H2020 – ICT-13-2018-2019

Machine Learning to Augment Shared Knowledge in Federated Privacy-Preserving Scenarios (MUSKETEER)

Grant No 824988

D6.1 Assessment Framework design and specification

September 19

Imprint

Contractual Date of Delivery to the EC: 30 September 2019

Author(s): Ángel Navia-Vázquez (UC3M), Jesús Cid-Sueiro (UC3M), Luis Muñoz-González (IMP)

Participant(s): Tree Technologies, IMP

Reviewer(s): Davide Dalle Carbonare (ENG), Susanna Bonura (ENG), Mathieu Sinn (IBM)

Project: Machine learning to augment shared knowledge in federated privacy-preserving scenarios (MUSKETEER)

Work package: WP6

Dissemination level: Public

Version: 1.5

Contacts: Angel Navia-Vázquez - angel.navia@uc3m.es

Website: www.MUSKETEER.eu

Legal disclaimer

The project Machine Learning to Augment Shared Knowledge in Federated Privacy-Preserving Scenarios (MUSKETEER) has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 824988. The sole responsibility for the content of this publication lies with the authors.

Copyright

Executive Summary

This deliverable (D6.1 Assessment Framework design and specification) is the result of the task T6.1 (Design and specification of the assessment framework) and describes the common evaluation framework that will be used to evaluate the behaviour and main performance indicators of the implemented algorithms. We have followed the Goal Question Metric method, and the main goals are to assess the scalability, computational efficiency, performance, security, and data value estimation capabilities of the proposed and implemented schemes.

Document History

Version	Date	Status	Author	Comment
1	31 July 2019	For internal review	Angel Navia-Vázquez, Jesús Cid-Sueiro	First draft
2	2 sept 2019, review inputs	For internal review	Susanna Bonura, Davide Dalle Carbonare	Update
3	9 sept 2019	For internal review	Angel Navia-Vázquez, Jesús Cid-Sueiro	Update
4	10 Sept 2019	For internal review	Luis Muñoz-González	Update
5	14 sept 2019	For internal review	Angel Navia-Vázquez	Update
6	Final Version			Update

Table of Contents

List of Figures

List of Acronyms and Abbreviations

ABBREVIATION	DEFINITION
AUC	Area Under (ROC) Curve
BMI	Body Mass Index
CA	Consortium Agreement
DOW	Description of Work
DV	Data Value
FS	Feature Selection
GA	Grant Agreement
GQM	Goal, Questions and Metrics
IDP	Industrial Data Platform
IDR	Intermediate Data Representation
MK	Master Key
ML	Machine Learning
MLA	Machine Learning Algorithm
MN	Master Node
PERT	Program evaluation and review technique
PHE	Partial Homomorphic Encryption
PK	Public Key
POM	Privacy Operation Mode
PP	Privacy Preserving
PPML	Privacy Preserving Machine Learning (a.k.a. Privacy Preserving Data Mining)
ROC	Receiver Operating Characteristics
SMC	Secure Multiparty Computing
SQL	Structured Query Language
TA	Task Alignment

1 Introduction

1.1 Purpose

The MUSKETEER project aims at building an Industrial Data Platform (IDP) such that different users can contribute with data to solve a given Machine Learning (ML) task without compromising the confidentiality of the data. The core component of the platform providing the capability of training ML models while preserving confidentiality in the data, is named as the ML library (comprising several ML Algorithms) developed under different Privacy Operation Modes (POMs). The deliverable D6.1 describes the assessment methodology that will be

used to characterize the behaviour and main performance indicators of the implemented algorithms in the Machine Learning library, from a technical point of view. The general context of the assessment and the other perspectives of analysis are described in Deliverable D2.3.

In this document we took inspiration from the Goal Question Metric (GQM) method [Solingen] to define the main common evaluation framework to be used to to assess the scalability, computational efficiency, performance, security, and data value estimation capabilities of the proposed and implemented ML schemes under the different Privacy Operation Modes (POMs), according to Tasks T6.2, T6.3 and T6.4 in WP6.
The GQM was also used for the pilot goals and KPIs definition (task T2.3) to be adopted for the assessment of the platform from the use-case perspectives in WP7, as described in D2.3. In this document, we will expand the definition of the assessment corresponding to the above mentioned tasks, which are described in D2.3 as goals G3.1, G3.2 and G4.1.

The evaluation experiments described in this document focus on the evaluation of the ML components from a general point of view, and therefore we will not use the datasets provided by the use cases. As described in Section 5, we will use standard open datasets, such that the experiments can easily be replicated by other researchers.

1.2 Related documents

As indicated in the PERT diagram below, the results from this deliverable will provide input to WP7 (User Cases), such that they can better decide which POM/algorithms are the most adequate to solve a given task. Although not shown in the PERT, the inputs are mainly defined by the project KPIs and objectives that concern the performance of the implemented Machine Learning algorithms, as well as other declared pertinent objectives and user specifications, as summarized in Section 3 below.

Deliverable D2.3 completely defines the MUSKETEER evaluation framework from several different points of view or perspectives: business, end users' cases, technical and market/business. That document provides a context for D6.1, which focuses on the technical assessment of the developed ML libraries under the different POMs.

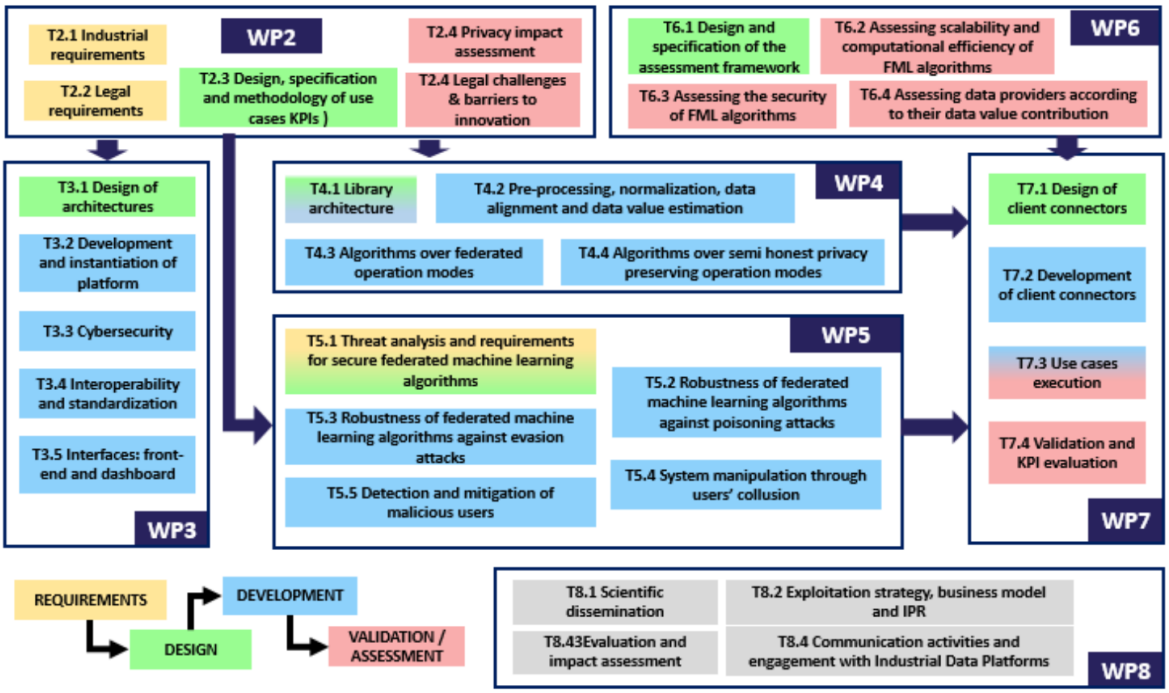

Figure 1 MUSKETEER's PERT diagram

1.3 Document structure

This document is structured as follows:

- The current section (Introduction), presents the purpose of the document, as well as the relationship with other WPs in the project.
- In Section 2 we describe the selected assessment methodology: Goal, Question, Metrics [Solingen].
- In Section 3 we briefly revisit some of the specific and pertinent objectives in the project, as well as the associated KPIs.
- In Section 4 we detail the full list of Goals, Questions and Metrics to be used during the assessment.
- In Section 5 we describe the selected datasets and their characteristics.
- In Section 6 we give some detail on the experiments to be carried out to fulfil the GQM methodology described in Section 4.
- In Section 7 we draw some preliminary conclusions.
- Section 8 collects the main references.

2 Assessment methodology

We will approach the assessment problem using the paradigm of Goals, Questions and Metrics (GQM) [Solingen], which is illustrated in the following Figure.

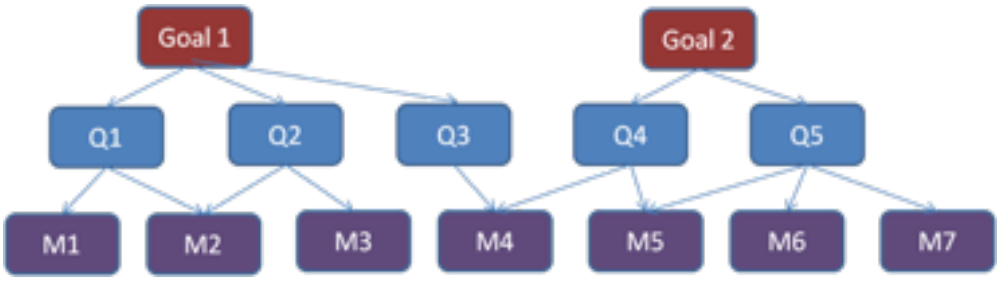

Figure 2 Goals, Questions and Metrics (GQM) paradigm example

The GQM methodology used in agile environments, allows for identifying and further refining a collection of explicit measurement goals. After the goal identification phase, one or more questions can be defined for every goal. Finally, one or metrics are described to answer those questions:

- Goals define what the project wants to improve;
- Questions refine each goal to a more quantifiable way;
- Metrics indicate the metrics required to answer each question.

In what follows, we provide a more detailed description of the specific formulation of the GQM methodology to be used here:

1. **Goals**: Specifying formal measurement goals is the first step in the GQM process. These goals have to be defined in an understandable way and with a clear structure. For this purpose, the template adopted in MUSKETEER (as declared in D2.3) supports the definition of measurement goals by specifying, FOR EACH GOAL:
   a. **Analysis**: the object under measurement
   b. **For the purpose of**: understanding, controlling, or improving the object
   c. **With respect to**: the quality focus of the object that the measurement focuses on
   d. **From the viewpoint of**: the people that measure the object. Two viewpoints were defined: 'technical' (i.e. technical partners involved in WP7 (T7.4): ENG, TREE, IMP, IDSA, KUL) and 'business' (i.e. end user partners involved in WP7 (T7.4): FCA, COMAU, B3D, HYGEIA)
   e. **In the context of**: the environment in which measurement takes place.

2. **Questions**: The second step is the definition of questions. Questions are refinements of goals to a more operational level, which is more suitable for interpretation. By answering the questions, one should be able to conclude whether a goal is reached.

3. **Metrics**: Once goals are refined into a list of questions, metrics should be defined that provide all the quantitative information to answer the questions in a satisfactory way. Therefore, metrics are a refinement of questions into a quantitative process and/or product measurement. After all these metrics have been measured, sufficient information should be available to answer the questions.

3 Project KPIs and objectives

In this section we briefly summarize the objectives and KPIs as reflected in the Grant Agreement that are pertinent to WP6 (assessment of Machine Learning algorithms, MLA) and this deliverable. We revisit some of them as they appear in the DOW [GrantAgreement], and we provide further clarification when needed. We have only included here those directly related to the assessment described in this deliverable, the full list can be found in D2.3 or in the DOW itself.

Objective 1. Machine Learning over a high variety of different privacy-preserving scenarios.

O1.1. Definition of several Privacy Operation Modes (POMs) to provide compliance with the legal and confidentiality restrictions of most industrial scenarios.

KPI: Distributed efficiency (speedup/number of users) superior to 0.8, while preserving privacy.

O1.2. Creating predictive models without directly exposing them to the data consumers (training data remains in the installations[1] of data providers).

KPI: The federated training will achieve comparable accuracy as the traditional local computing (decentralization will not affect the accuracy).

O1.3. Correct combination of different concepts of federated machine learning, differential privacy, homomorphic encryption, secure multiparty computation and distributed computing to improve the scalability of machine learning algorithms over every POM.

[1] Within the premises/organizational boundaries.

KPI: Faster than current SMC privacy-preserving alternatives[2] such as [PySyft] or [SecureML].

O1.4 Complete library of algorithms, having algorithms of different complexity levels.

KPI: Number of implemented algorithms. In supervised learning, it will contain at least a classification and regression alternative of linear models, kernel methods, trees and deep neural networks. It will also include one unsupervised technique for clustering and data decomposition.

O2. Providing robustness against external and internal threats

O2.1. Providing analysis and requirements for secure federated machine learning algorithms. We will consider vulnerabilities during training and at runtime, including the possibility of abuse from the users of the platform.

O2.2 The POMs will be designed to allow for a secure information exchange among the platform users.

KPI: Provide 8 working and robust[3] POM for the use cases.

O2.3. Including defensive mechanisms for the federated machine learning algorithms against poisoning and evasion attacks by detecting and mitigating the effect of such attacks.

KPI: The defensive mechanisms will be capable of reducing the effect of poisoning (for reasonable levels of data poisoning, e.g. less than 20% of poisoning in the training dataset) and evasion attacks, compared with unsecured federated machine learning algorithms.

[2] The platforms mentioned in the proposal are orientative. Since this is a very dynamic field, we will identify at the moment of the assessment which are the most suitable/available ones for comparison purposes.

[3] By "working and robust" we mean that those methods will have been benchmarked have been able to provide results on a variety of datasets/tasks.

O2.4. Providing mechanisms to detect and mitigate the effect of abusive users in the platform trying to compromise the learning process.

KPI: The defensive mechanisms will be capable of mitigating colluding users' attacks for reasonable scenarios (e.g. assuming a maximum of 20% of malicious users colluding to manipulate the platform), compared with unsecured federated machine learning algorithms.

O2.6. Developing a framework to test the security of federated machine learning against data poisoning, evasion attacks, and users' colluding attacks.

O3. Enhancement of the Data Economy

O3.1. Enhancing data providers to share their datasets thanks to the ability of creating predictive models without explicitly giving their datasets (using the FML concept), thus avoiding any possibility of personal/private information robbery[4].

KPI: Implementation of 8 different privacy operation modes to cover the different privacy needs given in industry.

O3.2. Allowing to measure the impact of every data owner on the accuracy of the predictive models, thus allowing to monetize their contributions as a function of their real data value.

KPI: Different data value estimation methods (one for every[5] POM).

O4. Providing a standardized and extensible architecture

[4] Maybe "Information leakage or theft" is a more adequate description.

[5] Some POMs may share the same techniques to estimate the data value. What is meant here is that, irrespectible of the POM that is being executed, a DV estimation method will be available.

O4.2. Allowing interoperability with Big Data frameworks by providing portability mechanisms to load and export the predictive models from/to other platforms.

KPI: MUSKETEER[6] will be capable to export the predictive models[7] to be loaded at least into Scikit-Learn, TensorFlow and Apache Spark

4 Goals, Questions and Metrics for ML assessment

The following goals have been identified from the corresponding task goals in WP6. Although we have renumbered them with respect to the analysis carried out in D2.3, the first three ones have a direct relationship with goals G3.1, G3.2 and G4.1, as defined in D2.3:

Goal 1: Assessing performance, scalability and computational efficiency of MLAs (G1)
Goal 2: Assessing the security of MLAs (G2)
Goal 3: Assessing data value extraction and monetization strategies (G3)

In these goals, the assessment will be carried out for every feasible combination of algorithm and POM. In what follows, we further refine every goal into sub-goals and provide their respective hierarchy of questions and metrics.

4.1 Goal 1 (G1): Assessing performance, scalability and computational efficiency

The objective here is to evaluate the developed MLAs to determine if their behaviour is as expected, mainly from the point of view of performance (Do they provide models as competitive as those obtained with other libraries?), scalability (Does the computational cost of the training procedure grow in a controlled manner such that the methods can be applied to an increasing amount of data or input features?) and computational efficiency (Do we need to provide an excessive amount of memory, computational or communication resources for the library to work?)

[6] By "MUSKETEER" we mean here the "Machine Learning Library at Musketeer".

[7] Using a standard model exportation format, to be selected.

G1	
Analyse	Machine Learning Algorithms (MLA)
For the purpose of	Evaluating
With respect to	Performance in distributed learning scenarios, scalability, computational overload, communication requirements, storage requirements, etc
From the viewpoint of	Technical perspective
In the context of	WP6: Assessment of data quality, scalability, computational efficiency and security

To facilitate the identification of questions, we further split Goal 1 into several sub-goals, namely:

G1: Assessing performance, scalability and computational efficiency of MLA

G1.1: Assessing the performance of MLA
G1.2: Assessing the reliability of MLA
G1.3: Assessing the scalability of MLA
G1.4: Assessing the computational efficiency of MLA

4.1.1 Questions for G1

In what follows we will define a collection of questions that better determine the Goal 1.

Identifier	Questions
G1.1_Q1	Is the ML library able to provide a data clustering in such a way that objects in the same group are more similar to each other than to those in other groups?
G1.1_Q2	Given a dataset, is the ML library able to provide predictions for unseen values of related non-categorical (real valued) variables?
G1.1_Q3	Given a training dataset, is the ML library able to provide predictions of the class of each data, according to some related categorical variable?
G1.1_Q4	Is the ML library able to provide correlation values among the features in data?
G1.1_Q5	Is the feature selection or extraction algorithm implemented in the ML library able to identify a relevant subset of features?

G1.2_Q1	Does each ML algorithm give comparable output working on the same data and in the same conditions in different sessions (reliability)?
G1.3_Q1	Does the training algorithm scale up when the dimension of the application scenario grows in terms of the amount of data?
G1.3_Q2	Does the training algorithm scale up when the dimension of the application scenario grows in terms of the amount of users (data providers)?
G1.3_Q3	Does the training algorithm scale up when the dimension of the application scenario grows in terms of the amount of input features?
G1.4_Q1	Are the MLAs faster than their counterparts in competing libraries?
G1.4_Q2	Are the transmission costs reasonable?
G1.4_Q3	Is the memory usage during training reasonable?

4.1.2 Metrics for G1

We define here the metrics and benchmarks for questions in Goal 1.

Identifier	KPI	Format	Method of collection and measurement	Benchmark
G1.1_Q1_M1	Unsupervised clustering metrics: silhouette coefficient, Calinksi-Harabasz index, Davier-Boulding index, contingency matrix.	Double	Test ML models: Clustering analysis with *n* groups. See experiment G1_E1.	We use as benchmark the performance obtained by a standard centralized library (Scikit-Learn, for instance) solving the same task. The assessment will be positive if both results do not differ by more than 5%.
G1.1_Q1_M2	Ground-truth based scores: adjusted Rand index, mutual information (MI), adjusted MI, normalized MI, homogeneity, completeness, v-measure	Double	Test ML models: Clustering analysis with *n* groups. See experiment G1_E1.	We use as benchmark the performance obtained by a standard centralized library (Scikit-Learn, for instance) solving the same task. The test will be positive if both results do not differ by more than 5%.

G1.1_Q2_M1	Mean squared error of the difference between the predicted value and the real value of the target variable	Double	Test ML models. See experiment G1_E2.	We use as benchmark the performance obtained by a standard centralized library (Scikit-Learn, for instance) solving the same task. The test will be positive if both results do not differ by more than 5%.
G1.1_Q3_M1	Standard classification metrics: Accuracy, AUC, Precision-Recall, Sensitivity-Specificity, etc	Double in [0, 1] or %	Test ML models. See experiment G1_E3.	We use as benchmark the performance obtained by a standard centralized library (Scikit-Learn, for instance) solving the same task. The test will be positive if both results do not differ by more than 5%.
G1.1_Q4_M1	Mean Square Error between estimated correlation values and reference values	Double	Test ML models: Compute correlations. See experiment G1_E4.	We use as benchmark the performance obtained by a standard centralized library (Scikit-Learn, for instance) solving the same task. The test will be positive if both results do not differ by more than 5%.
G1.1_Q5_M1	Performance loss or gain with respect to a reference feature set.	Double	Test ML models. See experiment G1_E5.	We use as benchmark the performance obtained by a standard centralized library (Scikit-Learn, for instance) solving the same task. The test will be positive if both results do not differ by more than 5%.
G1.1_Q5_M2	Number of coincident relevant features with respect to a reference feature set.	Integer	Test ML models. See experiment G1_E5.	We use as benchmark the performance obtained by a standard centralized library (Scikit-Learn, for instance) solving the same task. The test will be positive if both results do not differ by more than 5%.
G1.2_Q1_M1	Average of the standard deviation from the average	Double	Test ML models. See experiments G1_E1, G1_E2, G1_E3, G1_E4,	We consider the assessment positive if the standard deviation is low-

	of the normalized outputs calculated on same inputs in different sessions		G1_E5.	er than 5% of the mean value.
G1.2_Q1_M2	Statistics of performance metrics on same inputs in different sessions	Double	Test ML models. See experiments G1_E1, G1_E2, G1_E3, G1_E4, G1_E5.	We consider the assessment positive if the metrics are within a 5% interval of the mean value.
G1.3_Q1_M1	Training time vs data size.	Double	Test ML models. See experiments G1_E1, G1_E2, G1_E3, G1_E4. We estimate a regression model over the temporal trend and obtain the trend profile: coefficient and exponent of a linear, exponential or power-law model of metric vs problem size [Goldsmith, 2007].	We consider the assessment positive if the trend profile is less than 2.
G1.3_Q2_M1	Training time vs number of users	Double	Test ML models. See experiments G1_E1, G1_E2, G1_E3, G1_E4. We estimate a regression model over the temporal trend and obtain the trend profile: coefficient and exponent of a linear, exponential or power-law model of metric vs problem size [Goldsmith, 2007].	We consider the assessment positive if the trend profile is less than 2.
G1.3_Q3_M1	Training time vs number of features.	Double	Test ML models. See experiments G1_E1, G1_E2, G1_E3, G1_E4. We estimate a regression model over the temporal trend and obtain the trend profile: coefficient and exponent of a linear, exponential or power-	We consider the assessment positive if the trend profile is less than 2.

			law model of metric vs problem size [Goldsmith, 2007].	
G1.4_Q1_M1	Training time	Double	Test ML models. See experiments G1_E1, G1_E2, G1_E3, G1_E4.	We use as benchmark the training time in a competing platform. The assessment is positive if MUSKETEER is faster than the competing platforms.
G1.4_Q2_M1	Amount of information transmitted.	Double	Test ML models. See experiments G1_E1, G1_E2, G1_E3, G1_E4. We compute the transmission ratio: information transmitted (bytes) / size of the training dataset (bytes). If encrypted data is used, the reference value will be computed with respect to the encrypted data.	The assessment is positive if the transmission ratio is less than 10 (same order of magnitude).
G1.4_Q2_M2	Time dedicated to data transmission.	Double	Test ML models. See experiments G1_E1, G1_E2, G1_E3, G1_E4. We compute the transmission dedication: fraction of time used for transmission with respect to the total training time.	The assessment is positive if the transmission dedication is less than 0.5 on an experimental setup where master and worker nodes are run in the same machine for better control of the used processors but they communicate through a communication service located in the IBM cloud.
G1.4_Q3_M1	Amount of memory used during training.	Double	Test ML models. See experiments G1_E1, G1_E2, G1_E3, G1_E4. We estimate the storage ratio: total information stored in master and workers on memory (bytes) / size	The assessment is positive if the storage ratio is less than 10 (same order of magnitude).

			of the training dataset (bytes). If encrypted data is used, the reference value will be computed with respect to the encrypted data.	

4.2 Goal 2 (G2): Assessing the security

In this section we will describe the specific Goals, Questions and Metrics defined to evaluate the security of the MLAs.

G2	
Analyse	**Security**
For the purpose of	Evaluating
With respect to	Robustness to attacks (poisoning, evasion, users' collusion)
From the viewpoint of	Technical perspective
In the context of	WP5 (Security and Trustworthiness of Federated Machine Learning Algorithms) and WP6 (Assessment of data quality, scalability, computational efficiency and security)

4.2.1 Questions for G2

In what follows we will define a collection of questions that better determine the Goal 2.

Identifier	Questions
G2_Q1	Is the training model robust to poisoning attacks?

G2_Q2	How robust is the trained model against evasion attacks?
G2_Q3	Is the training model robust to user's collusion?

4.2.2 Metrics for G2

We define here the metrics and benchmarks for questions in Goal 2.

Identifier	KPI	Format	Method of collection and measurement	Benchmark
G2_Q1_M1	Loss in performance	Double	Evaluate the difference in the test performance of the model evaluated on a clean dataset and on a poisoned dataset (considering different levels of poisoning up to 20%). See experiment G2_E1.	See the clarification notes below.
G2_Q1_M2	Effectiveness of poisoning attack evaluated on a set of target data points.	Double	Evaluate the performance of the model trained on a poisoned dataset (considering different levels of poisoning up to 20%) on a set of data points previously defined as the target for the attack. See experiment G2_E2.	See the clarification notes below.
G2_Q2_M1	Percentage of successful evasion attacks	Double	Number of successful attacks over the total number of attempts evaluated on a set of test data points. See experiment G2_E3.	See the clarification notes below.

G2_Q2_M2	Average minimum perturbation required to craft successful evasion attacks	Double	For a set of test data points, evaluate the average minimum perturbation needed to produce an error in the trained model. See experiment G2_E4.	See the clarification notes below.
G2_Q3_M1	Percentage of malicious or faulty users successfully detected	Double	Number of malicious/faulty users detected over the total number of malicious/users present during the training of the model. See experiment G2_E5.	See the clarification notes below.
G2_Q3_M2	Percentage of benign users incorrectly detected as malicious	Double	Number of benign users detected as malicious over the total number of benign users present during the training of the model. See experiment G2_E6.	See the clarification notes below.

Clarifications on the metrics for G2:

- ***Performance*** in metrics G2_Q1_M1 and G2_Q1_M2 can be measured in different ways depending on the tasks or learning algorithms used. For example, in classification datasets, performance typically refers to the classification error (i.e. number of data points incorrectly classified over the total number of data points evaluated).
- Metric G2_Q1_M1 allows to evaluate the robustness of the training algorithm to **indiscriminate attacks**, i.e. those that aim to degrade the overall performance of the system. Thus, evaluating the loss in performance compared to the case where we train the model on a similar clean dataset provides an indicator of robustness against data poisoning.
- In some cases, attackers may **target** their attacks to **specific subsets of inputs**. To measure the robustness of the algorithms against these attack scenarios, metric

G2_Q1_M2 evaluates the robustness of the learning algorithms to targeted poisoning attacks.

- The metric in ***G2_Q2_M1*** can be evaluated as a function of the perturbation introduced in the adversarial examples (i.e. the attack data points). Typically different levels for this perturbation are considered to analyse the robustness of the system for different attack's strength.
- In metric G2_Q2_M2, the ***average minimum perturbation*** will be measured according to different norms typically used in the research literature, such as L1, L2 (Euclidean norm) or L-infinity norms. The analysis of different norms can provide more insights about the robustness of the models to adversarial attacks at test time. High values for the average minimum perturbation imply more robustness.
- Evaluating whether the assessment of the proposed metrics is positive or negative can be **subjective** as it **depends on several factors** including the dataset, the learning algorithm tested, the type of attack implemented and its strength. Related KPIs on the security of the learning algorithms do not require specific figures for assessing the security of the system. However, these metrics are useful to assess and compare the robustness of learning algorithms (including defensive capabilities against these attacks or not), which allows to select models according to the security requirements needed for a specific application.

4.3 Goal 3 (G3): Assessing data value extraction and monetization strategies

In what follows we will define a collection of questions that better determine the Goal 3.

G3	
Analyse	Task Alignment and Data Value
For the purpose of	Evaluating
With respect to	Utility for a given task under different operation modes.
From the viewpoint of	Technical perspective
In the context of	WP6: Assessment of data quality, scalability, computational efficiency and security

4.3.1 Questions for G3

In what follows we will define a collection of questions that better determine the Goal 3.

Identifier	Questions
G3_Q1	Is the task alignment procedure able to detect which are the most relevant data contributions to solve a given problem?
G3_Q2	Is the data value estimation method able to reward every participant according to the real data value of their data contribution?

4.3.2 Metrics for G3

We define here the metrics and benchmarks for questions in Goal 3.

Identifier	KPI	Format	Method of collection and measurement	Benchmark
G3_Q1_M1	Error rate of the task alignment method.	Double	Test ML models: as described in experiment G3_E1.	The benchmark will be the full-knowledge (gold-standard) solution. The assessment is considered positive if MUSKETEER is able to detect the users with data aligned to the task and exclude the other.
G3_Q2_M1	Error in reward estimation with respect to the full-knowledge approach.	Double	Test ML models: as described in experiment G3_E2.	The benchmark will be the full-knowledge (gold-standard) solution. The assessment is considered positive if MUSKETEER is able to estimate the reward within a 5% deviation from the gold-standard solution.

5 Selected datasets and their characteristics

In this section we briefly describe the potential datasets selected to carry out the assessment experiments. To facilitate the replicability of the experiments by other researchers we will rely on publicly available datasets. We propose to use a collection of datasets with a wide range of characteristics to explore their behavior under different conditions: number of

training patterns, number of features, type of features (numerical, categorical), type of target values (continuous for regression, discrete for classification, not available for clustering, etc.). For every experiment, one or more public datasets will be selected, to facilitate the replication of experiments by other researchers.
As previously mentioned, the evaluation experiments described in this document focuses on the evaluation of the ML components from a general point of view, and therefore we will not use the datasets provided by the use cases. The datasets from use case pilots will be used in WP7 for the final validation.

As a tentative list, we expect to use datasets in the experiments among the following (this list may be subject to some variations, according to the project's needs):

- **MNIST**: (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is widely used for training and testing in the field of machine learning [MNIST]. The input data are pixels values of 28x28 images, and the targets are the '0-9' labels.
- **Fashion-MNIST**: is a dataset with *Zalando's* articles images, consisting on a training set of 60,000 examples and a test set of 10,000 examples with images belonging to 10 different classes, including clothing and shoes [Xiao et al., 2017]. Similar to MNIST, the input data are pixel values of 28x28 images, and the targets are the '0-9' labels.
- Pima Indians **Diabetes**: The dataset consists of several medical predictor (independent) variables and one target (dependent) variable. Independent variables include the number of pregnancies the patient has had, their BMI, insulin level, age, etc. The goal is to predict the onset of diabetes based on diagnostic measures [Diabetes].
- Red **Wine** Quality Index: contains 1,599 red wines with 11 variables on the chemical properties of the wine. At least 3 wine experts rated the quality of each wine, providing a rating between 0 (very bad) and 10 (very excellent) [Wine].
- **Adult** income: Prediction task is to determine whether a person makes over 50K dollars a year. The input variables are both numerical and categorical. [Adult]
- **w8a**: This dataset consists on a webpage binary classification task. The task is learning to classify whether a webpage belongs to a certain category based on 300 features. [w8a]
- **Susy**: classification problem to distinguish a signal process which produces **su**per-**sy**mmetric particles from a background process which does not. [Susy]

- **Covertype** Data Set: Predicting forest cover type from cartographic variables only (no remotely sensed data). The actual forest cover type for a given observation (30 x 30 meter cell) was determined from US Forest Service (USFS) Region 2 Resource Information System (RIS) data. Independent variables were derived from data originally obtained from US Geological Survey (USGS) and USFS data. Data is in raw form (not scaled) and contains binary (0 or 1) columns of data for qualitative independent variables (wilderness areas and soil types). [Covertype]
- **Boston** Housing Dataset: contains information collected by the U.S Census Service concerning housing in the area of Boston Mass. It was obtained from the StatLib archive [Boston]. It has two prototasks: nox (**Boston-nox**), in which the nitrous oxide level is to be predicted; and (**Boston-price**) price, in which the median value of a home is to be predicted. [Boston]
- **YearPredictionMSD** Data Set: Prediction of the release year of a song from audio features. Songs are mostly western, commercial tracks ranging from 1922 to 2011, with a peak in the year 2000. [YearPredictionMSD]
- **Mopsi** User locations (Joensuu). Subset of GPS locations from Mopsi, Finland, in Joensuu area. [Mopsi]
- **Spambase** is a dataset for binary classification consisting on 4,601 emails containing both *good* and *spam* emails. Each email is represented as a vector with 57 features (taking values in the interval [0, 1]) representing the frequency of appearance of specific keywords in the email [Hopkins et al., 1999].

5.1 Datasets characteristics

In the following table we will summarize the characteristics of the used datasets. In the references section we also provide the link to the download site. The characteristics indicated in the table correspond to the situation of the dataset in the moment when they were downloaded. In case an additional transformation is needed for a particular experiment, it will be described in the corresponding experimental section. The type of tasks that a dataset can be used to are summarized in the last column).

Dataset Name	Number of Patterns[8] (train)	Number of Patterns (validation*)	Number of Patterns (test*)	Number of Features	Type of features	Targets	Tasks[9]
MNIST	60,000		10,000	784	int	cat	MCC, BC*, CL, CO
Fashion-MNIST	60,000		10,000	784	int	cat	MCC, BC*, CL, CO
Diabetes	768			8	Int, float	int	BC, CO
Wine	1,600			12	float	float	R, CO
Adult	28,000	4,561	16,281	123	float, cat	int	BC, R, CO
w8a	39,749	10,000	14,951	300	Sparse int	cat	BC, CO
Susy	3,500,000	500,000	1,000,000	18	float	cat	BC, CO
Covertype	581,012			54	float, categ.	cat	MCC, CO
Boston	51,630			14	float, int	float	R, CO
YearPredictionMSD	463,715		51,630	90	float	int	R, CO
Joensuu	6,014			2	float	-	CL, CO
Spambase	4,601			57	float	int	BC, CO

(*) If the partition is available in the original dataset.

(**) If the original dataset does not provide validation or test sets, we define them by splitting the training set.

[8] Training patterns or records.

[9] BC: Binary Classification, MCC: Multi-Class Classification, CL: Clustering, R: Regression, CO: Correlation estimation.

6 Experimental setup

We assume that, additional to the training data from every user, local validation and test datasets are available. This assumption is only needed for the assessment purpose, in the normal operation of the MUSKETEER platform those datasets are not mandatory.

The local validation dataset will be used to adjust some of the parameters of the model, if needed (hyperparameter selection, working point, etc.)

After the training is complete, the performance is evaluated using the local test set.

The experiments described in this section will be run for every available pair algorithm/POM, such that conclusions on the behavior of every algorithm under every POM conditions can be extracted.

The ML library will be designed to be as modular and easy to use as possibly. After deciding which POM to use, the corresponding objects to that POM will be loaded, and the ML models will have a unique form of use (much in the line of other libraries, where every model has .fit() and .predict() methods). Therefore, there is no need of establishing any experimental distinction among the different POMS, because all of them will be benchmarked in the same way.
A full detailed description of these experiments and the needed steps to reproduce them will only be available when the final version of the MUSKETEER architecture and ML libraries are available.

6.1 Experiments for Goal 1

***G1_E1**: We select datasets MNIST and Joensuu and apply to them the corresponding[10] clustering algorithm from a standard library to obtain a reference performance value (benchmark). We execute the same training in MUSKETEER and compare the results using the described metrics. We run several experiments with a different number of training patterns, number of users and number of features.*

***G1_E2**: We select datasets Wine, Adult and Boston and apply to them the corresponding regression algorithm from a standard library to obtain a reference performance value (benchmark). We carry out the same training in MUSKETEER and compare the results using the described metrics. We run several experiments with a different number of training patterns, number of users and number of features.*

[10] As an example, if the MUSKETEER clustering method implements *k*-means, then the comparison will be done with respect to the results of the *k*-means algorithm in the reference library (Scikit-Learn, for instance).

***G1_E3**: We select datasets MNIST and Covertype (MCC) and Diabetes and w8a (BC) and apply to them the corresponding classification algorithm from a standard library to obtain a reference performance value (benchmark). The Susy dataset is a very large one, and it could be used to test some of the algorithms/POMs. We carry out the same training in Musketeer and compare the results using the described metrics. We run several experiments with a different number of training patterns, number of users and number of features.*

***G1_E4**: We select datasets YearPredictionMSD, Adult and Boston and apply to them the corresponding correlation estimation algorithm from a standard library to obtain a reference performance value (benchmark). We carry out the same training in Musketeer and compare the results using the described metrics. We run several experiments with a different number of training patterns, number of users and number of features.*

***G1_E5**: We select datasets Diabetes, Adult and YearPredictionMSD and apply to them the corresponding feature selection/extraction algorithm from a standard library to obtain a reference performance value (benchmark). We carry out the same training in Musketeer and compare the results using the described metrics. We run several experiments with a different number of training patterns, number of users and number of features.*

6.2 Experiments for Goal 2

***G2_E1:** We select datasets MNIST, Fashion-MNIST and Spambase. We perform indiscriminate poisoning attacks against the tested learning algorithm varying the fraction of poisoning points compromised across all the platform users, ranging from 0% (no attack) to 20%. We test the algorithm's robustness against these attacks by analysing the loss in performance (evaluated on a separate test set). We run several experiments with different attack strategies (which will be developed in T5.2) varying the number of training examples used to train the learning algorithm.*

Note: This experiment may require to be reformulated, depending on the poisoning attack strategies developed in task T5.2. For example, [Bhagoji et al., 2019] introduced poisoning attacks targeting federated learning algorithms using model poisoning, where one or several users send malicious model updates to the central node that aggregate the model. These attacks do not require injecting malicious points in the training dataset. Instead, attackers manipulate directly the model updates that are sent to the server. In view of this, we may add an extra experiment to consider both model and data poisoning attacks or we may reformulate experiment G2_E1.

***G2_E2**: We select datasets MNIST, Fashion-MNIST and Spambase. We perform targeted poisoning attacks against the tested learning algorithm varying the fraction of poisoning points compromised across all the platform users, ranging from 0% (no attack) to 20%. We test the algorithm's robustness against these attacks by analysing the loss in performance, evaluated on the set of points targeted by the attacker. We run several experiments with different attack strategies (which will be developed in T5.2) varying the number of training examples used to train the learning algorithm and the number of points targeted by the attacker.*

Note: As in the previous case, this experiment may need to be reconsidered or an additional experiment will be included depending on the attacks developed and considered in task T5.2.

***G2_E3:** We select datasets MNIST, Fashion-MNIST and Spambase. Given a test dataset, we select an attack strategy to craft adversarial perturbations for all the samples in the test set given a metric (e.g. L1, L2 or L-infinity norms) and a value for the maximum perturbation allowed to create the adversarial perturbations (according to the selected metric). With these settings, we measure the fraction of successful adversarial examples, i.e. those that produce an error when tested on the tested learning algorithm.*

Note: The attack strategies to be used in this experiment will be defined and developed in task T5.3. Targeted and indiscriminate attack strategies can be considered here.

***G2_E4:** We select datasets MNIST, Fashion-MNIST and Spambase. Given a test dataset, we select an attack strategy to generate adversarial examples and a metric to measure the adversarial perturbation introduced by the attacker (e.g. L1, L2 or L-infinity norms). For each example in the training dataset we measure the minimum adversarial perturbation required (according to the attack strategy and the metric selected) to produce an error in the target algorithm. Finally we measure the average minimum perturbation across all the samples in the test dataset. Larger values are indicative of more robust algorithms.*

***G2_E5:** We select datasets MNIST, Fashion-MNIST and Spambase. We perform coordinated poisoning attacks with user's collusion against the tested algorithm varying the fraction of colluding users from 0% to 20%. Using the detection techniques that we will develop in task T5.5, we will measure the fraction of malicious users detected. We run several experiments with different attack strategies (which will be developed in T5.4).*

G2_E6: *We select datasets MNIST, Fashion-MNIST and Spambase. We perform coordinated poisoning attacks with user's collusion against the tested algorithm varying the fraction of colluding users from 0% to 20%. Using the detection techniques that we will develop in task T5.5, we will measure the fraction of benign users that are incorrectly detected as malicious (i.e. we measure the false positive rate of the detection algorithm). We run several experiments with different attack strategies (which will be developed in T5.4).*

6.3 Experiments for Goal 3

G3_E01*: We select datasets MNIST, Adult, and YearPredictionMSD, and partition the data among a given number of workers. We degrade some of the data partitions by adding noise, deleting some of the values, replacing values by random ones, etc. and we evaluate if the task alignment procedure is able to detect the perturbed data chunks. We measure the error in the detection of the irrelevant data chunks.*

G3_E02*: We select datasets MNIST, Adult, and YearPredictionMSD, and partition the data among a given number of workers (different sizes and possibly different data distributions). We run a brute-force procedure on the undistributed data to estimate the actual contribution of every chunk to the task solution and use this solution as a golden reference. We evaluate if the data value estimation methods proposed in D4.2 are able to estimate the real contribution of every worker. We compare the coincidences between the golden reference solutions with respect to the Musketeer estimations.*

7 Conclusions

In this document we have revisited and summarized the project objectives and the KPIs related to the general evaluation of the Machine Learning algorithms implemented under the different POMs. To develop an assessment procedure to evaluate the correct verification of all of them, we have proposed to adopt the Goal-Questions-Metrics methodology also used in other Work Packages in the project. We have defined several Goals that cover all of the evaluation objectives. Then we have decomposed them into simpler and more operative questions to finally define the metrics that will be used to verify the achievement of the goals. For the evaluation purpose we will use several publicly available datasets instead of the data from the use cases. The first reason is to facilitate the replicability of the experi-

ments by other researchers. The second reason for that choice is that the final use case performance will be evaluated in WP7. We have briefly described the characteristics of the selected datasets and the experiments to be carried out, although more detail about the experiments will be provided when the platform architecture and Machine Learning library is in a more developed stage. Summarizing, we have adopted a formal approach to fulfill the assessment of the proposed and implemented schemes from the point of view the scalability, computational efficiency, performance, security, and data value estimation.

8 References

[Adult] https://archive.ics.uci.edu/ml/datasets/adult

[Bhagoji et al. 2019] A.N. Bhagoji, S. Chakraborty, P. Mittal, S. Calo. "Analyzing Federated Learning through an Adversarial Lens." International Conference on Machine Learning, pp. 634-643, 2019.

[Boston] http://lib.stat.cmu.edu/datasets/boston

[Covertype] https://archive.ics.uci.edu/ml/datasets/covertype

[Diabetes] https://github.com/LamaHamadeh/Pima-Indians-Diabetes-DataSet-UCI

[Fashion-MNIST] https://github.com/zalandoresearch/fashion-mnist

[GrantAgreement] Grant Agreement number: 824988 — MUSKETEER — H2020-ICT-2018-2020/H2020-ICT-2018-2

[Goldsmith, 2007] Goldsmith, Simon F., Alex S. Aiken, and Daniel S. Wilkerson. "Measuring empirical computational complexity." *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*. ACM, 2007.

[Hopkins et al., 1999] M. Hopkins, E. Reeber, G. Forman, J. Suermondt. "Spambase Data Set." Hewlett-Packard Labs.

[MNIST] http://yann.lecun.com/exdb/mnist/

[Mopsi] http://cs.uef.fi/mopsi/data/

[PySyft] https://github.com/OpenMined/PySyft

[SecureML] https://eprint.iacr.org/2017/396.pdf

[Solingen] R. van Solingen, E. Berghout (1999). The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development, McGraw-Hill

[Susy] https://archive.ics.uci.edu/ml/datasets/SUSY

[Wine] https://archive.ics.uci.edu/ml/datasets/wine+quality

[w8a] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary/w8a

[Xiao et al., 2017] H. Xiao, K. Rasul, R. Vollgraf. "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms." arXiv preprint arXiv:1708.07747, 2017.

[YearPredictionMSD] https://archive.ics.uci.edu/ml/datasets/yearpredictionmsd