electronics

MDPI

Article

Federated Learning for Privacy-Preserving Basketball Game Outcome Classification: Key-Factor Analysis and Comparison with Centralised Machine Learning Models

Bita Farhadi [1], Md. Rajib Hasan [2,3,*], Noor H. S. Alani [3], Nurul I. Sarkar [2] and Huiguo Xu [1]

[1] EIT Data Science and Communication College, Zhejiang Yuexiu University, Shaoxing 312000, China; 20242705@zyufl.edu.cn (B.F.)
[2] Department of Computer and Information Sciences, Auckland University of Technology, Auckland 1010, New Zealand
[3] School of Computing, Eastern Institute of Technology, Napier 4110, New Zealand
* Correspondence: rhasan@eit.ac.nz or r.hasan@autuni.ac.nz

Abstract

Clubs, leagues and data vendors hold game-level performance data separately, and sharing such data is often restricted. This study examines which box-score differences are associated with National Basketball Association (NBA) game results and whether federated learning (FL) can match centralised models without pooling raw data. Using 49,899 games from the 1979–80 to 2024–25 seasons, we built home-minus-away differentials for 14 box-score statistics, measured their association with the result, and compared logistic regression (LR), support vector machine (SVM) and k-nearest neighbours (KNN) classifiers. Hyperparameters were chosen on the 2023–24 season, and the 2024–25 season was used only for testing. Because the inputs are recorded during the game being classified, the task is post-game outcome classification rather than pre-game forecasting. An FL framework was then simulated with the 30 franchises as clients. Field-goal percentage ($r = 0.62$) and field goals made ($r = 0.57$) showed the strongest associations with winning. All classifiers far exceeded the 0.549 accuracy of always predicting a home win; SVM reached 0.9081 and LR 0.9088, a difference that was not significant. Federated LR matched an identically configured centralised LR (paired difference 0.00 percentage points; 95% CI −0.43 to 0.43) and beat franchise-only models by 0.94 points. Under severe label skew, FedProx and a simple intercept recalibration recovered most of the lost accuracy. With client-level differential privacy, accuracy stayed above 0.90 at $\varepsilon \approx 4$. A pre-game variant based on rolling averages reached 0.63 accuracy, again with no loss under federation.

Keywords: federated learning; privacy preservation; differential privacy; non-independent and identically distributed data; distributed machine learning; communication-efficient learning; basketball analytics; game outcome classification; correlation analysis

Academic Editor: Juan Ernesto Solanes Galbis

Received: 4 November 2026
Revised: 21 November 2026
Accepted: 25 November 2026
Published: 26 November 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Big data technologies have made competition data far easier to collect and integrate, and machine learning is now an established tool in sports analytics [1–4]. Patterns found in large volumes of match data help researchers identify the factors that shape a game and

give coaches and athletes material for reviewing performance. Basketball, invented by James Naismith in 1891, is among the most widely followed sports in the world [5–10]. The National Basketball Association (NBA) has nearly eight decades of history, and leagues such as the EuroLeague in Europe and the Chinese Basketball Association (CBA) in Asia have helped spread the professional game internationally [11–15]. The NBA now has 30 teams, and every game produces detailed records that teams use for analysis and planning. Compared with conventional statistical summaries, machine learning offers a systematic way to clean, mine and model these records, including their temporal structure, in relation to win–loss results [16–20].

Buhamra et al. [21] used Grand Slam tennis match data to build regression-based prediction models for the four Grand Slam tournaments, demonstrating the feasibility of statistical and machine learning methods for sports performance prediction. He and Choi [22] introduced a stacked ensemble framework for NBA analytics that predicts game outcomes by combining several machine learning models, including naive Bayes, AdaBoost, multilayer perceptron (MLP), k-nearest neighbours (KNN), XGBoost, decision trees and logistic regression, to improve predictive performance.

Zhong [23] applied machine learning to forecast CBA game results for the 2021–2024 seasons. The Four Factors and Defence–Offence models and their extended versions were used with pre-game data only as input features, and several algorithms, including SVM, naive Bayes, KNN, logistic regression, MLP and XGBoost, were tested. The detailed Four Factors model with logistic regression achieved the highest accuracy (85.49%). Alves and Barbosa [24] investigated artificial intelligence in basketball analytics, focusing on the strengths and weaknesses of deep learning models for predicting game results. They focused on long short-term memory (LSTM) networks because they can capture the temporal sequence of game data. They showed that LSTM-based methods captured time-related information in NBA and Women's National Basketball Association (WNBA) data, yielding prediction accuracies of 65.50% and 67.48%, respectively. Gifford and Bayrak [25] applied decision trees and logistic regression to forecast National Football League (NFL) game results, with reported accuracies of 97% and 83%, respectively. Researchers have also explored several directions in basketball analytics, such as talent recognition [26], deep-learning-based free-throw prediction [27], and explainable artificial intelligence (AI) for NBA game prediction [28]. Song et al. [29] proposed a real-time prediction model that combines historical game data with bookmakers' betting lines, and Cai et al. [30] developed a hybrid ensemble learning framework that combines weighted feature construction, bagging strategies and Support Vector Machines to improve prediction accuracy and model generalisation.

However, all of these studies assume that game data can be pooled on a single server. Official box scores are public, but much of the information that clubs actually rely on is not: optical and wearable tracking, physiological and load-monitoring records, scouting reports and internally coded play data are commercially sensitive and are held separately by clubs, leagues and data vendors [7]. Sharing them raises confidentiality and data-governance concerns. Federated learning (FL) addresses this problem by training a shared model across several data holders while the raw data stay local; only model updates are exchanged and aggregated by a coordinating server [31–33]. FL has been studied widely in mobile, edge and healthcare settings, where statistical heterogeneity across clients, that is, non-independent and identically distributed (non-IID) data, is recognised as a central challenge for convergence and accuracy [32,34,35]. To the best of our knowledge, no study has examined whether basketball game outcomes can be modelled federatively with franchises acting as data silos, nor how the natural heterogeneity of club data, partial client participation and formal privacy protection affect performance relative to centralised learning.

https://doi.org/10.3390/electronics15235534

To address this gap, this study is guided by three research questions (RQs):

- **RQ1:** Which differences in technical statistics between opposing teams are most strongly associated with the outcome of an NBA game?
- **RQ2:** How do three classical classifiers—logistic regression, support vector machine and k-nearest neighbours—compare in accuracy, discrimination and temporal generalisation when classifying game outcomes from these differential features?
- **RQ3:** Can a federated model trained across franchise-level clients, without pooling raw data, match centralised performance, and how robust is it to non-IID data, partial participation and client-level differential privacy?

Correspondingly, the research objectives are (i) to construct a home-minus-away differential feature set and quantify its association with game results using Pearson and point-biserial correlation analyses; (ii) to tune and compare LR, SVM and KNN models with chronological model selection, a held-out test season and time-series cross-validation; and (iii) to design and evaluate a horizontal FL framework for game-outcome classification and to quantify its accuracy, sensitivity to heterogeneity, communication cost and privacy–utility trade-off.

The study makes three contributions. First, it provides a reproducible benchmark of LR, SVM and KNN on home-minus-away box-score differentials, with strictly chronological model selection and testing, a trivial home-win baseline, several test seasons, and an explicit check of which variables reveal the final score. The differential representation itself is a standard and intuitive choice; we use it because it keeps the federated model small and easy to interpret. Second, and more importantly, it presents what is, to our knowledge, the first federated learning study of basketball game outcomes in which each NBA franchise acts as a client. FedAvg [31] and FedProx [34] are evaluated under natural, IID and Dirichlet label-skewed partitions, partial client participation, intercept recalibration and client-level differential privacy that also covers the preprocessing statistics. Third, it states the threat model and the limits of the claims. The box-score task is post-game classification, the clients are simulated from one public dataset, and a pre-game variant built from rolling averages is reported to show how the framework behaves when only information available before the game is used.

The remainder of this paper is organised as follows. Section 2 describes the data, feature engineering, classifiers, federated learning framework and evaluation protocol. Section 3 presents the correlation analysis and the centralised and federated results. Section 4 discusses the findings in relation to the research questions and the study's limitations, and Section 5 concludes the paper.

2. Materials and Methods

2.1. Data Overview and Preprocessing

2.1.1. Data Description

We obtained NBA game data spanning the 1946–47 to 2024–25 seasons from the public Kaggle dataset "NBA Dataset: Box Scores and Stats (1947–Today)" compiled by E. Moore (see the Data Availability Statement). The dataset covers all games in each season, including the regular season, playoffs, conference finals and the NBA Finals. In this research, the matchup data between opposing teams are the primary object of study. The raw dataset contains 143,758 team-level records, corresponding to 71,879 games. Each record includes several statistical indicators, such as team score, opponent score, assists and blocks. Table 1 lists the main technical indicators used in the study.

Table 1. Technical statistical indicators.

Field Name	Full Name	Abbreviation
teamScore	Team Score	PTS
opponentScore	Opponent Score	Opp PTS
assists	Assists	AST
blocks	Blocks	BLK
steals	Steals	STL
fieldGoalsAttempted	Field Goals Attempted	FGA
fieldGoalsMade	Field Goals Made	FGM
fieldGoalsPercentage	Field Goal Percentage	FG%
threePointersAttempted	Three-Pointers Attempted	3PA
threePointersMade	Three-Pointers Made	3 PM
threePointersPercentage	Three-Point Percentage	3P%
freeThrowsAttempted	Free Throws Attempted	FTA
freeThrowsMade	Free Throws Made	FTM
freeThrowsPercentage	Free Throw Percentage	FT%
reboundsDefensive	Defensive Rebounds	DRB
reboundsOffensive	Offensive Rebounds	ORB
reboundsTotal	Total Rebounds	TRB
foulsPersonal	Personal Fouls	PF
turnovers	Turnovers	TOV
plusMinusPoints	Plus-Minus	+/−
numMinutes	Minutes Played	MIN

2.1.2. Data Preprocessing

Data cleaning aims to produce a consistent dataset without noisy or missing values. The pattern of missing values was examined first. Variables with a large proportion of missing values, as shown in Table 2, can harm model construction, so they were dropped. Because the NBA data cover a long history, early seasons differ in data completeness and playing style, and the cleaning had to be checked carefully. Of the 143,758 records, some items have a missing rate of up to 98% and low correlation with the dependent variable (Table 2); these items were removed. We also removed records without complete box-score statistics, which restricted the modelling dataset to games from the 1979–80 season onward: 99,798 team-level records, or 49,899 games. Of these, the 48,517 games played up to the end of the 2023–24 season form the training data and the 1382 games of the 2024–25 season form the independent test set (759 home-team wins and 623 away-team wins).

Table 2. Large-sample missing conditions.

Data Item	Null Count	Percentage
q1Points	141,008	98.09%
q2Points	141,008	98.09%
q3Points	141,010	98.09%
q4Points	141,010	98.09%
benchPoints	141,004	98.08%
biggestLead	141,004	98.08%
biggestScoringRun	141,004	98.08%
leadChanges	141,004	98.08%
pointsFastBreak	141,004	98.08%
pointsFromTurnovers	141,004	98.08%
pointsInThePaint	141,004	98.08%
pointsSecondChance	141,004	98.08%

timesTied	141,004	98.08%
timeoutsRemaining	141,004	98.08%
seasonWins	141,004	98.08%
seasonLosses	141,004	98.08%
coachId	143,758	100.00%

2.2. Feature Engineering

The models use the statistical differences between the two rival teams in each game. Each difference is obtained by subtracting the away team's statistic from the home team's statistic. According to the score difference, each game receives a binary label: 1 indicates a home-team win and 0 an away-team win. The procedure is as follows.

Step 1: Identification of home and away team labels. In game.csv, the home and away teams are identified by matching the primary keys of the team statistics file (gameId, teamCity + teamName, and opponentTeamCity + opponentTeamName) with the corresponding keys in game.csv (gameId, homeTeamCity/Name, and awayTeamCity/Name). The gameId uniquely identifies each game and is used to cross-match team information across the datasets. We compare the home-team columns of game.csv (homeTeamCity + homeTeamName) iteratively with the team-information columns (teamCity + teamName and opponentTeamCity + opponentTeamName) in the team statistics file, and we apply the same procedure to the away-team information (awayTeamCity + awayTeamName). When a match is found, the team is assigned a home/away label (teamType). Table 3 shows the resulting structure.

Table 3. Structure of the rule result section.

GameId	TeamName	TeamType	TeamScore	Assists
42400407	Indiana Pacers	Away Team	91	17
42400407	Oklahoma City Thunder	Home Team	103	20

Step 2: Validation of binary labels (0/1). Once the home and away labels have been assigned, the score difference for each game (differ_score) is computed as the home-team score minus the away-team score, and this value determines the outcome. When differ_score is greater than 0, the home team wins, and the game is labelled 1; otherwise, the away team wins, and the game is labelled 0. The resulting data are stored in TeamStatistics_differences_result.csv, part of whose structure is shown in Table 4.

Table 4. Partial structure of TeamStatistics_differences_result.csv.

GameId	HomeTeam	AwayTeam	Differ_Score	Result
42400407	Oklahoma City Thunder	Indiana Pacers	12	1

Step 3: Selection of model inputs. Fourteen differentials were used as inputs: assists, blocks, steals, field-goal attempts, three-point attempts, three-pointers made, three-point percentage, free-throw attempts, free throws made, free-throw percentage, offensive rebounds, total rebounds, personal fouls and turnovers. The score difference, the game identifier and the game date were excluded. Three further variables were excluded on purpose. Points satisfy PTS = 2·FGM + 3PM + FTM, and in our data this identity holds for 99.9% of games, so field goals made (FGM), together with the three-pointers and free throws made already in the set, would let a linear model rebuild the score difference, and hence the label. Field-goal percentage (FG% = FGM/FGA) combined with field-goal attempts recovers FGM almost exactly, and defensive rebounds equal total minus offensive rebounds in virtually every game, so DRB carries no information beyond TRB

and ORB. These variables are still analysed in the correlation study, and Section 3.5 reports the classifiers with them included.

2.3. Centralised Classifiers and Model Selection

All centralised models were implemented in Python (3.12.12) with scikit-learn [36]. Each game is represented by the vector x of its 14 differential features, with label y = 1 for a home win and y = 0 otherwise. The features were Z-score standardised:

$$x_{\text{scaled}} = \frac{x - \mu}{\sigma} \tag{1}$$

where μ and σ are the mean and standard deviation of each feature. They were always estimated on the training portion of the data in use: the training seasons for the final models and the training block of each cross-validation fold, so that no information from a validation or test period entered the scaling.

Logistic regression (LR) models the probability of a home win as $P(y = 1 \mid x) = 1/[1 + \exp(-(w^{\top}x + b))]$ and was fitted with the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) solver by minimising the L2-penalised log-loss

$$J(\beta) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)] + \lambda \sum_{j=1}^{14} \beta_j^2 \tag{2}$$

where p_i is the predicted probability for game i and λ sets the penalty strength (scikit-learn uses the equivalent inverse parameter C). Weighting the classes by their inverse frequency was treated as a further hyperparameter. The support vector machine (SVM) solves the soft-margin problem

$$\min_{w,b,\xi} \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \xi_i \tag{3}$$

subject to

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1,2, \ldots, n \tag{4}$$

with the radial basis function (RBF) kernel

$$K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2) \tag{5}$$

where C penalises margin violations and γ sets the kernel width. The AUC of the SVM was computed from its decision values; Platt scaling [37] is a monotone transformation and does not change this ranking. K-nearest neighbours (KNN) assigns a game to the majority class among its K nearest training games in Euclidean distance:

$$\hat{y}(x) = \underset{c \in C}{\text{argmax}} \sum_{j \in N_k(x)} I(y_j = c) \tag{6}$$

where $N_K(x)$ is the neighbour set and $I(\cdot)$ is the indicator function.

Model selection used only seasons that precede the test season. The 1979–80 to 2022–23 seasons (47,156 games) formed the training block and the 2023–24 season (1361 games) the validation block. Each grid point was fitted on the training block and scored on the validation block; the setting with the highest validation accuracy was then refitted on all seasons up to 2023–24 and evaluated once on the 2024–25 season, which played no part in any choice. Table 5 lists the grids, the selected values and their validation accuracy.

Table 5. Hyperparameter grids and selected values (selection on the 2023–24 validation season).

Model	Grid	Selected	Validation Accuracy
LR	C ∈ {0.001, 0.01, 0.1, 0.2, 0.5, 1, 2, 5, 10, 50, 100}; class weighting ∈ {none, balanced}	C = 0.5, balanced	0.9236
SVM (RBF)	C ∈ {0.1, 1, 10, 100}; γ ∈ {1/14, 0.01, 0.1}	C = 100, γ = 0.01	0.9192
KNN	K ∈ {1, 3, 5, …, 51}; Euclidean distance; uniform weights	K = 25	0.8935

2.4. Federated Learning Framework

2.4.1. Problem Formulation and Clients

A horizontal, cross-silo FL setting [32,33] was adopted. Each game was assigned to the franchise of its home team, so each client holds the differential feature vectors of the games it hosted. Because each vector contains statistics of both teams, this partition assumes that the host club records and keeps the complete box score of its home games; it is a modelling device for a cross-silo deployment rather than a description of how NBA data are owned today, and Section 3.6.4 compares it with assigning each game to the away franchise or to both franchises. Historical franchise names were mapped to their current franchises (for example, Seattle SuperSonics to Oklahoma City Thunder, New Jersey Nets to Brooklyn Nets, Vancouver Grizzlies to Memphis Grizzlies, Washington Bullets to Washington Wizards, Charlotte Bobcats to Charlotte Hornets, and New Orleans/Oklahoma City Hornets to New Orleans Pelicans), giving K = 30 clients. The federated objective is

$$\min_w F(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w), \qquad F_k(w) = \frac{1}{n_k} \sum_{i \in \mathcal{D}_k} \ell\,(w; x_i, y_i) + \frac{\lambda}{2} \| w \|^2 \tag{7}$$

where $\mathcal{D}_k$ is the local dataset of client k with n_k games, $n = \sum_k n_k$, and ℓ is the logistic loss (federated LR) or the hinge loss (federated linear SVM). Figure 1 shows the architecture. The client data are naturally non-IID: local training sizes range from 950 to 1871 games (quantity skew), and home-win rates range from 0.50 to 0.70 (label skew), as shown in Figure 2.

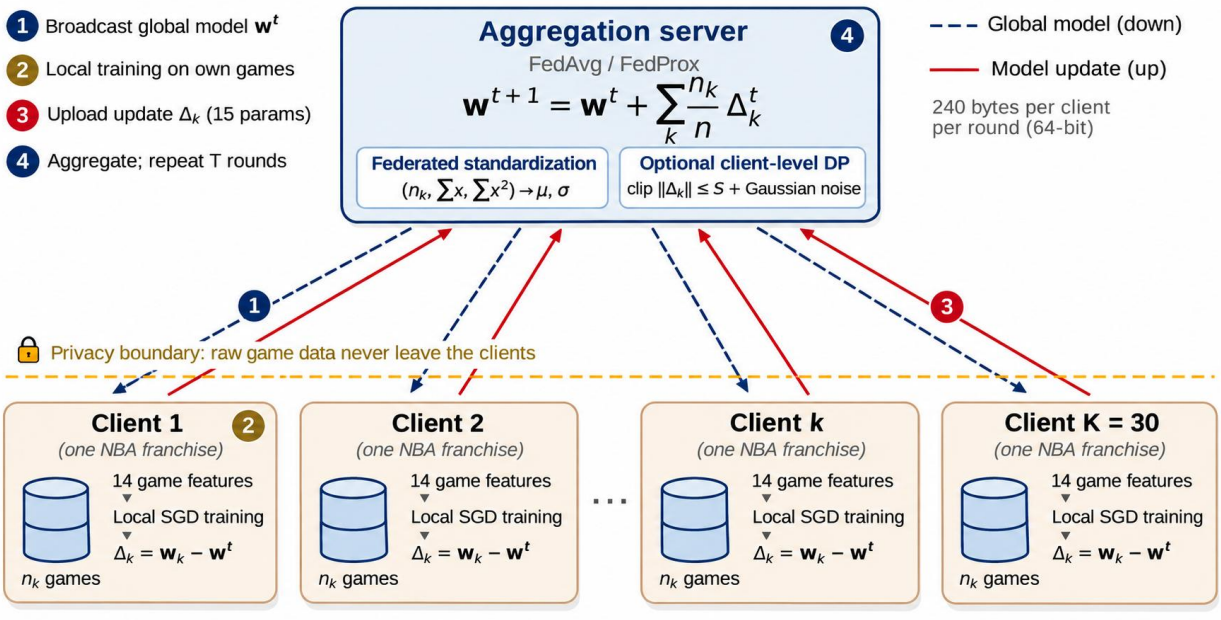


Figure 1. Proposed federated learning framework for basketball game outcome classification. Each franchise trains locally on its own games; only model updates (15 parameters) are exchanged with the aggregation server.

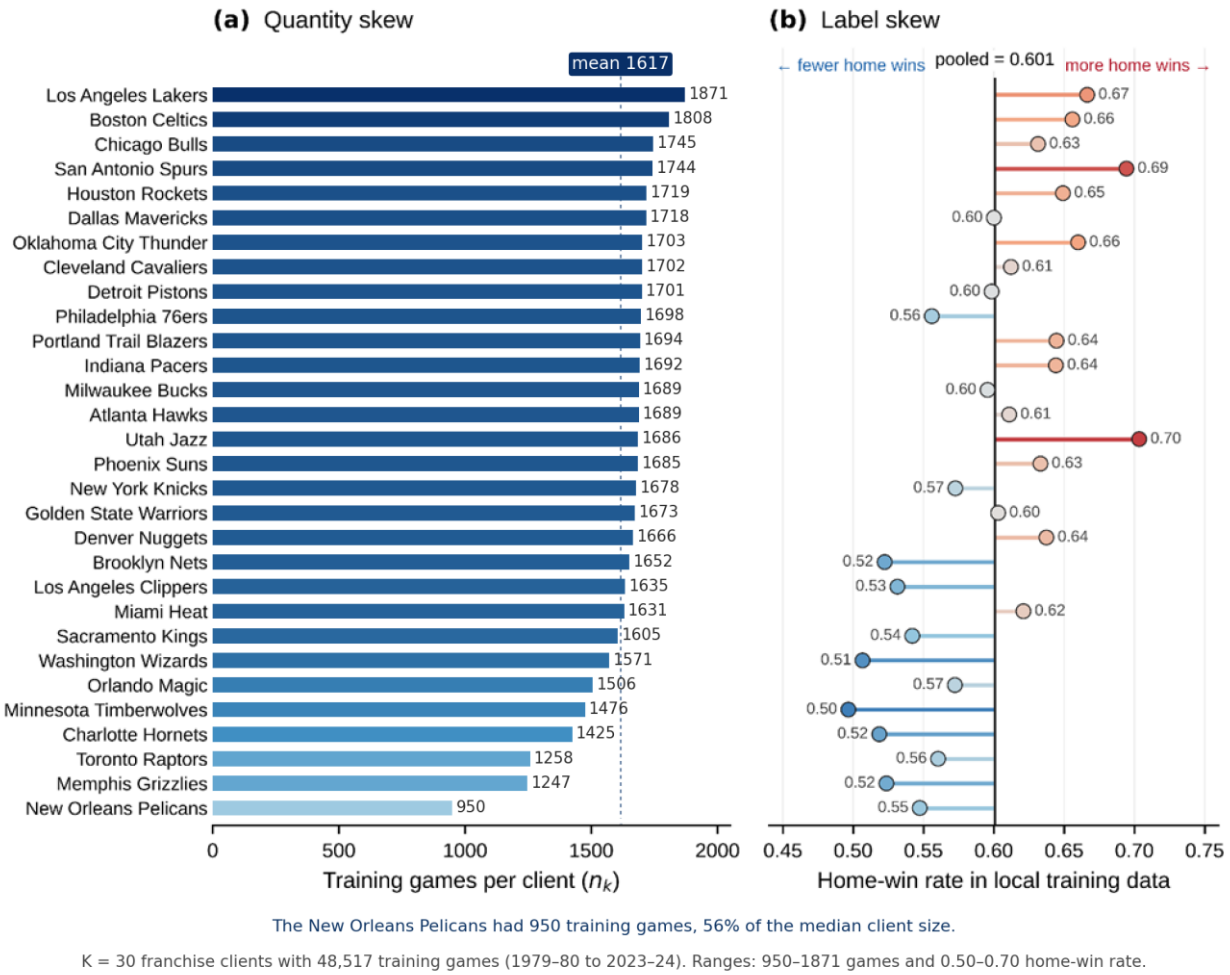


Figure 2. Statistical heterogeneity of the 30 franchise clients in the training data: (**a**) number of training games per client; (**b**) home-win rate per client (dashed line: pooled rate).

2.4.2. Federated Standardisation

To reproduce the Z-score standardisation of Equation (1) without sharing raw data, each client sends only the sufficient statistics $(n_k, \sum_i x_i, \sum_i x_i^2)$ of its features once, before training. The server computes the global mean $\mu = \sum_k \sum_i x_i / n$ and standard deviation $\sigma = \sqrt{\sum_k \sum_i x_i^2 / n - \mu^2}$, which are identical to the centralised values and are broadcast to all clients. In the non-private experiments, these statistics are released once in the clear; Section 2.4.5 describes how they are privatised.

2.4.3. FedAvg and FedProx

In each communication round t, the server broadcasts the global model w^t to a set S_t of m clients (all K clients unless stated otherwise). Each selected client runs E epochs of mini-batch stochastic gradient descent (batch size B, learning rate η) on its local objective and returns the update $\Delta_k^t = w_k^t - w^t$. In FedAvg [31], the server aggregates

$$w^{t+1} = w^t + \sum_{k \in S_t} \frac{n_k}{\sum_{j \in S_t} n_j} \Delta_k^t \quad (8)$$

FedProx [34] adds a proximal term to each local objective, $F_k(w) + \frac{\mu}{2} \| w - w^t \|^2$, which limits client drift under heterogeneous data.

2.4.4. Data Partitions

Three partitioning schemes were compared: (i) the *natural* franchise partition described above; (ii) an *IID* partition in which the training games are shuffled randomly and divided into 30 clients with the same sizes as the natural clients; and (iii) a *Dirichlet label-skew* partition [38], in which, for each class, the proportion assigned to each of the 30 clients is drawn from Dir(α) with $\alpha \in \{0.5, 0.1\}$; smaller α produces more severe non-IID data.

2.4.5. Client-Level Differential Privacy

To protect each franchise's entire contribution, DP-FedAvg [39] was applied. Each client update is clipped to an ℓ_2 norm of at most S, so one client changes the sum of updates by at most S. The server sums the clipped updates, adds Gaussian noise with standard deviation z·S and divides by the fixed number of participants m; this is equivalent to adding $\mathcal{N}(0, (zS/m)^2\mathbf{I})$ to the uniform average, where z is the noise multiplier. Aggregation in the DP runs is therefore uniform rather than weighted by n_k, so client sizes are not used. The cumulative privacy loss over T rounds is computed with Rényi differential privacy (RDP) [40]. For the Gaussian mechanism with sensitivity S, the RDP of order a per round is $a/(2z^2)$; composing over T rounds and converting to (ε, δ)-DP gives

$$\varepsilon = \min_{a>1} \left[\frac{Ta}{2z^2} + \frac{\log(1/\delta)}{a-1} \right] \tag{9}$$

No amplification from client subsampling is claimed [41], so the reported ε is an upper bound for a given noise multiplier. The guarantee is stated for add/remove-one-client adjacency [42]: two federations are neighbours if one contains an additional franchise with all of its games. Under replace-one-client adjacency, the sensitivity doubles, which is equivalent to halving z; both values are reported. The threat model is an honest-but-curious server that follows the protocol and sees every message the clients send, namely the standardisation statistics and each participating client's update in each round. Clients are assumed to follow the protocol. Secure aggregation [43], which would hide individual updates from the server, was not implemented, so FedAvg and FedProx on their own provide data locality but no formal privacy guarantee; the formal guarantee comes only from the DP mechanism. The standardisation statistics of Section 2.4.2 are also released under DP. Each client clips every game's features to fixed public bounds b (40 for assists, 25 for blocks and steals, 50 for field-goal and free-throw attempts and free throws made, 50 for three-point attempts and total rebounds, 30 for three-pointers made, offensive rebounds, fouls and turnovers, and 1 for the percentage differences; no training game exceeded these bounds), divides by b, and reports the vector of its local first and second moments clipped to an L2 norm of 0.25. The server averages these vectors uniformly and adds Gaussian noise with multiplier $z_s = z/\sqrt{(0.1T)}$, so that this one release costs as much RDP as 10% of the training rounds when T = 50. Client sizes are not released. The total privacy loss is the RDP sum of the T training rounds and this release.

2.4.6. Baselines and Experimental Settings

The federated models were compared with (i) a centralised LR fitted on the pooled training data with exactly the same 14 features, standardisation and objective as the federated LR (unweighted logistic loss with $\lambda = 10^{-4}$, which corresponds to $C = 1/(\lambda n) = 0.206$ in scikit-learn), solved to convergence with L-BFGS; (ii) the same objective optimised on a single pooled client with the federated SGD routine; and (iii) a local-only baseline in which each franchise trains an LR on its own games without collaboration and predicts its own 2024–25 home games. This matched reference differs from the class-weighted LR selected in Section 2.3; for the same features, the two configurations give 0.9038 (class-

weighted, C = 0.5) and 0.9088 (unweighted) on the test season, so the comparison with FL always uses the matched model. The federated hyperparameters are listed in Table 6. The learning rate was selected from {0.01, 0.05, 0.1, 0.2} using the 2023–24 season as a validation set (validation accuracies of 0.9001, 0.9184, 0.9192 and 0.9140, respectively). Every federated configuration was repeated with five random seeds, and the mean ± standard deviation is reported. Besides the global test metrics, the accuracy on each franchise's 2024–25 home games was computed to assess fairness across clients (mean, 10th percentile and worst client).

Table 6. Federated learning experimental settings.

Setting	Value
Clients (K)	30 franchises (home team of each game)
Training/test games	48,517 (1979–80 to 2023–24)/1382 (2024–25)
Features	14 standardised differential features + bias (15 parameters)
Algorithms	FedAvg, FedProx ($\mu \in \{0.01, 0.1\}$), DP-FedAvg
Local models	Logistic loss (LR), hinge loss (linear SVM); L2 penalty $\lambda = 10^{-4}$
Local training	E = 1 epoch (E = 5 in ablation and DP runs), B = 32, $\eta = 0.1$
Communication rounds	T = 100 (DP: T = 50 with E = 1, or T = 10 with E = 5)
Participation	m = 30 clients per round (m = 10 in ablation)
Partitions	Natural (home franchise), IID, Dirichlet $\alpha \in \{0.5, 0.1\}$; away-franchise and both-franchise assignments
DP parameters	Update clipping S = 0.5; $z \in \{1, 2, 4, 8, 16\}$; statistics clipping 0.25; $\delta = 10^{-5}$
Repetitions	5 random seeds

2.5. Evaluation Metrics and Validation Protocol

Accuracy, precision, recall, F1-score and the area under the receiver operating characteristic curve (AUC) were reported, with a home-team win as the positive class. Accuracy is the share of correctly classified games; precision is the share of predicted home wins that were home wins; recall is the share of actual home wins that were identified; F1 is the harmonic mean of precision and recall; and AUC measures how well the model ranks home wins above away wins, independently of the decision threshold. Because home teams win more often, every result is compared with the trivial rule that always predicts a home win, which scores 0.5492 on the 2024–25 season.

The data were split chronologically into a training block, a validation block (the 2023–24 season, used for every hyperparameter choice) and a test block (the 2024–25 season, used once). Temporal generalisation was assessed in two further ways. First, five-fold time-series cross-validation was run on the training seasons; in each fold, the earlier games were used for training and the next block for evaluation, and standardisation was refitted inside each fold. Second, a rolling-origin evaluation used each of the 2020–21 to 2024–25 seasons in turn as the test season, with all earlier seasons as training data. Differences in accuracy between two models evaluated on the same games were assessed with a paired bootstrap over games (5000 resamples, 95% percentile intervals) and with the exact McNemar test on the discordant games [44].

3. Results

3.1. Correlation Analysis

3.1.1. Pearson Correlation Analysis

Pearson correlation analysis was used to describe how the team-level box-score variables relate to the team score Y and to each other:

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \quad (10)$$

The Pearson coefficient quantifies the linear relationship between variables and ranges from −1 to 1. A coefficient with $|r| \geq 0.7$ is usually regarded as strong. As depicted in Figure 3, the variables most strongly linked to points are those that make up scoring: field goals made (0.85), field-goal percentage (0.67) and assists (0.61). Assists accompany most made baskets, so their correlation with points is expected. Turnovers are the only variable with a negative correlation with scoring (−0.12), which is consistent with possessions lost before a shot can be taken.

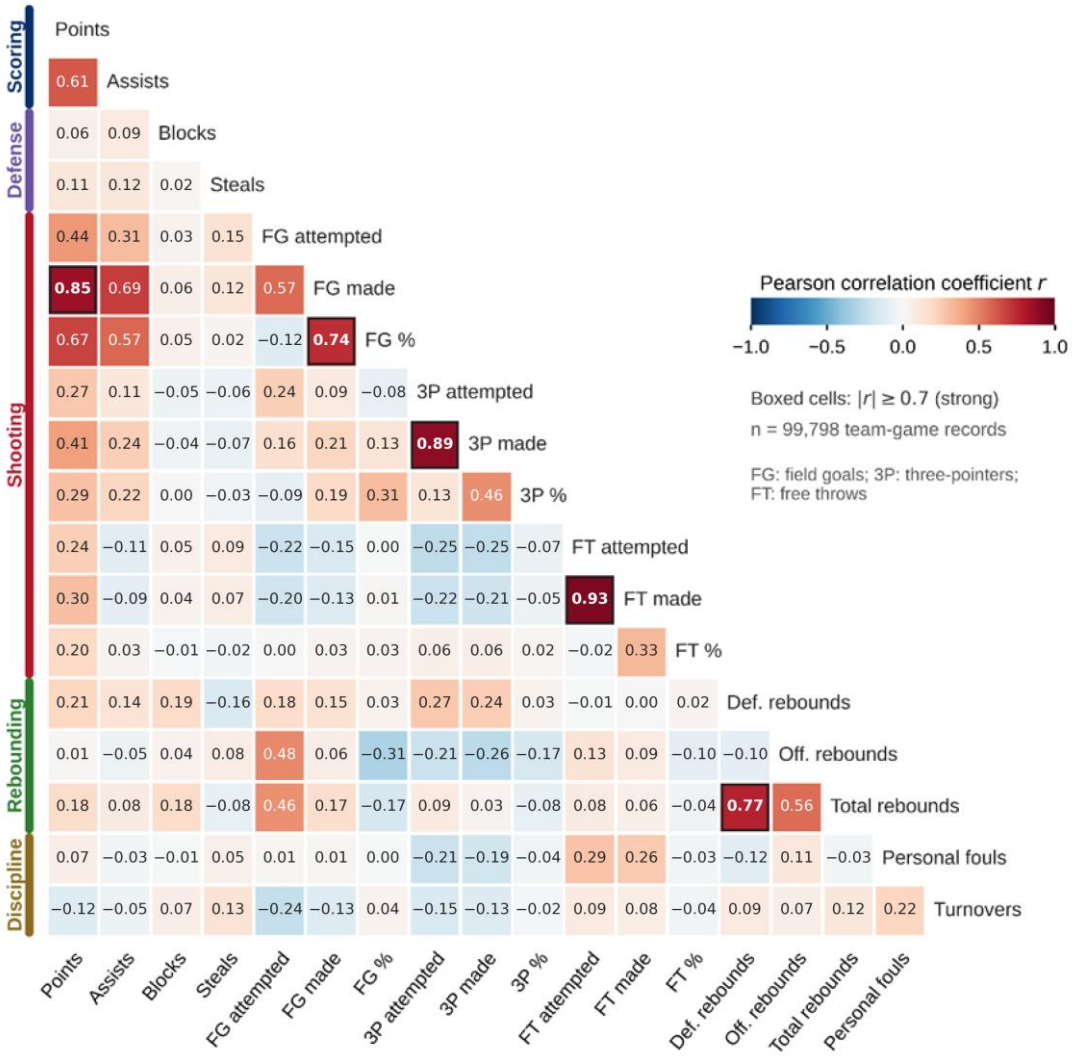


Figure 3. Pearson correlation coefficient heatmap.

The correlation analysis also reveals multicollinearity among several independent variables (Table 7), which can make regression coefficients unstable. Much of it is arithmetic: three-pointers made are bounded by three-point attempts, free throws made by free-throw attempts, and total rebounds contain defensive rebounds.

Table 7. Partial multicollinearity.

Feature Pair	Coefficient	Strength
3-Point Made–3-Point Attempted	0.89	Very strong
Free Throws Made–Free Throws Attempted	0.93	Very strong
Total Rebounds–Defensive Rebounds	0.77	Strong
Points–Field Goals Made	0.85	Very strong

3.1.2. Point-Biserial Correlation Analysis

The classification task distinguishes home wins (result = 1) from away wins (result = 0). Before modelling, the association between each feature and the result was measured.

Compared with Pearson correlation, point-biserial correlation is better suited to combinations of a binary and a continuous variable. Based on the mean difference, the standard deviation and the sample sizes, it evaluates the relationship as follows:

$$r_{pb} = \frac{\bar{X}_1 - \bar{X}_0}{s} \sqrt{\frac{n_1 n_0}{n(n-1)}} \tag{11}$$

where $\bar{X}_1$ and $\bar{X}_0$ are the means of the continuous variable when the binary variable equals 1 and 0, s is the standard deviation of the continuous variable, n_1 and n_0 are the numbers of samples in each class, and n is the total number of samples.

As shown in Table 8, the score difference (differ_score) has the strongest association with the result (r = 0.7951) because it defines the label; it was therefore excluded from model training. The difference in field-goal percentage (differ_fieldGoalsPercentage, r = 0.6186) and in field goals made (differ_fieldGoalsMade, r = 0.5719) follow, then defensive rebounds (differ_reboundsDefensive, r = 0.5068) and assists (r = 0.4511). Personal fouls (−0.2205) and turnovers (−0.1706) are weakly negatively associated with the result, and field-goal attempts show almost no association (−0.0987). These coefficients describe co-occurrence within the same game and should not be read as causal effects. Several of the variables are mechanically tied to scoring: made shots are points, and a team that is trailing late in a game tends to foul deliberately and attempt more shots. We therefore use them to describe which box-score margins accompany winning, not to recommend tactics.

Table 8. Correlation between feature variables and the dependent variable.

Variable Name	r	Strength
differ_score	0.7951	Strong positive
differ_fieldGoalsPercentage	0.6186	Moderate positive
differ_fieldGoalsMade	0.5719	Moderate positive
differ_reboundsDefensive	0.5068	Moderate positive
differ_assists	0.4511	Weak positive
differ_reboundsTotal	0.3522	Weak positive
differ_threePointersPercentage	0.3256	Weak positive
differ_freeThrowsMade	0.2747	Weak positive
differ_threePointersMade	0.2705	Weak positive
differ_freeThrowsAttempted	0.2314	Weak positive
differ_foulsPersonal	−0.2205	Weak negative
differ_blocks	0.2149	Weak positive
differ_steals	0.2081	Weak positive
differ_turnovers	−0.1706	Weak negative
differ_freeThrowsPercentage	0.1514	Weak positive
differ_fieldGoalsAttempted	−0.0987	Weak negative
differ_reboundsOffensive	−0.0625	Weak negative

differ_threePointersAttempted	−0.0339	Weak negative

3.2. Hyperparameter Selection

Figure 4 shows the validation accuracy (2023–24 season) across the three grids. For LR, the class-weighted model was best for C ≥ 0.5 (0.9236), while the unweighted model peaked at C = 0.01 (0.9199); accuracy changed by less than 0.01 for C between 0.1 and 100. For KNN, accuracy rose steeply up to K ≈ 15 and then fluctuated within about 0.01, with the maximum at K = 25 (0.8935). For the RBF SVM, $\gamma = 0.01$ gave the best and most stable results at every C, and C = 100 was marginally best (0.9192).

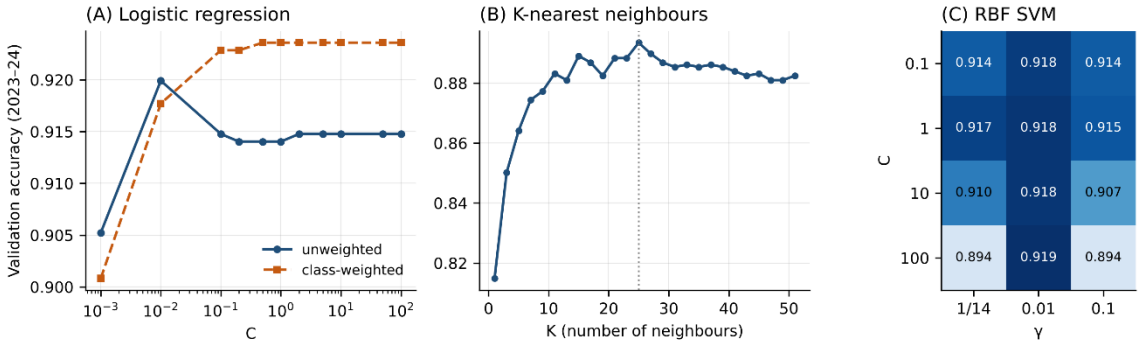


Figure 4. Hyperparameter selection on the 2023–24 validation season: (**A**) LR accuracy versus C with and without class weighting; (**B**) KNN accuracy versus K (dotted line: selected K); (**C**) RBF SVM accuracy over C and γ.

3.3. Centralised Classification Performance

Figure 5 shows the confusion matrices of the three selected models on the 2024–25 season (623 away wins and 759 home wins), and Table 9 reports the metrics together with the always-home baseline and the unweighted LR used as the federated reference. All classifiers were far above this baseline, which reflects the home advantage common to team sports [45]: the weakest, KNN, exceeded it by 33.79 percentage points. SVM reached the highest accuracy among the selected models (0.9081, F1 0.9173) and the selected class-weighted LR the highest precision (0.9253) and AUC (0.9704). However, the paired comparisons do not support a ranking of SVM and LR. SVM was 0.43 points ahead of the selected LR (95% CI −0.43 to 1.37; McNemar $p = 0.44$), and the unweighted LR, which scored 0.9088, was 0.07 points ahead of SVM ($p = 1.00$). Only KNN was clearly worse, by 2.10 points relative to SVM ($p = 0.005$) and 1.66 points relative to LR ($p = 0.040$). KNN made most of its additional errors on away wins (94 false home wins against 55 for LR), which is consistent with its sensitivity to the denser home-win region of the feature space. The class-weighted LR shifts errors from away wins to home wins: it misses more home wins but raises fewer false alarms, which gives it the highest precision and the lowest recall of the three.

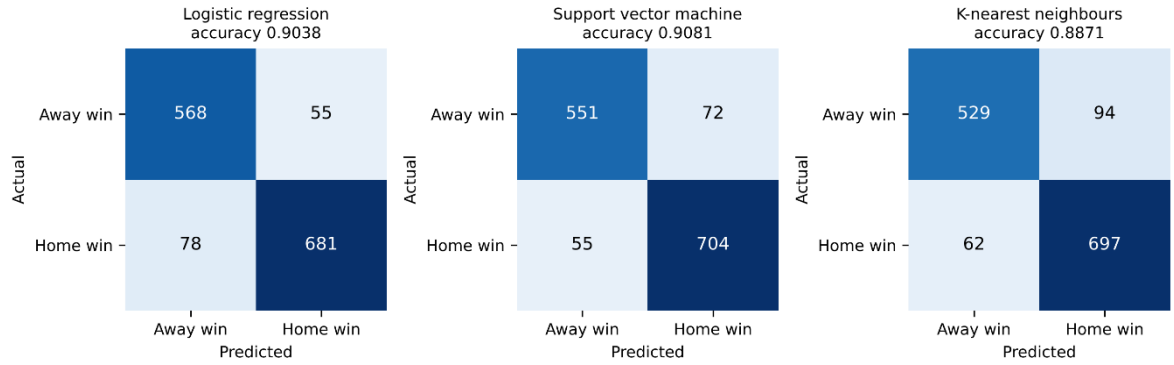


Figure 5. Confusion matrices of the selected centralised models on the 2024–25 test season.

Table 9. Centralised classification results on the 2024–25 test season.

Model	Accuracy	Precision	Recall	F1-Score	AUC
Always predict home win	0.5492	0.5492	1.0000	0.7090	0.5000
LR (class-weighted, C = 0.5)	0.9038	0.9253	0.8972	0.9110	0.9704
LR (unweighted, C = 0.206) [1]	0.9088	0.9137	0.9209	0.9173	0.9703
KNN (K = 25)	0.8871	0.8812	0.9183	0.8994	0.9543
SVM (RBF, C = 100, γ = 0.01)	0.9081	0.9072	0.9275	0.9173	0.9691

[1] Same objective as the federated LR (Section 2.4.6); not selected on the validation season.

3.4. Temporal Generalisation of the Centralised Models

Table 10 summarises the five-fold time-series cross-validation, with standardisation refitted inside each fold. The fold means were 0.9005 ± 0.0073 for LR, 0.9024 ± 0.0077 for SVM and 0.8608 ± 0.0113 for KNN, and AUC stayed above 0.92 in every fold. Accuracy rose slightly from the earliest to the latest fold for all models, as more training seasons became available.

Table 10. Five-fold time-series cross-validation on the training seasons (mean ± standard deviation over folds; scaling fitted within each fold).

Model	Accuracy	Precision	Recall	F1-Score	AUC
LR	0.9005 ± 0.0073	0.9417 ± 0.0028	0.8864 ± 0.0073	0.9132 ± 0.0049	0.9695 ± 0.0043
SVM	0.9024 ± 0.0077	0.9129 ± 0.0066	0.9230 ± 0.0061	0.9179 ± 0.0049	0.9674 ± 0.0059
KNN	0.8608 ± 0.0113	0.8591 ± 0.0120	0.9146 ± 0.0067	0.8859 ± 0.0066	0.9360 ± 0.0087

A single test season gives limited evidence about temporal robustness, so Table 11 repeats the evaluation for five consecutive test seasons. LR and SVM stayed between 0.898 and 0.924, KNN remained 2–4 points lower, and the home-win baseline ranged from 0.541 to 0.573. Neither LR nor SVM was better in every season, which supports the conclusion of Section 3.3 that the two are practically equivalent on this task. The federated columns are discussed in Section 3.6.1.

Table 11. Rolling-origin evaluation: accuracy on each test season with all earlier seasons used for training (federated results: mean over three seeds).

Test Season	*n*	Always Home	LR	SVM	KNN	LR (Matched)	FedAvg-LR	Local-Only LR
2020–21	1354	0.5406	0.9025	0.9062	0.8708	0.9077	0.9047	0.9010
2021–22	1373	0.5528	0.9104	0.9119	0.8747	0.9119	0.9097	0.8980
2022–23	1372	0.5729	0.9016	0.8980	0.8688	0.9052	0.9021	0.8929
2023–24	1361	0.5481	0.9236	0.9192	0.8935	0.9140	0.9187	0.9060
2024–25	1382	0.5492	0.9038	0.9081	0.8871	0.9088	0.9088	0.8994

Mean	6842	0.5527	0.9084	0.9087	0.8790	0.9095	0.9088	0.8995

3.5. Effect of the Excluded Variables

Table 12 adds the three excluded variables to the LR inputs, keeping the selected hyperparameters. Adding field-goal percentage raised the accuracy from 0.9038 to 0.9884, and adding field goals made gave perfect classification (1.0000) because, with three-pointers and free throws already present, the model can reconstruct the score difference. Adding defensive rebounds changed nothing, as expected from DRB = TRB − ORB. The excluded variables would therefore have inflated the reported accuracies through near-direct access to the final score; the base set leaves the model to infer the result from the other aspects of play.

Table 12. LR results on the 2024–25 season with the excluded variables added.

Input Set	Accuracy	AUC
Base set (14 features)	0.9038	0.9704
+FG%	0.9884	0.9996
+FGM	1.0000	1.0000
+DRB	0.9038	0.9703
+FG%, FGM, DRB	1.0000	1.0000

3.6. Federated Learning Results

3.6.1. Federated Versus Centralised and Local-Only Training

Table 13 summarises the federated results on the 2024–25 test season. With identical features, standardisation and objective, the centralised LR solved with L-BFGS reached 0.9088 accuracy and an AUC of 0.9703, and the same objective optimised on one pooled client with our SGD routine reached 0.9096. FedAvg over the 30 franchise clients reached 0.9090 ± 0.0003. Its predictions agreed with those of the centralised model on 99.4% of test games; of the eight discordant games, each model classified four correctly, so the paired difference was 0.00 points (95% CI −0.43 to 0.43; McNemar $p = 1.00$). Over the five rolling test seasons of Table 11 (6842 games), the pooled difference between FedAvg and the matched centralised LR was −0.07 points (95% CI −0.26 to 0.12), and the per-season difference ranged from −0.37 to +0.51 points. Franchises training only on their own games reached 0.8994. FedAvg was 0.94 points better on the 2024–25 season (95% CI 0.14 to 1.74; McNemar $p = 0.035$) and 0.94 points better over the five seasons (95% CI 0.51 to 1.37; $p < 0.001$). The federated model was also better for the worst-served franchise (0.8148 vs. 0.8085) and at the 10th percentile of clients (0.8460 vs. 0.8283). A federated linear SVM (hinge loss) performed similarly (0.9082 ± 0.0005). In short, collaboration without pooling recovered the accuracy of pooled training within about half a percentage point, and the benefit over isolated training, although small, was consistent across seasons.

Table 13. Federated learning results on the 2024–25 test season (mean ± standard deviation over five seeds; client columns are accuracies on each franchise's home games).

Configuration	Accuracy	Prec.	Recall	F1	AUC	Client Mean	Client 10th Pct.	Worst Client
Centralised LR, same objective (L-BFGS)	0.9088	0.9137	0.9209	0.9173	0.9703	0.9090	0.8418	0.8148
Centralised LR, same objective (SGD, one client)	0.9096	0.9171	0.9183	0.9177	0.9708	–	–	–
Local-only LR (no collaboration)	0.8994	0.9079	0.9091	0.9085	0.9683	0.8996	0.8283	0.8085
FedAvg-LR, natural	0.9090 ± 0.0003	0.9140	0.9209	0.9174	0.9702	0.9090	0.8460	0.8148
FedProx-LR ($\mu = 0.01$), natural	0.9090 ± 0.0003	0.9140	0.9209	0.9174	0.9702	0.9090	0.8460	0.8148
FedProx-LR ($\mu = 0.1$), natural	0.9080 ± 0.0003	0.9127	0.9204	0.9166	0.9701	0.9082	0.8390	0.8148

FedAvg-LR, IID	0.9090 ± 0.0005	0.9142	0.9207	0.9174	0.9703	0.9090	0.8446	0.8148
FedAvg-LR, Dirichlet $\alpha = 0.5$	0.9049 ± 0.0043	0.8976	0.9339	0.9152	0.9704	0.9047	0.8382	0.7874
FedAvg-LR, Dirichlet $\alpha = 0.1$	0.8890 ± 0.0113	0.8581	0.9581	0.9048	0.9704	0.8884	0.8216	0.7706
FedProx-LR ($\mu = 0.1$), Dirichlet $\alpha = 0.1$	0.9033 ± 0.0025	0.8919	0.9378	0.9142	0.9703	0.9028	0.8396	0.7805
FedAvg-LR, natural, m = 10	0.9087 ± 0.0008	0.9139	0.9204	0.9172	0.9703	0.9088	0.8418	0.8148
FedAvg-LR, natural, E = 5	0.9082 ± 0.0003	0.9136	0.9199	0.9168	0.9702	0.9084	0.8418	0.8049
FedAvg linear SVM, natural	0.9082 ± 0.0005	0.9123	0.9215	0.9169	0.9702	0.9083	0.8432	0.8000

3.6.2. Effect of Non-IID Data

The natural franchise partition behaves almost like the IID partition because its label skew is mild (home-win rates of 0.50–0.70; Figure 2). Synthetic label skew had a clear effect (Figure 6): with Dirichlet $\alpha = 0.5$, accuracy fell to 0.9049 ± 0.0043, and with $\alpha = 0.1$, it fell to 0.8890 ± 0.0113, with higher variance across seeds and a worst-client accuracy of 0.7706. The AUC remained unchanged (0.9704), while precision fell and recall rose (0.8581 and 0.9581 for $\alpha = 0.1$), which suggests that the global model still ranked games correctly but placed its decision threshold too low. FedProx with $\mu = 0.1$ recovered accuracy to 0.9033 ± 0.0025.

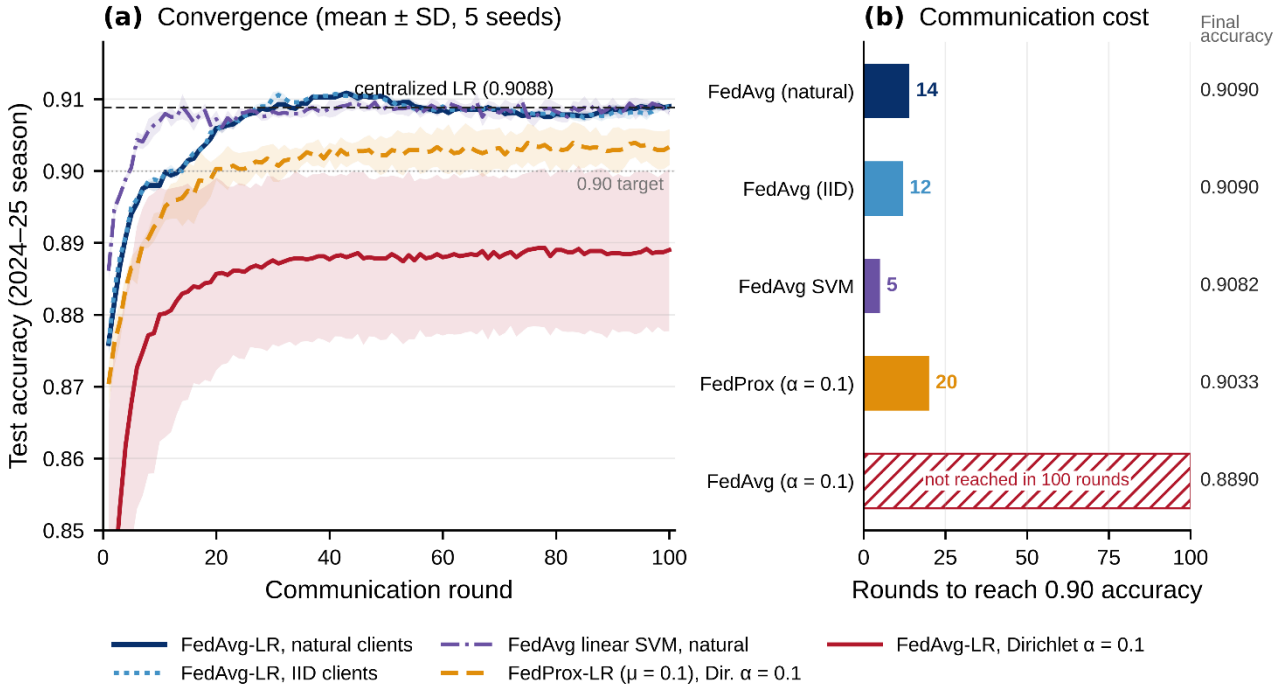


Figure 6. Test accuracy versus communication round for federated logistic regression and linear SVM under different client partitions (mean of five seeds). The dashed horizontal line is the centralised LR reference.

To test whether the loss is mainly an intercept shift, and whether FedProx does more than correct it, we trained on the seasons up to 2022–23, used the 2023–24 season as validation data held by the clients, and compared two simple post hoc corrections with FedProx (Table 14). In the intercept correction, the weights are frozen and only the bias is refitted by a one-dimensional Newton step on the validation log-loss; each client sends the gradient and curvature of its validation games, two numbers per round. In the threshold correction, the cut-off that maximises validation accuracy is chosen. Under $\alpha = 0.1$, refitting the intercept raised FedAvg from 0.9016 ± 0.0065 to 0.9074 ± 0.0008, essentially the accuracy of FedProx (0.9075), and refitting the intercept of FedProx added nothing (0.9074). The learned intercept moved from 1.42 under FedAvg to 1.21 after refitting, close

to the centralised value of 1.26, while the AUC did not change. The benefit of the proximal term under this kind of label skew is therefore largely a correction of the intercept, which a two-scalar recalibration achieves equally well. Threshold tuning on the validation season was slightly worse than either, probably because the chosen cut-off fits the validation season too closely. On the natural partition, neither correction helped.

Table 14. Accuracy on the 2024–25 season after training on seasons up to 2022–23, without and with post hoc recalibration on the 2023–24 season (mean ± standard deviation over five seeds; centralised LR: 0.9088).

Partition	Algorithm	No Correction	Intercept Refit	Threshold Tuned	AUC
Natural	FedAvg	0.9088 ± 0.0000	0.9071 ± 0.0004	0.9054 ± 0.0003	0.9702
Natural	FedProx (μ = 0.1)	0.9087 ± 0.0005	0.9077 ± 0.0004	0.9036 ± 0.0014	0.9701
Dirichlet α = 0.5	FedAvg	0.9045 ± 0.0025	0.9065 ± 0.0008	0.9039 ± 0.0010	0.9703
Dirichlet α = 0.5	FedProx (μ = 0.1)	0.9069 ± 0.0013	0.9068 ± 0.0003	0.9052 ± 0.0005	0.9702
Dirichlet α = 0.1	FedAvg	0.9016 ± 0.0065	0.9074 ± 0.0008	0.9035 ± 0.0011	0.9702
Dirichlet α = 0.1	FedProx (μ = 0.1)	0.9075 ± 0.0008	0.9074 ± 0.0009	0.9049 ± 0.0012	0.9702

3.6.3. Communication Efficiency and Partial Participation

Each client exchanges only the 15 model parameters per round (240 bytes upstream and downstream in 64-bit precision), so 100 rounds with all 30 clients require 720 kB in total, roughly one-eighth of the size of the training feature matrix itself (5.8 MB), which never leaves the clients. FedAvg-LR first exceeded 0.90 test accuracy after 14 rounds with the natural partition (12 rounds IID). Sampling only m = 10 of the 30 clients per round reduced per-round communication by two-thirds with negligible loss (0.9087 ± 0.0008). Increasing local computation to E = 5 epochs reached 0.90 accuracy after only three rounds, with a final accuracy of 0.9082, and the federated linear SVM converged within five rounds, illustrating the trade-off between local computation and communication.

3.6.4. Sensitivity to the Data-Holding Assumption

Because the main partition gives each game to its home franchise, we repeated FedAvg with each game assigned to the away franchise and with each game held by both franchises. Accuracy was 0.9098 ± 0.0007 and 0.9083 ± 0.0003, respectively, against 0.9088 for the home partition, with the same AUC (0.970). The conclusions therefore do not depend on which club is assumed to hold a game. Real deployments would add differences that a single public dataset cannot reproduce, such as club-specific tracking systems, data schemas, missing-data patterns and access policies; these are discussed in Section 4.

3.6.5. Privacy–Utility Trade-Off

Table 15 and Figure 7 report client-level DP-FedAvg. A noise-free run of the DP code (0.9107 at T = 50) differs slightly from the main FedAvg result (0.9090 at T = 100) for two reasons: stopping at 50 rounds gives 0.9101 with ordinary weighted FedAvg, and clipping with uniform averaging gives 0.9107. Both differences are within 0.002 and reflect the optimisation path rather than a change in the model. With one local epoch and 50 rounds, noise multipliers of 1 and 2 ($\varepsilon \approx 59$ and 23) cost almost nothing but offer little protection. The single-digit regime is more relevant: z = 8 gave $\varepsilon \approx 4.6$ and 0.8987 accuracy, and z = 16 gave $\varepsilon \approx 2.2$ and 0.8787. Because ε grows with the number of rounds, five local epochs and ten rounds gave a better trade-off: 0.9016 at $\varepsilon \approx 4.1$, 0.8919 at $\varepsilon \approx 2.0$ and 0.8425 at $\varepsilon \approx 1.0$ (add/remove adjacency; the replace-one values are roughly twice as large). Privatising the standardisation statistics as well was cheap at moderate budgets (for example 0.9072 at $\varepsilon \approx 24.7$) but costly in the single-digit regime (0.8671 at $\varepsilon \approx 4.3$), because with only 30 clients,

the noise on the second moments distorts the feature scales. With 30 clients, each franchise's update carries a large weight in the average, so single-digit guarantees cost up to about three points of accuracy for $\varepsilon \geq 2$ and more below that. Because the noise is added to an average over m clients, a federation with more clients (for example, several leagues) would reach the same accuracy with a larger noise multiplier and therefore a smaller ε; for a fixed noise multiplier and number of rounds, ε itself does not depend on the number of clients.

Table 15. Privacy–utility trade-off of client-level DP-FedAvg on the 2024–25 season ($S = 0.5$, $\delta = 10^{-5}$; mean ± standard deviation over five seeds). ε is given for add/remove-one-client adjacency, with replace-one adjacency in brackets.

E, T	z	ε (Updates Only)	Accuracy	AUC	ε (Updates + Statistics)	Accuracy
1, 50	1	58.9 (167.9)	0.9091 ± 0.0017	0.9696	63.1 (181.2)	0.9081 ± 0.0019
1, 50	2	23.2 (58.9)	0.9072 ± 0.0010	0.9696	24.7 (63.1)	0.9072 ± 0.0010
1, 50	4	10.0 (23.2)	0.9041 ± 0.0023	0.9688	10.6 (24.7)	0.8874 ± 0.0171
1, 50	8	4.6 (10.0)	0.8987 ± 0.0034	0.9659	4.9 (10.6)	0.8472 ± 0.0459
1, 50	16	2.2 (4.6)	0.8787 ± 0.0041	0.9546	2.3 (4.9)	0.7208 ± 0.0815
5, 10	1	20.2 (50.3)	0.9062 ± 0.0014	0.9685	21.4 (53.8)	0.9067 ± 0.0026
5, 10	2	8.8 (20.2)	0.9049 ± 0.0031	0.9682	9.3 (21.4)	0.8939 ± 0.0105
5, 10	4	4.1 (8.8)	0.9016 ± 0.0042	0.9672	4.3 (9.3)	0.8671 ± 0.0315
5, 10	8	2.0 (4.1)	0.8919 ± 0.0115	0.9623	2.1 (4.3)	0.7544 ± 0.0482
5, 10	16	1.0 (2.0)	0.8425 ± 0.0343	0.9267	1.0 (2.1)	0.6716 ± 0.1216

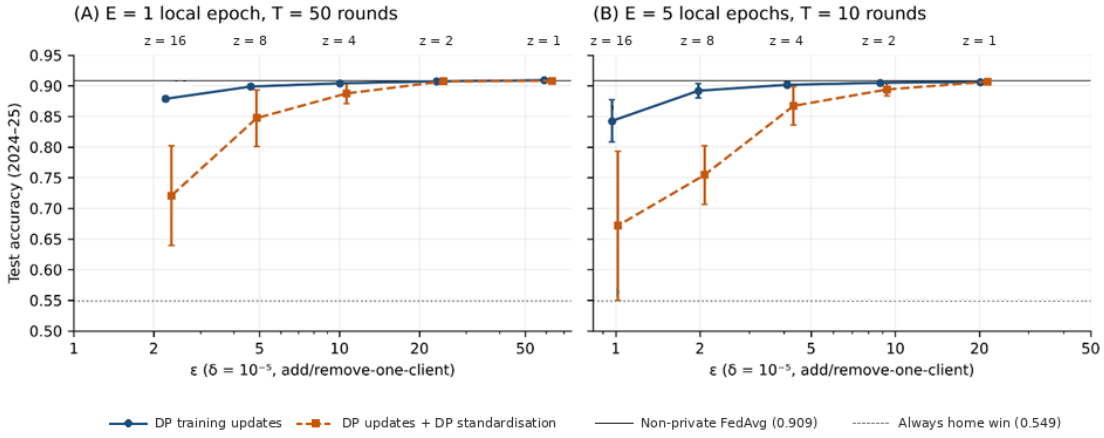


Figure 7. Privacy–utility trade-off of client-level DP-FedAvg: test accuracy versus ε (log scale) for (**A**) one local epoch and 50 rounds and (**B**) five local epochs and 10 rounds, with exact or differentially private standardisation statistics. Error bars show the standard deviation over five seeds.

3.7. Pre-Game Variant

To show how the framework behaves when only pre-game information is used, each team's mean statistics over its previous ten games (minimum five, computed strictly before the game date and across season boundaries) were differenced in the same home-minus-away manner. The 18 inputs were the 14 statistics of the main model plus points scored, points conceded, win rate and scoring margin. With the same split and settings (48,392 training and 1382 test games), the matched centralised LR reached 0.6331 accuracy (AUC 0.6832) and FedAvg 0.6336 ± 0.0017 (AUC 0.6833), against 0.5492 for always predicting a home win. This is in the range reported for pre-game NBA models [24] and

well below the box-score results, which confirms that the 0.90 accuracies above come from in-game information. It also shows that federation costs nothing in the pre-game setting.

4. Discussion

With respect to RQ1, shooting efficiency and possession-related statistics showed the strongest associations with the result. Field-goal percentage and field goals made had the largest point-biserial correlations, followed by defensive rebounds and assists, while turnovers and personal fouls were weakly negatively associated with winning. This ordering agrees with the emphasis on shooting efficiency, rebounding and ball security in earlier work [22,23,30] and with the Four Factors framework [46]. It is, however, an association measured within the same game, and several variables are partly a consequence of the score rather than a cause of it. The LR weights must also be read with care because of multicollinearity (Table 7): in the matched LR, total rebounds receive a large positive weight (4.37) and offensive rebounds a large negative weight (−3.12), which together act as the defensive-rebound effect that the model was not given directly.

With respect to RQ2, all three classifiers were far above the always-home baseline on the unseen 2024–25 season. SVM and LR were statistically indistinguishable, both on the test season and across five rolling seasons, while KNN was consistently 2–4 points lower. The class-weighted LR selected on the validation season trades recall for precision, while the unweighted LR has the higher accuracy on 2024–25; the difference between them was not significant. LR remains attractive because it is interpretable and needs only 15 parameters, which is what makes it cheap to federate.

With respect to RQ3, franchise-level collaboration without data pooling matched centralised training on this task: the paired difference was within half a point on the test season and across 6842 games from five seasons, and federation was consistently better than isolated per-club training. This agrees with theory and evidence that FedAvg performs well when client heterogeneity is moderate [31,34]. Under strong synthetic label skew, the expected degradation appeared [35,38], but mainly as an intercept shift; FedProx corrected it, and so did a two-scalar recalibration on validation data. Personalised schemes that keep part of the model local, such as the mutual-learning design that Wang et al. [47] use for computed tomography sites with different scanners and protocols, are a natural next step if clubs differ in more than their label balance. Applied studies also show that real cross-institution federations rarely enjoy complete, clean labels: Qiu et al. [48] trained vertically federated heart-sound classifiers when most positive labels were missing. Sports data share this problem, since many in-game events are coded inconsistently or not at all, and privacy protection has to work alongside such label noise. Our DP results show what formal protection costs at this scale. Budgets of $\varepsilon \approx 20$–60, which leave accuracy untouched, offer weak protection; budgets of $\varepsilon \approx 2$–8 cost up to about three points, and the cost falls if fewer communication rounds are used or more clients take part.

Several limitations should be noted. First, the main features are box-score statistics recorded during the game being classified. The models therefore classify outcomes from game performance, which is useful for identifying key factors and for post-game analysis, but they are not forecasts; the pre-game variant (Section 3.7) shows the accuracy one can expect when only past games are used. Second, the federated clients were simulated from a single public dataset. In a real cross-silo deployment, clubs would hold richer and more heterogeneous data, such as tracking and physiological records, collected with different systems, schemas, missing-data patterns and access policies, and client dropout, asynchronous updates and secure aggregation [43] would need to be handled. The practical claims of this study should be read in that light. Third, the DP accountant ignores amplification by subsampling [41], so the reported ε values are upper bounds, and the privacy guarantee assumes an honest-but-curious server without secure aggregation.

Fourth, only linear models were federated; extending the framework to nonlinear and deep models, personalised FL and vertical FL across data vendors is a promising direction for future work.

5. Conclusions

This study compared centralised and federated machine learning for classifying NBA game outcomes from box-score differentials. The analysis covered 49,899 complete games from the 1979–80 to 2024–25 seasons. Pearson and point-biserial correlation analyses described how team performance indicators relate to scoring and winning, and LR, SVM and KNN were tuned on the 2023–24 season and tested on the 2024–25 season.

Field-goal percentage, field goals made and defensive rebounds showed the strongest associations with winning; these are associations within a game, not causal effects. All classifiers were far above the 0.549 accuracy of always predicting a home win. SVM (0.9081) and LR (0.9088 unweighted, 0.9038 class-weighted) could not be separated statistically, and KNN (0.8871) was consistently weaker. Field goals made and field-goal percentage were excluded from the inputs because they allow the final score to be reconstructed, and including them raises accuracy to 0.99–1.00.

Federated LR matched an identically configured centralised LR (paired difference 0.00 points on the test season and −0.07 points across five seasons) and was about one point better than franchise-only models. Each round exchanged only 15 parameters per client, and training was robust to partial participation. Under severe label skew, FedProx and an intercept recalibration on validation data recovered most of the loss. Client-level differential privacy gave a controllable trade-off: with five local epochs and ten rounds, accuracy stayed above 0.90 at $\varepsilon \approx 4$ for the training updates, although privatising the preprocessing statistics as well was costly at that budget with only 30 clients.

Overall, the findings show that collaborative training without pooling raw data is feasible and practically lossless for this task. They do not show that box-score models forecast games: the pre-game variant reached 0.63. Future research will test the framework on data that clubs genuinely hold privately, such as tracking and physiological records, investigate personalised and adaptive aggregation for heterogeneous clients, use secure aggregation, and extend the study to other leagues and multimodal data.

Author Contributions: Conceptualisation, B.F. and M.R.H.; methodology, B.F., M.R.H. and N.H.S.A.; software, B.F., M.R.H., H.X. and N.I.S.; validation, M.R.H., N.H.S.A. and B.F.; formal analysis, H.X. and B.F.; investigation, M.R.H.; resources, H.X. and B.F.; data curation, M.R.H.; writing—original draft preparation, B.F., M.R.H. and N.H.S.A.; writing—review and editing, N.I.S. and N.H.S.A.; visualisation, M.R.H.; supervision, B.F. and M.R.H.; project administration, B.F. and N.H.S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The NBA game data analysed in this study were obtained from the public Kaggle dataset "NBA Dataset: Box Scores and Stats (1947–Today)" by E. Moore, available at https://www.kaggle.com/datasets/eoinamoore/historical-nba-data-and-player-box-scores (TeamStatistics.csv) accessed on 16 October 2026. The processed data and the code used for the centralised and federated experiments are available from the corresponding author upon reasonable request.

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

Abbreviations

The following abbreviations are used in this manuscript: AUC, area under the receiver operating characteristic curve; CI, confidence interval; CBA, Chinese Basketball Association; DP, differential privacy; FedAvg, federated averaging; FedProx, federated proximal optimisation; FL, federated learning; IID, independent and identically distributed; KNN, k-nearest neighbours; L-BFGS, limited-memory Broyden–Fletcher–Goldfarb–Shanno; LR, logistic regression; LSTM, long short-term memory; NBA, National Basketball Association; NFL, National Football League; RBF, radial basis function; RDP, Rényi differential privacy; SGD, stochastic gradient descent; SVM, support vector machine.

References

1. Bai, Z.; Bai, X. Sports Big Data: Management, Analysis, Applications, and Challenges. *Complexity* **2021**, *2021*, 6676297. https://doi.org/10.1155/2021/6676297.
2. D'Urso, P.; De Giovanni, L.; Swartz, T. Editorial: Big Data and Data Science in Sport. *Ann. Oper. Res.* **2023**, *325*, e1496. https://doi.org/10.1007/s10479-023-05344-z.
3. Ghosh, I.; Ramasamy Ramamurthy, S.; Chakma, A.; Roy, N. Sports Analytics Review: Artificial Intelligence Applications, Emerging Technologies, and Algorithmic Perspective. *WIREs Data Min. Knowl. Discov.* **2023**, *13*, e1496. https://doi.org/10.1002/widm.1496.
4. Cust, E.E.; Sweeting, A.J.; Ball, K.; Robertson, S. Machine and Deep Learning for Sport-Specific Movement Recognition: A Systematic Review of Model Development and Performance. *J. Sports Sci.* **2019**, *37*, 568–600. https://doi.org/10.1080/02640414.2018.1521769.
5. Naismith, J. *Basketball: Its Origin and Development*; Bison Books ed.; University of Nebraska Press: Lincoln, NE, USA, 1996.
6. Markovits, A.S. Basketball as a Democratizing Force in the United States: The Prominence of Jews and African-Americans in What Has Come to Be Known as "The Liberal's Game". In *Demokratie: Modus und Telos. Beiträge für Anton Pelinka*; Markovits, A.S., Rosenberger, S.K., Eds.; Böhlau: Vienna, Austria, 2001; pp. 259–282.
7. Stein, M.; Janetzko, H.; Seebacher, D.; Jäger, A.; Nagel, M.; Hölsch, J.; Kosub, S.; Schreck, T.; Keim, D.A.; Grossniklaus, M. How to Make Sense of Team Sport Data: From Acquisition to Data Modeling and Research Aspects. *Data* **2017**, *2*, 2. https://doi.org/10.3390/data2010002.
8. Dutt-Mazumder, A.; Button, C.; Robins, A.; Bartlett, R. Neural Network Modelling and Dynamical System Theory: Are They Relevant to Study the Governing Dynamics of Association Football Players? *Sports Med.* **2011**, *41*, 1003–1017. https://doi.org/10.2165/11593950-000000000-00000.
9. Beldon, Z. Basketball. In *Routledge Handbook of Global Sport*; Naurright, J., Zipp, S., Eds.; Routledge: London, UK, 2020; pp. 36–46. https://doi.org/10.4324/9781315714264.
10. Wyckoff, E.B. *The Man Who Invented the Game of Basketball: The Genius of James Naismith*; Enslow Publishing: Berkeley Heights, NJ, USA, 2013.
11. Nourayi, M.M. A Historical Perspective of Professional Basketball and Rules Changes in North America. *J. Sports Games* **2020**, *2*, 1–9.
12. Nelson, M.R. *The National Basketball League: A History, 1935–1949*; McFarland: Jefferson, NC, USA, 2009.
13. Surdam, D.G. *The Rise of the National Basketball Association*; University of Illinois Press: Urbana, IL, USA, 2012.
14. Staffo, D.F. The Development of Professional Basketball in the United States, with an Emphasis on the History of the NBA to Its 50th Anniversary Season in 1996–1997. *Phys. Educ.* **1998**, *55*, 9.
15. Jozsa, F.P., Jr. *The National Basketball Association: Business, Organization and Strategy*; World Scientific: Singapore, 2010. https://doi.org/10.1142/7822.
16. Bunker, R.; Susnjak, T. The Application of Machine Learning Techniques for Predicting Match Results in Team Sport: A Review. *J. Artif. Intell. Res.* **2022**, *73*, 1285–1322. https://doi.org/10.1613/jair.1.13509.
17. Gu, W.; Foster, K.; Shang, J.; Wei, L. A Game-Predicting Expert System Using Big Data and Machine Learning. *Expert Syst. Appl.* **2019**, *130*, 293–305. https://doi.org/10.1016/j.eswa.2019.04.025.
18. Yoo, C.; Ramirez, L.; Liuzzi, J. Big Data Analysis Using Modern Statistical and Machine Learning Methods in Medicine. *Int. Neurourol. J.* **2014**, *18*, 50–57. https://doi.org/10.5213/inj.2014.18.2.50.

19. Rahman, A. Statistics-Based Data Preprocessing Methods and Machine Learning Algorithms for Big Data Analysis. *Int. J. Artif. Intell.* **2019**, *17*, 44–65.
20. Ratner, B. *Statistical and Machine-Learning Data Mining: Techniques for Better Predictive Modeling and Analysis of Big Data*, 3rd ed.; Chapman and Hall/CRC: Boca Raton, FL, USA, 2017.
21. Buhamra, N.; Groll, A.; Brunner, S. Modeling and Prediction of Tennis Matches at Grand Slam Tournaments. *J. Sports Anal.* **2024**, *10*, 17–33. https://doi.org/10.3233/JSA-240670.
22. He, G.; Choi, H.S. Stacked Ensemble Model for NBA Game Outcome Prediction Analysis. *Sci. Rep.* **2025**, *15*, 29983. https://doi.org/10.1038/s41598-025-13657-1.
23. Zhong, Y. Enhancing Game Outcome Prediction in the Chinese Basketball League through a Machine Learning Framework Based on Performance Data. *Sci. Rep.* **2025**, *15*, 23788. https://doi.org/10.1038/s41598-025-08882-7.
24. Alves, J.M.; Barbosa, R.S. Machine Learning for Basketball Game Outcomes: NBA and WNBA Leagues. *Computation* **2025**, *13*, 230. https://doi.org/10.3390/computation13100230.
25. Gifford, M.; Bayrak, T. A Predictive Analytics Model for Forecasting Outcomes in the National Football League Games Using Decision Tree and Logistic Regression. *Decis. Anal. J.* **2023**, *8*, 100296. https://doi.org/10.1016/j.dajour.2023.100296.
26. Mahmood, Z.; Daud, A.; Abbasi, R.A. Using Machine Learning Techniques for Rising Star Prediction in Basketball. *Knowl.-Based Syst.* **2021**, *211*, 106506. https://doi.org/10.1016/j.knosys.2020.106506.
27. Deng, H. Basketball Free Throw Hit Probability Prediction System Based on Deep Learning Object Detection and Pose Estimation. *Syst. Soft Comput.* **2025**, *7*, 200334. https://doi.org/10.1016/j.sasc.2025.200334.
28. Wang, Y.; Liu, W.; Liu, X. Explainable AI Techniques with Application to NBA Gameplay Prediction. *Neurocomputing* **2022**, *483*, 59–71. https://doi.org/10.1016/j.neucom.2022.01.098.
29. Song, K.; Gao, Y.; Shi, J. Making Real-Time Predictions for NBA Basketball Games by Combining the Historical Data and Bookmaker's Betting Line. *Physica A* **2020**, *547*, 124411. https://doi.org/10.1016/j.physa.2020.124411.
30. Cai, W.; Yu, D.; Wu, Z.; Du, X.; Zhou, T. A Hybrid Ensemble Learning Framework for Basketball Outcomes Prediction. *Physica A* **2019**, *528*, 121461. https://doi.org/10.1016/j.physa.2019.121461.
31. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Agüera y Arcas, B. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA, 20–22 April 2017; PMLR: 2017; Volume 54, pp. 1273–1282.
32. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. https://doi.org/10.1561/2200000083.
33. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.* **2019**, *10*, 12. https://doi.org/10.1145/3298981.
34. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated Optimization in Heterogeneous Networks. In Proceedings of Machine Learning and Systems (MLSys), Austin, TX, USA, 2–4 March 2020; Volume 2, pp. 429–450.
35. Zhu, H.; Xu, J.; Liu, S.; Jin, Y. Federated Learning on Non-IID Data: A Survey. *Neurocomputing* **2021**, *465*, 371–390. https://doi.org/10.1016/j.neucom.2021.07.098.
36. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
37. Platt, J.C. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In *Advances in Large Margin Classifiers*; Smola, A.J., Bartlett, P., Schölkopf, B., Schuurmans, D., Eds.; MIT Press: Cambridge, MA, USA, 1999; pp. 61–74.
38. Hsu, T.-M.H.; Qi, H.; Brown, M. Measuring the Effects of Non-Identical Data Distribution for Federated Visual Classification. *arXiv* **2019**, arXiv:1909.06335. https://doi.org/10.48550/arXiv.1909.06335.
39. McMahan, H.B.; Ramage, D.; Talwar, K.; Zhang, L. Learning Differentially Private Recurrent Language Models. In Proceedings of the 6th International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
40. Mironov, I. Rényi Differential Privacy. In Proceedings of the 2017 IEEE 30th Computer Security Foundations Symposium (CSF), Santa Barbara, CA, USA, 21–25 August 2017; pp. 263–275. https://doi.org/10.1109/CSF.2017.11.
41. Mironov, I.; Talwar, K.; Zhang, L. Rényi Differential Privacy of the Sampled Gaussian Mechanism. *arXiv* **2019**, arXiv:1908.10530. https://doi.org/10.48550/arXiv.1908.10530.
42. Dwork, C.; Roth, A. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.* **2014**, *9*, 211–407. https://doi.org/10.1561/0400000042.

43. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.B.; Patel, S.; Ramage, D.; Segal, A.; Seth, K. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS), Dallas, TX, USA, 30 October–3 November 2017; pp. 1175–1191. https://doi.org/10.1145/3133956.3133982.
44. Dietterich, T.G. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Comput.* **1998**, *10*, 1895–1923. https://doi.org/10.1162/089976698300017197.
45. Jamieson, J.P. The Home Field Advantage in Athletics: A Meta-Analysis. *J. Appl. Soc. Psychol.* **2010**, *40*, 1819–1848.
46. Kubatko, J.; Oliver, D.; Pelton, K.; Rosenbaum, D.T. A Starting Point for Analyzing Basketball Statistics. *J. Quant. Anal. Sports* **2007**, *3*, 1. https://doi.org/10.2202/1559-0410.1070.
47. Wang, H.; Zhang, X.; Guo, H.; Ren, X.; Wang, S.; Fan, F.; Ma, J.; Zeng, D. Federated Metadata-Constrained iRadonMAP Framework with Mutual Learning for All-in-One Computed Tomography Imaging. *Cyborg Bionic Syst.* **2025**, *6*, 0376. https://doi.org/10.34133/cbsystems.0376.
48. Qiu, W.; Quan, C.; Yu, Y.; Kara, E.; Qian, K.; Hu, B.; Schuller, B.W.; Yamamoto, Y. Federated Abnormal Heart Sound Detection with Weak to No Labels. *Cyborg Bionic Syst.* **2024**, *5*, 0152. https://doi.org/10.34133/cbsystems.0152.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.