Journal of
Sensor and Actuator Networks

MDPI

Article

Reduction in Data Imbalance for Client-Side Training in Federated Learning for the Prediction of Stock Market Prices

Momina Shaheen [1,*], Muhammad Shoaib Farooq [1] and Tariq Umer [2]

[1] School of Systems and Technology, University of Management and Technology, Lahore 54000, Pakistan; shoaib.farooq@umt.edu.pk
[2] Department of Computer Science, COMSATS University Islamabad Lahore Campus, Lahore 54000, Pakistan; tariqumer@cuilahore.edu.pk
* Correspondence: s2018288003@umt.edu.pk

Abstract: The approach of federated learning (FL) addresses significant challenges, including access rights, privacy, security, and the availability of diverse data. However, edge devices produce and collect data in a non-independent and identically distributed (non-IID) manner. Therefore, it is possible that the number of data samples may vary among the edge devices. This study elucidates an approach for implementing FL to achieve a balance between training accuracy and imbalanced data. This approach entails the implementation of data augmentation in data distribution by utilizing class estimation and by balancing on the client side during local training. Secondly, simple linear regression is utilized for model training at the client side to manage the optimal computation cost to achieve a reduction in computation cost. To validate the proposed approach, the technique was applied to a stock market dataset comprising stocks (AAL, ADBE, ASDK, and BSX) to predict the day-to-day values of stocks. The proposed approach has demonstrated favorable results, exhibiting a strong fit of 0.95 and above with a low error rate. The R-squared values, predominantly ranging from 0.97 to 0.98, indicate the model's effectiveness in capturing variations in stock prices. Strong fits are observed within 75 to 80 iterations for stocks displaying consistently high R-squared values, signifying accuracy. On the 100th iteration, the declining MSE, MAE, and RMSE (AAL at 122.03, 4.89, 11.04, respectively; ADBE at 457.35, 17.79, and 21.38, respectively; ASDK at 182.78, 5.81, 13.51, respectively; and BSX at 34.50, 4.87, 5.87, respectively) values corroborated the positive results of the proposed approach with minimal data loss.

Keywords: data imbalance; edge networks; federated learning; stock market prediction

Citation: Shaheen, M.; Farooq, M.S.; Umer, T. Reduction in Data Imbalance for Client-Side Training in Federated Learning for the Prediction of Stock Market Prices. *J. Sens. Actuator Netw.* **2024**, *13*, 1. https://doi.org/10.3390/jsan13010001

Academic Editors: Laura Verde, Fiammetta Marulli, Rosario Catelli and Giovanni Paragliola

Received: 2 November 2023
Revised: 14 December 2023
Accepted: 19 December 2023
Published: 21 December 2023

1. Introduction

The proliferation of the internet of things (IoT) and edge devices has led to a substantial surge in data generation. It has been forecasted to surpass 30 billion, resulting in a potential global data volume of approximately 163 zettabytes (trillion gigabytes) [1,2]. The cost of computation, processing, and communication associated with storing data in data centers is significant. However, this is not the only concern as the sensitivity of edge data with regard to privacy has resulted in mass data-sharing hesitancy by the general public. In the context of industries, safeguarding data from privacy breaches and cyber-attacks is an emerging challenge. To preserve the confidentiality of data residing on edge nodes, federated learning (FL) is a widely adopted approach [3,4]. FL has emerged as a promising approach to tackle critical concerns such as access rights, privacy, security, and access to heterogeneous data [5,6]. FL facilitates the creation of a collective learning model via multiple nodes without the need to exchange their data samples [7], thus saving the cost of communication and storage of data to central servers while preserving the privacy of edge data [8,9]. This technique has successfully found applications in diverse domains, including mobile traffic prediction and monitoring [10,11], healthcare [12,13], the

internet of things [14–16], transportation and autonomous vehicles [17], digital twin [16], blockchain [18], disaster management [19,20], natural language processing [21], knowledge extraction [22], agriculture [23], pharmaceutics, and medical sciences [24,25].

The approach of FL differs from that of distributed machine learning (DML), where the data are initially centralized on a server and subsequently partitioned into subsets for the purpose of learning tasks. In this scenario, the sample size follows a uniform distribution and is both independent and identically distributed (IID) [26]. In contrast, FL distributes the algorithm for processing across edge devices rather than concentrating the data on a central server [27,28], as presented in Figure 1. As a result, it can be observed that FL possesses a greater number of training subsets in comparison to DML. This may lead to non-identical distribution of data (non-IID), as stated in [29]. Most classification tasks exhibit imbalanced class distributions, which can lead to biased machine learning algorithms [30]. The problem of imbalanced distribution poses a significant challenge. In supervised learning, models require labeled training data for updating their parameters. The imbalance of the training data is in the variation of the number of samples for different classes/labels. Its major solutions are ensemble learning and sampling techniques [31]. The under-sampling method is straightforward to implement as it involves sampling the data to achieve a balanced proportion [32,33]. Under sampling techniques, examples from the training dataset that belong to the majority class are removed to better balance the class distribution. However, the implementation of this technique involves a large dataset, whereas the local data of the edge devices in the FL network are generally limited in size.

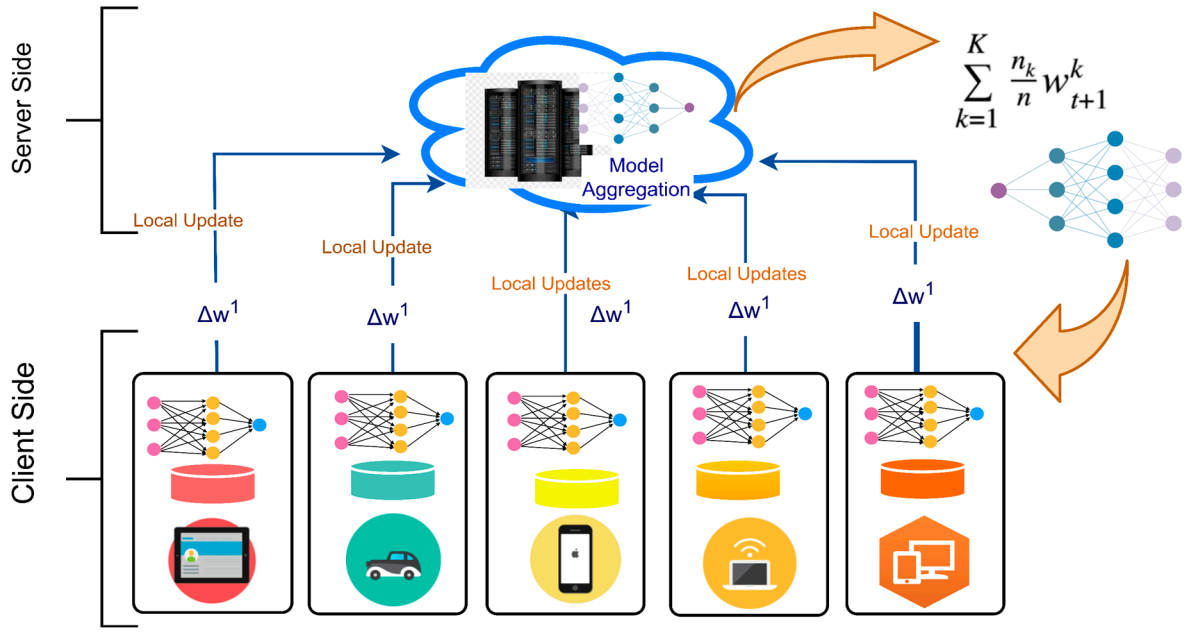

Figure 1. Federated learning approach where edge devices train the model with their local data and send the trained model to FL server. FL server then aggregates the model to make a global model and then send the updated model to edge nodes.

The edge nodes use the available data to train a collaborative model in an FL setting. The distribution of data originating from edge devices is contingent upon their usage patterns. For instance, when comparing cameras situated in natural habitats to those placed in parks, the latter are inclined to capture a greater number of images featuring individuals. To facilitate a deeper comprehension, these imbalances can be classified into three distinct types: (1) Size imbalance, which refers to a scenario in which the size of the data sample on each edge node is irregular. (2) Local imbalance, also known as non-IID, refers to a scenario where not all nodes in a system follow the same data distribution [34]. (3) The phenomenon of global imbalance refers to the situation where the distribution of data across all nodes

in a system is characterized by a significant class imbalance, as noted in reference [35]. FL aims to create a global model that generalizes well to unseen data. However, if certain classes are underrepresented during training due to data imbalance, the model may not generalize effectively to those classes during inference.

To summarize, FL is a proficient distributed machine learning approach that offers the added benefit of protecting privacy; however, it encounters difficulties in managing datasets that are unbalanced or skewed [35]. In FL, models are trained on local datasets from different clients. If some clients have a disproportionately large or small number of samples from a particular class, the model may become biased toward those classes. Similarly, clients with more data may have a larger influence on the global model during aggregation, potentially overshadowing contributions from clients with smaller datasets. This can lead to a lack of fairness in the learning process. This study aims to address this issue to enhance the precision of the results by tackling the issue of local data imbalance in a federated setting by balancing the classes of data and by addressing the challenge of imbalanced data while maintaining privacy and minimizing communication overhead.

Contribution: The uneven distribution of data can result in bias during the training phase of a model, leading to reduced accuracy in FL applications [34,35]. The primary contribution of this paper is the resolution of imbalanced data issues through the implementation of class estimation and balancing technique that adjusts local imbalance via class estimation and data augmentation on the client side:

- A novel approach comprising balanced federated learning (Bal-fed) has been proposed for implementation on the FL setting to achieve a class balance to increase the training accuracy with lower computation costs.
- The proposed approach harnesses class estimation and balancing to train a linear regression (LR) machine learning algorithm in an FL setting. To assess its applicability, this approach is implemented on stock market data (text and numerical data).
- This approach has been shown to enhance the accuracy of the model in FL settings by demonstrating an accuracy rate exceeding 95% in the FL environment, even after mitigating the issue of data imbalance.

The present article is organized into five primary sections. Section 2 elucidates the studies conducted and the outcomes of the experiments executed to address predicaments and challenges like the problem under consideration. The third section of the paper provides a detailed account of the methodology and materials employed in conducting the experiments. Section 4 encompasses the setup and execution of the experiment. The outcomes of the applied methodology. The experiment is concluded in Section 5, where the corresponding results are concluded and future directions in this area are highlighted.

2. Related Work

In the realm of distributed data, FL is a developing methodology that aims to address privacy concerns [36]. The advancement of novel frameworks aimed at enhancing healthcare technologies has been the focus of numerous research endeavors [37–41]. Similarly, the industrial applications of FL are emerging as a potential area of research. The application of deep CNN-based methods in industrial systems holds immense potential for revolutionizing various aspects of operations. In industrial settings, deep CNNs excel in tasks such as image recognition, defect detection, and predictive maintenance. For instance, in quality control processes, a well-trained deep CNN can accurately identify defects in manufactured products by analyzing visual data from production lines. Moreover, predictive maintenance becomes more effective as deep CNNs analyze sensor data to detect subtle anomalies indicative of machinery wear or potential failures. This review underscores the transformative impact of deep CNNs in enhancing efficiency, reducing downtime, and ensuring the overall reliability of industrial systems. A plethora of approaches and frameworks exist for FL; however, only a limited number of studies have been conducted to assess the efficacy of data balancing in FL approaches and frameworks [42]. This section

provides a comprehensive overview of the experiments conducted, with a particular focus on those that are relevant to our study.

The production and assembly of data using nodes in a network often occurs in a non-IID manner, as noted in previous studies [34,43–45]. For instance, individuals who use cellular devices may frequently utilize language in the context of predicting the subsequent word. Additionally, the quantity of data distributed among nodes may vary significantly. The enhancement of the convergence of the FL algorithm can be achieved through the quantification of the statistical heterogeneity of the data. Recent studies have introduced and applied various techniques and instruments for computing statistical heterogeneity through metrics [46]. However, it is not possible to measure these metrics before the training process.

To improve the performance of machine learning models, Verma et al. proposed novel methodologies particularly in the context of highly imbalanced datasets [47]. The researchers analyzed the participants' performances in various settings. The model in question was an artificial intelligence (AI) model that was developed using an FL approach, which involved the integration of data from multiple sources.

The algorithms utilized for the autonomous computation of updates by clients, based on their local data, in the present model were established by Konecny et al. (2016) [48]. The updated data can be transmitted to a central server by them. The central server computes a new global model by amalgamating the changes made by the clients. The primary users of this system are mobile devices, and the efficiency of their communication is of utmost importance. This study proposes two methods, namely structured updates [49] and sketch updates [50], to mitigate the expenses associated with up-link transmission.

Nilsson et al. [51] conducted a benchmark of three FL algorithms. Storing the data on the server led to a comparative analysis of the efficacy of the three methods. Federated averaging, commonly referred to as FedAvg, is a distributed machine learning algorithm that enables the training of models on decentralized data. Federated stochastic variance reduced gradient and CO-OP are among the algorithms utilized in federated learning. The algorithms underwent testing with non-independent and identically distributed data. The concept of independent and identically distributed (IID) variables is a fundamental principle in probability theory and statistics. The present study investigates various data partitioning techniques applied to the MINIST dataset. The investigation revealed that the FedAvg algorithm demonstrated the highest level of accuracy.

The integration of FL and deep reinforcement learning (DRL) as a means of enhancing edge systems is suggested in [52]. The implementation of this concept has resulted in improvements in caching, networking, and mobile edge computing (MEC). To leverage edge nodes and facilitate device collaboration, they developed the "In-Edge AI" framework. Empirical evidence has demonstrated that this framework exhibits exceptional performance with minimal cognitive load. Ultimately, the authors deliberated on a range of challenges and opportunities to depict the promising future of "In-Edge AI" [53]. Xu et al. [54] conducted a survey to investigate the growth of FL in healthcare informatics. The authors provided a comprehensive overview of the vulnerabilities, statistical challenges, and privacy concerns associated with the issue at hand. They also proposed solutions to address these issues. The authors anticipate that their findings will serve as valuable resources for the researchers in the field of computational research on machine learning algorithms. Specifically, they claimed that their work will aid in the management of large amounts of distributed data while also considering privacy and health informatics [54].

Sattler et al. developed the clustered federated learning (CFL) approach [55] to tackle the issue of reduced accuracy in FL scenarios when the data distribution of local clients deviates. Federated multi-task learning (FMTL) is a technique that is facilitated using CFL. The geometric properties of the FL loss surface are utilized in the FMTL approach to facilitate the categorization of client populations based on the distribution of trainable data. It is recommended that the FL communication mechanism in CFL remain unchanged. The clustering quality is supported by robust mathematical guarantees, which are further

enhanced by the incorporation of deep neural networks (DNNs). CFL maintains a variety of client demographics over an extended period while ensuring privacy protection measures are in place. This approach allows for flexibility in data management. When comparing FL and CFL, the latter is claimed to be regarded as a post-processing technique that attains objectives that are either equivalent to or greater than those of the former [56].

Frameworks for secure FL were proposed in [57]. The authors presented a comprehensive FL platform that encompasses federated transfer learning (FTL), as well as vertical and horizontal FL. The authors presented concepts related to FL, discussed the necessary infrastructure, and explored the potential implications of FL implementation. The authors provided a comprehensive analysis of the progress made in this area. Furthermore, utilizing federated processes, experts suggested the establishment of data networks among enterprises to facilitate data sharing while ensuring the protection of end-users' privacy [58].

Recently, a proposed framework for agnostic FL optimized a centralized model [59]. The optimization of client distributions enables it to be suitable for any resulting target distribution. The authors expressed their opinion that the framework produces a perception of fairness. The authors proposed a rapid stochastic optimization method to tackle the optimization issues at hand. The authors also presented convergence bounds for the approach, assuming a given hypothesis set and a convex loss function. The authors utilized multiple datasets to demonstrate the advantages of their proposed techniques. The paradigm proposed by the authors has potential applicability in various learning contexts, including but not limited to domain adaptability, cloud computing, and drifting, as suggested in a previous research [45]. In the field of mobile devices, Bonawitz and colleagues have proposed a scalable production approach in [60]. The system was built upon the foundation of TensorFlow (TF). The authors introduced advanced theoretical ideas, discussed diverse challenges and their corresponding resolutions, and illustrated the problems and possible solution [61].

It is apparent from the literature that numerous frameworks and techniques have been developed to address challenges such as communication cost, statistical heterogeneity, convergence, and resource allocation. However, the challenge posed by class imbalance and data imbalance is often overlooked in FL. In the context of FL and imbalanced data, researchers have explored various methods to mitigate the impact of class imbalance on model performance. Some of the approaches include oversampling [62], under-sampling [33,63], class weights [64], and localized balancing [65]. The effectiveness of these techniques may vary depending on the specific characteristics of the dataset and the FL setup. This study continues to refine existing methods and proposes a new approach to improve the handling of imbalanced data in federated learning. This article aims to use class estimation and balancing by handling the imbalanced class distribution of the data using a class estimation approach with a balancing algorithm.

3. Materials and Methods

An FL approach, referred to as Bal-fed (balanced federated learning), is proposed and illustrated in Figure 2 in this paper. This approach aims to rebalance training by implementing certain techniques such as the process of selecting clients at the edge layer and estimating the class of clients being executed. To address potential data bias, a technique known as data augmentation [66] is employed on a global scale. The linear regression algorithm is employed to train the model over edge nodes, as illustrated in Figure 2. The updated model is subsequently transmitted to the central server for model aggregation using FedAvg. The problem of data imbalance is effectively addressed in centralized machine learning. In FL, it is imperative to maintain the confidentiality of personal information. The generation of synthetic data has been proposed as a viable approach to preserving privacy while still allowing for data analysis [67,68]. This method involves creating artificial data that mimic the statistical properties of the original data, without revealing any sensitive information. This statement is in accordance with the post-processing guarantees of dif-

ferential privacy (DP) [67,69]. Augenstein et al. investigated and presented the federated approach for generating synthetic data [70]. In the context of a federated environment, the technique of data synthesis can be employed. Furthermore, it is imperative to incorporate client estimation in the self-balancing approach.

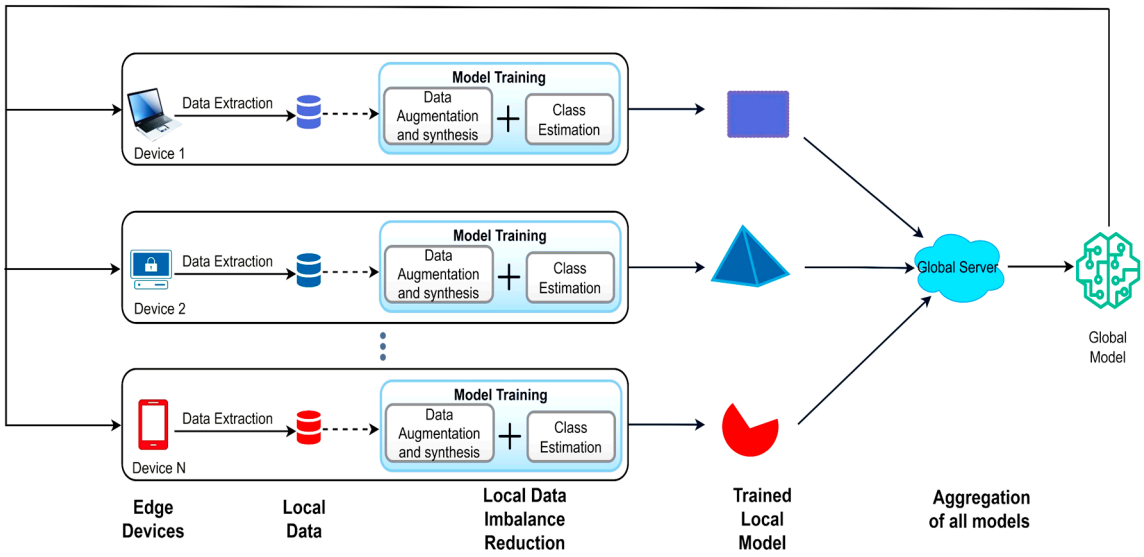

Figure 2. Proposed scheme for FL in the scenario of edge networks to reduce data skew problem.

3.1. Model Training on the Client Side

In FL settings, the raw data of clients could not be obtained due to privacy concerns. However, the class estimation and balancing scheme [71] can help the class distribution of client services on edge side according to their updated gradients. This class estimation technique can then be used with the application of data augmentation [66] to balance the classes and data evenly.

While training a model in FL, the expectation of gradient square for different classes has the following approximate relation [72]:

$$\frac{E||\nabla L(w_i)||^2}{E||\nabla L(w_j)||^2} \approx \frac{n_i^2}{n_j^2} \tag{1}$$

where L denotes the cost function of the training algorithm; n_i and n_j are the number of data samples for class i and class j, respectively, where $i \neq j$ and $i, j \in C$. Due to the correlation between class distribution and gradient, the class estimation [71] of class C_i, with class ratio $\frac{n_i^2}{\sum_j n_j^2}$, can be denoted as

$$R_i = \frac{e^{\beta \overline{||\nabla L^{aux}(w_i)||^2}}}{\sum_j e^{\beta \overline{||\nabla L^{aux}(w_j)||^2}}} \tag{2}$$

where β is a hyperparameter tuned for the normalization between classes. Thus, the composition vector $R = [R_1,..., R_C]$ that indicates the distribution of raw data can be obtained. Additionally, the Kullback–Leibler (KL) divergence can be employed to assess the class imbalance of each edge node using U (vector of classes with magnitude C). This can be defined as:

$$D_{KL}(R||U) = \sum_{i \in C} R_i log \frac{R_i}{R_j} \tag{3}$$

During FL training, after updating the model, the server has the capability to retrieve the local model from each client device. By employing the class estimation approach, we

can unveil the composition vector R_k for the chosen client k. The reward for client k is then defined as:

$$r^k = \frac{1}{D_{KL}(R^K||U)} \tag{4}$$

The class distribution can be unveiled based on the composition vector. Let $R^k(t)$ denote the composition vector of client k at time slot t. Consequently, the class ratio can be approximated through the sample mean of the composition vector, expressed as:

$$\overline{R}^{\,k} = \frac{\sum_{t=1}^{T^k} R^k\,(t)}{T^k} \tag{5}$$

With the estimated composition vector $\overline{R}$ and reward r of each client, we can design the client selection scheme with minimal class imbalance according to Algorithm 1.

Algorithm 1: Class Balancing Algorithm

Initialize
Set $S_t = \emptyset$ and $R_{total} = \emptyset$
$k_0 = \arg\max_k \hat{r}^k$
St←St ∪ {k_0}
while |St| < K do
Select $k_{min} = \arg\min_k D_{KL}((R_{total}+\bar{R}^k)||U)$ for $k \in K\backslash St$
Set St←St ∪ {k_{min}}, R_{total}←R_{total} + $\bar{R}^k$ min.
end while
Outputs: St

Data augmentation in data analysis is a technique used to increase the amount of data available for analysis [66]. It is a set of algorithms that construct synthetic data from an available dataset by creating modified copies of existing data or generating new synthetic data from the current data. Synthetic data commonly involves introducing minor variations in the data, to which the model's predictions should remain invariant [73]. Additionally, synthetic data can capture combinations of distant examples that might be challenging to infer otherwise. Data augmentation serves as a valuable tool to enhance training by supplying machine learning models with a more diverse and representative dataset, thereby improving accuracy and robustness. [66,70]. This technique has been extensively studied and applied in various fields, as evidenced by the numerous references to it in the literature [66]. Regularization is a technique commonly used during the training of machine learning models to prevent overfitting. It helps to reduce the variance of the model by adding a penalty term to the loss function, which discourages the model from fitting the noise in the training data.

In FL, the computation is performed at client sides, so it is better for edge nodes to bear a lower computation cost. Linear regression algorithms have low computational requirements [74] compared to more complex models such as SVM, Random Forest, and DL [75–78], making them suitable for large datasets or scenarios where computational resources are limited. For the FL setting, LR takes only 7.6 s for the training round, while Random Forest (RF) takes 515 s and SVM takes 4989 s [74]. Thus, we used this algorithm for local training of the clients' data. Moreover, this algorithm produces comparable outcomes to convolutional neural networks (CNNs) while minimizing computational cost. LR is well-suited for scenarios where both the dependent and independent variables are continuous, such as in the analysis of stock market datasets.

The value of a variable can be predicted through the utilization of linear regression analysis [79,80], which is based on the value of another variable. The predictability of the dependent variable is a crucial aspect to consider. By manipulating the independent variable, a hypothesis can be formulated regarding the anticipated value of the dependent variable [75]. This type of analysis involves the utilization of one or more independent variables to accurately predict the value of the dependent variable. The coefficients of the

linear equation are then calculated based on this prediction. It involves fitting a line or surface to minimize the discrepancies between the anticipated and observed output values. A collection of paired data can be utilized to create basic linear regression models that employ the "least squares" approach to determine the optimal-fit line:

$$y = \beta 0 + \beta 1X + \varepsilon \quad (6)$$

The variable y is the predicted value of the dependent variable (y) for a given value of the independent variable (x) [81]. The $\beta 0$ represents the intercept, which is the predicted value of y when x equals 0. On the other hand, $\beta 1$ is the regression coefficient that indicates the expected change in y as x increases. The variable x is considered as the independent variable, as it is expected to have an influence on y. The variable ε in the equation represents the error of the estimate, which quantifies the amount of variation present in the estimate of the regression coefficient.

LR aims to determine the line of best fit for a given set of data. This is achieved by identifying the regression coefficient ($B1$) that minimizes the total error (e) of the model. Linear regression commonly employs the mean squared error (MSE) as a metric to evaluate the accuracy of the model. The mean squared error (MSE) is computed through:

- A calculation of the deviation of the observed y-values from the predicted y-values for each corresponding x-value.
- A calculation of the square of each of these distances.
- A calculation of the mean for each of the squared distances.

3.2. Federated Averaging (FedAvg) for Model Aggregation

The benchmark FedAvg algorithm was utilized for the purpose of global model aggregation. A subset of the federation's members, consisting of clients/devices, was randomly selected to receive the initial global model synchronously, as outlined in Algorithm 2 [82]. In the current round of training, the local model of each selected client is updated by utilizing local data. The process of updating the server with client information is described in references [82,83]. To enhance the overall model, the server computes the average of all updates received from the client. Once the model parameters have reached convergence, as determined by appropriate criteria, the process is repeated with an additional round of training.

Algorithm 2: FedAvg (Federated Averaging). There are n clients, B is the local minibatch size, E is the number of local epochs per communication round, η is the learning rate, and fi is the local loss function.

Server Executes
initialize ω_0;
for each round $t = 0, 1, \ldots$ **do**
for each client $i = 0, \ldots, n-1$ in parallel
$\omega^t \leftarrow$ ClientUpdate(i, ω_t)
$_{t+1}$ n_k k
$\omega_{t+1} \leftarrow \sum_{i=1}^{n}{}_{n}\ \omega_{t+1}$
ClientUpdate (i, ω): *//Run on client i*
for each local epoch e from $0, \ldots, E-1$ **do**
for each minibatch b of size B **do**
$\omega_{e+1} \leftarrow \omega_e - \eta\sigma\ fi(\omega_e; b)$
return ω_E to server

The process of gradient descent occurs on the client side, while the aggregation of the averaged clients' updates takes place on the server. The variable k denotes the set of clients indexed by k, while η is the learning rate. The level of client computation is regulated by three crucial parameters. The present study considers the fraction of clients, denoted by i, that engage in computation during each round. Additionally, the study examines

the impact of local minibatch size *(B)* and the number of local epochs *(E)* on the overall performance of the system.

3.3. Implementation

The proposed framework must be executed to achieve a balanced training process, as illustrated in Figure 3. To mitigate overfitting, data augmentation techniques are utilized to increase the amount of data by either creating new synthetic data from the existing data or by adding modified copies of the current data. In addition, future enhancements to this study may involve the implementation of data synthesis techniques, whereby a novel dataset is generated from the existing one. The input for the process is in the form of .CSV data, which are then utilized to generate a synthetic dataset using differential privacy (DP) techniques.

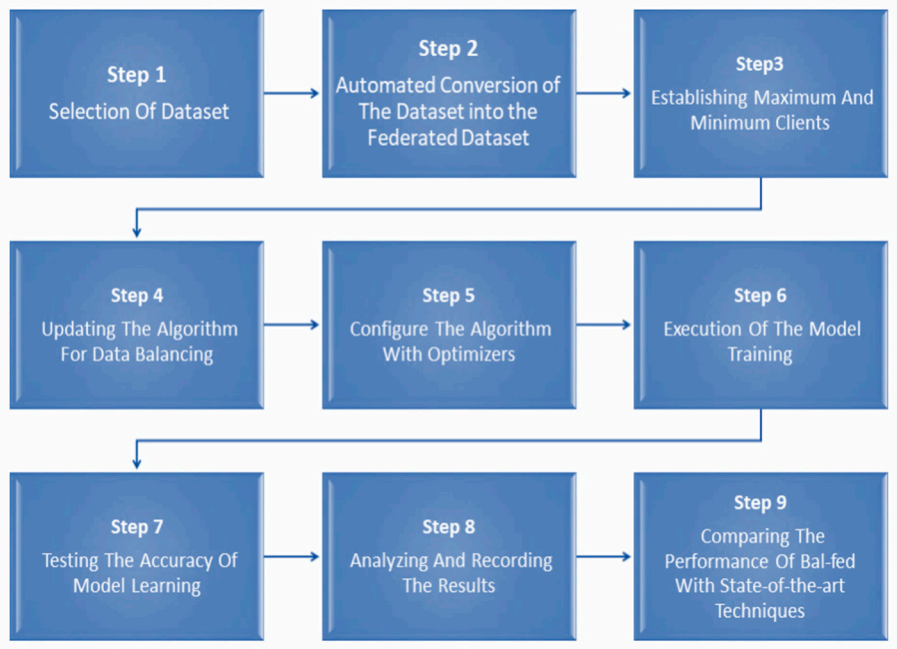

Figure 3. Schematic workflow sequence to evaluate the prediction performance of Bal-fed.

It is necessary to establish unique settings for model training and aggregation processes. The sequential workflow for the proposed scheme is elucidated in Figure 3. The collected data are transformed into federated data through an automated process. The data are distributed randomly among all clients. Subsequently, the Bal-fed technique is implemented.

Stock market dataset is utilized to evaluate the efficacy of the model in predicting stock prices. In this study, American Airline Group (AAL), Adobe (ADBE) AutoDesk Inc. (ADSK), and the Boston Scientific Corporation (BSX) were utilized as the primary subjects of investigation. The dataset underwent conversion to randomly distributed datasets to render them to be appropriate for the FL framework. The utilization of Bal-fed in handling the stock market comprises both numerical and textual data for predictive modeling tasks, which can enhance the fitness of the model in the FL setting, particularly for a diverse range of problems.

In this study, we employed the Flower framework [83] for FL [84] and other computational tasks. Developers can simulate integrated FL algorithms on their data by utilizing Flower. Additionally, testing of novel algorithms can be conducted. The researchers will identify suitable locations for conducting various types of research and provide comprehensive examples. The development of machine learning models is facilitated via the Python code, which is easily comprehensible to humans.

4. Results and Discussion

This section comprises the findings obtained from the present study. In this section, we present numerical results to showcase the effectiveness of the proposed algorithms. Our technique was evaluated by conducting tests on stock market data.

4.1. Evaluation Measure

The evaluation of an algorithm's performance is typically represented by a confusion matrix, which provides insight into the occurrence of errors. The matrix illustrates the number of predicted outcomes from the test data that correspond to the correct class, as well as the number of outcomes that are incorrectly assigned to other classes. The inputted data within the matrix are instrumental in assessing and determining the evaluation metrics of the algorithms. The quality of the classifier will be evaluated using the commonly used parameter accuracy [85], which is defined as:

$$Accuracy\ (Acc) = \frac{TP + TN}{N} \tag{7}$$

The value of N is determined by the sum of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). TN refers to the number of correctly identified negative cases, while FN represents the number of negative cases that were incorrectly identified as positive. TP denotes the number of correctly identified positive cases, and FP represents the number of positive cases that were incorrectly identified as negative [76].

The mean squared error (MSE), mean absolute error (MAE), and R-squared metrics are primarily utilized to assess the rates of prediction error and the performance of model in regression analysis [86–88]. Relative MSE is also calculated to provide greater insight about the model generalization. MAE represents the difference between the original and predicted values, calculated by averaging the absolute differences over the dataset. MSE represents the difference between the original and predicted values by squaring the average difference over the dataset. R-squared (coefficient of determination) represents how well the values fit compared to the original values. The value from 0 to 1 is interpreted as a percentage. The relative MSE considers the percentage error rather than the absolute error, thus providing insights into how well a model generalizes across datasets. A lower relative MSE indicates that the model is robust and performs well across a range of data samples.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{8}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{9}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{10}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{11}$$

In the given context, y represents the true price, $\hat{y}$ represents the predicted price, and n denotes the number of samples in the test dataset. Similarly, relatively MSE can be defined as:

$$Relative\ MSE(\%) = \left(\frac{MSE}{\overline{y}}\right) \times 100\% \tag{12}$$

where $\overline{y}$ is the mean and computed by:

$$\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{13}$$

The training time is quantified to examine the latency in machine learning model training. This time duration is influenced by factors such as the model's complexity, dataset size, and the efficiency of the processing framework. Reducing training delays provides real-time advantages for prediction and classification.

4.2. Analysis of Results

Real-time data were collected from the Y-finance API for the purposes of this experimental study. The present study analyzes the financial data of prominent stock market companies, namely American Airline Group (AAL), Adobe (ADBE) AutoDesk Inc. (ADSK), and Boston Scientific Corporation (BSX), which have significant market capitals. The retrieved data were systematically organized chronologically from 1 January 2013 to 1 February 2023, in a continuous manner. The data for each stock were organized into separate CSV files, each containing 2517 records. Each individual entry within the dataset comprised three distinct variables: the date of the recorded observation, the closing price of the asset, and the corresponding prediction. The data were transformed into a federated dataset through an automated technique. The model presented in this study was developed through the utilization of linear regression [88] within the Flower framework. The evaluation function used, i.e., R-squared, MSE, MAE, and RMSE, and the detailed results are presented in Table 1. The minimum number of edge nodes has been established as 20. Each node trains a linear regression algorithm with class estimation and balancing (Algorithm 1) using its own data and subsequently transmits the gradient of loss from the model to the server. The FL server employs the FedAvg algorithm (Algorithm 2) to update the model parameters. To update each local model, the updated parameters are distributed to the edge nodes. This iterative data processing process is executed in a convergence fashion without data sharing until the training termination requirements (100 rounds of local data training) are met. This model is trained and refined through multiple iterations to attain an optimal state where further training does not significantly improve performance. Each client processes its own local data independently in a decentralized or distributed training approach. We opted for a 90% and 10% partitioning of data for the purposes of training and testing, respectively.

Table 1. Subset of the variations of R-squared, RMSE, MAE, and MSE with respect to training iteration in stock data.

Stock	Training Iterations	R-Squared	Mean Absolute Error	Mean Squared Error	Relative MSE	RMSE
AAL	80	0.67	5.70	116.34	3.82	10.78
	81	0.68	5.70	55.50	1.82	7.45
	82	0.69	6.86	55.0	1.81	11.04
ADBE	80	0.98	15.07	691.52	2.94	26.29
	81	0.98	17.8	457.35	3.01	26.59
	82	0.98	17.95	436.03	1.94	21.38
ADSK	80	0.97	9.93	189.43	1.41	13.76
	81	0.98	7.77	184.26	1.37	13.57
	82	0.98	6.77	182.78	1.36	13.51
BSX	80	0.98	5.02	35.60	1.21	5.92
	81	0.98	13.84	34.60	1.21	5.88
	82	0.98	5.02	34.50	1.21	5.87

The resulting data-frame consisted of columns labeled as Date, Open, and Close. The outcomes of the proposed methodology for predicting stock data are illustrated via a line graph in Figure 4. The predicted values align with the actual values, demonstrating consistency between the model's projections and the observed outcomes. Subsequently, a linear model was fitted to the graph and subsequently displayed, followed by the formulation of observations. The scatter graph in Figure 5 displays the resulting graph and the fitted model. The dataset comprised 20 clients and 100 communication rounds, with each

communication round consisting of 5 epochs. The rationale for reducing the number of communication rounds with this dataset is due to its high accuracy rate of 95%, achieved within only 75 rounds. Therefore, the reduction in learning time and communication cost is achieved.

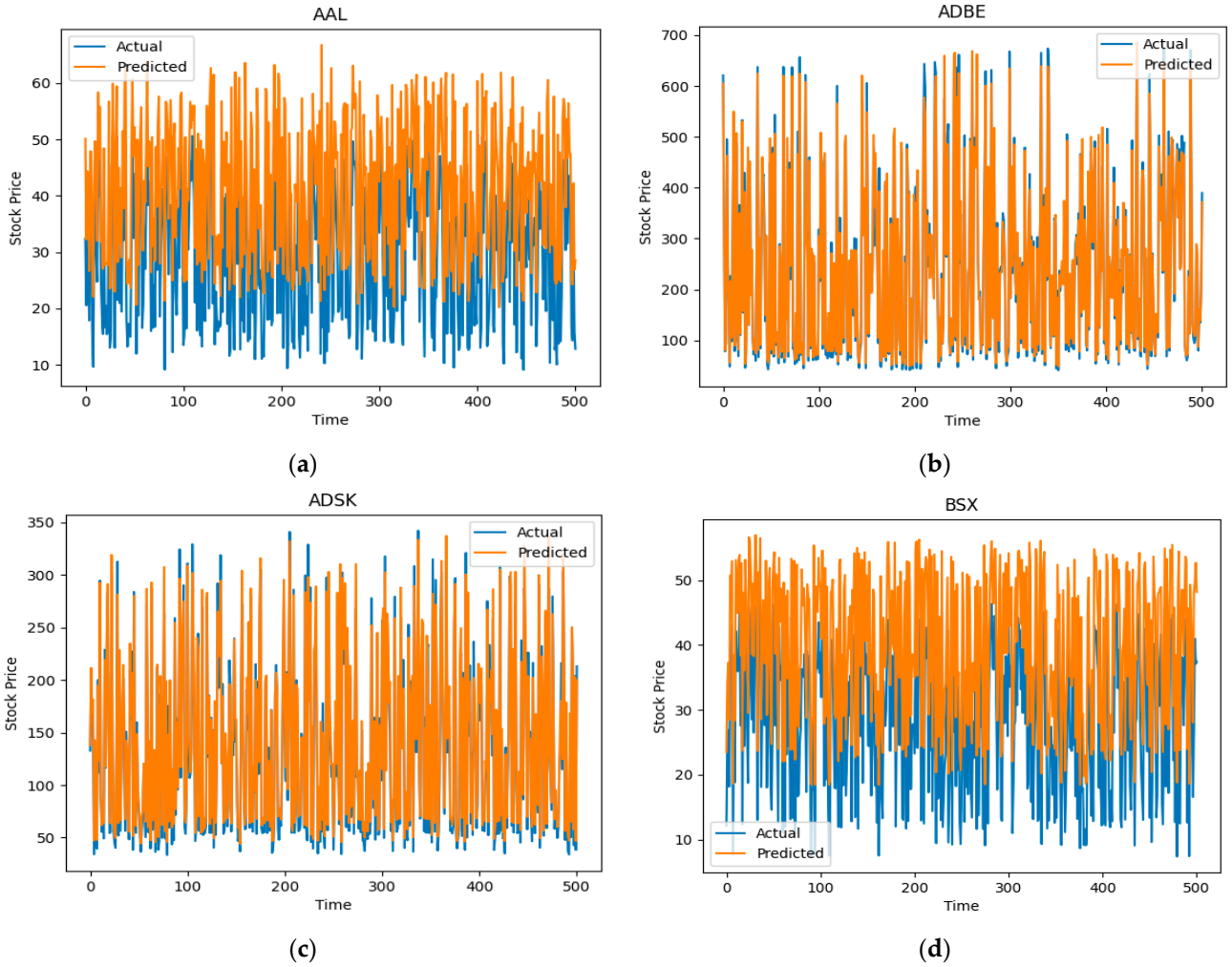

Figure 4. Prediction of stock prices using Bal-fed scheme of balancing: (**a**) predicted price of AAL; (**b**) predicted price of ADBE; (**c**) predicted price of ASDK; (**d**) predicted price of BSX.

MAE represents the average absolute difference between predicted and actual values, offering insight into the average magnitude of errors. A lower MAE is preferable. MSE is the average of squared differences between predicted and actual values, assigning more weight to larger errors. Lower MSE values are indicative of better performance. RMSE, the square root of MSE, measures the average magnitude of errors in the same units as the target variable, providing another assessment of predictive performance. R-squared gauges how well the model's predictions explain variance in the actual data, with a range from 0 to 1, with 1 signifying a perfect fit. In Table 1, R-squared values are relatively high (e.g., 0.97, 0.98) in most cases, suggesting that the model is performing well in capturing the variation in the stock prices.

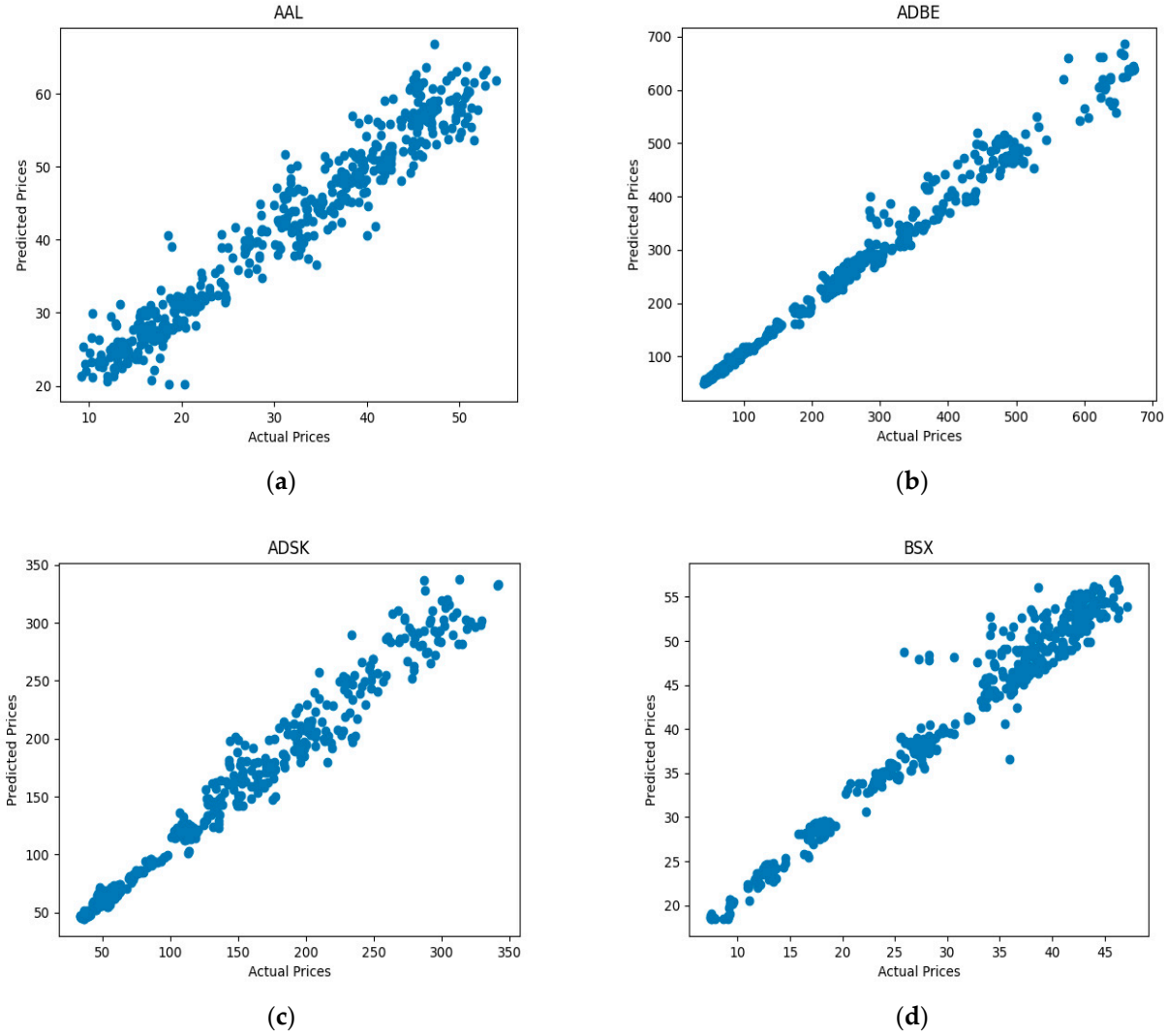

Figure 5. The fitted results of stock price prediction with respect to the actual price in scatter graphs: (**a**) predicted price vs. actual prices of AAL; (**b**) predicted price vs. actual prices of ADBE; (**c**) predicted price vs. actual prices of ASDK; (**d**) predicted price vs. actual prices of BSX.

For stock AAL during the 80th training iteration, R^2 is 0.69, MAE is calculated as 5.70, MSE is 16.347, and RMSE 10.78643. This suggests that, in the 80th training round for stock AAL, the model had a moderate R-squared value, indicating a moderate fit, as is also presented in Figure 4a. The errors in predictions had relatively low magnitudes (as indicated by low MAE, MSE, and RMSE values).

Table 2 provides a detailed snapshot of how the predictive model performs for each stock and iteration, providing insights into the model's accuracy and precision across different training scenarios. ADBE shows consistently high R-squared values (around 0.98), indicating a strong fit of the model to the actual data. The MAE values range from 15 to 17.96, which represents the average magnitude of errors in predicting Adobe's stock prices. MSE and RMSE values also provide insights into the model's accuracy and the magnitude of errors. ADSK also demonstrates high R-squared values (above 0.977), indicating a good fit of the model to the data. The model seems to improve in terms of MAE over training rounds, decreasing from 9.78 to 6.78. MSE and RMSE values show the overall accuracy and precision of the model in predicting Autodesk's stock prices. BSX exhibits high and consistent R-squared values (above 0.984), indicating a strong correlation between predicted and actual stock prices. The MAE values show variations, with the lowest value at 5.03, suggesting relatively small errors in predictions. MSE and RMSE values provide additional

insights into the accuracy and precision of the model for Boston Scientific. To summarize, for all the stocks except for AAL (ADBE, ADSK, BSX), the model demonstrates high R-squared values, indicating a strong fit. The variations in MAE, MSE, and RMSE across training rounds provide additional details about the model's performance in predicting stock prices for these specific companies.

The model training iterations were set to 100 times; although the higher accuracy and lesser error measures began to be reported only at 80 iterations. This made the accuracy of the Bal-fed much higher. By the 100th iteration, R-squared values reported to be near 0.98, which provides greater fit (as seen in Figure 6a). In the beginning, the R^2 values for AAL and BSX fluctuated until the 61st and 40th iterations, respectively, but after that, the values started to stabilize and were reported to be higher.

Figure 6. Result of analysis measures with respect to each iteration count over local data using the Bal-fed model. (**a**) Value of MAE with respect to each count. (**b**) Value of MSE with respect to each count. (**c**) Value of R-squared with respect to each count. (**a**,**d**) Value of RMSE with respect to each count.

The MSE, MAE, and RMSE continued to become lower as the number of iterations increased, as depicted in Figure 6. The decline in MSE, MAE, and RMSE is observed for AAL at 122.03, 4.89, and 11.04, respectively. For ADBE, the values of MSE, MAE, and RMSE are recorded as 457.35, 17.79, and 21.38, respectively. For ASDK, the MSE, MAE, and RMSE presented 182.78, 5.81, and 13.51, respectively. At the end, for BSX, the MSE, MAE, and RMSE recorded at 34.50, 4.87, and 5.87, respectively. These values prove the positive results of the proposed approach with minimal data loss, whereas R-squared values for AAL, ADBE, ASDK, and BSX were recorded as 0.95, 0.98, 0.98, and 0.98, respectively.

4.3. Comparative Analysis

To compare the efficacy of the Bal-fed technique with the FL approach without class estimation and balancing, we utilize a simple federated learning (FL) technique with randomly distributed data across 20 clients. The data are randomly distributed without balancing classes, and linear regression is used for client-side training without applying class estimation. For model aggregation, the FedAvg (Algorithm 2) technique is employed for global model aggregation. The resulting R-squared values are reported as 0.66, 0.79, 0.78, and 0.77 for AAL, ADBE, ADSK, and BSX, respectively.

Comparing the performance with the proposed Bal-fed technique, which employs an estimation and balancing of classes (Algorithm 1), the following results are observed in Table 2.

The results suggest that the Bal-fed technique with LR and balanced classes achieves higher accuracy (95.01%) compared to the FL technique without class estimation (79.6%). Additionally, Bal-fed results in lower data loss and a slightly longer training time, indicating a trade-off between accuracy and processing time. The lower R-squared values in the FL technique may be attributed to data imbalance issues over edge data. The predictive values achieved a 95% accuracy rate with minimal data loss, as demonstrated in Table 2. The statistical metrics of MSE, relative MSE, MAE, RMSE, and R-squared prove that the Bal-fed model has demonstrated sufficient accuracy in predicting stock prices data. In fact, its performance surpasses that reported in the literature, where a maximum accuracy of 85% was achieved [30].

Table 2. Results of stock data applied to evaluate the prediction performance of Bal-fed.

Evaluation Measure	Result with LR-Employed Bal-Fed	Result without Estimation
Accuracy	95.01%	79.6%
Data loss	19	43
Training time in seconds	14	10.7

5. Conclusions and Future Work

In the context of centralized machine learning, the consolidation of all local data onto a single server poses significant privacy concerns. FL is a machine learning approach that involves utilizing users' data to train a model, which is subsequently transmitted to the server. The sharing of confidential data on cloud servers is not conducted in this manner. Rather, solely the outcomes or trained models are uploaded. This approach exhibits greater efficiency in terms of generalization, addressing privacy concerns, and ensuring the accuracy of the system. However, a significant issue arising from using FL is the presence of data imbalance. To address this issue, we employed the class estimation and data balancing technique. We developed a novel approach for managing class distribution in which augmented data are generated automatically and distributed separately to each client for local model training. This method is designed to update gradients without requiring access to customers' data information. Furthermore, the proposed approach involves implementing class estimation and data balancing mechanism to mitigate the adverse effects of data imbalance. The Bal-fed technique has been implemented on AAL, ADBE, ADSK, and BSX stock price data collected for last 10 years. The iterative training of Bal-fed is in a decentralized manner, without sharing data between iterations. The training continues until a termination condition (100 rounds of local data training) is met. The data are partitioned into training and testing sets to evaluate the model's performance. The utilization of this technique has yielded favorable outcomes with limited data loss. R-squared values are relatively high (e.g., 0.97, 0.98) in most cases, suggesting that the model is performing well in capturing the variation in the stock prices. The model exhibits strong fits within 75 to 80 iterations for stocks (ADBE, ADSK, BSX) with consistently high R-squared values, indicating accuracy. While AAL shows a moderate fit, the model's

accuracy improves by the 100th iteration, as reflected in the decreasing MSE, MAE, and RMSE values across stocks.

To the best of our knowledge, this is the first study which provides a combination of techniques which can effectively reduce data imbalance and maintain high accuracy with a reduced computation time while ensuring privacy across various learning approaches. The proposed approach addresses privacy concerns and regulatory compliance, offering a novel method for organizations to leverage. The future research goals include advancing efforts to improve the Bal-fed ability, exploring the use of long short-term memory (LSTM) techniques for stock price prediction, and creating hybrid models that integrate technical analysis, fundamental analysis, and news analytics. The authors also express an interest in applying the Bal-fed approach to image data and extending its application to a broader range, such as medical image diagnosis.

Author Contributions: Conceptualization, M.S.F. and M.S.; methodology, M.S.; software, M.S.; validation, M.S., and T.U.; formal analysis, M.S.; investigation, M.S.; resources, T.U.; writing—original draft preparation, M.S. and M.S.F; writing—review and editing, T.U.; visualization, M.S.; supervision, T.U. and M.S.F.; project administration, M.S.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used for this research are uploaded to GitHub and can be shared on request.

Conflicts of Interest: The authors have no conflicts of interest.

References

1. Lionel, V. Internet of Things (IoT) and non-IoT Active Device Connections Worldwide from 2010 to 2025(in billions). 2022. Available online: https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/#:~:text=The%20total%20installed%20base%20of,that%20are%20expected%20in%202021 (accessed on 30 November 2023).
2. Petroc, T. Volume of Data/Information Created, Captured, Copied, and Consumed Worldwide from 2010 to 2020, with Forecasts from 2021 to 2025. 2023. Available online: https://www.statista.com/statistics/871513/worldwide-data-created/ (accessed on 30 November 2023).
3. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*; 2017; pp. 1273–1282. Available online: https://proceedings.mlr.press/v54/mcmahan17a.html (accessed on 30 November 2023).
4. Beltrán, E.T.M.; Pérez, M.Q.; Sánchez, P.M.S.; Bernal, S.L.; Bovet, G.; Pérez, M.G.; Pérez, G.M.; Celdrán, A.H. Decentralized Federated Learning: Fundamentals, State of the Art, Frameworks, Trends, and Challenges. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 2983–3013. [CrossRef]
5. Dinh, C.T.; Tran, N.H.; Nguyen, M.N.; Hong, C.S.; Bao, W.; Zomaya, A.Y.; Gramoli, V. Federated learning over wireless networks: Convergence analysis and resource allocation. *IEEE/ACM Trans. Netw.* **2020**, *29*, 398–409. [CrossRef]
6. Luping, W.; Wei, W.; Bo, L. Cmfl: Mitigating communication overhead for federated learning. In Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 7–9 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 954–964.
7. Arsalan, A.; Umer, T.; Rehman, R.A. 3 Federated Learning Technique in Enabling Data-Driven Design for Wireless Communication. In *Data-Driven Intelligence in Wireless Networks: Concepts, Solutions, and Applications*; CRC Press: Boca Raton, FL, USA, 2023; p. 23.
8. Paragliola, G. Evaluation of the trade-off between performance and communication costs in federated learning scenario. *Future Gener. Comput. Syst.* **2022**, *136*, 282–293. [CrossRef]
9. Paragliola, G.; Coronato, A. Definition of a novel federated learning approach to reduce communication costs. *Expert Syst. Appl.* **2022**, *189*, 116109. [CrossRef]
10. Paragliola, G. Application of Federated Learning Approaches for Time-Series Classification in eHealth Domain. *Procedia Comput. Sci.* **2022**, *207*, 3545–3552. [CrossRef]
11. Shaheen, M.; Farooq, M.S.; Umer, T.; Kim, B.-S. Applications of federated learning; taxonomy, challenges, and research trends. *Electronics* **2022**, *11*, 670. [CrossRef]
12. Phyu, H.P.; Stanica, R.; Naboulsi, D. Multi-slice privacy-aware traffic forecasting at RAN level: A scalable federated-learning approach. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 5038–5052. [CrossRef]
13. Paragliola, G. A federated learning-based approach to recognize subjects at a high risk of hypertension in a non-stationary scenario. *Inf. Sci.* **2023**, *622*, 16–33. [CrossRef]

14. Rahman, A.; Hasan, K.; Kundu, D.; Islam, M.J.; Debnath, T.; Band, S.S.; Kumar, N. On the ICN-IoT with federated learning integration of communication: Concepts, security-privacy issues, applications, and future perspectives. *Future Gener. Comput. Syst.* **2023**, *138*, 61–88. [CrossRef]
15. Zhao, Y.; Zhao, J.; Jiang, L.; Tan, R.; Niyato, D. Mobile Edge Computing, Blockchain and Reputation-based Crowdsourcing IoT Federated Learning: A Secure, Decentralized and Privacy-preserving System. *arXiv* **2019**, arXiv:1906.10893.
16. Lu, Y.; Huang, X.; Zhang, K.; Maharjan, S.; Zhang, Y. Communication-efficient federated learning for digital twin edge networks in industrial IoT. *IEEE Trans. Ind. Inform.* **2020**, *17*, 5709–5718. [CrossRef]
17. Pokhrel, S.R.; Choi, J. Federated Learning with Blockchain for Autonomous Vehicles: Analysis and Design Challenges. *IEEE Trans. Commun.* **2020**, *68*, 4734–4746. [CrossRef]
18. Li, L.; Qin, J.; Luo, J. A Blockchain-Based Federated-Learning Framework for Defense against Backdoor Attacks. *Electronics* **2023**, *12*, 2500. [CrossRef]
19. Farooq, M.S.; Tehseen, R.; Qureshi, J.N.; Omer, U.; Yaqoob, R.; Tanweer, H.A.; Atal, Z. FFM: Flood forecasting model using federated learning. *IEEE Access* **2023**, *11*, 24472–24483. [CrossRef]
20. Tehseen, R.; Farooq, M.S.; Abid, A. A framework for the prediction of earthquake using federated learning. *PeerJ Comput. Sci.* **2021**, *7*, e540. [CrossRef]
21. Marulli, F.; Verde, L.; Marrore, S.; Campanile, L. A Federated Consensus-Based Model for Enhancing Fake News and Misleading Information Debunking. In Proceedings of the Intelligent Decision Technologies: Proceedings of the 14th KES-IDT 2022 Conference, Rhodes, Greece, 22 June 2022; Springer Nature: Singapore, 2022; pp. 587–596.
22. Marulli, F.; Verde, L.; Marrone, S.; Barone, R.; De Biase, M.S. Evaluating efficiency and effectiveness of federated learning approaches in knowledge extraction tasks. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6.
23. Mehta, S.; Kukreja, V.; Yadav, R. Advanced Mango Leaf Disease Detection and Severity Analysis with Federated Learning and CNN. In Proceedings of the 2023 3rd International Conference on Intelligent Technologies (CONIT), Hubli, India, 23–25 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.
24. Silva, S.; Gutman, B.A.; Romero, E.; Thompson, P.M.; Altmann, A.; Lorenzi, M. Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data. In Proceedings of the International Symposium on Biomedical Imaging, Venice, Italy, 8–11 April 2019; Volume 2019, pp. 270–274.
25. Antunes, R.S.; André da Costa, C.; Küderle, A.; Yari, I.A.; Eskofier, B. Federated learning for healthcare: Systematic review and architecture proposal. *ACM Trans. Intell. Syst. Technol. (TIST)* **2022**, *13*, 1–23. [CrossRef]
26. Li, X.L.S.; Lv, L.; Ding, Z. Mobile app start-up prediction based on federated learning and attributed heterogeneous network embedding. *Future Internet* **2021**, *13*, 256. [CrossRef]
27. Guendouzi, B.S.; Ouchani, S.; Assaad, H.E.; Zaher, M.E. A systematic review of federated learning: Challenges, aggregation methods, and development tools. *J. Netw. Comput. Appl.* **2023**, *220*, 103714. [CrossRef]
28. Ye, M.; Fang, X.; Du, B.; Yuen, P.C.; Tao, D. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Comput. Surv.* **2023**, *56*, 1–44. [CrossRef]
29. Liu, Y.; Ma, Z.; Liu, X.; Ma, S.; Nepal, S.; Deng, R. Boosting Privately: Privacy-Preserving Federated Extreme Boosting for Mobile Crowdsensing. *arXiv* **2019**, arXiv:1907.10218.
30. Yang, M.; Wang, X.; Zhu, H.; Wang, H.; Qian, H. Federated learning with class imbalance reduction. In Proceedings of the 2021 29th European Signal Processing Conference (EUSIPCO), Dublin, Ireland, 23–27 August 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 2174–2178.
31. Wang, L.; Xu, S.; Wang, X.; Zhu, Q. Addressing class imbalance in federated learning. In Proceedings of the AAAI Conference on Artificial Intelligence, virtual, 2–9 February 2021; Volume 35, pp. 10165–10173.
32. Yen, S.J.; Lee, Y.S. Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset. In Proceedings of the Intelligent Control and Automation: International Conference on Intelligent Computing, ICIC 2006, Kunming, China, 16–19 August 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 731–740.
33. Lin, W.C.; Tsai, C.F.; Hu, Y.H.; Jhang, J.S. Clustering-based undersampling in class-imbalanced data. *Inf. Sci.* **2017**, *409*, 17–26. [CrossRef]
34. Seol, M.; Kim, T. Performance Enhancement in Federated Learning by Reducing Class Imbalance of Non-IID Data. *Sensors* **2023**, *23*, 1152. [CrossRef] [PubMed]
35. Duan, M.; Liu, D.; Chen, X.; Tan, Y.; Ren, J.; Qiao, L.; Liang, L. Astraea: Selfbalancing federated learning for improving classification accuracy of mobile deep learning applications. In Proceedings of the 2019 IEEE International Conference on Computer Design, ICCD, Abu Dhabi, UAE, 17–20 November 2019; pp. 246–254.
36. Marulli, F.; Bellini, E.; Marrone, S. A security-oriented architecture for federated learning in cloud environments. In Proceedings of the Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 34th International Conference on Advanced Information Networking and Applications (WAINA-2020), Caserta, Italy, 15–17 April 2020; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 730–741.
37. Rahman, A.; Hossain, M.S.; Muhammad, G.; Kundu, D.; Debnath, T.; Rahman, M.; Khan, M.S.I.; Tiwari, P.; Band, S.S. Federated learning-based AI approaches in smart healthcare: Concepts, taxonomies, challenges and open issues. *Clust. Comput.* **2023**, *26*, 2271–2311. [CrossRef] [PubMed]

38. Subramanian, M.; Rajasekar, V.; V.E., S.; Shanmugavadivel, K.; Nandhini, P.S. Effectiveness of Decentralized Federated Learning Algorithms in Healthcare: A Case Study on Cancer Classification. *Electronics* **2022**, *11*, 4117. [CrossRef]
39. Cremonesi, F.; Vesin, M.; Cansiz, S.; Bouillard, Y.; Balelli, I.; Innocenti, L.; Silva, S.; Ayed, S.S.; Taiello, R.; Kameni, L. Fed-BioMed: Open, Transparent and Trusted Federated Learning for Real-world Healthcare Applications. *arXiv* **2023**, arXiv:2304.12012.
40. Farooq, M.S.; Younas, H.A. Beta Thalassemia Carriers detection empowered federated Learning. *arXiv* **2023**, arXiv:2306.01818.
41. Nguyen, D.C.; Ding, M.; Pathirana, P.N.; Seneviratne, A.; Zomaya, A.Y. Federated learning for COVID-19 detection with generative adversarial networks in edge cloud computing. *IEEE Internet Things J.* **2021**, *9*, 10257–10271. [CrossRef]
42. Berghout, T.; Benbouzid, M.; Bentrcia, T.; Lim, W.H.; Amirat, Y. Federated Learning for Condition Monitoring of Industrial Processes: A Review on Fault Diagnosis Methods, Challenges, and Prospects. *Electronics* **2023**, *12*, 158. [CrossRef]
43. Yang, L.; Huang, J.; Lin, W.; Cao, J. Personalized federated learning on non-IID data via group-based meta-learning. *ACM Trans. Knowl. Discov. Data* **2023**, *17*, 1–20. [CrossRef]
44. Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; Chandra, V. Federated learning with non-iid data. *arXiv* **2018**, arXiv:1806.00582. [CrossRef]
45. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated opti mization in heterogeneous networks. *arXiv* **2020**, arXiv:1812.06127.
46. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [CrossRef]
47. Verma, D.C.; White, G.; Julier, S.; Pasteris, S.; Chakraborty, S.; Cirincione, G. Approaches to address the data skew problem in federated learning. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*; International Society for Optics and Photonics: Bellingham, WA, USA, 2019; Volume 11006, p. 110061I.
48. Konečný, J.; McMahan, H.B.; Yu, F.X.; Richtárik, P.; Suresh, A.T.; Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv* **2016**, arXiv:1610.05492.
49. Wu, X.; Yao, X.; Wang, C.L. FedSCR: Structure-based communication reduction for federated learning. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *32*, 1565–1577. [CrossRef]
50. Rothchild, D.; Panda, A.; Ullah, E.; Ivkin, N.; Stoica, I.; Braverman, V.; Gonzalez, J.; Arora, R. Fetchsgd: Communication-efficient federated learning with sketching. In Proceedings of the International Conference on Machine Learning, virtual, 12–18 July 2020; 119, pp. 8253–8265. Available online: https://proceedings.mlr.press/v119/rothchild20a.html (accessed on 30 November 2023).
51. Nilsson, A.; Smith, S.; Ulm, G.; Gustavsson, E.; Jirstrand, M. A performance evaluation of federated learning algorithms. In Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning, Rennes, France, 10–11 December 2018; pp. 1–8.
52. Wang, H.; Wu, Z.; Xing, E.P. Removing confounding factors associated weights in deep neural networks improves the prediction accuracy for healthcare applications. In Proceedings of the BIOCOMPUTING 2019: Proceedings of the Pacific Symposium, San Diego, CA, USA, 29 October–4 November 2018; World Scientific: Singapore, 2018; pp. 54–65.
53. Nori, M.K.; Yun, S.; Kim, I.M. Fast federated learning by balancing communication trade-offs. *IEEE Trans. Commun.* **2021**, *69*, 5168–5182. [CrossRef]
54. Xu, J.; Glicksberg, B.S.; Su, C.; Walker, P.; Bian, J.; Wang, F. Federated learning for healthcare informatics. *J. Healthc. Inform. Res.* **2021**, *5*, 1–19. [CrossRef] [PubMed]
55. Sattler, F.; Muller, K.-R.; Samek, W. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 3710–3722. [CrossRef]
56. Sattler, F.; Müller, K.R.; Wiegand, T.; Samek, W. On the Byzantine Robustness of Clustered Federated Learning. In Proceedings of the ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 8861–8865. [CrossRef]
57. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–19. [CrossRef]
58. Lim, W.Y.B.; Luong, N.C.; Hoang, D.T.; Jiao, Y.; Liang, Y.C.; Yang, Q.; Niyato, D.; Miao, C. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2031–2063. [CrossRef]
59. Mohri, M.; Sivek, G.; Suresh, A.T. Agnostic federated learning. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 4615–4625.
60. Bonawitz, K.; Eichner, H.; Grieskamp, W.; Huba, D.; Ingerman, A.; Ivanov, V.; Kiddon, C.; Konečný, J.; Mazzocchi, S.; McMahan, B.; et al. Towards federated learning at scale: System design. In Proceedings of the Machine Learning and Systems, Stanford, CA, USA, 31 March–2 April 2019; Volume 1, pp. 374–388.
61. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 308–318.
62. Xiao, C.; Wang, S. Triplets Oversampling for Class Imbalanced Federated Datasets. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: Research Track: European Conference, ECML PKDD 2023, Turin, Italy, 18–22 September 2023; Koutra, D., Plant, C., Rodriguez, M.G., Baralis, E., Bonchi, F., Eds.; Proceedings, Part II (1 ed., pp. 368–383). (Lecture Notes in Computer Science; Volume 14170). Springer: Berlin/Heidelberg, Germany, 2023; pp. 368–383.

63. Zhang, J.; Li, A.; Tang, M.; Sun, J.; Chen, X.; Zhang, F.; Chen, C.; Chen, Y.; Li, H. Fed-cbs: A heterogeneity-aware client sampling mechanism for federated learning via class-imbalance reduction. In Proceedings of the International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; pp. 41354–41381.
64. Ma, Z.; Zhao, M.; Cai, X.; Jia, Z. Fast-convergent federated learning with class-weighted aggregation. *J. Syst. Archit.* **2021**, *117*, 102125. [CrossRef]
65. Liang, P.P.; Liu, T.; Ziyin, L.; Allen, N.B.; Auerbach, R.P.; Brent, D.; Salakhutdinov, R.; Morency, L.P. Think locally, act globally: Federated learning with local and global representations. *arXiv* **2020**, arXiv:2001.01523.
66. Wong, S.C.; Gatt, A.; Stamatescu, V.; McDonnell, M.D. Understanding data augmentation for classification: When to warp? In Proceedings of the 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Gold Coast, Australia, 30 November–2 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–6.
67. Dwork, C.; Roth, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* **2014**, *9*, 211–407. [CrossRef]
68. Abay, N.C.; Zhou, Y.; Kantarcioglu, M.; Thuraisingham, B.; Sweeney, L. Privacy preserving synthetic data release using deep learning. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Dublin, Ireland, 10–14 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 510–526.
69. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating noise to sensitivity in private data analysis. *J. Priv. Confidentiality* **2016**, *7*, 17–51. [CrossRef]
70. Augenstein, S.; McMahan, H.B.; Ramage, D.; Ramaswamy, S.; Kairouz, P.; Chen, M.; Mathews, R. Generative models for effective ml on private, decentralized datasets. *arXiv* **2019**, arXiv:1911.06679.
71. Bejjanki, K.K.; Gyani, J.; Gugulothu, N. Class imbalance reduction (CIR): A novel approach to software defect prediction in the presence of class imbalance. *Symmetry* **2020**, *12*, 407. [CrossRef]
72. Anand, R.; Mehrotra, K.G.; Mohan, C.K.; Ranka, S. An improved algorithm for neural network classification of imbalanced training sets. *IEEE Trans. Neural Netw.* **1993**, *4*, 962–969. [CrossRef]
73. Shorten, C.; Khoshgoftaar, T.M.; Furht, B. Text data augmentation for deep learning. *J. Big Data* **2021**, *8*, 1–34. [CrossRef]
74. Pourroostaei Ardakani, S.; Du, N.; Lin, C.; Yang, J.C.; Bi, Z.; Chen, L. A federated learning-enabled predictive analysis to forecast stock market trends. *J. Ambient. Intell. Human Comput.* **2023**, *14*, 4529–4535. [CrossRef]
75. Shaheen, M.; Awan, S.M.; Hussain, N.; Gondal, Z.A. Sentiment analysis on mobile phone reviews using supervised learning techniques. *Int. J. Mod. Educ. Comput. Sci.* **2019**, *11*, 32. [CrossRef]
76. Ahmad, F.; Najam, A. Video-based face classification approach: A survey. In Proceedings of the 2012 International Conference of Robotics and Artificial Intelligence, Rawalpindi, Pakistan, 22–23 October 2012; IEEE: Piscataway, NJ, USA; pp. 179–186.
77. Ahmad, F.; Najam, A.; Ahmed, Z. Image-based face detection and recognition: "state of the art". *arXiv* **2013**, arXiv:1302.6379.
78. Ahmad, F.; Ahmed, Z.; Najam, A. *Soft Biometric Gender Classification Using Face for Real Time Surveillance in Cross Dataset Environment*; INMIC: Islamabad, Pakistan, 2013.
79. Su, X.; Yan, X.; Tsai, C.L. Linear regression. *Wiley Interdiscip. Rev. Comput. Stat.* **2012**, *4*, 275–294. [CrossRef]
80. Montgomery, D.C.; Peck, E.A.; Vining, G.G. *Introduction to Linear Regression Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2021.
81. Maulud, D.; Abdulazeez, A.M. A review on linear regression comprehensive in machine learning. *J. Appl. Sci. Technol. Trends* **2020**, *1*, 140–147. [CrossRef]
82. Li, X.; Huang, K.; Yang, W.; Wang, S.; Zhang, Z. On the convergence of fedavg on non-iid data. *arXiv* **2019**, arXiv:1907.02189.
83. Flower. *Flower a Friendly Federated Learning Framework*. 2022. Available online: https://fower.dev/ (accessed on 24 August 2022).
84. Zhou, Y.; Ye, Q.; Lv, J. Communication-efficient federated learning with compensated overlap-fedavg. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *33*, 192–205. [CrossRef]
85. Patro, V.M.; Patra, M.R. Augmenting weighted average with confusion matrix to enhance classification accuracy. *Trans. Mach. Learn. Artif. Intell.* **2014**, *2*, 77–91.
86. Das, K.; Jiang, J.; Rao, J.N.K. Mean squared error of empirical predictor. *Ann. Stat.* **2004**, *32*, 818–840. [CrossRef]
87. Willmott, C.J.; Matsuura, K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Clim. Res.* **2005**, *30*, 79–82. [CrossRef]
88. Chicco, D.; Warrens, M.J.; Jurman, G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Comput. Sci.* **2021**, *7*, e623. [CrossRef]