PROTEUS
Scalable online machine learning for predictive analytics and real-time interactive visualization
687691

D6.2 PROTEUS evaluation and impact assessment

Lead Author: Graham Hesketh, Giuseppe Maio, Rachel Finn & Kush Wadhwa, Trilateral Research

With contributions from: DFKI, BU, AMIII

Reviewer: Hamid Bouchachia

<table>
<thead>
<tr>
<th>Deliverable nature:</th>
<th>Report (R)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dissemination level:</td>
<td>Public (PU)</td>
</tr>
<tr>
<td>Contractual delivery date:</td>
<td>30/11/2018</td>
</tr>
<tr>
<td>Actual delivery date:</td>
<td>30/11/2018</td>
</tr>
<tr>
<td>Version:</td>
<td>1.0</td>
</tr>
<tr>
<td>Total number of pages:</td>
<td>31</td>
</tr>
<tr>
<td>Keywords:</td>
<td>KPIs, impact assessment, evaluation</td>
</tr>
</tbody>
</table>
Abstract
This deliverable performs an impact assessment of PROTEUS. We define and use KPIs to assess the impact of PROTEUS technologies and evaluate the collective performance of its components. KPIs are grouped into 1) PROTEUS components KPIs, made up of data process KPIs, i.e. PROTEUS’ hybrid engine evaluations, a Lara language evaluation, online machine learning KPIs, and data visualisation KPIs; and 2) industrial value KPI, made up of a process improvement KPI. We identified a great deal of PROTEUS- driven impacts to both AI and Smart Manufacturing in the EU. These include: the design and implementation of a hybrid computation system able to process both data-at-rest and data in motion; the design and implementation of Lara, a high-level language to express online machine learning using streaming data; the development of PROTEIC, an online interactive visualisation dashboard; the implementation of an open source library, SOLMA, containing a great deal of innovative machine learning algorithms that specialise in online streaming data analysis; the computation of a process improvement KPI to be used to assess the extent to which machine-learning classifiers generate value for steel coil manufacturers.
Executive summary

This deliverable performs an impact assessment of PROTEUS. PROTEUS components KPIs provide information on the project’s contributions to machine learning as well as smart manufacturing in Europe, as demonstrated in an industrial use case with an ArcelorMittal steel coil factory.

Some specific results include:

- While latency measurements of 1 second are normally acceptable, the PROTEUS hybrid computation engine for batch and streaming data processing delivers millisecond latencies, providing near real-time interactive analysis and defect alerts to workers on the AMIII factory floor.
- PROTEUS’ abstraction language, LARA, simplifies the definition and optimization of user defined functions for batch and streaming data analytics, reducing the learning curve and removing barriers to mastering online machine learning techniques.
- PROTEUS’ machine learning library SOLMA provides algorithms for regression and classification with batch and streaming data which were not available in existing batch processing libraries such as Flink-ML. In the AMIII use case, these new tools for predictive analytics enable the early stage prediction of steel coils as defective, prompting rapid cost saving intervention strategies.
- PROTEUS’ powerful data visualization platform for streaming data, PROTEIC, can generate 28 plot types supported by multiple interactive features that put big data insights at the disposal of AMIII factory workers.

Taken together these impacts represent powerful innovations to both the EU’s machine learning, big data and smart-manufacturing industries.
## Document Information

<table>
<thead>
<tr>
<th>IST Project Number</th>
<th>687691</th>
<th>Acronym</th>
<th>PROTEUS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Full Title</strong></td>
<td>PROTEUS evaluation and impact assessment</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Project URL</strong></td>
<td><a href="http://www.proteus-bigdata.com">www.proteus-bigdata.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>EU Project Officer</strong></td>
<td>Martina EYDNER</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Deliverable Number</strong></td>
<td>6.2</td>
<td><strong>Title</strong></td>
<td>PROTEUS evaluation and impact assessment</td>
</tr>
<tr>
<td><strong>Work Package Number</strong></td>
<td>WP6</td>
<td><strong>Title</strong></td>
<td>Communication, exploitation and dissemination</td>
</tr>
<tr>
<td><strong>Date of Delivery</strong></td>
<td>Contractual M36</td>
<td><strong>Actual</strong></td>
<td>M36</td>
</tr>
<tr>
<td><strong>Status</strong></td>
<td>version v.1 final ■</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Nature</strong></td>
<td>report ■ demonstrator □ other □</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Dissemination level</strong></td>
<td>public ■ restricted □</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Authors (Partner)</strong></td>
<td>TRI</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Responsible Author</strong></td>
<td>Name</td>
<td>E-mail</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Partner</td>
<td>Phone</td>
<td></td>
</tr>
</tbody>
</table>

## Abstract

(for dissemination)

.
.

## Keywords

KPIs, impact assessment, evaluation

## Version Log

<table>
<thead>
<tr>
<th>Issue Date</th>
<th>Rev. No.</th>
<th>Author</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>13/11/2018</td>
<td>0.1</td>
<td>Giuseppe Maio</td>
<td>Changes made following Graham Hesketh’s review.</td>
</tr>
<tr>
<td>26/11/2018</td>
<td>0.2</td>
<td>Giuseppe Maio</td>
<td>Changes made following Graham Hesketh’s review</td>
</tr>
<tr>
<td>27/11/2018</td>
<td>0.3</td>
<td>Giuseppe Maio</td>
<td>Changes made following Graham Hesketh’s review</td>
</tr>
<tr>
<td>27/11/2018</td>
<td>0.4</td>
<td>Giuseppe Maio</td>
<td>Changes made following Graham Hesketh’s review</td>
</tr>
<tr>
<td>28/11/2018</td>
<td>0.5</td>
<td>Giuseppe Maio</td>
<td>Changes made following Graham Hesketh’s review</td>
</tr>
<tr>
<td>29/11/2018</td>
<td>0.6</td>
<td>Graham Hesketh, Giuseppe Maio</td>
<td>Comments added from DFKI, AMIII and BU</td>
</tr>
<tr>
<td>30/11/2018</td>
<td>1.0</td>
<td>Graham Hesketh, Giuseppe Maio, Rachel Finn &amp; Kush Wadhwa</td>
<td>Changes made following review of DFKI, AMIII and BU</td>
</tr>
</tbody>
</table>
## Table of Contents

Executive summary ........................................................................................................... 3
Document Information ....................................................................................................... 4
Table of Contents ................................................................................................................ 5
List of figures ....................................................................................................................... 7
Abbreviations ...................................................................................................................... 8
Introduction .......................................................................................................................... 9

1. PROTEUS’ Hybrid Engine: Advances in Batch & Stream Processing ......................... 11
   1.1 PROTEUS Hybrid Model .......................................................................................... 11
       1.1.3 PROTEUS User-level Programming API ......................................................... 12
   1.2 Benchmarking PROTEUS’ hybrid engine for batch and streaming data ................. 13
       1.3 Summary .............................................................................................................. 13

2. PROTEUS’ Declarative Language LARA .................................................................... 15
   2.1 Summary .............................................................................................................. 15

3. PROTEUS’ Advancement of Online Machine Learning .................................................. 16
   3.1 PROTEUS’ Online Machine Learning Library SOLMA ........................................ 16
   3.2 Evaluation of classification algorithms: flatness accuracy use case ......................... 18
       3.2.1 WAPA Algorithms Applied to Flatness Positions on a Coil .............................. 18
       3.2.2 Large Scale Experiment for Classifying Coils Using ONLMSR ....................... 19
   3.3 Evaluation of regression algorithms: flatness regression prediction use case .......... 19
       3.4 Summary .............................................................................................................. 20

4. PROTEUS’ Data Visualisation Platform PROTEIC ...................................................... 21
   4.1 Plot types in PROTEIC ............................................................................................ 21
   4.2 Interactive functionality in PROTEIC ...................................................................... 24
   4.3 Factory Expert’s end-user evaluation of PROTEIC .................................................. 25
       4.3.1 User’s positive feedback on PROTEIC features .............................................. 26
       4.3.2 User’s suggestions for the improvement of PROTEIC ...................................... 26
   4.4 Summary .............................................................................................................. 26

5. Industrial Value KPI ..................................................................................................... 27
   5.1 Process improvement KPI ....................................................................................... 27
   5.2 Summary .............................................................................................................. 29

6. Barriers to Impact ....................................................................................................... 30
   6.1 Issues with the data ............................................................................................... 30
   6.2 Issues with factory implementation ....................................................................... 30
   6.3 Departure of PROTEUS Partners ......................................................................... 31
   6.4 Summary .............................................................................................................. 31

7. Conclusion ..................................................................................................................... 32
References ................................................................. 33
List of figures

Figure 1. Conventional hybrid architecture and PROTEUS hybrid architecture .................................................. 12
Figure 2. Code Snippet written in Lara ................................................................................................................. 15
Figure 3. Annotation Plot & Gauge Chart ............................................................................................................ 21
Figure 4. Area Plots ........................................................................................................................................ 22
Figure 5. Heat Maps ........................................................................................................................................ 22
Figure 6. Line Charts ...................................................................................................................................... 22
Figure 7. Scatterplots ..................................................................................................................................... 23
Figure 8. Streamgraphs .................................................................................................................................. 23
Figure 9. Network Graphs & Swim lane ........................................................................................................... 23
Figure 10. Sun Bursts ....................................................................................................................................... 24
Figure 11. Parallel Coordinates ....................................................................................................................... 24
Figure 12. Example of the “pause” interaction feature implementation in PROTEIC ........................................... 25
Figure 13. Current implementation of the confidence band board ................................................................. 25
Abbreviations

AI: Artificial Intelligence
BDVA: Big Data Value Association
DPA: Distributed passive-aggressive
EFFRA: European Factories of the Future Research Association
ETL: Extract, Transform, Load
Flink-ML: Machine learning library for Flink
GDP: Gross Domestic Product
IoT: Internet of Things
KPI: Key Performance Indicators
LASSO: Least absolute shrinkage and selection operator
ONLMSR: Online Normalised Least Mean Squares regression
PPP: Public Private Partnership
RMSE: Root mean squared error
SAX: Symbolic Aggregate approximation
SME: Small and Medium-sized Enterprises
SOLMA: Scalable Online Machine Learning Algorithm
WAPA: Weighted averaging passive-aggressive

PROTEUS’ Partners:
AMIII: ArcelorMittal
BU: Bournemouth University
DFKI: Deutsches Forschungszentrum für Künstliche Intelligenz
LMDP: Lambdoop
TREE: Treelogic
TRI: Trilateral Research
Introduction

The PROTEUS project is focused on developing data analytics prototypes for streaming and batch data processing. These techniques are to be implemented in a use case with leading steel company ArcelorMittal (industrial user) to solve challenging issues in its steelmaking process. ArcelorMittal has defined requirements for a technological solution, based on online, scalable machine learning that is to be able to better control and track defects in the steel coils they produce resulting in continuous improvements in the steelmaking process.

Using these industrial user-defined requirements as a starting point, the project built Scalable Online Machine Learning (SOLMA) algorithms, a hybrid processing engine and data visualisation components to align the project outputs to the end-user industrial requirements as far as possible. Furthermore, the project sought to ensure these machine learning and data analytics innovations were effective beyond the specific AMIII use case and applicable to other domains, but particularly other smart manufacturing scenarios.

This deliverable performs an impact assessment of the PROTEUS toolset components. It presents a set of KPIs previously developed in D2.8-D2.11 and we use these KPIs to assess the impact of PROTEUS technologies and evaluate the collective performance of its components. PROTEUS’ KPIs have undergone a continuous process of revision starting immediately after the proposal stage. They thus reflect the adjustments made to the objectives of the project as well as its development. The KPIs presented herein provide both quantitative and qualitative measures to estimate PROTEUS’ impacts in the context of smart manufacturing in Europe. In what follows, we provide definitions of artificial intelligence and smart manufacturing, describe the policy context of smart manufacturing in Europe and its strategic importance, and present PROTEUS KPIs which we discuss more in detail in the following pages.

Artificial intelligence (AI) refers to computer systems able to discover, process, analyse, and interpret patterns in datasets providing intelligence for decision making (Software.org 2018, p. 2). AI is an umbrella term encompassing reasoning and machine learning (including neural networks, statistical learning and computational intelligence). Any technique that enables computers to emulate human behaviour using a set of algorithms can be defined as AI (ibid p.4). Machine learning is an approach that uses statistical methods to allow applications to learn from experience. Machine learning models use algorithms that take advantage of information from training datasets to make predictions or recognitions when fed new data (ibid, p. 5).

AI and machine learning contributed to advancements in many industries. Some of these are listed below.

- In Agriculture, AI contributed to making farmers more productive, creating smart tractors, killing weeds more accurately etc. (ibid, p. 5);
- In Transportation, AI contributed to creating smarter traffic lights to reduce waiting time at red lights, using autonomous boats to monitor fish stock, cutting train delays, etc. (ibid, p.7).
- In Manufacturing, AI contributed to alternative product design, materials improvement, detecting defects, preventing breakdowns, optimising assembly lines (ibid., p.12).

The use of AI in the manufacturing sector and within factories has revolutionised the sector giving birth to what is known as smart manufacturing. The term smart manufacturing broadly defines a series of technology-driven measures aiming to optimise manufacturing processes. It consists for instance of employing computer controls, big data modelling, and other innovations to improve the efficiency
of the manufacturing sector (Manufacturing Tomorrow 2017). The National Institute of Standards and Technology refers to smart manufacturing as “fully-integrated collaborative manufacturing systems that respond in real time to meet changing demands and conditions in the factory, in the supply network and in customer’s needs” (NIST 2014).

The manufacturing sector in Europe accounts for 2 million enterprises, 33 million jobs, and 6% of Europe's GDP. Smart manufacturing is responsible for 64% of private sector Research & Development expenditure and for 49% of innovation expenditure in Europe (EC 2018).

The Big Data Value Association (BDVA) contributed to set the EU’s smart manufacturing agenda in the last decade. The BDVA highlighted three smart manufacturing research/policy areas that are of strategic interest to the Public Private Partnership (PPP) between the EU and the European Factories of the Future Research Association (EFFRA) (BDVA 2018), aiming to advance smart manufacturing in Europe. These are:

1. **Smart Factories** characterised by data-generation internal to the production processes, where smart analytics is needed for safety, optimisation and diagnosis of the plant as well as the health and safety of blue-collar workers;
2. **Smart Supply Chain** characterised by data generation within an ecosystem of suppliers, providers, distributors, and retailers; analytics is functional to supply chain integration and collaboration between white collar workers;
3. **Smart Product Lifecycle** in which the data is generated by the product itself in its life-cycle and analytics is functional to product monitoring (ibid p. 9).

The PROTEUS project’s mission is to implement scalable machine learning algorithms for predictive analytics using streaming data. This is likely to have many applications but an important use-case considered herein, aims to improve ArcelorMittal’s (AMIII) factory processing to detect defective steel coils before these are marketed as high-quality ones, resulting in economic loss. The data used throughout the PROTEUS project was supplied by AMIII. The PROTEUS use case thus addresses the first research/policy area, **Smart Factories**. Once defective coils are detected, they can be re-used for other industrial purposes or re-marketed in a more profitable way for the manufacturer; this process is known as re-manufacturing (ibid, p.9).

We grouped the KPIs considered for the impact assessment of PROTEUS into two categories: **PROTEUS components KPIs** and **Industrial value KPIs**. PROTEUS components KPIs are made up of data process KPIs, i.e. PROTEUS’ hybrid engine evaluations (Section 1), a Lara language evaluation (Section 2), online machine learning KPIs (Section 3), and data visualisation KPIs (Section 4) to reflect the four technical components of PROTEUS. PROTEUS components KPIs provide information on the project’s contribution to AI and Big Data Analytics in Europe. The Industrial value KPI consists of a process improvement KPI (Section 5) to assess the extent to which PROTEUS generates added value for AMIII. This KPI can be extended outside AMIII’s use case to the manufacturing sector more broadly, clarifying the extent to which the use of a machine learning classifier generates value for manufacturers. This deliverable also considers the barriers to PROTEUS’ impacts (Section 6).
1. PROTEUS’ Hybrid Engine: Advances in Batch & Stream Processing

The PROTEUS hybrid engine is a data analytics and workflow platform for analysing streaming and historical data, such as that coming from Arcelor Mittal’s (AMIII) sensors installed in their factory infrastructure (see D3.2, D3.3, D3.9). It is referred to as a hybrid processing engine, for its ability to deal with both streaming and historical data simultaneously in one stream rather than in two distinct processing applications. The hybrid processing engine enables PROTEUS to develop real-time analysis algorithms and predictive models which will take into account both the historical data that AMIII has been collecting for more than a decade, as well as current rapidly produced data coming from the sensors in the steel factory.

The standard approach for analysing high-volume data streams (e.g., log data) is to run periodic batch jobs (e.g., hourly). Batch processing frameworks (e.g., Hadoop) primarily focus on static data and often have difficulty handling latency-sensitive workloads, such as interactive analyses, where latency is the time delay required for packets of data to be retrieved, displayed or processed.

In smart manufacturing there is a need for low latency responses to complicated data queries over large volumes of data and across infinite streams of data. Such queries include statistical analysis and machine learning, for example, and this demands quality real time processing capabilities. PROTEUS provides accurate results in such analyses because it can use more data and therefore perform better training of predictive models. The use case of AMIII, is a good example of a time-critical application which required intensive data analytics applied to both streaming data coming from sensors and batches of historical data of past production processes. In the AMIII steel factory, the system must quickly alert factory workers to any anomaly detected (e.g., malformed steel coils) in order for them to fix the issues, but it must also be able to make accurate predictions of steel quality and that requires processing large amounts of historical data. This demonstrates the need for an integrated platform which is able to provide instant answers and detect anomalies (low latency) but also to be able to use historical data in order to e.g., train predictive models.

In addition to low latency, the system is required to be simple and customisable so that it can handle various use-cases and adjustments to changing factory processes and associated changing data access requirements. Like any big data processing, scalability is also critical. PROTEUS provides user simplicity through its PROTEUS Application Programming Interface (API) and scalability via its ability to improve processing performance with more parallel computing power (i.e., more compute nodes).

In the remainder of this section, we present and evaluate PROTEUS’ advances with respective to the hybrid scenario of batch and stream processing. In Section 1.1 we first describe why Apache Flink was chosen as a base from which to build rather than Apache Spark or Apache Storm. We then introduce the innovation of the PROTEUS hybrid model and its user-friendly programming interface, the PROTEUS API. In Section 1.3, experiments benchmarking the low-latency scalable performance of the PROTEUS hybrid computation model are described.

1.1 PROTEUS Hybrid Model

In this section we give an overview of the innovations brought to stream and batch processing through the PROTEUS hybrid computation model.

1.1.1 Choosing Apache Flink for PROTEUS
Apache-Flink is a popular streaming data processing system and an integral component in PROTEUS. Karimov et al (2018) benchmarked the performance of Apache-Flink against similar popular systems such as Apache-Storm and Apache-Spark. According to Karimov et al 2018, there are a number of benefits offered by Apache Flink that make it particularly well suited to a hybrid engine such as PROTEUS.

These benefits and their importance to the AMIII use case, made Apache Flink the best platform upon which to build PROTEUS. In addition, this will also enable the scalability and applicability of PROTEUS to other industries which are also characterised by requirements for near real-time interventions and analysis, such as the gaming industry (Karimov, 2018).

1.1.2 PROTEUS hybrid architecture and side inputs

Despite its name being hybrid, the PROTEUS implemented architecture consists of a single data type. That is, instead of using system architecture like in Figure 2 (a), PROTEUS uses a simpler and more efficient one, as shown in Figure 2 (b). Instead of using two abstractions over which programs are defined, PROTEUS defines everything over a stream processor.

![Figure 1. Conventional hybrid architecture and PROTEUS hybrid architecture](image)

Again, this hybrid model was appropriate for the PROTEUS use-case because it allowed AMIII historical data and AMIII real-time sensor data to be analysed in an integrated manner.

A key concept in the operation of the PROTEUS hybrid engine is the introduction of side inputs. A side input is a secondary input of a stream that enables a number of important functionalities, which in PROTEUS include the joining of streaming data with static data, the joining of streaming data with slowly evolving data and, crucially, dynamic machine learning model updates. In the AMIII factory, it is important than an analyst can control these operations in a simple and efficient way if they are to make regular adjustments to changing factory processes and their associated varying data access requirements. In order to be able to fully control these side input operations, the end-user was provided with what is referred to as the PROTEUS application programming interface (API) described in Section 1.1.3.

1.1.3 PROTEUS User-level Programming API

The Proteus API is integrated with the Flink API and it lets users create new side inputs (for instance, from a historical data set) and access them in user-defined functions. This is important because the user-defined functions will embody the data access and processing customisation required for various use-cases and manufacturing scenarios. The design of the PROTEUS API introduces a few changes to the User-level Programming Java and Scala API, including new methods in the Streaming Execution Environment in order to let users create side inputs from an already existing Data Stream. The PROTEUS API simplifies the development of machine learning algorithms and enables their quick deployment in a production environment. This innovation provides agility in smart
manufacturing which enables the end user to adjust to new use-cases and changing factory processes with minimal difficulty.

1.2 Benchmarking PROTEUS’ hybrid engine for batch and streaming data

In addition to successfully performing these integrated analyses, the hybrid engine needed to also perform them in ways which were at least as good as existing tools. An important measure of performance for a batch and streaming data processor where near real-time information is required is latency. Generally a standard benchmark for acceptable performance is a latency measurement of 1 second; for example Amazon Web Services sets the default end-to-end latency of its streaming processing service AWS Kinesis to 1 second.

In D3.9 partners performed experiments to benchmark the performance of PROTEUS’ hybrid engine. Latency was benchmarked at maximum throughput using an Intel Xeon CPU E5530 2.40GHz with 16 CPUs. The experiments were also performed on both 2-node and 4-node clusters to investigate the performance scalability that would come with more nodes and thus more parallel computing power. The Latency measurements are summarised in the table below. Latency reduces when training data increases and more noticeably when more nodes are used, indicating the performance benefits of scaling. In the experiments, PROTEUS’ hybrid engine was able to achieve millisecond latencies.

<table>
<thead>
<tr>
<th></th>
<th>20% training data</th>
<th>40% training data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 nodes</td>
<td>4 nodes</td>
</tr>
<tr>
<td>End-to-end average latency (seconds)</td>
<td>0.69</td>
<td>0.54</td>
</tr>
<tr>
<td>Hybrid operator average latency (seconds)</td>
<td>0.81</td>
<td>0.67</td>
</tr>
</tbody>
</table>

As mentioned in the introduction of this section, the standard approach for analysing high-volume data streams primarily focused on static data and had difficulty handling latency-sensitive workloads, such as interactive analyses. In PROTEUS there was a need for low latency responses to long-running complex queries over high-volume, infinite streams of data and these results show that PROTEUS has delivered against that need. Specifically, the hybrid engine within PROTEUS detects and displays these anomalies in fractions of a second, enabling operators to intervene as quickly as possible. This section was critical to supporting the machine learning capabilities in Section 3.

1.3 Summary

This section has outlined the capabilities of the PROTEUS hybrid engine for batch and streaming data. It described why Flink was chosen as the starting point from which to build PROTEUS. It then demonstrated the innovation of the PROTEUS hybrid computation engine, with its single processor for both batch and streaming built with side inputs that enable the joining of streaming data with static data, the joining of streaming data with slowly evolving data and, crucially, dynamic machine learning model updates, all of which are controlled through the PROTEUS API. Ultimately, where latency measurements of 1 second are generally acceptable, PROTEUS delivers millisecond latencies, providing near real-time interactive analysis and defect alerts to workers on the AMIII factory floor.
2. PROTEUS’ Declarative Language LARA

Another notable contribution in PROTEUS was the design and implementation of Lara, a high-level language in which it is simple to code streaming machine learning (see D3.6, D3.7, D3.8, in particular T3.3). This is the first stable declarative programming language specifically tailored for online machine learning using streaming data. Other programming languages for machine learning, i.e., Python and R, specialise mainly in offline algorithms for data-at-rest stored locally or in the cloud. LARA allows users to build algorithms that see the data only once and use no memory, i.e., online streaming algorithms. Because big data analytics is dominated by offline algorithms, LARA is a significant contribution to European big data analytics allowing users to easily use the PROTEUS-based hybrid engine built on top of Flink.

The API closely resembles languages such as R, Matlab, and Python libraries such as NumPy, as shown in Figure 3. Data science experts are traditionally accustomed to such languages, and this innovation makes the language more accessible to and easier to implement by a wide population of data scientists. Partners thus enabled data scientists, who do not have system-specific knowledge, to efficiently develop scalable machine learning analytics through Lara.

```java
// read measurements into the DataBags A and B
val A = readCSV(...) //
val B = readCSV(...) //
// SELECT a₁, ..., aₜ, b₁, ..., bₑ //
// FROM A, B
// WHERE A.id = B.id
val X = for {
    a <- A
    b <- B
    if a.id == b.id
    } yield (a₁, ..., aₜ, b₁, ..., bₑ)

// Convert the DataBag X into a Matrix
val M = X.toMatrix() //
// Calculate the mean of each column of the matrix
val means = for (c <- M.cols()) yield mean(c) //
// Compute the deviation of each cell of M
// to the cell’s column mean.
val U = M - Matrix.fill(M.numRows, M.numCols)
    ((i,j) => means(j)) //
// compute covariance matrix
val C = 1 / (U.numRows - 1) * U.t %*% U
// compute singular value decomposition
// e.g. rescale M, reduce dimensions, etc.
```

Figure 2. Code Snippet written in Lara

2.1 Summary

LARA allows data scientists to build algorithms for streaming data, facilitating their activities by performing all the mathematical operations related to the streaming data processing in the background. Hence, data scientists can focus on the algorithmic development without worrying about engine issues. This will reduce the learning curve and reduce barriers to entry when mastering online machine learning techniques and ultimately make it more accessible to more smart manufacturing data scientists.
3. PROTEUS’ Advancement of Online Machine Learning

One of PROTEUS’ key objectives is to advance machine learning in the smart manufacturing sector. As stated in a Software.org report, “the test of strength for tomorrow’s manufacturing economy isn’t how things are built, but whether our manufacturing future is built with software, the cloud, and data” (Software.org 2018, p. 3). PROTEUS aims to pave the way for the re-orientation of the European manufacturing sector towards smart manufacturing so that manufacturers can tap into the opportunities provided by software development, cloud architecture and big data. PROTEUS aims to have a substantive impact on employment in Europe as manufacturers are now hiring more software developers than production line workers (ibid., p. 4). Manufacturers are adopting software solutions for 3D design, additive manufacturing, and internet of things-driven technologies to design, build and deliver new products (ibid, p.3).

An adequate set of machine learning tools for data analytics applied to streaming data is a must in smart manufacturing. Although FlinkML provides a nice set of traditional machine learning algorithms on top of Flink Batch API, there is no support for data streams. For stream data processing such as that performed in PROTEUS, data analytics must satisfy new requirements including low memory usage, low processing time, items that can be processed at most once and prediction should be possible at any stage.

Here, online learning plays a prominent role where the learning process can be seen as a game between the algorithm and the reality/nature: given an input at time t, first the algorithm predicts the output, then the reality reveals the true output, finally the algorithm suffers a loss (≥0) and self-adjusts. This process repeats as new data becomes available. The proposed algorithms are implemented in SOLMA using some pipeline abstractions (i.e., Estimator, Transformer, and Predictor pattern) (see D4.5).

SOLMA represents one of PROTEUS’ main achievements in which new algorithms enabling scalable online learning were developed (details to be found in D4.1- D4.5). A gentle coverage of SOLMA and its algorithms is to be found in this paper by Jamil et al (2018), in this presentation by Bouchachia (2018), and on GitHub (PROTEUS 2018). In Section 3.1 we summarise the impact of the SOLMA library and the way it surpasses the state of the art.

In Section 3.2 we demonstrate some online machine learning algorithms from SOLMA in a use-case with AMIII data on steel coils. The algorithms are for classification in supervised machine learning and we discuss performance in terms of accuracy and true positive rates.

In Section 3.3 we demonstrate some online machine learning algorithms from SOLMA again in a use-case with AMIII data on steel coils but this time the algorithms are for regression in supervised machine learning and we discuss performance in terms of the coefficient of determination, $R^2$, root mean squared error (RMSE) and computational complexity $O(n)$.

3.1 PROTEUS’ Online Machine Learning Library SOLMA

PROTEUS’ innovative SOLMA contribution goes beyond what is achievable with FlinkML. Within SOLMA, several algorithms were developed. Stream-data-specific components, i.e., Transformers (Reservoir samplers; Online PCA; Random Projection; SVD), Estimators (Moments; Heavy hitters), and Predictors (VHT) were used to build the SOLMA abstraction. SOLMA is a library entirely built for online learning, where data is used sequentially only once and where the algorithms operate with
“no memory” (Jamil et al. 2018). SOLMA is therefore completely independent from Flink-ML. This is an important contribution, placing PROTEUS as an innovator of AI big data analytics in Europe.

SOLMA is the first open source machine learning library which specialises in online learning. Some of the algorithms provided in SOLMA are also written in other languages (R, Python) and will be made available as open source on Github. This allows for community-driven developments and enhancements of SOLMA. The tables below show some of the functionality that exists in FlinkML which is for batch processing and the new functionality included in SOLMA for stream and batch hybrid processing. These new functionalities are a significant advance for big data stream and hybrid processing that will have a particularly notable impact in smart manufacturing as it allows the predictive power of regression and classification algorithms to be applied in stream and hybrid processing, which in the AMIII use case enables steel coils to be predicted as defective or otherwise as described in Sections 3.2 and 3.3. The table below summarises the new functionalities provided by the SOLMA library for streaming and batch data that were not available previously in batch processing libraries such as FlinkML.

**SOLMA Functionalities**

<table>
<thead>
<tr>
<th>Basic Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online moments: simple mean, simple variance, weighted mean, weighted variance, exponentially weighted mean and variance, moving average aggregation algorithm;</td>
</tr>
<tr>
<td>Online sampling: simple reservoir sampling, weighted reservoir sampling, adaptive reservoir sampling;</td>
</tr>
<tr>
<td>Incremental principle component analysis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classification Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online support vector machines (OSVM)</td>
</tr>
<tr>
<td>Online bi-level stochastic gradient for support vector machines (OBSG-SVM)</td>
</tr>
<tr>
<td>Online passive-aggressive algorithms (PA)</td>
</tr>
<tr>
<td>Online weighted passive-aggressive algorithms (WAPA)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Regression algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online ridge regression (ORR)</td>
</tr>
<tr>
<td>Online shrinkage via limit of Gibbs sampling (OSLOG)</td>
</tr>
<tr>
<td>Aggregation algorithm for regression (AAR)</td>
</tr>
<tr>
<td>Competitive online ridge regression (COIRR)</td>
</tr>
<tr>
<td>Online Least Absolute Shrinkage and Selection Operator (LASSO)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Drift handling and anomaly detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online weighted averaging passive-aggressive algorithm (WAPA)</td>
</tr>
<tr>
<td>Online Normalised Least Mean Squares regression (ONLMSR)</td>
</tr>
<tr>
<td>Anomaly detection using incremental PCA (AD-IPCA)</td>
</tr>
</tbody>
</table>

Moreover, SOLMA’s algorithms could be extended to every production line of items made up of a single component, i.e., steel, wood, glass, whose quality is monitored continuously through sensors. Industry representatives from various sectors such as food, healthcare, wood, i.e. Unilever, XIM and Sodra, have approached partners and enquired about SOLMA’s functionalities. This proves its potential as an innovator of European smart manufacturing.
3.2 Evaluation of classification algorithms: flatness accuracy use case

In this section, we judge the performance of some SOLMA algorithms developed in PROTEUS when applied to the AMIII data. It should be stressed from the outset that these performance indicators are as much, if not more, about the predictive power of the data as they are the performance of the algorithm. None the less, these metrics are useful for comparing different algorithms applied to the same data in a controlled experiment. In the AMIII use case and in machine learning more generally, these enable analysts to perform model selection, that is, deciding on which algorithm works bests for different use cases.

3.2.1 WAPA Algorithms Applied to Flatness Positions on a Coil

The use case considered herein is the classification of points on the surface of a steel coil as flat or not. In the AMIII factory, if a steel coil exceeds 60 I-Units (in the middle zone which discards the head and tail parts of the coil which are not representative of flatness) it is deemed to not be flat and thus defective for AMIII purposes. This value was thus used as a threshold for classifying sections of the steel to build data for classification with supervised machine learning.

The full details are reported in D4.4, but it suffices to say here that the algorithms used for classification are online weighted averaging passive-aggressive (WAPA) algorithms. More precisely, two variants of this model were explored, one in which a smoothing parameter is adjusted and tuned, which is referred to as DWAPA and one in which the smoothing parameter is exclusively set to 1, which is referred to simply as a distributed passive-aggressive (DPA) algorithm. Both these classes of model were further generalised by experimenting with the inclusion of another penalty term that was either linear (I) or quadratic (II).

As this was a classification problem, the appropriate evaluation metrics are the accuracy, precision and recall. A comparison of the accuracies for all model variants is shown below. The higher accuracy metric in the table shows that the DWAPA models outperform the DPA models when predicting flatness, based on an analysis of 298 individual coils.

<table>
<thead>
<tr>
<th>Data</th>
<th>DWAPA</th>
<th>DWAPA-I</th>
<th>DWAPA-II</th>
<th>DPA</th>
<th>DPA-I</th>
<th>DPA-II</th>
</tr>
</thead>
<tbody>
<tr>
<td>298 coils</td>
<td>99.10</td>
<td>99.10</td>
<td>99.10</td>
<td>98.56</td>
<td>98.56</td>
<td>98.56</td>
</tr>
</tbody>
</table>

The accuracy metric alone however, is not an ideal indicator of performance owing to the significant class imbalance in the dataset. In fact, there were 22326 flat positions while there were only 213 non-flat positions, with the majority of them near the beginning of the coil. Consequently, an algorithm that simply predicted all positions as flat would return an accuracy of 99.055%. Nonetheless, the DWAPA models showed an improvement by predicting 22338 positions correctly giving an accuracy of 99.10%, indicating that it recognised some of the 213 non-flat positions. To explain the KPI assessment in a more useful format we give the confusion matrix for this experiment below, where 1 indicates flat and -1 indicates non-flat:

```
Predicted: 1 (flat)          | Label: 1 (flat) | Label: -1 (non-flat) |
----------------------------|-----------------|----------------------|
Predicted: 1 (flat)          | 22317           | 192                  |
Predicted: -1 (non-flat)     | 9               | 21                   |
```
This means that the algorithm correctly predicted 22317 points as flat and 21 points as non-flat. From the confusion matrix, we can also report the following conventional metrics for classification problems:

<table>
<thead>
<tr>
<th>True Positive Rate (Recall)</th>
<th>True Negative Rate</th>
<th>False Positive Rate</th>
<th>False Negative Rate</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.999596883</td>
<td>0.098591549</td>
<td>0.90140845</td>
<td>0.000403117</td>
<td>0.525825397</td>
</tr>
</tbody>
</table>

From this table we see that although the true positive rate is high, the true negative rate is actually quite low and only around 10% of the 192 actual non-flat positions were predicted as non-flat. Similarly, the precision is not as high as we would hope owing to the high false positive rate. This is due to the high-class imbalance in the data set.

### 3.2.2 Large Scale Experiment for Classifying Coils Using ONLMSR

An additional large-scale experiment was performed in which AMIII steel coils were classified as faulty (defective) or non-faulty using information provided by AMIII. This experiment used all suitable coils available which corresponded to 9450 coils of different lengths, approximately 1.1 million data points. This demonstrates PROTEUS ability to handle large volumes of high velocity data. The algorithm used is the Online Normalised Least Mean Squares regression (ONLMSR) discussed in D2.11. The confusion matrix for the experiment is shown below, where the non-faulty class is taken as positive and the faulty as negative:

<table>
<thead>
<tr>
<th></th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Non-Faulty (0)</td>
</tr>
<tr>
<td>Predicted</td>
<td></td>
</tr>
<tr>
<td>Non-Faulty (0)</td>
<td>4150</td>
</tr>
<tr>
<td>Faulty (1)</td>
<td>2</td>
</tr>
</tbody>
</table>

This means that the algorithms correctly predicted 4150 non-faulty coils and 4609 faulty coils. The number of false negatives was extremely small: out of 9450 total coils, only 2 non-faulty coils were misclassified as faulty. The number of false positives was higher: out of 9450 total coils, 689 faulty coils were misclassified as non-faulty.

<table>
<thead>
<tr>
<th>True Positive Rate (Recall)</th>
<th>True Negative Rate</th>
<th>False Positive Rate</th>
<th>False Negative Rate</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9995</td>
<td>0.87</td>
<td>0.13</td>
<td>0.0005</td>
<td>0.8576</td>
</tr>
</tbody>
</table>

The high true positive and true negative rates achieved here with the AMIII data show that significant predictive power has been achieved by PROTEUS and that is something that can guide mitigation and intervention strategies in the factory leading to significant cost savings. We discuss how the confusion matrix in this section can be used to calculate cost savings in the factory in Section 5.

### 3.3 Evaluation of regression algorithms: flatness regression prediction use case

In this section we evaluate some of SOLMA’s algorithms for regression when applied to the AMIII data. Two algorithms are compared. The first is Least Absolute Shrinkage and Selection Operator (LASSO), see D4.5 for details, and the other is Online Normalised Least Mean Squares regression
(ONLMSR) and is discussed in D2.11. As this is a regression problem, as mentioned, the appropriate evaluation metrics are the $R^2$, RMSE and MAE. These indicate the extent to which values predict by the models above correspond to the actual values in the data. In particular, $R^2$ provides information on the amount of variation in the data that is explained by the model and thus sheds light on the predictive strength of the model. $R^2$ ranges from 0 to 1: the closer it to 1 the better. The table below shows that the ONLMSR algorithm performs quite well accounting for 40% of the variations in the data. RMSE and MAE are two measures for the typical difference between predicted and observed values: the smaller these are the better. As we can observe in the table below, compared to LASSO (RMSE= 24.48; MAE= 9.51), the ONLMSR algorithm is characterised by smaller measures of RMSE (20.58) and MAE (4.37). Hence, ONLMSR is a better fit for the AMIII’s data.

ONLMSR provides significant improvement in terms of accuracy and time in comparison to LASSO. LASSO time complexity is $O(n^2)$. In contrast, ONLMSR is $O(n)$.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>RMSE</th>
<th>$R^2$</th>
<th>MAE</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>LASSO</td>
<td>24.48</td>
<td>0.09</td>
<td>9.51</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>ONLMSR</td>
<td>20.58</td>
<td>0.4</td>
<td>4.37</td>
<td>$O(n)$</td>
</tr>
</tbody>
</table>

### 3.4 Summary

In this section we have seen how PROTEUS’ machine learning library SOLMA has added a number of useful algorithms for regression and classification with batch and streaming data. These algorithms were not available in Flink-ML, for example. In the AMIII use case, these new tools for predictive analytics enable the early stage prediction of steel coils as defective, prompting rapid cost saving intervention strategies. As different algorithms are better suited to different problems, it was shown that optimising conventional machine learning performance metrics such, e.g. minimising false positive rates and false negative rates or maximising $R^2$, can help select which algorithm is best suited to the task at hand. In Section 5 it will also be shown how these same metrics can be used to estimate cost savings in the AMIII factory.
4. PROTEUS’ Data Visualisation Platform PROTEIC

In this section we will evaluate the impact of PROTEUS’ visualisation tools in the broad context of smart manufacturing. Visualisation tools are the principle enablers of analytics-driven decision making in a factory, constituting the core value of smart-manufacturing.

Recent research on visualisation tools for smart manufacturing highlights the lack of tools that strike a balance between subject matter expertise and technical/AI know-how (Moayen and Iskander 2017). On one hand, there are mainly-statistically driven “one-size-fit all” visualisation tools, in which minimal industrial expertise is included (ibid.). On the other hand, there are non-statistical tools, mainly based upon subject matter expertise which use minimal statistical inputs.

We evaluate PROTEUS’ visualisation tool considering the extent to which it represents a balanced solution between subject matter expertise and technical/AI know-how. To do this, we carried out a qualitative assessment of PROTEUS-driven technical and industrial integration, considering the PROTEUS data visualisation dashboard, PROTEIC. This qualitative assessment is supported by a KPI measuring the number and types of plots that PROTEIC can generate (Section 4.1), the number of functions e.g. zoom and notifications the plots have (Section 4.2), and a factory expert’s user feedback on PROTEIC (Section 4.3).

4.1 Plot types in PROTEIC

In this section, we present the Plots that PROTEIC dashboard can generate. The number of plots variations is an important KPI for any visualisation platform as there is so often an end user need to visualise data in myriad ways. The PROTEIC dashboard can generate 28 plot types including: annotation plots; area plots; bar charts; gauge charts; heatmaps; line charts; scatterplots; streamgraphs; network graphs; sunbursts, swim lanes; parallel coordinates.

Heatmaps are useful as they can show the different level of flatness of a coil; parallel coordinates allow multi-variable comparisons across coils; scatterplots visualise the strength of the association between a coil’s variable and its flatness level; gauge charts allow factory operators to monitor the coil’s flatness level live and identify when it reaches problematic levels; finally, line charts allow factory operators to analyse a particular part of the streaming signal in depth. Below we show some of these plots.

![Figure 3. Annotation Plot & Gauge Chart](image-url)
Figure 4. Area Plots

Figure 5. Heat Maps

Figure 6. Line Charts
Figure 7. Scatterplots

Figure 8. Streamgraphs

Figure 9. Network Graphs & Swim lane
4.2 Interactive functionality in PROTEIC

The different interactive functions available in PROTEIC include zoom, pause, an alarm triggering system and window signaling errors, and a save and retrieve function.

As far as the zoom function is concerned, a zoom on the desired area of a plot is provided in many of the PROTEIC plots. The factory operator simply has to mark the region of the plot that he/she wants to zoom in and the screen adjusts instantaneously and automatically.

This zoom function is useful when used in conjunction with the pause function, as it allows factory operators to inspect an area of the coil in detail, without the need to see the time series moving or have an axis scale that is too imprecise. The pause function is desirable for online monitoring, especially when the frequency rate is high: this can hinder the visualisation of the trends that are of interest to the expert. After the expert’s close inspection and validation, the normal online representation can be easily reactivated.
With regards to the alarm triggering system and window signalling errors, partners developed a visualisation tool capable of monitoring many signals coming from the plant sensors and triggering alarms when the values of the signals are exceeding a given boundary. In the last version of the tool, the safety zone adjusts automatically as a new point is included: the tool computes the interval in each point and then connects the different points. Figure 15 below shows the tool. The safety zone is represented by the grey rectangle; the red dots represent the “problematic” exits from the exit zones. The visualisation tool uses the window signalling error function to put red dots on the extreme values to signal anomalies to the user. This is very visual and can help the operators focus on these risky points.

Finally, regarding the save and retrieve function, coil visualisations can be saved and retrieved after a given time frame. This function provides the powerful capability of crosschecking coils from different time periods, different castings or different steel grades, or simply to re-visualise a coil which was problematic and compare it with another coil with a “normal” behaviour.

4.3 Factory Expert’s end-user evaluation of PROTEIC
AMIII has involved a high-level plant expert in providing feedback for the design and validation of TREE’s PROTEIC visualisation tool. This is an achievement showing progress towards a well-balanced integration of industrial know-how and technical expertise: as plant experts or operators (with other priorities and daily duties) are rarely involved in research activities, such as PROTEUS. This shows how PROTEIC successfully fulfilled end users’ needs.

Below the feedback of the plant expert following implementation of PROTEIC in the AMIII’s facilities.

4.3.1 User’s positive feedback on PROTEIC features

The expert highlighted that the tool is really intuitive and self-explanatory. Additionally, the expert considered the links to the PROTEUS website very useful as it provides an overview of the Project and the technologies used underneath. The expert thought that having a streaming monitoring of certain crucial variables, such as that of PROTEIC, would be a remarkable improvement to their current situation, where operators can only have historical visualisations of past coils. Hence, the expert was really interested in PROTEIC. In fact, the expert highlighted once again that the tool is so easy to use that they would be in favour of installing it at the plant.

The factory operator’s generally positive feedback on PROTEIC is due to the fact that AMIII worked closely with TREE, advising on the variables and the visual components to include in PROTEUS’ visualisation dashboard. For instance, following AMIII’s recommendations, TREE included into PROTEIC a heating map, an alarm triggering system, and a window signalling errors. The final version of PROTEIC is thus the result of continuous interactions between TREE and AMIII and reflects the industrial expertise of the latter.

4.3.2 User’s suggestions for the improvement of PROTEIC

The expert has shared their interest in introducing other kinds of models to the tool. In fact, the expert mentioned that it would be nice to have a drop-down list with the algorithms that can be selected to visualise the graph. The expert also mentioned that it would be great to be able to move the graph plots in the web just by dragging them around. Thus, variables could be ordered wherever the user prefers them. The expert mentioned that it would be useful to obtain the value of the outlier as you are approaching the cursor. Moreover, they would add the possibility of adding some notes somewhere. This is because although there are many outliers in a certain region in a given coil, not all of them are as critical or just a reminder to check this coil afterwards.

4.4 Summary

PROTEIC brings together statistical/ AI techniques for anomaly detection, i.e., the algorithms of the SOLMA library whose results are displayed in PROTEIC, and AMIII’s subject matter expertise. PROTEUS, through its open-source nature, has thus contributed to advanced knowledge within the sector, providing a freely available template, PROTEIC, which other manufacturers can deploy in various industries.

As a matter of fact, because of the way in which PROTEIC gathers coil data and displays coil variables, any manufacturer producing coils can potentially reuse this dashboard and tap into PROTEUS knowledge base. PROTEIC thus provides the basis for further research on innovative
ways to integrate AI expertise and industrial know-how in the smart manufacturing to fulfil end users’ needs. This could be greatly beneficial to European Small and Medium-sized Enterprises (SME) working in the manufacturing sector. These accounted for “58% of total employment and 42% total added-value in the sector in 2016” (EC 2017, p.15).

5. Industrial Value KPI

Partners developed an industrial value KPI to assess the extent to which PROTEUS provided added-value to ArcelorMittal. This KPI can be used by manufacturers outside the steel industry as well: those who manufacture products which may be defective but can be still marketed can refer to it. Additionally, this KPI helps decision making on the adoption of smart-manufacturing. It may not be worth it for manufacturers with relatively low production costs to install sensors and use predictive machine learning tools, such as PROTEUS, to help production. Conversely, smart-manufacturing may be profitable for manufacturers having high production costs. Such a KPI could be used to support small and medium enterprises (SMEs) characterised by large variations in production costs.

5.1 Process improvement KPI

In this section, we report on an important process KPI which determines the economic benefit a predictive classifier would introduce to the steel manufacturing process, in comparison to the default scenario in which all coils are assumed to be high quality, flat and thus not defective. Although we do not install PROTEUS in the AMIII factory in this project, it is important to note that the metric for evaluating success is available for future reference. This analysis would be applied to the true positives and false positives in the confusion matrix presented in Section 3.3.2 for example, although it is not possible to yet calculate economic savings without knowledge of processing costs in the factory. Additionally, it could be extended outside the steel industry to the smart manufacturing sector more broadly.

The predictive classifier aims to identify non-flat coils early on so that they can be downgraded and assigned to a new use earlier than they would be otherwise. This will save the company the extra processing and handling costs associated with trying to mistakenly market defective coils as high quality only for them to be sent back. The algorithms defined in D2.11 would perform this function in a fully deployed version of PROTEUS.

If a coil is genuinely flat and the classifier predicts it to be so (true positive, TP), it would be handled (i.e., further processed, marketed, shipped and sold) as if it is high quality and the company will make the monetary value $V_{high}$. If a coil is genuinely non-flat and the classifier predicts it to be so (true negative, TN), it would be handled as if it is low quality and the company will make the monetary value $V_{low}$, where $V_{high} > V_{low}$.

If a coil is genuinely flat but the classifier predicts it to be non-flat (false negative, FN), it would be handled as if it is low quality and the company will make the monetary value $V_{low}$. If a coil is genuinely non-flat but the classifier predicts it to be flat (false positive, FP), it would be handled as if it is high quality only for it to be sent back, downgraded and resold as low quality, and thus the company will make the monetary value $V_{low} - V_{process}$, where $V_{process}$ is the monetary value associated with the wasted additional processing and mishandling of the coil. This last scenario is the worst possible outcome.
If the factory operates in full accordance with the predictions of the classifier, the total monetary value made, \( V_{\text{classifier}} \), will be:

\[
V_{\text{classifier}} = TP * V_{\text{high}} + TN * V_{\text{low}} + FN * V_{\text{low}} + FP * (V_{\text{low}} - V_{\text{process}})
\]

\[
= P * V_{\text{high}} + N * V_{\text{low}} - FN * \Delta V - FP * V_{\text{process}}
\]  

(1)

Where \( P = TP + FN \) is the total number of positive (flat) coils, \( N = TN + FP \) is the total number of negative (non-flat) coils and \( \Delta V = (V_{\text{high}} - V_{\text{low}}) \) is the monetary value difference between the high and low quality coils. It is evident that FP introduce a larger penalty in value than FN when \( V_{\text{process}} > \Delta V \). However, if the classifier worked perfectly such that \( FP = FN = 0 \), then the monetary value made would be:

\[
V_{\text{PerfectClassifier}} = P * V_{\text{high}} + N * V_{\text{low}}
\]  

(2)

Alternatively, if the classifier simply predicted every coil to be flat such that \( FP = N, FN = 0 \), then the monetary value made would be:

\[
V_{\text{Pred.AllFlat}} = P * V_{\text{high}} + N * (V_{\text{low}} - V_{\text{process}})
\]  

(3)

The condition set out in equation (3) represents the default current operation mode of the factory in that the absence of a classifier is equivalent to the assumption that all coils are flat at the section of the operation and so they continue to be processed accordingly right up until the final checks prior to dispatching to customers. It should be noted though, that at the very final checks for flatness and quality the company will operate a conservative approach when uncertain to the flatness so as to avoid sending out coils of insufficient quality but that is a separate judgement made much later to the early judgement made by the classifier that is the focus of this analysis.

Consequently, it is possible to establish an upper bound on the value added by machine learning in the factory by comparing a classifier that works perfectly to the assumption that all coils are flat:

\[
V_{\text{PerfectClassifier}} - V_{\text{Pred.AllFlat}} = N * V_{\text{process}}
\]  

(4)

However, the value added by a general, possibly imperfect, classifier relative to one which predicts all coils to be flat, would be:

\[
V_{\text{Classifier}} - V_{\text{Pred. All Flat}} = TN * V_{\text{process}} - FN * \Delta V
\]

\[
= N * V_{\text{process}} - FP * V_{\text{process}} - FN * \Delta V
\]  

(5)

Thus, a general classifier adds value if and only if:

\[
\frac{FN}{TN} < \frac{V_{\text{process}}}{\Delta V}
\]  

(6)

Or alternatively in terms of the false omission rate \( F.O.R \), value is added if and only if:

\[
F.O.R = \frac{FN}{TN + FN} < \frac{1}{1 + \frac{\Delta V}{V_{\text{process}}}}
\]  

(7)

Finally, it is worth noting that it is often possible for a classifier to reduce FP at the expense of FN, or vice versa, by tuning a decision threshold parameter \( \theta \). In which case, to minimise costs and maximise value one defines the optimal decision threshold parameter as:

\[
\theta_{\text{optimal}} = \arg\min_{\theta} FP(\theta) * V_{\text{process}} + FN(\theta) * \Delta V
\]  

(8)
5.2 Summary
The KPI above can be used by manufacturers outside the steel industry: those who manufacture products which may be defective but can be still marketed can refer to it. This KPI shows that the creating added value through a machine learning classifier depends not only on the performance of the classifier itself, i.e., how good it is at detecting defective products (the classifier will detect actually defective products as well as high-quality products erroneously classified as defective), but on producers’ costs and returns.

This KPI helps decision-making on the adoption of machine learning classifiers in smart-manufacturing. It may not be worth it for manufacturers with relatively low production costs to install sensors and use predictive machine learning tools, such as those used for the PROTEUS solution, to help production. Conversely, machine-learning based smart-manufacturing may be profitable for manufacturers having high production costs.

Hence, this KPI represents a useful contribution of PROTEUS to smart manufacturing providing producers with the necessary tools to make decision on investments in smart manufacturing. In addition, this KPI adds value to the PROTEUS technical components described in this deliverable. Through, this KPI, manufacturers could in fact decide whether it is worth to invest in internal R&D and build upon the open source components of PROTEUS, i.e., LARA, SOLMA and PROTEIC, in order to enhance their smart-manufacturing capabilities through predictive analytics.
6. Barriers to Impact

In this section we deal with the obstacles faced during the PROTEUS project and how the project pivoted to minimise their effect on the projects impact. We start by presenting the issues with the AMIII’s data (Section 6.1) and continue by addressing issues with the PROTEUS factory implementation (Section 6.2), as well as some comments on the departure of some of the PROTEUS partners (Section 6.3).

6.1 Issues with the data

Issues with the data were recorded by partners in a log of data issues which listed all the anomalies with the data. These include, but were not limited to:

- faulty variable dimensionality with some variables being one-dimensional while they were expected to be two-dimensional
- an unexpected mix of 1D and 2D even though they were expected to be 1D
- anomalous repetition of variables’ values
- non-existing variables which were originally listed in the variable’s mapping
- variable duplications

Most notably though, a major obstacle stemmed from the fact that the variables were anonymised and so the physical nature of the variable was unknown. This inhibited intuitive insights from guiding the analysis. Moreover, position x on the input variables is very different from that position x on the output (target) variables. These issues could have resulted from faulty sensors in the Hot Strip Mill production line.

Furthermore, it should be noted that the data generation process was particularly onerous and complex due to the multifaceted nature of the data, made up of both data-at-stream and data-in-motion, and its proprietary format required a great deal of pre-processing before access could be granted to partners.

It was also evident that there was little obvious predictive power in the variable from which to predict targets such as flatness. This had a particular impact on results in Section 3. Despite this, the project did work hard to extract as much insight from the data as possible and these efforts ultimately resulted in reasonable predictions for flatness as reported in Section 3.

6.2 Issues with factory implementation

Unfortunately, the implementation of the technologies and machine learning models developed by PROTEUS in the AMIII’s facility is not feasible during the course of the project due to difficulty in getting agreement from AMIIIs management and legal teams. PROTEUS has had to re-think the pilot methodology to evaluate the utility of the PROTEUS solution. It should be noted that AMII, like most large manufacturing enterprises, has a conservative tendency in terms of applying new models in the plant, unless they have been widely tested/validated (PROTUES is still a prototype). This does not mean that all the work developed within the PROTEUS project is not going to be used; simply, it may take some time to implement the PROTEUS prototype online in plant. For these reasons, partners decided to evaluate the PROTEUS integrated prototype by gathering feedback from a senior expert within the plant (see Section 3.3 above) in relation to the data visualisation components. In addition, some important KPIs and impact theory has been derived ready for evaluation in a factory setting at a later date (see Section 5.1 above).
6.3 Departure of PROTEUS Partners

The unfortunate premature departure of two of PROTEUS partners (TREE and LMDP) meant that the consortium lacked some of the technical capabilities, skill and hardware components required to improve, implement and evaluate further some of the technological components of PROTEUS. This was the case for the technical evaluation of the PROTEIC visualisation dashboard, and the installation of the software components needed to run the hybrid computation system. That said other partners stepped forward and took on more responsibility and the knock-on effects were mitigated as much as possible.

6.4 Summary

In this section we have discussed some of the impediments faced during the PROTEUS project including issues with the quality and predictive power of the data in the AMII use case and the understandable reluctance of AMIII to implement the system in the factory within the timeframe of the project. In addition, the departure of partners TREE and LMDP slowed progress but the effects of this were minimised as other partners stepped forward to take on more responsibility.
7. Conclusion

This deliverable performed an impact assessment of PROTEUS. PROTEUS components KPIs provided information on the project’s contributions to machine learning as well as smart manufacturing in Europe, as demonstrated in an industrial use case with an ArcelolorMittal steel coil factory.

Some specific results include:

- While latency measurements of 1 second are normally acceptable, the PROTEUS hybrid computation engine for batch and streaming data processing delivers millisecond latencies, providing near real-time interactive analysis and defect alerts to workers on the AMIII factory floor.

- PROTEUS’ abstraction language, LARA, simplifies the definition and optimization of user defined functions for batch and streaming data analytics, reducing the learning curve and removing barriers to mastering online machine learning techniques.

- PROTEUS’ machine learning library SOLMA provides algorithms for regression and classification with batch and streaming data which were not available in existing batch processing libraries such as Flink-ML. In the AMIII use case, these new tools for predictive analytics enable the early stage prediction of steel coils as defective, prompting rapid cost saving intervention strategies.

- PROTEUS’ powerful data visualisation platform for streaming data, PROTEIC, can generate 28 plot types supported by multiple interactive features that put big data insights at the disposal of AMIII factory workers.

Taken together these impacts represent powerful innovations to the EU’s machine learning, big data and smart-manufacturing industries. PROTEUS will find application all industries characterised by requirements for near real-time interventions and analysis such as the gaming industry and PROTEUS has already attracted the attention of industry representatives from various sectors such as food, healthcare and wood, with Unilever, XIM and Sodra approaching partners to enquire about the functionality of PROTEUS’ streaming and batch machine learning library SOLMA.
References


PROTEUS (2018). *PROTEUS* (GitHub Repositories). Available at: https://github.com/PROTEUS-H2020
