Innovation Economics

https://www.aimspress.com/journal/InnoEcon

InnoEcon, 1(1): 68–85.
DOI:10.3934/InnoEcon.2026004
Received: 07 April 2026
Revised: 16 June 2026
Accepted: 22 June 2026
Published: 14 July 2026

Research article

Intelligent customer retention prediction using distributed federated learning

Prakash Choudhary[1,*], **Amandeep Pooni**[2] **and Ravi Raj Choudhary**[3]

[1] Department of Computer Science and Engineering, Central University of Rajasthan, Rajasthan, India

[2] Department of Computer Science and Engineering, National Institute of Technology, Hamirpur, Himachal Pradesh, India

[3] Department of Computer Science, Central University of Rajasthan, Rajasthan, India

* **Correspondence:** Email: prakash.choudhary@curaj.ac.in.

Abstract: Customer retention is a key business objective, as retaining existing customers improves profitability and long-term growth. Predicting customer churn enables organizations to identify at-risk customers and implement effective retention strategies. However, customer data are often distributed across organizations and cannot be shared due to privacy, regulatory, and competitive concerns.
This study explored privacy-preserving approaches for customer retention prediction using internet service provider (ISP) data. We applied federated learning and a distributed form of incremental machine learning on internet service provider data to study and compare privacy conserving methods of predicting customer churn. With the federated learning approach, we divided and distributed data across different clients, thereby training individual models on those clients and the resulting models were averaged on a central server. In the distributed incremental learning approach, instead of transferring all models to a central server, a single model is trained and passed on to the next server to train on new data utill all clients are covered and the resultant model is synchronized across all clients. The findings provide practical insights for organizations seeking to balance predictive performance, customer retention, and data privacy requirements.

Keywords: stacked ensemble model; user retention; federated learning; business analytics

JEL Codes: C45, C53, C81, C84, C88, L84

1. Introduction

User retention is the continued use of a product or feature by customers. User churn is a subset of user retention metrics that looks at the number of users who have stopped using a product over a specific period of time. User retention is an important metric as it directly corresponds to the revenue

generated by an enterprise and its business decisions. Retaining existing users is cheaper than acquiring new ones because significant marketing costs are required to acquire new users, whereas existing users only need some form of incentive to prevent them from either completely stopping the use of a service or switching to a competitor offering better incentives. The acquisition cost can be five times higher than the retention cost Ali and Shabn (2024).

Federated learning McMahan et al. (2017) (also known as collaborative learning) is a machine learning technique that trains an algorithm across multiple decentralized edge devices or servers holding local data samples without exchanging them. This approach stands in contrast to traditional centralized machine learning techniques, where all local datasets are uploaded to a single server, as well as to classical decentralized approaches, which often assume that local data samples are identically distributed. Federated learning can be applied to enable collaboration among different organizations to create more generalized user retention prediction models that can be used by organizations that do not yet have a large user base.

Although federated learning preserves privacy to a large extent, recent advancements have led to various methods for tracing general data patterns from trained models, as well as other types of attacks Yang et al. (2020); Ma et al. (2020). While these methods do not completely compromise the underlying data, they remain a concern for applications such as healthcare and ad tracking, where data patterns and metadata can be as valuable as the actual data. Therefore, we propose a hybrid federated learning architecture that employs incremental learning in a distributed manner, offering enhanced privacy preservation compared to standard federated learning.

The primary objectives of this study are as follows:

1. Develop a privacy-preserving and scalable framework for customer retention prediction using federated and distributed incremental learning techniques.
2. Propose a hybrid architecture for federated learning to enhance privacy as compared to traditional federated learning.
3. Evaluate privacy-preserving customer-retention-prediction approaches and compare their effectiveness with conventional centralized machine learning.

We implemented federated learning and a modified distributed incremental learning framework, and compared their performance with a centralized machine learning model.

The paper has been structured as follows: Section 2 presents a review of relevant research studies that were examined and referenced to support this work. Section 3 describes the dataset, preprocessing steps, and the model training and testing procedures. It also discusses the different federated learning architectures used for model training, and presents diagrams illustrating each approach. Section 4 presents and discusses the outcomes of the experiments conducted in this study. Section 5 summarizes the key findings and conclusions of the study. It also highlights the study's limitations and suggests directions for future research.

2. Literature review

The architectural motivation for the framework proposed in this study has been derived from sources directly addressing user retention and customer churn problems, as well as applications of federated learning in medicine and COVID-19 detection. Recent advancements in federated learning have largely focused on preserving the data privacy of hospitals and other medical organizations.

Naz et al. (2018) discussed different types of churners as well as several approaches for predicting customer churn. There are two main types of customer churn depending on who initiates the termination of the service relationship. If a customer voluntarily leaves a service, they are classified as voluntary churners. If they are removed from the service due to reasons such as non-payment of dues or violation of terms of service, they are classified as involuntary churners. Voluntary churners can be further categorized into deliberate and incidental churners. Deliberate churners leave a service primarily due to dissatisfaction, whereas incidental churners may be forced to leave due to personal circumstances despite their willingness to continue using the service. Among these groups, deliberate churners are of particular importance to businesses because they can potentially be retained by addressing their concerns or offering incentives. However, identifying such customers is essential, which is where churn prediction models play a significant role. The authors also discussed various attributes that influence churn. Some attributes, such as service usage duration, service cost, and bill payment history, are common across many user retention applications, while others are domain-specific. For example, in the telecommunications industry, factors such as prepaid or postpaid subscription plans can significantly affect churn prediction.

Sikri et al. (2024) evaluated several machine learning algorithms, including perceptron, multi-layer perceptron, naïve bayes, logistic regression, K-nearest neighbors(KNN), and decision trees, for improving customer retention in the telecommunications sector. In Amin et al. (2023), a novel adaptive learning approach for customer churn prediction was proposed using a naïve bayes classifier combined with a genetic algorithm-based feature weighting strategy.

In Lalwani et al. (2022), various machine learning techniques were implemented to develop a customer-churn prediction system, including logistic regression, naïve bayes, support vector machines(SVMs), decision tree, random forest, extra trees classifier, and boosting algorithms such as AdaBoost, XGBoost, and CatBoost. The results demonstrated that AdaBoost and XGBoost outperformed the other methods.

Similarly, Chaubey et al. (2022) presented a customer-purchase-behavior prediction system using machine learning classification techniques. The results indicated that a hybrid ensemble stacking classifier (KnnSgd) outperformed other methods, including logistic regression, decision trees, K-nearest neighbors, naïve bayes, support vector machines, random forest, and stochastic gradient descent.

Shi et al. (2021) proposed Fed-Ensemble, a framework that combines federated learning and ensemble learning. Traditional federated learning based on federated averaging can result in high variance. Fed-Ensemble reduces this variance and achieves better generalization performance. After training all base models, predictions are generated using an ensemble strategy according to the following equation, where k denotes the type of base model.

$$f_w(x) = \frac{1}{K} \sum_{k=1}^{K} f_{w_k}(x) \tag{1}$$

Sheller et al. (2020), analyzed how federated learning models compare with centralized learning models. Various collaborative learning paradigms were also explored and compared with federated learning. It was concluded that the benefits offered by federated learning outweigh the errors introduced by combining models trained on different datasets.

Machine learning models trained on data from a single institution are sometimes overfitted due to factors such as demographics and the type of hospital, and therefore cannot be applied on a generalized

scale. Traditionally, institutions need to collaborate and share patient data to create a common data pool for training machine learning models. This method is known as collaborative data sharing (CDS). In contrast, federated learning is a distributed learning paradigm that does not require participating organizations to share data. Such an approach is referred to as data-private collaborative learning. However, due to privacy restrictions, this approach cannot be applied in all use cases.

Rodan et al. (2014) applied negative correlation learning (NCL) to an ensemble of multi-layer perceptron (MLP) models to address the problem of class imbalance. In user retention applications, the number of users who churn is typically much lower than the number of users who continue using the service. NCL applied to an MLP was compared with other methods such as artificial neural network(ANN), KNN, SVM, Bagging, and AdaBoost, and it performed on par with the best-performing models in the comparison group.

Hu et al. (2021) compared various types of federated learning with traditional distributed learning. Distributed machine learning differs from federated learning as it still requires a central server to handle all the data, while the processing is decentralized. In contrast, in federated learning, both the data and processing are decentralized. Federated learning can be divided into three primary types: horizontal federated learning, vertical federated learning, and federated transfer learning. Horizontal federated learning focuses on clients with similar data characteristics but different data samples. Vertical federated learning focuses on clients with overlapping data samples but different characteristics, while federated transfer learning focuses on scenarios where the overlap of both data samples and data characteristics is limited. Depending on the underlying model, federated learning frameworks can also be categorized as federated supervised learning (the federated linear algorithm, federated support vector machine, and federated decision tree algorithm), federated semi-supervised learning (federated match and federated logistic regression), and federated unsupervised learning (the federated GAN network).

Dou et al. (2021) used CT scans from hospitals in Hong Kong. These data were used to train individual models, which were later combined. The data were also aggregated and used to train a centralized model for performance comparison. The trained model was validated both internally and externally using data from hospitals in China and Germany. The experiment was conducted using six configurations, namely, three individual models corresponding to three data centers, their ensemble (combining different model types), one joint model, and one federated learning model. The evaluation metrics included receiver operating characteristic (ROC) analysis for lesion detection, area under the curve (AUC) for each ROC, and mean average precision (mAP) for evaluating bounding-box accuracy. The results were also compared with interpretations provided by radiologists. Among the individual models, the model trained on the largest dataset achieved the highest AUC. The federated learning model performed comparably to the best individual model and the ensemble model.

Several recent studies have investigated the application of federated learning to churn and user retention prediction. In Okonkwo and Englama (2025), an AI-enhanced federated learning framework was proposed for real-time customer churn prediction, demonstrating improved decision-making performance while preserving user data privacy. Similarly, Krishnan (2024) presented a telecom churn prediction model based on federated neural networks using the FedAvg algorithm. Their approach achieves competitive accuracy and enables distributed model training across geographically dispersed datasets without requiring the sharing of raw data. However, these approaches are largely based on static models and do not adequately capture the dynamic nature of user behavior. Moreover, traditional machine learning techniques for churn prediction were comprehensively reviewed in Imani et al. (2025),

where the effectiveness of models such as decision trees, random forests, and neural networks was highlighted. The study also emphasized key challenges, including the lack of privacy-preserving mechanisms and limited adaptability to evolving data environments.

To address the issue of dynamic data distributions, recent research has focused on federated continual and incremental learning methods. The work presented in Wu et al. (2024) introduced a federated class-incremental learning framework (FedCLASS) that incorporates self-distillation techniques to mitigate catastrophic forgetting during model updates. In addition, incremental federated learning was explored in other application domains. For instance, Pekar et al. (2024) proposed an incremental federated learning framework for network traffic classification that adapts to evolving data patterns and improves classification performance over time. While this work highlights the importance of handling streaming and non-stationary data, it does not directly address user-centric tasks such as churn or retention prediction.

3. Methodology

For centralized learning, all data and models are located on the same client, as shown in Figure 1. For federated learning, the dataset is divided among a varying number of clients, as shown in Figure 2. Each client has a separate model that is trained on the data available for that particular client. After the training phase, models from all clients are sent to a central server, where they are aggregated using a weighted average to form a generalized model. This central model is then distributed to all clients and can be used for prediction purposes.

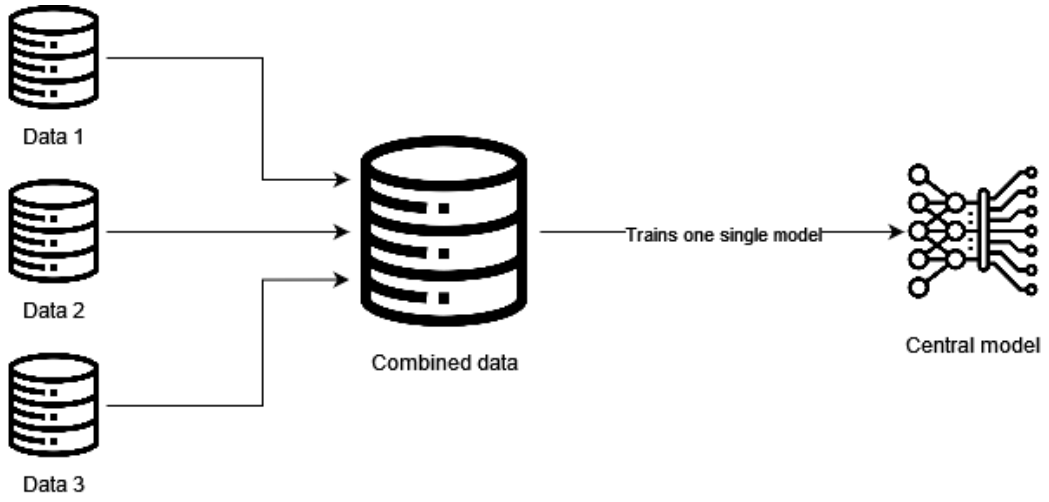

Figure 1. Centralized learning model.

In our distributed incremental learning model, a single model is trained and then passed to the next server for training on new data until all clients have been covered. The resultant model is then synchronized across all clients. Logistic regression, MultinomialNB, BernoulliNB, and Perceptron models have been used for training, and the results of centralized machine learning, federated learning, and distributed incremental learning were compared based on accuracy, precision, recall, and F1-score. The time performance metric measures only the training time and does not include server communication time.

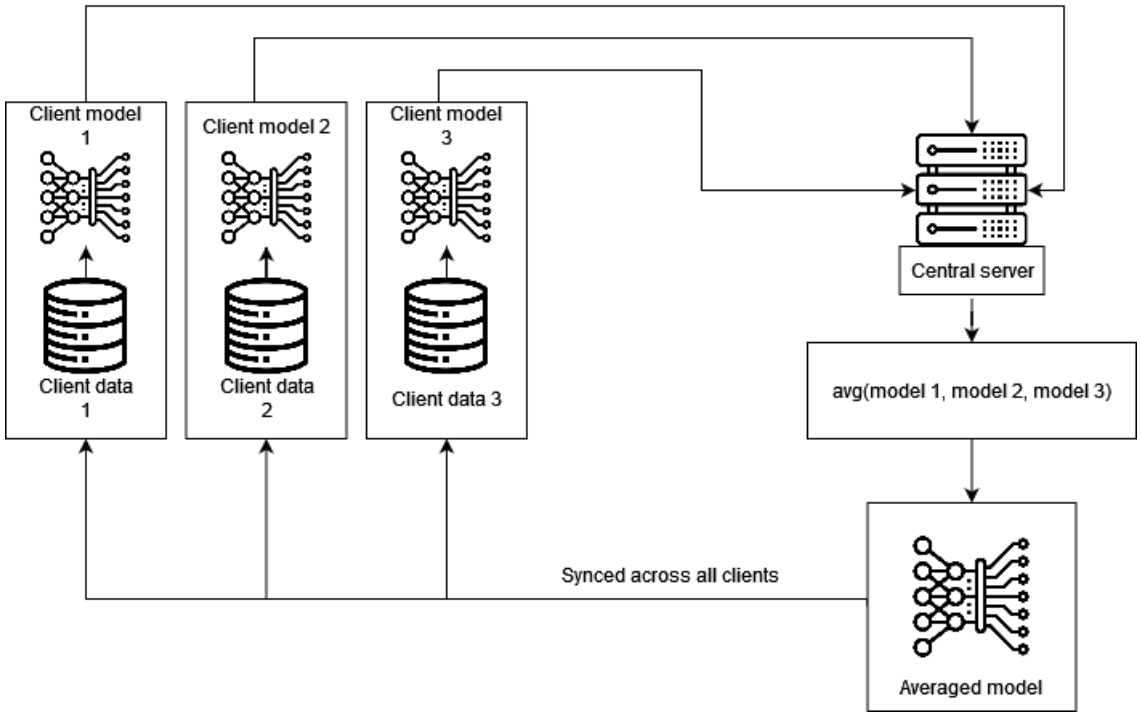

Figure 2. Federated learning model.

3.1. Problem statement

Creating a generalized user retention prediction model requires a diverse dataset, which is often distributed across various organizations that may or may not be willing to share their data with each other or with a third-party organization due to privacy concerns.

3.2. Dataset

In this study, we used the publicly available Internet Service Provider (ISP) Customer Churn dataset obtained from Kaggle Kunt (2021), Do et al. (2017). The dataset contains a total of 72,274 customer records, including both churned and non-churned users. Specifically, 40,050 instances correspond to customers who have churned, while 32,224 correspond to customers who have retained the service.

The dataset consists of 11 attributes describing customer demographics, service usage patterns, and account-related information. For this study, 10 relevant features were selected for model training and prediction, while redundant or less informative attributes were excluded. These features include variables related to subscription type, service usage behavior, and billing information, which are commonly used in churn analysis.

The data span the period from 2007 to 2019, capturing long-term customer behavior and enabling the analysis of temporal patterns in churn. Additionally, the dataset exhibits class imbalance and variability in feature distributions, making it suitable for evaluating machine learning models under realistic conditions.

To support federated learning, the dataset was partitioned into multiple subsets representing different clients or nodes. Each subset was used for local model training, thereby simulating a distributed data

environment in which raw data are not shared across nodes. This setup enables the evaluation of privacy-preserving learning approaches while maintaining predictive performance.

3.3. Preprocessing

During preprocessing, the primary identifier attribute (Customer ID) was removed, as it does not contribute to the prediction task. Missing values present in certain attributes were handled by replacing them with the mean value of the respective feature to ensure data consistency.

To analyse the impact of dataset size on model performance in both federated and distributed incremental learning settings, experiments were conducted using varying sample sizes ranging from 10,000 records to the full dataset of 72,274 records. This enabled the evaluation of scalability and learning behavior under different data volumes.

All numerical features were normalized using the MinMaxScaler, which scales input values to the range (0, 1). This normalization ensures uniformity across features and improves the convergence behavior of machine learning models when evaluated using different estimators.

3.4. Model selection and rationale

The logistic regression, bernoulli naïve bayes (BernoulliNB), multinomial naïve Bayes (MultinomialNB), and Perceptron models were selected because they are lightweight, computationally efficient, and well-suited for high-dimensional textual and categorical datasets commonly used in prediction and classification problems. These models also provide strong baseline performance and are widely adopted in machine learning research for comparative evaluation.

Logistic regression was selected because it is a robust linear classifier that performs well on sparse and high-dimensional data. It provides probabilistic outputs, supports regularization to reduce overfitting, and offers good interpretability of feature importance. Logistic regression generally performs effectively when the relationship between variables is approximately linear.

BernoulliNB was chosen because it is particularly suitable for binary or Boolean feature representations, where variables indicate the presence or absence of attributes. This makes it effective for datasets with binary features such as text occurrence vectors. It is computationally fast and works well even with limited training data.

MultinomialNB was selected due to its effectiveness in handling frequency-based discrete features, especially in text classification problems where word occurrence counts are important. It performs efficiently on sparse datasets and requires relatively low computational resources.

The **Perceptron** model was included because it is an efficient online learning algorithm capable of handling large-scale and streaming datasets. It is computationally inexpensive and suitable for linearly separable data. In addition, the Perceptron can adapt incrementally during training, which is beneficial in distributed and dynamic learning environments.

These models were preferred over more complex approaches such as random forest, support vector machines (SVMs), or deep learning models because the objective of this research was to evaluate computationally efficient and scalable algorithms suitable for distributed and federated learning environments. Complex models generally require higher computational power, increased communication overhead, longer training time, and larger datasets, which may not be practical in resource-constrained or decentralized systems. Therefore, the selected models provide a balanced

trade-off between prediction performance, interpretability, scalability, and computational efficiency.

3.5. Training and testing

The training and testing process in this study is carried out in three different architectural settings to evaluate model performance in incremental, centralized, federated, and distributed learning environments. For consistency, the same set of machine learning models and hyperparameters is maintained across all experimental setups.

Logistic regression is trained with a maximum of 500 iterations, using an L2 regularization penalty to prevent overfitting. The optimization is performed using the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) solver, which is well-suited to handling large-scale datasets.

For probabilistic modeling, naïve bayes classifiers are employed using both bernoulli and multinomial variants. In both cases, additive smoothing (Laplace/Lidstone smoothing) with a parameter value of 1.0 is applied to handle zero-frequency issues and improve model generalization.

Additionally, a perceptron model is utilized as a linear classifier. The training process is controlled using a stopping criterion of 10^{-3}, where the optimization terminates once the change in loss between successive iterations falls below this threshold, ensuring efficient convergence.

Model performance is evaluated under varying data distributions and client configurations to assess the effectiveness of each approach in handling distributed and evolving datasets.

3.6. Centralized machine learning

For centralized machine learning, four estimators were used. In this configuration, the entire dataset is available in memory, and all models are trained using the same dataset. The ratio of training to testing data is 70:30. The prediction results for centralized machine learning are presented in Table 1, while the computational time performance of the various models is shown in Table 2.

Table 1. Centralized machine learning results for different estimators.

Classification	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.79	0.80	0.83	0.82
BernoulliNB	0.82	0.93	0.73	0.82
MultinomialNB	0.67	0.66	0.84	0.74
Perceptron	0.74	0.90	0.59	0.71

Table 2. Training time for centralized machine learning for different estimators.

Classification	Time taken (seconds)
Logistic Regression	0.534
BernoulliNB	0.031
MultinomialNB	0.016
Perceptron	0.122

In the centralized machine learning setup, four different estimators were employed to evaluate model performance. Since the entire dataset is available at a single location, all models are trained using the complete dataset without any data partitioning.

The dataset is divided into training and testing sets using a 70:30 ratio, where 70% of the data is used for training and the remaining 30% is used for testing. This split ensures a fair evaluation of model generalization performance.

A variety of estimators, including linear and probabilistic models, are utilized to analyze their effectiveness under a centralized learning paradigm. The prediction results obtained from these models are presented in Table 1, while the corresponding computational time performance is summarized in Table 2.

This centralized setup serves as a baseline for comparison with federated and distributed incremental learning approaches, enabling an assessment of the trade-offs in terms of accuracy, efficiency, and scalability.

3.7. Federated learning

For federated learning, logistic regression was used as the base model for all clients. The dataset was divided equally among all clients, along with one additional shard reserved for testing purposes. On each client, the local testing-to-training data ratio was set to 0.1, as there was no requirement to evaluate performance immediately using local data. Instead, testing was performed using the separate data shard created from the original dataset split, as this provides a better assessment of the generalization capability of the federated learning model.

Each client's model was trained solely on the data available at that client. Once training had been completed across all clients, the models were aggregated using a weighted average, as shown in the following equation, where N is the total number of clients, d_i represents the number of data entries on client i, and D denotes the total number of data entries across all clients.

$$avg\_weights = \sum_{i=1}^{N} \frac{d_i}{D} client\_weights_i \tag{2}$$

The resultant model is synchronized across all clients, ensuring that each client possesses the same aggregated model, which is more generalized than any individual client model. For model evaluation, the testing data shard is used, and a testing-to-training ratio of 0.9 is applied, as the majority of this data is intended for testing purposes. Predictions are then generated using this testing dataset.

Table 3. Federated learning using logistic regression with a varied number of clients.

Number of clients	Accuracy	Precision	Recall	F1-score
5	0.84	0.70	0.33	0.44
10	0.83	0.75	0.23	0.36
20	0.82	1.00	0.23	0.38
30	0.79	1.00	0.29	0.45
40	0.76	1.00	0.29	0.45
50	0.71	1.00	0.26	0.42

The data were partitioned six different times to simulate varying numbers of clients. The results obtained from the federated learning model under different client configurations are presented in Table 3. The corresponding time performance for varying numbers of clients is shown in Table 4.

Table 4. Training time for federated learning using logistic regression with varied number of clients.

Number of clients	Time taken (seconds)
5	0.684
10	0.837
20	1.182
30	1.653
40	1.828
50	2.193

3.8. Distributed incremental learning

Incremental learning He et al. (2018) has traditionally been used to perform machine learning on large datasets where the entire dataset cannot be accommodated in the available memory of the system on which the model is being trained. Consequently, partial fitting is typically employed using a stream of data batches, allowing the model to learn sequentially, as shown in Figure 3.

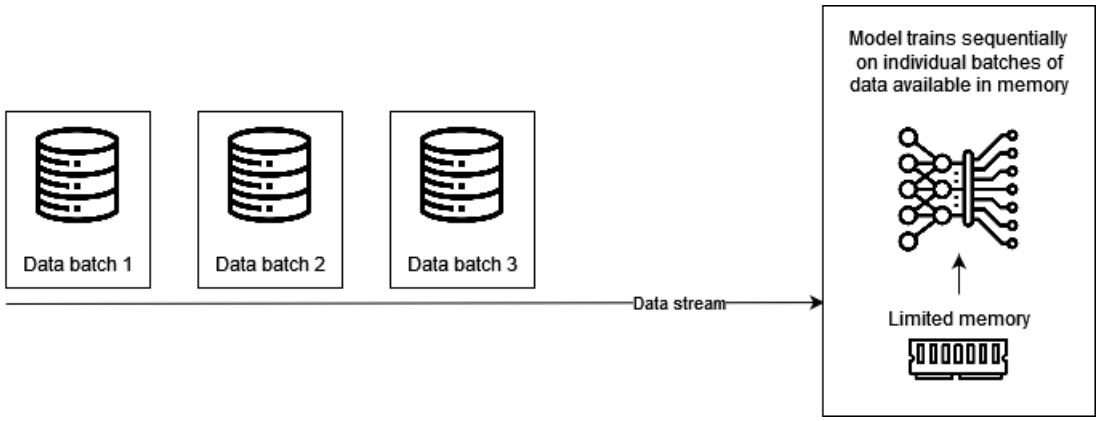

Figure 3. Incremental learning model.

By transferring the model across clients, we have developed a modified version of federated learning in which the model treats the data on each client as part of a continuous data stream and applies the partial fit method at each client, as shown in Figure 4. Once all clients have been covered, the model is synchronized across all clients, resulting in a model that has been trained using data from all clients without requiring any client's data to leave its local environment.

The dataset was divided using a training-to-testing ratio of 70:30. No client maintains an independent model of its own. Three base models were used for distributed incremental learning: BernoulliNB, MultinomialNB, and perceptron. Each base model was trained sequentially on the same dataset partition to ensure a fair comparison.

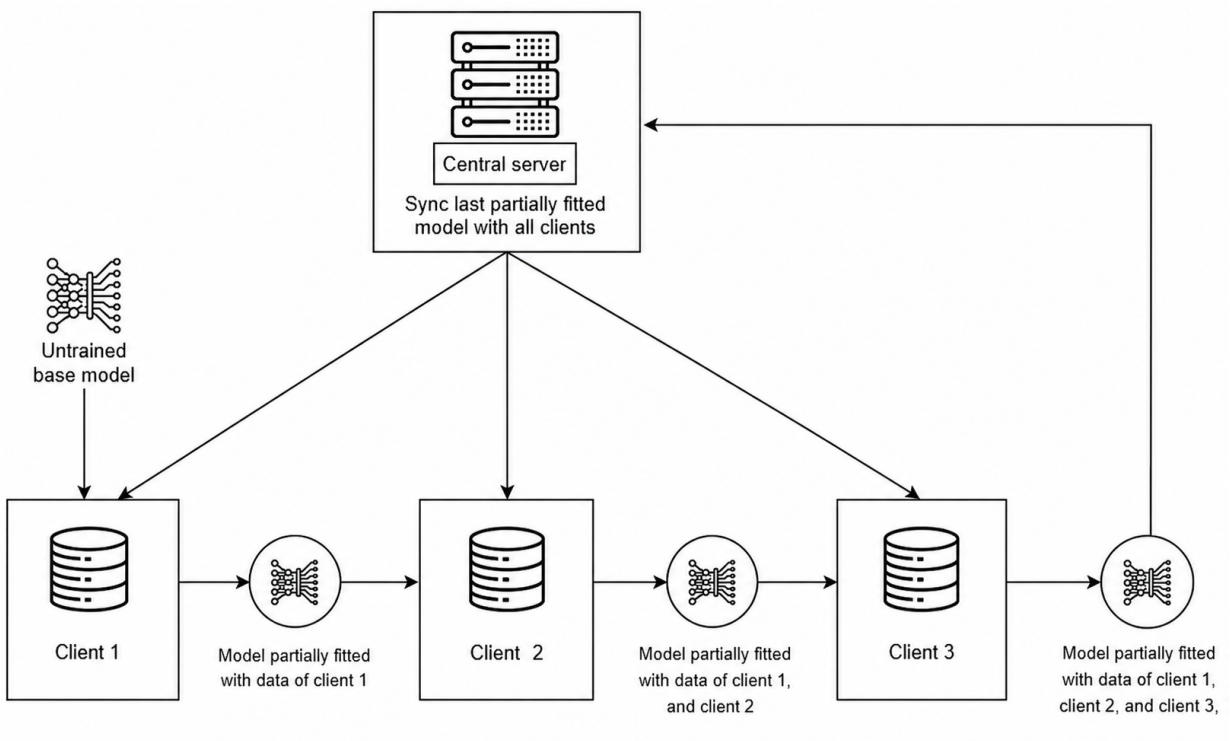

Figure 4. Distributed incremental learning model.

The data were partitioned six different times to simulate varying numbers of clients, consistent with the federated learning setup. The results obtained from the distributed incremental learning model under different client configurations are presented in Tables 5, 6, and 7. The corresponding time performance of the different models for varying numbers of clients is presented in Table 8.

Table 5. Distributed incremental using BernoulliNB with a varied number of clients.

Number of clients	Accuracy	Precision	Recall	F1-score
5	0.82	0.93	0.73	0.82
10	0.82	0.93	0.73	0.82
20	0.82	0.93	0.73	0.82
30	0.82	0.93	0.73	0.82
40	0.82	0.93	0.73	0.82
50	0.82	0.93	0.73	0.82

Table 6. Distributed incremental using MultinomialNB with a varied number of clients.

Number of clients	Accuracy	Precision	Recall	F1-score
5	0.67	0.66	0.84	0.74
10	0.67	0.66	0.84	0.74
20	0.67	0.66	0.84	0.74
30	0.67	0.66	0.84	0.74
40	0.67	0.66	0.84	0.74
50	0.67	0.66	0.84	0.74

Table 7. Distributed incremental using perceptron with a varied number of clients.

Number of clients	Accuracy	Precision	Recall	F1-score
5	0.77	0.73	0.95	0.82
10	0.79	0.79	0.86	0.82
20	0.59	0.58	1.00	0.73
30	0.56	0.56	1.00	0.71
40	0.66	0.62	0.98	0.76
50	0.67	0.63	0.97	0.77

Table 8. Training time for distributed incremental learning using different estimators with varied number of clients.

Classification	Time taken (seconds) by a varied number of clients					
	5	10	20	30	40	50
BernoulliNB	0.014	0.017	0.024	0.031	0.037	0.044
MultinomialNB	0.015	0.020	0.029	0.038	0.049	0.059
Perceptron	0.044	0.053	0.060	0.067	0.074	0.083

4. Results and discussion

Results are compared based on both the machine learning models and the number of clients used. Training time performance is also measured for different models and varying numbers of clients.

4.1. Performance metrics

The primary metric used for comparing performance across different architectures, models, and numbers of clients is accuracy, which is defined as the ratio of correct predictions to the total number of predictions. Additional evaluation metrics include precision, which is calculated as the number of true positives divided by the sum of true positives and false positives; recall, which is calculated as the number of true positives divided by the sum of true positives and false negatives; and the F1-score, which is the harmonic mean of precision and recall.

4.2. Results and analysis

The best outcome for the centralized model was achieved by BernoulliNB, with precision and recall values of 0.93 and 0.73, respectively. For federated learning, logistic regression achieved precision and recall values of 0.70 and 0.33, respectively, when using five clients. These values are lower than those obtained by the centralized logistic regression model, which achieved precision and recall values of 0.80 and 0.83, respectively.

In the distributed incremental learning model, both naïve Bayes classifiers performed similarly to the centralized learning model, with no noticeable effect resulting from the number of clients. The perceptron model in the distributed incremental learning setup exhibited lower precision and higher recall compared with the centralized model, achieving values of 0.73 and 0.95, respectively, when using five clients.

The following tables present a comprehensive comparison of the results obtained from the experiments.

4.3. Discussion

For logistic regression, the federated learning model achieves better accuracy but a very reduced precision, recall, and F1-score as compared to centralized machine learning. Distributed incremental learning models for BernoulliNB and MultinomialNB perform the same as their centralized counterparts across all metrics. Both of these naive Bayes methods also show no effect upon varying the number of clients. By applying distributed incremental learning in the federated learning architecture, we created a model which for most estimators performs very close if not equally well as compared to centralized machine learning. Our federated learning model, though, gives us better accuracy than both the centralized model and the distributed learning model gives us far less precision, recall, and F-1 scores. In practice, distributed incremental learning will consume significantly more time as the model is trained sequentially across the whole set of clients and transit time from one client to another will also add up. In federated learning, in each round, training is performed in a parallel manner for all the individual client models and all the models can be sent to the server all at once.

The main advantage of using distributed incremental learning is how models are handled, and this has a direct impact on data privacy. While in traditional federated learning models, we can back trace the general data patterns used to train a particular model from each client, thus having access to characteristics of each client. In distributed incremental learning, the presence of only one model ensures that data characteristics from every previous client in the sequence is mixed up, and except for the first client, the data pattern for any particular client cannot be deduced. This has direct implications for removing one of the major privacy concerns faced by current federated learning architectures. In this study, focus has been made to maintain uniformity among the types of estimators used in performing machine learning. But different estimators have different uses and some of them are not compatible or applicable for intended-use-cases. In distributed incremental learning, the use of logistic regression is omitted as the partial fit method is not available for logistic regression. Classifiers like the random forest classifier, decision tree classifier, and KNN classifier that offer even better accuracy are not used due to the unavailability of partial fitting methods for these classifiers.

Based on the experimental results, we propose the following guidelines: (i) centralized machine learning may be preferred when data sharing is permissible and maximum predictive performance is

required; (ii) federated learning is suitable when data privacy and regulatory compliance are critical concerns; and (iii) distributed incremental learning can be adopted in environments where data arrive continuously and computational resources are distributed across multiple locations.

5. Managerial implications and practical implementation guidelines

Customer retention is often more cost-effective than acquiring new customers. However, modern organizations typically store customer information across multiple business units, regions, or partner organizations, making it difficult to develop comprehensive churn prediction models whilst complying with privacy regulations.

The proposed privacy-preserving framework enables organizations to predict customer churn without transferring or centralizing sensitive customer data. This approach helps businesses improve customer retention whilst maintaining customer trust and regulatory compliance.

How the proposed approach supports marketing decisions: Rather than replacing existing marketing strategies, the proposed system provides marketers with an early warning mechanism that identifies customers at risk of leaving.

Marketing teams can use these predictions to:

1. Prioritize high-risk customer segments.
2. Design personalized retention campaigns.
3. Allocate retention budgets more efficiently.
4. Improve customer experience through proactive interventions.
5. Measure the effectiveness of retention initiatives over time.

5.1. Practical implementation framework

The deployment process can be organized into five steps:

- **Step 1- Define retention objectives:** Identify business goals, such as reducing monthly churn rates, increasing customer lifetime value, or improving subscription renewal rates.
- **Step 2- Identify relevant customer data sources:** Customer information may exist across multiple branches, regions, subsidiaries, or business partners. Examples of useful customer data include:
    1. Service usage records
    2. Customer complaints
    3. Billing information
    4. Contract history
    5. Customer support interactions

  The data remain within each organization and are not shared externally.
- **Step 3- Train privacy-preserving models:** Each participating organization trains the model locally using its own customer data.
    1. In federated learning, locally trained models are aggregated centrally without sharing raw data.
    2. In distributed incremental learning, a model is sequentially updated across participating organizations.

- **Step 4- Generate churn risk scores:** The trained model assigns a churn probability score to each customer.
- **Step 5- Implement retention actions:** Marketing teams can use churn scores to trigger interventions such as:

  1. Personalized offers
  2. Loyalty programs
  3. Service upgrades
  4. Proactive customer support
  5. Contract renewal incentives

5.2. Business benefits and challenges

Selecting an appropriate privacy-preserving learning approach requires organizations to balance business objectives, technical capabilities, and operational constraints. Although both federated learning and distributed incremental learning enable collaborative customer retention prediction without sharing raw customer data, they differ in terms of scalability, communication requirements, implementation complexity, and sensitivity to data heterogeneity.

Table 9. Comparison of federated learning and distributed incremental learning from a business and implementation perspective for privacy-preserving customer retention prediction.

Aspect	**Federated Learning**	**Distributed Incremental Learning**
Customer data sharing	Not required	Not required
Privacy protection	High	High
Regulatory compliance	Strong	Strong
Communication overhead	Moderate to high	Low
Implementation complexity	Moderate	Low to moderate
Risk of model bias	Lower	Higher if data are highly heterogeneous
Scalability	High	Moderate

Table 9 highlights the trade-offs that decision-makers should consider when selecting a privacy-preserving approach for customer retention prediction. Federated learning offers greater scalability and a reduced risk of model bias through parallel model training, although it requires more communication and coordination. In contrast, distributed incremental learning reduces communication overhead and may be easier to deploy in resource-constrained environments, but it can be more sensitive to variations in customer data across participating organizations. The choice between these approaches should therefore depend on organizational priorities, available technical resources, and the complexity of the data ecosystem.

5.3. Organizational considerations

Successful implementation requires close collaboration among marketing, analytics, and IT teams to align business objectives with technical capabilities. Key stakeholder responsibilities include:

1. **Marketing teams:** define business objectives and retention strategies.

2. **Data analysts:** interpret churn predictions and evaluate campaign effectiveness.
3. **IT teams:** manage secure model training and system integration.

6. Conclusions

For customer retention prediction, the federated learning model achieved higher overall accuracy; however, its precision, recall, and F1-score were lower than those obtained through centralized machine learning. In contrast, the distributed incremental learning models based on BernoulliNB and MultinomialNB delivered performance comparable to their centralized counterparts across all evaluation metrics, while remaining unaffected by variations in the number of participating clients.

The results demonstrate that privacy-preserving learning approaches can effectively support customer retention analytics without requiring organizations to share sensitive customer data. In particular, distributed incremental learning achieved predictive performance close to that of centralized machine learning while maintaining data confidentiality across participating organizations. This makes it a promising solution for businesses operating in privacy-sensitive environments where direct data-sharing is restricted by regulatory, legal, or competitive considerations.

In our work, data are distributed across multiple clients, where local models are trained independently and subsequently aggregated on a central server. Furthermore, we propose a distributed sequential learning approach that enhances privacy compared with conventional federated learning. In this approach, a single model is sequentially trained across clients, which progressively blends data patterns from different clients. While the privacy of the first client may still be vulnerable during the initial training stage, the model becomes increasingly difficult to associate with any specific client's data as it is updated across subsequent clients. Since customer data remain within their originating organizations and only model parameters are transferred, the risk of exposing customer information is significantly reduced.

The proposed approach enables organizations to leverage broader customer insights for churn prediction and retention planning while retaining control over their proprietary customer information. Such capabilities can assist marketers in identifying customers at risk of churn, designing targeted retention campaigns, and improving customer lifetime value without compromising data privacy.

Overall, the study highlights the potential of privacy-preserving learning frameworks for customer retention strategies. Future research can explore additional machine learning techniques to further improve predictive performance and broaden their application across different industries.

Author contributions

Prakash Choudhary: Conceptualization, Formal Analysis, Investigation, Methodology, Software, Supervision, Writing – Review and Editing.

Amandeep Pooni: Validation, Visualization, Writing – Original Draft Preparation, Conceptualization, Data Curation.

Ravi Raj Choudhary: Formal Analysis, Investigation, Software, Supervision, Writing – Review and Editing.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Funding details

The authors received no financial support for the research of this article.

Conflict of interest

The authors declare that they have no conflict of interest.

References

Ali N, Shabn OS (2024) Customer lifetime value (clv) insights for strategic marketing success and its impact on organizational financial performance. *Cogent Bus Manag* 11. https://doi.org/10.1080/23311975.2024.2361321

Amin A, Adnan A, Anwar S (2023) An adaptive learning approach for customer churn prediction in the telecommunication industry using evolutionary computation and naive bayes. *Appl Soft Comput* 137: 110103. https://doi.org/10.1016/j.asoc.2023.110103

Chaubey G, Gavhane PR, Bisen D (2022) Customer purchasing behavior prediction using machine learning classification techniques. *J Amb Intel Hum Comp* 14: 16133–16157. https://doi.org/10.1007/s12652-022-03837-6

Do D, Huynh P, Vo P, et al. (2017) Customer churn prediction in an internet service provider. *2017 IEEE International Conference on Big Data (Big Data)*, Boston, MA, USA, 3928–3933. https://doi.org/10.1109/BigData.2017.8258400

Dou Q, So TY, Jiang M, et al. (2021) Federated deep learning for detecting covid-19 lung abnormalities in ct: A privacy preserving multinational validation study. *npj Digit Med* 4: 60. https://doi.org/10.1038/s41746-021-00431-6

He H, Chen S, Li K, et al. (2018) Incremental learning from stream data. *Ieee T Neur Net Lear* 22: 1901–1914. https://doi.org/10.1109/TNN.2011.2171713

Hu K, Li Y, Xia M, et al. (2021) Federated learning: A distributed shared machine learning method. *Complexity* 2021: 20. https://doi.org/10.1155/2021/8261663

Imani M, Joudaki M, Beikmohammadi A, et al. (2025) Customer churn prediction: A systematic review of recent advances, trends, and challenges in machine learning and deep learning. *Mach Learn Knowl Extr* 7: 105. https://doi.org/10.3390/make7030105

Krishnan M (2024) Enhancing customer churn prediction in telecom using federated learning. Available from: https://norma.ncirl.ie/id/eprint/7953.

Kunt MS (2021) Internet service provider customer churn dataset. Available from: https://www.kaggle.com/datasets/mehmetsabrikunt/internet-service-churn.

Lalwani P, Mishra MK, Chadha JS, et al. (2022) Customer churn prediction system: A machine learning approach. *Computing* 104: 271—294. https://doi.org/10.1007/s00607-021-00908-y

Ma C, Li J, Ding M, et al. (2020) On safeguarding privacy and security in the framework of federated learning. *Ieee Network* 34: 242–248. https://doi.org/10.1109/MNET.001.1900506

McMahan HB, et al. (2017). Communication-efficient learning of deep networks from decentralized data. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Fort Lauderdale, Florida. https://doi.org/10.48550/arXiv.1602.05629

Naz NA, Shoaib U, Sarfraz MS (2018) A review on customer churn prediction data mining modeling techniques. *Indian J Sci Technol* 11: 1–7. https://doi.org/10.17485/ijst/2018/v11i27/121478

Okonkwo R, Englama KS (2025) Ai-enhanced real-time customer churn prediction via federated learning for privacy-preserving and optimized marketing decision. *Eur J Comput Sci Inform Technol* 13: 32–39. https://doi.org/10.37745/ejcsit.2013

Pekar A, Makara LA, Biczok G (2024) Incremental federated learning for traffic flow classification in heterogeneous data scenarios. *Neural Comput Applic* 36: 20401–20424. https://doi.org/10.1007/s00521-024-10281-4

Rodan A, Fayyoumi A, Faris H, et al. (2014) Negative correlation learning for customer churn prediction: A comparison study. *Sci World J* 7. https://doi.org/10.1155/2015/473283

Sheller MJ, Edwards B, Reina GA, et al. (2020) Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data. *Sci Rep* 10: 12598. https://doi.org/10.1038/s41598-020-69250-1

Shi N, Lai F, Al Kontar R, et al. (2021) Fed-ensemble: Improving generalization through model ensembling in federated learning. *J Big Data* 6: 28. https://doi.org/10.1186/s40537-019-0191-6

Sikri A, Jameel R, Idrees SM, et al. (2024) Enhancing customer retention in telecom industry with machine learning driven churn prediction. *Sci Rep* 14: 13097. https://doi.org/10.1038/s41598-024-63750-0

Wu Z, He T, Sun S, et al. (2024) Federated class-incremental learning with self-distillation (fedclass). *arXiv*. https://doi.org/10.48550/arXiv.2401.00622

Yang X, Feng Y, Fang W, et al. (2020) An accuracy-lossless perturbation method for defending privacy attacks in federated learning. *WWW '22: Proceedings of the ACM Web Conference 2022*, 732–742. https://doi.org/10.1145/3485447.3512233