Indonesian Journal of Electrical Engineering and Computer Science
Vol. 40, No. 2, November 2025, pp. 1011~1022
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v40.i2.pp1011-1022

Development and integration of a privacy computing gateway for enhanced interoperability

Akhila Reddy Yadulla, Vinay Kumar Kasula, Bhargavi Konda, Mounica Yenugula, Supraja Ayyamgari
Department of Information Technology, University of the Cumberlands, Kentucky, United States

Article Info

Article history:

Received Nov 11, 2024
Revised Aug 2, 2025
Accepted Oct 15, 2025

Keywords:

Application layer
Communication layer
Interoperability
Privacy computing
Protocol layer
Three-tiered system architecture

ABSTRACT

A new design of privacy computing gateway stands as the solution to secure efficient interoperability between heterogeneous platforms. The growing importance of data privacy, along with rising collaborative data analysis operations, creates an immediate need for standardized privacy-preserving frameworks that are adaptable to diverse situations. A three-layered architecture consisting of application protocol and communication layers receives support from an Adaptation mechanism designed for compatibility between separate privacy computing systems. Testing of the framework uses standard machine learning methods together with horizontal and vertical federated learning using diverse data quantities and feature distribution patterns. The gateway achieves satisfactory model performance and protects data privacy integrity in combination with platform interoperability. area under the curve (AUC) along with F1 score metrics, proves that the proposed system reaches performance equivalence with centralized models when operating within privacy-limited environments. The research introduces an effective solution for securing cross-platform data sharing that will enable secure inter-sector collaboration in finance, healthcare, and government applications.

Corresponding Author:

Vinay Kumar Kasula
Department of Information Technology, University of the Cumberlands
6178 College Station Dr, Williamsburg, Kentucky, United States
Email: vkasula19501@ucumberlands.edu

1. INTRODUCTION

The rapid advancement of privacy-preserving computation technologies is largely driven by an increasing need for secure, privacy-compliant data-sharing solutions across sectors such as finance, healthcare, and government. These technologies empower “data usability without visibility,” allowing users to derive valuable insights without directly accessing sensitive information. Key methods include federated learning, secure multiparty computation (SMPC), homomorphic encryption, and differential privacy [1]-[3]. For example, in healthcare, federated learning enables multiple hospitals to collaboratively train machine learning models on shared patient data while strictly adhering to privacy regulations. This collaborative approach drives advancements in diagnostics and treatment recommendations, facilitating improvements in patient care without compromising data privacy. However, while each privacy-preserving technology offers unique strengths, they are often developed on isolated architectures with distinct algorithmic frameworks, resulting in limited interoperability. This lack of seamless communication among systems leads to “data archipelagos”—broad clusters of isolated data that exacerbate existing “data silo” issues. In critical fields that depend on cross-institutional data collaboration, such as finance and healthcare, this fragmentation forces organizations to adopt multiple platforms to achieve cross-provider data sharing. Consequently, operational

complexity rises, and infrastructure costs grow as institutions are required to invest in various platforms and compatibility solutions [4]. To address these limitations, researchers are actively exploring standardization frameworks and interoperability protocols aimed at bridging these disparate systems. Recent efforts have focused on developing modular, open-source privacy-preserving solutions that promote cross-platform functionality, enabling smoother and more efficient data sharing [5], [6]. Despite these advancements, achieving scalable and robust interoperability solutions that preserve privacy and security across diverse ecosystems remains a challenging task. Significant research and development efforts are essential to address these interoperability barriers, which currently stand as a major obstacle to the widespread adoption of privacy-preserving computation. By overcoming these challenges, industries could unlock the full potential of privacy-preserving technologies, fostering collaborative data initiatives that maintain strong privacy safeguards.

2. DEVELOPMENT OF INTEROPERABILITY

Privacy computing refers to a class of information technologies that enable data analysis and computation while ensuring data is not leaked. It spans multiple fields, including data science, cryptography, and artificial intelligence [7]-[9]. As privacy computing technology continues to evolve, the issue of interoperability between privacy computing platforms has become increasingly prominent. Interoperability in privacy computing means enabling the interaction and collaboration of data, algorithms, and computing power across different systems through standardized interfaces and interaction protocols, allowing users to jointly complete the same privacy computing tasks.

The progression of interoperability in privacy computing platforms has evolved through three distinct stages, each reflecting significant advancements in cross-platform compatibility and the development of industry-wide standards. The first stage focuses on the foundational technologies that enable different platforms to interact, addressing basic compatibility issues and ensuring that privacy computing systems can communicate with one another. The second stage marks the introduction of more sophisticated protocols and interfaces that facilitate seamless data exchange and privacy-preserving computations across platforms, allowing for greater flexibility and efficiency. In the third and final stage, industry-wide standardization efforts take place, leading to the establishment of universal protocols and frameworks that enable wide-scale adoption and collaboration across various platforms, ensuring interoperability without compromising privacy. These stages reflect the ongoing evolution of privacy computing and its role in enabling secure, privacy-preserving collaboration in the digital world [10].

Stage 1: Basic Interoperability Among Platforms from Different Vendors

In the initial stages of privacy computing, data providers typically implemented privacy-preserving systems tailored to client-specific requirements, often based on existing or emerging platforms. To achieve basic interoperability, vendors engaged in one-to-one technical integrations, creating custom configurations to ensure compatibility. This phase required unified management of nodes and resources, alongside the design of specialized algorithms and workflows to coordinate platform interactions. Although these integrations were individually customized and allowed one vendor to take the lead, they were effective at meeting immediate business needs by establishing compatibility through mutual agreements on shared algorithms.

Stage 2: Advanced Interoperability Among Platforms from Different Vendors

As privacy computing platforms expanded in scope and scale, vendors encountered heightened interoperability challenges. The one-on-one integration model, though initially sufficient, began to struggle under the complexity of multi-party interactions, resource management, and a lack of standardization in communication processes. Consequently, vendors sought more sophisticated interoperability approaches, focusing on establishing advanced interoperability standards. This included creating communication protocols, message formats, and standardized encryption methods to enable seamless, higher-level interactions among diverse platforms. The goal of this stage was to support more scalable, systematic cross-platform functionality as client demands for privacy-preserving data sharing grew.

Stage 3: Industry-Wide Interoperability Standards

With a proliferation of independent privacy computing architectures, “data silos” emerged as isolated, non-communicating systems, inhibiting the seamless flow of data across platforms. As privacy computing gained traction across various industries, the limitations imposed by this fragmented landscape became apparent. Establishing unified industry-wide standards for interoperability thus became a critical goal, enabling broader integration across privacy platforms. By defining standardized communication protocols, message formats, and encryption mechanisms, the industry aimed to foster collaboration among different privacy computing platforms. This stage of interoperability would enable privacy-preserving data sharing on a much larger scale, providing the foundation for cohesive, industry-wide privacy computing ecosystems.

3. CHALLENGES OF INTEROPERABILITY

Privacy computing involves intricate principles and diverse platform architectures, making interoperability a challenging goal. To achieve seamless compatibility among different privacy computing platforms, it is essential to bridge architectural differences while preserving each platform's unique functionality and ensuring compatibility across systems [11]-[14]. This level of integration presents substantial challenges due to the underlying complexity of privacy-preserving technologies. A key obstacle stems from the diversity of fundamental principles that shape each platform. Each provider of privacy computing technologies has its own proprietary methods and algorithms, which are central to the computation and exchange of data. These differences in algorithmic design create unique data processing and interaction models, complicating communication between platforms with disparate computational logic. This variation in algorithmic structures means that platforms cannot easily "speak the same language," making interoperability a complex task to address. Further complicating matters, privacy computing platforms vary significantly in their functional components, such as communication modules, encryption protocols, resource and task management systems, model management frameworks, node management, and authorization protocols. These differences reflect each provider's unique technological approach and the specific application environments they prioritize [15], [16].

Such diversity in platform architectures creates a multifaceted landscape where integration requires overcoming the distinct implementations within each system, which becomes the first major hurdle for interoperability. Another layer of complexity arises from vendor-related differences. With numerous providers offering privacy computing solutions, each with its own set of standards, achieving interoperability across multiple platforms is increasingly challenging. The sheer volume of variations across platforms amplifies the difficulty of building a cohesive, interoperable environment that can function seamlessly while maintaining each platform's proprietary standards.

4. PATHWAYS TO ACHIEVING INTEROPERABILITY

Privacy computing serves as a vital technology to balance the flow of data with the need for privacy protection. As the large-scale application of privacy computing grows, achieving interoperability between platforms becomes essential for cross-platform functionality and efficiency. This section discusses interoperability pathways from a platform architecture standpoint, classifying privacy computing platforms into three key layers: the application layer, algorithm layer, and primitive layer. Each layer fulfills distinct functions within privacy computing, and effective interoperability solutions must address each level separately. Consequently, interoperability in privacy computing is categorized into three types: application layer interoperability, algorithm layer interoperability, and primitive layer interoperability.

Application Layer Interoperability enables seamless communication between platforms at the application level. This level facilitates system management functions such as node discovery and resource allocation, allowing for business-level integration across platforms. By standardizing interactions and management processes, application-layer interoperability allows privacy-preserving computations to function smoothly across different system interfaces. Algorithm Layer Interoperability focuses on creating standardized algorithmic frameworks applicable across various platforms. Here, the algorithms' design principles are transparent and shared among providers, enabling different vendors to implement the same algorithms with consistent interaction processes. This layer allows for flexible interoperability by making algorithms interoperable despite differences in the underlying technology stacks. Primitive Layer Interoperability addresses the most granular level, where the smallest components, or computational primitives, form the basis of privacy-preserving protocols. For example, in secure multi-party computation (MPC) with the ABY3 protocol, platforms need to adhere to fundamental principles like data encryption and partitioning to ensure secure, distributed computation. By defining these primitives, different platforms can independently implement protocol steps, ensuring compatibility at the most foundational level. Through abstraction and standardization at the primitive layer, platforms achieve compatibility in core functions, allowing for mid-layer algorithms and application-layer services to interoperate seamlessly. To achieve full interoperability across these three layers, protocol processes and code implementations must be standardized.

Given that platforms often vary in openness and come from different providers, three main strategies facilitate this goal: protocol-level interoperability, SDK-level interoperability, and client-level interoperability. This paper introduces an innovative adaptation mechanism to foster interoperability across the application and algorithm layers for heterogeneous privacy computing platforms. This mechanism ensures that privacy computing platforms, regardless of underlying architectural differences, can work together effectively, advancing both privacy protection and efficient data sharing in a broad range of industries.

5. SYSTEM DESIGN

5.1. Design objectives

The core objective of this system design is to address the interoperability challenges between heterogeneous privacy computing platforms at both the application and algorithm layers, thereby facilitating collaborative gains across federated learning platforms. The adaptation mechanism within the TrustGate gateway aims to achieve this by utilizing an Adaptation module that leverages an adapter mechanism for application-layer adaptation and a mapping engine and algorithm management for interoperability at the algorithm layer. Ultimately, this approach integrates heterogeneous privacy computing platforms such as FATE and SecretFlow within the TrustGate gateway, achieving seamless interoperability at both the application and algorithm layers.

5.2. Architectural design

The structure of the interoperability Adaptation module, illustrated in Figure 1, highlights its role within the TrustGate gateway in establishing seamless interactions between diverse privacy computing platforms [17]. This section outlines the core principles and functionalities of the Adaptation mechanism. The Adaptation module consists of several essential components, including adapter management, a mapping engine, algorithm management, a computation engine, data interfaces, data measurement, and system management. Through adapter management, the module enables configurations for clients across multiple privacy computing platforms. By leveraging various adapters, the Adaptation module achieves application-layer interoperability, allowing diverse privacy computing clients to communicate and manage resources effectively. This modular approach also ensures algorithm-layer interoperability, with the mapping engine synchronizing algorithm parameters across platforms, enabling cross-platform functionality. The design underscores key aspects such as security, control, and measurability while maintaining flexibility for expansion and integration. Application-Layer Interoperability: At this layer, the Adaptation module includes multiple adapters designed to interface with other privacy computing platforms. This setup enables cross-platform interoperability and centralized resource management. Adapter management functions facilitate node and resource management, promoting efficient utilization across heterogeneous platforms [18], [19].

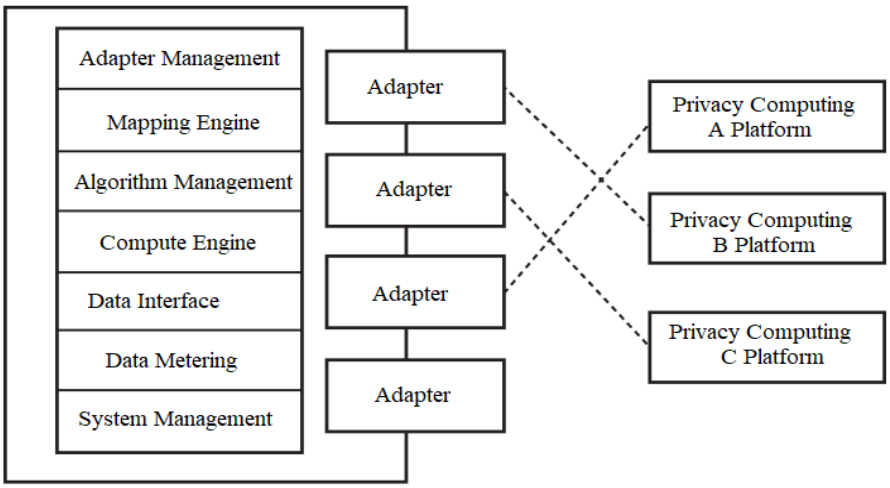

Figure 1. Interoperability adaptation module

Node management: This functionality handles the creation, modification, and deletion of nodes across platforms, maintaining critical node data (e.g., name, description, port, platform) and offering detailed display and search functionalities. This approach ensures organized and accessible node information across systems. Resource Management: This component focuses on resource allocation, supporting the sharing and coordination of various assets among platforms at the application level. Algorithm-Layer Interoperability: At the algorithm layer, the adaptation module employs a mapping engine and algorithm management to synchronize algorithm parameters and facilitate communication between platforms with different architectures. Key processes include adapting platforms, enabling different privacy computing systems to exchange data, and achieving interoperability. Routing and synchronizing algorithm data: coordinating data and parameter exchanges across systems, ensuring consistent algorithm behavior. Task Synchronization:

Ensuring task progress and status updates are shared across platforms. Together, these capabilities allow the Adaptation module to achieve interoperability at both the application and algorithm layers. By integrating heterogeneous platforms like FATE and SecretFlow, the module leverages adapter management, a mapping engine, and algorithm management to ensure compatibility and efficient resource sharing across diverse privacy computing ecosystems. This integration framework provides a foundation for flexible, scalable interoperability across the privacy computing landscape, enhancing system compatibility and data-sharing capabilities in a secure, measurable manner.

6. SYSTEM IMPLEMENTATION

6.1. Implementation architecture

This paper proposes a multi-layer, loosely coupled system architecture, as illustrated in Figure 2. This architecture integrates technologies from federated learning, blockchain, and big data platforms [20]-[22]. The platform is built on a big data infrastructure, leveraging the computational power, storage, and network resources of the big data platform. Data is stored on HDFS and processed using Spark resources. The platform consists of four main components: TrustGate, SecretFlow, FATEClient, and WebManager.

- TrustGate: TrustGate serves as the entry point for external systems interacting with the privacy computing platform, managing secure connections and trust verification with external networks. A critical part of TrustGate is its Adaptation Module, which ensures seamless interoperability between various heterogeneous platforms involved in privacy computing. This module is responsible for adapting protocols, data formats, and communication standards to enable secure and trustworthy interactions. TrustGate also generates trustworthy evidence, maintaining an audit trail that supports accountability and verification of all operations.
- SecretFlow: SecretFlow is the core component responsible for privacy-preserving operations within the system. It handles Secure MPC, which enables multiple parties to jointly compute a function over their inputs without revealing the inputs to each other. This module incorporates a variety of encryption algorithms, secure intersection techniques, and feature engineering tools that are essential for enabling secure federated learning. SecretFlow also supports a range of cryptographic protocols that enhance data confidentiality, protecting sensitive information from unauthorized access.
- FATEClient: FATEClient is responsible for managing the integration and transformation of FATE protocols (Federated AI Technology Enabler), a popular framework for federated learning. This component facilitates connectivity with FATE-based nodes across the network, allowing nodes to participate in federated learning while preserving data privacy. FATEClient ensures that data is prepared, processed, and exchanged according to FATE protocols, supporting collaborative computation without compromising data ownership or privacy.
- WebManager: WebManager oversees system management and user interaction, providing a web-based interface for monitoring and control. This component manages the storage of management data, such as logs, metadata, and process states, and supports process management to streamline operations and workflows within the platform. It includes visualization tools that present key metrics, workflow statuses, and computation results to users in an accessible and intuitive manner.

Overall, this architecture provides a robust and scalable platform for privacy-preserving federated learning applications. By building on big data infrastructure, such as HDFS for distributed storage and Spark for high-performance processing, the system can handle large datasets and complex computations efficiently. This multi-layer design, combined with the distinct roles of each component, ensures flexibility, security, and interoperability, making it well-suited for modern privacy computing environments.

6.2. Cross-platform architecture analysis

To facilitate interoperability within privacy computing, a three-layer system architecture can be adopted and structured to address the complexities of cross-platform integration. The first layer, the communication layer, focuses on establishing secure and efficient data transfer mechanisms between different systems, ensuring compatibility across platforms. The second layer, the protocol layer or interaction layer, is responsible for defining the rules and standards for how systems communicate, ensuring that data can be exchanged accurately and securely between platforms. The third layer, the application layer, operates at a higher level, integrating the various functionalities needed for specific applications, such as privacy-preserving data analytics or secure computations. This layered approach enables scalable and efficient interoperability, supporting the development of advanced privacy computing systems that can function seamlessly across different environments and platforms.

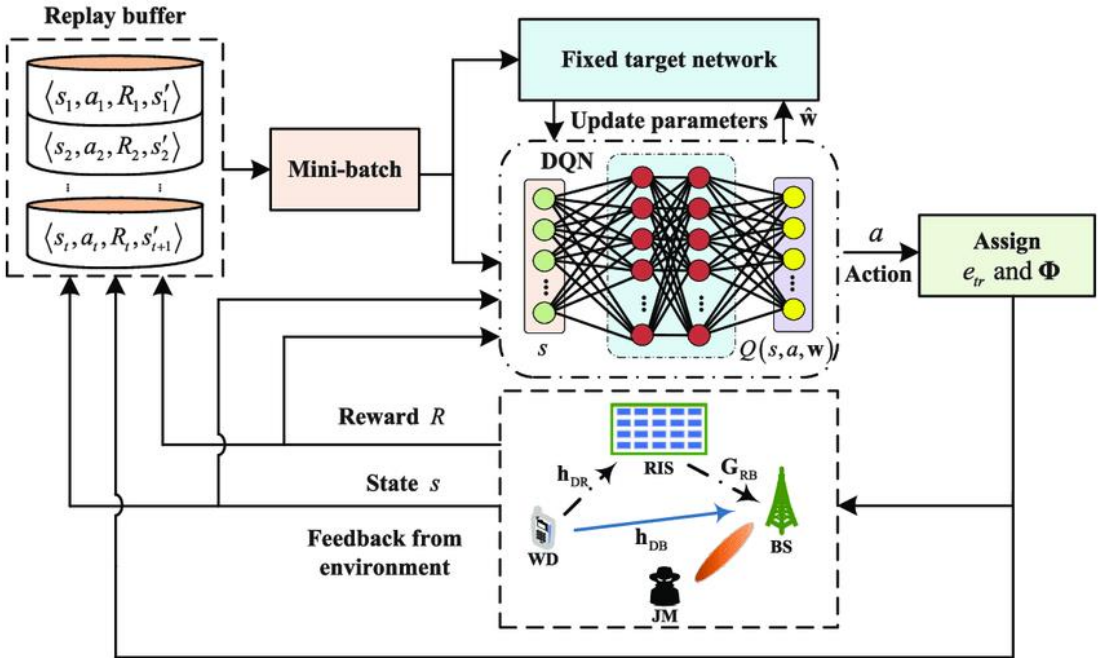

Figure 2. Network structure of the adaptive pilot design algorithm based on dueling DQN

6.2.1. Application layer

The application layer defines the essential communication requirements and the interoperability protocol stack. This involves establishing a collaborative framework for cross-platform privacy computing, which includes management processes and protocols. At this layer, task orchestration, scheduling, execution, monitoring, and evidence storage are unified through standardized rules, ensuring that all platforms involved can coordinate seamlessly. Clear definitions for each type of computing task's implementation processes ensure that interactions meet predefined interoperability standards, regardless of platform differences.

6.2.2. Protocol layer (interaction layer)

The protocol or interaction layer establishes standard procedures and requirements for each phase of cross-platform interactions. Organized across nodes, resources, and algorithm execution, this layer provides normative processes for critical interaction components such as discovery, authentication, application, and authorization. Detailed requirements for connection invocation further strengthen protocol layer functionality, ensuring that cross-platform engagements meet high standards for security and compatibility. This structure enables different systems to communicate in a consistent, secure manner without compromising each platform's unique characteristics.

6.2.3. Communication layer

The communication layer is the foundation of interoperability, providing standardized guidelines for all cross-platform data exchanges. This layer includes selecting communication frameworks, defining interfaces, standardizing data formats, and creating transmission protocols to handle inter-platform communication. These elements enable consistent data transmission and management, facilitating real-time integration. Importantly, the Adaptation module within the TrustGate gateway oversees these standards, serving as the coordinator that facilitates inter-platform communication. By centralizing adaptation processes, the communication layer enhances efficiency and reliability in cross-platform interoperability for privacy computing. Overall, this three-layered architecture—coordinated by the Adaptation module within TrustGate—provides a robust framework for achieving cross-platform interoperability in privacy computing systems, ensuring scalable, secure integration across platforms with diverse infrastructures.

6.3. Function verification

This study integrates the TrustGate gateway with the FATE and SecretFlow privacy computing platforms to validate interoperability. The validation process focuses on a "single migration and integration" business scenario, leveraging joint model training and federated learning to assess model performance. A

binary classification model is built using federated learning algorithms to predict potential single migration and integration users.

a) Feature selection: Feature selection involves identifying crucial user attributes that enhance predictive model accuracy by focusing on key behavioral and value metrics. This process examines basic user information, such as interaction patterns on the platform, to gain insights into engagement. Additionally, metrics like purchasing habits and engagement levels help in refining prediction outcomes. The type of device users interact with, whether mobile or desktop, is also considered to identify relevant features. Social connections are further analyzed to understand user relationships and influence, which enhances the model's ability to predict behaviors accurately.
b) Model training: Model training in this system is designed to be secure and iterative, aiming to enhance model accuracy while preserving data privacy. Initially, data samples are divided to represent two collaborating parties, with RSA encryption applied to establish secure intersections between the datasets, ensuring confidentiality. For algorithm selection, the Vertical Logistic Regression (LR) algorithm is compared against SecureBoost, with SecureBoost selected due to its superior performance in this application. To optimize the model, key parameters such as tree depth and the number of child nodes are carefully adjusted, which enhances the model's effectiveness and accuracy.
c) Performance comparison: Performance comparison in this system is assessed by evaluating predictive accuracy and model results under varying sample sizes and feature counts. For sample size testing, data samples are split horizontally, allowing performance to be compared across different data volumes to see how sample size impacts accuracy. Similarly, feature count testing is performed by vertically splitting data samples, introducing additional features incrementally to observe their effect on model performance. This approach provides insight into how both the amount of data and the complexity of features contribute to the model's predictive power. Ultimately, the performance metrics from these comparisons help determine the optimal balance of data volume and feature complexity for achieving high accuracy.

The TrustGate gateway, when integrated with federated learning platforms like FATE and SecretFlow, enables seamless interoperability across complex data-sharing environments. By leveraging TrustGate's secure communication framework, sensitive data can be processed without direct exposure, ensuring privacy-preserving operations. The integration of FATE and SecretFlow allows for decentralized machine learning, where models are trained on distributed data while maintaining data confidentiality. This combined functionality demonstrates superior performance in tasks such as predicting user migration patterns, as it can effectively handle large datasets while preserving user privacy. The approach not only streamlines the data-sharing process but also ensures that privacy is safeguarded throughout, making it highly suitable for applications that require secure and scalable data analysis.

6.3.1. Horizontal federated scenarios

The experimental results for horizontal federated learning with the TrustGate interoperability gateway are shown in Table 1. The F1 score shows no significant trend with changes in data volume, while the area under the curve (AUC) increases with larger data volumes [23]-[26]. As illustrated, the fitted curve reflects the actual situation, with the AUC increasing as the number of data points grows from 500,000, although the rate of increase diminishes.

Table 1. Experimental results of the privacy computing platform with integrated TrustGate interoperability gateway

Sample size (number of records per side)	Number of features on host side/important features	Number of features on host side/important features	AUC	Precision	Recall	F1 Score	Threshold (best metric)
1,000,000 (500,000)	35/30	38/32	0.810324	0.5604	0.240	0.352	0.75
2,000,000 (1,000,000)	36/28	39/34	0.812556	0.5902	0.250	0.360	0.73
4,000,000 (2,000,000)	37/29	40/35	0.814789	0.5801	0.255	0.365	0.72
5,000,000 (2,500,000)	38/31	41/36	0.817432	0.6005	0.245	0.370	0.70

6.3.2. Vertical federated scenarios

The study assesses the performance of vertical federated learning on a TrustGate-integrated platform, exploring how different conditions, including variations in data volume, feature dimensions, and the distribution of crucial features, affect the learning process. It analyzes how these factors influence the

model's ability to accurately classify and predict outcomes in federated settings, where data is distributed across different parties. To evaluate the effectiveness of the learning process, the study uses key performance metrics such as the AUC and the F1 score, which measure the model's precision, recall, and overall ability to make correct predictions. The impact of changing feature distributions, including the concentration of important features, is closely examined to understand how these factors affect the model's accuracy and generalization. Ultimately, the study provides insights into optimizing vertical federated learning for improved performance in real-world applications.

a) Vertical federated learning with varying data volumes

Table 2 presents the results of vertical federated learning with progressively increasing data volumes. On the TrustGate interoperability gateway, as the number of training samples grows, both AUC and F1 scores show improvement. This trend aligns with conventional machine learning, where model performance typically benefits from a larger dataset. The conclusion indicates that the TrustGate-integrated platform leverages increased data volume effectively, enhancing model accuracy and robustness.

Table 2. Results of vertical federated learning with varying data volumes

Number of samples	AUC	Precision	Recall	F1 Score
500,000training/ 5 million validation	0.7950	0.590	0.210	0.3100
2 million training/ 5 million validation	0.8020	0.560	0.230	0.3300
5 million training/ 5 million validation	0.8100	0.610	0.250	0.3500

b) Vertical federated learning with increasing feature dimensions

Results for vertical federated learning with rising feature dimensions are detailed in Table 3. With an increase in the number of feature dimensions, the model's AUC and F1 scores also show notable improvement. This outcome suggests that, similar to traditional machine learning, vertical federated learning on the TrustGate-integrated platform becomes more effective as additional feature dimensions are introduced, providing the model with richer data inputs that improve predictive power.

Table 3. Results of vertical federated learning with increasing feature dimensions

Number of samples	Number of features (Host)	Number of features (Guest)	AUC	Precision	Recall	F1 Score
5 million training/ 5 million validation	8	8	0.82	0.42	0.28	0.34
5 million training/ 5 million validation	12	12	0.84	0.47	0.32	0.38
5 million training/ 5 million validation	18	18	0.86	0.49	0.35	0.41
5 million training/ 5 million validation	25	25	0.88	0.53	0.37	0.45

c) Vertical federated learning with varying important feature distributions

Table 4 provides results for vertical federated learning across different distributions of key features. In this scenario, the model's AUC and F1 scores remain stable despite variations in the distribution of important features. This finding suggests that on the TrustGate-integrated platform, vertical federated learning performance is robust to changes in the distribution of critical features. Therefore, the platform can handle feature distribution shifts without significant impacts on model accuracy or stability.

Table 4. Results of vertical federated learning experiments with different distributions of important features

Number of samples	Distribution of importance features	AUC	Precision	Recall	F1 Score
5 million training/ 5 million validation	None of the important features are with the label party	0.81	0.52	0.23	0.32
5 million training/ 5 million validation	3 important features are with the label party	0.82	0.50	0.25	0.34
5 million training/ 5 million validation	6 important features are with the label party	0.83	0.55	0.27	0.36
5 million training/ 5 million validation	All important features are with the label party	0.84	0.58	0.29	0.38

6.4. Performance validation

The performance of interoperability between the FATE and SecretFlow privacy computing platforms, integrated with the TrustGate gateway, was tested. Following standardized performance evaluation procedures for privacy computing products, tests were conducted under specific hardware resources, data sets, algorithm requirements, and result conditions to simulate actual demand scenarios and assess the accuracy metrics of the privacy computing platforms. The privacy computing platforms strictly adhere to privacy protection principles, ensuring the confidentiality of user input data, maintaining the secrecy of intermediate data, and preventing the exposure of global intermediate data. In federated learning with joint modeling, measures are taken to protect sensitive information, such as local gradients, and to prevent leakage. The integrated TrustGate gateway with the FATE privacy computing platform continues to support differential privacy technology. Introducing noise makes individual data contributions difficult to determine, thus reducing the risk of data leakage during model training. FATE employs differential privacy to ensure model training privacy and also supports homomorphic encryption, allowing computations to be performed on encrypted data and thereby protecting data privacy. FATE uses homomorphic encryption to execute computations while maintaining data privacy.

To meet the accuracy requirements for evaluation, real-world joint modeling scenarios related to single migration and integration were selected for testing. In this setup, the TrustGate gateway's Adaptation module enabled seamless interoperability between the privacy computing platform and conventional machine learning algorithms. Modeling training was conducted using consistent datasets, feature selections, and training parameters. Model performance metrics, such as AUC (Area Under the Curve) and KS (Kolmogorov-Smirnov) values, were used to compare results from privacy-computing-based federated models with baseline models trained on plaintext data. The experiment was deemed successful if these metrics remained within a predefined error margin. The experiment involved a comparative analysis of traditional centralized modeling against federated learning using the TrustGate interoperability gateway's privacy computing capabilities. Modeling effects and prediction accuracy were examined for cases with identical sample features and varying feature sets, as shown in Table 5. Trends in AUC and F1 values were observed across different feature configurations. Modeling Results: AUC Values: Performance rankings observed were as follows: Traditional modeling with 20 features < Federated Learning with 41 features < Traditional modeling with 41 features. F1 Values: The same performance order was noted—Traditional modeling with 20 features < Federated Learning with 41 features < Traditional modeling with 41 features. The findings indicate that federated learning on the TrustGate privacy computing platform demonstrates performance levels comparable to traditional machine learning models trained on complete feature sets. Both AUC and F1 metrics showed significant improvements over models with a limited number of features, reflecting enhanced model accuracy and predictive power. These results underscore the effectiveness of the TrustGate interoperability gateway in supporting robust modeling in privacy-sensitive federated learning environments.

Table 5. Comparison of modeling effectiveness between federated learning on the trustgate interconnection gateway privacy computing platform and centralized machine learning experimental results

Comparison item	Sample size	Number of features	AUC	Precision	Recall	F1 Score
Privacy Computing Platform (Interconnection Gateway)	2 million training/ 5 million validation	41	0.792	0.502	0.225	0.311
Traditional Modeling (All Features)	2 million training/ 5 million validation	41	0.796	0.536	0.223	0.315
Traditional Modeling (Partial Features)	2 million training/ 5 million validation	20	0.779	0.452	0.184	0.241

6.5. Experimental conclusions

The experimental results provide strong evidence of the successful integration and validation of the interoperability platform, which combines the TrustGate gateway with the FATE and SecretFlow privacy computing frameworks. This integration has proven effective in both functional and performance aspects, with the TrustGate gateway enabling seamless communication between the two privacy computing platforms. Through this integration, the system was able to conduct both horizontal and vertical federated learning experiments within the "single migration and integration" scenario, showcasing its ability to support diverse federated learning operations.

Performance and Federated Learning Validation: The performance of the federated learning models was found to be comparable to that of traditional model training methods, provided that the same sample size was used in both cases. This indicates that the integration of the TrustGate gateway with the FATE and

SecretFlow platforms does not compromise the performance of federated learning compared to centralized training models. Additionally, as the feature dimensions of the models were increased, the effectiveness of federated model training showed noticeable improvements. This progression highlights that the TrustGate-enabled platform can scale with the complexity of data, enhancing model performance as more features are incorporated.

Validation of Privacy and Security Features: The validation experiment was conducted using the FATE framework integrated with the TrustGate gateway, confirming several critical aspects related to privacy and security in privacy-preserving computations. Specifically, the experimental results validated the following aspects: Data Invisibility: The system ensured that sensitive data remained invisible during the federated learning process, effectively preserving privacy; Data Trustworthiness: The system demonstrated that the data used for training and analysis could be trusted, as it was securely processed through the federated learning framework without direct access to sensitive data; Data Measurability: The ability to accurately measure the performance of federated learning models, while maintaining privacy, was confirmed, indicating that data metrics could be reliably used to assess model effectiveness without compromising privacy; Achieving Interoperability: A key component of the experimental validation was the introduction of the Adaptation framework, which played a pivotal role in enabling interoperability between heterogeneous federated learning platforms, such as FATE and SecretFlow. This framework allowed for the smooth exchange of data and model parameters between the different platforms, achieving interoperability in federated learning scenarios. Some specific experiments focused on this interoperability yielded positive results, meeting the expected outcomes in terms of system performance and integration between the platforms.

Our research both validates previous proof of federated learning efficiency through its new capability of seamless operation between various privacy platforms. The implemented integration solves one of the main deployment challenges in existing privacy-preserving machine learning by enabling heterogeneous system functionality. The research established that using TrustGate in a federated learning system enables both performance standards and privacy requirements to be achieved. The system delivers cooperative machine learning for various distributed networks while maintaining full data security, together with precise model performance. The findings demonstrate potential applications in analytic processes involving sensitive data that belong to finance organizations and healthcare and telecommunications sectors.

6. CONCLUSION

Privacy computing is becoming essential for secure collaboration in sectors like finance, healthcare, and government. To unlock its full potential, future work must focus on making different systems work together seamlessly and on creating shared global standards. These efforts will make it easier for organizations to collaborate while protecting sensitive data. Technologies like TrustGate show that secure and effective cross-platform learning is possible without sacrificing data privacy or performance. Moving forward, research should aim to deploy such solutions in real-world settings, support a wider range of data types, and align with international data protection laws. This will help build a safer, more connected digital environment where privacy and innovation go hand in hand.

FUNDING INFORMATION

Authors state no funding involved.

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability does not apply to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Federated learning: Collaborative machine learning without centralized training data," *Google AI Blog*, 2017. https://ai.googleblog.com/2017/04/federated-learning-collaborative.html.

[2] I. Arévalo and J. L. Salmeron, "A chaotic maps-based privacy-preserving distributed deep learning for incomplete and non-IID datasets," *IEEE Transactions on Emerging Topics in Computing*, vol. 12, no. 1, pp. 357–367, Jan. 2024, doi: 10.1109/TETC.2023.3320758.

[3] K. Bonawitz, P. Kairouz, B. McMahan, and D. Ramage, "Federated learning and privacy: Building privacy-preserving systems for machine learning and data science on decentralized data," *Queue*, vol. 19, no. 5, pp. 87–114, Oct. 2021, doi: 10.1145/3494834.3500240.
[4] K. Bonawitz, P. Kairouz, B. Mcmahan, and D. Ramage, "Federated learning and privacy," *Communications of the ACM*, vol. 65, no. 4, pp. 90–97, Mar. 2022, doi: 10.1145/3500240.
[5] A. Google, "Privacy-enhancing technologies for federated learning," *Research Paper. Retrieved from*, 2021. https://research.google/pubs/archive/43050.pdf.
[6] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, and V. Ivanov, "Towards federated learning at scale: System design," in *Proceedings of the Conference on Systems and Machine Learning (SysML)*, Nov. 2019, pp. 1–16, doi: 10.1109/IEEECONF44664.2019.9049066.
[7] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, Jan. 2019, doi: 10.1145/3298981.
[8] M. H. Alsharif, R. Kannadasan, W. Wei, K. S. Nisar, and A. H. Abdel-Aty, "A contemporary survey of recent advances in federated learning: Taxonomies, applications, and challenges," *Internet of Things (Netherlands)*, vol. 27, p. 101251, Oct. 2024, doi: 10.1016/j.iot.2024.101251.
[9] C. Ren *et al.*, "Advances and open challenges in federated learning with foundation models," *arXiv preprint arXiv:2404.15381*, 2024.
[10] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Foundations and trends® in machine learning*, vol. 14, no. 1–2, pp. 1–210, 2021.
[11] D. Chai, L. Wang, L. Yang, J. Zhang, K. Chen, and Q. Yang, "A survey for federated learning evaluations: goals and measures," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 10, pp. 5007–5024, Oct. 2024, doi: 10.1109/TKDE.2024.3382002.
[12] Z. Zhao *et al.*, "Towards efficient communications in federated learning: A contemporary survey," *Journal of the Franklin Institute*, vol. 360, no. 12, pp. 8669–8703, Aug. 2023, doi: 10.1016/j.jfranklin.2022.12.053.
[13] T. Yang, G. Andrew, H. Eichner, and F. Beaufays, "Applied federated learning: Improving Google keyboard query suggestions," *Google AI Research Blog*, 2018. https://research.googleblog.com/2018/04/applied-federated-learning-improving.html.
[14] S. Banabilah, M. Aloqaily, E. Alsayed, N. Malik, and Y. Jararweh, "Federated learning review: Fundamentals, enabling technologies, and future applications," *Information Processing and Management*, vol. 59, no. 6, p. 103061, Nov. 2022, doi: 10.1016/j.ipm.2022.103061.
[15] K. Singhal, H. Sidahmed, Z. Garrett, S. Wu, J. Rush, and S. Prakash, "Federated reconstruction: Partially local federated learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 11220–11232, 2021.
[16] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, May 2020, doi: 10.1109/MSP.2020.2975749.
[17] Z. Yang, M. Chen, K. K. Wong, H. V. Poor, and S. Cui, "Federated learning for 6G: applications, challenges, and opportunities," *Engineering*, vol. 8, pp. 33–41, Jan. 2022, doi: 10.1016/j.eng.2021.12.002.
[18] S. Feng, H. Yu, and X. Li, "Multi-participant multi-class vertical federated learning," *arXiv preprint arXiv:2001.11154*, 2020.
[19] L. Yang *et al.*, "A survey on vertical federated learning: From a layered perspective," *arXiv preprint arXiv:2304.01829*, 2023.
[20] Y. Liu *et al.*, "Vertical federated learning: concepts, advances, and challenges," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 7, pp. 3615–3634, Jul. 2024, doi: 10.1109/TKDE.2024.3352628.
[21] X. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2019.
[22] R. Yan *et al.*, "Label-efficient self-supervised federated learning for tackling data heterogeneity in medical imaging," *IEEE Transactions on Medical Imaging*, vol. 42, no. 7, pp. 1932–1943, Jul. 2023, doi: 10.1109/TMI.2022.3233574.
[23] Y. Liu, T. Fan, T. Chen, and Q. Yang, "FATE: An industrial-grade platform for collaborative learning with data protection," *Journal of Machine Learning Research*, vol. 22, no. 226, pp. 1–6, 2021.
[24] M. Arafeh, H. Otrok, H. Ould-Slimane, A. Mourad, C. Talhi, and E. Damiani, "ModularFed: Leveraging modularity in federated learning frameworks," *Internet of Things (Netherlands)*, vol. 22, p. 100694, Jul. 2023, doi: 10.1016/j.iot.2023.100694.
[25] H. R. Roth *et al.*, "Nvidia flare: Federated learning from simulation to real-world," *arXiv preprint arXiv:2210.13291*, 2022.
[26] H. R. Roth *et al.*, "Empowering federated learning for massive models with Nvidia flare," *In Federated Learning Systems: Towards Privacy-Preserving Distributed AI*, 2025.

BIOGRAPHIES OF AUTHORS

Dr. Akhila Reddy Yadulla holds an MS in Computer Science from Northwestern Polytechnic University in 2016, an MS in Information Technology Management from Campbellsville University in 2019, and a Ph.D. in Information Technology from the University of the Cumberlands in 2024. She is an accomplished Senior Software Engineer with over eight years of experience in Java full-stack development and expertise in the Software Development Life Cycle. Dr. Yadulla is proficient in Java, Spring Boot, Hibernate, RESTful APIs, Angular, React, Microservices, SQL/NoSQL databases, DevOps, and Agile methodologies. Her research focuses on Cybersecurity, Artificial Intelligence, Data Science, Big Data, Blockchain Technologies, Amazon Web Services, and Machine Learning. She combines deep technical knowledge, advanced problem-solving skills, and AI-driven innovative solutions in cutting-edge technologies. Notably, she led a team in developing a secure and scalable microservices architecture for significant telecommunication and banking platforms, resulting in a major increase in system performance. She can be contacted at email: Akhilareddyyadulla@ieee.org.

Vinay Kumar Kasula has a master's in technology from Osmania University, followed by a second Master's from Northwest Missouri State University. A deep-rooted interest in research and advanced technologies has driven his academic journey. This passion led him to pursue and complete a Ph.D. at the University of the Cumberlands, where he graduated in 2024, focusing on areas including Blockchain, Artificial Intelligence, Deep Learning, Machine Learning, Data Analytics, and Cybersecurity. With over a decade of industry experience, currently as a Senior Systems Analyst, he has applied and expanded his expertise in these domains, continually exploring and integrating innovative solutions to meet evolving technological challenges. This blend of education, research, and practical experience fueled his commitment to advancements in these cutting-edge fields. He can be contacted at email: Vinaykasula.phd@ieee.org.

Bhargavi Konda works as a senior systems analyst in the human resources information systems department for a healthcare company. She mainly works with data analysis and generates reports for various departments in the organization. She has been one of the major contributors in maintaining employee records, supporting all HR departments, and especially working with payroll operations from her team. She earned her master's degree from Osmania University (2001-2004). She received the gold medal and Star Student of the Year awards for her outstanding performance. She was placed in an IT consulting company right after her master's, and she worked as an Oracle technical consultant from 2004 to 2012. In that period, she was awarded the Outstanding Trainer award for her commitment and dedication. She has experience implementing applications in the banking and manufacturing sectors. She was passionate about higher studies, so she started to pursue a Ph.D. in IT in 2019. She earned a Ph.D. degree in Information Technology in 2024 from the University of the Cumberlands. She is excited to work on many journals and publications in the future and wants to continue her research. She can be contacted at email: Bhargavikonda@ieee.org.

Mounica Yenugula is a skilled IT professional with a strong academic background and experience as a Cloud Engineer. She holds a Ph.D. in Information Technology from the University of Cumberlands, where her research focused on critical security and trust issues in cloud environments. Her academic journey began with a Bachelor of Technology in Electronics and Communication Engineering from Koneru Lakshmaih University and continued with a Master of Science in Electrical Engineering from California State University, Los Angeles. As a certified Cloud Engineer, Mounica has successfully implemented cloud solutions within the financial sector, driving digital transformation. Her active involvement in the Women in IT community and her role as a judge for the Globee Awards demonstrate her commitment to industry advancement and innovation. She can be contacted at email: ymounica.phd@ieee.org.

Supraja Ayyamgari is currently a Ph.D. candidate in Information Technology at the University of the Cumberlands, USA. She holds a robust academic foundation with a bachelor's degree in computer science from Jawaharlal Nehru University, India, a Master's degree in Information Technology from Southern New Hampshire University, USA, and an MBA from New England College, USA. Alongside her academic accomplishments, Supraja has nearly eight years of diverse industry experience in IT, where she has held roles such as Developer, Full Stack Engineer, and DevOps Specialist. She now leads critical infrastructure and cloud transformation projects in the financial sector. Her research interests include blockchain, cryptography, cloud computing, augmented reality (AR), and virtual reality (VR). Supraja is dedicated to advancing her expertise and is eager to contribute to academic publications and journals in the future. She can be contacted at the following email: Sayyamgari32587@ucumberlands.edu.