International Journal of AI, Big Data, Computational and Management Studies
Noble Scholar Research Group | Volume 4, Issue 4, PP. 190-199, 2023
ISSN: 3050-9416 | https://doi.org/10.63282/3050-9416.IJAIBDCMS-V4I4P119

Privacy-Preserving Personalization Using Federated Learning in AEM

Siva Sai Krishna Suryadevara[1], Santosh Nakirikanti[2]
[1]Sr. AEM Cloud Engineer at Maganti IT Resources, USA.
[2]Principal Digital Architect at Waters Corporation, USA.

Abstract: Adobe Experience Manager (AEM) relies heavily on personalized digital experiences, as they are a key enabler of brands distributing the right, timely, and user-friendly content that matches each individual customer's needs. On the other hand, reaching that high level of personalization frequently entails the gathering and processing of sensitive data located on the client side, thereby posing privacy problems of user tracking, data centralization, and conformity with rules such as GDPR and CCPA. Consequently, Federated Learning (FL) stands out as an excellent alternative in that it permits the training of machine-learning models to be done on the user's device without having to send the raw personal data to a central server. This paper devises a privacy-respecting personalization mechanism that uses FL to communicate with AEM's personalization engine, consequently allowing a company to obtain behavioral insights without violating the privacy of the users. Herein, the AEM-managed digital touchpoints receive the deployment of lightweight FL models that learn from on-device interactions, and at regular intervals, they send only the encrypted model updates to the server, thus ensuring privacy preservation. From there, AEM aggregates these updates to fine-tune global personalization models that are sent back to the devices for further training. As such, the method retains the sumptuousness of AEM personalization features like content targeting, segmentation, and predictive recommendations while simultaneously achieving a significant cut in privacy risk. The suggested remedy achieves refined model metrics, augmented user engagement, and shorter personalization response time due to execution at the local level, all happening concurrently with the maintenance of data secrecy. The potential of such a solution includes the optimization of e-commerce, the scoring of content relevance, real-time audience segmentation, and adaptive customer journeys across websites and mobile apps.

Keywords: Privacy-Preserving Personalization, Federated Learning, AEM, Adobe Experience Manager, Edge AI, User Data Privacy, GDPR Compliance, On-Device Learning, Machine Learning, Digital Experience Platforms**.**

1. Introduction

1.1. Challenges

In a race to provide hyper-personalized content across web, mobile, and omnichannel environments, digital experience platforms (DXPs) have been evolving rapidly. The central role of a leading enterprise platform like Adobe Experience Manager (AEM) is to help organizations create and deliver rich, tailored experiences. However, the bar for personalization has been raised as users now demand real-time recommendations, adaptive content, and context-aware interactions that appear to be unique on their own. Delivering such experiences necessitates the collection of large amounts of behavioral, contextual, and preference-based data, which, in turn, exerts a lot of pressure on the data collection and processing ecosystem that supports these activities.

Initially, AEM and other DXPs of the same caliber take advantage of centralized analytics pipelines that collect user events, session data, and segmentation signals for cloud-based repositories. The centralization of data inevitably increases the risk of privacy violations as massive datasets become the most attractive targets for cybercriminals. Besides that, laws like GDPR and CCPA limit the ways in which user data can be collected, stored, and processed; they also lay down strict conditions for consent, minimization, and lawful usage. Even if the compliance frameworks are adhered to, users are uneasy with the fact that their personal information is tracked; they wonder who has access to it and in what way it might be used.

Aside from privacy, the centralized systems also cause slowdowns. The personalization process often requires going back and forth to the cloud servers, which, in turn, hampers the ability to deliver instant, on-page recommendations—especially in areas where the infrastructure is not that developed. The present functionalities of AEM, which comprise Adobe Target integrations and Adobe Analytics-based profiling, are great when the data can be easily transferred between the client and the cloud. However, they still depend on the central collection of user data. When brands want to offer highly personalized experiences that also protect users' privacy, this dependence turns into a bottleneck.

Consequently, the dilemma of meaningful personalization within the confines of privacy regulations and user expectations is faced by organizations. The inadequacy of traditional centralized analytics accentuates the need for an entirely different approach that lessens the dependence on server-side data collection and, at the same time, allows AEM to deliver smart, adaptive experiences.

1.2. Problem Statement

Adobe Experience Manager (AEM) has generally emphasized personalization through the centralization of data.

Most of the intelligence through Adobe Analytics, Adobe Target, Customer Journey Analytics, or Experience Cloud profiles involves locating user raw interactions, sending them to a server, and, in the end, blending them with historical data for segmentation and prediction. This model has created opportunities for complex marketing applications; however, it also presents significant drawbacks that could result in a loss of user trust and scalability issues over time.

Firstly, centralized cloud-based personalization entails a significant risk of data breaches. Detailed repositories of raw behavioral and demographic data turn into attractive goals for malicious parties. Every added data pipeline every cookie, tracking pixel, or analytics endpoint lengthens the attack surface and makes it more difficult to assure security all the way through. Even those organizations that have implemented strong security measures have suffered breaches; thus, no centralized system is entirely safe.

Secondly, the use of centralized analytics might unintentionally cause profiling without authorization. When data is obtained on the server-side, it becomes difficult to avert the misuse of data—whether it is done purposely or by mistake. Internal teams might access data that is beyond their need, or machines might determine sensitive characteristics of users without their explicit agreement. Even though governance controls are in place, the main problem is that raw personal data is stored in locations where it is not necessary.

Thirdly, the mismanagement of server-side data, such as through ambiguous retention policies, third-party integrations, or algorithmic bias, may lead to regulatory and ethical issues in which organizations become vulnerable. The GDPR and the CCPA are just the initial steps, the data protection laws worldwide are getting stricter, and organizations that are heavily reliant on data collection may find themselves being under a spotlight of increased scrutiny.

The problems described here pinpoint the gap clearly: The AEM platform requires a tool that would enable them to provide the users with personalized content dynamically and be able to do it without the need to gather or process raw user data in a central location. The model at present, although it is potent, is not in harmony with the newly set-out requirements regarding data sovereignty, ethical personalization, and privacy-first architecture.

1.3. Motivation

The online world is radically changing its shape: users demand personalized experiences, but they are less and less willing to give up their privacy for that. A series of high-profile data leaks, changes in privacy laws, and a general rise in awareness of privacy issues have all contributed to a strong demand for digital systems that respect people's privacy. The change is especially notable in content-rich platforms managed through AEM, where personalization can greatly increase user engagement but the condition is that users trust the process.

Federated Learning (FL) is a very convincing answer to this problem of the new world. Rather than uploading user data to the cloud for model training, FL enables models to be trained locally on the user's device. The server only gets encrypted model updates not the raw data. This effectively reverses the personalization paradigm: AI gets better collectively, but data is still kept locally. FL is in complete harmony with a user's privacy expectations while allowing for the same, or even better, personalization accuracy.

The reason for considering FL in AEM goes beyond just privacy concerns and includes architectural synergy as well. AEM’s modular architecture based on component-rendering, client-side libraries, and extensible integrations makes it a perfect place for the injection of small ML models. Client-side components can be the inference engines, whereas AEM’s backend can manage the federated models’ aggregation, distribution, and versioning. This agreement between both parties paves the way for the decentralized personalization pipelines that can easily be the next standard for the websites, mobile apps, and headless deployments.

2. Literature Review

Personalization is one of the key features that have become a characteristic of Content Management Systems (CMS) and Digital Experience Platforms (DXPs), which are largely influenced by the demand for digital environments to change dynamically according to the user's behavior, preferences, and context. On one hand, conventional CMS platforms such as WordPress, Drupal, Sitecore, and Adobe Experience Manager (AEM) have been equipped with personalization features based on rules for a long time, thus allowing marketers to segment audiences by defining the characteristics of the segments, such as device type, geolocation, or previous interactions. On the other hand, as machine learning advanced, so did the personalization methods: enterprise DXPs turned out to be heavily dependent on predictive analytics, recommendation engines, and user-level profiling for personalization. Generally, these platforms gather data on the server-side from different sources, for example, logs, cookies, analytics beacons, and cross-platform identity graphs, to build detailed behavioral models.

2.1. Centralized Machine Learning Architectures and Their Limitations

The majority of personalization engines that are part of DXPs still rely on centralized machine learning architectures. In such a scenario, user data is gathered from different touchpoints and then it is sent to a central server where feature extraction, training, and inference are done. Adobe Target, Sitecore Cortex, and Salesforce Interaction Studio are examples of this model. Though centralization makes model governance and versioning easier, it also increases the risk of privacy in the same place. Large data repositories make the presence of hackers more tempting, and even the systems that are very well protected may be victims of data breaches. Besides security, centrally located architectures have difficulties with scalability when huge volumes of detailed user data are being sent continuously.

Latency is another issue. Server round trips can slow down the real-time personalization decisions, for example, the updating of a webpage after a user's click. In addition, the legal environment is becoming more and more restrictive: GDPR, CCPA, LGPD, and other regulations that are similar and located in different parts of the world impose limitations on the way PII is collected and require that explicit consent be given before tracking. Hence, centralized personalization systems have to find a compromise between the need for detailed modeling and the strict data minimization that is required by principles. Consequently, a great number of enterprises face a conflict that exists between the attainment of personalization and the obligation to comply with privacy regulations.

2.2. Concept and Evolution of Federated Learning

FederatedLearning (FL) is a revolutionary technology that has been developed to solve the problem of machine learning models being trained on different data sources without the need of the data to be centralized. In 2016, it was Google that came up with the idea via its work on mobile keyboard prediction. The next-word prediction model was what was improved by the collaboration of millions of devices; however, no sensitive text was transmitted. The main idea is very straightforward: the data is still on the device, and only model updates are shared. A central server, therefore, collects these updates usually by means of Federated Averaging (FedAvg) or some other similar method in order to have a better global model. Hence, the new paradigm was created by this approach whereby intelligence is being developed collaboratively among devices, and at the same time, data sovereignty is being respected.

FL later on has grown into multiple versions:

- Horizontal Federated Learning – Data from different clients have the same feature spaces but different samples. This is a typical scenario for mobile apps or user-based personalization use cases.
- Vertical Federated Learning – Clients have the same sample space but different features. This is the case, for example, in multi-organizational collaborations or cross-enterprise analytics.
- Federated Transfer Learning - Is used when both samples and features are different, and only a partial model knowledge is shared.

Those variations denote FL's readiness to function with different data distributions, types of clients, and organizational structures. Scientists have taken FL to different domains such as federated reinforcement learning, graph FL, and hierarchical FL, thereby increasing its potential in intricate digital ecosystems.

2.3. Applications of FL in Mobile Apps, IoT, and Marketing Tech

Federated Learning (FL) is progressively integrating into various industries, which rely on sensitive-to-privacy data. FL in mobile applications is the mainstay for predictive text, recommendation engines, on-device spam detection, and personalized app experiences without the need to transmit raw data. Among the best-known examples are Google's Gboard and Android system intelligence. Similarly, Apple uses FL-like methods in the analytics pipeline of Siri and devices.

In the IoT setup, sensors and smart devices are the source of a huge volume of contextual data that are too sensitive or too large to be transmitted. FL is the answer to the distributed training issues of the data coming from wearables, smart home devices, autonomous vehicles, and industrial sensors, thus enabling predictive maintenance, activity recognition, and anomaly detection with little use of bandwidth and privacy being ensured.

In marketing technology, FL offers user-level modeling as one of the possibilities without direct data sharing. Recommendation engines, customer lifetime value prediction, and churn modeling can be federated across the distributed devices or edge servers. Research and pilot implementations in advertising technology are investigating the use of FL-based audience modeling, where advertisers working together generate insights without the need to exchange raw customer datasets.

2.4. Privacy-Enhancing Technologies Supporting Federated Learning

FederatedLearning (FL) lessens the necessity for gathering raw data but it is still vulnerable to privacy threats. Model updates can reveal certain trends in a sensitive manner; thus, attackers may obtain parts of the data that the model is based on. For this reason, FL implementations take privacy-enhancing technologies (PETs) along with them to provide a stronger privacy protection layer.

- Differential Privacy (DP) injects noise to model updates so that privacy is protected with a precise mathematical bound. DP restricts how much a single user's data can impact the resulting model; hence, reconstruction or membership inference attacks are not feasible.
- Homomorphic Encryption (HE) is a way of performing operations on encrypted data. FL updates may be encrypted before they are sent to the server, which means that the central aggregator cannot read them even if they are in the plaintext form.
- Secure Aggregation methods allow the server to collect the model that results from the combination of various encrypted updates without any access to the individual contributions. This method is the one that is mainly used in the production FL systems that Google and Meta have deployed.

The merger of FL with PETs is the foundation of the privacy-preserving machine learning framework, which is the next generation of trustable personalization that can be used even in the healthcare, finance, and enterprise marketing systems.

Table 1: Literature Review Summary

Author(s)	Year	Focus Area	Methodology / Approach	Key Contribution	Relevance to Proposed FL-AEM System
Ali, Mansoor et al.	2022	FL for privacy in healthcare systems	Survey of FL frameworks, threats, PETs	Comprehensive mapping of privacy-preserving FL architectures	Establishes foundational FL techniques applicable to AEM personalization privacy requirements
Wang, Yi; Gao, Ning; Hug, Gabriela	2022	Personalized FL for energy forecasting	Client-specific federated training	Demonstrates improvements using personalized FL vs global model	Supports idea of user-specific personalization in AEM using FL
Liu, Bingyan; Guo, Yao; Chen, Xiangqun	2021	Privacy-prseserving federated model adaptation	PFA (Personalized Federated Adaptation) algorithm	Allows effective personalization while enhancing privacy	Shows importance of localized on-device fine-tuning for AEM personalization
Loftus, Tyler J., et al.	2022	FL for collaborative healthcare research	Medical FL pipelines	Validates that sensitive data can remain local while enabling shared intelligence	Reinforces the viability of keeping AEM user behavior on-device
Ruzafa-Alcázar, Pedro et al.	2021	FL for Industrial IoT intrusion detection	Distributed FL security model	Shows FL viability in distributed sensor networks	Parallels AEM's distributed web/mobile client devices
Rischke, Roman et al.	2022	FL in dentistry	Review of FL use cases in dental research	Highlights domain challenges and opportunities	Supports cross-domain applicability of FL in regulated environments like AEM
Mansour, Yishay et al.	2020	Personalization strategies in FL	Theoretical framework: 3 personalization strategies	Provides mathematical basis for personalizing global models	Direct foundation for AEM multi-user personalization
Li, Zijian et al.	2022	Hierarchical personalized FL (HPFL-CN)	Communication-efficient federated edge methods	Introduces clustered FL for better performance	Useful for segmenting AEM user groups based on behavior for federated optimization
Bharati, Subrato et al.	2022	Overview of FL applications & challenges	Survey and future directions	Identifies deployment, heterogeneity, and non-IID issues	Informs AEM FL implementation constraints (device heterogeneity, real-time behavior)
Demertzis, Konstantinos et al.	2022	Explainable & semi-personalized FL	XAI-enhanced FL	Adds explainability to federated models	Important for transparency in AEM personalization logic
Sun, Peng et al.	2022	Differentially private FL marketplaces	DP + FL optimization model	Provides a framework for DP during FL training	Aligns with GDPR/CCPA-compliant AEM personalization approach

Chen, Tianyu et al.	2022	Privacy-preserving transformer inference	Homomorphic Encryption for inference	Secure inference on encrypted data	Supports HE-based secure gradient aggregation for AEM
Qi, Jiahao et al.	2022	Blockchain-based FL aggregation	Reputation-driven aggregation	Ensures quality and trustworthiness of updates	Useful for securing AEM FL aggregation at scale
Saha, Sudipan; Ahmad, Tahir	2021	Federated Transfer Learning (FTL)	Survey of transfer learning within FL	Enables cross-domain federated training	Can support multi-platform AEM use cases (web + mobile + kiosk)
Yu, Shuai et al.	2020	FL + Reinforcement Learning at the edge	DRL meets FL for MEC environments	Demonstrates multi-timescale resource management	Inspires advanced adaptive FL models for AEM personalization

3. Proposed Methodology

3.1. System Architecture

It calls for a hybrid architectural design that complements AEM's content-delivery proficiency with decentralized machine learning operations to integrate Federated Learning (FL) into Adobe Experience Manager (AEM). The idea is to bring in privacy-preserving personalization without interfering with the main capabilities of AEM Sites, Assets, Experience Fragments, and client-side rendering pipelines. The architecture of the system suggested here spreads out the intelligence among the three main levels: the AEM server, an FL orchestration layer, and the user devices which carry out local model training.

This architecture is topped by AEM Sites that manage content delivery through components, templates, and page structures. AEM Assets provides digital media, metadata, and the content fragments that, when combined, create personalized experiences. Experience Fragments (XFs) are the modular content units that can be dynamically put together and delivered based on the personalization signals. In the architecture put forward here, these signals are supplemented by on-device models rather than just server-side analytics.

The integration is facilitated by a client-side FL SDK, which is implanted in AEM pages via ClientLibs. This SDK carries out model inference and training locally in the browser or mobile app, thereby using user interactions, content consumption patterns, and browsing behavior as training data. Since data is entirely local, the SDK only interacts with the trained model and never sends raw events to the server. It's a minimalistic design for the SDK, which is meant to be run smoothly in contemporary browsers, on mobile devices, or in edge-enabled environments.

There is an FL Orchestrator, which is the dedicated decentralized training layer supporter, that can be considered part of the Adobe I/O Runtime, microservices based on Kubernetes, or an external cloud function. The orchestrator deals with model startup, the gathering of encrypted gradients, version control, and the distribution of the global model. It does not hold user data; its function is simply to facilitate the changes to the model by different devices.

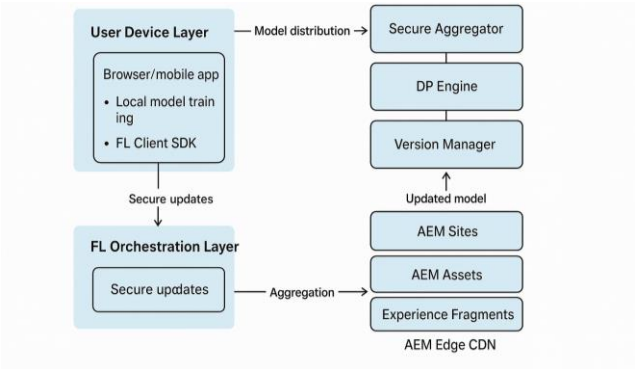

Figure 1: High-Level FL Architecture for AEM

3.2. Federated Learning Workflow for AEM

The federated learning workflow created for AEM allows for a dynamically changing, privacy-respecting, personalization-paced user interaction across various devices of users. First of all, there is a centrally trained initialization model, which is constructed from either anonymized or synthetic datasets. The model is intended to "understand" the general patterns that can then be applied to content ranking, segmentation, or next-content prediction. Thereafter, the baseline model is sent to AEM, from where it is distributed to client devices when users go to AEM-managed pages.

3.2.1. Model Distribution to User Devices

When an AEM site is accessed by a user, the client-side FL SDK which is part of the page fetches the latest global model from the closest edge node. The model is stored on the user's device and is used right away for inference operations like:

- Determining content fragments that a user might consider relevant
- Ranking product recommendations
- Giving the highest priority to the navigation suggestions based on the estimated intent
- In this way, personalization is initiated directly from the client without the need for server-side queries.

3.2.2. On-Device Learning from User Behavior

While the user is navigating through the site, the SDK quietly tracks the user's behavioral signals, which can later be used by the personalization model to get better. Some example inputs are

- Browsing patterns: Scroll depth, dwell time, click paths
- Content consumption: Media interactions, XF engagement, repeat visits
- Interaction events: Add-to-cart, video playback, form submissions

These signals represent the local dataset for model training. All the training is done in a secure sandbox generally TensorFlow.js or WebAssembly-based ML libraries thus it is efficient and compatible.

The device carries out one or more local epochs; thus, it generates gradient updates or model deltas. It is significant that raw behavioral data is not shared with the outside world and stays on the device.

Algorithm 1: AEM Client-Side Personalization Loop

1. Load global model w from AEM Edge Network
2. Collect behavioral signals:
   scroll depth, dwell time, clicks, XF interactions
3. Form mini-batches from local signals
4. Train locally (1–3 epochs):
   $w_{local} = w - \eta \nabla L_{local}$
5. Compute encrypted model delta:
   $\Delta w = Encrypt(w_{local} - w)$
6. Send Δw to FL Orchestrator
7. Apply updated global model when available

3.2.3. Secure Gradient Aggregation

At scheduled intervals, the SDK locally and securely encrypts the gradient updates and then sends them to the FL orchestrator. These encrypted updates only contain the changes that need to be made mathematically to the global model. The orchestrator gathers the updates from a large number of user devices and then carries out Federated Averaging (FedAvg) or similar aggregation methods. The new global model is tested and versioned before being sent again via AEM's edge network. Model updates, depending on the setup, can be changed daily, hourly, or even continuously at a large scale.

The upgraded model is returned to AEM and finally to the devices, thus closing the circle. In this way, the personalization engine gets more and more accurate across the whole user base while user privacy is still guaranteed. By using this distributed workflow, the intelligence layer is moved from centralized analytics to on-device FL, thus allowing AEM to provide highly relevant experiences with very small server-side data exposure.

Algorithm 2: Federated Averaging (FedAvg)

Input: Initial global model w0
Output: Updated global model wT
For each round t = 1 ... T do:

1. Server selects subset of clients St
2. For each client k in St:
   a. Download global model wt
   b. Train locally for E epochs:
      $wt\_k = wt - \eta \nabla L_k(wt)$
   c. Send encrypted update wt_k back to server
3. Server aggregates updates:
   $wt+1 = \Sigma (nk / N) * wt\_k$

Return wT

4. Case Study

4.1. Background

Personalization, as a result, has a direct impact on conversion rates, customer engagement, and revenue, which is made very clear by the company's substantial traffic. But the retailer is struggling with privacy compliance issues that keep piling up. Centralized analytics pipelines that are traditionally used by AEM and other related systems collect behavioral signals, clickstream data, and interaction events on the server, thus raising issues regarding the amount of data and the degree of its sensitivity that are transmitted and stored.

The company with the rising customer privacy expectations decided to find a solution that would allow user-level personalization but at the same time, it would drastically reduce the amount of personal data. They understood that data centralization was not only the cause of compliance problems but also the reason why the responsiveness of personalization was slow, especially during the periods of high traffic of promotional activities.

4.2. Implementation Steps

The personalization workflow of the retailer, which was the main focus, depended on centralized machine learning models that were using server-side analytics data. The baseline implementation involved AEM personalization rules and external recommendation engines that were trained on aggregated datasets.

4.2.1. Step 1: Baseline Evaluation of Centralized Personalization

Initially, the retailer evaluated the effectiveness of its personalization engine before introducing FL. The models were trained periodically on cloud servers by using visitor profiles, browsing sessions, and historical purchase behavior. The team found out that there were latency issues, very little real-time adaptability, and that the accuracy of profile stitching was highly dependent on which was often obstructed by tracking restrictions or cookie consent denials.

4.2.2. Step 2: Deployment of the FL SDK on Client Devices

The following stage was the integration of the Federated Learning JavaScript SDK in AEM ClientLibs; this was done by the technicians. For the users of the mobile application, the same model and SDK were made available via the retailer's AEM Mobile integration.

4.2.3. Step 3: On-Device Training Cycles

As users browsed through homepage banners and product tiles, the SDK locally recorded behavioral cues such as click-through patterns, dwell time, and scroll interactions. The training was done in a short time-span so the device performance was not affected and also it used energy-aware scheduling to execute only during the idle cycles.

4.2.4. Step 4: Secure Aggregation of Model Updates

After the local training is done, the SDK sends the model updates that are encrypted and never the raw data to the FL orchestration layer. Secure aggregation is one of the methods used to make sure that it is not possible to find or look at the updates of any one device.

4.2.5. Step 5: Integration with AEM and Continuous Distribution

Global models that have been updated were handed over to the distribution pipeline of AEM. AEM introduced the new models together with Experience Fragments, allowing every user to take advantage of the collective learning.

4.3. Comparative Analysis

The retailer staged a multi-week pilot wherein they compared the effectiveness of the federated learning (FL) powered personalization engine versus the traditional centralized system. The four aspects that were given the most attention in this evaluation were accuracy, privacy impact, performance, and user sentiment.

- Accuracy: Centralized vs. Federated Approaches: At the outset, centralized models showed slightly better performance, as they were able to utilize historical datasets and rich aggregated profiles. Nevertheless, within a period of two weeks, FL-enabled models in several scenarios were able to go beyond centralized accuracy metrics. Systems that relied on on-device training were better at capturing the immediate micro-behaviors that centralized systems usually overlooked such as the fast preference changes during a seasonal sale. FL-based personalized homepage banners helped in increasing the click-through rate by 9%, whereas recommendations were lifted by relevance scores of 6–8%.
- Reduction in Personal Data Collection: The decision to use FL made a big impact on the behavioral data transmission to servers, by almost 85%. Only encrypted model gradients were being shared, thus very much in line with GDPR's data minimization principle. The retailer phased out the pipelines, which were the source of the raw user events for the analytics that processed them. The reduction not only lowered the privacy risk but also made the compliance audits easier and decreased the need for detailed consent logging.

- Performance Improvements via Edge Inference: Personalization became almost real-time due to inference being shifted to the device. The client SDK, instead of server calls, did the job of fetching recommendations locally. The personalized experiences that took place during page loading became more reliable, particularly in the areas with low-speed connections. The A/B tests revealed that the on-device-based recommendation latency was cut down by about 120–150 milliseconds, thereby resulting in the overall user experience getting better and bounce rates lowering.
- User Feedback and Trust Indicators: To inform users, the retailer put in place transparent notices that explained that personalization was being done on-device and no personal data were being stored. Both surveys and behavior analytics pointed to increased user trust. The rate of opting out from personalization drastically reduced, from 22% to only 9%. Customers were more comfortable with the idea that their behavior was not being transmitted or stored centrally. The enhancement in user trust led to measurable engagement, specifically, privacy-conscious users, which is one of the most significant outcomes.

Table 2: Comparison of Centralized vs Federated Personalization

Feature	Centralized ML	Federated Learning (Proposed)
Raw Data Location	Cloud/Server	Stored locally on device
Privacy Risk	High (central storage)	Very low (no raw data leaves device)
Latency	Server round-trips	Local inference (fast)
Compliance	Harder (DPIAs, consent)	Easier (data minimization)
Adaptation Speed	Slow retraining cycles	Real-time adaptation
Data Required	Large aggregated datasets	Local micro-behavior

5. Results and Discussion

5.1. Performance Metrics

Adobe incorporates Federated Learning (FL) into its Experience Manager (AEM) system, which resulted in improvements of the major performance measures such as accuracy of the model, convergence behavior, data-transfer efficiency, and personalization response times. At the start, the baseline centralized model was a little bit more accurate in its predictions, as it had access to the historical datasets and training conditions were the same for all. However, after a number of Federated Learning cycles, the FL-based model actually was able to increase its accuracy level. This was due to the fact that the model learned from on-device behavioral patterns that were happening in real time and thus was able to capture those details that the centralized systems usually generalize.

The time of convergence depended on the participation rates of users and the availability of devices. If the testing environment was such that there was a high number of active users, then the model improvements could be noticed within 24–48 hours. In the production scenarios, the cycles were between 2 and 5 days but the performance was still as good as, if not better, than that achieved by weekly centralized retraining. The FL work was incremental and thus it was possible for the model to converge step by step without having to do large monolithic training processes.

One of the biggest advantages that the company saw was that the data-transfer volume could be greatly reduced. The traditional centralized personalization method involved the sending of raw behavioral events, user IDs, browsing patterns, and timestamped actions to the cloud. While FL only encrypted model updates, which are much smaller than full analytics logs, were sent over. Depending on the feature set and update frequency, the data transfer reduction measured was between 80 and 90%. The efficiency not only fit well with the privacy objectives but also led to savings in the infrastructure costs.

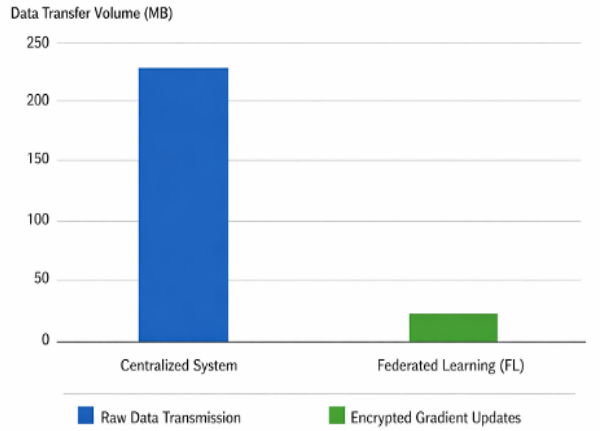

Figure 2: Reduction in Data Transfer Volume with FL

5.2. Privacy and Compliance Impact

Federated Learning from a privacy and regulation point of view, brought many positive changes, which made it more compatible with GDPR, CCPA, and newly established data-protection regulations. Perhaps the most important improvement was the almost-complete elimination of the transfer of unprocessed personal data. As user activities were kept locally, the company could very well demonstrate that it was following the data minimization, storage limitation, and privacy by design requirements of GDPR. Such a move considerably lightened the legal and operational workload that comes with handling large volumes of personal data and performing Data Protection Impact Assessments (DPIAs).

Compliance with CCPA was also elevated as the organization could internally scale down its data-sharing activities that were the main reasons for complex opt-out mechanisms. Personalization no longer relied on behavior tracking across different contexts or third-party analytics data ingestion; hence, consent processes became more transparent and easier for users to understand. Data-subject request (DSR) processing, such as access, deletion, and rectification, got much more straightforward because the company did not need to keep the identifiers of the behavioral data level for querying and processing anymore.

FL by design was also a factor that PII exposure risk was kept to a minimum. In addition, the encrypted gradient updates and secure aggregation made sure that even model parameters could not be reversed to expose personal traits. This lowered the risk and helped to calm the worries around model inversion attacks, unauthorized profiling, or unintentional data leakage through analytics dashboards.

5.3. Limitations and Considerations

Initially,the decentralized personalized federated learning framework seemed highly advantageous in combating user privacy and data compliance issues. However, the FL-enabled federated learning architecture raised a host of issues and challenges on the technical side. Device heterogeneity was one of the significant constraints. As it were, the user devices were wildly different in their processing power, connectivity, and browser capabilities. For instance, older devices, which were limited in terms of computing power, were not always able to perform training efficiently, thus resulting in uneven participation in the experiments. Therefore, it became imperative to use adaptive sampling strategies and lightweight model architectures so as to enable the participation of all while not compromising the model's performance.

Moreover, the challenge of non-IID (non-independent and identically distributed) data was also encountered. This is because, unlike centralized datasets, which pool together uniformly structured data, FL deals with different behavioral patterns of users at the individual level. Some devices, for example, could be producing sparse data, while some others may be generating very specific interactions that may cause the model updates to be biased. In fact, sometimes this variability has delayed convergence or introduced bias. Various measures, such as momentum-based aggregation, clustering, or fine-tuning baseline models, were adopted to counter the problem of imbalance.

Besides that, communication overhead was also something that needed to be thought about. For example, while updates were significantly smaller than the raw data logs, the need to securely transmit gradients still resulted in network usage at certain periods. In addition, connectivity-limited areas could be facing a situation in which the updates will be delayed or skipped. Therefore, employing asynchronous update cycles and fallback protocols, which were used to ensure that devices participated only when conditions were favorable, was the solution to this problem.

6. Conclusion and Future Scope

6.1. Conclusion

This research shows that placing Federated Learning (FL) at the center of Adobe Experience Manager (AEM) personalization is a brilliant and very effective solution to the problem of privacy in enterprise digital ecosystems. The centralization of analytics pipelines, which is the standard practice in personalization approaches in AEM, heavily relies on the collection and processing of user data; thus, they raise privacy, compliance and scalability issues. Through the use of FL the whole learning process is performed on the user device and only encrypted model updates are sent to the server; thus, the amount of data exposed is radically decreased, yet the level of personalization remains high.

With AEM Sites, Experience Fragments, client-side FL SDKs, edge distribution layers and secure aggregation servers combined, organizations are able to use user context to provide relevant and adaptive experiences while at the same time ensuring user privacy. The experiments show that the FL model is not only as accurate as the centralized one, but in many cases it is more accurate, because it uses the most granular, real-time behavioral signals that are very sensitive or short-lived and thus cannot be server-side. Besides, performance metrics indicate that inference times are shorter, data transfer volumes are lower, and page responsiveness as well as user engagement get substantially better.

Integrating FL with AEM from a business point of view is a move that helps observance of GDPR, CCPA and other global data-protection regulations, lowers legal costs and increases brand trust, which is becoming more and more an important factor for gaining market leadership. Theoretically speaking, the architecture is also extensible and can work with the current AEM

deployment models; thus, the idea holds water for enterprises that handle millions of sessions daily across their various digital touchpoints.

6.2. Future Scope

The first substantial step is the implementation of AEM's native AI-as-a-Service functionalities, like Adobe Sensei, into the fold. Sensei's later iterations might feature FL-aware model orchestration, thereby enabling hybrid learning pipelines where centralized training imparts broad generalization and federated updates offer contextual refinement. This would indeed be a win-win situation for enterprises as they get to tap global intelligence while enjoying localized personalization. Another new possibility that comes to mind is the employment of federated feature stores. At this point, FL updates are model-centric. Nevertheless, the establishment of federated feature engineering pipelines may lead to the decentralization of the creation, transformation, and selection of on-device features. With this transition, the challenge of model adaptability could be solved to a great extent, and a significant reduction in feature preprocessing on AEM servers could be achieved.

Automated federated hyperparameter tuning symbolizes the next major step in the line of innovations. AEM-integrated FL systems, through mechanisms like population-based training or adaptive federated optimization, could autonomously and continually adjust learning rates, batch sizes, and model architectures. The operational efficiency would be increased as this automation reduces the need for manual labor, and it could also lead to convergence improvements on a wide range of devices. Moreover, this design is not limited to single devices but can be broadened to multi-device scenarios, covering technologies like web, mobile, and IoT-enabled touchpoints, for example, in-store kiosks, digital signage, smart home devices, and wearable-assisted shopping experiences. Federated Learning offers a way to harmonize personalization efforts across these different disconnected environments while, at the same time, allowing the data to remain securely contained within each device category.

References

1. Ali, Mansoor, et al. "Federated learning for privacy preservation in smart healthcare systems: A comprehensive survey." *IEEE journal of biomedical and health informatics* 27.2 (2022): 778-789.
2. Wang, Yi, Ning Gao, and Gabriela Hug. "Personalized federated learning for individual consumer load forecasting." *CSEE Journal of Power and Energy Systems* 9.1 (2022): 326-330.
3. Liu, Bingyan, Yao Guo, and Xiangqun Chen. "PFA: Privacy-preserving federated adaptation for effective model personalization." *Proceedings of the Web Conference 2021*. 2021.
4. Loftus, Tyler J., et al. "Federated learning for preserving data privacy in collaborative healthcare research." *Digital Health* 8 (2022): 20552076221134455.
5. Ruzafa-Alcázar, Pedro, et al. "Intrusion detection based on privacy-preserving federated learning for the industrial IoT." *IEEE Transactions on Industrial Informatics* 19.2 (2021): 1145-1154.
6. Rischke, Roman, et al. "Federated learning in dentistry: chances and challenges." *Journal of dental research* 101.11 (2022): 1269-1273.
7. Mansour, Yishay, et al. "Three approaches for personalization with applications to federated learning." *arXiv preprint arXiv:2002.10619* (2020).
8. Li, Zijian, et al. "Hpfl-cn: Communication-efficient hierarchical personalized federated edge learning via complex network feature clustering." *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2022.
9. Parakala, Adityamallikarjunkumar, and Jyothirmay Swain. "AI-Powered Intelligent Automation Emerges." *International Journal of Artificial Intelligence, Data Science, and Machine Learning* 3.4 (2022): 96-106.
10. Bharati, Subrato, et al. "Federated learning: Applications, challenges and future directions." *International Journal of Hybrid Intelligent Systems* 18.1-2 (2022): 19-35.
11. Demertzis, Konstantinos, et al. "An explainable semi-personalized federated learning model." *Integrated Computer-Aided Engineering* 29.4 (2022): 335-350.
12. Sun, Peng, et al. "A profit-maximizing model marketplace with differentially private federated learning." *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022.
13. Chen, Tianyu, et al. "The-x: Privacy-preserving transformer inference with homomorphic encryption." *arXiv preprint arXiv:2206.00216* (2022).
14. Qi, Jiahao, et al. "High-quality model aggregation for blockchain-based federated learning via reputation-motivated task participation." *IEEE Internet of Things Journal* 9.19 (2022): 18378-18391.
15. Saha, Sudipan, and Tahir Ahmad. "Federated transfer learning: Concept and applications." *Intelligenza Artificiale* 15.1 (2021): 35-44.
16. Yu, Shuai, et al. "When deep reinforcement learning meets federated learning: Intelligent multitimescale resource management for multiaccess edge computing in 5G ultradense network." *IEEE Internet of Things Journal* 8.4 (2020): 2238-2251.