and real time manner, so they can anticipate the value of multiple and diverse parameters that affect the final dimensional properties of the obtained product.
- Some of the parameters involved in the production process are metal thickness, width and flatness measurement, which determinates the quality of the product, in addition to temperature, vibration intensity, tension in the rollers and the speed of the plate of the machinery. These parameters are obtained in real time through a sensor network installed around the facility. Real time visualization and prediction based on these parameters are the main objective of the new tools. In order to achieve the desired objective, these new visualization tools have to deal with the most representative characteristics of Big Data :
- Variety of data: Data generated by the sensors are both quantitative and qualitative. The current process defines 34 different structured data schemas with a total of 7870 variables. This information is complemented in real-time with the data generated by an internal model that evaluates the flatness of the coil. This semi-structured data, stored in SIG format, contains data in two formats: time series and maps (similar to a heat map) that represent the flatness of the coil. Each SIG file stores 42 time series/maps associated with the coils, including their flatness as a target variable among others.

- Velocity of data: The generation rate of each variable varies from sensor to sensor but always in the millisecond scale. As a general guideline, the generation rate is in the range of 32 and 500 milliseconds.
- Volume of data: Historical data from 2010 to now of hot strip mill process includes ~700000 records for each one of the 7870 variables. In addition, the historical data of SIG information is around 500 gigabytes. Historical data has to be used in real-time simulation to enrich the predictive models in real-time. Volume of data streams is a key issue here due to the 7870 variables plus the time series and maps generated in the millisecond scale and to be processed with low latency for predicting the flatness.
- Discussed tools must provide an adequate and suitable kit for visualization, such as collections of charts that represent different kinds of data in a meaningful way, regarding the diagram in use and appropriate dashboards enabling to choose between the different charts and properties of variables. It is important to allow operators dive in the behaviour of the variables, therefore different variables must be highly comparable considering many facts such as periods of time and correlation factors.
- Different periods of time involve data at rest obtained from historical saved data and real time obtained data. It is also important to enable suitable prediction visualization of variable behaviour, using meaningful trend lines and other prediction techniques. These relationships must be expressed using the adequate visualization tool.
- The system will make use of machine learning techniques with the objective of build the predictive model and to ensure that the model evolves properly, operators will have the choice to analyse it and visually obtain information about it
- It is important to allow users to visualize quickly (real time) any changes in the production process in an intuitively way to streamline decision-making without going into considerable losses in final product development.
- A what-if analysis could be interesting to perform in order to predict possible changes in some of the variables involved in the process and anticipate possible losses or gains related to the production process.
- Business Intelligence users should be able to perform analytical data operations in order to improve the production process constantly and get better business profits.

3.3 Existing solutions analysis

3.3.1.1 Software tools

Currently there exist different visualization tools oriented towards the data analysis. Most of them are oriented to the data analysis from relational databases, but gradually visualization tools intended for use in Big Data environments are growing. Common used tools will be analysed in this section in order to take an approximation of available data analytic operations and data visualizations that provide each tool.

Tableau Software [7]

Family of interactive data visualization products focused on business intelligence and data analytics over graphical visualization. It's not an open source tool, and offers five main products in order to fit customer needs.

- Desktop
- Server
- Online
- Mobile
- Reader

Tableau last release has a Hadoop connector integrated in order to get data from HDFS.

Provides different analytics operations:

- Data cleaning
- Ad-hoc operations
- Advanced operations
- R integration
- Sorting operations
- Grouping operations
- Calculates groups operations
- Filtering operations
- Trend-Lines
- Cohort-Analysis
- What-If Analysis
- Sampling
- Timing Analisis
- Continous to discrete operations

Url	www.tableau.com
Open Source	No
Big Data Connector	Yes, Hadoop connector in order to retrieve data from HDFS
Main feature	Powerful analytics operations

CartoDB [8]

CartoDB is a SaaS cloud computing platform that provides GIS and web mapping tools for display in real time in a web browser. CartoDB users can use the company's free platform or deploy their own instance of the open source software. It was built on open source software including PostGis and PostgreSQL. The tool uses javascript extensively in the front end web application, back end Node.js based APIs, and for client libraries.

CartoDB is split into four components:

- The web application, where users can manage data and create custom maps. Users who aren't technically inclined can use an intuitive interface to easily create custom maps and visualizations. Advanced users can access a web interface to use SQL to manipulate data and apply map styles using a cartography language similar to css.
- Maps API that acts as a dynamic tile service, which creates new tiles based on client requests.
- SQL API, where PostgreSQL-supported SQL statements can be used to retrieve data from the database. The SQL API serves data in various formats including JSON, GeoJSON, and CSV.
- CartoDB.js library, which can wrap the Maps and SQL APIs into complete visualizations or be used to integrate data into other web applications.

Data types supported:

- CSV
- TSV
- SHP
- KML, KMZ (google Earth format)
- XLS, XLSX (Excel)

- GEOJSON
- GPX (Datos de gps)
- OSM,BZ2 (Open Street Maps)
- ODS (Open Spreadsheet)

Url	www.cartodb.com
Open Source	Yes
Big Data Connector	Yes
Main Feature	Real time data visualization in maps
Written in	Ruby, Javascript

Gephi [9]

Gephi is an open-source network analysis and visualization software package written in Java on the NetBeans platform, initially developed by students of the UTC in France.

Url	gephi.org
Open Source	Yes
Big Data Connector	Yes
Main Features	• Real time visualization • Layout Algorithms • Dynamic Filtering • Extensible with plugins • Metrics
Written in	Java, OpenGL
Operatin System	Linux, Windows, MAC OS

Kibana [10]

Kibana is an open source data visualization plugin for ElasticSearch. It provides visualization capabilities on top of the content indexed on an Elasticsearch cluster. Users can create bar, line and scatter plots, or pie charts and maps on top of large volumes of data.

Kibana uses JSON based query language, and allows the following data analytics operations:

- Count: computes the number of rows that matches a specified criterion.
- Average: computes the average of a specific dataset.
- Sum: computes the sum of specific dataset.
- Min: computes the minimum element of a specific dataset.
- Max: computes the maximum element of a specific dataset.
- Unique count: computes the number of unique values in a specified criterion.
- Percentiles: show the point at which a certain percentage of observed value occurs.
- Range: allow a field to have values between the lower and upper bounds.
- Date Range: allow a field to have values between the lower and upper data bounds
- Filters: Allow a field to have values that suits in specific user criteria.

- Standard Deviation: quantify the amount of variation or dispersion of a set of data values.

Kibana provides different visualization charts:

- Area Chart: A type of presentation graphic that emphasizes a change in values by filling in the portion of the graph beneath the line connecting various data points.
- Data Table: Standard spreadsheet table into chart that can easily be sorted and paged
- Line Charts: Data points connected with a straight line
- Pie Charts: Visualize data as "slices of pie," or proportions of a whole.
- Vertical Bar Chart: displays data as vertical bars.
- Dashboards: 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.

Url	www.elastic.co/products/kibana
Open Source	Yes
Operatin System	Multi platform
Written in	JavaScript
Main Features	Extended visualization charts and data analytics operations

Caravel

Caravel is a software tool for data analysis and visualization through charts called "slides". Caravel allows the creation of dashboards, slide collections with meta information such as size, position and CSS styles and export them as JSON or CSV. It provides many kinds of visualizations such as timeliness, heat maps, world maps, trend lines, tables and more.

Data sources are tables contained in different database systems like MySQL, Postgres, Presto or Redis, which are queried using Caravel's web interface, allowing us selecting, filtering and aggregating data interactively through the browser.

It is not possible to query multiple tables at a time, although Caravel use views to perform joins between different tables.

Caravel allows the following operations over data, depending on the kind of slide:

- Mean – Computes the average of a dataset.
- Sum – Computes the sum of a dataset.
- Min – Computes the minimum element of a dataset.
- Max – Computes the maximum element of a dataset.
- Stddev – Computes the standard deviation of a dataset.
- Var – Computes the variance of a dataset.
- Median – Computes the median of a dataset.

Caravel provides many kinds of charts, the most relevant are:

- Line charts – Data points connected with a straight line.
- Area charts – Can be visualized in stacked, stream or expanded form.
- Table – Standard spreadsheet table into chart that can easily be sorted and paged.
- Pie charts – Visualize data as "slices of pie," or proportions of a whole.

- Bars – A good way to visualize one or more categories of data, particularly if each category has sub-categories. Can be visualized grouped or stacked.
- Scatter plot – Chart that displays numeric coordinates along the X- and Y-axis.

Caravel is built upon Python and Flask App Builder. It depends on SQLAlchemy to access databases and D3 for chart drawing.

Url	https://github.com/airbnb/caravel
Open Source	Yes
Written in	Python, Flask App Builder
Main Features	• Rich set of visualizations to analyze data • Flexible way to extend the capabilities • An extensible, high granularity security model allowing intricate rules on who can access which features, and integration with major authentication providers.

R

R is a system for statistical computation and graphics. It consists of a language plus a run-time environment with graphics, a debugger, access to certain system functions, and the ability to run programs stored in script files. It is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis
- graphical facilities for data analysis and display either on-screen or on hardcopy
- a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities,
- easy installable packages to extend the core.

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering,) and graphical techniques, and is highly extensible.

Url	http://www.r-project.org
Operating System	Multi platform
Open Source	yes

SAS Visual Analytics

Visualization tool developed by SAS. It's commonly used to see a graphical data report from many different sources (as datawarehouse, rdbm). Also it has functions of data analytics.

In SAS documentation:

- Big Data environments has been added as new data source, but only Hadoop environment suits in that new feature. SAS VA Tool can only read any data stored as SASHDAT (SAS datatype) in HDFS filesystem.
- SAS VA can load structured data only (CSV, Spreadsheets..etc)
- Considering this, it has been concluded that is not a valid tool for applying in the project

Url	http://www.sas.com/en_ca/software/business-intelligence/visual-analytics.html
Open Source	No
Operating System	Windows, IBM mainframe, Unix/Linux, OpenVMS Alpha
Written in	C
Big Data Connector	Yes, but only to HDFS service and only can read SASHDAT data

Conclusion

Most of data visualization tools have big data connectors, but only for Hadoop environments so they are not extrapolated to environments intended for real-time analysis (such Flink environment).

In conclusion, the PROTEUS project should have its own visualization tool and analytical data operations.

3.3.1.2 Software libraries

Data visualization is the presentation of data in a pictorial or graphical format. It enables decision makers to see data presented visually, so they can understand difficult concepts or identify new patterns.

Static visualizations can offer only precomposed "views" of data, so multiple static views are often needed to present a variety of perspectives on the same information. The number of dimensions of data is limited, too, when all visual elements must be present on the same surface at the same time. Representing multidimensional datasets fairly in static images is notoriously difficult. Due to this static images are commonly represented in formats like PNG or JPG user interaction, visual effects and pattern identifying are not possible.

The following sections describes a set of technologies and libraries that enables web browsers to visualize data in an interactive and scalable way.

3.3.1.2.1 Web Graphic Technologies

HTML5 is revision of the Hypertext Markup Language (HTML) standard, the standard programming language for describing content and appearance of websites. HTML5 includes some new features such as syntax clean-ups, new JavaScript language features and APIs, mobile capabilities, and breakthrough multimedia support.

Using HTML5, developers and designers can create graphic experiences by using standards-based technology. Although graphics on the web is not a new concept, HTML5 greatly enhances the user experience by eliminating the installation of plug-ins, the installation of which has been linked to 50% site abandonment, since graphics are currently delivered natively by the browser.

This section describes a set of graphic technologies that are currently supported by the most of the web browsers: SVG, Canvas and WebGL:

SVG
Scalable Vector Graphics (SVG) is a language for describing two-dimensional vector and mixed vector/raster graphics in an XML format. It has two parts: an XML-based file format and a programming API for graphical applications. Key features include shapes, text and embedded raster graphics, with many different painting styles. It supports scripting through languages such as ECMAScript and has comprehensive support for animation. SVG is used in many business areas including web graphics, animation, user interfaces and high-quality design. SVG builds upon many other successful standards such as XML (SVG graphics are text-based and thus easy to create), JPEG and PNG for image formats, DOM for scripting and interactivity, SMIL for animation and CSS for styling. SVG is interoperable. It allows three types of graphic objects: vector graphic shapes, images and texts. They can be grouped, styled, or transformed. SVG can be interactive and dynamic.

Canvas
Canvas is a part of HTML5 that allows dynamic, scriptable rendering of 2D shapes and bitmap images. It is a low level, procedural model that updates a bitmap and does not have a built-in scene graph. It is the typical choice for most HTML5 games.

Canvas consists of a drawable region defined in HTML code with height and width attributes. JavaScript code will access the area through a 2D API, which contains a full set of drawing functions. All the content drawn by Canvas API is rendered as a bitmap, in single HTML5 DOM elements. It has a set of limitations: (i) Event handling is very difficult, because of drawn objects become part of the canvas bitmap (all the elements drawn in the bitmap are treated as a single element); (ii) Moving an object across the canvas requires to redraw the entire canvas for each movement, since it is entirely treated as a bitmap.

WebGL
Web graphics library (WebGL) is an image library that allows creating real-time rendered and interactive 3D graphics. It is also a JavaScript API that allows graphics rendering within any compatible web browser without the use of plug-ins. It is integrated completely into all the web standards of the browser, allowing GPU accelerated usage of physics and image processing and effects as part of the web page canvas. Despite of WebGL is widely supported in modern browsers, its availability depends on GPU (graphics processing unit) supporting it. GPU is a specialized electronic circuit and a single-chip processor designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer. It was presented as a processing unit with rendering engines that are capable of processing a minimum of 10 million polygons per second. Therefore, the main difference between WebGL and other rendering technologies is the intensive use of GPU that WebGL does, by allowing web browsers to rapidly render 2D and 3D graphics.

WebGL is available both in computers and mobile devices. The problem of mobile devices is that GPU processor is very limited, so that FPS (frames per seconds) offered by this devices is smaller compared to traditional computers.

3.3.1.2.2 Web Visualization libraries

We have classified these libraries by its purpose into three different categories: 3D, drawing and charting libraries.3D and drawing libraries allow general purpose rendering of 3D and 2D graphics respectively, while charting libraries are specifically designed for charting only.

3D libraries
These libraries allow the rendering of 3D graphics using mainly WebGL technologies. Apart from the libraries described in this section, there are other libraries that can render in-browser 3D graphics. The former are not covered in this document since they rely on proprietary technologies.

- **BabylonJS:** JavaScript framework mainly for building 3D videogames in HTML5. It can be used to render any kind of 3D graphics in WebGL, including charts.
- **Three.js:** JavaScript 3D library that is meant to be easy to use. It can render to SVG, Canvas, WebGL and CSS.
- **PlayCanvas:** Open source 3D engine for the development of 3D games and interactive applications. It renders WebGL.

Drawing libraries

The libraries described in this section allow manipulation of SVG or Canvas elements for 2D drawing. Some of them can be used to draw any kind of visualisation or graphics on the web, while others have been specifically designed for data visualisation.

- **D3** [11]**:** Widely used JavaScript library specifically designed for building dynamic data visualisations for the web. It renders to SVG and supports direct DOM elements manipulation and data binding to SVG elements, allowing the creation of any kind of elaborated and interactive visualisations.
- **Fabric.js:** JavaScript library that provides an object model for Canvas, to aid the creation of objects and simplify the use of its API. This library eases the use of the complex Canvas API while retaining its features.
- **Phaser:** Framework for the development of HTML5 games that renders to WebGL and Canvas. It can also be used to create interactive charts.
- **Two.js:** 2D drawing library that can render to WebGL, Canvas and SVG using the same API. It requires Underscore.js and Backbone.js events.

Charting libraries

These libraries can plot various kinds of charts that are rendered with one of the three technologies discussed earlier in this document. Some can only support one specific type of charts while others provide a wide catalog of charts or customisation options.

- **amCharts:** Standalone charting library that offers a variety of customizable charts for different kinds of data, including geographic data (SVG maps). It provides responsive design in all of the charts and renders to SVG. Its charts can load data as JSON, CSV or JavaScript arrays, it also supports real-time data.
- **ArborJS:** Simple graph visualisation library that renders to SVG and Canvas. It has the ability to render graphs exclusively.
- **ChartJS:** Lightweight and standalone charting library. ChartJS provides different interactive and responsive chart that can be customized and rendered to SVG. It does not provide a mechanism for streaming data.
- **CytoscapeJS:** Library for visualisation and analysis of networks. It is dependency free and renders to Canvas.
- **Dygraphs:** Fast JavaScript charting library that provides interactive charts. It can handle big data sets with millions of points and supports error bars. The charts are customizable, allow for touch controls in mobile devices and support streaming data.
- **EJSCharts:** JavaScript charting library that renders interactive charts in Canvas. It can read data from JSON, CSV, XML and JavaScript arrays, and provides support for streaming data loaded with Ajax.
- **EnvisionJS:** JavaScript library for creating fast and interactive visualizations. It supports streaming data and renders to SVG.
- **EpochJS:** Charting library written in JavaScript that renders to SVG and Canvas. It supports streaming data but it only supports six basic charts.
- **Highcharts:** JavaScript charting library that offers interactive charts and supports a wide variety of interactive chart types. It renders to SVG and Canvas and supports exporting to several formats and printing. HightCharts its not open source, although it is free for non-commercial uses.
- **KartoGraph.js:** Lightweight JavaScript library for building interactive maps without the need of a mapping service. It renders SVG maps and runs standalone.
- **Leaflet:** Lightweight JavaScript library for creating mobile-friendly interactive maps. It is intended to provide good performance and usability and it can be extended with plugins. It supports tile maps, WMS, vector layers, image overlays, and interactive features.
- **NVD3:** Charting library written in JavaScript that intends to provide reusable and customizable d3.js modules. While it depends on d3.js and renders to SVG it lacks support for streaming data.
- **Plotly.js:** Built on top of d3.js and stack.gl, plotly.js is a high-level, declarative charting library. It ships with 20 chart types, including 3D charts, statistical graphs, and SVG maps. Plotly provides a streaming API for real-time data.
- **OpenLayers:** JavaScript library to load, display and render maps from multiple sources. It can render tiled layers and vector layers and it provides mobile support. The maps can be rendered in WebGL and Canvas, and it allows fully customised maps.
- **Polymaps:** JavaScript library for image and vector tiled maps that render to SVG. Its goal is to provide large-scale data overlays on interactive tiled maps while retaining performance.
- **RickShaw :** JavaScript toolkit for creating interactive graphs that support streaming data. It depends on d3.js and renders to SVG.
- **ZingChart:** Charting library with a great number of chart types and customization options. It provides responsive design and supports streaming data with thousands of data points. ZingChart is not Open Source, although it is free to use with certain restrictions.

3.3.1.2.3 Analysis

HTML5 Graphic Technologies

In order to choose the right HTML5 graphic technology, we need to analyse the features they offer, and when technology becomes ideal depending on the scenario. Although some common features like styling or 2D geometric model are supported by all the technologies, there are other complex features (such as interactive or 3D model) that, depending on the rendering style, are supported or not.

The following table summarizes the main feature differences between SVG, Canvas and WebGL graphic technologies:

Table 1 Comparison of HTML5 graphic technologies

	SVG	**Canvas**	**WebGL**
Interactive (natively)	**X**		
Multiple DOM elements	**X**		
Native 2D support	**X**	**X**	**X**
Native 3D support			**X**
Pixel Manipulation		**X**	**X**
Scalable web graphics	**X**		
Styling	**X**	**X**	**X**

Libraries

Table 1 shows a comparison of technologies and technical characteristics of each charting library, as well as chart types available in each of them.

Conclusion

In this section we have studied the most common charting libraries available to find one that meets all the needs of the project. This requirements are: ability to load real-time streaming data, interactive visualisations, support for massive volume of data points, suitability for scientific data analytics and machine learning, usability and enough extensibility to make custom visualisations.

While some of the studied libraries meet one or more of the requirements, there is none that meets all of them at the same time.

We therefore conclude that there is no library available that meets all of the requirements to suitably perform the tasks needed in PROTEUS, thus the need of a new data visualisation library specifically designed for predictive analytics and real-time interactions with extremely large datasets.

The new charting library will be built on top of D3, since it is already the state of the art library for data visualisation on the web. One potential constraint of D3 is that it renders to SVG, a technology that limits the number of elements displayed on the browser, contrary to Canvas and WebGL technologies. However, this limitation can be overcome with various techniques of data aggregation and incremental methods, and there is no available library that matches the capabilities of D3 and renders to Canvas or WebGL at the same time.

	Render technology			Size	Basic charts					Statistical		Temporal		Hirarchical					Flow			Network	Spatial		Matrix	
	SVG	Canvas	WebGL	Aprox. Size (KB)	Bars	Columns	Lines	Pie	Area	Scatter plot	Box plot	Timeline	Gant	Tree map	Dendrogram	Wind rose	Polar	Organizational	Alluvial diagram	Sankey diagram	Swinlane	Graphs	SVG Maps	WMS / Tile Maps	Heat map	Chord diagram
amCharts	X			213	X	X	X	X	X	X			X				X						X		X	
ArborJS	X	X		1																		X				
ChartJS		X		51		X	X	X	X								X									
Cytoscape	X	X		263																		X				
Dygraphs	X			123	X		X																			
EJSCharts	X			194	X	X	X	X	X	X																
EnvisionJS		X		104			X																			
EpochJS	X			69		X	X	X	X	X															X	
Highcharts	X	X		165	X	X	X	X	X	X	X	X		X		X	X						X		X	
KartoGraphs	X			66																			X			
Leaflet	X			131																				X		
NVD3	X			211	X	X	X	X	X	X																
Plot.ly	X		X		X	X	X	X	X	X	X			X			X						X		X	
OpenLayers	X			509																				X		
Polymaps	X	X		34																			X	X		
RickShaw	X			75		X	X		X																	
ZingChart		X		633	X	X	X	X	X	X	X	X		X		X	X						X		X	X

Table 2. Visualization libraries comparision

4 Data processing

This section analyse the industrial case in terms of data processing requirements.

4.1 Current state

Currently ArcelorMittal processes data with the system which is installed in other facility than the production one. To deal with the defect detection, a measurement system is developed that assigns a punctuation on the coil depending on the flatness defects found. The score is assigned as a result of the measurement to analyse the overall quality of coil. The current data processing system is not operating on real time basis and there is no upper bound guarantee about the computation results.

4.2 Needs and gaps

In terms of data processing, ArcelorMittal should address several needs to be solved:

- Processing data with low latency. Processing the sensor data with low latency is essential to detect the defects in coil production. As ArcelorMittal indicated, the defect inspection of the existing system can take several days. This can cause unforeseen results of the production. Therefore, high latency is a serious issue that needs to be addressed.
- Providing upper bound on the computation results. This is important as the system should provide the computation result with a worst case guarantee.
- Fault tolerance and high availability. If the sensor data is massive, we need to provide a distributed system to process the data and therefore, availability of such a system becomes essential. For example, if any computation or data processing unit goes down for any reason, the system should continue to operate and detect the defects.
- Flexible data processing system that can be configured with high level APIs. For example, if the production process changes or the computational model of a global score changes in the future, the amount of changes in the data processing system should be minimal and it should be done through high level APIs.
- Unified batch-stream data processing. Another need for the data processing system in ArcelorMittal is supporting the hybrid (batch and stream) computation. Especially in defect or anomaly detection systems the hybrid computation should be used. For example, the global score can be assigned more "confidently" if the real time data is compared with historical data and the result is used as a parameter when computing global score.
- Scalable data processing. When the data obtained from sensors are massive, the computational time should be competitively same (and not exponentially increasing) provided that we increase the number of computational units.
- Machine learning capability of the system. Because, the processing of coil is a repeated process, detecting defects could be easy if the system "learns" from its previous actions. Here the presence of machine learning algorithms in the system is relevant. This is a need in current data processing system. The goal is to train a classifier which will predict whether a mixer has high chances of being defective, based on the given measurements. For this task, we have to gather data from various log files residing in different production plants and join them in order to get all measurements of a mixer throughout its production.
- Declarative language. The absence of a high-level declarative language for interfacing/interacting with the kernel of the current data processing system, especially for machine learning is can be considered as a deficiency. Machine learning analysts need not to be database or system expert necessarily. For

example, SQL users need not to know about the internals and the structure of the processing details making SQL even more popular; hence the importance of equipping the current system with a declarative language. Most of the languages proposed on top of parallel dataflow engines are based on dataflow abstractions where the user specifies his program in terms of transformations (e.g., joins or maps) of bags of data. The need for highly declarative languages led to various SQL-like abstractions on top of these engines, like Apache Hive [12], Spark SQL and Impala [13]. While these languages are also capable of using user-defined functions (UDFs), they are foremost designed for Select-Project-Join (SPJ) queries and are not suitable for more complex analysis tasks found in machine learning. This stems from the inability of these languages to encode iterations and control flow, as well as the inability to optimize complex algorithms that go beyond simple data flows.

4.3 Existing solutions analysis

Currently there are several partial solutions to address the requirements highlighted in the previous sections. In this section we explain the key solutions from each category and analyse if they can fulfil those requirements.

The concept of combining batch and stream processing has been initially proposed by Nathan Marz as Lambda Architecture [14]. It is a fault-tolerant robust system and able to serve a wide range of workloads. For example, Apache Hadoop and Hive are good examples of batch layer of the Lambda architecture and Apache Storm as a low latency layer for faster reads and writes. However, there are several drawbacks of this architecture.

- Maintenability: the Lambda architecture consists of several systems which can be painful to maintain. Moreover, separate code has to be written and maintained for Summingbird [15] which was developed in Twitter, is an example of such an architecture. It has high level APIs which make developers' life easy. Another approach was Lambdoop [16]. However, as it is mentioned above, maintaining and running several systems synchronously comes with a cost.
- Lack of a unified programming interface. It is hard to program such a system as it lacks a common programming interface. That is, the architecture is general and gives freedom to use arbitrary frameworks as its components. Moreover, the Lambda architecture has not provisioned for machine learning models that need to be continuously updated. It should, thus, be extended in order to fit the setting and requirements in our use case.

Creating a system which addresses most of the above problems with relational database technologies was the research conducted in Brown University and MIT. They developed S-Store that treats the streaming data as transactional events to a database. However, this system faces scalability issues as it focuses on OLTP, rather than OLAP and complex analytics.

There are several streaming and batch data processing systems like Apache Flink, Apache Spark and etc. Those systems support streaming and batch processing in their engine. However, they either do not combine batch and stream processing in one execution engine or they have very high latency.

The Berkeley Data Analytics Stack (BDAS) is a framework that can serve a basis for the needs of hybrid computation. That is, BDAS combines batch and stream processing under a common dataflow API based on resilient data bags. Spark Streaming, standard batch processing in Spark and Spark SQL are the main components of this system. One drawback of Spark, especially in our context related with machine learning is that it does not provide a direct access to mutate state in asynchronous manner. Another drawback is that Spark has high latency when executing streaming queries. The reason is that it makes streaming computations in micro-batch fashion which can be a limiting factor when executing complex queries. Therefore, Apache Spark and BDAS cannot serve as a solution to the needs mentioned in above section.

While relational domain-specific languages (DSLs) such as Pig, Hive, or Spark SQL/DataFrame are a good fit for ETL (extract transform load) tasks, programming machine learning algorithms in those languages is cumbersome. To this end, R-like DSLs such as SystemML's DML or Apache Mahout's Samsara [17] were proposed. These DSLs offer linear algebra and control flow primitives suitable for expressing ML algorithms, but only provide limited, non-intuitive support for classic ETL tasks. This strict separation of programming paradigms introduces fundamental problems. To overcome these problems, we argue for the unification of relational and linear algebra into a common theoretical foundation. To achieve this goal, first we need to explore and reason about optimizations across the two algebras in a suitable intermediate language

representation (IR). Second, we need to showcase the added benefits of unification and the optimizations that come thereof, by defining a common DSL with high-level programming abstractions for both relational and linear algebra. In line of the benefits offered by other UDF-heavy dataflow API's, the proposed DSL should be embedded in a host-language like Scala (e.g. Spark RDDs, Samsara) rather than external (e.g., Pig, DML).

5 Catalogue of software requirements

5.1 Subsystem Predictive Analytics

5.1.1 Functional requirements

Predictive Analytics	
ID: FR-1.1	Batch processing algorithms
Version	0.0.1 2016-04-28
Subsystem	Predictive algorithm framework
Dependencies	
Description	The system should allow to launch machine learning algorithms using the batch processing framework.
Priority	HIGH
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.2	Stream processing algorithms
Version	0.0.1 2016-04-28
Subsystem	Predictive algorithm framework
Dependencies	
Description	The system should allow to launch machine learning algorithms using the stream processing framework.
Priority	HIGH
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.3	Hybrid processing algorithms
Version	0.0.1 2016-04-28
Subsystem	Predictive algorithm framework
Dependencies	FR-1.2, FR-1.1
Description	The system should allow to launch machine learning algorithms using a hybrid processing framework.
Priority	HIGH
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.4	Support to Flink
Version	0.0.1 2016-04-28
Subsystem	Predictive algorithm framework
Dependencies	FR-1.1, FR-1.2, FR-1.3
Description	The predictive algorithm framework should be compatible with Apache Flink.
Priority	MODERATE
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.5	Behaviour analysis
Version	0.0.1 2016-04-28
Subsystem	Predictive algorithm library
Dependencies	FR-1.1, FR-1.2, FR-1.3
Description	The predictive analytics should have algorithm to detect behaviour analysis.
Priority	MODERATE
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.6	Forecasting
Version	0.0.1 2016-04-28
Subsystem	Predictive algorithm library
Dependencies	FR-1.1, FR-1.2, FR-1.3
Description	The predictive analytics should have capabilities to forecast variables.
Priority	MODERATE
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.7	Anomaly detection
Version	0.0.1 2016-04-28
Subsystem	Predictive algorithm library
Dependencies	FR-1.1, FR-1.2, FR-1.3
Description	The predictive analytic framework should have capabilities to detect anomalies in different signals.
Priority	MODERATE
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.8	Automatic parameters calibration
Version	0.0.1 2016-04-28
Subsystem	Predictive algorithm library
Dependencies	
Description	The system should be able to adjust the algorithm parameters automatically adapting to the signal distribution.
Priority	MODERATE
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.9	Automatic analysis of the algorithm quality
Version	0.0.1 2016-04-28
Subsystem	Algorithms quality
Dependencies	
Description	The system should be measures to analysis of the quality of each algorithm using unsupervised techniques.
Priority	MODERATE
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.10	System to create experiments
Version	0.0.1 2016-04-28
Subsystem	Algorithms quality
Dependencies	FR-1.9
Description	The system should allow create different experiments and validate de quality for this experiments with different configurations.
Priority	LOW
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.11	Real time metrics
Version	0.0.1 2016-04-28
Subsystem	Algorithms quality
Dependencies	FR-1.9
Description	The system should define a framework to create real time metrics that get information about state and results of the algorithm.
Priority	LOW
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.12	Algorithm monitoring
Version	0.0.1 2016-04-28
Subsystem	Algorithms quality
Dependencies	FR-1.9
Description	The system should monitor the defined parameters for each configured algorithm in the system.
Priority	LOW
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.13	Parallels pipelines
Version	0.0.1 2016-04-28
Subsystem	Algorithms quality
Dependencies	FR-1.9
Description	The system should launch parallels pipelines to identify problems or improvements between different parameter configuration or algorithm implementations.
Priority	LOW
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.14	Split datasets
Version	0.0.1 2016-04-28
Subsystem	Algorithms quality
Dependencies	FR-1.13
Description	The system should provide tools to split the datasets or data streams in different parts to analyse and compare the results.
Priority	LOW
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.15	Evaluation datasets
Version	0.0.1 2016-04-28
Subsystem	Algorithms quality
Dependencies	FR-1.9
Description	The system should include analysed datasets to can evaluate the obtained results and compare this information with objective data.
Priority	LOW
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.16	Replay datasets
Version	0.0.1 2016-04-28
Subsystem	Algorithms quality
Dependencies	FR-1.15
Description	The system should be able to replay datasets with the same conditions.
Priority	LOW
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.17	Replay streaming datasets
Version	0.0.1 2016-04-28
Subsystem	Algorithms quality
Dependencies	FR-1.16
Description	The system should reproduce stored datasets like a data stream, with this functionality the system could test stream processing algorithms.
Priority	LOW
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.18	Support with continuous integration tools
Version	0.0.1 2016-04-28
Subsystem	Algorithms quality
Dependencies	FR-1.9
Description	The system should support to integrate with continuous integration tools like Jenkins or CircleCI.
Priority	LOW
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.19	Support other frameworks
Version	0.0.1 2016-04-28
Subsystem	Predictive algorithm library
Dependencies	FR-1.4
Description	The system should support to integrate with other processing frameworks.
Priority	LOW
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.20	Data reduction
Version	0.0.1 2016-04-28
Subsystem	Predictive algorithm library
Dependencies	
Description	The system should include technique for the feature selection.
Priority	LOW
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.21	Feature selection pipelines
Version	0.0.1 2016-04-28
Subsystem	Predictive algorithm library
Dependencies	FR-1.20, FR-1.13
Description	The system should evaluate different feature selection algorithm and check the result in independent pipelines.
Priority	HIGH
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.22	Online feature selection
Version	0.0.1 2016-04-28
Subsystem	Predictive algorithm library
Dependencies	FR-1.20
Description	The system should include techniques the online feature selection.
Priority	MODERATE
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.23	Historical data re-processing
Version	0.0.1 2016-04-28
Subsystem	Predictive algorithm library
Dependencies	
Description	When the algorithm parameters change the system should recalculate the historical results to adapt this information to the new environment.
Priority	LOW
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.25	Cache to store models
Version	0.0.1 2016-04-28
Subsystem	Distributed cache
Dependencies	
Description	The system should include a cache to store the model of the different algorithms.
Priority	HIGH
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.26	Persistent cache
Version	0.0.1 2016-04-28
Subsystem	Distributed cache
Dependencies	FR-1.25
Description	The cache should store the information in a data store.
Priority	HIGH
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.27	Distributed cache
Version	0.0.1 2016-04-28
Subsystem	Distributed cache
Dependencies	FR-1.25
Description	The cache should be distrusted.
Priority	HIGH
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.28	Client based cache
Version	0.0.1 2016-04-28
Subsystem	Distributed cache
Dependencies	FR-1.25
Description	The cache should store part of the information in the client memory.
Priority	HIGH
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.29	Cache with fault tolerance
Version	0.0.1 2016-04-28
Subsystem	Distributed cache
Dependencies	FR-1.25
Description	The cache should be able to works when some nodes are down.
Priority	HIGH
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.30	Invalidation of caches
Version	0.0.1 2016-04-28
Subsystem	Distributed cache
Dependencies	FR-1.25
Description	The cache should include a method to invalidate caches.
Priority	HIGH
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.31	Feedback method
Version	0.0.1 2016-04-28
Subsystem	Feedback
Dependencies	
Description	The system should provide a method to get feedback about the algorithm results.
Priority	HIGH
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.32	Feedback analysis
Version	0.0.1 2016-04-28
Subsystem	Feedback
Dependencies	FR-1.31
Description	The system should analysed with different techniques and include in the QA cycle.
Priority	HIGH
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.33	Feedback adaptation
Version	0.0.1 2016-04-28
Subsystem	Feedback
Dependencies	FR-1.31
Description	The system should include a framework to adapt semi-supervised algorithms to the captured feedback.
Priority	HIGH
Status	UNCOVERED
Comments	

Predictive Analytics	
ID: FR-1.34	Adaptivity
Version	0.0.1 2017-07-20

Subsystem	Feedback
Dependencies	FR-1.34
Description	The system, models and algorithms should be able to adapt dynamically to new conditions, evolving on time.
Priority	HIGH
Status	UNCOVERED
Comments	

5.1.2 Non-functional requirements

- **Compatibility**

Predictive analytics	
ID: NFR-1.1.1	Flink ecosystem compatibility
Version	0.0.1 2016-04-28
Subsystem	Predictive Framework
Dependencies	
Description	The framework should be compatible with the Flink ecosystem.
Priority	HIGH
Status	UNCOVERED
Comments	

Predictive analytics	
ID: NFR-1.1.2	Flink ecosystem compatibility
Version	0.0.1 2016-04-28

Subsystem	Distributed cache
Dependencies	
Description	The cache should be compatible with the Flink ecosystem.
Priority	HIGH
Status	UNCOVERED
Comments	

Predictive analytics	
ID: NFR-1.1.3	Scala compatibility
Version	0.0.1 2016/04/28
Subsystem	Distributed cache
Dependencies	
Description	The distributed cache must be implemented and compatible with Scala 2.10.*.
Priority	HIGH
Status	UNCOVERED
Comments	

Predictive analytics	
ID: NFR-1.1.4	Java compatibility
Version	0.0.1 2016/04/28
Subsystem	Distributed cache
Dependencies	
Description	Distributed cache must be implemented and compatible with Java 1.8+
Priority	HIGH
Status	UNCOVERED
Comments	

- **Reliability**

Predictive analytics	
ID: NFR-1.2.1	Testability and continuous integration
Version	0.0.1 2016-04-28
Subsystem	All
Dependencies	
Description	The software functionality and quality should be continuously tested and monitored with automated unit tests to prevent regressions and defects
Priority	MODERATE
Status	UNCOVERED
Comments	

- **Others**

Predictive analytics	
ID: NFR-1.3.1	Documentation
Version	0.0.1 2016-04-28
Subsystem	All
Dependencies	All software will be documented using the standards methodologies
Description	
Priority	LOW
Status	UNCOVERED
Comments	

Predictive analytics	
ID: NFR-1.3.2	Open-source
Version	0.0.1 2016-04-28
Subsystem	Distributed Cache
Dependencies	
Description	The source code of the distributed cache must be available as Open Source
Priority	HIGH
Status	UNCOVERED
Comments	

5.2 Subsystem Visual Analytics

5.2.1 Functional requirements

Visual Analytics	
ID: FR-2.1	Interactivity
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	
Description	The system should allow the definition of events on user interaction
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.2	Real-time visualization
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	FR-2.1
Description	Ability to visualize real-time data coming from external sources
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.3	Visualisation pausing
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	FR-2.1
Description	Ability to pause and resume the real-time visualisations on demand by the user
Priority	MODERATE
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.4	Ability to configure visualizations
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	FR-2.1
Description	Options such as sizes, events and chart styles must be user-configurable
Priority	MODERATE
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.5	Flink support
Version	*0.0.1 2016-04-28*
Subsystem	Visualisation library
Dependencies	
Description	The charts must be able to display visualisations of incremental operations carried out by Apache Flink.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.6	Static data visualisation
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	FR-2.1
Description	Ability to visualise data-at-rest
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.7	Color palettes
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	
Description	The user must be able to specify the color palette to be applied to the charts.
Priority	HIGH, MODERATE, LOW
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.8	Export to image format
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	
Description	Library must allow to export svg graphs into PNG format
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.9	Graph snapshots
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	
Description	The user should be able to select an area in the chart and take snapshots of the visualisation for later comparison with other areas.
Priority	LOW
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.10	Chart categories
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	
Description	The visualisation should include charts from the following categories: basic, statistics, temporal, spatial, matrix, flow, hierarchical, networks.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.11	Touch support
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	
Description	The user should be able to interact with the visualisations using a touchscreen.
Priority	MODERATE
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.12	Sum Operation
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental low level operation that allows another high level operations.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.13	Substract Operation
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.14	Multiply operation
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.15	Division operation
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.16	Split operation
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Operation that allows to split string chains in smallest chains.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.17	Contains Operation
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	String operation that allows search a pattern inside string chain
Priority	MODERATE
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.18	Lenght
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental low level operation that returns the string length of specific input.
Priority	MODERATE
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.19	Count operation
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental low level operation that computes the how often appears a specifi element.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.20	Average operation
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental low level operation that computes the incremental average of a data stream and allows another high level operations.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.21	Median operation
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental low level operation that computes the median of a data stream and allows another high level operations.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.22	Mode operation
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental low level operation that computes the mode of a data stream and allows another high level operations.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.23	Variance operation
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental low level operation that computes the variance of a data stream and allows another high level operations.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.24	Covariance operation
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental low level operation that computes the covariance of a data stream and allows another high level operations.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.25	Standard deviation
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental low level operation that computes the standard deviation of a data stream and allows another high level operations.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.26	Correlation
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental low level operation that computes the correlation between two variables of a data stream and allows another high level operations.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.27	Percentage operation
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental low level operation that computes the percentage of different values over a total in a data stream and allows another high level operations.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.28	Max operation
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental low level operation that computes the max value of a data stream and allows another high level operations.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.29	Min operation
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental low level operation that computes the min value of a data stream and allows another high level operations.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.30	Group by
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental low level operation that allows the grouping of values given a key. Operators will be applied to values of a key.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.31	Order by
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental low level operation that orders values of a data stream and allows another high level operations.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.32	Set default value for specific input value
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Operation that detects invalid domain values and allows its substitution.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.33	Set continuous variable
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Treat data stream values as continuous variables.
Priority	MODERATE
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.34	Set discrete variable
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Treat data stream values as discrete variables.
Priority	MODERATE
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.35	DateDiff
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental low level operation that returns the difference between dates
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.36	DateGet
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Returns the month,year,day or time of specific date
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.37	DateAdd
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental low level operation that returns the result given by addition operation, or substract operation.
Priority	MODERATE
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.38	DateDiff
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental low level operation that returns the difference between dates
Priority	MODERATE
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.39	Time period selection
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Operators will allow to specify a time period to select the values that will part of the input of the operator.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.40	Operator recovery
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Operators will allow state recovery of data in case of failure.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual Analytics	
ID: FR-2.41	Adaptivity
Version	0.0.1 2017-07-20
Subsystem	Incremental Analytics Engine
Dependencies	
Description	The visual analytics system should be able to adapt dynamically to new conditions, evolving on time when required.
Priority	HIGH
Status	UNCOVERED
Comments	

5.2.2 Non-functional requirements

- **Compatibility**

Visual analytics	
ID: NFR-2.1.1	HTML5 compatibility
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	
Description	The visualisation library must be compatible with browsers that support the HTML5 standard
Priority	HIGH
Status	UNCOVERED
Comments	

Visual analytics	
ID: NFR-2.1.2	CSS3 compatibility
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	
Description	The visualisation library must be compatible with browsers that support the HTML5 standard
Priority	HIGH
Status	UNCOVERED
Comments	

Visual analytics	
ID: NFR-2.1.3	ES6 compatibility
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	
Description	The visualisation library must be compatible with the ECMAScript 6 language
Priority	MODERATE
Status	UNCOVERED
Comments	

Visual analytics	
ID: NFR-2.1.4	Websocket support
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	
Description	The visualisation library must be able to consume data from a WebSocket API
Priority	HIGH
Status	UNCOVERED
Comments	

Visual analytics	
ID: NFR-2.1.5	Flink compatibility
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental operations used in backend must be build using Flink 1.0.*.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual analytics	
ID: NFR-2.1.6	Scala compatibility
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental operations must be implemented and compatible with Scala 2.10.*.
Priority	HIGH
Status	UNCOVERED
Comments	

Visual analytics	
ID: NFR-2.1.7	Java compatibility
Version	0.0.1 2016/04/28
Subsystem	Incremental Analytics Engine
Dependencies	
Description	Incremental operations must be implemented and compatible with Java 1.8+
Priority	HIGH
Status	UNCOVERED
Comments	

- **Reliability**

Visual analytics	
ID: NFR-2.2.1	Testability and continuous integration
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	
Description	The software funcionality and quality should be continuosly tested and monitored with automated unit tests to prevent regressions and defects
Priority	MODERATE
Status	UNCOVERED
Comments	

Visual analytics	
ID: NFR-2.2.2	Responsive web design
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	NF-2.1.1, NF-2.1.2, NF-1.11
Description	The layout of the User Interface should be able to adapt to different screen sizes and respond optimally with any device that support Dynamic HTML technologies.
Priority	MODERATE
Status	UNCOVERED
Comments	

Visual analytics	
ID: NFR-2.2.3	Support for future chart extensions
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	
Description	The system should provide a modular and extensible design and take into consideration possible future extensions of the chart catalog
Priority	MODERATE
Status	UNCOVERED
Comments	

- **Portability**

Visual analytics	
ID: NFR-2.3.1	Reusability
Version	0.0.1 2016-04-28
Subsystem	
Dependencies	
Description	Visualisation library
Priority	HIGH
Status	UNCOVERED
Comments	

- **Efficiency**

Visual analytics	
ID: NFR-2.4.1	Responsiveness
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	
Description	The user interface must minimize the delay between user interactions and system responses, so that the user is immediately aware of the performed tasks
Priority	MODERATE
Status	UNCOVERED
Comments	

- **Others**

Visual analytics	
ID: NFR-2.4.1	Documentation
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	
Description	
Priority	LOW
Status	UNCOVERED
Comments	

Visual analytics	
ID: NFR-2.4.2	Open-source
Version	0.0.1 2016-04-28
Subsystem	Visualisation library
Dependencies	
Description	The source code of the visualisation library must be available as Open Source
Priority	HIGH
Status	UNCOVERED
Comments	

5.3 Subsystem Data Processing

5.3.1 Functional requirements

Data Processing	
ID: FR-3.1.1	*Data input*
Version	0.0.1 2016-05-04
Subsystem	Hybrid engine
Dependencies	
Description	The input is categorized as static and dynamic inputs. Static input can be historical data and dynamic input is real time data received from coil production sensors.
Priority	HIGH
Status	UNCOVERED
Comments	

Data Processing	
ID: FR-3.1.2	*Join operation*
Version	0.0.1 2016-05-04
Subsystem	Hybrid engine
Dependencies	Data input
Description	Join operation takes place between static and dynamic (stream) data. We put the stream data into windows and treat static data as single window. The result of this operation is set of windows or *WindowedStream*.
Priority	HIGH
Status	UNCOVERED
Comments	

Data Processing	
ID: FR-3.1.3	*Union operation*
Version	0.0.1 2016-05-04
Subsystem	Hybrid engine
Dependencies	Data input
Description	Union operation takes place between static and dynamic (stream) data. We put the stream data into windows and treat static data as a single window. The result of this operation is set of windows or *WindowedStream*.
Priority	HIGH
Status	UNCOVERED
Comments	

Data Processing	
ID: FR-3.1.4	*Hybrid workflow*
Version	0.0.1 2016-05-04
Subsystem	Hybrid engine
Dependencies	Hybrid operations
Description	The structure of workflows in Hybrid engine is same as it is in streaming workflow of Apache Flink, as the previous is built on top of latter. However, there are some changes to be done in managing the state in hybrid operators, constructing mutable windows (that can be converted from both static and dynamic data) and etc.
Priority	HIGH
Status	UNCOVERED
Comments	

Data Processing	
ID: FR-3.1.5	*System output*
Version	0.0.1 2016-05-04
Subsystem	Hybrid engine
Dependencies	Hybrid operations
Description	The output is integrated with batch and online computation results that allows us to efficiently generate up-to-date intermediate results which will be visualized by the visualization layer.
Priority	HIGH
Status	UNCOVERED
Comments	

Data Processing	
ID: FR-3.1.6	Adaptivity
Version	0.0.1 2017-07-20
Subsystem	Hybrid engine
Dependencies	
Description	The data processing system should be able to adapt dynamically to new conditions, evolving on time when required.
Priority	HIGH
Status	UNCOVERED
Comments	

5.3.2 Non-functional requirements

- **Usability**

Data Processing	
ID: FR-3.2.1	*Usability*
Version	0.0.1 2016-05-04
Subsystem	Hybrid engine
Dependencies	Hybrid operations
Description	To use the hybrid engine, it is essential to provide high level APIs to users. That is, the users of the system need not to know the internals of data processing engine. Moreover, the declarative language is another important part of system usability. Machine learning experts can use this language to express their queries and the system will automatically convert them to workflows.
Priority	HIGH
Status	UNCOVERED
Comments	

- **Security**

Data Processing	
ID: FR-3.2.2	*Security*
Version	0.0.1 2016-05-04
Subsystem	Hybrid engine
Dependencies	Hybrid operations, Data input
Description	This module is a utility that lets program code run in a security context provided by the Hadoop security user groups. The secure context will for example pick up authentication information from Kerberos.
Priority	HIGH

Status	UNCOVERED
Comments	

- **Reliability**

Data Processing	
ID: FR-3.2.3	*Reliability*
Version	0.0.1 2016-05-04
Subsystem	Hybrid engine
Dependencies	Hybrid operations, Data input
Description	The most important factors underpinning Hybrid Engine are their reliability and high availability support. By default, there is a single JobManager instance per Flink cluster. This creates a single point of failure (SPOF): if the JobManager crashes, no new programs can be submitted and running programs fail. With JobManager High Availability, it is possible to recover from JobManager failures and thereby eliminate the SPOF. So, configuration of high availability for both standalone and YARN clusters is essential part of system.
Priority	HIGH
Status	UNCOVERED
Comments	

- **Portability**

Data Processing	
ID: FR-3.2.4	*Portability*
Version	0.0.1 2016-05-04
Subsystem	Hybrid engine
Dependencies	Hybrid operations

Description	We will provide a declarative language for expressing analytical queries that can spawn either batch or online computations, without requiring any changes to the analysis algorithm. By this way, the system will be portable with other platforms.
Priority	HIGH
Status	UNCOVERED
Comments	

- **Efficiency**

Data Processing	
ID: FR-3.2.5	*Efficiency*
Version	0.0.1 2016-05-04
Subsystem	Hybrid engine
Dependencies	Hybrid operations
Description	To achieve the efficiency, we provide optimized architectures for automated distribution of tasks over clusters for effective stream data mining and historical data mining
Priority	HIGH
Status	UNCOVERED
Comments	

6 Conclusions

This deliverable presents the catalogue of requirements that the software modules to be developed in the context of PROTEUS. The requirements are derived from the ArcelorMital case study providing guidance to the whole project. However, the software solutions proposed in PROTEUS will be generic enough for any stream-based processing use case.

Software requirements are divided in the subsystems:

- Predictive Analytics: this subsystem will address the requirements of building fast prediction models for online data streams.
- Visual Analytics: in this subsystem we will develop a visualization system to help end-user understanding both data and predictions as well as an interactive tool for data exploration
- Data processing: this subsystem will address the requirements of combining data-at-rest and data-at-motion on top a scalable data processing engine as Apache Flink.

The final aim of PROTEUS is to obtain a system that is interactive and fulfils the requirements associated with scalable online machine learning, hybrid data processing and interactive real-time visual analytics.

References

[1] "Storm, distributed and fault-tolerant realtime computation." [Online]. Available: http://storm-project.net/. [Accessed: 10-Jun-2013].

[2] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: cluster computing with working sets," p. 10, Jun. 2010.

[3] "MLlib | Apache Spark." [Online]. Available: https://spark.apache.org/mllib/. [Accessed: 10-Apr-2015].

[4] Apache Software Foundation, "Apache Flink," 2014. [Online]. Available: https://flink.apache.org/. [Accessed: 29-Mar-2016].

[5] "SQLstream Blaze | Real-time Processing for Big Data in Motion." [Online]. Available: http://www.sqlstream.com/. [Accessed: 23-May-2016].

[6] "DataTorrent. Unified Batch and Stream Processing Platform for Enterprises | DataTorrent." [Online]. Available: https://www.datatorrent.com/. [Accessed: 23-May-2016].

[7] "Tableau Desktop | Tableau Software." [Online]. Available: http://www.tableau.com/products/desktop. [Accessed: 31-Mar-2015].

[8] "CartoDB." [Online]. Available: https://cartodb.com/. [Accessed: 23-May-2016].

[9] "Gephi - The Open Graph Viz Platform." [Online]. Available: https://gephi.org/. [Accessed: 23-May-2016].

[10] "Kibana." [Online]. Available: https://www.elastic.co/products/kibana. [Accessed: 23-May-2016].

[11] "D3.js - Data-Driven Documents." [Online]. Available: http://d3js.org/. [Accessed: 01-Apr-2015].

[12] A. Thusoo, J. Sen Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy, "Hive," *Proc. VLDB Endow.*, vol. 2, no. 2, pp. 1626–1629, Aug. 2009.

[13] "Cloudera Impala: Real-Time Queries in Apache Hadoop, For Real | Apache Hadoop for the Enterprise | Cloudera." [Online]. Available: http://blog.cloudera.com/blog/2012/10/cloudera-impala-real-time-queries-in-apache-hadoop-for-real/. [Accessed: 10-Jun-2013].

[14] N. Marz and J. Warren, *Big Data Principles and best practices of scalable realtime data systems*. Manning Publications Co., 2014.

[15] Twitter, "Summingbird," 2013. [Online]. Available: http://www.infoq.com/news/2014/01/twitter-summingbird.

[16] R. Casado, "Lambdoop, a framework for easy development of Big Data applications," in *NoSQL Matters Barcelona*, 2013.

[17] "Apache Mahout: Scalable machine learning and data mining." [Online]. Available: http://mahout.apache.org/. [Accessed: 11-Jun-2013].