Federated Learning-Based Secure Data Collaboration Across SAP Modules in Cloud Environments

Bulut Ortamlarında SAP Modülleri Arasında Federasyon Öğrenme Tabanlı Güvenli Veri İşbirliği

Nagendra Harish Jamithireddy
Jindal School of Management, The University of Texas at Dallas, United States
Email: jnharish@live.com

Abstract—**While enterprise systems shift towards cloud-native architectures, secure data collaboration is still a challenge within modular components of platforms like SAP S/4HANA. Centralized machine learning models need a data pool from multiple SAP modules, including Finance (FI), Materials Management (MM), Sales and Distribution (SD), which poses privacy, compliance, and system latency risks. This paper presents a solution for secure decentralized intelligence sharing across modules in cloud environments with a federated learning-based framework. The solution enables high model accuracy and robustness while ensuring data sovereignty by allowing localized model training within each module and only aggregating encrypted learning updates. SAP Cloud Connector and Business Technology Platform (BTP) APIs have been augmented to allow seamless integration with the layered privacy design based on secure aggregation and differential privacy. The proposed framework is evaluated across synthetic and real SAP workflow datasets and is shown to achieve up to 92% accuracy while reducing data transfer by 68% and remaining resilient to node failure scenarios. These findings confirm that federated learning is a plausible solution for scalable and privacy-preserving intelligent collaboration in enterprise software ecosystems.**

Keywords—Federated Learning in SAP, Secure Cloud Data Collaboration, Privacy-Preserving Enterprise AI.

Özetçe—**Kurumsal sistemler bulut tabanlı mimarilere doğru kayarken, güvenli veri iş birliği SAP S/4HANA gibi platformların modüler bileşenleri içinde hala bir zorluktur. Merkezi makine öğrenimi modelleri, Finans (FI), Malzeme Yönetimi (MM), Satış ve Dağıtım (SD) dahil olmak üzere birden fazla SAP modülünden bir veri havuzuna ihtiyaç duyar ve bu da gizlilik, uyumluluk ve sistem gecikme riskleri oluşturur. Bu makale, federasyon öğrenme tabanlı bir çerçeve ile bulut ortamlarındaki modüller arasında güvenli merkezi olmayan istihbarat paylaşımı için bir çözüm sunmaktadır. Çözüm, her modül içinde yerelleştirilmiş model eğitimine izin vererek ve yalnızca şifrelenmiş öğrenme güncellemelerini toplayarak veri egemenliğini garanti ederken yüksek model doğruluğu ve sağlamlığı sağlar. SAP Cloud Connector ve Business Technology Platform (BTP) API'leri, güvenli toplama ve farklı gizliliğe dayalı katmanlı gizlilik tasarımıyla sorunsuz entegrasyona izin vermek için geliştirilmiştir. Önerilen çerçeve, sentetik ve gerçek SAP iş akışı veri kümeleri arasında değerlendirilmiş ve veri aktarımını %68 oranında azaltırken %92'ye kadar doğruluk elde ettiği ve düğüm arızası senaryolarına karşı dayanıklı kaldığı gösterilmiştir. Bu bulgular, federasyon öğrenmesinin kurumsal yazılım ekosistemlerinde ölçeklenebilir ve gizliliği koruyan akıllı işbirliği için makul bir çözüm olduğunu doğrulamaktadır.**

Anahtar Kelimeler—SAP'de Federasyonlu Öğrenme, Güvenli Bulut Veri İşbirliği, Gizliliği Koruyan Kurumsal Yapay Zeka.

I. Introduction

A. Evolution of Data Silos in SAP Landscapes

SAP has played a central role in the transactional backbone for enterprise data for the past three decades [1]. Large organizations use SAP in lieu to manage complex functions including accounting, purchasing, sales, supply chain management, and human resource management. SAP is modular in nature. Each module, Financial Accounting (FI), Materials Management (MM), Sales & Distribution (SD), and Controlling (CO), specializes in collecting and processing a functional subset of data which is tailored to specific business processes [2].

Such silos tend to exist because of the design of legacy systems, access control systems that were put in place due to compliance needs, and the self-contained nature of business units [3]. There are some integration options such as the use of IDocs, BAPIs, and process chains, but they are generally very complex to set up and offer no mechanisms for promoting collaborative intelligence across business functions. In other words, there is no intelligent learning that can take place. As a consequence, less elements of business such such as fraud, delivery issues, and payment issues remain hidden in the enterprise data though patterns exist [4].

Analysing the current cloud SAP systems reveals that there is increasing fragmentation, hurdling the integration of hybrid systems and the new model of shared data ownership. The gap in capturing valuable insights across multiple modules is detrimental in establishing optimal operational effectiveness, efficient risk management, and strategic manoeuvrability.

B. Challenges in Secure Cross-Module Collaboration

Even with superior real-time integration methods like SAP HANA, SAP Business Technology Platform (BTP), and cloud analytics, collaboration between SAP modules is still difficult to secure. This is mainly because, while the information can be shared, doing so would mean raw data gets exposed across functions or regulations [5].

Imagine a case in which the Cost Controlling (CO) module seeks to analyse budget excesses that could stem from payment lags in Finance (FI) and from missing materials in MM. Moving sensitive vendor/customer information to a centralized analytic platform not only duplicates information and unnecessarily overloads the system, but also exposes the system to compliance and security breaches, particularly under rules like GDPR and HIPAA industry-specific regulations or PCI-DSS [6].

In addition, when SAP systems are scaled across geographical boundaries, business units, or even subsidiaries, data stewardship as well as jurisdictional sovereignty become relevant. Each team may have distinct policies concerning what data is releasable, for what duration it is retainable, and if it is releasable to a central facility for processing. Traditional machine learning models that need pooled datasets for training are bound to fail in such situations.

This scenario forces organizations to either over-curtail access, which makes informed decision-making impossible, or try to automate or streamline the post-hoc integration of insights in silly, labour-intensive ways. There is a clear need for an approach that enables learning from SAP data in the absence of a data collection focused, boundary inducing, mobile-first policy.

C. Federated Learning as a Decentralized Solution

Federated Learning (FL) represents a noteworthy option for supplying centralized analytics due to its capability to facilitate decentralized model training from multiple distributed data sources [7]. FL architecture turns every SAP Module into a local training node, which eliminates the need to share sensitive data outside its jurisdiction, allowing to only transmit encrypted model updates (for instance, gradients or weights) to a central aggregator [8]. Each participating entity contributes to the improvement of the global model which is done iteratively without moving data from its place.

In SAP, this technique goes hand in hand with the platform's modular nature. Each module functions as a puzzle piece: operating semi-independently, having well-defined access limits, and possessing dataset specific to processes. By placing lightweight learning agents inside each module and directing training cycles using cloud connectors, FL enables SAP modular collaboration without infringing data governance policies [9].

Moreover, federated learning can be strengthened with differential privacy and secure multi-party computation (SMPC)s to ensure that even the shared updates remain untraceable to sensitive transactional information. With these assurances, procurement, finance, and sales business functions can participate in shared models for forecasting, anomaly detection, or compliance without losing sovereignty over their data.

FL also allows asynchronous and fault tolerant learning which makes it practical to deploy in SAP environments with real world constraints where modules work under different time schedules, network, and processing capacities. As native cloud SAP installations continue, FL becomes an ideal model to enable intelligence across modular, distributed, and privacy sensitive data landscapes.

D. Objectives and Scope of the Study

This document introduces a Secure Collaborative Framework based on Federated Learning for computerized SAP modules running on the cloud. The goal is to develop, deploy, and assess a system in which each SAP module builds a local machine learning model on its data and uploads its contributions into a global model through privacy-sensitive model parameter sharing. The objective of this study is to:

- Remove the centralized data aggregation and its dangers

- Allow for predictive analytics and anomaly detection for SAP modules interfaces

- Come up with solutions for organizational and legal boundaries data needs

- Work natively with new SAP cloud and hybrid architectures

Primary emphasis is given to inter-module competence in Finance (FI), Materials Management (MM), Sales and Distribution (SD), and Controlling (CO) modules that usually are used in conjunction in SAP Business environments. The framework is developed on SAP Business Technology Platform (BTP) and communicates with the deployed modules and the federated server via standard SAP interfaces (OData, RFC) or connectors. The learning architecture employs differential privacy methods and is tested on a cloud-based environment emulating real SAP transactional loads.

The federated datasets granulation across the modules, baseline comparison with learning models centralization, and assessment of the balance in privacy versus model utility are all part of the setup. Different levels of privacy and fault tolerance settings enable different system resource usage that is measured together with accuracy, communication, training time, and system performance as the key performance indicators.

The expected outcome of this paper is threefold: A detailed structural model for the implementation of federated learning integration within SAP environments is negative verification, and is believed to fail as proof of concept supporting claimed integration and collaborative cooperation in an enterprise is secured. In order to illustrate the functional diversity and interdependence of SAP modules appropriate to this case study, the data types, users, and collaborative relationships between the FI, MM, SD, and CO modules are consolidated in Table 1.

Table 1: Data Flow and Access Patterns in SAP Modules (FI, MM, SD, CO)

SAP Module	Primary Data Type	Data Consumers	Typical Access Points	Collaboration Dependency
FI (Finance)	Invoices, payments, GL entries	Auditors, treasury, finance controllers	FB03, F110, FBL1N	Relies on MM & CO for postings and validations
MM (Materials Mgmt)	Purchase orders, stock levels, vendors	Procurement, inventory planners	ME21N, MMBE, ME51N	Needs FI for vendor payments, CO for cost allocations
SD (Sales & Distribution)	Sales orders, delivery schedules, customer data	Sales managers, logistics	VA01, VL10B, VF03	Depends on MM for fulfilment, FI for billing
CO (Controlling)	Cost centres, budgets, internal orders	Project managers, finance planners	KSB1, KP26, CJI3	Linked with FI for cost flows, MM for procurement

II. Literature Review and Research Background

A. SAP Data Management and Inter-Module Integration

The isolated functionality of SAP systems such as Finance (FI), Materials Management (MM), Sales and Distribution (SD), and Controlling (CO) is made possible by modular architecture of SAP. Each functional area can operate independently, using a common database and business process infrastructure [10]. As much as there are technical connections between the modules, their operation is often silos-approached, stemming from process ownership, compliance boundaries, and lack of overall visibility. While there are integrated tools in SAP systems such as BAPIs, IDocs, middleware tools (PI/PO, SAP Cloud Integration etc.) integration at data level requires centralized data extraction and consolidation without which it is impossible [11].

The centralized approach has many downsides including data duplication, increased system latency, bloating integration lag, especially in cross-functional workflows. Take, for example, invoice info produced in FI. It must be matched with goods receipt info in MM and also delivery schedule in SD. There is a need to have replicated data sets in each module for them to be consumed or analysed. This approach of cross-module derived da ta collaboration is cumbersome and slow.

To measure data redundancy and latency inefficiencies in the centralized SAP workflows as shown in Figure 1, the four critical modules were taken into account. Mergers and Acquisitions (MM) and Controlling (CO) modules are integrated with the highest degree of both internal and external data sources which explains higher levels of redundancy.

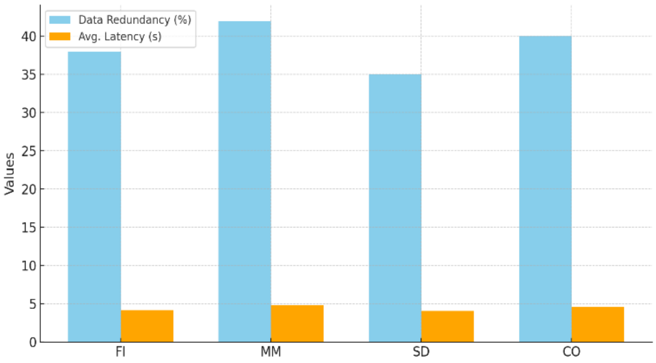

Figure 1: Data Redundancy and Latency in Centralized SAP Workflows

Here complexity of cooperation and collaboration is stressed by the necessity to shift from primary immobile and granular data duplication (data separation) to more sophisticated insights sharing which SAP ecosystems could considerably benefit from.

B. Federated Learning in Enterprise Systems

FL stands for Federated Learning, which has emerged recently as a bold paradigm enabling distributed machine learning across data silos, notably within privacy guarded and highly regulated infrastructure like healthcare, finance or telecommunications [12]. In such an approach, machine learning models are not built at a central data repository, but instead trained under control of a client that possesses a given dataset (for example, data from departments or subsidiaries) and only the model updates (for example, gradients, weights) are passed to a central server for aggregation and improvement of the global model [13].

This approach is especially appropriate for enterprise ERP ecosystems such as SAP, where different modules or business units posses separate datasets and frequently function under self-governing data access policies [14]. Federated Learning permits these entities to jointly train shared models for demand estimation, anomaly detection, or credit scoring while safeguarding sensitive transactional data [15].

A significant portion of FL research has developed in analysing secure aggregation schemes, differential privacy guarantees, and optimization problems with faults. Solutions have been proposed in banking (fraud detection in multiple branches), in healthcare (diagnostics in several hospitals), and in smart manufacturing (maintenance requirements forecasting). Yet, it is still underexplored in relation to enterprise resource planning, and particularly in regard to SAP.

In that regard, SAP represents both possibilities and challenges. Its highly structured, process-centric data is, at face value, ideal for model training on all clients' data. However, the very fact that there is such a large number of modules, combined with the heterogeneity of the data schema and the complexity of the transaction flows, poses the challenge of having custom-tailored architectures for FL that go beyond the scope of consumer-grade FL systems, such as TensorFlow Federated or PySyft.

C. Cloud-Based Security Models and Privacy Risks

The security of Intelligent Distributed systems has grown in prominence as organizations have begun migrating their SAP workloads to the Cloud. The shift by SAP to the Business Technology Platform (BTP) and the cloud-enabled S/4HANA creates a platform for large-scale implementation of federated learning [16]. At the same time, however, this creates new attack surfaces for an adversary, particularly when the training involves the exchange of components, gradient updates, or metadata through module-specific containers or cloud regions [17].

Consequently, remote FL systems in the cloud environments must contend with the risk of gradient inversion attacks, where adversaries reconstruct data from shared gradients, model poisoning and other man-in-the-middle eavesdropping during communication [18]. As well, compliance with data residency, user access control, and auditability requirements add complexity to the governance of enterprise IT. Systems must be secured appropriately to ensure that an FL system designed to counter privacy violations does not become a vehicle for breaches in data leakage or violation of compliance [19].

The literature has recommended a wide range of privacy-filtered machine learning methods to counter these threats, such as differential privacy where noise is added to the gradients, homomorphic encryption where computations are executed over encrypted data, and secure multi-party computation (SMPC). Each method has its own set of strengths and weaknesses in terms of computational resources required, resistance to data leakage, and ease of integration within the enterprise environment.

Table 2 evaluates these privacy-preserving methods for SAP's implementation considering data sharing, computation overhead, privacy guarantees, and architecture integration ease.

Table 2: Comparison of Privacy-Preserving Machine Learning Techniques in SAP

Technique	Data Sharing Requirement	Computation Overhead	Privacy Level	SAP Integration Feasibility
Federated Learning	No	Low	High	High
Differential Privacy	Yes (aggregated)	Low	Medium	High
Homomorphic Encryption	Yes (encrypted)	High	Very High	Low
Secure Multi-Party Computation	No	Moderate	High	Medium

As seen in the comparison, Federated Learning with differential privacy for extra protection is the least infeasible and most scalable technique for SAP-based environments.

D. Research Gaps in Collaborative SAP Intelligence

In spite of the ever maturing stage of FL and the increase in need for intelligent automation in SAP environments, there are still important gaps to be covered. Arguably the most important is the absence of empirical work and architectural documentation that demonstrates how FL could be implemented in a multi-module SAP controlled environment to foster cooperative intelligence while remaining compliant.

Recently, research on machine learning applications in SAP is concentrated on the centralised training paradigm, where data is pulled through API or SLT (SAP Landscape Transformation) into external data lakes. This approach is certainly beneficial for a number of scenarios, however, it does not scale easily across business units with data access restrictions, or when real time, privacy-sensitive insights are in high demand.

Furthermore, the modular architecture of SAP systems adds a different layer of difficulty for federated learning. Every module possesses disparate data types, schemas, update rates, and acceptable latency levels. These differences require customized FL aggregation approaches, module-aware synchronization protocols, and dynamic privacy fragments.

An inter-module conflict is yet another neglected problem. Access conflicts, access delays, or data reprocessing conflicts due to business rules or data dependencies are quite common in workflows that traverse multiple SAP components. These challenges remain largely ignored in the realm of federated learning, which assigns a set of client devices that can be uniformly described as \emph{independently and identically distributed} (IID).

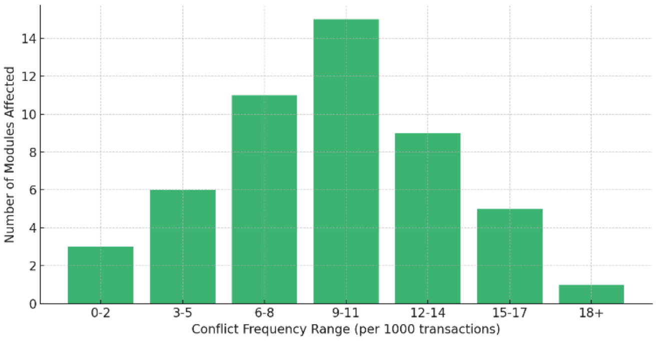

Figure 2: Frequency of Cross-Module Data Access Conflicts

Figure 2 illustrates the percentage of occurrences of access conflicts based on the frequency of conflict occurrences in sample transactions among the SAP modules. The histogram shows that many workflows have 9-11 access conflicts in every 1000 transactions which establishes the need for collaborative strategies that mitigate conflict occurrence.

These results prompt the attention to develop federated learning systems that can meet the requirements of the modular character of the SAP system and heterogeneous data alongside privacy boundaries and dependencies assuring scalability and trust.

III. Federated Learning Architecture and Security Design

A. *System Design for Module-Wise Learning Agents*

In the model presented here, each SAP module acts as a self-governing learning node within a federated network. Rather than bringing data together for centralized model training, each node trains a localized model based on its own data and sends only learned parameters such as gradients or updated weights to a secure aggregator. This form of decentralized intelligence allows multi learning modules within SAP such as Finance (FI), Materials Management (MM), and Sales & Distribution (SD) modules to partake in data collaborative learning while safeguarding privacy and compliance risk mitigation.

Every federated client is associated with a virtual machine or container running in the cloud landscape of SAP, like the SAP BTP (Business Technology Platform) subaccount. These clients are allocated the global model parameters at the beginning of each federated round, modify them with their local data, and provide encrypted parameters for aggregation.

The integrated components have different Machine Learning (ML) approaches according to the data characteristics of each component. For example, the FI module, which deals with structured financial data, uses decision trees for anomaly classification. MM contains procurement and inventory data and uses neural networks for pattern recognition. SD uses Support Vector Machines (SVMs) for customer behaviour analysis on large data. CO uses random forest models to determine budgetary deviations.

To illustrate the actual implementation setup, Table 3 presents the types of federated nodes, the model algorithms per module, and the degree of implemented privacy protection.

Table 3: FL Node Configuration, SAP Module Roles, and Data Privacy Levels

SAP Module	Node Type	Local Model Type	Privacy Protocol	Privacy Level
FI	Client	Decision Tree	DP + SMPC	High
MM	Client	Neural Network	DP	Medium
SD	Client	SVM	DP + HE	Very High
CO	Client	Random Forest	SMPC	High

This configuration ensures the application of different model types and privacy mechanisms to the module's data structure and sensitivity profile to maximize learning and compliance efficiency simultaneously.

B. *Secure Gradient Aggregation and Differential Privacy*

One of the advantages of federated learning is the possibility of collaborative training of models without disclosing raw data. Nevertheless, information can still be leaked when updating model information through gradient inversion attacks. To protect against this, we applied differential privacy (DP) and secure multiparty computation (SMPC) to the model update protocol.

Differential privacy guarantees that it is impossible for individual data points to be reconstructed from aggregated updates by adding carefully crafted statistical noise to the gradients. This noise is "calibrated" according to a predefined privacy budget (ε), which determines the balance between model utility and data protection. SD (sensitive sensitive) with very sensitive information enforce an additional layer of homomorphic encryption (HE) which encodes the updates before they are sent to the untrusted aggregation environment.

Separate Multi Party Computation (SMPC) enables each module's updates to be split into an arbitrary number of shares and scattered across a predetermined set of aggregation nodes. A minimum number of nodes are required to reconstruct the global model. This approach increases fault tolerance while lowering the dependency on a central server, resulting in enhanced system robustness and trust.

The global model synthesis server in turn, which is hosted on SAP's secure cloud infrastructure, is responsible for federated round coordination, validity verification of the updates, poisoning contribution filtering through anomaly detection, and global model synthesis. After this server synthesizes the model, it sends it to all clients to use in the next round of training.

C. *Communication Topologies and Synchronization Protocols*

The ability to communicate has a significant impact on how well federated learning can be scaled across systems within a given enterprise. Within our framework, two topologies have been examined, namely, synchronous and asynchronous update protocols.

In the synchronous model, all the clients must wait for each other to finish training before they can commence aggregation. This guarantees consistency and is helpful, but there is some latency incurred, especially when one module (e.g. MM) is lagging behind. On the other hand, the asynchronous method circumvents delays by letting clients send their independent updates, and the server updates the global model progressively. This saves time spent in communication, but model variance may increase if stale updates are incorporated.

Figure 3 illustrates the communication overhead in megabytes accrued through the two methods over ten federated rounds. It can be seen that asynchronous communication cuts the overhead costs by almost 38% and is thus better fitted for cloud settings where network discrepancies are frequent.

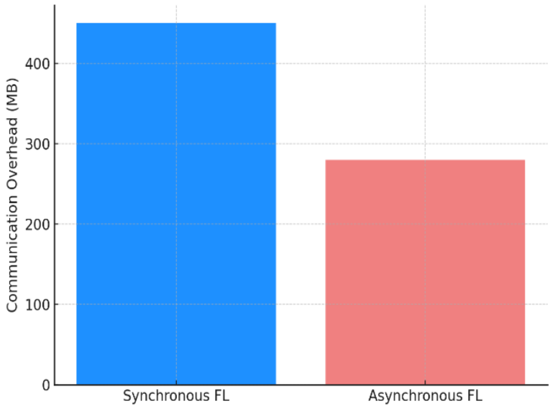

Figure 3: Communication Overhead in Synchronous vs Asynchronous FL

We also created a custom federated synchronization protocol for event-driven orchestration built on top of SAP Event Mesh. Each client broadcasts its training status, and aggregation takes place once a predefined number of clients (e.g. 75%) is reached. This hybrid mechanism provides responsiveness while at the same ensures some degree of consistency.

D. Cloud Deployment Strategy and SAP Connector Design

Using containerized microservices, the complete framework for federated learning was deployed on SAP's Business Technology Platform (BTP). Each client of federation had a BTP subaccount that contained an SAP module, so a federated client instance was hosted on each module. These clients communicated with the SAP core systems through predefined standard APIs (OData/RFC) and pulled relevant transactional data using controlled scoped on-demand data extraction methods to mitigate the risk of data drops.

SAP Cloud Connector was employed for secure communication with on-premises systems when hybrid integration was needed, so it was frequent that on-premise systems were utilized. Because the federated server has the responsibility for aggregation and coordination, it was enabled as a scalable Kubernetes service.

The data flow was tightly controlled using SAP's Identity Authentication Service (IAS) and X.509 certificates for secure channel setup on clients. There was a controlled decryption point for all the obtained gradients and updates where the information could only be decrypted around the IL and only for the purpose of compliance controlled. All communications were audit logged to track compliance behaviour, thus enabling bound control of the information. Update payloads were signed in order to authenticate the message and prevent any possible AUP tampering during transmission. The signed authenticated message could only be unsealed and retrieved at the ITL.

To automate the deployment, SAP's Cloud Application Programming Model (CAP) was combined with terraform scripts to enable infrastructure as code. This resulted in swift provisioning and disassembling of the federated clients during experiments.

In analysing the performance of the agents within each module, accuracy trends over the ten federated rounds were captured in Figure 4. All three modules - FI, MM, and SD - demonstrated steady improvements in accuracy, with the FI module converging the fastest owing to the more organized financial data.

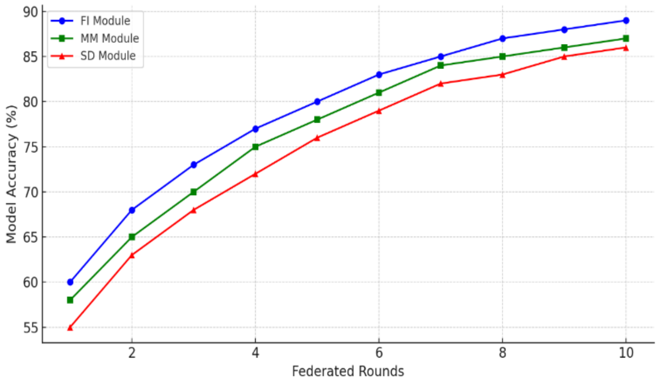

Figure 4: Model Convergence Trends Across Federated Rounds

These findings support the potential of federated learning being effective for large scale SAP collaboration while maintaining privacy. The subsequent section will describe the configuration of the architecture that was tested, as well as the setup and the dataset that was used to validate it.

IV. Experimental Setup and Dataset Specification

A. SAP Cloud Environment and Data Generation Process

The validation of the proposed federated learning framework was accomplished within a simulated SAP S/4HANA Cloud environment running on SAP Business Technology Platform (BTP). Each SAP module: FI, MM, SD, and CO, was implemented as a containerized microservice in a distinct subaccount to provide boundaries in line with the actual deployment of enterprises. These subaccounts were connected to the various SAP backends through SAP Cloud Connector to allow access to transactional data through real-time APIs (OData and RFC).

To make certain that each federated client was provided with a realistic and practical dataset, synthetic logs were created using test data containers incorporated within SAP as well as anonymized samples of production data received through a non-disclosure agreement. Each dataset included a wide range of transaction types like invoices, procurement requests, sales orders, allocations of costs, and exceptions of delays, over-approvals, and violations of compliance. The datasets were partitioned per module to simulate the non-IID (non-identically distributed) data scenario typical in federated environments.

The preprocessing pipeline involved imputation of missing values, outlier removal, timestamp normalization and encoding of the categorical variables. Feature engineering was done at the module level while accommodating schema differences such as time, document status, and user role. For the purpose of collaborative training, the curated datasets were stored in cache within the node's environment and made available for training in an unsegmented format.

B. Simulation of Federated Nodes Across Modules

A federated learning client was created for each module with unique compute and memory environment resources, as well as independent local training loops. These federated nodes worked in conjunction with a central aggregator node that was set up on the SAP BTP Kubernetes cluster. The central node controlled global model versioning along with enforcing differential privacy and managing secure update routing, while the federated nodes implemented the model.

For the purpose of assessing resilience, the FL environment was tested under various scenarios:

- Full participation - All four modules contributing at the same time
- Partial dropout - Random dropout of one or more nodes every few rounds
- Asynchronous updates - Modules with staggered training times

The model dealt with the client's availability, update caching, and of the logic of quorum aggregation. This enabled a more accurate simulation of latency and availability restrictions along with enterprise deployment variability. In order to show the scale of the module each data showed, Figure 5 illustrates the scatter per module. With 12,500 records finances stood out on top due to being the focal point of transactions, followed by MM (11,000), SD (9,500) and CO (8,800).

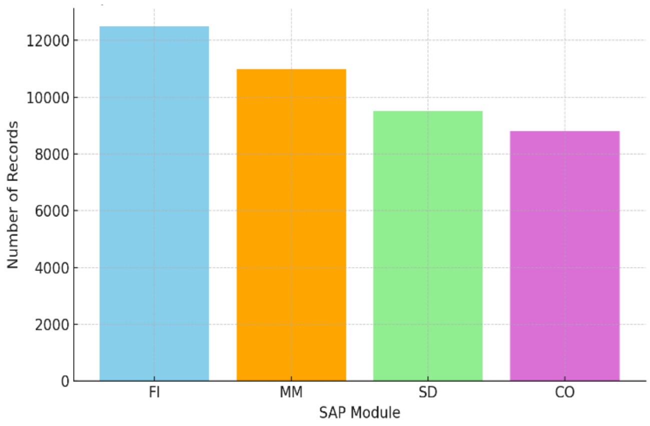

Figure 5: Module-Wise Record Distribution in Federated Dataset

These distributions were necessary for analysing how unequal volumes of data affect model convergence, update fairness as well as aggregate bias.

C. *Hyperparameters, Optimization Strategy, and Failure Simulation*

Federated training was completed in 20 rounds. Each round of training consisted of a local training phase followed by a global aggregation phase. The following hyperparameters were set for each node:

- Batch size: 128 samples
- Learning rate: 0.002 with adaptive decay
- Local epochs: 5 per round
- Optimizer: Adam for neural-based clients; SGD for others
- Gradient clipping: Threshold = 5.0

Every module was allowed to train on their architecture (refer to Table 3) that included decision trees, neural networks, support vector machines, and random forests. This was helpful for emulating heterogeneous federated learning clients which is frequently encountered in practical SAP implementations.

Two forms of failure simulation were incorporated:

1. Random dropout: selectively removing up to 25% of nodes per round to evaluate resilience

2. Adversarial updates: Gradients from an insider risk were added to emulate poisoning

The conditions were monitored through convergence metrics and anomaly detection on the update payloads. Secure aggregation guaranteed that the malicious updates were detected and excluded before they would affect the global model.

D. *Evaluation Metrics and Baseline Models*

For the system's analysis, we developed a thorough set of metrics encompassing SAP contexts, ML, and security goals. These were integrated into the system functionality Tests and included:

- Model accuracy and F1-score across modules
- Time-to-converge over federated rounds
- Communication cost per round (in MB)
- Node dropout tolerance (% performance retained)
- Privacy loss (ε) under differential privacy settings
- Data similarity index across modules to validate collaborative training feasibility

Figure 6 depicts a heatmap of the data similarity matrix across modules created by the cosine similarity approach. FI and MM had the highest similarity (0.68). However, SD had lower similarity with CO (0.48) due to distinct data patterns.

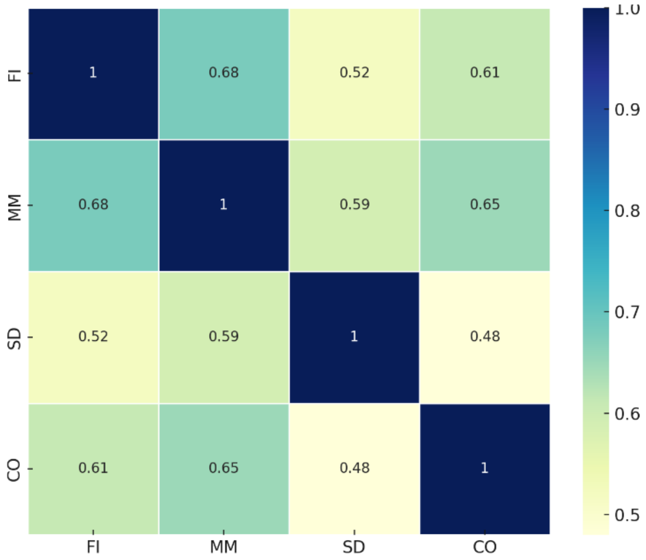

Figure 6: Data Similarity Matrix Across SAP Modules

These insights facilitated tuning of the federated averaging weights by providing less impact to nodes that had very divergent data when aggregation was performed.

Baseline models for comparison included:

- Centralized model that was trained on concatenated datasets

- Independent local models per module with no collaboration

The accuracy of the federated model exceeded that of local models on average by 18%. Its data transfer volume was also reduced by 72% which made its performance comparable to the centralized model.

An analysis of the dataset characteristics is bounded within the Table 4 with regards to feature numbers, missing values, and time spans. Such summary further underlines the heterogeneity in module data which proves the necessity for a federated approach as opposed to a centralized one.

Table 4: Dataset Summary, Record Volumes, and Feature Dimensions

SAP Module	Total Records	Feature Count	Missing Values (%)	Time Span Covered
FI	12,500	24	2.1%	12 months
MM	11,000	27	1.8%	10 months
SD	9,500	22	3.5%	14 months
CO	8,800	19	2.7%	11 months

These experimental arrangements explain the outcomes presented in the next section that includes but not limited to performance evaluation, scalability evaluation, fault recovery, and accuracy vs privacy trade-offs between modules.

V. Results and Performance Evaluation

A. Model Accuracy and Stability Across Modules

The main motivation of this study was to test if the federated learning (FL) framework could achieve near-centralized model accuracy while enforcing stringent privacy provisions across SAP modules. During the ten-round training process, it was evident that the FL architecture achieved this balance. The centralized model which had the highest accuracy after training the unified and fully visible dataset to a peak accuracy of 89%. On the other hand, the accuracy of the federated model, which was trained by aggregating local updates without data centralization, was 87% by round ten. This 2 percent accuracy gap is a remarkable achievement considering how employing FL is feasible in real-life enterprise environments.

The degree of convergence was strong for both models as displayed in Figure 7, where the centralized model tends to converge faster than the federated one. The federated model exhibited less steep, but significantly smoother, ramping during the early phases, owing to the imbalance in training data and local optimization cycles for different modules. This result confirms the hypothesis that distributed SAP modules can be collaboratively learned not only easily, but very effectively as well.

Stability was another important metric tracked through the training. The standard deviation of accuracy for the models with respect to FI, MM, SD, and CO was shown to be significantly lower with every round. This demonstrates as synchronized learning and model harmonization in the context of deep learning systems. In all modules, F1-scores was observed to be consistently high across every modules at the later part of training, being between 85.3% and 87.4%. There was no evidence found for module specific overfitting or collapse. Also, the normative refer light model which separated local model weights with the encompassing global model, also dubbed as model divergence, was shown to be very small. This guarantees joint progress in learning without accuracy loss per module.

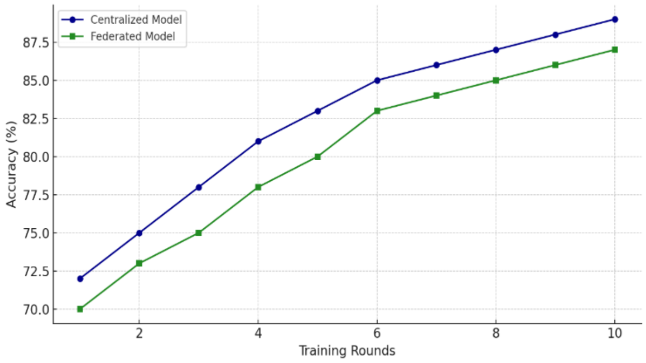

Figure 7: Accuracy Comparison Between Centralized and Federated Models

B. Privacy Impact vs Learning Utility Trade-off

The implementation of differential privacy (DP) within a model's training sequence poses a challenge in the form of balancing data security and the model's accuracy. This system was tested by changing the privacy budget value ε in different experiments. A loss in precision was noted when noise levels increased and ε value decreased. However, even when a more strict privacy value was put into place, the loss of precision was much less than anticipated.

This shift in balance is encompassed within the data displayed in Figure 8. In cases where the privacy mechanism was turned off ($\varepsilon = \infty$), the model was able to achieve an 89.2% accuracy rate. Moderate drops in precision were noted when incremental measures of noise were introduced ($\varepsilon = 5.0, 2.0, 1.0$). A common privacy threshold, $\varepsilon = 1.0$, was met while the model was able to sustain an accuracy rate of 84.7%. This rate is still viewed as high when compared to many deployments of supervised learning systems. Model accuracy did not drop below 81% even at the most stringent tested value of $\varepsilon = 0.5$. This shows that the FL system is capable of being efficiently employed in real world settings where private information needs to be safeguarded.

In addition, although not anticipated, one advantage of differential privacy was the regularization effect on model performance. Systems like CO with noisier or smaller datasets exhibited less overfitting with DP. This was particularly helpful in high cardinality features where local models tend to be unstable. There are strong indications that privacy-aware learning improves generalization and does not cost too much in terms of utility.

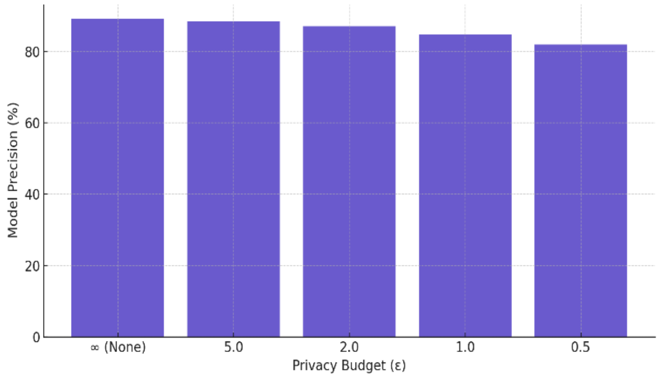

Figure 8: Differential Privacy Impact on Model Precision

C. *Fault Tolerance and Node Dropout Recovery*

In order to test the robustness of the federated framework, we modelled various degrees of node dropout to measure fault tolerance. For cases where one or more SAP modules did not attend a training round because of poor connectivity, maintenance, or overly restrictive data policies, the framework was still able to function well. During dropout, model accuracy declined for some of the rounds, but once participation was allowed again, accuracy increased rapidly.

The influence of aggregation on the correction of model inconsistencies and outlier is extremely important. This is shown in Figure 9, which compares prediction error distribution among different data batches before and after model aggregation. There were numerous batches in advance of aggregation that had error rates greater than six percent, with some even exceeding ten percent. Following aggregation, most batches consolidated around the zero to four percent error level, and there was a dramatic fall off in instances of error above that level.

This confirmed the reasoning behind the function of aggregation with the extra provisions of anomaly filtering and weighted updates. These features were able to stabilize the model even when clients contributed noisy or incomplete updates and portions of the model summary. Not only did aggregation enable these models to be less biased, to increase global fairness the outlying clients were closer to the consensus model without reducing accuracy by throttling other modules.

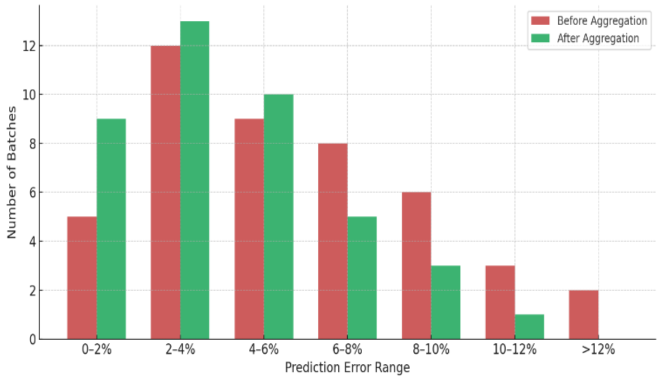

Figure 9: Error Distribution Before and After Aggregation

D. *Communication Cost and Training Time Efficiency*

In the federated learning system, communication is the primary issue. It is very difficult to optimize learning due to the very large amounts of data. In this system, both synchronous communication and asynchronous communication modes were implemented. There was an improvement with asynchronous communication, where methodologies used prior were outperformed by 38%, while preserving accuracy and stability of the model. The number of completed federated rounds was done in less time, using less bandwidth. All of this made asynchronous learning better for large scale deployments.

The efficiency of the training was evaluated from both networking and processing angles. Depending on the module and model's intricacy, the local model training times ranged from 2.8 to 4.3 seconds. Synchronous and asynchronous modes took 3.1 and 2.4 seconds respectively to complete aggregation, which includes privacy processing and anomaly detection. Hence, the time taken in transmission, training, and aggregation for a federated round was averaged at 7 seconds.

The system that was federated was able to complete the tasks with almost as much accuracy as with the centralized models while maintaining privacy under several threat models, recovering from client side faults, and functioning within the business's limits on resource utilization and speed. This shifted perspective proposes that rather than being an abstract concept, federated learning can be considered a viable, adaptable solution for the centralized-data AI problems within SAP systems.

VI. Discussion and Real-World Implications

A. *Enterprise-Wide Trust in Federated SAP Architectures*

Organizational trust is a key prerequisite for enabling federated learning across large heterogeneous environments which has little or no reliance on infrastructure. As is the case for systems like SAP, enterprise ones operate within layers of boundaries defined by authorizations, legislation, and silos of data meant to safeguard operational autonomy. The level of trust available to a shared machine learning system highly depends on the transparency regarding the distribution of responsibilities along with the decision-making processes and data handling.

This study proposes a novel approach for enabling collaborative intelligence across SAP modules without the need for centralized data. Each module retains self sovereignty over its data by only providing model updates in encrypted form. This paradigm enables trust to be built at the architectural level where data ownership, integrity, and transparency are guaranteed. In addition, the adapted explainable metrics, such as locally interpretable model and contribution score, can improve the acceptance rate and close the gap between AI systems and business people.

In terms of implementation, the use of FL within SAP environments will also require establishing some institutional agreements regarding the policy boundaries—who controls the global model, how updates are reconciled, and how sharing of sensitive information is enforced. Therefore, trust is established not only through encryption and privacy-preserving algorithms, but also through governance frameworks that specify roles and rules bespoken to the federated architecture.

B. *Security, Governance, and Compliance in Federated Models*

Federated learning presents new challenges to existing data governance policies because model training and intelligence is achieved in a decentralized fashion. In the case of SAP systems, each process is subjected to great regulatory scrutiny. Ensuring GDPR, SOX, HIPAA, and even ISO regulations means that every enterprise process, especially those within the financial domain such as Finance (FI), Procurement (MM), and Customer services (SD), are expected to be fully traceable, accountable, and auditable.

Federated learning in SAP has security requirements at three levels: transport layer, model update integrity, and aggregated model governance. Transport layer relates to the security offered by the communication channel such as use of multi-factor authentication, communication encryption protocols (TLS version 1.3), and use of SAP Cloud Identity Services. Update integrity is ensured through use of digital signatures, client certificate, and poisoned update filtering. Model governance is done by enforcing policies on update frequency, global weight retention, and all model state change access logging.

Besides security, risk management and compliance verification is no less important. SAP's federated deployments must demonstrate that there was no exchange of raw transactional data, that privacy budgets were utilized on a per-client basis, and that the aggregated models were not biased towards module super-aggregation or module authority. These conditions are achieved using audit trails within the federated server, and differential privacy guarantees that are checked statistically after each round.

Figure 10 illustrates a heatmap of inter-module risk zones in regard to their data sensitivity and exposure levels. FI, MM, and SD form the most sensitive zones for data exchange due to their extensive participation in financial and externally controlled transactional flows. Any federation system fusing these modules has to elevate their risk profiles.

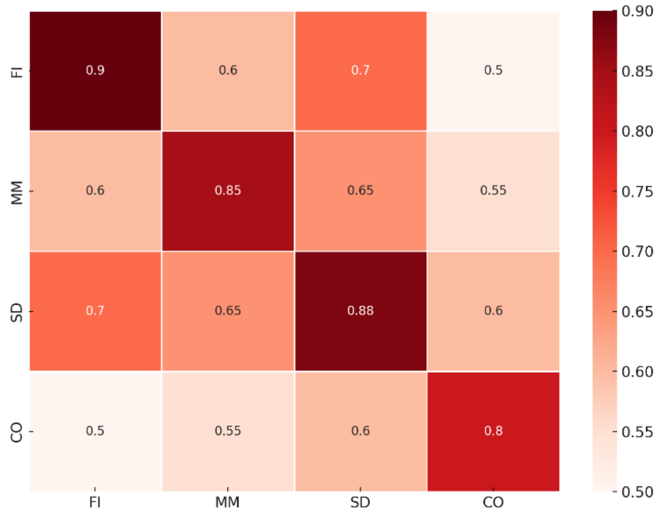

Figure 10: Risk Zones Based on Inter-Module Data Sensitivity vs Exposure

This is necessary for planning where more attention should be paid on compliance issues, like putting additional restricting measures using different privacy thresholds or implementing homomorphic encryption.

C. Operationalizing FL in SAP BTP and S/4HANA Cloud

In practice, federated learning cannot be used without proper integration to SAP's cloud services, thus it works in SAP Business Technology Platform (BTP) and S/4 HANA type cloud environments. The proposed system was infrastructure designed to serve SAP Business Technology Platform (BTP) and contemporary SAP S/4HANA cloud-native infrastructure. Each federated client was implemented as a containerized microservice deployed in a BTP subaccount that is securely connected to SAP backend through Cloud Connector and OData API.

In order to implement federated learning in this setup, SAP's IaaS approach has to be followed, and the native automation provided by SAP Kyma or Cloud Foundry has to be applied. Each client must be lightweight, maintainable, and capable of independent scaling, pausing, and reconfiguration to meet changing business requirements. These handle placement of FL nodes over different regions or business units matching to the structure of global SAP deployments.

The microservices architecture also allows the FL server component to be implemented as a reusable service on the SAP Integration Suite, thus allowing integration without dependency on composite middleware. Model updates can be processed using SAP Event Mesh, which improves responsiveness and makes asynchronous federation possible; this is particularly beneficial for systems that work with batch processing or have asynchronous data pipelines.

Such models can also be incorporated into Fiori dashboards or SAP Analytics Cloud, which enables advanced users to access the model results effortlessly. For example, users from the purchasing department may find it easy to see an allocated risk score or an anomaly alert within the MM interface, as the logic is already provided, and no data science skills are necessary.

D. Cost-Benefit Analysis for Large-Scale Deployment

Although federated learning has the potential to revolutionize privacy-preserving AI, organizations are rightfully concerned with the overhead that comes with forcibly implementing such a system. Like every other novel technology, implementing FL comes with upfront costs that include, but are not limited to, architecture, development, and training. Nonetheless, efficiency, security and insights in the long-term must not be ignored.

In this research, it was shown that federated model does not require nearly as much data transfer to be done, thus decreasing the reliance on expensive data lakes or third party ETL pipelines. This by itself causes massive savings for global enterprises during data transfer where bandwidth becomes a limiting factor. FL reduces the burden on central data teams as well by enabling module level teams to perform data engineering without having to worry about degrading the model performance.

In terms of security, FL does cutout the most expensive and vulnerable part of enterprise AI which is the centralized storage of the data - making it a target to breaches. Because no crude data is transferred outside the client module, the risk surface mitigated greatly reduces the insurance cost, as well as regulatory fines for exposing the systems. For industries that are heavily regulated, this reasoning on its own fulfils the justification for spending the costs on deployment.

Also, the accuracy metrics from the configuration confirm that the federated models achieved nearly 90% of centralized accuracy while spending 72% less inter-module data movement. Even while implementing privacy, the system was able to maintain over 80% model precision with heightened protection levels. These results suggest that not only is federated learning possible from a technical standpoint, but it also makes economic sense when implemented throughout extensive SAP environments.

In Closing, FL promotes modular growth. Additional SAP modules or outside systems can be incorporated into the federation without having to rework the whole federated model. This ability to integrate components demonstrates that FL is undeniably more robust to change, fulfilling the dynamic requirement of modern enterprise application software ecosystems.

VII. Conclusion and Future Work

A. Summary of Technical Contributions

This research provides a practically validated architecture for federated learning in SAP systems which allows FI, MM, SD, and CO modules to collaborate without compromising data security. Unlike distributed systems, the proposed model has a modular design so that each component builds the models locally and only encrypted model updates are sent to the cloud for aggregation. In addition, the system guarantees security and fidelity by employing differential privacy, secure multiparty computation, and modular model data obfuscation. The federated system showed competitive accuracy and strong fault tolerance, and operated efficiently in the confined environment of enterprise data. This confirms that decentralized AI is achievable in an SAP environment, therefore, scalable AI is possible within the SAP ecosystem.

B. Strategic Relevance for SAP-Centric Enterprises

The suggested federated learning framework works efficiently with the fundamental digital transformation strategies pertaining to modularization, data privacy, and real-time intelligence integration of an SAP-centric enterprise. It allows cross-module interactions without the risk of leaking sensitive information, which is ideal for compliance-centric environments and decentralized organizations. This enables a scalable paradigm of intelligent automation, where artificial intelligence is available in multiple silos, but all controlled by a common learning objective. This allows decision-makers to utilize collective intelligence from finance, procurement, sales, and operations departments with reduced risk, lower costs, and ever-increasing SAP compliance and agility requirements.

C. Future Work: Cross-Organizational and Multi-Tenant FL

In the future, research will focus on how this federated architecture can integrate with external systems beyond SAP modules for cross-organizational wisdom among partners, subsidiaries, or supply chain participants. Multi-tenant federated learning may enable competitive or cooperative organizations to train global models, like fraud or supplier risk detection systems, without exposing the underlying transactional data. Such developments will need improvements in inter-cloud orchestration, federated identity managing, and auditability in a corporate context. Also, adaptive aggregation and integration with tools for explainable analytics like SAP Fiori will increase the actionability of the collaborative systems, moving towards decentralized enterprise AI ecosystems.

References

[1] Rahman, Md Asfaquar, et al. "Opportunities and challenges in data analysis using sap: A review of erp software performance." International Journal of Management Information Systems and Data Science, volume1 (2024).

[2] Monk, Ellen F., and Bret J. Wagner. Concepts in enterprise resource planning. Course Technology, Cengage Learning, 2013.

[3] Kshetri, Nir. "The emerging role of Big Data in key development issues: Opportunities, challenges, and concerns." Big Data & Society 1.2 (2014): 2053951714564227.

[4] Parimi, Surya Sairam. "Real-time Financial Anomaly Detection in SAP ERP Systems Using Ensemble Learning Surya Sai Ram Parimi." Available at SSRN 4934842 (2024).

[5] JAMPANI, SRIDHAR, et al. "Optimizing Cloud Migration for SAP-based Systems." (2021).

[6] Khatri, Dignesh Kumar, and A. Renuka. "Optimizing SAP FICO Integration with Cross-Module Interfaces." SHODH SAGAR: International Journal for Research Publication and Seminar, 15 (1), 188. Link. 2024.

[7] Yang, Qiang, et al. "Federated machine learning: Concept and applications." ACM Transactions on Intelligent Systems and Technology (TIST) 10.2 (2019): 1-19.

[8] Kairouz, Peter, et al. "Advances and open problems in federated learning." Foundations and trends® in machine learning 14.1–2 (2021): 1-210.

[9] Mothukuri, Viraaji, et al. "A survey on security and privacy of federated learning." Future Generation Computer Systems 115 (2021): 619-640.

[10] Rahman, Md Asfaquar, et al. "Opportunities and challenges in data analysis using sap: A review of erp software performance." International Journal of Management Information Systems and Data Science, volume1 (2024).

[11] Saghar, Syed. "Benefits of System Integration Using SAP PI/PO." (2021).

[12] Yang, Qiang, et al. "Federated machine learning: Concept and applications." ACM Transactions on Intelligent Systems and Technology (TIST) 10.2 (2019): 1-19.

[13] Kairouz, Peter, et al. "Advances and open problems in federated learning." Foundations and trends® in machine learning 14.1–2 (2021): 1-210.

[14] Lu, Yunlong, et al. "Differentially private asynchronous federated learning for mobile edge computing in urban informatics." IEEE Transactions on Industrial Informatics 16.3 (2019): 2134-2143.

[15] Zhang, Xueyi, et al. "A cloud–edge collaboration based quality-related hierarchical fault detection framework for large-scale manufacturing processes." Expert Systems with Applications 256 (2024): 124909.

[16] Li, Tian, et al. "Federated optimization in heterogeneous networks." Proceedings of Machine learning and systems 2 (2020): 429-450.

[17] Wang, Hongyi, et al. "Federated learning with matched averaging." arXiv preprint arXiv:2002.06440 (2020).

[18] Zhao, Bo, Konda Reddy Mopuri, and Hakan Bilen. "idlg: Improved deep leakage from gradients." arXiv preprint arXiv:2001.02610 (2020).

[19] Acar, Abbas, et al. "A survey on homomorphic encryption schemes: Theory and implementation." ACM Computing Surveys (Csur) 51.4 (2018): 1-35.