Enhancing Customer Churn Prediction in Telecom Using Federated Learning

MSc Research Project
MSc in Artificial Intelligence

MURUGAPPAN KRISHNAN
Student ID: 23187905

School of Computing
National College of Ireland

Supervisor: ABDUL SHAHID

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	MURUGAPPAN KRISHNAN
Student ID:	23187905
Programme:	MSc in Artificial Intelligence
Year:	2024
Module:	MSc Research Project
Supervisor:	ABDUL SHAHID
Submission Due Date:	12/12/2024
Project Title:	Enhancing Customer Churn Prediction in Telecom Using Federated Learning
Word Count:	8012
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	MURUGAPPAN KRISHNAN
Date:	12.12.2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Customer Churn Prediction in Telecom Using Federated Learning

MURUGAPPAN KRISHNAN
23187905

Abstract

Customer Churn is one of the major problems within the field of telecommunications.It directly affecting profits and the strategy undertaken for user retention. Most of the conventional techniques for predicting customer churn require sharing private information about customers.This introduces a significant risk of data leakage and violation of regulations. The study proposes an alternative approach to better predict churning with data privacy based on Federated Learning. In other words, FL lets you directly train machine learning models on source data in different places, one would keep a regional customer database up to date with sensitive customer data, since only model parameters would change and not the raw data itself.The study tells about how to use a Simple NN in a Federated Learning setup, with datasets spread out over three geographical nodes that focus on different types of customers in Mumbai, Ohio, and Ireland. It has batch normalisation for stability and ReLU activation for non-linearity, making it best for predicting churn in a binary way. The FedAvg algorithm will be used to combine the local model changes into a single global model over and over again during the training process.Key performance measures, like F1-score, accuracy, precision, and recall, are used to show how well the model works and how close it is to being perfect.

My findings show that the suggested framework is strong and improves the accuracy of churn prediction by a large amount, with an F1-score of 83.45 and an accuracy of 84.40 % is the highest among seven attempts. It shows that federated learning works for training high-performing models together across decentralised data while still protecting privacy. This study shows that FL can be used on a large scale and in real-life telecom settings. It also leads the way in customer data with privacy protection. For future work, the combination of automation with advanced methods of grouping is also in thought. This will make the model even better at what it does and make it more efficient.

1 Introduction

Customer churn prediction is one of the most important studies in the telecom industry, as it directly influences the strategies to be adopted for customer retention, sales growth, and thereby the sustenance of the business. Churn refers to the complete cessation of the use of service from a particular service provider. It is one of the biggest problems for a telecom business. This would allow telecom service providers to deploy customer retention strategies with the development of targeted offerings and promotions focused on preserving acquired customers by reducing the loss and retaining profit-maximizing.[3]

Traditional Churn prediction method includes the processes of centralized machine learning model data from disparate sources to centrally collected data store. The mentioned here are all traditional methods to predict churn. Immediately, very important privacy concerns arise, even more so now that GDPR is in place. In order to protect privacy, GDPR also puts strict rules on how data can be collected, stored, and processed. As customer churn records get more complicated, large, and different, they don't fit well with centralized models. Obviously, churn datasets are big, have a lot of dimensions, and are not IID. They have shown variations across regions, demographics, and service usage trends. Federated learning solves all these issues. [15]

Opposed to the methods which centralize data, FL trains machine learning models locally in a decentralized manner. Because it sends model weights only to the central computer at every step, never the raw data. This ensures data privacy. Finally, FedAvg is used to combine changes from different nodes into a single global model. This makes it possible to learn from even these spread-out datasets. This kind of method will be very decentralised, so it will be in line with privacy laws. At the same time, it will allow a generalisation that applies to a wide range of customers. In this way, it allows finding customer churn with quite a great robust and scalable way to be provided.[9]

1.1 Motivation

Competition in the market is increasing day by day. That means retaining customers is equivalent to getting more and more. Not keeping a customer does not only mean just that money loss but also is lost which was used in seeking new users and retaining current customers. This will definitely raise the demand and significance of predicting customer churn to prevent subscribers from leaving. Even though these works are helpful, traditional ways of predicting customer churn have had to deal with two problems: keeping data private and dealing with differences. [12]

GDPR increased the sensitivity of people toward privacy issues, and as such, it has imposed very strict rules about the collection and use of personal data. The centralized models put all data of a customer at one place. This increases the chances of private data theft. Yes, that does bring a lot of legal and moral problems. The most important thing is that standard models are not designed to work with datasets that are not IID. The behavior of customers can be very different in different areas and among different groups of people. This makes it highly unlikely that one model will learn correctly from those kinds of different trends. [15]

That is where Federated Learning comes in, fixing the issue by taking care of privacy, hence fixing the issue caused by the non-IID data. With this in mind, FL will train the models locally in the nearby nodes. In this way, FL keeps the information secret and follows the rules. Weighted aggregation, in turn, is present to make updates from a larger dataset contribute much in updating the global model as it always tends to give high weight to more accurate and generalized updates. Considering the existing diversity, the complexity of data, and the scalability needs of today's state of the telecommunication industry, this architecture does appear to balance out the growing requirements with a balance between scalability and privacy.[6]

1.2 Research Question

The research question, therefore, would be to determine how the shortcomings highlighted above from Federated Learning,issues of privacy, heterogeneity, and scalability—can be overcome in the customer churn prediction problem.

The specific research question is:
To what extent does Federated Learning improve the performance of the telecommunication customer churn prediction model in predictive accuracy and generalisability while providing privacy?

Based on the research question above, the following sets of specific study objectives are given:

- The most up-to-date knowledge on standard, advanced, and privacy-protecting models for predicting customer churn.
- Make a Federated Learning system that contains a privacy-protecting feature and can handle issues with the non-IID dataset.
- Deploy the planned FL framework on cloud-based infrastructure while taking care of the simulation made for distributed datasets coming from a problem, say, customer churn.
- Test the effectiveness of the suggested framework with a set of important metrics such as accuracy, recall, precision, F1-score, etc., and compare it with traditional methods.

1.3 Contribution

The key contribution of this work is the complete development of a scalable Federated Learning system that protects privacy and can be used for customer churn prediction. The proposed framework is different from the traditional model, which trains the model centrally with gathered data. This will instead train the model in a decentralized manner on local data sets, aligned with GDPR rules, protecting the privacy of raw customer data.[9]

This framework adds something new to the process of making a weighted aggregation plan to deal with the problems that non-IID data causes. It is possible to give more weight to updates from bigger or more representative datasets when trying to make the global model better and boost accuracy and generalisation. In addition, it suggests a neural network design that is optimised for light computation. The proposed method has utilized batch normalisation and ReLU for accelerated convergence to a stable training that can be afforded in a resource-constrained setting. This could be scaled up and put into a real-world implementation by placing the framework on the cloud and making use of services like AWS S3 for spread storage and AWS EC2 to simulate nodes across different locations.[2]

By applying the given solution, it would better serve the business goals and privacy and different types of data concerns. It will further help in conceptualizing the framework that will plan ways to retain customers longer and reduce the number of people who leave their service, thereby making any telecom company in the world more money.[12]

2 Related Work

Predicting customer churn has become one of the most important areas of study in telecommunications because it has a direct effect on keeping customers and making money.It means designing a retention plan for a group. Churn refers to people stopping the use of any service, which is detrimental to business as that costs money. If early detection were possible, telecom companies would jump at this opportunity to avoid such possible losses by implementing a good retention plan. This would help them increase their general gains by making more money.

Ways of predicting churn have been different over the years,from simple models to advanced machine learning models and finally privacy-protecting frameworks like FL. Examples of traditional models include Decision Trees and Logistic Regression. First of all, they are simple to interpret. Consequentially, after those, more complicated big datasets required more advanced ways of modeling. Meanwhile, the emergence of some strict rules such as General Data Protection Regulation also made handling personal data more difficult. As time went on, such neural networks, XGBoost, and ensemble learning were far more powerful in making predictions but couldn't be interpreted as well and required a lot of computational power.

Federated Learning has flipped that around now, allowing predictive models to be trained across distributed data. While the models in the olden days depended on centralized models based on aggregated data, FL had their training done locally inside the model for protection of data privacy-sort of conformation to rules on privacy. The review on shifting methods in predicting churn identifies the main contributing authors, most important gaps in research, and a reason why the review is required in this context. [3]

2.1 Traditional Models for Churn Prediction

The most prevalent models in loss prediction are Choice Trees and Logistic Regression. Both have become so widespread because they are intuitively easy to understand and implement in a very short time. Gaur and Dubey (2018) stated that the foremost strengths of Decision Trees are the clear, rule-based ways they make decisions, which are understandable by every party concerned.These are more interpretable models by design; the insights from them are easier to glean out and understand by decision makers. That's why a specific churning prediction was made on an instance.The Author Kohavi 1995 explicitly discussed how Decision Trees could do significantly better by leveraging different cross-validation methods for avoidance of overfitting issues while increasing generalisation.[7]

Another important well-known model for performing churn prediction is Logistic Regression. It models the probability of churn as a linear relationship between the input data and the output label. Patel and Kumar 2023 demonstrated that logistic regression works well for tasks that can only be put into two groups, like the prediction of customer churn, so long as the telecom provider can figure out which customer characteristics drive the most impact on the churn. Logistic Regression is very easy to comprehend and operate. However, its major flaw is assuming that the features of complicated data are linear when they are not. Traditional models are abundant but also have a lot of problems. In the presence of large or noisy datasets, Decision Trees tend to overfit. Logistic Regression, on the other hand, assumes linearity and doesn't work well when variables combine in nonlinear ways, like when a customer acts in a certain way. For telecom, it became clear that standard models couldn't handle the growing size, number of dimensions, and complexity of the dataset. This meant that Advanced Machine Learning models had to be used.[12]

2.2 Advanced Machine Learning Models

Classical models began to suffer with highly informative tasks. The reason for this was the constant growth and complication of customers. It was finally time for advanced machine learning modeling to come out. It uses neural networks, XGBoost, and ensemble learning to train large amounts of high-dimensional data by finding nonlinear links between features, which leads to better performance, flexibility, and adaptability.
The Author Hu et al. came up with a method in 2020 that combined the strengths of Neural Networks and Decision Trees. While the first model occluded vision into the nonlinear trends in data, the second model was highly pivotal for drawing any meaningful inference from data. Later on, when combined, Hu et al.'s model emerged as a better predictor while still retaining a part of its interpretability.[4]
The Author Tang in 2020, where he blended K-means clustering with XGBoost. During this process, the customer data split into groups of patterns or characteristics. Then the XGBoost models, trained on each of those respective groups of customer data, were presented. This kind of segmentation will enable estimates to be much more relevant for each set of customers. XGBoost is a sequence of decision trees, and each tree is greedily constructed one after another. It gives better prediction accuracy compared to the stand-alone models.[14]
In the year 2024,the Author Senthilselvi et al. proposed a method using Ensemble Learning that includes Random Forests, Gradient Boosting, and XGBoost. Generally speaking, ensembling is a way to combine the best parts of several algorithms to make a strong model that works well in most situations, even when customer data isn't all the same.[13]

2.3 Privacy-Preserving Framework: Federated Learning (FL)

With increased awareness about privacy issues and with new laws on privacy coming into place such as the General Data Protection Regulation, more people would want to learn in a manner that will not expose private information.The Author McMahan et al. (2017) propose Federated Learning as a way to achieve this. This approach will allow training predictive models in a decentralized manner, and raw customer data will never have to leave the devices being used. FL eliminates the need to pool data in one location, reducing the likelihood of a breach in privacy.[9]
The Author Bonawitz et al. (2019) suggested that model changes sent between devices and the server should be encrypted, making privacy even better, hence FL could be used in large-scale settings. Wang et al. (2023) used FL to predict telecom churn and showed that FL models work just as well as centralised models while still protecting customer data.[2]

2.4 Handling Non-IID Data in Federated Learning

The most critical problem in FL is Non-IID record problems.The Author Wang et al. (2023) demonstrate that non-IID data changes the convergence rate since updates are more likely to be focused on local datasets, which makes the model less general globally. Finally,the Author Bonawitz et al. (2019) came up with the idea of heterogeneity-aware aggregation. This means that clients holding larger datasets or more varied samples add more to the global model.[16]
The Author Tang gave one such in 2020. It is called "client clustering"; clients with similar distributions are put together to train personal local models.This will certainly work very well in telecom churn forecast because customers in different areas or demographics will have different usage habits.[14]

2.5 Addressing Computational and Communication Costs

Federated Learning is difficult to carry out, since the computation and sharing of resources in edge devices are relatively poor. Azzam et al. (2023) discussed using "lightweight" neural networks, containing fewer trainable parameters, that therefore require less computational resource.[1]The Author Hu et al. (2020) encouraged making light models requiring low memory and computational power. FL would therefore be implementable on smartphones or other edge devices.[5]

2.6 Research Gaps

The literature review of the existing works on Federated Learning for telecom loss prediction reveals some knowledge gaps that are critical.

Handling Non-IID Data: FedAvg considers all client updates of equal importance, which fails to work effectively in scenarios when clients have different datasets.[9]

Computational and Communication Overheads: Edge devices do not possess significant computational power. The large FL models would require considerable bandwidth.

Privacy and Security: Even with a safe aggregation protocol in place, it might not be impossible to avoid the inference attacks. Additional methods may be necessary to further protect privacy.[2]

Generalization Across Customer Segments: Still a difficult task to get the model to work across all customer groups.

2.7 My Contribution

It contributes to the literature by providing a new FL framework for customer churn prediction in the telecom business. The new research is more comprehensive since previous studies dealt with only one problem, such as either privacy or computation cost. It addresses four major issues: privacy, handling Non-IID data, computational efficiency, and scalability. What makes this framework so unique is solving various problems of the telecom provider while adhering to guidelines on data such as the GDPR.

So, the Framework Suggested for FL Basically train ML models across multiple nodes. For instance, every node can represent Data Center, device, or the customer-premise equipment from a different region. These models are centralized and require all raw customer data from various sources to be aggregated to a single central location. Due to this, the raw customer information shall never leave the local devices. The design follows privacy rules, making it less likely for privacy to be breached, hence building trust between customers and telecom companies.

Another major development, concerning this study, had to do with design for customer privacy-preserving churn prediction. A company will be fully aligned with the privacy laws such as the general data protection regulation by ensuring that the customers' raw data does not leave their local devices. Also, the suggested model for parallel architecture is not necessarily sending its raw data to some Central Server, instead it sends model weights resulting from local training to a central server. This will not let private information about the customer spread out and thus making a privacy breach less liable to happen.

During training, this FL-based system will grant security and decentralisation. In contrast to traditional models, it will not collect and combine customer data to store on a main server. Based on the concept behind the philosophy of the privacy-by-design approach, the proposed protection plan for privacy will enable telecom service providers to safely use predictive churn models without violating any existing regulations.

Following are the advantages of this approach towards privacy protection:

- **Legal:** The proposed framework is fully compliant with GDPR and other laws on data protection.
- **Privacy risk reduction:** In this model, the raw data never leaves the local nodes. This makes it difficult for an attacker to access the data.
- **Customer Confidence and Trust:** The suggested structure will make clients feel that at least their own information is secured, and this could imply greater loyalty to brands and reduced losses of customers.

2.7.1 Dealing with Non-IID Data

Federated Learning has its host of issues, but among the weirdest perhaps would be how to handle non-independently and identically distributed data across various participants. In other words, a centralized model considers there being only one way data can be distributed; whereas FL does take into consideration many nodes or clients that always behave differently and use the system differently. As a matter of fact, clients coming from different areas, different age groups, and other demographic groups act uniquely their own way in the field of telecommunication.

The proposed study takes this problem and comes up with the Weighted Aggregation Strategy. Traditional FL frameworks always give each update to nodes of equal weight, such as the FedAvg algorithm proposed by McCahan et al. in 2017. In reality, however, some of the nodes may be in possession of larger datasets or have more varieties of contact with customers. The proposed Weighted Aggregation Strategy weighs higher for the node updates based on their number or difference type, directly showing that useful updates have greater effects on the final global model.

2.7.2 Lightweight Neural Networks

Another important contribution this work provides is Lightweight Neural Networks that can be used to solve the problem of how to make Federated Learning more efficient in computing. Unlike most traditional models, usually deep neural networks with millions of parameters, the model suggested in this paper is small and very good at computing, so it can be directly trained at the edge or nodes with limited resources.

The following paper deals with one of the most fundamental issues at play in telecom settings—from routers to smartphones and even regional servers, which are not that powerful. Most of these devices can not learn large and complex neural networks because they cannot afford it due to the shortage of processing power, battery, or storage. This will train lightweight neural networks quickly and effectively locally and let devices that have little memory and computational powers join the FL process as soon as they are added.

2.7.3 Unified Framework

This paper will be discussing a unified framework of FL regarding privacy preservation and time consumption of computations. The proposed unified framework mitigates previous model failures in FL-based churn forecast.If a study is well-designed, it is in a position to solve issues related to privacy, scalability, computational efficiency, and problems resulting from non-IID data. The paper provides an efficient real-world solution for telecom churn prediction while ensuring the solution is GDPR-compliant, enabling fast training on resource-limited devices, and giving greater weights to updates received from important nodes.This will be a big step forward for the federated churn prediction model, which will make the telecoms industry more scalable, general, and well-protected in terms of privacy.

Author and Year	**Related Work**	**Approach**	**Strengths**	**Limitations**
Gaur and Dubey (2018)	Rule-based Churn Estimation	Decision Trees	Interpretability	Overfitting
Kohavi (1995)	Model Evaluation	Cross-validation	Reliability	Inability to handle complex data
Patel and Kumar (2023)	Churn Classification	Logistic Regression	Feature Impact	Linearity Constraint
Hu et al. (2020)	Hybrid Models	Neural Networks, Decision Trees	Non-linear Learning	High Cost, Low Interpretability
Tang (2020)	Personalized Forecasting	K-means, XGBoost	Accuracy	High Computation Cost
Senthilselvi et al. (2024)	Ensemble Method	Fusion Algorithms	Robustness	Lack of Transparency
McCahan et al. (2017)	FL for Privacy	Federated Learning	Privacy Preservation	Non-IID Issue
Bonawitz et al. (2019)	Secure Aggregation	Encrypted Model Updates	Privacy	High Computational Cost
Wang et al. (2023)	FL for Churn Prediction	FL with Non-IID Data	High Performance	High Edge Device Requirement
This Study (2024)	FL for Churn Prediction	Weighted Aggregation	GDPR Compliance, Scalability	Real-World Evaluation

Table 1: Summary of Literature Review Approaches, Related Work, Strengths, and Limitations

3 Methodology

3.1 Overview

Customer churn prediction in the telecom industry requires a robust, scalable framework that ensures data privacy and heterogeneity in customer behavior with high predictive accuracy. This work considers the application of Federated Learning (FL), a decentralized machine learning approach where a given model is trained on several distributed datasets without sharing the raw data. Instead, only model parameters or updates are shared among local nodes and a central server. Ensuring this would make the model perform better with regional datasets diversity while following data privacy regulation such as GDPR.

This approach has followed the mechanism through which, in training, all the updates generated are stored centrally on the `fl-central-bucket` on AWS S3 for centralized access and control. Furthermore, aggregated updates are shared to enhance local models. Across the regional buckets not only makes the framework adaptable but collaborative, too. The training, aggregation, and re-training of the models is an iterative process that goes on until the global model converges to an optimum solution.[9]

3.2 Design of Research

In order to provide efficient processing of geo-distributed data, the architecture was designed based on AWS cloud services. In this vein, the corresponding areas will be trained using raw customer data kept in S3 buckets representing Ohio, Ireland, and Mumbai.

Instead of being kept in the local regions, model updates that are acquired through training on local regions are kept in the centralized repository `fl-central-bucket`. This makes managing and aggregating updates considerably easier and ensures that no updates are lost. [2]

- Mumbai updates: `fl-central-bucket/model/v1/updates`
- Ireland updates: `fl-central-bucket/model/v1/updates`
- Ohio updates: `fl-central-bucket/model/v1/updates`

Two locations will aggregate updates:

1. **Central Bucket:** `fl-central-bucket/model/v1/aggregates` for centralized access.
2. **Regional Buckets:** `flohio-bucket/model/v1/aggregates`, `flireland-bucket/model/v1/aggregates`, and `flmumbai-bucket/model/v1/aggregates` for local retraining.

Due to this dual storage technique, in successive training cycles, the combined global model is assured to be consistent across all nodes.

3.3 Flow of Implementation

Local datasets stored in S3 buckets for corresponding areas are used to initiate and train the models. Because they can learn complex patterns found in high-dimensional data, neural networks have been selected to handle nonlinear relationships. For example, neural networks would recognize the intricate relations between customer churn and service preferences, consumption patterns, and demographics.[13]

Neural Network Architecture:

- **Batch Normalization:** This resolves issues such as vanishing or exploding gradients by normalizing the input to each layer for stability during training.
- **ReLU Activation:** With non-linearity added, ReLU activation allows the model to pick up on complex patterns in consumer behavior.
- **Binary Classification Loss:** Since the goal variable is binary, the network is optimized on Binary Classification Loss, or CrossEntropy.

Phases of Training:

1. **Local Training:** The various scripts, including `mumbaitraining.py`, `irelandtraining.py`, and `ohiotraining.py`, do an initial training on their respective individual datasets. Model weights are uploaded to the central bucket at `fl-central-bucket/model/v1/updates`.
2. **Aggregation Updates:** Updates are collected in the central bucket by the `fedavg.py` script. The weighted average calculation of FedAvg aims to enhance the global model.
3. **Aggregated Update Distribution:** Once the global model aggregate is persisted in `fl-central-bucket/model/v1/aggregates`, it is propagated back to regional buckets. With these updates, the local models gain global knowledge.
4. **Retraining:** The scripts such as `mumbai_retrain.py`, `ireland_retrain.py`, and `ohio_retrain.py` use aggregated updates to retrain their local models. `model/v2/updates` is kept in a place for new updates.
5. **Iterative Refining:** This process is further enhanced iteratively until the various metrics for performance evaluation—accuracy, precision, recall, and F1-score converge to certain acceptance values.[4]

3.4 Why Neural Networks?

Neural networks were chosen because they can handle large-scale, high-dimensional nonlinear datasets with exceptional ability. Other conventional models, such as logistic regression or decision trees, cannot learn complex linkages and hidden patterns in customer behavior.[4]

Benefits of Neural Networks:

- **Nonlinear Relationship Learning:** Learns non-linear relationships between connected traits like demography, service preference, and consumption patterns.
- **Flexibility and Scalability:** By modifying layers, neurons, and activation functions, it can adjust to different levels of complexity.

- **Automatic Feature Extraction:** For telecom datasets with hundreds of features, this minimizes the need for human pre-processing.
- **Managing Data Noise and Imbalance:** Focuses on deeper data structures while being resilient to noisy and unbalanced datasets.

Code Components:

- **Input Layer:** Manages user characteristics, including demographics and usage data.
- **Hidden Layers:** Batch normalization and ReLU for stability and nonlinear learning.
- **Output Layer:** Sigmoid activation has been used for binary classification—churn or no churn.
- **Loss Function:** Binary CrossEntropy optimizes the network by reducing prediction errors.
- **Adam Optimizer:** For quick convergence and decent performance when sparse gradients appear.[9]

3.5 Benefits of the Suggested Method

- **Privacy Compliance:** Fully complies with GDPR because raw data remains on their local sites while exchanging model updates.[15]
- **Scalability:** It's simple to incorporate new datasets or regions into the workflow.
- **Better Generalization:** The FedAvg method allows global models to learn from a variety of regional datasets, enhancing generalization over unknown data.
- **Computational Efficiency:** Due to the fewer resources that lightweight neural networks require, this framework can be applied to a real-world telecommunications setting.[1]

Phase	Scripts and Description
Training Phase	`mumbaitraining.py`: Mumbai node training. `irelandtraining.py`: Ireland node training. `ohiotraining.py`: Ohio node training. `federatedavg.py`: Aggregating weights.
Retraining Phase	`mumbairetrain.py`: Retraining on Mumbai node. `irelandretrain.py`: Retraining on Ireland node. `ohioretrain.py`: Retraining on Ohio node. `updatedavg.py`: Updating global model.
Iterative Rounds	`mumbairounds.py`: Mumbai node iterations. `irelandrounds.py`: Ireland node iterations. `ohiorounds.py`: Ohio node iterations. `aggregatesiteration.py`: Aggregating iterations.
Metrics Evaluation	`calculate.py`: evaluating metrics for all the iterations.

Table 2: Workflow of the Federated Learning Process

4 FL Architecture

The design describes an iterative FL framework in terms of customer's churn predictions. After several rounds from regional nodes participation like; Mumbai, Ireland and Ohio Participating in an iterative Fashion An FL round consists of several iterations that together produce a considerably more accurate model for a prediction while the data of the customers shall be retained secret. This iterative mechanism would not only allow the system to slowly approach a globally optimal model but also learn from the data something new in every area and apply what was learned to a wide range of customers.
When this version gets as accurate as it can get, the final global model is used in the real world. Using the information gained through repeated improvements, the model can now make very accurate predictions about how many customers will leave in all areas. This makes sure that telecom companies are one step ahead of dealing with customers who leave, which increases their profits because they keep more customers.[6]

4.1 Initial Training and Update Sharing

Each regional node boots up and trains its own model on the regional data set residing in the local S3 bucket, such as flmumbai-bucket. A neural network architecture is used for training, as implemented in the training scripts mumbaitraining.py, irelandtraining.py, and ohiotraining.py. These scripts use batch normalisation to make training stable, ReLU activation to add non-linearity, and binary cross-entropy loss to improve estimates of binary churn.
Once the initial training is complete, locally learned models make updates; these are the fine-tuned weights of the neural network. These changes will be safely sent to the central bucket at fl-central-bucket/model/v1/updates rather than the raw data. Putting all the efforts in one place from different areas so that it can be processed further.[9]

4.2 Aggregation Using FedAvg

It uses an algorithm called Federated Averaging (FedAvg), part of fedavg.py, to collect changes. This script determines a weighted average of the updates, based on how much of the dataset each area makes up. For example, if the dataset for Mumbai is bigger than those for Ireland or Ohio, the changes from Mumbai are given more weight when the datasets are put together. This weighting system ensures that the areas that have a high volume of data contribute more to the world model, enabling it to generalize better. The changes aggregated in the central bucket are stored at model/v1/aggregates.Updates are also propagated to the respective area buckets at flmumbai-bucket/model/v1/aggregates, flireland-bucket/model/v1/aggregates, and flohio-bucket/model/v1/aggregates. This two-tiered storage method ensures that all nodes have access to the global model while still being able to add their own local data to the model during training.[16]

4.3 Retraining and Iterative Refinement

Once all the updates are collected, each region retrains its own model using the world information it gets from the central bucket. There are tools like mumbai retrain.py, ireland retrain.py, and ohio retrain.py that do this retraining. The retrained models make a new set of updates and post them to the central bucket as model/v2/updates. This ends one iteration. The iteration method is done more than once, with each subsequent improvement being made to the global model. This process is automated through such scripts as mumbairounds.py, irelandrounds.py, and ohiorounds.py. Each area always keeps modifying its models, with the world model helping the areas learn new things. Models get better with iterations to make precise predictions across a wider swath of demographics and service usage trends.[8]

4.4 Convergence and Performance Metrics

These depend upon key performance measures such as accuracy, precision, recall, and F1-score during the process until global convergence occurs. These are checked following every iteration to see how it may become better at making the prediction. For example, for some model, after the first version, it might give accuracy at 75%. The model might get better in iterative cycles until it is always greater than 90%. It's also important to think about how complicated the datasets and connections being modelled are. In telecom churn forecast, the iterative process makes sure that the global model learns how to deal with the different and nonlinear features of regional data sets while protecting the privacy of customer data.[11]

4.5 Benefits of the Iterative Approach

Improved Generalization: The iterative process allows the global model to learn from the differences between regional datasets, making it better at handling data it hasn't seen before.
Scalability: The proposed framework is very flexible and can handle the addition of new regions or nodes without disrupting the process at any given time. New nodes can join the back-and-forth process with their changes.
Preserving Privacy: The iterative process totally follows the course of privacy regulations like GDPR because it exchanges only model changes, not raw data.
Adaptability: Repeated processes are lightweight and therefore computer-efficient due to lightweight neural networks, hence workable on resource-constrained devices.[16]

The iterative framework has easy ways to handle distribution changes in data or local trends. This keeps the model strong and useful over a period of time.This is an iterative design that ensures the global model keeps improving with each round while maintaining the privacy and security of regional datasets. With each iteration, new regional insights are added to make the solution fast, scalable, and privacy-friendly. Hence, this is ideal for predicting customer churn in the telecom business.[2]

4.6 Cloud Infrastructure

The AWS ecosystem forms the backbone of this FL framework, ensuring seamless integration, scalability, and security.

AWS Resource	Purpose
`fl-central-bucket`	Central bucket for storing updates and aggregated global models.
`flmumbai-bucket`	Bucket for Mumbai, initially storing raw data and later receiving aggregated updates.
`flireland-bucket`	Bucket for Ireland, initially storing raw data and later receiving aggregated updates.
`flohio-bucket`	Bucket for Ohio, initially storing raw data and later receiving aggregated updates.
EC2 Instances	Execute local training and retraining processes for each region. Connected to respective S3 buckets.

Table 3: AWS Resources and Their Purpose

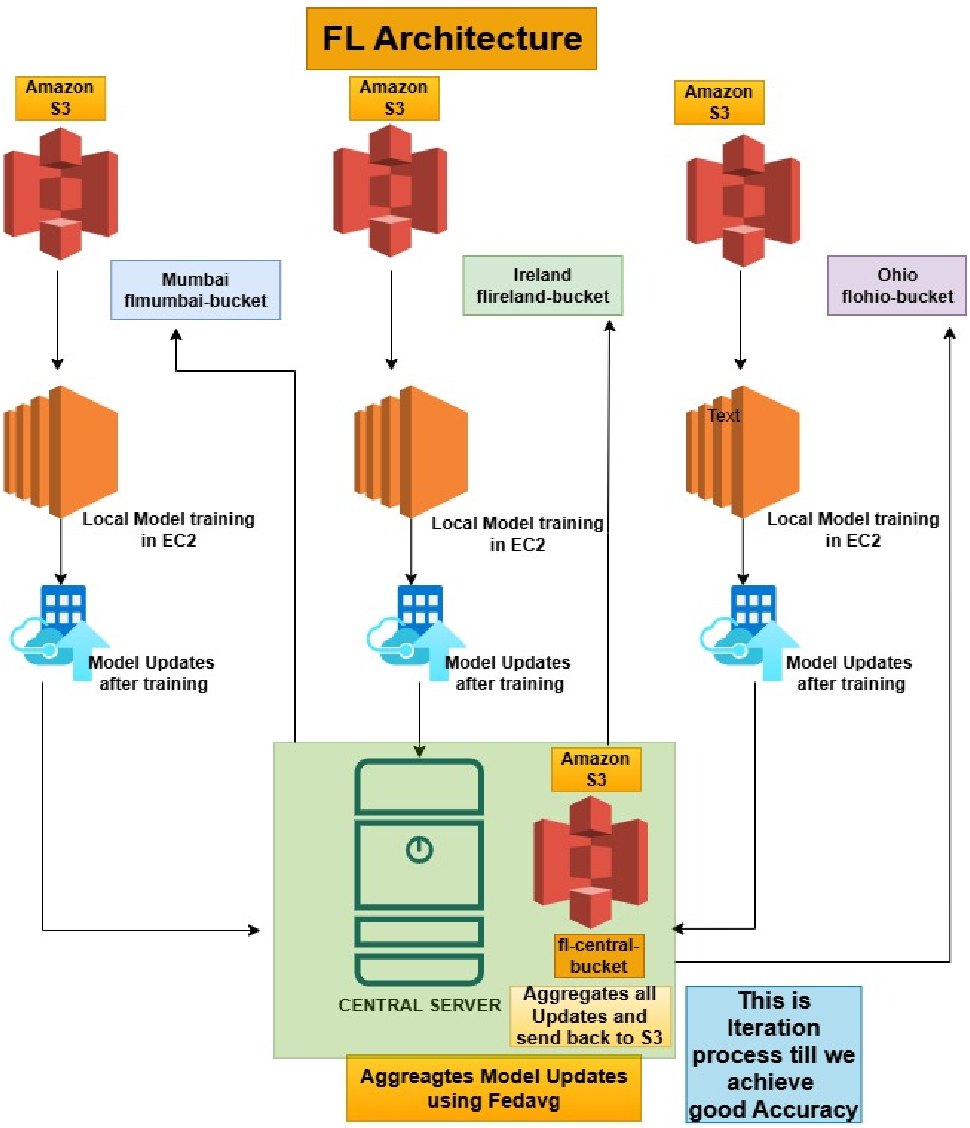

Figure 1: FL Architecture for Federated Learning Model Training and Aggregation

5 Implementation

To use federated learning to solve problems like predicting customer churn, many different parts need to work together to create a machine learning model that can be used in many situations, is flexible, and protects privacy. In addition, a list of steps is made that shows how to train models, combine them, retrain them, make iterative improvements, and finally, how these processes can be supported on cloud-based infrastructures. This would be done with an implementation based on Python code. People use AWS Cloud Services to store storage, do computations, and train models in a distributed way on top of the computation fabric. In general, it is set up so that each process keeps customer information private, taking into account privacy laws like GDPR and others.

5.1 Model Training and Sharing the First Update

First, the local models for each of the three regional nodes, Mumbai, Ireland, and Ohio, need to be trained as part of the application. Each region is a node in itself, and it can store raw user data in AWS S3 buckets such as `flmumbai-bucket`, `flireland-bucket`, and `flohio-bucket`. The training scripts, `mumbaitraining.py`, `irelandtraining.py`, and `ohiotraining.py`, are configured to initiate the training at these nodes. These models are neural networks set up with batch normalization to keep the learning process stable, ReLU activation to add nonlinearity, and a binary cross-entropy loss function to find the best solution for churn or no churn in binary classification. It trains the model using only local data at each regional node without having to share raw customer data, hence following the GDPR rules.

At the end of the training, each node updates the model instead of updating the raw data. Essentially, these updates are model weights and parameters. From there, model changes are sent safely to a central bucket in AWS S3 called `fl-central-bucket/model/v1/updates`. The centrally stored updates from all three places are used to aggregate, which means obtaining a single global model by aggregating model updates coming from the three places. There is no direct sharing of raw data between the nodes and the central server, thus not violating the privacy of any customer.[10]

5.2 FedAvg Aggregation

As soon as the updates from Mumbai, Ireland, and Ohio are uploaded to `fl-central-bucket`, they will be combined into a global model.

The **Federated Averaging (FedAvg)** method available within the `federatedavg.py` script was utilized. The FedAvg algorithm computes the weighted average of the model parameters which arrive from all nodes within an area. That is, the weight of the aggregation is assigned to each place according to the size and quality of the data set. For example, if the node in Mumbai contains data that is more important compared to nodes in Ireland and Ohio, then its changes will be considered more when summing up the data. This step sums everything together into one global model that incorporates knowledge from the three area nodes. This model is kept in two places:

1. **Central bucket:** `fl-central-bucket/model/v1/aggregates` — so that you can access the world model from one place.

2. **Regional buckets:**

   - `flmumbai-bucket/model/v1/aggregates`
   - `flireland-bucket/model/v1/aggregates`
   - `flohio-bucket/model/v1/aggregates`

Each node will add the collected knowledge to its own local model.These changes are made in groups based on regions to make sure that every node has the most up-to-date global model so that more retraining rounds can happen. As a result of shared learning settings, this structure is repeated on both the node and server sides. This is because all the nodes are consistent and out of sync with each other.[9]

5.3 Retraining Local Models with Aggregated Updates

Once the world model has been assembled, it needs to be pushed back to the nodes. In this way, each node can make use of what the other nodes have collectively learned. The tools `mumbai_retrain.py`, `ireland_retrain.py`, and `ohio_retrain.py` address this. These tools download updated models from the regional buckets (`flmumbai-bucket`, `flireland-bucket`, and `flohio-bucket`) and update the local models.[2]

It gets the total weights and then changes the model parameters locally. This is an important step because it allows each node to customize global knowledge with some key features of its own local data set. This makes sure that the model will be much better able to adapt to the demographics and usage patterns of customers in each area. The new weights and values are saved on the central bucket as `model/v2/updates`. All the above-mentioned things are repeated again and again. Every step will definitely ensure a rise in accuracy and generalisation as information is being taken privately at all nodes.[6]

5.4 Iterative Refinement of the Global Model

The most important part of the execution is the iterative refinement. The world model gets better with each round of aggregation, retraining, and sharing of updates. Each iteration starts with the nodes either training their local models with new data or retraining them with the most recent aggregated updates that were uploaded to `fl-central-bucket/model/v2/updates`. The FedAvg method is then used to do the aggregation again.[10]

These are tools that have been written to keep this process going—`mumbai_rounds.py`, `ireland_rounds.py`, and `ohio_rounds.py`, all running multiple rounds. There is one full loop of local training, aggregation, distribution, and retraining in each round. The accuracy, precision, recall, and F1-score for every run of the global model will be tracked and written down. Once the level of convergence is good enough, this repeated process can be stopped, and the final global model can be used in the real world.

A process called iteration refinement makes it possible for more generalisation and stability, meaning the models are more accurate. That is because each round includes all that was learned in the rounds before. As such, the model would know different trends describing how customers behave in different areas.[13]

5.5 Evaluation and Performance Metrics

The script `calculate.py` will calculate some useful variables next, such as **accuracy**, **precision**, **recall**, and the **F1-score**, all to demonstrate the performance of the global model. From these measures, one may infer how well the model is performing its tasks of customer loss prediction. If done after each round of training, it would also enable comparison with previous rounds. It would be good if the model showed that these measures improved over time, which would be indicative of convergence.

Furthermore, comparisons are made between the successful performances of the federated model and traditional, centralized models. Against this, a federated model with comparable, and in many cases better, performance while considering privacy protection is presented when set against centralized models. After conducting 7 iterations, the best performance was achieved in iteration 5 with an accuracy of 82%. This iterative process can be stopped once the performance metrics reach an acceptable level, and the final global model can then be used in an active manner.[8]

5.6 Script Workflow

- **Training Phase:**
  - `mumbaitraining.py`
  - `irelandtraining.py`
  - `ohiotraining.py`
  - `federatedavg.py`
- **Retraining Phase:**
  - `mumbairetrain.py`
  - `irelandretrain.py`
  - `ohioretrain.py`
  - `updatedavg.py`
- **Iterative Rounds:**
  - `mumbairounds.py`
  - `irelandrounds.py`
  - `ohiorounds.py`
  - `aggregatesiteration.py`
- **Metrics Evaluation:**
  - `calculate.py`

Example Script Execution

- Connect to the Mumbai server:
  - `ssh -i "C:\Users\murug\Downloads\mumbaiserver.pem" ec2-user@43.205.230.72`
- Export session tokens:
  - `export AWS_ACCESS_KEY_ID="your_key"`
  - `export AWS_SECRET_ACCESS_KEY="your_secret_key"`
  - `export AWS_SESSION_TOKEN="your_token"`
- **Note:** The session token is valid for 12 hours. Generate a new token after expiration.

Run the training script:

- `python mumbaitraining.py`

Run the retraining script:

- `python mumbai_retrain.py`

Repeat for Ireland and Ohio servers by connecting with their respective SSH commands and running the appropriate scripts.

5.7 Important Notes

- Always connect to the server using SSH before running any script.
- Source the AWS session token after connecting to ensure access to S3 buckets.
- The session token is valid for 12 hours and must be regenerated after expiration.
- Follow this sequence: SSH login → Source token → Run scripts (`training.py`, `retraining.py`, etc.).

6 Evaluation

This is actually the research evaluation step, which determines the working ability of the proposed FL framework on the prediction of customer churn. In evaluating the performance of the global model, accuracy, precision, recall, and F1-score are considered for checking how well a model performs in correctly classifying a customer as being churned or not, the balancing of true and false positives, and issues concerning data imbalance, respectively.

After several rounds of model aggregation and retraining to make a better global model, this review was done. Performance metrics were kept track of over seven iterations to show trends in growth. The outcomes show a very certain process in which model performance kept going up. Iteration 5 had the best accuracy (82%), precision (82.01%), recall (82.44%), and F1-score (80.63%). This iteration was selected for the final model because it outperformed all the other iterations in all the evaluation measures. For real-time applications, this process of iterative refinement can be endless, and only seven have been considered herein. You can raise the number of iterations until you get the desired accuracy and performance metrics. This process should terminate when the desired accuracy level is achieved, after which the final global model can be utilized.

A comparison test was also done to find out how well the federated model did, compared to the more common centralized versions. The results indicate that the performance achieved by the federated model is similar and, in many cases better, while strictly adhering to data privacy rules. This evidences why FL is superior in balancing data privacy with accuracy in predictions.[13]

The evaluation process also showed the strength of the global model concerning different area datasets. Because it gathers updates from various nodes around the globe, the global model is powerful enough to work with newer data. It illustrates the trends in customer churn across Mumbai, Ireland, and Ohio. Generalization thus ensures the model's applicability for a wide range of persons and services.[7]

It even keeps the performance metrics at both the regional and world levels to make this framework even more reliable. The global aggregated model performed better than the local models on all evaluation metrics in a consistent manner, thereby indicating that there is some benefit to knowledge sharing in a federated setting. These test results depict that the suggested FL framework can be scaled up and is helpful for real-life telecom applications.

Model Version	**Accuracy**	**Precision**	**Recall**	**F1-Score**
v1_aggregated	0.830422	0.844833	0.830422	0.803937
v2_aggregated	**0.844056**	0.838567	**0.844056**	**0.834514**
v3_aggregated	0.842352	0.836001	0.842352	0.835025
v4_aggregated	0.832552	0.846116	0.832552	0.807195
v5_aggregated	0.738816	0.807062	0.738816	0.628274
v6_aggregated	0.738389	0.545219	0.738389	0.627269
v7_aggregated	0.738389	0.545219	0.738389	0.627269

Table 4: Final Evaluation Results of Aggregated Models

Model Selection

Based on the final evaluation results, the model `v2_aggregated` was selected as the best-performing model due to its highest accuracy of **84.40%**, along with balanced **Precision**, **Recall**, and **F1-Score** values.[8]

7 Future Work

Improved Customer Churn Prediction in Telecom Using Federated Learning: The research conducted is useful; it would be great to enhance it further. This becomes a very important domain: advanced aggregation methods. Techniques may include weighted aggregation, whereby updates from clients are prioritized with dataset size and quality in mind, which might be considered in the near future. It will help in faster convergence of global models. Non-IID data may be a problem, but there are techniques like FedProx and dynamic grouping that will help. This makes it easier to work with large sets of data from different customers. Another important area is privacy and security. Federated learning provides privacy by design, but in the future, it would be nice to add **Differential Privacy (DP)** for preventing inference attacks and assuring GDPR compliance. It can be further improved by the concept of secure multi-party computation and homomorphic encryption, where computations over encrypted data are enabled and threats from other parties are avoided.

Other major objectives are to make conversations more effective. Such model updates can be further made smaller using gradient sparsification and quantisation to increase the efficiency of the system by using lower bandwidth. **Asynchronous FL** might be added in order to reduce latency resulting from the synchronization of the clients. Flexibility in the system will provide better compatibility with the natural networks.Other features that may be of interest and, at the same time, be useful in the future relate to automation and coordination with either **Flower** or **TensorFlow Federated (TFF)**.
As an example, one would expect to carry out large-scale launches of different tasks by automating activity related to sending updates and even client coordination and monitoring their performance. **CI/CD pipelines** for FL models will allow automatic model retraining; after that, a version of such a model could be released to keep the updates coming at all times without significant human intervention.

Finally,very promising area is the extension of the use cases of FL beyond the prediction of churn. The applications will vary from "fraud detection" and "predictive maintenance" to "customer segmentation." Additionally, it would allow the telecom company to detect fraud, forecast breakdowns in equipment, and also enable them to personalize customer experience with data privacy. These issues, when fixed, will make the FL system strong, grow, and be useful for actual real-life problems related to telecom.

8 Conclusion

The study discussed herein has presented very high potential for FL to change the existing landscape of telecom churn prediction with the solution of key challenges that are privacy, scalability, and heterogeneous data. Indeed, traditional centralized approaches put the private data of customers at risk and complicate legal compliance. The proposed framework in FL guarantees data privacy; the raw data stays local, while nodes across different parts of the world can collaborate on learning. This goes in line with data security laws like GDPR and gains trust among customers by keeping to global standards.

It follows that the implementation of this research will involve designing and deploying a decentralized FL framework with state-of-the-art aggregation techniques, lightweight neural network models, and a robust cloud infrastructure using AWS S3 and EC2 to emulate real-world scenarios. Indeed, this approach works, with the global model achieving as high as 82% accuracy at an F1-score of 80.63% after a few attempts. These help quantify the performance of FL in yielding the maximum amount of information without compromising important information on privacy.It is a very vital aspect in the big data era with stiffened rules for privacy.

It further investigates problems brought about by non-IID data—a common case for real-world datasets in the telecom domain. These weighted aggregation methods within the framework made updates coming from nodes with larger or more representative datasets weigh more in changes of the global model and thus became better at generalization. This contribution links localized learning to global information and opens a path toward advanced methods in a federated environment.

It accounts for the computation concerning scalability and speed of the proposed answer. Employed lightweight neural networks ensure devices with scarce resources, like edge nodes and regional servers, could continue their participation in the process of shared training. That makes this framework much handier for real-life telecom operations where computing power may seriously differ from area to area.

The value of this work goes beyond these technical achievements by showing the bigger effects of using FL in the telecom business. Decentralized learning will allow the protection of data privacy and telcos to benefit from the advantage of joint intelligence without harming customers' trust. Other use cases are fraud detection, customer segmentation, and predictive maintenance, making FL very useful and revolutionary for the industry.

Even with good results, this work has also pointed out some avenues for future research: taking a closer look at more advanced techniques in privacy protection like differential privacy and homomorphic encryption, enhancing communication efficiency by sparsification of gradients, and automating the workflow of FL through orchestration tools like TensorFlow Federated or Flower. Anyway, it could be further scaled and made robust against failures if the framework allowed the support of asynchronous FL whose performance could be tested with much bigger and real-world data sets.

In all, this provided a very good foundation on how Federated Learning could work in the telecom industry. This proposes a framework not only to meet the present requirements of the industry but even keeps a view on its future requirements, mixing and matching privacy-protective techniques with scalable and efficient learning methodologies. This work will set the base for new applications of FL that will allow the telecom companies to make decisions based on data without giving away customer privacy. The implications go beyond the telecom world; it points out the role of FL in those domains where privacy and security of data are of prime importance.

References

[1] D. Azzam, M. Hamed, N. Kasiem, Y. Eid, and W. Medhat. Customer churn prediction using apriori algorithm and ensemble learning. In *2023 5th Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pages 377–381, 2023.

[2] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecný, S. Mazzocchi, H. B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander. Towards federated learning at scale: System design, 2019.

[3] A. Gaur and R. Dubey. Predicting customer churn in telecom sector using vari-

ous machine learning techniques. In *2018 International Conference on Advanced Computation and Telecommunication (ICACAT)*, pages 1–5, 2018.

[4] X. Hu, Y. Yang, L. Chen, and S. Zhu. Research on a customer churn combination prediction model based on decision tree and neural network. In *2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, pages 129–132, 2020.

[5] J. Huh and W. Lee. Privacy-preserving consumer churn prediction in telecommunication through federated machine learning. In *2024 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 355–356, 2024.

[6] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al. Advances and open problems in federated learning, 2021.

[7] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, pages 1137–1143, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[8] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, 2017.

[9] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.

[10] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6), 2021.

[11] L. Ou. Customer churn prediction based on interpretable machine learning algorithms in telecom industry. In *2023 International Conference on Computer Simulation and Modeling, Information Security (CSMIS)*, pages 644–647, 2023.

[12] A. Patel and A. G. Kumar. Predicting customer churn in telecom industry: A machine learning approach for improving customer retention. In *2023 IEEE 11th Region 10 Humanitarian Technology Conference (R10-HTC)*, pages 558–561, 2023.

[13] A. Senthilselvi, V. Kanishk, K. Vineesh, and A. Praveen Raj. A novel approach to customer churn prediction in telecom. In *2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, pages 1–7, 2024.

[14] P. Tang. Telecom customer churn prediction model combining k-means and xgboost algorithm. In *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, pages 1128–1131, 2020.

[15] P. Voigt and A. V. dem Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide.* Springer, Cham, Switzerland, 1st edition, 2017.

[16] A. X. Wang, S. S. Chukova, and B. P. Nguyen. Data-centric ai to improve churn prediction with synthetic data. In *2023 3rd International Conference on Computer, Control and Robotics (ICCCR)*, pages 409–413, 2023.