Tailored IoT & BigData Sandboxes and Testbeds for Smart, Autonomous and Personalized Services in the European Finance and Insurance Services Ecosystem

Infinitech

D4.15 – Encrypted Data Querying and Personal Data Market - III

Revision Number	**3.0**
Task Reference	T4.5
Lead Beneficiary	FBK
Responsible	Bruno Lepri
Partners	FBK, GFT, HPE, IBM, INNOV, UNIC
Deliverable Type	Demonstrator (DEM)
Dissemination Level	Public (PU)
Due Date	2022-03-31
Delivered Date	2022-03-31
Internal Reviewers	ATOS, JSI and GFT
Quality Assurance	INNOV
Acceptance	WP Leader Accepted and Coordinator Accepted
EC Project Officer	Beatrice Plazzotta
Programme	HORIZON 2020 - ICT-11-2018
	This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement no 856632

Contributing Partners

Partner Acronym	Role[1]	Author(s)[2]
FBK	Lead Beneficiary	Gabriele Santin, Bruno Lepri
IBM	Contributor	Fabiana Fournier, Inna Skarbovsky
JSI	Internal Reviewer	Maja Srkjanc
ATOS	Internal Reviewer	Nuria Ituarte Aranda
GFT	Internal Reviewer	Maurizio Megliola
INNOV	Quality Assurance	John Soldatos

Revision History

Version	Date	Partner(s)	Description
0.1	2022-01-15	IBM, FBK	First draft of ToC
0.1	2022-01-20	IBM	Revised ToC
0.2	2022-03-01	FBK	Initial contributions of Sections 1, 2, and 3
0.3	2022-03-04	FBK	Revision and finalization of Sections 1, 2, and 3
0.7	2022-03-10	FBK, IBM	Executive summary
1.0	2022-03-16	FBK	First Version for Internal Review
2.0	2022-03-	ATOS, JSI, and GFT	Version for Quality Assurance
3.0	2022-03-31	FBK	Version for Submission

[1] Lead Beneficiary, Contributor, Internal Reviewer, Quality Assurance

[2] Can be left void

Executive Summary

This deliverable (D4.15) is the last of three deliverables planned in the scope of Task 4.5 of the INFINITECH project. The purpose of this task is to overcome the current limitations of standard data sharing paradigms, and this ambitious goal is achieved by the design and implementation of a framework for securely querying, processing, and analyzing data over the INFINITECH permissioned blockchain infrastructure. This goal will enable decentralized, federated, and secure execution of machine learning (ML) algorithms and lay the foundation for a market of insights obtained by running ML algorithms.

In this framework, deliverable D4.15 describes the finalization of the algorithmic aspects of the *Insights Sharing and Provenance* conceptual module jointly designed by IBM and FBK in deliverable D4.14 [11]. In particular, we describe the details of the modeling choices and the design of the Federated Learning algorithm that constitute the ML core of the module and present its application on a benchmark dataset. We remark that this module is based on a secure and auditable execution framework developed by IBM, and built on top of IBM Hyperledger Fabric [2, 7], the blockchain platform selected by the INFINITECH project.

In this deliverable we first recall the structure of the federated learning algorithm, which is based on Random Forests (RF) [3, 5], and describe how it can implement a secure sharing of ML-derived insights, instead of raw data, among different organizations (e.g., banks, insurance companies, etc.). We then discuss the details of the implementation of the algorithm, including the formulation of the methods and its properties. We finally report several aspects of its usage on a benchmark example.

Overall, this report provides a detailed account of the finalized version of the Federated Learning algorithm that has been developed by FBK together with IBM, as presented in Deliverable D4.14 “Encrypted Data Querying and Personal Data Market – II” [11]. This algorithm, together with the blockchain framework developed by IBM and discussed in Deliverable 4.12 [13](submitted at the same time), constitute a Minimum Viable Product (MVP) that will permit INFINITECH partners and INFINITECH users to cooperatively solve financial and insurance challenges using ML approaches, but without losing the control of the data they own.

Table of Contents

List of Figures

List of Tables

Abbreviations/Acronyms

Abbreviation	Definition
AD	Anomaly Detection
Bacc	Balanced Accuracy
FL	Federated Learning
FN	False Negative
FP	False Positive
ML	Machine Learning
MVP	Minimum Viable Product
OPAL	Open Algorithms
PCA	Principal Component Analysis
Prec	Precision
Rec	Recall
RF	Random Forest
TP	True Positive
TN	True Negative

1. Introduction

The current deliverable is the last one of a series of three deliverables whose aim is to describe the activities conducted in the Task 4.5 “Secure and Encrypted Queries over Blockchain Data” of the INFINITECH project. The main objective of this task is the design and implementation of a framework for querying encrypted data over the INFINITECH permissioned blockchain infrastructure and for running Machine Learning (ML) algorithms on these data and enabling a personal data market.

As already mentioned in the deliverable D4.13 “Encrypted Data Querying and Personal Data Market – I” [10] (submitted at M14), the inspiration for this framework comes from two recent approaches, that is ENIGMA [22] and Open Algorithms (OPAL) [18]. Both approaches provide a mechanism for the privacy-preserving sharing of data across multiple data repositories. In particular, OPAL introduces the concept of moving the ML algorithms to the data repositories, where each data repository participating in the computation performs all its computations behind the firewalls. Additionally, ENIGMA introduces the notion of Multi-Party Computation (MPC) [19] that gives the data repositories the ability to collectively perform an algorithm computation that produces some results without revealing the raw data.

In Deliverable D4.14 “Encrypted Data Querying and Personal Data Market – II” [11] (submitted at M22), we revised our initial proposal for an INFINITECH framework for securely accessing, managing, and sharing data across financial and insurance institutions. We described the novel *Insights Sharing and Provenance* conceptual module jointly designed by IBM and FBK. More precisely, we introduced the five components of this module, namely (i) *Identity Manager and User/Client Authentication*, (ii) *Federated Learning Artifacts Store*, (iii) *Artifacts Usage Audit*, (iv) *Secured Execution*, and (v) *Tokenization*. This module was built on top of Hyperledger Fabric (simply Fabric) [2, 7], the blockchain platform selected by the INFINITECH project.

The current deliverable constitutes the last step in the full specification of the module, and its implementation in an MVP. In achieving this goal, the D4.15 is complemented by Deliverable D4.12 “Blockchain Tokenization and Smart Contracts - III” [13] (submitted at the same time), which describes the implementation of the blockchain solution.

In Section 2, we introduce in detail the federated learning algorithm, based on Random Forests (RFs), implemented as the basis for the INFINITECH framework for securely sharing insights.

Section 3 analyzes instead the details of the application of the algorithm on a benchmark dataset for fraud detection. Through this experiment the ML algorithm is analyzed from different points of view, and its features are discussed.

Finally, we draw some conclusions, and we discuss possible extension of the Federated Learning model beyond the details discussed in this deliverable and the MVP produced within the INFINITECH project.

We highlight that the content of this deliverable was largely used for the paper “A Framework for Verifiable and Auditable Federated Anomaly Detection”, authored by Gabriele Santin (FBK), Inna Skarbovsky (IBM), Fabiana Fournier (IBM), and Bruno Lepri (FBK). The paper is currently uploaded on ArXiv (https://arxiv.org/abs/2203.07802) and under submission to the special issue “Trustable, Verifiable, and Auditable Federated Learning” of the journal “IEEE Transactions on Big Data”.

1.1 Objectives of the Deliverable

The main goals of Task 4.5 “Secure and Encrypted Queries over Blockchain Data” are the design and implementation of a framework for querying encrypted data over the INFINITECH permissioned blockchain infrastructure, namely Hyperledger Fabric, and for running ML algorithms on them, as well as creating the foundation for a personal data market where individuals and organizations will be able to trade their data or insights in exchange for tokens or other assets.

These goals encompass, in this third deliverable, the following specific objective:

- To describe the ongoing design, research and implementation work on the federated learning algorithms selected as the basis for the INFINITECH framework for securely sharing ML insights. More specifically, we introduce an approach based on Random Forests (RF) [5], also providing the details of its architecture, and describing its results in a first application to a fraud detection task.

1.2 Insights from other Tasks and Deliverables

Deliverable D4.15 "Encrypted Data Querying and Personal Data Market – III" is released in the scope of WP4 "Interoperable Data Exchange and Semantic Interoperability" activities and documents the implementation of the Federated Learning part within the collaborative work performed by IBM and FBK within the context of tasks T4.4 "Tokenization and Smart Contracts Finance and Insurance Services" and T4.5 "Secure and Encrypted Queries over Blockchain Data".

This document relies on previous and current work reported in the following deliverables:

- D4.7 "Permissioned Blockchain for Finance and Insurance - I" [8] (submitted at M11) which revolves around the first version of blockchain applications carried out in the scope of the INFINITECH project.
- D4.8 "Permissioned Blockchain for Finance and Insurance - II" [9] (submitted at M19) presents the second version of the blockchain activities in the INFINITECH project.
- D4.10 "Blockchain Tokenization and Smart Contracts – I" [12] (submitted at M14) motivates the usage of tokenization in blockchain networks in the financial and insurance sectors and describes the first round of activities carried out on tokenization in the INFINITECH project.
- D4.13 "Encrypted Data Querying and Personal Data Market – I" [10] (submitted at M14) motivates the need for a paradigm change concerning the currently dominant model of siloed data collection, management, and exploitation as well as it illustrates the first round of activities carried out in INFINITECH for designing and implementing a framework for securely accessing, managing, and sharing data or ML insights between customers, financial and insurance institutions.
- D4.14 "Encrypted Data Querying and Personal Data Market – II" [11] (submitted at M22) introduces the design of the module, and presents at a high level the design principles and initial developments in both the algorithmic part, and the blockchain solution.
- D4.12 "Blockchain Tokenization and Smart Contracts - III" [13] (submitted at the same time) which provides in-depth technical details of the implementation of the *Blockchain based federated learning environment and data marketplace* introduced in D4.14 [11].

1.3 Updates with respect to the Previous Version (D4.14)

The current deliverable, D4.15 "Encrypted Data Querying and Personal Data Market – III" contains the full specification, description and testing of the algorithm introduced in D4.14 "Encrypted Data Querying and Personal Data Market – II" [11]. Section 2 "Federated Learning" develops and specifies the content of Section 3 in D4.14.

1.4 Structure

We organized the structure of D4.15 as follows:

- Section 2 introduces in detail the federated learning algorithm, based on RF, implemented as the basis for the INFINITECH framework for securely sharing insights.
- Section 3 analyzes in detail the application of the algorithm on a benchmark dataset for fraud detection.
- In Section 4 we conclude the deliverable describing the possible extension of the federated learning approach to other tasks and scenarios.

2 Federated Learning

In this section we introduce the federated learning (FL) algorithm, implemented as the basis for the INFINITECH framework for securely sharing ML insights, and we provide the details of its properties.

In the field of ML, an increasing amount of new attention is devoted to the issues of data ownership, data privacy, and data trading. In this setting, multiple related aspects are being analyzed and systematized within the framework of FL [15]. This new field deals with the study of various scenarios where multiple agents own separate batches of data, and they are willing to cooperate for the construction of different ML models that are possibly distributed across the nodes. This collaboration leverages different communication strategies to overcome the limitations of the single agents, which can be due to scarcity of data or scarcity of computational resources, but with the important constraint that data should never leave the location where it resides. This approach is in stark contrast with more traditional data-centralized methods, and it paves the way for several new algorithms that focus on various aspects of data ownership.

In particular, we focus on Anomaly Detection (AD) [4] systems that are common in the financial industry, such as fraud detectors or default predictors. The peculiar characteristic of these applications is that a classifier must be trained to identify anomalous cases, i.e., events that are unusual compared to the most frequent patterns observed in the data. In particular, anomalous examples are scarce by definition. Consequently, different agents such as banks, financial institutions, insurance companies may foresee a benefit in collaborating with their peers in order to trade knowledge and improve their individual models. On the other hand, the data that is used to train these systems is usually shared with caution, since it typically comprises sensitive personal information regarding the financial position or the individual characteristics of the clients. Moreover, the possession of these data is often an important asset for the single agents, which are possibly not willing to give them away once for all, but would rather like to develop an on-purpose sharing. This option is inherently difficult with easily copyable digital data.

To this end, we present a fully decentralized FL system where multiple agents collaborate for the training of one model per agent, and which is (i) privacy preserving by design, (ii) robust to changes in the network topology and to asynchronous communications, and (iii) resistant to malicious intrusions and adversarial attacks.

The system is designed so that each agent trains an ensemble classifier [4, 22], that is a ML model that is made of multiple simple estimators that are combined as atomic building blocks. This structure makes it easy to iteratively improve local models as well as exchanging knowledge between agents by sharing the top performing blocks. We use in particular Random Forests (RF) [3, 5] as ensemble models, as they are well-suited for AD problems and robust to missing data, but we comment below how this is not a restrictive choice and other ensembles could be adopted. Moreover, the chosen design of the ML algorithm permits to integrate the system in the BC infrastructure that guarantees trustable and verifiable execution of the algorithm and certifies the communication between the nodes.

2.1 Architecture of federated learning algorithm

The algorithm is based on the use of RFs [3], which are flexible and effective learning algorithms, commonly used for classification and AD tasks. We extend this method to work in a federated approach by designing suitable rules for the sharing and merging of these learners between different nodes.

2.1.1 Agents and atomic operations

The federation is composed of $\mathbb{N}$ agents or nodes. Each node has an individual dataset, and its goal is to train an ensemble classifier based on RF. These RF are ensemble classifiers made of a number of binary decision trees as simple building blocks. We consider three atomic operations to modify an ensemble: one enlarges the ensemble, one keeps its size bounded, and one selects the top performing estimators. They are defined as follows:

1. **ADD:** The node starts by training a RF containing a fixed number n_estimators of trees. After the first iteration, if the fit operation is executed again then the existing forest is enlarged by training an additional number n_estimators of trees. Each newly trained tree is marked with the ID of the node, and with an incremental and unique ID of the trees in this node. In this way, at each stage of the algorithm it is possible to uniquely identify the identity and source of each tree in the federation. If the number of trees exceeds a prescribed number n_max, the CROP operation is executed.
2. **GET_TOP:** Once a set of trained trees is available, the single node can rank them according to their importance in the prediction. Since each node has access only to its own training data, it is important to implement a strategy that is able to rank the nodes using only these data, but on the other hand that avoids overfitting, i.e. the excessive fine tuning of the classifier on the training data that can possibly prevent a meaningful generalization on new data [4]. We implement a greedy method [6] based on Gaussian Process (GP) [20], which is able to select a subset of the trees which provides an almost-optimal minimization of the posterior variance. In this way, restricting the ensemble to the selected nodes makes it possible to keep the largest part of the information contained in the full model. This mechanism is used to rank the single trees from the most important to the less important.
3. **CROP:** To satisfy the memory limit of each node, a CROP operation is implemented. In particular, the memory constraints are met by imposing a maximum depth max_depth on the newly trained trees (see point 1), and additionally imposing that each node stores at most a number max_estimators of trees. To enforce this condition, whenever new trees are added to the node (either using the FIT operation of point 1, or by the federated operations described in the next section) the trees are ranked using the GET_TOP operation of point 2, and only the top max_estimators are kept, while the other ones are discarded.

2.1.2 Federated learning

In addition to the single node learning, we have implemented a federated strategy that allows several nodes to collaborate in solving a common problem. The group of agents is partially connected according to a network represented by an undirected graph, so that each node can communicate only with its set of neighbors. This network is possibly time-varying, and this allows us to model temporary interactions and communication failures.

To manage the communication, each node has a registry with a slot for each of the other nodes. We assume that each node can write a message to its slot in the registry of each node to which it is connected. Using the registry and the atomic operations on the ensemble, we implemented the three fundamental operations that each agent can perform to change its status at each iteration:

- FIT: A number n_new of novel decision trees are trained by the agent on its own dataset, and they are added to the local RF. If the resulting number of estimators is larger than n_max, then the method CROP is used to keep only the best ones.
- SHARE: The agent identifies its top n_share estimators and writes them to the registry of each of its first order neighbors. If a registry slot contains already some estimators from previous communications, they are overwritten.
- GET: The agent reads its registry slots to collect all the estimators received in the previous iterations (if any), and adds them to its current ensemble by using the ADD method. If this operation makes the ensemble larger than n_max, excess estimators are removed by a call to the CROP method.

.

2.2 Properties of the algorithm

The entire algorithm is completely decentralized, since it only requires the existence of a communication network and the agreement on a set of initial parameters. The model supports time-varying networks, and it allows for completely asynchronous communication, including the option for different nodes to join or leave the federation at different times.

It is worth to note that all the operations, except for GET_TOP, are well defined for any type of ensemble classifier, and do not require further specification to be implementable. The only method-specific operation is thus GET_TOP, which requires defining a way to rank the estimators within an ensemble. We remark that similar design principles as those used here for RF could be adopted to work with more general ensembles. In this sense, the present algorithm may be understood as a family of algorithms, parametrized by the method that is used to promote some estimators with respect to other ones. The importance of this ranking system is reflected in the fact that we are employing a registry with slots that stores only the last written information. In this way, when a node reads its registry via the GET method, it only reads the result of the most recent call of GET_TOP transmitted by its neighbors.

This solution is used also to guarantee that the registry has a bounded memory footprint, since in this way it needs to store at most n_share times N estimators at each time. Similarly, the bound n_max on the number of estimators held by each single node controls the size of each ensemble classifier. These two requirements can be translated to memory bounds if we assume that each estimator has a maximal memory size.

Moreover, the only operation that can create new estimators is FIT. Whenever this method is called, the newly constructed estimators are labeled with an identifier comprising the identifier of the creator node, and a progressive estimator counter maintained by the node itself. In this way each estimator in the federation is uniquely identified, and it is always possible to know which nodes trained it.

Moreover, communication between different nodes amounts only at the exchange of estimators via the SHARE and GET methods. Both the operations of creation and sharing are thus easily secured by means of the BC integration that is discussed in the deliverable D4.12 “Blockchain Tokenization and Smart Contracts - III” [13].

3 Evaluation on a fraud detection dataset

This section gives a detailed account of the tests of the federated learning algorithm we have run on a fraud detection dataset. Before presenting and discussing the actual results, we provide specifications of the communication network, of the preparation of the dataset, and of the setup of the algorithm's hyperparameters.

3.1.1 Dataset

We have built the algorithm to solve the *Credit Card Fraud Detection* problem published as a Kaggle challenge in November 2016 [1]. The dataset collects credit card transactions performed in Europe during two days in September 2013. The data is anonymized by publishing only 28 features resulting from a Principal Component Analysis (PCA) [14] of the original data. Additionally, each transaction contains a timestamp, the amount transferred, and a label indicating whether the transaction was fraudulent. We do not make use of the amount of the transaction and of the timestamp for this task.

The dataset is highly unbalanced, meaning that most of the transactions are legit ones. Table 1 reports the statistics of the entire dataset.

Table 1: Dataset for the federated learning use case on fraud detection.

Number of transactions	Number of frauds	Percentage of frauds	Features per transaction
284807	492	0.173%	28

3.1.2 Task and metrics

The dataset is associated with the task of predicting if a new transaction is a fraudulent or a legal one, given the 28 features used to describe it.

For AD problems, it is important to adopt metrics that are suitable to measure the performances of an algorithm in this task. We summarize the metrics that we will use in Table 2.

Table 2: Metrics for the federated learning use case on fraud detection.

Metric	Abbreviation	Description
True Positive	TP	The number of frauds that are correctly detected.
False Positive	FP	The number of legitimate transactions that are wrongly marked as frauds.
True Negative	TN	The number of legitimate transactions that are correctly detected.
False Negative	FN	The number of frauds that are wrongly marked as legitimate transactions.

Recall	Rec	The quantity TP / (TP + FN), that is the ratio between the correctly identified frauds, and the total number of actual frauds (detected or not).
Precision	Prec	The quantity TP / (TP + FP), that is the ratio between the correctly identified frauds, and the total number of transactions marked as frauds (correctly or not).
Balanced Accuracy	BAcc	The quantity (TP / (TP + FN) + TN / (TN + FP))/2, that is the mean between the normalized number of true positives and true negatives, each normalized by the total number of either positive or negative samples in the dataset.

3.1.3 Communication networks

We consider 20 nodes organized according to three different communication networks. We remark that, although the algorithm supports time-varying networks, we stick to a simpler static-network scenario for testing.

The networks are a fully disconnected one (D), a pairwise connected network (P), and fully connected one (C) (see Figure 2). These three networks model an increasing level of connectivity, from non-interacting nodes in D, to locally connected nodes in P, to fully connected nodes in C.

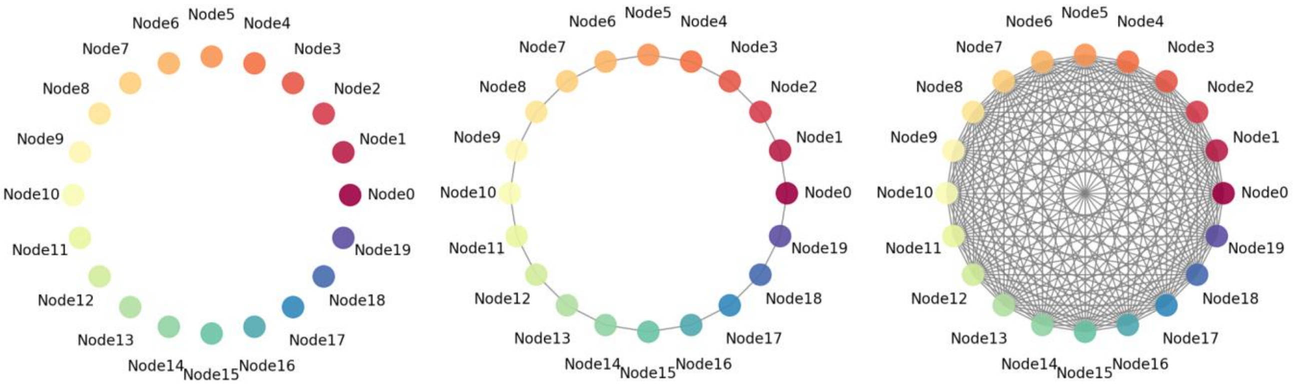

Figure 1 - Communication networks for the federated learning use case

3.1.4 Preparation of the dataset

To assign a set of data to each node, we split the dataset described in Section 3.1.1 in an unbalanced manner, in order to simulate the presence of nodes owning data of different quality. This is done by non-uniform randomized sampling of the positive and negative classes, in such a way that each transaction is assigned to a unique node. The statistics of the resulting set of ten datasets is summarized in Table 3.

For testing purposes only, we also create a shared test set that is used to assess the performance of the algorithm in an unbiased manner, using the metrics defined in Section 3.1.2. The set is obtained by collecting a random sample of the 10% of each of the ten datasets. We remark that the existence of such a centralized test set is not required by the actual architecture, but just used here to measure the performances.

The 20 datasets have a variable distribution of positive and negative examples. We report in Table 3 some relevant statistics.

Table 3: Statistics of the distributed dataset.

	Minimal value		Maximal value	
	Node	Value	Node	Value
Number of samples	Node9	3297	Node13	28249
Number of frauds	Node2	0	Node3	49
Fraud ratio	Node2	0.00%	Node11	0.41%

3.1.5 Hyperparameters setup

The algorithm is flexibly parametrized by several parameters that are set to specific values in these experiments. Their name, role, and value are defined in Table 4.

We remark that at this stage there has been no particular efforts to fine tune these parameters, since our interest is in obtaining a first insight into the effectiveness of the method. For this reason, they have been set either to default values (`sample_size` and `max_depth`) or set to values that produce a relatively small model that is fast to train and test.

Table 4: Name, role, and value of the parameters used in our experiments.

Name	Description	Value
`n_estimators`	Number of new trees that are trained at each execution of the FIT operation.	10
`max_depth`	Maximal depth of each trained tree.	10
`max_estimators`	Maximal number of trees that are stored in the RF of each node.	50
`n_share`	Number of trees that are sent to the neighbors via the SHARE operation.	10

3.1.6 Accuracy results

We use these metrics defined in Section 3.1.2 to assess the improvement of the federated models over the scenario where each node is isolated. To this end, for each node we compute on the test set the metric in the two federated cases (pairwise connected and fully connected) and their difference with the corresponding value in the disconnected case. We compute the mean and median of these differences over the 20 nodes. These values are reported in Figure 2 (left), and it can be observed that overall there is a significant increase (0.1-0.2) both in the mean and the median, and for all the three metrics. This confirms that, apart from the case of single nodes, the federation is very effective to improve the classifiers.

To offer an additional insight into the functioning of the sharing mechanism, we visualize in Figure 2 (right) the same metrics, but computed over the train sets of each single node. In this case, it is remarkable to observe that both the mean and median are negative, meaning that the accuracy is decreasing on the train set when entering the federation. Since the test metrics are instead increasing, this is a good sign that the federated algorithm is able to equip each node with a model that has an accuracy that goes far beyond the own dataset, and is effectively able to share insights not present in each single node.

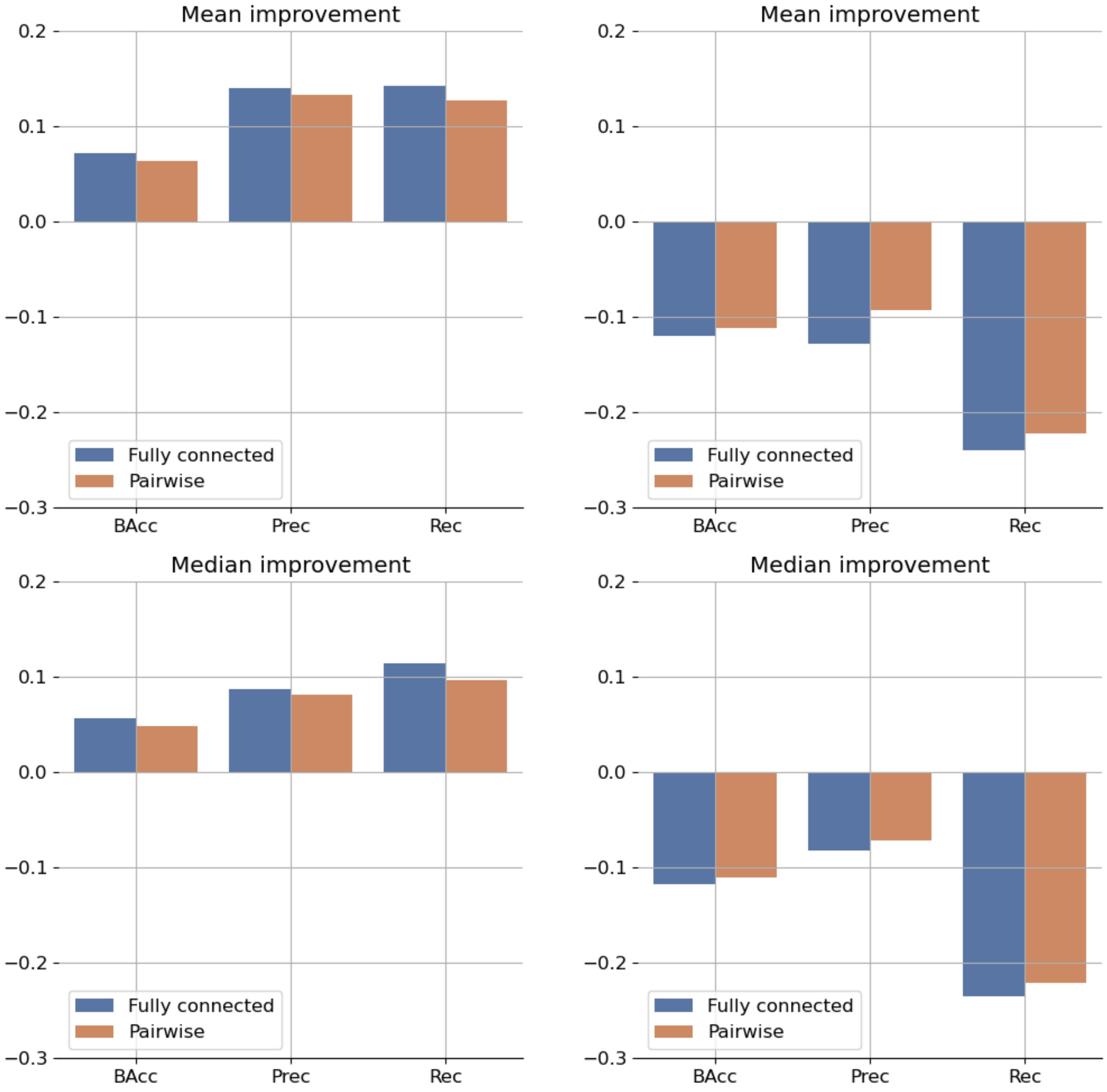

Figure 2 - Mean and median improvement in the three metrics over the disconnected case for the two federated scenarios (Fully connected and Pairwise). The metrics are computed over the train set (right) and the test set (left).

3.1.7 Mixing of the estimators

It is interesting to note now how the network connection influences the propagation of the shared estimators among the different nodes. To this end, Figure 4 shows the final status of the RF of each node, and in particular the distribution of the trees within each RF according to the source node. Namely, since each estimator is uniquely identified, it is possible at each moment to check where the estimators of each node have been fitted. In the figure, we show in each row the origin of the estimators of each node. In the disconnected case (left panel) there is no mix, and indeed each node owns only estimators that it fitted itself. In the fully connected case (right panel) a quite uniform mixing can instead be observed, with the addition that some nodes (Node0, Node2, Node5, Node6, Node8, Node10) produce almost no estimators that are used by the other ones. The fact that the mixing is quite stable among the nodes is an indication of the effectiveness of the sharing and ranking mechanism. In the intermediate case of the pairwise connected nodes (central panel) the mixing reflects the connection pattern, since each node holds estimators from its direct neighbors. In this case it is worth remarking that the estimators are effectively transmitted beyond the first order neighbors of a node, and this suggests that even a not fully connected network may be effective for the federation to work.

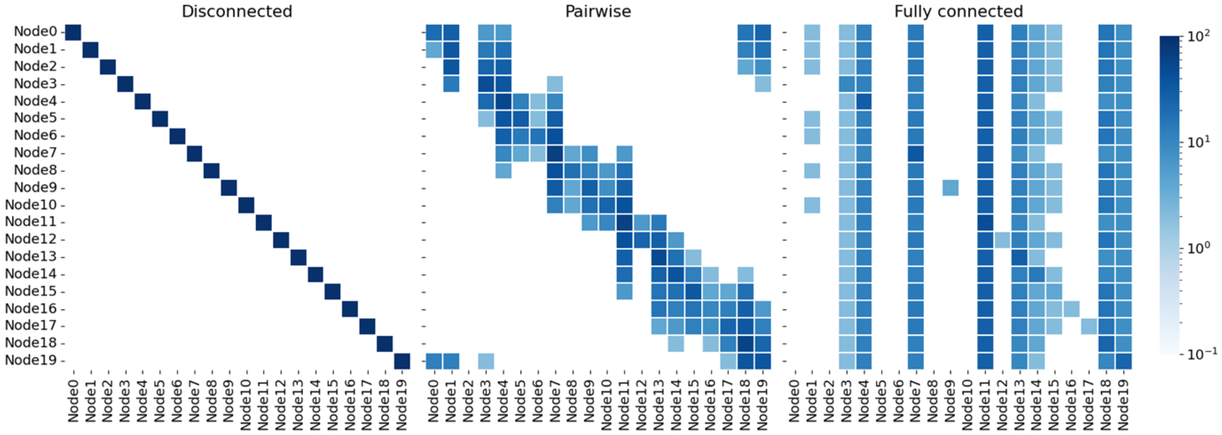

Figure 3 - Origin of the estimators selected by each node at the end of the iteration for the three connection settings. Each row represents a node, and the columns indicate the origin of its estimators. The values of each row are normalized as percentages which sum to 100%.

4 Conclusions and perspectives

The current deliverable has been developed around the objective of describing the design, research and implementation work on the federated learning algorithms selected as the basis for the INFINITECH framework for securely sharing ML insights.

As the last of the series of three deliverables (D4.13, D4.14, D4.15 “Encrypted Data Querying and Personal Data Market”, [10, 11]), this deliverable provided the final details of the implementation of the algorithm and demonstrated its testing and functioning. The results have shown that the proposed solution is effective in achieving the desired results of decentralized and asynchronous federated learning, which can ultimately be used for querying encrypted data over the INFINITECH permissioned blockchain infrastructure and for running ML algorithms on these data.

Together with the deliverable D4.12 “Blockchain Tokenization and Smart Contracts - III” [13] (submitted at the same time), which provides in-depth technical details of the implementation of the *Blockchain based federated learning environment and data marketplace*, Deliverable D4.15 is one of the two final pillars of the collaboration of FBK and IBM in Task 4.4 “Tokenization and Smart Contracts Finance and Insurance Services” led by IBM and Task 4.5 “Secure and Encrypted Queries over Blockchain Data” led by FBK.

As documented also in the deliverable D4.12 “Blockchain Tokenization and Smart Contracts - III” [13] (submitted at the same time), FBK and IBM collaboration on Task 4.4 “Tokenization and Smart Contracts Finance and Insurance Services” and Task 4.5 “Secure and Encrypted Queries over Blockchain Data” has resulted in the following achievements:

- Implementation of a blockchain based marketplace for assets (i.e., ML insights) managing and their trading using digital tokens.

- Implementation of a blockchain based environment for the secure execution of federated machine learning algorithms materialized for the case of fraud detection.

- A movie (presented during the 2nd Workshop on Blockchain Applications for Digital Finance held on March 2, 2022) emphasizing the highlights of the work (https://www.youtube.com/watch?v=H8M8PMIA8YU&list=PL9suUK-Ys8V3Dkzm7qmZnIb-VIc1eSMwp&index=21)

- A deep dive demo (step by step) of the MVP BC technical aspects available in the INFINITECH project marketplace (https://www.youtube.com/watch?v=J7ekCHSoWrg&list=PL9suUK-Ys8V3Dkzm7qmZnIb-VIc1eSMwp&index=22)

- Submission of a joint paper with FBK titled A Framework for Verifiable and Auditable Federated Fraud Detection to the IEEE Transactions on Big Data journal special issue on "Trustable, verifiable, and auditable federated learning". The code of the FL algorithm is published at https://github.com/GabrieleSantin/federated_fraud_detection.

The proposed algorithmic solution has the potential to be extended to more general scenarios, and it will be the basis of further investigation. In particular, mechanisms for the optimization of the communication network and extension to other Machine Learning classifiers will be the focus of further research.

This report should be read in conjunction with D4.12 - Blockchain Tokenization and Smart Contracts - III” which complements the technical details of the MVP with the BC implementation.

The current deliverable constitutes the final report of Task 4.5 and concludes the activities of the specific task.

Table 5: Conclusions (TASK Objectives with Deliverable achievements)

Objectives	Comment
Designing and developing an innovative federated learning framework for securely sharing ML insights and enabling data markets.	We have designed and developed an innovative federated learning approach based on Random Forests. We also documented its performances on fraud detection tasks. The federated learning framework is the ML basis of the MVP, developed by IBM and FBK within tasks T4.4 and T4.5, that provides the mechanisms for controlling and managing the access rights of the ML insights (i.e., the outcomes of the federated learning algorithm for fraud detection) stored on the BC and for trading these insights via digital tokens.

Table 6: Conclusions (map TASK KPI with Deliverable achievements)

KPI	Comment
Realize a blockchain based federated learning environment	*Target Value = 1* The joint MVP, developed by IBM and FBK within tasks T4.4 and T4.5, realizes the devised framework for the secure execution of federated machine learning algorithms exploiting BC technology in a fraud detection use case.
Realize a blockchain based data marketplace	*Target Value = 1* The joint MVP, developed by IBM and FBK within tasks T4.4 and T4.5, provides the mechanisms for controlling and managing the access rights of the assets stored in the BC (i.e., the outcomes of the federated learning algorithm for fraud detection) and for trading these assets via digital tokens.

5 Appendix A: Literature

[1] "Credit Card Fraud Detection", 2016. [Online]. Available: https://www.kaggle.com/mlg-ulb/creditcardfraud

[2] "Hyperledger Fabric – Hyperledger" (2020). [Online]. Available: https://www.hyperledger.org/use/fabric. [Accessed 5-October-2020].

[3] "Random Forest Regressor" (2021) [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html#sklearn.ensemble.RandomForestRegressor

[4] Bishop, C.M. (2006). Pattern recognition. Machine learning 128, no. 9.

[5] Breiman, L. (2001). Random Forests. Machine Learning, 45 (1), pp. 5-32.

[6] De Marchi, S., Schaback, R., Wendland, H. (2005). Near-optimal data-independent point locations for radial basis function interpolation, Adv. Comput. Math. 3, 23, 317-330.

[7] Gaur, N., Desrosiers, L., Ramakrishna, V., Novotny, P., Baset, S.A., and O'Dowd, A. (2018). Hands-on Blockchain with Hyperledger: Building decentralized applications with Hyperledger Fabric and Composer. Packt Publishing.

[8] INFINITECH project, Deliverable D4.7 "Permissioned Blockchain for Finance and Insurance - I"

[9] INFINITECH project, Deliverable D4.8 "Permissioned Blockchain for Finance and Insurance - II"

[10]INFINITECH project, Deliverable D4.13 "Encrypted Data Querying and Personal Data Market – I"

[11]INFINITECH project, Deliverable D4.14 "Encrypted Data Querying and Personal Data Market – II"

[12]INFINITECH project, Deliverable D4.10 "Blockchain Tokenization and Smart Contracts – I"

[13]INFINITECH project, Deliverable D4.12 "Blockchain Tokenization and Smart Contracts - III"

[14]Jolliffe, I.T. (2002). Principal Component Analysis, second edition, New York: Springer-Verlag New York.

[15]Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R.G.L., El Rouayheb, S., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P.B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konecný, J., Korolova, A, Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Raykova, M., Qi, H., Ramage, D., Raskar, R., Song, D., Song, W., Stich, S.U., Sun, Z., Suresh, A.T., Tramér, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F.X., Yu, H., and Zhao, S. (2021). Advances and open problems in federated learning. Foundations and trends in machine learning, 14 (1-2), pp. 1-210.

[16]Long G., Tan Y., Jiang J., Zhang C. (2020) Federated Learning for Open Banking. In: Yang Q., Fan L., Yu H. (eds) Federated Learning. Lecture Notes in Computer Science, vol 12500. Springer, Cham.

[17]McMahan, H. B., Moore, E., Ramage, D., Hampson, S., Aguera y Arcas, B. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data, AISTAT.

[18]Oehmichen, A., Jain, S., Gadotti, A., and de Montjoye, Y.-A. (2019). OPAL: High performance platform for large-scale privacy-preserving location data analytics. In Proceedings of BigData 2019: pp. 1332-1342.

[19]Pinkas, B., and Lindell, Y. (2009). A proof of security of Yao's for two-party computation. J Cryptol 22, 161-188.

[20]Rasmussen, C. E., Williams, C. K. I. (2006). Gaussian Processes for Machine Learning, The MIT Press.

[21]Sagi, O., and Rokach, O. L. (2018). Ensemble learning: A survey, WIREs Data Mining and Knowledge Discovery.

[22] Zyskind, G., Nathan, O., and Pentland, A. (2015). Decentralizing privacy: Using blockchain to protect personal data. In Proceedings of IEEE Symposium on Security and Privacy Workshops: 18.