Computing (2025) 107:3
https://doi.org/10.1007/s00607-024-01364-0

REGULAR PAPER

Simulafed: an enhanced federated simulated environment for privacy and security in health

Jose M. Rivas[1,2] · Carlos Fernandez-Basso[1,2] · Roberto Morcillo-Jimenez[1,2] · Juan Paños-Basterra[1,2] · M. Dolores Ruiz[1,2] · Maria J. Martin-Bautista[1,2]

Received: 19 September 2024 / Accepted: 22 October 2024 / Published online: 20 November 2024

Abstract

Federated learning enables collaborative data analysis without the need to share sensitive information among participants, addressing privacy concerns in domains such as healthcare and finance. However, current federated simulation environments face challenges like limited flexibility in experiment configuration and difficulties ensuring data privacy. We present SimulaFed, a federated simulation environment based on a custom architecture that offers a personalised and configurable approach to data analysis using the Docker platform. SimulaFed allows researchers to create experiments tailored to their specific needs, ensures communication privacy, and incorporates various security and governance techniques. We demonstrate the effectiveness of SimulaFed through a real-world medical case, implementing and comparing two privacy-preserving federated algorithms for association rule mining: Tassa's and Chahar's algorithms. Our experiments show that while Tassa's algorithm performs better in environments with a moderate number of participants due to lower computational and communication overhead, Chahar's algorithm, though offering robust security through homomorphic encryption, suffers from efficiency limitations owing to high encryption and decryption costs. These findings provide valuable insights into the performance and limitations of existing algorithms, highlighting the need for more efficient methods in federated settings.

Keywords Federated learning · Simulation environment · Data privacy · Scalable platform · Application in healthcare data

Mathematics Subject Classification 68T01 · 68T09 · 68T10 · 68U35 · 68W15

Extended author information available on the last page of the article

1 Introduction

As the digital era progresses, the importance of data privacy and security has escalated. Federated Learning (FL) offers a transformative solution by enabling data analysis and machine learning without the need to share the data itself. Globally, growing concerns about data privacy have accelerated the adoption of federated learning techniques. In the healthcare sector, hospitals and research institutions across different countries want to collaborate on medical research and patient care analytics without violating strict privacy regulations such as the European Union's General Data Protection Regulation (GDPR) [1] and the United States' Health Insurance Portability and Accountability Act (HIPAA) [2]. For instance, collaborative studies on rare diseases require the pooling of data from different locations around the world, but the sharing of sensitive patient information is severely restricted [3]. Federated learning allows these institutions to train models on decentralised data while maintaining patient confidentiality.

Similarly, in the financial industry, international banking institutions need to detect fraud and money laundering activities that span multiple countries and jurisdictions. Sharing customer transaction data across borders raises significant privacy and compliance concerns [4]. Federated learning enables these organisations to collaboratively analyse patterns of fraudulent activity without exposing sensitive customer data, improving security while complying with privacy regulations.

However, a significant challenge remains in accurately simulating real-world scenarios in which FL could be applied. Although current methods provide a foundation, they often fail to fully capture the complexity of real-life data interactions. There is a clear need for a simulation environment that offers advanced features such as adaptability, support for both supervised and unsupervised learning, a comprehensive simulation suite, a flexible API, and open-source accessibility, all while prioritising data privacy and security [5].

This paper aims to describe such an environment. Our proposed solution focuses on creating an adaptable simulation platform that combines the current Federated Learning simulation needs and is designed for future challenges. This research is not limited to compare existing frameworks, but lays the foundations for a base environment in which further simulations of real-world characteristics can be built and improved. The scope of this work includes the design and development of the simulation environment and the provision of initial simulation results to demonstrate its potential.

Our proposal, called SimulaFed, makes several necessary contributions to the field of Federated Learning. Development of an adaptable simulation platform that can be easily customised to replicate diverse real-world scenarios, accommodating different data distributions and participant configurations. It gives support for both supervised and unsupervised learning, including the implementation of association rule mining techniques; the ability to incorporate comprehensive security measures such as Secure Multi-Party Computation (SMC), Differential Privacy (DP), and Homomorphic Encryption (HE) (see Sect. 4.3).

SimulaFed ensures robust privacy protection while offering a highly flexible open-source API layer that simplifies the integration of new algorithms and methods, fostering further research and development. Moreover, it improved scalability and real-world applicability by using Docker and Docker Compose (see Sect. 4), thus facilitating the transition from experimental setups to practical applications in different domains. Together, these contributions extend the capabilities of the Federated Learning simulation and provide a robust foundation for future advances in privacy-preserving collaborative data analysis.

Our work represents a significant advancement in federated learning simulation environments. SimulaFed is a flexible and algorithm-agnostic framework, enabling researchers to implement their own algorithms or seamlessly integrate third-party algorithms. Unlike many existing frameworks, SimulaFed supports federated learning of unsupervised algorithms, as demonstrated by our implementation of Association Rule Mining (ARM) algorithms. This capability is essential, as unsupervised learning plays a vital role in deriving insights from unlabelled data, which is prevalent in real-world applications.

Our framework also enables dynamic resource allocation, along with the ability to dynamically adjust the number of participants in an experiment. This flexibility allows for the simulation of different configurations, accurately refleting different real-world scenarios. Furthermore, SimulaFed supports both centralised and decentralised governance architectures to accommodate different organisational requirements and collaboration models.

By implementating and comparing Tassa's and Chahar's privacy-preserving ARM algorithms within SimulaFed, we have gained valuable insights into their performance and practical deployment considerations. Our findings highlight the strengths of our framework in supporting such implementations and underscore the importance of flexibility and configurability in federated learning environments. These findings contribute to a deeper understanding of how different algorithms function in federated settings, laying a solid foundation for future developments in this field.

This work presents a tool that allows us to perform different experiments in federated environments with a wide range of parameters, integrating different techniques available in the literature used to guarantee privacy and confidentiality in federated environments. The contributions of this study are as follows:

1. A review of some prominent frameworks in federated learning and their respective strengths and weaknesses is conducted, comparing our tool against different criteria.
2. A tool called SimulaFed is proposed, which is parameterisable and capable of simulating a federated system by integrating a wide configuration, allowing the researcher to perform various experiments with different parameters quickly and agilely.
3. An overview of the system's functionality is provided through its various components, demonstrating its application in a real-world healthcare context. We analyse

the scalability and performance of the developed simulation system through different configurations.

By thoroughly analysing existing frameworks and identifying their limitations (as detailed in Tables 1 and 2), our work advances the state of the art by providing a federated learning simulation environment that combines flexibility, scalability, and support for unsupervised learning algorithms. SimulaFed's capacity to dynamically allocate resources and adjust participant configurations enables researchers to model and simulate a wide range of real-world scenarios, addressing a critical gap in current federated learning research tools.

While this paper focuses primarily on the healthcare, the challenges addressed by SimulaFed are prevalent across multiple industries. Data privacy concerns and the need for collaborative analysis without sharing sensitive information are critical in numerous sectors. By providing a flexible and secure federated learning environment, SimulaFed has the potential to transform data collaboration practices across various domains.

The paper is organised as follows: Sect. 2 explains the fundamental concepts of federated learning and the challenges associated with its simulation. Section 3 reviews existing work in the field, identifying areas for improvement. Section 4 describes the SimulaFed architecture, security measures, and deployment strategies. Section 5 presents the results of our experiments with a health real-world use case, validating the benefits of our approach. Section 6 concludes the paper with a summary of our findings and potential directions for future research.

2 Fundamentals and applications of federated learning

This chapter establishes the basic context and importance of FL, especially in the healthcare sector. Here we explain what FL is, its general benefits and challenges, and how it is specifically applied in the healthcare sector. This chapter prepares the reader to understand why we need specific tools such as frameworks and simulation environments to carry out FL.

2.1 Introduction to federated learning (FL)

Federated Learning (FL) is a collaborative machine learning paradigm that enables multiple clients to learn a global model without exposing their data to each other [6]. It is particularly useful in scenarios where data privacy is a concern, such as in healthcare [7], finance and banking [8], telecommunications [9], manufacturing and industry 4.0 [10], SmartCities and IoT [11], Retail [12], and Energy [13]. The process involves clients training models locally and then sending weight updates to a central server, which aggregates these updates to create a global model [14].

FL emerges as a paradigm-shifting approach in the landscape of machine learning, where the training process is distributed across multiple devices or servers. Rather than pooling data into a central repository, FL allows for the model to be

Table 1 Comparison of Federated Learning Frameworks

Framework	Communication	Governance	Supported learning types	Dynamic resource allocation	Flexibility and configurability	Limitations
TensorFlow federated	Built-in security protocols (TLS); limited custom security	Centralised	Supervised	No	Limited (TensorFlow only; limited custom algorithms)	Only supervised learning; limited flexibility; no dynamic resource allocation
FATE	Built-in security protocols (TLS); supports homomorphic encryption; limited custom security	Centralised	Supervised, Unsupervised	No	Medium (supports custom algorithms; some ML libraries)	Centralised only; complex setup; no dynamic resource allocation
Flower	Built-in security (TLS); allows adding custom security protocols	Centralised and Decentralised	Supervised, Unsupervised	No	High (supports multiple ML libraries; custom algorithms)	No dynamic resource allocation
PySyft + PyGrid	Built-in security protocols; supports homomorphic encryption; allows custom security protocols	Decentralised	Supervised	No	Medium (custom algorithms; limited integration)	Only supervised learning; no dynamic resource allocation; setup complexity
OpenFL	Built-in security (TLS); limited custom security	Centralised	Supervised	No	Limited (some integration)	Only supervised learning; limited flexibility; no dynamic resource allocation
IBM Federated learning	Built-in security within IBM ecosystem; limited custom security	Centralised	Supervised	No	Limited (IBM ecosystem only)	IBM-only; only supervised learning; limited flexibility; no dynamic resource allocation

Table 1 (continued)

Framework	Communication	Governance	Supported learning types	Dynamic resource allocation	Flexibility and configurability	Limitations
NVIDIA Clara	Built-in security (TLS); supports homomorphic encryption; limited custom security	Centralised	Supervised	No	Limited (healthcare-specific)	Proprietary; healthcare-specific; only supervised learning; limited flexibility; no dynamic resource allocation
Substra	Built-in security using blockchain; allows custom security protocols	Decentralised	Supervised	No	Medium (custom algorithms; complex integration)	Only supervised learning; complex setup; no dynamic resource allocation
SimulaFed	Security protocols and encryption guided by algorithm	Centralised and Decentralised	Supervised, Unsupervised	Yes	High (supports any ML library; custom algorithms; configurable nodes)	Higer latency in the communication system due to the unsupervised algorithm

Table 2 Comparison of Federated Learning Simulation Frameworks

	LEAF [27]	PySyft + PyGrid [46]	FedML [47]	TFF [29]	SimulaFed
Realism and Precision	High	Medium	Low	Medium	High
Flexibility and Configurability	High	High	Medium	Medium	High
Performance	Medium	Medium	Medium	Medium	High
Interoperability	Medium	High	Medium	High	High
Validation and Verification	Medium	Low	High	Medium	Medium
Security and Privacy	Medium	High	High	Medium	High
Visualisation	Low	Low	Low	Low	Low
Reproducibility	High	Medium	High	Medium	Medium
Extensibility	Medium	High	High	High	High
Dynamic Resource Allocation	No	No	No	No	Yes
Supported Learning Types	Supervised	Supervised	Supervised	Supervised	Supervised, Unsupervised

trained locally on users' devices, with only model updates being communicated to a central server. This not only preserves bandwidth but also enhances users' privacy by not transferring sensitive data over the network. The core principle of FL is to maintain data sovereignty and privacy, thereby reducing the risk of data exposure and breach. Despite these benefits, FL presents unique security and privacy challenges, as the distributed nature of the model training process can expose the system to new attack vectors.

2.2 Federated learning in the healthcare sector

Federated learning (FL) has emerged as a transformative approach in the healthcare sector, addressing critical challenges related to data privacy, security, and the need for large, diverse datasets. Traditional centralised machine learning models require data to be aggregated in a single location, which poses significant privacy risks, especially in healthcare where patient data is highly sensitive [15, 16]. FL mitigates these concerns by enabling the training of machine learning models across distributed datasets without the need to share raw data, thus preserving patient privacy [17, 18].

As far as we know and have been able to review, there are no specific federated learning systems applied to health, but there are certainly applications of federated systems applied to health. One of the primary applications of FL in healthcare is in medical image analysis and human behaviour recognition, where it allows for the development of robust models without compromising data privacy [15]. For instance, in the context of brain tumour segmentation, FL has been successfully implemented to train models on data from multiple healthcare centres located in

different countries, demonstrating its capability to handle real-time distributed networking and maintaining high accuracy while preserving privacy [16].

FL is also crucial in the realm of mobile health (mHealth) applications, where data is often siloed and patients are concerned about privacy implications. By enabling collaborative training of models without sharing raw data, FL facilitates remote monitoring, diagnostic support, and treatment planning, thereby enhancing the quality of healthcare services [19]. This is particularly important for monitoring self-care ability, health status, and disease progression in patients using sensor devices [19].

In addition to these applications, FL has been leveraged to predict heart diseases by integrating IoT-generated health data with Electronic Health Records (EHRs). This approach not only improves the accuracy of disease prediction but also ensures data privacy, encouraging broader participation from healthcare providers [20, 21]. The use of advanced algorithms like the soft-margin L1-regularised Support Vector Machine (SVM) further enhances the computational efficiency and scalability of these predictive models [20].

Moreover, the potential of FL extends to collaborative frameworks that combine blockchain technology to ensure secure and trusted data aggregation. For example, the HealthFed framework leverages FL and blockchain to enable privacy-preserving and distributed learning among multiple clinician collaborators, ensuring the secure aggregation of local model updates [22].

Despite its promising applications, FL in healthcare still faces challenges such as latency, security risks, and the need for robust communication protocols. Addressing these issues is crucial for the widespread adoption of FL in healthcare settings [17, 23]. Nonetheless, the ability of FL to maintain data privacy while enabling the development of accurate and generalizable models makes it a valuable tool in advancing healthcare technologies and improving patient outcomes [23].

3 Evaluation and comparison of federated learning frameworks and simulation environments

As the field of Federated Learning advances, selecting the most suitable framework and simulation environment has become essential for effective research and development. A framework in FL provides the tools and libraries needed to implement and execute distributed learning algorithms while ensuring data remains on local devices. Examples include TensorFlow Federated (TFF), FATE, and PySyft + PyGrid [4, 24, 25].

In contrast, a simulation environment is a controlled setting that allows researchers to test and optimise these algorithms under various conditions before deployment. These environments are crucial for replicating real-world scenarios, managing diverse data distributions, and ensuring the validity of simulation results [26, 27]. Examples include LEAF and FedML.

Frameworks facilitate the practical implementation and execution of FL algorithms, while simulation environments allow thorough testing and optimisation in

controlled conditions. It is crucial that these tools enable a seamless transition from simulation to production to ensure the scalability and efficiency of FL systems.

In the previous chapter, we discussed the fundamental principles and technical challenges of FL, emphasising the critical role of robust simulation environments in overcoming these challenges. Building on this foundation, this chapter evaluates existing FL frameworks and simulation environments to determine their suitability for various research and application scenarios. The goal is to identify the most effective tools and highlight the need for advanced solutions like SimulaFed, which can overcome current limitations and better support the diverse requirements of FL research.

By conducting this evaluation, we aim to provide a clear understanding of the available options and their respective advantages and disadvantages. This will enable researchers and practitioners to make informed decisions when selecting frameworks and simulation environments, ultimately enhancing the effectiveness and impact of their FL projects.

3.1 Characteristics desirable in federated learning frameworks

In evaluating frameworks for Federated Learning (FL), it is essential to establish criteria that reflect the practical needs and challenges encountered in real-world applications. The selected criteria are based on their relevance to performance, security, and usability in a federated context.

Effective *communication* is crucial for FL frameworks, as it directly impacts the efficiency and security of the learning process. This involves ensuring *security* through adherence to established standards like TLS and end-to-end encryption to maintain data integrity and confidentiality during transmission [4]. Additionally, *performance* is essential, requiring high-performance, low-latency communication methods achieved through advanced protocols and optimised data handling techniques [28]. Furthermore, the communication architecture should exhibit *modularity and flexibility*, allowing adaptation to various protocols and requirements [26]. Lastly, *interoperability* is vital for seamless communication across different frameworks and platforms, ensuring the FL framework can operate effectively in diverse environments [25].

Governance refers to the way the learning process is managed and coordinated within the framework. Effective governance ensures that the framework can handle various organisational structures and policies. This involves *centralised governance*, where control is managed by a central entity, simplifying management but potentially creating a single point of failure and scalability issues. Alternatively, *decentralised governance* distributes control among multiple entities, increasing robustness and scalability but adding complexity to coordination and communication [24].

Supported Learning Types refers to the framework's ability to support different types of learning tasks, which enhances its versatility. This includes *supervised learning*, which is essential for tasks where the training data includes labels and the desired output is known, and *unsupervised learning*, which is crucial for discovering hidden patterns in unlabelled data, expanding the framework's applicability [4].

Dynamic Resource Allocation is vital for optimising performance and scalability in FL frameworks. Efficient resource management ensures optimal performance and load balancing across multiple nodes. This capability enhances scalability and overall system performance, as the framework should dynamically manage computational resources like CPU shares across multiple nodes [25, 26].

These criteria provide a comprehensive basis for evaluating FL frameworks, ensuring that they meet the essential requirements for real-world deployment.

3.2 Evaluation of specific federated learning frameworks

To provide a comprehensive overview of available Federated Learning (FL) frameworks, we evaluate several prominent frameworks based on the criteria established in the previous section. This evaluation highlights the strengths and limitations of each framework in terms of communication, governance, supported learning types, dynamic resource allocation, and overall limitations, specifically focusing on their adequacy as simulation environments.

TensorFlow Federated (TFF) is a framework developed by Google for experimenting with FL using the TensorFlow infrastructure. It offers a secure and flexible API but is primarily suited for development rather than production applications. Additionally, TFF is limited to centralised governance and only supports supervised learning [29]. While TFF excels in secure communication and ease of use, it lacks comprehensive simulation tools and support for unsupervised learning tasks.

FATE (Federated AI Technology Enabler), developed by WeBank, excels in high-performance and secure computing, offering a modular architecture for scalability. It supports both supervised and unsupervised learning. However, its centralised governance and lack of comprehensive simulation tools limit its flexibility and adaptability [30]. FATE's robust security measures are offset by its limitations in providing detailed simulation capabilities.

Flower Framework is highly flexible and platform-agnostic, allowing integration with various machine learning systems. Its modular design supports high scalability, but its simulation capabilities are limited. Flower can adapt to both centralised and decentralised governance, supporting supervised and unsupervised learning [31]. Despite its adaptability, Flower's limited simulation tools hinder its application in complex scenarios.

PySyft + PyGrid combines WebRTC and Websockets for secure communication, providing high adaptability and a decentralised governance model. While it supports supervised learning, it has limitations in unsupervised learning support and lacks a comprehensive simulation environment [32]. The framework's strong security features are marred by its insufficient simulation capabilities.

Open Federated Learning (OpenFL) supports interoperability between TensorFlow and PyTorch. It offers a basic simulation environment and centralised governance, suitable for supervised learning tasks. However, it is limited in handling more complex or unsupervised learning scenarios [33]. OpenFL's basic simulation tools limit its effectiveness in detailed FL studies.

IBM Federated Learning provides secure and highly scalable solutions primarily for the IBM ecosystem. Its main limitation is the lack of a real-use simulation environment, restricting its application to development and testing within the IBM infrastructure [34]. IBM's strong security and scalability are offset by its inadequate simulation capabilities.

NVIDIA Clara offers encrypted and secure communication with high scalability. However, its proprietary nature and healthcare-specific design limit its adaptability across different sectors and its use for unsupervised learning tasks [35]. Clara's proprietary nature restricts its broader applicability and simulation scope.

Substra uses blockchain for transparency and security, being highly scalable but complex to integrate. While it supports supervised learning well, it lacks a comprehensive simulation environment and faces challenges in supporting unsupervised learning [36]. Substra's blockchain integration adds complexity and limits its simulation capabilities.

In summary, although these frameworks offer valuable features, as shown in Table 1, they present significant limitations regarding simulation capabilities. This evaluation underscores the need for advanced simulation environments to address these shortcomings, which will be examined in detail in a series of critical limitations.

As summarised in Table 1, existing federated learning frameworks often lack support for unsupervised learning algorithms and dynamic resource allocation. For instance, while TensorFlow Federated [29] and PySyft + PyGrid [32] provide secure environments, they are limited to supervised learning and do not offer comprehensive simulation capabilities. SimulaFed fills these gaps by supporting both supervised and unsupervised learning types, including association rule mining algorithms, and by allowing dynamic adaptation of resources and participants.

We are faced with *lack of detailed and precise simulation capabilities*, where many frameworks do not provide comprehensive tools for simulating real-world FL scenarios. There exist *limitations in flexibility and configurability*, often a lack of adaptability to different scenarios and research needs, which hampers detailed experimentation. Another important limitation is the *difficulties in interoperability*, which presents a challenge in integrating with other tools and systems and limits the versatility of the frameworks.

The *need for better visualisation and validation tools* requires creating enhanced tools for the visualisation and validation of results to achieve thorough analysis and debugging. Most frameworks do not support dynamic management of computational resources, which affects scalability and performance. *Dynamic resource allocation not supported* is one of the limitations of this type of system. The last key detachable is the *limitations in support for unsupervised learning types*, where many frameworks are primarily designed for supervised learning, with limited support for unsupervised learning tasks.

3.3 Desirable characteristics in federated learning simulation environments

In order to create a robust and effective simulation environment for Federated Learning (FL), it is essential to identify and integrate key characteristics that address the

unique challenges and requirements of FL. Below, we discuss these desirable characteristics, providing justifications for their importance.

Realism and Precision A simulation environment must accurately replicate the conditions and complexities of real-world FL scenarios. This includes handling diverse data distributions, participant configurations, and network conditions to ensure the validity and applicability of the simulation results [28].

Flexibility and Configurability The ability to easily configure and adjust the simulation environment to reflect various real-world scenarios is crucial. This includes supporting different data distributions (IID[1] and non-IID[2]), participant setups, and varying resource allocations [26, 37].

Performance High-performance simulation environments can efficiently handle large-scale simulations with numerous participants and extensive data sets. This includes optimising resource utilisation and ensuring minimal latency and overhead during simulations [25, 38].

Interoperability The simulation environment should seamlessly integrate with various FL frameworks and other machine learning tools. This ensures flexibility and ease of use across different platforms and systems [39].

Validation and Verification Robust mechanisms for validating and verifying the results of simulations are necessary to ensure their accuracy and reliability. This includes providing tools for debugging, testing, and performance evaluation [40].

Security and Privacy Given the sensitive nature of the data often used in FL, robust security and privacy measures must be integrated into the simulation environment. This includes techniques such as Secure Multi-Party Computation (SMC), Differential Privacy (DP), and Homomorphic Encryption (HE) to protect data throughout the simulation process [4, 41–43].

Visualisation Effective visualisation tools are essential for interpreting and analysing the results of simulations. This includes dashboards, graphical representations of data flows, and real-time monitoring of simulation progress [44].

Reproducibility To ensure the scientific validity of simulation studies, the environment must support reproducibility. This includes providing mechanisms for saving and sharing simulation configurations, datasets, and results [45].

Extensibility The simulation environment should be designed to allow for easy extension and adaptation to future requirements and advancements in FL. This includes modular architectures and open-source frameworks that facilitate community contributions and ongoing development [39].

In the next section, we will evaluate existing simulation environments for FL based on these criteria, highlighting their strengths and limitations in addressing the needs of FL research and practice.

[1] Independent and Identically Distributed.

[2] Non-Independent and Identically Distributed.

3.4 Evaluation of specific simulation environments

The environments selected for this evaluation are widely used in federated learning research and have publicly available code, ensuring their representativeness and accessibility. These environments were chosen because they offer a broad perspective on the capabilities and limitations present in current FL simulation tools. The results of this evaluation are summarised in Table 2.

Moreover, as shown in Table 2, our framework offers high flexibility and configurability, interoperability, and support for both centralised and decentralised governance models. This versatility distinguishes SimulaFed from other frameworks, facilitating the simulation of diverse real-world scenarios and fostering extensive research opportunities.

From the evaluation, several key limitations in current simulation environments are identified. As shown in Table 2, most frameworks lack detailed and precise simulation capabilities. Additionally, there are significant limitations in flexibility and configurability to adapt to different scenarios. The interoperability with other tools and systems also poses challenges, and there is a need for better visualisation and validation tools. Dynamic resource allocation is not supported by most frameworks, and there are limitations in support for unsupervised learning types.

LEAF offers high flexibility and configurability, but falls short in visualisation and dynamic resource allocation. *PySyft + PyGrid* is strong in security and privacy, but limited in ease of use and lacks comprehensive simulation tools. *FedML* has high flexibility and configurability and is limited in visualisation capabilities. *TFF* integrates well with TensorFlow, but lacks support for unsupervised learning and dynamic resource allocation.

Once we have analysed the various frameworks from the literature, we will analyse our tool, SimulaFed, in the next section, covering all the features it maintains.

4 SimulaFed framework

In the field of Federated Learning (FL), both a robust framework and a comprehensive simulation environment are crucial. A framework facilitates the practical implementation and execution of FL algorithms, while a simulation environment allows for thorough testing and optimisation in controlled conditions. It is essential that these tools enable a seamless transition from simulation to production to ensure the scalability and efficiency of FL systems.

Our comparative analysis reveals that existing FL frameworks, despite their valuable features, have significant limitations, particularly in simulation capabilities, flexibility, and support for advanced learning types. SimulaFed addresses these gaps by offering a comprehensive set of features tailored to the evolving needs of FL research and practice.

SimulaFed provides secure and robust *communication* by integrating different algorithms within the framework. Initially, the Chahar [48] and Tassa [49] algorithms have been integrated, allowing the communication between nodes to the integrated federated learning algorithms uses a cutting-edge homomorphic

encryption-based approach, allowing operations to be performed on encrypted data without decryption. This advanced technique ensures the privacy of local data during the aggregation and updating process of the global model.

It also integrates *governance* by implementing a centralized communication protocol based on message authentication, where a miner node and a combiner node manage the governance of the system, allowing the encryption and decryption of keys to send data to different nodes. It also implements a protocol for decentralized governance among nodes using multi-party computation techniques, where nodes collaborate to compute a common function without revealing their private data.

As an *open-source platform*, SimulaFed encourages collaboration and continuous improvement, making it an ideal tool for researchers aiming to advance the field of FL. By bridging the gaps identified in existing frameworks, SimulaFed provides a secure, adaptable, and scalable environment for both simulation and production, supporting the diverse requirements of FL research.

SimulaFed supports both *supervised and unsupervised learning*, providing flexibility for diverse machine learning tasks. Its detailed simulation environment allows for extensive testing and fine-tuning of various parameters, supporting the simulation of up to 100 nodes. This scalability is crucial for conducting exhaustive tests and understanding the performance of FL algorithms under different conditions.

One of SimulaFed's standout features is advanced *dynamic resource allocation*, ensuring optimal performance and load balancing across multiple nodes for efficient large-scale simulations. Our tool allows users to adjust and distribute resources because the execution of the different nodes is carried out through containers.

While developing SimulaFed, we acknowledge that leveraging the full potential of the platform may require adequate computational resources, especially when simulating environments with a very large number of nodes. However, this is a common consideration in distributed systems and depends on the hardware on which SimulaFed is deployed rather than a limitation of the platform itself.

Simulated environments are virtual representations of real-world systems, realities, or scenarios created to experiment, train, research, or develop new ideas in a controlled and safe environment. These environments are used in various disciplines and sectors, including education, medicine, engineering, defence, scientific research, and entertainment. We have implemented a simulation environment based on federated computational models capable of creating a system that represents a federated environment.

The degree of *realism and precision* can vary from simplified models with two nodes to highly complex simulations with multiple nodes that can interact while maintaining data privacy through governance. SimulaFed allows researchers to interact with the simulated environment by configuring files that enable experiments based on modified environmental variables. It gives the system the capability of *flexibility and configurability*. We aim to enhance its adaptability and robustness. The *performance* of SimulaFed is enhanced through the integration with Docker, which provides high scalability and flexibility, limited only by the computational resources of the host machine.

One of the essential features of SimulaFed is its support for Security and Privacy. While the framework provides basic security inherent in federated learning-since

data remains on local nodes and only model updates are shared-it allows users to implement advanced security techniques within their algorithms. Techniques such as Secure Multi-Party Computation (SMC), Differential Privacy (DP), and Homomorphic Encryption (HE) can be integrated by users to enhance data protection throughout the simulation process. This flexibility enables users to tailor the security measures to their specific needs and research goals. In future work, we plan to include implementations of these advanced security techniques within SimulaFed, providing users with ready-to-use tools to enhance data protection in their simulations.

Additionally, SimulaFed supports different data distributions and topologies, allowing users to simulate various real-world scenarios. The framework permits the implementation of algorithms like those proposed by Chahar [48] and Tassa [49], which enable the selection of different data distribution strategies, such as Independent and Identically Distributed (IID) and Non-Independent and Identically Distributed (non-IID) executions. This capability enhances the realism and applicability of the simulations."

SimulaFed has a *validation and verification* system that currently performs a simple verification of the execution process, ensuring that all nodes have executed correctly and providing a warning if they have not. The *reproducibility* in this environment allows for reduced enforced repetition of various experiments by modifying configurations. SimulaFed has a logging system that records the executions of different experiments.

The flexible API design of SimulaFed simplifies the integration of new algorithms, enhancing its *interoperability and extensibility*. As an API-REST service, it also supports the creation of microservices capable of communicating with other frameworks. Although it is not currently connected to any, it is ready to do so at any time.

Finally, SimulaFed has, in the field of simulated environments, a series of *limitations* that will be addressed in future research, such as *visualisation* aspect has yet to be essayed, but we will focus on implementing a user-friendly interface that comforts our users.

In summary, simulated environments provide a safe and controlled means for experimentation, learning, and development. Next, we will detail the architecture of the SimulaFed system.

4.1 Overview of the architecture

In a Federated Learning environment, the deployment and interaction of nodes with other participants can be a complex task. With a dynamic number of participants and a wide array of resources required by each node, implementing a system of these characteristics with Docker, Docker Compose, and Dockerfile offers us the flexibility to adjust different characteristics of our system, making it more adaptable and efficient.

Docker's containerisation technology plays a crucial role in our federated learning environment. It allows us to encapsulate each participating node as an independent, isolated entity, mirroring the autonomy of real-world entities in a

federated setup. This emulation ensures efficient resource allocation and replicates the collaborative nature of federated data analysis, making our system more efficient and robust.

The Docker Compose configuration file serves as the backbone of our environment, efficiently deploying nodes and managing communication channels. By encapsulating these aspects in the Docker Compose configuration, the deployment of our system and communication channels are efficiently managed, contributing to a cohesive and functional federated environment. The Docker Compose technology thus enables the quick and simple deployment of the required containers with the specified parameters that will adjust to the requirements of our federated environment experimental setting. It should also be noted that Docker Compose sets up a single network for the framework. Each container for a service joins the default network and can be both reachable by other containers on that network and discoverable by them at a hostname identical to the container name, which significantly facilitates inter-container communication.

The Dockerfile of each participating node is instrumental in encapsulating their functionalities and dependencies. These files define the environment within which each node operates, ensuring consistency across the distributed architecture. By specifying the base image, installing necessary packages, and incorporating essential scripts, the Dockerfile streamlines the setup process for each node. This approach not only simplifies deployment, but also enables reproducibility and scalability. The Dockerfile of every participant node can be analysed, allowing the specification of their dependencies. It is important to note that the requirements file will vary from node to node, and since each role is associated with specific tasks, their code dependencies will vary. For instance, in some algorithms like the one proposed in [48], there are nodes with special roles like Miner and Combiner, that involve specific tasks of the algorithm.

By emulating a federated environment using Docker, Docker Compose and Dockerfile, we provide a practical and scalable platform for testing and refining our system, enabling seamless validation of a federated algorithm. These technologies also transition from an experimental setup like the one we have implemented to a more real-world setting since the containerisation technology of Docker enables the simple deployment of containers to the desired hosts without requiring extensive overhauls.

4.2 System configuration

The collaborative dynamics of a federated simulated environment are implemented through a network of participating nodes. Each node represents an independent participant fulfilling a role (see Fig. 1), contributing to the collective data analysis process. These nodes are designed to emulate the real-world scenario of distributed entities collaborating towards a common goal. We will now cover their distinct roles, functionalities, and interactions that can appear in a federated learning/mining process.

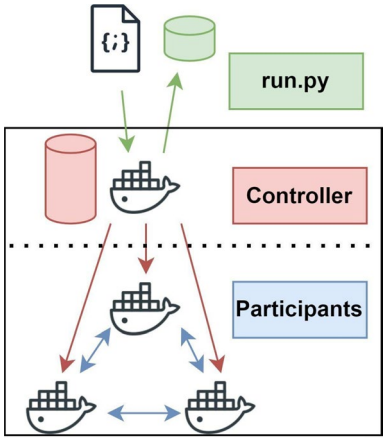

Fig. 1 Architectural view of an implemented system

4.2.1 Controller

In a real-world environment, data is already present in the nodes and can follow the characteristics explained in Section 2.3.1; however, in the experimental setting, data needs to be added to the nodes, generally by splitting a preobtained dataset. This opens up a wide array of possibilities, from the origin of the data to how it is distributed among the nodes.

One type of distribution is *Independent and Identically Distributed (IID)*, which means that each sample in a data set is drawn from the same probability distribution and is statistically independent of the other samples. This implies that the data at each node is representative of the overall data distribution. In practice, however, data in federated systems is often *Non-Independent and Identically Distributed (non-IID)*, meaning that the data is not uniformly distributed across the participating nodes, either in terms of quantity, data labels, or feature space. For example, different clients may collect data in different contexts or environments, resulting in different distributions; and/or data is not independent. For example, data collected by a single client may be correlated because it comes from the same user or device.

Additionally, in FL, model updates are typically exchanged between the central server and the participating devices during training. However, when dealing with many devices, the communication overhead becomes a significant concern. Transferring model parameters or gradients from each device to the central server and aggregating them can be time-consuming and resource-intensive. This bottleneck requires efficient communication protocols and strategies to minimize network traffic and latency, such as compression techniques, selective device participation, or hierarchical aggregation schemes.

These settings can result in interesting insights when comparing how our federated environment performs with diverse data distributions. With these requirements in mind, a container named `db controller` is created with Docker to allow the user to manage the data distribution before starting the algorithm process. The main tasks that the user can configure are the following:

- *Dataset selection* The user can visualise the datasets that are available to load them into the participant nodes. The user selects the desired file based on the experiment they want to conduct. To select it, they must modify the `DEFAULT DATASET FILENAME` option in the `.env` file by specifying the full path. The user can add their own simply by moving the desired `.csv` into the designated `/datasets/` folder in the db controller.
- *Distribution type selection* Whether IID or non-IID, this ensures that the data distribution strategy is in line with the analytical objectives. This allows the user to replicate conditions similar to those found in the real world, where nodes' data will vary based on their location, collection tools and other factors. An example of this distribution could be a dataset containing different medical units with values *['Cardiology', 'Coronary Care', 'Dialysis', 'Gastroenterology', ⋯]*. The user could choose to distribute all instances of *'Cardiology'* data to one node and instances of *'Dialysis'* to another node. This would help researchers to highlight differences in data according to the selected category. Once the user has selected the splitting parameters in the `DEFAULT DATASET DISTRIBUTION` option in the `.env` configuration file, the `db controller` script scans the Docker Compose network and sends the data to the participants.

4.2.2 Participant nodes setting

The participant nodes simulate real-life entities participating in the federated process. These nodes also act as data holders, receiving the data using the distribution selected by the user.

In complying with the algorithms, the participating nodes compute the subprotocols necessary to obtain the desired results. It is important to note that participant nodes have different roles internally according to their container id (participants with ids 1, 2, and M may have distinct roles), and they can also scan the Docker Compose network to detect all other participants and their role in the learning/mining process. Data sharing and communication among participant nodes are pivotal for collaborative analysis in our enhanced federated environment. This section outlines how these crucial aspects are managed through the use of API calls. Each participant node is equipped with its API application, enabling it to send and receive data over HTTP seamlessly.

API calls enable the coordination and execution of actions between nodes on a network, using specific endpoints to transfer data and communicate parameters. Docker compose simplifies the deployment and management of containers, by creating a standard network that enables interaction between nodes and a scalable, manageable deployment process. To achieve this, our federated environment obtains the configuration through the `.env` file by modifying the parameters `N_PARTICIPANTS`, `PARTICIPANT_NAME`, `CONTROLLER_NAME`, and `DEFAULT_NETWORK_NAME`. In this way the system will create the participants needed for the experiment in question.

4.2.3 Script to run the experiments

An external script using Docker Compose is crucial in managing the experimentation process in the advanced federated environment setup. This script `experiment.py` is given a dictionary of parameter options representing different configurations for the federated experiments. It orchestrates the iterative execution of Docker Compose, dynamically modifying the environment variables in the `.env` file according to the specified parameters. By initiating and terminating experimental runs based on the provided options and exploring different settings such as the number of participants, transactional details, and algorithm variations. This approach allows efficient and automated testing of federated algorithms under diverse conditions, providing insights into their performance across different scenarios.

4.3 Security strategies

Our architecture is designed to be inherently flexible, allowing for the interplay of established privacy-preserving techniques used in the federated learning setting to protect sensitive data while allowing collaborative model training, such as Secure MultiParty Computation (SMC) [50, 51], Differential Privacy (DP) [52, 53] and cryptographic techniques [54–56]. Adapting the integration of additional protocols as required by evolving security landscapes and use cases within FL. This versatility is critical, as it enables our framework to support a wide range of privacy-preserving strategies and to adapt to new advancements in security technology, thus maintaining robust defence mechanisms in the face of ever-changing cyber threats and privacy concerns. Our framework is not prescriptive, but allows for the selection and integration of these techniques to create a bespoke security solution tailored to the unique requirements of each FL scenario.

Secure Multiparty Computation (SMC) ensures that during a computation, participants only know their inputs and the final results, without any knowledge about the inputs of other participants [50]. This can be achieved without relying on a trusted third party [51]. Instead, SMC utilises communication among the participants to perform the computation securely. The aim of SMC is to achieve the same outcome without needing a trusted third party. This involves establishing secure communication channels between the participants and executing cryptographic protocols to jointly compute the desired results while preserving the confidentiality of individual inputs. Among this type of techniques is the *secure sum* [50], which allows the computation of the sum of a given value among the different participating nodes. The *secure set union* [57] is used to share the standard sets of elements without revealing the contents, and a one-way commutative hash function is utilised in different techniqueslike the *secure size of set intersection* [48].

Differential privacy (DP) [58] is one of the main approaches proven to ensure strong privacy protection in data analysis [52]. DP protects the users' privacy by adding noise to the original dataset or the learning parameters [53]. Thus, an attacker could not retrieve an individual's sensitive information in the training dataset. A comprehensive overview of various attacks is provided in [59], highlighting

significant threats such as Membership Inference Attacks, Training Data Extraction, and Model Extraction

Our framework has also been designed to be adaptable to various cryptographic techniques, such as Homomorphic Encryption(HE) [48, 54], which allows computations to be performed on encrypted data without first decrypting the data, keeping it private and secure even in addition to multiplication computations [48]. Commutative Encryption [55] and Elliptic Curve Cryptography (ECC) [56, 60] provide layers of security for data during its transit and storage. Techniques like Secret Sharing [61] and Oblivious Transfer [62] can be seamlessly integrated to provide additional mechanisms to protect data from unauthorised access and collusion.

In summary, we can achieve secure communication between the nodes, allowing us to adapt the experiments to any desired federated algorithm. Currently, the Tassa [49] and Chahar [48] algorithms are incorporated, with the ability to configure the execution by dynamically assigning different resources and modifying the various parameters in the `.env` file of the system. By using Docker, we were able to configure the experiments to run on up to 100 nodes, allowing us to observe different behaviours of our system. Governance can be centralised or decentralised, depending on the distribution of communication among the different nodes involved in the experiment. SimulaFed has implemented a set of machine learning algorithms that support supervised and unsupervised learning problems. The implementation of the REST API allows us to have decentralised resources; but, certain limitations need improvement. These limitations are due to scalability issues related to hardware requirements that do not have high capacity, and to the parametrisation of the algorithms, which could be extended to include specific encryption techniques.

5 SimulaFed in a healthcare real-life setting

This section provides an overview of the system's functionality through its various components, demonstrating its application in a real healthcare context. Specifically, we describe an example involving the analysis of patient health data to extract relationships in medical records across various departments using association rules within the federated learning paradigm. It is important to note that sensitive data has been properly anonymised to ensure compliance with the General Data Protection Regulation (GDPR).

In Fig. 2, we depict the workflow of our use case. It focuses on two key components applied to a set of patient medical records: the processing of the dataset to execute the algorithm and the application of association rules within a federated learning environment.

5.1 Data sources

The datasets used for the experiments follow the structure of the Andalusian Minimum Basic Data Set (CMDB) [63]. They are based on Electronic Health Records (EHR) and contain health records of real patients in two hospitals, covering different

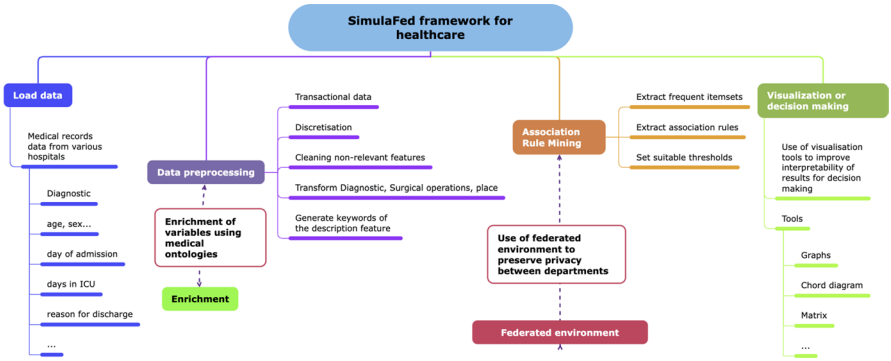

Fig. 2 Use Case workflow

departments (emergency, mental health, maternity, etc.). Each record is linked to a hospital episode. Due to the different data sources, the records have different formats. The characteristics of the raw datasets are shown in Table 3, providing key insights into their structure and scale, which lay the groundwork for the experiments and highlight the complexity of the healthcare data, strengthening the credibility and applicability of the research.

When extracting valuable knowledge from large datasets, an essential stage is to prepare the datasets for the effective application of data mining techniques. This crucial phase, known as data preprocessing, involves a series of data processing techniques to optimise the dataset for knowledge extraction. Several preprocessing modules have been introduced in the dataset used, focusing on the transformation of variables to better encapsulate the underlying information and the enrichment of the variables in the dataset with external data sources that can be queried in [63].

After the data pre-processing phase, the study delves into specific modules designed to enhance and enrich the data sets. Notable transformations include normalisation of values, conversion of factor variables, and refinement of medical data representations. For example, postcodes are transformed into

Table 3 Data source features description

Features	Codification	Method for enrichment
Diagnoses	International classification of diseases (ICD)	External API
Origin of patients	Spanish health service system	Database
Reason for discharge	International classification of diseases (ICD)	External API
Reason for admission	International classification of diseases (ICD)	External API
Surgical procedures	International classification of procedures	External API
Diagnostic tests	International classification of Test	External API
Departments	Spanish health service system	Database
Other data[1]	Andalusian health service system (SAS)	Database

[1]Rest of the variables related to the management and information of the Andalusian hospital system

geographical data, dates are parsed into detailed temporal attributes, and patient-related data such as age and hospitalization details are integrated. In addition, a data enrichment module is implemented to enhance variables from external coding dictionaries, further enriching the datasets with valuable information. All data and information related to the dataset is available at [63].

Dealing such large and complex datasets requires the use of advanced big data management algorithms, as emphasised by Andronie et al. [64], to efficiently process and analyse the data. By using these algorithms, we ensure that our system can cope with the scale and complexity inherent in healthcare data, facilitating effective federated learning.

5.2 Data processing

One critical phase in extracting valuable knowledge from large datasets is data pre-processing, which involves preparing the data for the application of a learning/mining technique. This proposal includes several pre-processing modules where some variables are transformed to better model the information they contain, external data are used to enrich the variables in the dataset, and continuous data are discretised using linguistic variables. A novel process of applying linguistic labels automatically or with expert guidance to improve interpretability has been applied in this case [65].

Initially, classical pre-processing is performed, eliminating irrelevant codes, normalising values, and transforming factor variables [63]. Healthcare-oriented data processing follows, converting variables like history codes, postcodes, and dates into more meaningful data representations. For instance, postcodes are mapped to municipalities, cities, and countries, while dates are expanded to include patient age, admission times, and hospital stay details. Additionally, a data enrichment module integrates external coding dictionaries to enhance variables like diagnoses and surgical procedures with additional context from external taxonomies and databases, as shown in [63].

Effective data pre-processing is crucial when dealing with large and complex healthcare datasets, as it has a significant impact on the quality of the extracted knowledge. Similar to the approaches described by Mavrogiorgou et al. [66], who proposed an optimized KDD process for collecting and processing ingested and streaming healthcare data, our methodology includes advanced pre-processing techniques tailored for healthcare applications. Additionally, the use of data processing tools for graph data modeling in big data analytics, as discussed by Voulgaris et al. [67], highlights the importance of selecting appropriate data structures and processing methods to effectively handle complex relationships within the data. By incorporating these strategies, we increase the robustness and effectiveness of our data pre-processing pipeline, ensuring that the subsequent federated learning processes are based on high-quality, well-prepared data.

5.3 Association rules in federated environment

In this section, we delve into the importance of experimental parameters, exploring how variations in the number of participants, transactions, features, and data distribution can help us understand the scalability, efficiency and effectiveness of the simulated system. For that, we will use a federated approach for association rule mining.

5.3.1 Number of participants (medical-units)

This parameter determines the size and complexity of our simulated federated environment. By varying the number of hospital departments, we can evaluate the scalability of our federated association rule mining algorithms. To perform an experiment to demonstrate the scalability of our system, we will vary the number of participants. Specifically, in this real use case, the number of participants is iteratively set to values within the interval [3, 20], keeping all other parameters the same.

Figure 3 shows a linear and scalable growth with respect to the number of participants. As we incrementally increase the number of participants, the output metrics (computational time) exhibit a proportional and consistent increase. This linear trend indicates that our federated system can efficiently handle additional participants efficiently without an exponential increase in resource requirements. Therefore, the system maintains scalability, making it possible to increase the number of participants while ensuring that performance remains predictable and manageable. This behavior is essential for practical applications where the number of participants can vary significantly.

5.3.2 Number of transactions

The number of transactions in our dataset is crucial, as it influences the data that participants need to share and process. This parameter affects the communication overhead and computational requirements of the federated learning process, so it is important to investigate the performance of the algorithms in relation to the transaction load. For this experiment, the number of transactions was iteratively set to

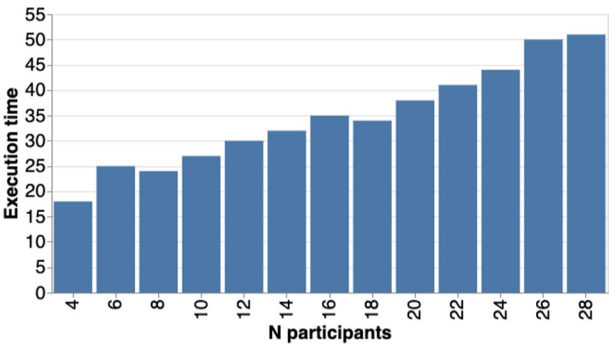

Fig. 3 Total execution time vs. the number of participants in the simulation

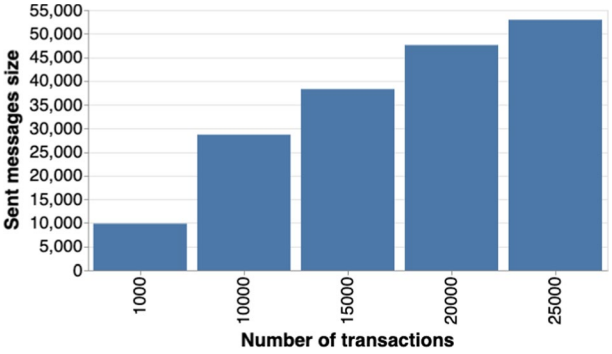

Fig. 4 Effect of the number of transactions on the size of the sent messages

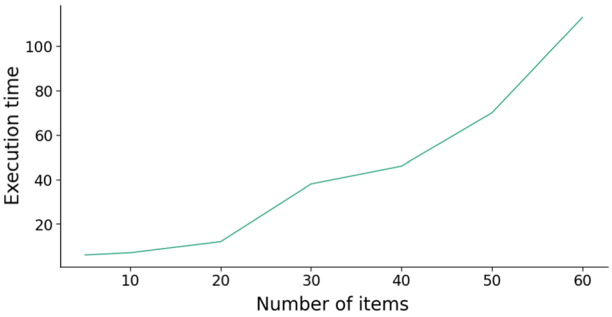

Fig. 5 Total execution time vs the number of items for the algorithm

values between 1000 and 20,000 by replicating some records in departments with lower workloads to facilitate this performance test. The following parameters were set: $participants = 12, minsupp = 0.2$, and $n_{items} = 32$.

5.3.3 Number of features

The number of features in our dataset is important because it affects the complexity of the mining task. Increasing the number of features can challenge algorithms in terms of computational efficiency and the discovery of meaningful association rules, as the computational complexity with respect to this parameter is exponential in exhaustive search algorithms. Exploring this parameter helps to assess the adaptability of algorithms to high-dimensional data. In this experiment, the number of selected features was iteratively set to values between 5 and 60, while keeping all other parameters constant ($participants = 12$ and $n_{transactions} = 5000$) (Figs. 4, 5).

The first metric to consider is the total execution time with respect to the number of items, a metric that behaves exponentially and is equal to that of the algorithm without the federated environment. After this experimentation, we can conclude that the higher dimensionality of the data affects the algorithm's performance.

This experiment has been very useful for analysing an algorithm's performance by varying one of the most important parameters in frequent itemset mining and association rules. Although most of the metrics behave similarly, the mining costs are drastically different and could be a real challenge in a production environment.

5.4 Discussion of the results

Performance tests have been conducted on the aforementioned dataset, including a comprehensive evaluation using the 57 most significant variables within a federated environment comprising different departments of the Clinical Hospital of Granada. This evaluation involved 35 departments (nodes) and led to the discovery of several interesting association rules. Detailed results and experimentation details are presented below.

The experiments were carried out with different threshold configurations. Here we present the results obtained with a minimum support of 0.1 and a minimum confidence of 0.8.

The derived set of rules has enabled the discovery of hidden patterns in the relationships between different diagnoses present in patients, linking them to patient characteristics. This type of relationship facilitates the study of co-morbidities within the dataset.

By examining the discovered patterns, several notable rules emerge. For example:

$$\begin{aligned} &\{Age = long, Timeofadmission = long\} \rightarrow \\ &\quad \{Reason\_medical\_discharge = death\} \end{aligned} \tag{1}$$

Federated environments allow for more general rules by extracting information from the whole set while preserving the privacy of the different nodes, taking advantage of a higher volume of transactions. This is why we can extract these types of more general rules, whereas if we had analysed by department, there would not have been enough data to establish general rules.

6 Conclusions

We have introduced and developed the SimulaFed system, a federated environment that offers a customized and configurable approach to data analysis, as well as flexibility, scalability and versatility in a simulation environment that supports both supervised and unsupervised learning. We have analysed the state-of-the-art approaches for simulating federated systems and evaluated their advantages and limitations. Our proposal aims to address these weaknesses by enabling researchers to create experiments tailored to their specific needs through the Docker platform. The architecture ensures communication privacy by means of several cryptographic techniques, Secure Multiparty Computation and differential privacy, generating final global results without sharing data from the nodes, providing robust privacy protection.

Our adaptive framework aims to achieve secure inter-node communication and adapts experiments for any desired federated learning/mining algorithm. Tested on up more than 100 nodes, our system shows versatility in governance (centralised or decentralised) and supports a wide range of machine learning algorithms.

We validated the system in a real-world medical scenario, avoiding direct access to confidential information while taking advantage of a larger volume of data to achieve more robust results using federated algorithms for mining association rules on patient health data across multiple departments. Performance tests demonstrated linear and scalable growth, efficient handling of a variable number of participants, and the ability to adapt to high-dimensional data. Furthermore, the results obtained show the potential to uncover hidden patterns and aid in the study of comorbidities.

Moving forward, we aim to expand the system to incorporate various Federated Learning algorithms from the literature, as well as to implement enhancements for handling the different requirements and challenges in enterprise (cross-silo) and mobile/edge (cross-device) federated learning scenarios. By continuing to evolve and adapt our framework, we work to meet the ever-changing demands of data analysis and security in federated environments.

Acknowledgements We would like to acknowledge support for this work from Grant PID2021-123960OB-I00 funded by MCIU/AEI/10.13039/501100011033 and by ERDF/EU (FederaMed project), and from DesinfoScan project: Grant TED2021-129402B-C21 funded by MCIU/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR. In addition the pre-competitive project of the Plan Propio of the "University of Granada".We would like to thank to Clinical Hospital San Cecilio for the data. Finally, the research reported in this paper is also funded by the European Union (BAG-INTEL project, grant agreement no. 101121309).

Funding The research reported in this paper was partially supported by Grant PID2021-123960OB-I00 funded by MCIU/AEI/10.13039/501100011033 and by ERDF/EU (FederaMed project), and from DesinfoScan project: Grant TED2021-129402B-C21 funded by MCIU/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR.

Data availability The used data is available on https://archive.ics.uci.edu/ml/index.php.

Declarations

Conflict of interest The authors declare that they have no Conflict of interest.

Ethics approval and consent to participate Not applicable.

References

1. Voigt P, Bussche A (2017) The eu general data protection regulation (gdpr). A Practical Guide, 1st Ed., Cham: Springer International Publishing 10(3152676):10–5555
2. Act A (1996) Health insurance portability and accountability act of 1996. Public Law 104:191
3. Brisimi TS, Chen R, Mela T, Olshevsky A, Paschalidis IC, Shi W (2018) Federated learning of predictive models from federated electronic health records. Int J Med Inform 112:59–67
4. Yang Q, Liu Y, Chen T, Tong Y (2019) Federated machine learning: concept and applications. ACM Transact Intell Syst Technol (TIST) 10(2):12–11219
5. Singh AK, Anand A, Lv Z, Ko H, Mohan A (2021) A survey on healthcare data: a security perspective. ACM Transact Multimidia Comput Commun Appl 17(2s):1–26

6. Jain S, Jerripothula KR (2023) Federated learning for commercial image sources. In: 2023 IEEE/CVF winter conference on applications of computer vision (WACV), pp. 6534–6543. IEEE
7. Rieke N, Hancox J, Li W, Milletarì F, Roth HR, Albarqouni S, Bakas S, Galtier MN, Landman BA, Maier-Hein K, Ourselin S, Sheller M, Summers RM, Trask A, Xu D, Baust M, Cardoso MJ (2020) The future of digital health with federated learning. npj Digit Med. https://doi.org/10.1038/s41746-020-00323-1
8. Chen Y, Liang L, Gao W (2023) Non trust detection of decentralized federated learning based on historical gradient. Eng Appl Artif Intell. https://doi.org/10.1016/j.engappai.2023.105888
9. Abad MSH, Ozfatura E, Gündüz D, Ercetin O (2020) Hierarchical federated learning across heterogeneous cellular networks, 2020, 8866–8870. https://doi.org/10.1109/ICASSP40776.2020.9054634
10. Qu Y, Pokhrel SR, Garg S, Gao L, Xiang Y (2021) A blockchained federated learning framework for cognitive computing in industry 4.0 networks. IEEE Transact Ind Inform 17(4):2964–2973. https://doi.org/10.1109/TII.2020.3007817
11. Singh S, Rathore S, Alfarraj O, Tolba A, Yoon B (2022) A framework for privacy-preservation of iot healthcare data using federated learning and blockchain technology. Future Gener Comput Syst 129:380–388. https://doi.org/10.1016/j.future.2021.11.028
12. Wang Y, Bennani IL, Liu X, Sun M, Zhou Y (2021) Electricity consumer characteristics identification: a federated learning approach. IEEE Transact Smart Grid 12(4):3637–3647. https://doi.org/10.1109/TSG.2021.3066577
13. Yang K, Jiang T, Shi Y, Ding Z (2020) Federated learning via over-the-air computation. IEEE Transact Wireless Commun 19(3):2022–2035. https://doi.org/10.1109/TWC.2019.2961673
14. Hartmann F, Rojas R (2018) Federated learning. PhD thesis, Free University of Berlin
15. Chaddad A, Wu Yihang (2024) Christian desrosiers: federated learning for healthcare applications. IEEE internet of things journal
16. Camajori Tedeschini B, Savazzi S, Stoklasa R, Barbieri L, Stathopoulos I, Nicoli M, Serio L (2022) Decentralized federated learning for healthcare networks: a case study on tumor segmentation. IEEE Access 10:8693–8708
17. Aouedi O, Sacco A, Piamrat K, Marchetto G (2022) Handling Privacy-Sensitive Medical Data With Federated Learning: challenges and Future Directions. IEEE Journal of Biomedical and Health Informatics
18. Kiourtis A, Mavrogiorgou A, Menesidou S-A, Gouvas P, Kyriazis D (2020) A secure protocol for managing and sharing personal healthcare data. Integr Citizen Cent Digital Health Soc Care 275:92–96. https://doi.org/10.3233/SHTI200701
19. Tongnian Wang, Yan Du, Yanmin Gong, Kim-Kwang Raymond Choo, Yuanxiong Guo (2022) Applications of federated learning in mobile health: scoping review. J Med Internet Res
20. Bebortta S, Tripathy S, Shakila B, Chowdhary CL (2023) Fedehr: a federated learning approach towards the prediction of heart diseases in IoT-based electronic health records. Diagnostics
21. Dhasaratha C, Hasan MK, Islam S, Khapre S, Abdullah S, Ghazal TM, Alzahrani AI, Alalwan N, Vo N, Akhtaruzzaman M (2024) Data privacy model using blockchain reinforcement federated learning approach for scalable internet of medical things. CAAI Transactions on Intelligence Technology
22. Houda ZAE, Hafid AS, Khoukhi L, Brik B (2023) When collaborative federated learning meets blockchain to preserve privacy in healthcare. IEEE Transact Netw Sci Eng 10(5):2455–2465
23. Rauniyar A, Hagos DH, Jha D, Håkegård JE, Bagci U, Rawat DB, Vlassov V (2024) Federated learning for medical applications: a taxonomy, current trends, challenges, and future research directions. IEEE Internet Things J 11(5):7374–7398
24. McMahan HB, Moore E, Ramage D, Hampson S, Arcas BA (2017) Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA
25. Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Bhagoji AN, Bonawitz K, Charles Z, Cormode G, Cummings R et al (2021) Advances and open problems in federated learning. Found Trends Mach Learn 14(1–2):1–210
26. Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, Ramage D, Segal A, Seth K (2019) Towards federated learning at scale: system design. In: Proceedings of the 2nd SysML Conference, Palo Alto, CA, USA
27. Caldas S, Konecny J, McMahan HB, Talwalkar A (2018) Leaf: A benchmark for federated settings. arXiv preprint arXiv:1812.01097

28. Li T, Sahu AK, Talwalkar A, Smith V (2020) Federated learning: challenges, methods, and future directions. IEEE Signal Process Mag 37(3):50–60
29. TensorFlow: TensorFlow Federated Documentation. Accessed: 2024-05-15 (2023). https://www.tensorflow.org/federated
30. FederatedAI: FATE (Federated AI Technology Enabler) Documentation. Accessed: 2024-05-15 (2023). https://github.com/FederatedAI/FATE/blob/master/README.md
31. Flower Labs: flower framework documentation. Accessed: 2024-05-15 (2024). https://flower.dev/docs/
32. OpenMined: PySyft and PyGrid documentation. Accessed: 2024-05-15 (2024). https://github.com/OpenMined/PySyft
33. OpenFL Developers: open Federated Learning (OpenFL) Documentation. Accessed: 2024-05-15 (2024). https://github.com/securefederatedai/openfl
34. IBM: IBM federated learning documentation. Accessed: 2024-05-15 (2024). https://ibmfl.res.ibm.com
35. NVIDIA: NVIDIA Clara Documentation. Accessed: 2024-05-15 (2024). https://docs.nvidia.com/clara/
36. Substra: Substra Documentation. Accessed: 2024-05-15 (2024). https://github.com/Substra/substra
37. Park J-I, Joe-Wong C (2024) Federated learning with flexible architectures. arXiv preprint arXiv:2406.09877
38. Sani L, Gusmão PPB, Iacob A, Zhao W, Qiu X, Gao Y, Fernandez-Marques J, Lane ND (2023) High-throughput simulation of federated learning via resource-aware client placement. arXiv preprint arXiv:2306.17453
39. Li L, Wang J, Xu C (2020) Flsim: an extensible and reusable simulation framework for federated learning. In: International conference on simulation tools and techniques, pp. 350–369. Springer
40. Youngblood SM, Pace DK, Eirich PL, Gregg DM, Coolahan JE (2000) Simulation verification, validation, and accreditation. J Hopkins APL Tech Dig 21(3):359–367
41. Byrd D, Polychroniadou A (2020) Differentially private secure multi-party computation for federated learning in financial applications. In: Proceedings of the first ACM international conference on AI in finance, pp. 1–9
42. Hasan J (2023) Security and privacy issues of federated learning. arXiv preprint arXiv:2307.12181
43. Park J, Lim H (2022) Privacy-preserving federated learning using homomorphic encryption. Appl Sci 12(2):734
44. Developer N (2024) Federated learning from simulation to production with NVIDIA FLARE. Accessed: 2024-07-31. https://developer.nvidia.com/blog/federated-learning-from-simulation-to-production-with-nvidia-flare/
45. Peregrina JA, Ortiz G, Zirpins C (2022) Towards a metadata management system for provenance, reproducibility and accountability in federated machine learning. In: European conference on service-oriented and cloud computing, pp. 5–18. Springer
46. Ryffel T, Trask A, Dahl M, Wagner B, Mancuso J, Rueckert D, Passerat-Palmbach J (2018) A generic framework for privacy preserving deep learning. arXiv preprint arXiv:1811.04017
47. He C, Li S, So KHR, Zhang X, Wang Q, Fang Z, Yoon J, Ding ZS, Li H, Koyejo S, et al (2020) Fedml: A research library and benchmark for federated machine learning. arXiv preprint arXiv:2007.13518
48. Chahar H, Keshavamurthy B, Modi C (2017) Privacy-preserving distributed mining of association rules using elliptic-curve cryptosystem and Shamir's secret sharing scheme. Sādhanā 42(12):1997–2007
49. Tassa T (2013) Secure mining of association rules in horizontally distributed databases. IEEE Trans Knowl Data Eng 26(4):970–983
50. Yao AC (1982) Protocols for secure computations. In: 23rd Annual symposium on foundations of computer science (sfcs 1982), pp. 160–164. IEEE
51. Kantarcioglu M, Clifton C (2004) Privacy-preserving distributed mining of association rules on horizontally partitioned data. IEEE Trans Knowl Data Eng 16(9):1026–1037
52. Dwork C, Roth A (2014) The algorithmic foundations of differential privacy. Found Trends Theor Comput Sci 9(3–4):211–407
53. Ji S, Lipton ZC, Elkan C, Naughton JF (2014) Differential privacy in machine learning: a survey and a user guide. arXiv preprint arXiv:1412.7584
54. Gentry C (2009) Fully homomorphic encryption using ideal lattices. In: STOC '09: Proceedings of the Forty-first annual ACM symposium on theory of computing, pp. 169–178. ACM

55. Pohlig SC, Hellman ME (1978) An improved algorithm for computing logarithms over gf(p) and its cryptographic significance. IEEE Trans Inf Theory 24(1):106–110
56. Aggarwal CC, Yu PS (2008) In: Aggarwal CC, Yu PS (eds.) A general survey of privacy-preserving data mining models and algorithms, pp. 11–52. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-70992-5_2
57. Clifton C, Kantarcioglu M, Vaidya J, Lin X, Zhu MY (2002) Tools for privacy preserving distributed data mining. ACM SIGKDD Explor Newsl 4(2):28–34
58. El Ouadrhiri A, Abdelhadi A (2022) Differential privacy for deep and federated learning: a survey. IEEE Access 10:22359–22380
59. Zhao J, Chen Y, Zhang W (2019) Differential privacy preservation in deep learning: challenges, opportunities and solutions. IEEE Access 7:48901–48911
60. Modi CN, Patil AR (2016) Privacy preserving association rule mining in horizontally partitioned databases without involving trusted third party (ttp). In: Proceedings of 3rd international conference on advanced computing, networking and informatics: ICACNI 2015, Volume 2, pp. 549–555. Springer
61. Shamir A (1979) How to share a secret. Commun ACM 22(11):612–613
62. Juan X, Yanqin Z (2010) Application of distributed oblivious transfer protocol in association rule mining. In: 2010 Second international conference on computer engineering and applications, vol. 2, pp. 204–207. IEEE
63. Fernandez-Basso C, Gutiérrez-Batista K, Morcillo-Jiménez R, Vila M-A, Martin-Bautista MJ (2022) A fuzzy-based medical system for pattern mining in a distributed environment: application to diagnostic and co-morbidity. Appl Soft Comput 122:108870
64. Andronie M, Lăzăroiu G, Iatagan M, Hurloiu I, Ştefănescu R, Dijmărescu A, Dijmărescu I (2023) Big data management algorithms, deep learning-based object detection technologies, and geospatial simulation and sensor fusion tools in the internet of robotic things. ISPRS Int J Geo Inf 12(2):35
65. Fernandez-Basso C, Ruiz MD, Martin-Bautista MJ (2020) A fuzzy mining approach for energy efficiency in a big data framework. IEEE Trans Fuzzy Syst 28(11):2747–2758
66. Mavrogiorgou A, Kiourtis A, Manias G, Kyriazis D (2021) An optimized kdd process for collecting and processing ingested and streaming healthcare data. In: 2021 12th international conference on information and communication systems (ICICS), pp. 49–56. IEEE
67. Voulgaris K, Kiourtis A, Karamolegkos P, Karabetian A, Poulakis Y, Mavrogiorgou A, Kyriazis D (2022) Data processing tools for graph data modelling big data analytics. In: 2022 13th international congress on advanced applied informatics winter (IIAI-AAI-Winter), pp. 208–212. IEEE

Authors and Affiliations

Jose M. Rivas[1,2] · Carlos Fernandez-Basso[1,2] · Roberto Morcillo-Jimenez[1,2] · Juan Paños-Basterra[1,2] · M. Dolores Ruiz[1,2] · Maria J. Martin-Bautista[1,2]

✉ Carlos Fernandez-Basso
cjferba@decsai.ugr.es

Jose M. Rivas
jose.rivas@ieee.org

Roberto Morcillo-Jimenez
robermorji@ugr.es

Juan Paños-Basterra
panosjuan@ugr.es

M. Dolores Ruiz
mdruiz@decsai.ugr.es

Maria J. Martin-Bautista
mbautis@decsai.ugr.es

[1] Department of Computer Science and A.I, University of Granada, Periodista Daniel Saucedo Aranda, s/n, Granada 18014, Spain

[2] Research Centre for Information and Communications Technologies (CITIC-UGR), University of Granada, Granada 18014, Spain