Journal of Big Data

RESEARCH Open Access

Data-driven prediction of soccer outcomes using enhanced machine and deep learning techniques

Ebenezer Fiifi Emire Atta Mills[1,4*†], Zihui Deng[2,4†], Zhuoqing Zhong[2,4†] and Jinger Li[3,4]

†Ebenezer Fiifi Emire Atta Mills, Zihui Deng and Zhuoqing Zhong equally to this work and share first authorship.

*Correspondence: eattamil@kean.edu

[1] School of Mathematical Sciences, Wenzhou-Kean University, Wenzhou 325060, Zhejiang, China
[2] Department of Computer Sciences, Wenzhou-Kean University, Wenzhou 325060, Zhejiang, China
[3] College of Business and Public Management, Wenzhou-Kean University, Wenzhou 325060, Zhejiang, China
[4] Academy of Interdisciplinary Research for Sustainability (AIRs), Wenzhou-Kean University, Wenzhou 325060, Zhejiang, China

Abstract

This paper introduces a novel framework for soccer game prediction using advanced machine learning and deep learning techniques, initially focusing on the Dutch Eredivisie League and later expanding to include the Scottish Premiership and the Belgian Jupiler Pro League. The methodology includes data preprocessing, feature engineering, model training, and testing. Various models are evaluated, including enhanced versions of Logistic Regression, XGBoost, Random Forest, SVM, Naive Bayes, Feedforward Neural Network, and Vanilla Recurrent Neural Network. Unlike existing studies that focus on end-of-game features, this research incorporates real-time features like half-time results and goals for in-game decision-making. Advanced data normalization and sampling methods, such as SVM-SMOTE and Near-Miss, are applied to improve model performance. Models are assessed using accuracy, recall, precision, F1-score, and Area under the ROC Curve. Results indicate that the Feedforward Neural Network excels in predicting game results, while Logistic Regression is best for predicting under and over 2.5 goals. The integration of Random Forest and XGBoost in a voting model consistently achieves the highest accuracy across both prediction tasks. The combined use of data from the three leagues further validates the models' robustness and generalizability. This study demonstrates the potential of machine and deep learning to enhance soccer game predictions through advanced techniques and comprehensive data analysis, making significant contributions to sports analytics.

Keywords: Deep learning, Feature engineering, Machine learning, Soccer prediction, Sports analytics

Introduction

Sports tournaments are embraced by individuals today and are followed with great enthusiasm. Soccer is one of these sporting events [1]. Sports managers employ artificial intelligence (AI) techniques, such as machine learning, for player selection and performance evaluation [2]. Moreover, artificial intelligence techniques predict intricate future events, such as stock market outcomes, game results, tournament and league predictions [3], bet odds, and team performances [4].

Soccer is a basic ball game where two teams compete to score by getting the ball into the opposing team's net, resulting in a win, draw, or defeat. Each match consists of two

45-min halves, plus extra time determined by the referee, and is governed by rules penalizing fouls. Over 85% of games result in either a draw or a team winning by one or two goals [5]. Tactics refer to a strategic plan aimed at achieving a specific goal [6], with the ultimate aim in sports being victory [7]. A coach's tactical choices are influenced by factors such as the player's physical condition, teamwork, technical skills, and the tactics of the opposing coach, as well as the game location. Winning is not determined by a single factor but by multiple contributing aspects [8].

In sports science, various studies on team performances have emerged recently. Gomez et al. [9] conducted a study on the performances of top-tier soccer teams at various periods of a game. Marcelino et al. [10] examined the collective motion patterns displayed by soccer players. Tan [11] attempts to develop a sports predictive analytics system based on AI and big data, examining the pivotal role these technologies play in the sports industry. Wunderlich and Memmert[4] extensively examined large datasets from soccer games and created a social network by analyzing player positions and passing patterns. Novillo et al. [12] introduces a novel methodology utilizing spatial multilayer networks to analyze the playing patterns of Real Madrid, FC Barcelona, and Getafe FC.

Charest and Sleep [13] review the role and effectiveness of predictive analytics in sports, focusing on the impact of sleep on athletic performance and recovery. Maglo et al. [14] highlights advances in deep learning, such as player re-identification, instance segmentation, and transformer-based architectures for semi-supervised learning, along with improvements in player pose estimation in dynamic sports environments. Beal et al. [15] survey AI and machine learning applications in sports, covering match outcome prediction, injury management, and other areas beyond performance metrics. Pappalardo et al. [16] propose using Multivariate Time Series (MTS) of player workload history to train a deep learning Convolutional Neural Network (CNN) to predict injury likelihood based on workload history.

Today, obtaining results and statistics about past games online is much easier than in many other domains. For this reason, predicting the outcomes of sports games has inspired many recent studies [17]. Given the extensive popularity of sports, there is also a growing interest in developing methodologies and strategies to predict game outcomes by analyzing various factors [18]. The choice of placing a wager on the game arises when a predicted outcome for a particular game is known. Although this is a crucial question, this study does not focus on it. Data scientists have utilized machine learning (ML) algorithms to predict soccer game results.[1] Geurkink et al. [21] employed Extreme Gradient Boosting (XGBoost) with 13 features to predict winning or losing soccer games.

Various ML models assist teams and managers in making optimal decisions to outperform opponents. Bunker and Susnjak [22] review various studies utilizing ML techniques to predict match outcomes in sports like soccer, rugby, and American football. ML algorithms used are Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), Decision Trees, and ensemble models. Key predictive factors encompass player statistics, team performance metrics, and external elements such

[1] Readers interested in a literature review of machine learning techniques in soccer and other sports outcome prediction are encouraged to refer to [19, 20].

as venue and weather conditions. For example, soccer outcomes have been predicted through analyses of large datasets using logistic regression and random forests. At the same time, rugby predictions have relied on linear regression models to evaluate team strength and home advantage. The importance of feature selection, particularly game-specific and time-dependent variables, is highlighted as essential for enhancing predictive accuracy. The paper underscores the complexity of sports data and notes that ML methods yield varying success based on the sport and dataset. Despite the challenges in developing universal predictive models, the field continues to advance, with future research encouraging interdisciplinary collaboration between sports performance analysis and machine learning.

Rahman [23] highlights that despite the advancements in machine learning techniques, the accuracy of predicting soccer league game results remains subpar due to the complexity of sports events. The primary difficulty encountered by researchers while analyzing soccer game datasets mostly rests in the process of feature engineering [24, 25]. Multiple endeavors have been made to identify meaningful associations among key characteristics extracted from soccer game data to construct a reliable predictive model [19].

This study addresses the identified gaps in predictive accuracy and feature engineering within sports analytics by employing enhanced machine learning and deep learning techniques, systematically improving model robustness and generalizability across the Dutch Eredivisie, Belgian Jupiler League, and Scottish Premiership. While predictive models often face challenges regarding the generalizability of their findings across different domains, this study addresses these concerns by incorporating data from multiple soccer leagues. Further, to enhance the robustness and applicability of the results, a combined dataset that merges these leagues is also analyzed, thereby broadening the scope and potential application of the findings in sports analytics.

Several machine learning models were evaluated, including enhanced versions of logistic regression, XGBoost, Random Forest, SVM, and Naive Bayes. Additionally, deep learning techniques, specifically Feedforward Neural Networks (FNN) and Vanilla Recurrent Neural Networks (RNN), were incorporated to address sports data's complexities and temporal dependencies.

The contribution of this research lies in several key customizations and enhancements. First, a robust feature engineering process was developed that incorporates real-time game features, such as half-time results and goals, departing from the conventional focus on end-of-game features. Additionally, 28 new features were introduced, designed to capture intricate aspects of team performance, player statistics, and tactical variations, enriching the model's input data and enhancing predictive accuracy.

To optimize model performance, hyperparameter tuning strategies were employed using grid search and Bayesian optimization techniques. This systematic approach ensured optimal model configurations, enhancing predictive performance and robustness. In addressing class imbalance challenges, data augmentation methods, including the Support Vector Machine Synthetic Minority Oversampling Technique (SVM-SMOTE), Near-Miss, and Random-OverSampling, were applied. These techniques improved the models' ability to learn from minority classes, resulting in more balanced and accurate predictions.

Recognizing the strengths and limitations of individual models, a voting model was developed that combines the predictions of finely tuned ML and DL models. This ensemble approach consistently achieved the highest accuracy across various prediction tasks, demonstrating the efficacy of integrating multiple model outputs. Enhancements to traditional DL models included advanced regularization techniques, dropout layers, batch normalization, and data augmentation strategies, enabling the models to capture complex, non-linear relationships within the data and improve generalization capabilities.

This study significantly advances the field of sports analytics by using enhanced machine learning and deep learning techniques. Through innovative feature engineering, advanced hyperparameter tuning, and advanced data augmentation methods, the study provides more accurate and robust predictions for soccer game outcomes.

The remainder of this paper is structured as follows. "Methodology" section discusses research methodology. "Data analysis" section analyzes the data. "Experimental result evaluation" section presents the experimental results, and "Conclusion" section discusses the findings. The last section presents the conclusion.

Methodology

The proposed methodology is presented in this section, and materials and methods are analyzed.

Proposed framework

As depicted in Fig. 1, the framework of this paper begins by acquiring data from the Dutch Eredivisie League, the Belgian Jupiler League, and the Scottish Premiership League. The initial data preprocessing involves dropping specified columns, encoding target variables for distinct classification tasks, and transforming categorical game results and binary over/under goal totals into numerical formats appropriate for model training. Numerous new features were developed from the original data during the feature engineering phase, such as Team State and Attack Strength. These features enrich the model's input data and enhance its predictive accuracy. The significance of all features was assessed using a Random Forest classifier, which identified the most influential factors within the dataset.

Various normalization techniques were employed in the model training phase, including min–max, max–abs, standardization (Std.), and robust scaling. These methods

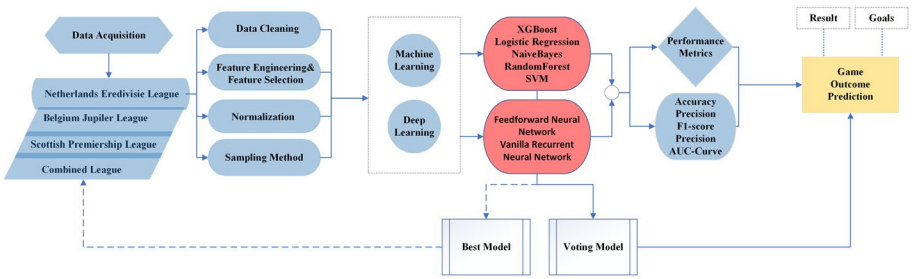

Fig. 1 Flow diagram of the proposed framework

adjust the scale of feature data, thus enhancing the model's stability and performance. Additionally, optional sampling methods like random under-sampling, near-miss, and support vector machine—synthetic minority over-sampling technique (SVM-SMOTE) were applied to balance the class distribution. These methods significantly improve the model's ability to effectively recognize minority classes. Each method can be selectively applied depending on the specific characteristics and needs of the dataset, allowing for flexible adaptation to various modeling scenarios.

Seven different models were initially tested with these features, all of which achieved acceptable results. Consequently, parameter tuning was carried out on all seven models to achieve the highest accuracy and identify the optimal predictive model. Finally, a soft voting model was utilized, which combines various trained models. Different combinations of models were experimented with to further optimize performance and achieve higher accuracy.

Data acquisition

Recent technological advancements have significantly broadened the scope of data collection in soccer, allowing for a comprehensive array of statistical metrics to be gathered for every individual game. Consequently, these data are readily available and utilized by various stakeholders, betting companies, and enthusiasts.

This study's dataset comprises data from the Dutch Eredivisie League, covering the 2017–2018 season to May 19th, 2024, and includes 1411 games. The results of these games are as follows: 644 Home Team Wins, 442 Draws, and 325 Away Team Wins. Additionally, the analysis classifies games according to the total goals scored, with 852 games registering fewer than 2.5 goals (categorized as 'Under') and 559 games recording more than 2.5 goals (categorized as 'Over').

The dataset utilized in this research is obtained from [26], which offers extensive data on various global soccer leagues from 1993–94 to the current period. This dataset encompasses historical game results, league rankings during each game, and bookmakers' closing odds. Including these odds is crucial as they reflect the bookmakers' specialized knowledge in predicting the outcomes of games and various other aspects of the games. These odds are typically available before the games commence. This study initially focuses on the Dutch Eredivisie League. Detailed information about the structure and rules of the league is available on the official Eredivisie website https://eredivisie.eu/home/.

Data cleaning

Throughout the dataset preparation process, several important data characteristics were identified. To maintain consistency in the study, data like the number of team offsides were excluded as they could not be standardized across all seasons. The main goal was to use only those data features that have been available since 2017, ensuring the integrity and comparability of the datasets.

Given the inherently high unpredictability of soccer game outcomes, incorporating a wide range of features is crucial. By including more features, a broader spectrum of game details can be captured, enhancing understanding and prediction capabilities. This

comprehensive approach allows models to account for various factors influencing the game, ultimately leading to more accurate and insightful predictions.

Focusing on complete and consistent data ensures that the machine and deep learning models are built on a stable and reliable foundation. This method emphasizes the clarity and integrity of the datasets, paving the way for powerful sports analytics applications that can better grasp the dynamics of soccer games.

Feature engineering

This paper's approach was refined by engineering new features from existing datasets to enhance the models for predicting soccer game outcomes. This feature engineering process allows for a deeper team performance analysis by creating additional variables.

In constructing statistical features within the analysis, two parameters, game History Window, and Goal Margin Diff, are employed to refine and direct the calculations. These parameters are essential for tailoring the statistical evaluation to specific research needs and game scenarios, thus enhancing the robustness and relevance of the derived metrics.

The game History Window parameter determines the number of recent games for calculating various statistical metrics. By defining the range of historical data included in the analysis, this parameter ensures that the generated statistics capture the team's current form and potential. It is particularly critical for analyses focused on patterns of consecutive wins, draws, or other performance trends, as it adjusts the temporal scope of the data under consideration. Similar approaches, such as using a history window in basketball analytics, emphasize the importance of selecting an optimal historical window to enhance predictive accuracy, as demonstrated in [27].

The Goal Margin Differential focuses on games where the goal difference meets or exceeds a specified threshold. This selective approach is pivotal for distinguishing between different types of victories-identifying games where a team won and did so with a significant margin. This parameter is crucial for assessing teams' offensive capabilities and defensive strength, providing a more detailed insight into their performance quality.

New features were introduced to enrich the dataset, such as Away Team State (A state for the Away Team, calculated using their recent performance metrics such as the number of wins in a recent window, goals scored and conceded, and their losses' goal difference), Home Team State, Home Attack Strength (A metric representing the attacking strength of the home team, derived from the home team's goals scored and the away team's goals against), Away Attack Strength, Away Win Rate (The percentage of away games won by the away team, calculated over a specified period and context), Home Win Rate, Home Loss Rate (The rate at which the home team loses games at home, assessed over a certain historical period), Away Loss Rate, Away Draw Rate (the proportion of away games that end in a draw for the away team, calculated across a specific period), and Home Draw Rate.

These engineered features provide important insights into team dynamics and are instrumental in improving the accuracy of the predictive models for soccer games.

Feature selection

The Random Forest model was exclusively used for feature selection to identify and select the most relevant features contributing to decision-making. Random Forest, an

ensemble of decision trees, inherently evaluates feature importance during the model training. Each tree in the forest considers a random subset of features to split on at each node, and the overall importance of each feature is determined by how much it contributes to reducing impurity across all trees. This method is particularly effective for reducing noise and avoiding overfitting by discarding irrelevant or less important features. Consequently, this approach helps enhance the model's accuracy and generalization capability by focusing only on the most significant predictors in the dataset.

Table 1 illustrates the comparison of feature importance between models predicting game results (Result) and models predicting whether the total goals in a game will be under or over 2.5 (U/O 2.5). The data shown in the table are derived from the Random

Table 1 Feature importance comparison between result and U/O 2.5 Models

Feature (result)	Importance (result)	Feature (U/O 2.5)	Importance (U/O 2.5)
Half-time result	0.0922	Home team half-time goals	0.1211
Home win bet odds	0.0786	Away team half-time goals	0.0991
Away win bet odds	0.0757	Away win bet odds	0.0657
Home team half-time goals	0.0707	Draw bet odds	0.0641
Away team half-time goals	0.0637	Home win bet odds	0.0596
Draw bet odds	0.0604	Half-time result	0.0556
Away wins in the window	0.0416	*Away wins in the window	0.0410
Home wins in the window	0.0406	*Home wins in the window	0.0406
Away team state	0.0314	*Away team state	0.0313
Home team state	0.0279	*Home team state	0.0307
Total away loss rate	0.0269	*Total home win rate	0.0285
Total away win rate	0.0265	*Total away win rate	0.0249
Total home win rate	0.0263	*Total away loss rate	0.0241
Total home loss rate	0.0258	*Total away draw rate	0.0238
Home attack strength	0.0240	*Total home draw rate	0.0231
Total away draw rate	0.0236	*Total home loss rate	0.0231
Total home draw rate	0.0234	*Away attack strength	0.0217
Away attack strength	0.0231	*Home attack strength	0.0211
Home goal differential	0.0212	*Away goal differential	0.0200
Away goals against	0.0203	*Away goals forward	0.0196
Away goal differential	0.0200	*Home goal differential	0.0192
Away goals forward	0.0199	*Away goals against	0.0189
Home goals forward	0.0185	*Home goals forward	0.0187
Home goals against	0.0184	*Home goals against	0.0158
Away loss	0.0118	*Away win	0.0103
Home wins margin goal	0.0113	*Home win	0.0102
Away win	0.0111	*Home draw	0.0101
Home draw	0.0108	*Away loss	0.0100
Home win	0.0108	*Home wins margin goal	0.0098
Away draw	0.0099	*Away draw	0.0087
Away losses margin goal	0.0095	*Away losses margin goal	0.0080
Home loss	0.0090	*Home loss	0.0080
Away wins margin goal	0.0076	*Away wins margin goal	0.0072
Home losses margin goal	0.0075	*Home losses margin goal	0.0063

* Indicates features created through feature engineering

Forest model analysis, highlighting how various features weigh differently depending on the target. This paper maintains all the features since their importance scores are non-zero, and incorporating a wide range of features is crucial given the inherently high unpredictability of soccer game outcomes.

Models

In the initial stage of the research, five distinct machine-learning models, two deep-learning models, and a voting model were evaluated to predict outcomes in soccer games within the Dutch Eredivisie League. The analysis was extended to the Scottish and Belgian Leagues to address the limitation of generalizability of the findings. The models implemented in this study are presented below.

Extreme gradient boosting

This model operates on gradient boosting with decision trees as the base learners. Gradient boosting is a technique where new models are successively added to correct the errors made by existing models [28]. Extreme Gradient Boosting (XGBoost) enhances this approach using a more sophisticated regularization framework, thus systematically addressing model overfitting while boosting accuracy. The information gained from splitting in the XGBoost algorithm can be mathematically described as follows:

$$\text{Gain} = \frac{1}{2}\left[\left(\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \lambda}\right) + \left(\frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \lambda}\right) - \left(\frac{\left(\sum_{i\in I} g_i\right)^2}{\sum_{i\in I} h_i + \lambda}\right)\right] - \gamma, \tag{1}$$

where I_L and I_R denote the instances on the left and right sub-trees, respectively; g and h represent gradients and Hessians, λ is the regularization parameter, and γ is the complexity control on tree growth. This formula balances between finding the best splits to maximize gain while controlling the growth of the tree complexity to ensure robustness and prevent overfitting.

Logistic regression

This model uses the logistic function to estimate the probability that an outcome variable equals a certain value. It is particularly used for binary classification tasks like predicting win vs. loss. The logistic regression function is modeled as follows:

$$\text{logit}\,(p) = \ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k. \tag{2}$$

Here, p is the probability of the dependent variable equaling a case (e.g., a win), $X_1, \ldots, X_k$ are the predictors, and $\beta_0, \beta_1, \ldots, \beta_k$ are the parameters to be estimated. This model estimates the log odds of the outcome as a linear combination of the predictors. The probabilities associated with each class are then used to classify data points by comparing them to a threshold value, commonly 0.5, to make predictions about the outcome.

Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' Theorem, assuming independence among features. It is particularly effective for large datasets due to its simplicity and efficiency [29]. The posterior probability for a class C_k given a feature vector x is calculated using the formula:

$$P(C_k|x) = \frac{P(C_k)\prod_{i=1}^{n} P(x_i|C_k)}{P(x)}, \tag{3}$$

where $P(C_k)$ is the prior probability of class C_k, $P(x_i \mid C_k)$ is the likelihood of feature x_i given class C_k, and $P(x)$ is the evidence, a scaling factor ensuring that the probabilities sum to one, calculated as $\sum_k P(C_k)\prod_{i=1}^{n} P(x_i \mid C_k)$. This model simplifies computation by assuming the features are conditionally independent given the class label, making it particularly suitable for high-dimensional data.

Random Forest

This model is an ensemble of decision trees, typically constructed with a randomized subset of data features to improve robustness over a single decision tree [30]. Random Forests address the overfitting problem inherent in individual trees by averaging or 'voting' across multiple tree predictions, effectively mitigating variance and bias. The predictive function for a Random Forest can be mathematically described as:

$$m_{M,n}(x; \Theta_1, \ldots, \Theta_M, D_n) = \frac{1}{M}\sum_{j=1}^{M} m_n\big(x; \Theta_j, D_n\big), \tag{4}$$

where $m_n(x; \Theta_j, D_n)$ represents the prediction from the j-th tree, M is the total number of trees, x is the input feature vector, Θ_j are the random parameters used during the training of each tree, and D_n is the training data. This formulation ensures that each tree contributes independently to the final outcome, reducing error by averaging the results.

Support vector machine

SVM works by identifying the optimal hyperplane that separates data classes in a high-dimensional space. A hyperplane is a decision boundary that maximizes the distance (margin) between the classes' nearest points (support vectors). This distance is maximized to increase the separation clarity and, consequently, the classification accuracy. Mathematically, SVM aims to solve the optimization problem of minimizing $\|w\|$ to maximize the margin M, under the constraints that ensure all data points are correctly classified with the maximum margin:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\|w\|^2 \\ &\text{subject to} \quad y_i(w \cdot x_i + b) \geq 1. \end{aligned} \tag{5}$$

This optimization problem ensures that the SVM model provides a robust classification with high generalization ability.

Feedforward Neural Network

A Feedforward Neural Network (FNN) consists of layers of interconnected nodes or neurons [31]. Each neuron in these networks performs computations by calculating a weighted sum of its inputs and then applying an activation function. The process is described by:

$$v_i = \sum_j w_{ij} \cdot x_j + b_i, \tag{6}$$

where v_i represents the weighted sum at the i-th neuron, w_{ij} denotes the weight from input j to neuron i, x_j are the input values, and b_i is the bias term for neuron i. The output y_i of each neuron is then obtained by applying a nonlinear activation function f:

$$y_i = f(v_i).$$

This architecture enables the FNN to learn complex patterns from the data, which applies to classification, regression, and pattern recognition tasks.

Vanilla Recurrent Neural Network

A Vanilla Recurrent Neural Network (RNN) is a neural network that processes data sequences by maintaining a hidden state that captures information from previous time steps. The architecture involves recurrent connections between hidden layers, which allows the network to learn temporal dependencies.

The following equations describe the operation of a Vanilla RNN:

$$h_t = \sigma(U_h x_t + W_h h_{t-1}), \tag{7}$$

$$y_t = O h_t, \tag{8}$$

where $x_t \in \mathbb{R}^n$ is the input at time step t, $h_t \in \mathbb{R}^k$ is the hidden state at time step t, $y_t \in \mathbb{R}^p$ is the output at time step t, U_h, W_h, and O are the network's weight matrices, and $\sigma(x) = \frac{1}{1+e^{-x}}$ is the logistic activation function applied component-wise.

The hidden state h_t serves as a memory of previous inputs, enabling the RNN to handle tasks like language modeling, speech recognition, and time series prediction. However, Vanilla RNNs often struggle with long-term dependencies due to issues like vanishing and exploding gradients.

Voting model

A voting technique that uses a majority voting approach to integrate predictions from multiple models. This method enhances prediction accuracy by aggregating class outputs from various pre-trained models and selecting the class that appears most frequently among them. The voting method aims to reduce variance and biases present in single models. The mathematical representation for hard voting is given by:

$$\text{class label} = \text{mode}\{c_1, c_2, \ldots, c_m\}, \tag{9}$$

where c_i represents the class prediction by the i-th classifier out of m total classifiers. This method selects the class with the highest frequency of predictions.

This comprehensive suite of models provides a robust framework for predicting outcomes using various aspects of machine learning techniques, each contributing uniquely to the final decision-making process.

Data augmentation

To address the imbalance in the datasets, five advanced data augmentation techniques were employed: support vector machine synthetic minority oversampling technique (SVM-SMOTE), synthetic minority oversampling technique with nearest neighbors (SMOTE-NN), near-miss (NM), and random-oversampling (ROS). Each method is crucial in creating balanced and accurate predictions across the dataset.

The SVM-SMOTE method uses support vector machines to identify and generate synthetic examples along the decision boundary. By carefully creating samples that accurately represent the minority class's decision space, data integrity is maintained while strengthening the model's ability to distinguish between classes.

Similarly, SMOTE-NN generates synthetic samples near the original based on their nearest neighbors. Expanding minority class clusters without replicating existing samples prevents overfitting and encourages a more generalized model that effectively handles varying class distributions.

The NM technique reduces the majority class by selecting samples that are closest to the minority class instances, helping to achieve a more refined decision boundary. It removes unnecessary majority class data while creating a more balanced training dataset that accurately represents both classes.

The random-oversampling technique increases minority class representation by randomly replicating its existing samples. By augmenting the dataset with more minority samples, the model becomes more adept at detecting patterns and making accurate predictions involving the minority class.

Applying these techniques to the leagues' dataset is crucial for developing predictive models that minimize inherent bias toward the majority class, thereby improving fairness and accuracy. Integrating these augmentation methods ensures that all outcomes are equally represented and prevents predictions from being skewed by imbalanced results, such as home wins.

Model evaluation

The performance of the classification model was assessed using a set of evaluation metrics, namely accuracy, recall, precision, and the F1-score. Complementing these, the area under the receiver operating characteristic (ROC) curve, commonly abbreviated as AUC, served as an integrative measure of the model's discriminatory capacity.

The accuracy metric quantifies a predictive model's aggregate performance across all categorical outcomes. In the context of this investigation, encompassing Home Win, Draw, Away Win, it is delineated as the quotient of the aggregate of correct predictions over the entirety of prognostications made:

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}} \quad (10)$$

It must be underscored that the accuracy metric might yield an overestimated view of model performance in instances of class imbalance, especially for the class with a preponderance of instances.

Recall, also known as sensitivity, is computed as the ratio of accurately discerned positive instances to the sum of actual positive instances, reflecting the model's capability to detect all positive instances:

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (11)$$

Precision, as a metric, is articulated as the ratio of correctly identified positive instances to the total number of instances adjudged as positive:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (12)$$

Elevated precision intimates the model's dependability in asserting the positivity of instances.

The F1-score is a harmonized mean of precision and recall, offering a singular metric for the balanced assessment of a model's precision and recall:

$$\text{F1-score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (13)$$

The rate of false positives is a fraction denoting the proportion of negative instances incorrectly classified as positive, utilized predominantly in the construction of the Receiver Operating Characteristic (ROC) curve:

$$\text{False Positive Rate} = \frac{\text{False Positive}}{\text{False Positive} + \text{True Negative}} \quad (14)$$

In evaluating the efficacy of a classification model, the area under the receiver operating characteristic (ROC) curve-denoted as AUC-serves as a critical metric. The ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier as its discrimination threshold is varied. A model with an AUC approaching unity indicates its superior capability to discriminate between classes with high accuracy.

The selection of an appropriate performance metric is inherently contingent upon the nature of the problem. Within the scope of this research, a comprehensive array of metrics is presented to facilitate comparative analysis, thereby contributing to a holistic assessment of model performance. Nevertheless, the ultimate metric for evaluation is predicated on accuracy. This predilection is attributed to the ubiquity of accuracy as a benchmark in classification model evaluation. It is attributed to its straightforwardness and interpretability, which are particularly advantageous in scenarios characterized by a finite set of target classes, as in the present study.

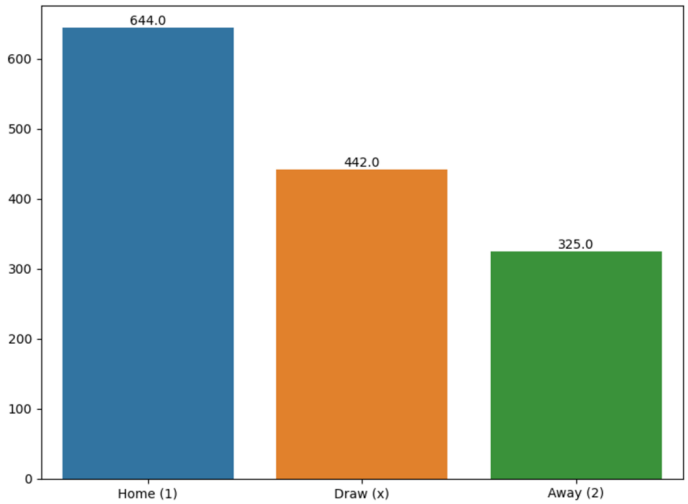

Fig. 2 Result distribution analysis

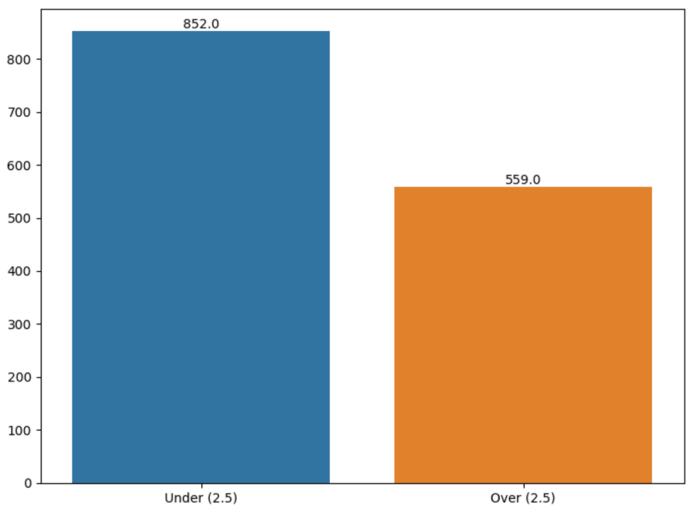

Fig. 3 U/O distribution analysis

With the methodology set, this paper applies these techniques to the collected data. The next section examines the data analysis, presenting how these methodologies manifest in practical application and the insights they yield.

Data analysis

In the following sections, the data from the Dutch Eredivisie league is analyzed. The analysis covers the distribution of home, draw, and away results for different teams and the bet odds distributions for these outcomes.

First, the overall distribution of game outcomes was examined for a comprehensive analysis. This is crucial for understanding the broader patterns in the league and enhancing predictive models. From Fig. 2, the total counts of home wins (644), draws (442), and away wins (325) are observed. This initial overview sets the stage for a deeper dive into the specific factors influencing these results.

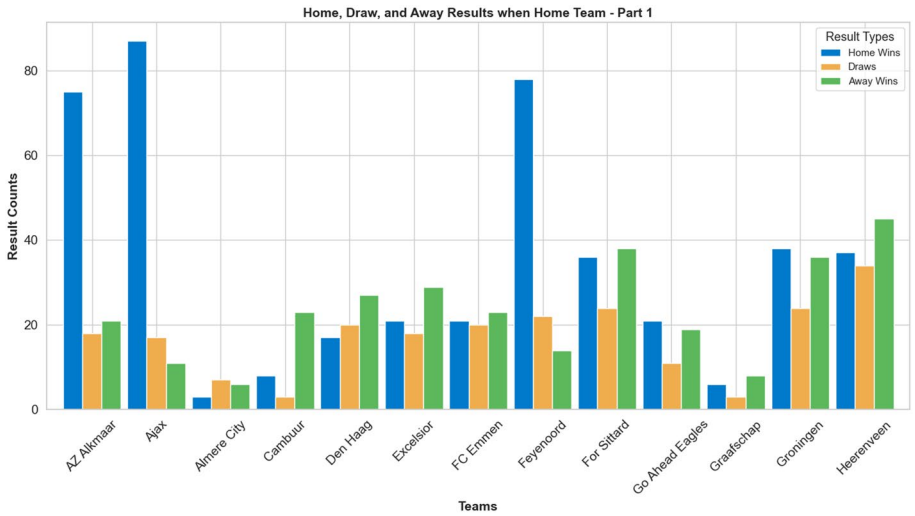

Fig. 4 Home Team results part 1

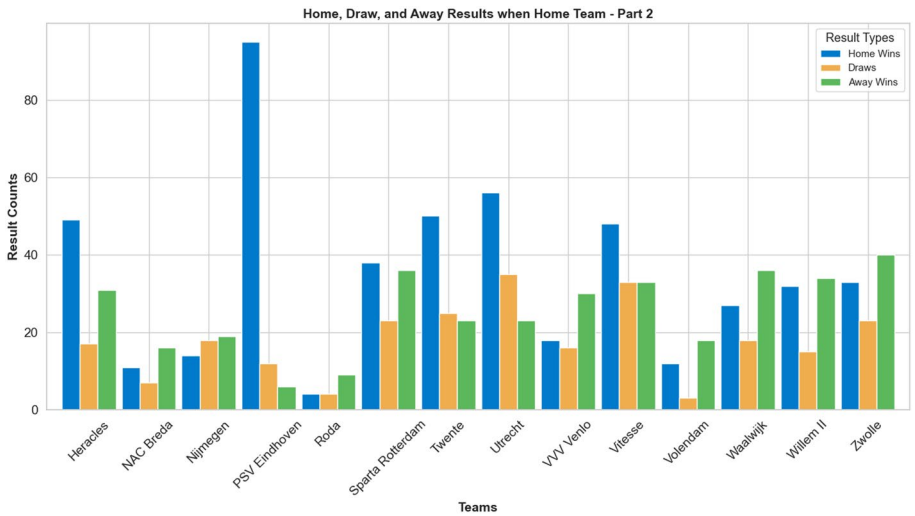

Fig. 5 Home team results part 2

Additionally, the target distribution of Under and Over (2.5) outcomes was analyzed, as shown in Fig. 3. The data shows that games resulting in Under 2.5 goals are more frequent (852 times) than Over 2.5 goals (559 times). This distribution helps understand the scoring patterns in the Eredivisie, which can further inform predictive models.

Following this, teams' performance as home and away teams was examined. Figures 4, 5, 6, and 7 illustrate these results. Figures 4 and 5 show the home performance; PSV Eindhoven, Ajax, and Feyenoord stand out with the highest number of home wins. These teams exhibit a strong home-field advantage, consistently winning many games at home. Figures 6 and 7 show the away performance; they are more evenly distributed across different teams. While no single team dominates regarding away

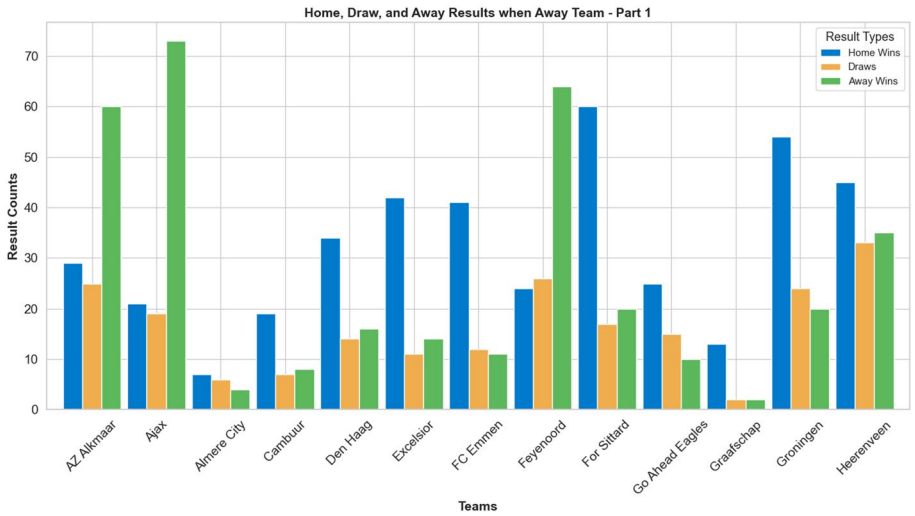

Fig. 6 Away team results part 1

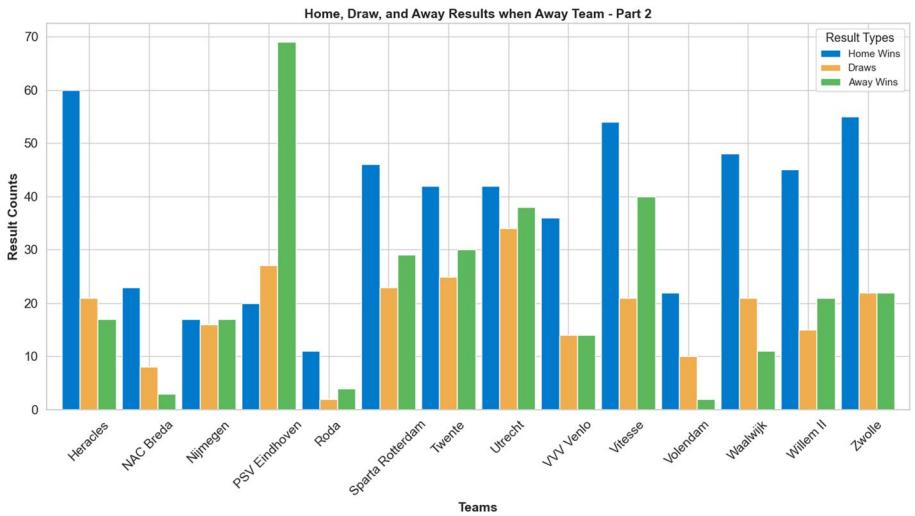

Fig. 7 Away team results part 2

wins, teams like Heracles and Zwolle perform better than others in away games. This indicates their relative strength and adaptability when playing away from home.

To further the understanding, the distribution of bet odds of home win, draw, and away win was analyzed, as shown in Figs. 8, 9, and 10, with the horizontal axis representing the bet odds. The bet odds of winning at home are heavily skewed towards lower values, suggesting that the home team is likelier to win. This is consistent with the strong home-field advantage observed in the performance analysis. The bet odds of a draw start at values around 2.3 and are concentrated around the higher values, indicating that draws are less likely by the betting market. Finally, the distribution of the bet odds of winning away from home is more spread out, indicating a lower

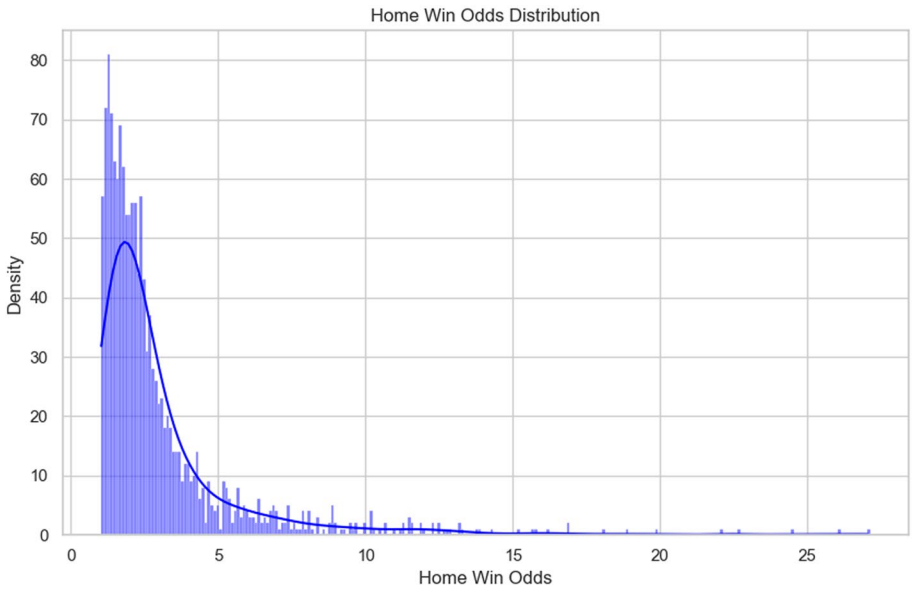

Fig. 8 Home win odds distribution

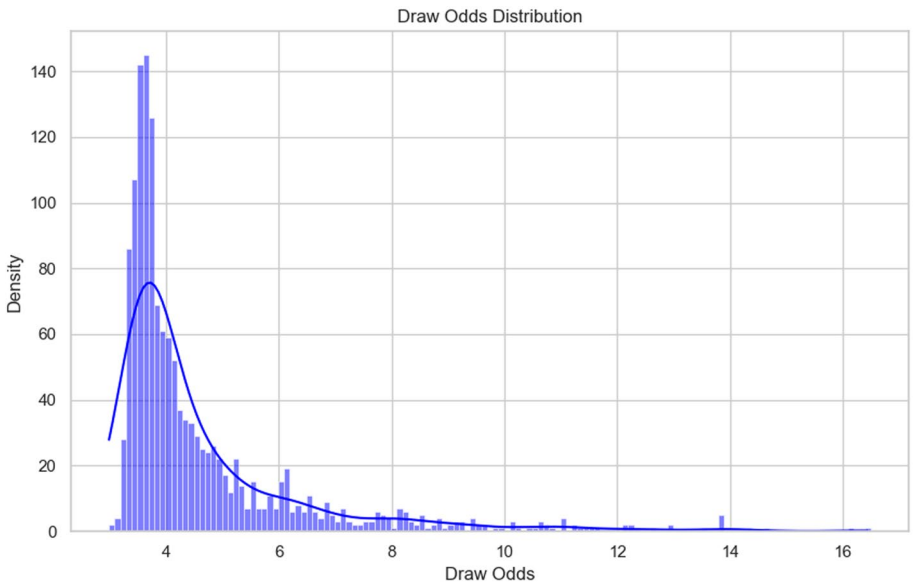

Fig. 9 Draw odds distribution

probability and higher variability of winning away from home. This is consistent with the finding that away games are more challenging for teams.

Key insights are derived by combining the results from Figs. 1 through 10. The data strongly supports the presence of a home-field advantage, with teams like PSV Eindhoven, Ajax, and Feyenoord benefiting significantly when playing at home, as indicated by their high win counts and favorable bet odds. Additionally, the analysis highlights the competitiveness of teams like Heracles and Zwolle in away games, suggesting their strategic strengths and resilience in less favorable conditions.

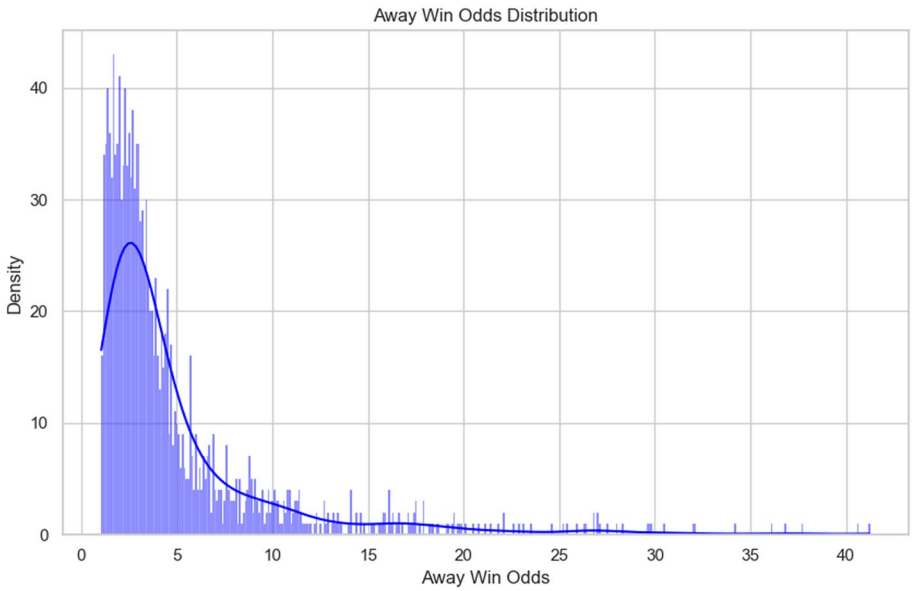

Fig. 10 Away win odds distribution

The bet odds distributions reflect the betting market's expectations. Lower bet odds for home wins correlate with a higher probability of home victories, while the higher and more variable bet odds for away wins and draws indicate greater uncertainty. Despite higher bet odds for draws, the number of draws (439) is significant. This indicates that while the betting market sees draws as less likely, they are a common outcome in the Eredivisie, adding complexity to predictive models. To address this, data augmentation techniques were implemented to adjust and enhance the data, aiming to improve the accuracy of draw predictions.

In conclusion, the comprehensive analysis of the Eredivisie league data reinforces the presence of a strong home-field advantage, the competitive nature of away games, and the intricate challenge of predicting draws. By employing enhanced data modeling techniques like data augmentation, a better understanding of game outcomes can be achieved, ultimately improving the ability to predict game results with greater accuracy.

Having explored the data, the focus now shifts to evaluating the experimental results derived from applying the methodologies. This section assesses the effectiveness of the different models in predicting soccer game outcomes.

Experimental result evaluation

At the initial stage of model development, several models are configured with basic parameters and subjected to preliminary training to gauge their performance. Once these models have been trained on the dataset, the Area Under the Curve (AUC) metric is employed to assess each model's overall performance. The AUC metric is particularly valuable as it provides a comprehensive view of how well each model distinguishes between the different classes across various threshold settings. This preliminary evaluation helps identify the most promising models, which can then be further fine-tuned and evaluated in subsequent stages of the development process.

Table 2 Initial performance comparison of models

Model	Accuracy	F1-Score	Precision	Recall
Logistic Regression	0.67	0.61	0.61	0.62
XGBoost	0.67	0.61	0.61	0.62
Random Forest	0.69	0.57	0.62	0.61
SVM	0.66	0.63	0.64	0.63
Naive Bayes	0.64	0.53	0.52	0.57
Vanilla RNN	0.65	0.58	0.57	0.59
FNN	0.66	0.55	0.55	0.58

Table 3 Classification metrics for initial models

Model	Class	Precision	Recall	F1-score
Logistic Regression	Home (0)	0.80	0.79	0.79
	Draw (1)	0.36	0.31	0.33
	Away (2)	0.67	0.76	0.71
XGBoost	Home (0)	0.74	0.84	0.79
	Draw (1)	0.31	0.16	0.21
	Away (2)	0.61	0.71	0.66
Random Forest	Home (0)	0.78	0.86	0.82
	Draw (1)	0.46	0.12	0.19
	Away (2)	0.60	0.85	0.71
SVM	Home (0)	0.81	0.74	0.77
	Draw (1)	0.38	0.47	0.42
	Away (2)	0.73	0.69	0.71
Naive Bayes	Home (0)	0.73	0.83	0.78
	Draw (1)	0.24	0.08	0.12
	Away (2)	0.60	0.79	0.69
Vanilla RNN	Home (0)	0.68	0.82	0.74
	Draw (1)	0.27	0.22	0.24
	Away (2)	0.77	0.72	0.75
FNN	Home (0)	0.75	0.84	0.79
	Draw (1)	0.31	0.10	0.15
	Away (2)	0.60	0.81	0.69

Initial experiments

The performances of the five machine learning models and two deep learning models are evaluated using the Dutch Eredivisie League dataset, with each model trained for two distinct prediction tasks. The first task aims to predict game results, while the second focuses on predicting whether a game would have over or under 2.5 goals.

The initial evaluation of predictive models for sports results is detailed in Tables 2 and 3. These tables provide insights into the accuracy, precision, recall, and F1 scores for various models, indicating their strengths and weaknesses.

The Random Forest model shows the highest overall accuracy (0.69) and performs well in predicting Home wins (F1-score of 0.82). However, it struggles significantly with Draw predictions (F1-score of 0.19). SVM and Logistic Regression models display similar overall accuracies (0.66 and 0.67, respectively) and balanced

Table 4 Initial performance comparison of models about U/O 2.5 goals

Model	Accuracy	F1-score	Precision	Recall
Logistic Regression	0.74	0.74	0.75	0.76
XGBoost	0.74	0.73	0.73	0.73
Random Forest	0.73	0.73	0.72	0.73
SVM	0.73	0.73	0.73	0.74
Naive Bayes	0.73	0.72	0.72	0.72
Vanilla RNN	0.74	0.81	0.75	0.88
FNN	0.73	0.70	0.71	0.70

Table 5 Classification metrics for initial models about U/O 2.5 goals

Model	Under precision	Over precision	Under recall	Over recall	Under F1-score	Over F1-score
Logistic Regression	0.67	0.78	0.68	0.78	0.68	0.78
XGBoost	0.67	0.78	0.68	0.78	0.68	0.78
Random Forest	0.66	0.79	0.69	0.76	0.68	0.77
SVM	0.65	0.81	0.74	0.73	0.69	0.77
Naive Bayes	0.66	0.78	0.68	0.76	0.67	0.77
Vanilla RNN	0.71	0.75	0.49	0.88	0.58	0.81
FNN	0.66	0.76	0.58	0.82	0.62	0.79

performances across Home and Away predictions, though SVM has a slightly better performance in predicting Draws (F1-score of 0.42).

XGBoost and Naive Bayes models have comparable accuracies (0.67 and 0.64, respectively), with XGBoost excelling in predicting Home and Away results but underperforming in Draw predictions (F1-score of 0.21). Naive Bayes also shows a notable deficiency in predicting draws (F1-score of 0.12).

Vanilla RNN and Feedforward Neural Network models achieve moderate overall accuracies (0.65 and 0.66, respectively). Vanilla RNN is slightly better at predicting Away results (F1-score of 0.75), whereas the Feedforward Neural Network performs overall balanced but struggles with Draw predictions (F1-score of 0.15).

This analysis highlights the need for model refinement, particularly in improving Draw predictions through parameter adjustments and possibly exploring more sophisticated modeling techniques.

Based on the classification metrics for U/O 2.5 models, several insights can be derived from Tables 4 and 5.

Logistic Regression demonstrates a well-balanced performance with high precision and recall for both "Under" and "Over" predictions, making it a reliable choice for general predictions. The Vanilla RNN model also shows strong performance, particularly with a high recall for "Over" predictions (0.88), although it has a lower recall for "Under" predictions (0.49).

Random Forest, XGBoost, and SVM models exhibit balanced performance across both "Under" and "Over" predictions, with metrics slightly lower than those of Logistic Regression and Vanilla RNN. These models provide robust general predictions, but their

performance may vary slightly depending on the specific scenario. Naive Bayes demonstrates moderate performance, with reasonably high precision and recall for "Under" predictions but slightly lower metrics for "Over" predictions. This indicates that while it is fairly reliable for "Under" predictions, it may need further refinement to improve its accuracy for "Over" outcomes. The Feedforward Neural Network shows a well-balanced performance with slightly lower metrics than Logistic Regression but still performs reliably for both "Under" and "Over" predictions.

Models like Logistic Regression and Vanilla RNN exhibit strong performance across both prediction categories, making them reliable choices for general predictions. Other models, such as Random Forest, XGBoost, and SVM, also provide solid performance but may require adjustments based on specific prediction needs. While moderately effective, Naive Bayes could benefit from additional tuning to enhance its predictive accuracy.

The AUC curves offer essential insights into the comparative efficacy of the machine learning and deep learning models used in this study. Figure 11 provides a visual representation of these comparisons, showcasing AUC curves for seven different models across two prediction tasks: game results and U/O 2.5 goals.

In predicting game results, the SVM, Logistic Regression, and Feedforward Neural Network models show the highest AUC scores (0.88), indicating their strong discriminative power in predicting game results. The RNN model also performs well with an AUC of 0.87. The Naive Bayes model demonstrates a slightly lower AUC (0.85), but it is still reasonably effective. XGBoost and Random Forest models follow closely with AUC scores of 0.84 and 0.84, respectively.

For the prediction of U/O 2.5 goals, the Logistic Regression model again leads with an AUC score of 0.83, followed closely by the RNN model with an AUC of 0.82 and the SVM model at 0.80. The Feedforward Neural Network also performs well with an AUC of 0.81. The XGBoost and Naive Bayes models have slightly lower AUC scores (0.76 each), while the Random Forest model shows an AUC of 0.75.

The AUC curves provide a visual comparison of model performance, helping to identify which models may be more suitable for further optimization and deployment in predicting soccer game results and goals scored. The SVM, Logistic Regression, and Feedforward Neural Network models stand out for their high performance in both tasks. It is important to note that these are preliminary inferences. More detailed model training will be conducted in the next steps.

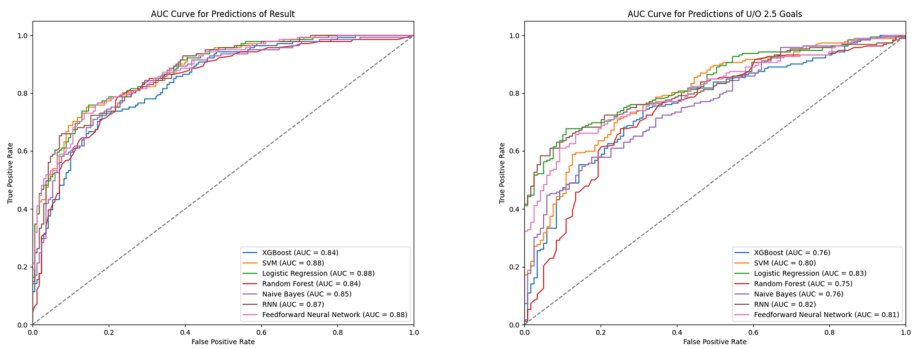

Fig. 11 AUC curves for the seven models about results and U/O 2.5 Goals

Further experimentation

After the initial evaluations, extensive parameter tuning was performed on each model to optimize their predictions. This phase involved adjusting various hyperparameters to enhance model accuracy and robustness. Additionally, a Voting Model was used to combine the predictions of these seven tuned models to achieve better overall performance. The results of these adjustments were systematically collected and analyzed, providing refined insight into the models' performances. Table 6 presents the specific performance of each model after parameter tuning and the combined results of the Voting Model.

The logistic regression model was the first model evaluated in the study. After applying hyperparameter tuning, the optimal configuration to maximize predictive accuracy was determined. This involved using the `liblinear` solver due to the specification of an l_1 penalty. Furthermore, the Standardization method was applied to enhance the model's performance. Additionally, to address the class imbalance, the Random-OverSampling sampler was employed. The highest predictive accuracy achieved through sevenfold cross-validation was 0.69.

The XGBoost model was the subsequent model evaluated in the analysis. The configuration of the model included setting the number of estimators to 500, a maximum depth of 10, an l_1 penalty of 0.9, an l_2 penalty of 0.8, a minimum child weight of 7, and a learning rate of 0.06. Subsequently, employing sevenfold cross-validation, a reported accuracy of 0.67 was achieved, signifying the model's effectiveness in predicting the target variable.

Following the XGBoost model, the analysis progressed to evaluate the Random Forest model. With 2000 estimators, the model was trained using the `RandomForestClassifier` from the sklearn library. Standardization was applied, and to mitigate class imbalance, random oversampling was employed. The Gini impurity measure was utilized for splitting, with 'sqrt' as the maximum features setting. The model's parameters included a minimum sample leaf of 9, a maximum depth of 46, and a minimum sample split of 12. Through this configuration, an accuracy of 0.72 was achieved through sevenfold cross-validation.

The Support Vector Machine (SVM) model was then evaluated. Instantiated using the `SVC` class from the sklearn library, the SVM model was configured with a regularization coefficient (C) of 0.7, a linear kernel, and a 'scale' gamma setting. Class weights were not adjusted, and SVM-SMOTE was employed to address class imbalance. Additionally,

Table 6 Tuned result performance comparison of eight models

Model	Normalizer	Sampler	Acc.	F1	Prec.	Rec.
Logistic Regression	Std.	ROS	0.69	0.66	0.65	0.66
XGBoost	Std.	None	0.67	0.59	0.60	0.61
Random Forest	Std.	ROS	0.72	0.63	0.69	0.64
SVM	Std.	SVM-SMOTE	0.66	0.62	0.62	0.63
Naive Bayes	Min–Max	NM	0.66	0.57	0.68	0.59
Vanilla RNN	Std.	None	0.71	0.58	0.64	0.61
FNN	Std.	None	0.73	0.68	0.73	0.67
Voting Model	None	None	0.83	0.81	0.86	0.79

probability calibration was enabled to enhance the model's reliability. Through sevenfold cross-validation, the SVM model achieved an accuracy of 0.66.

Next, the Naive Bayes model was evaluated. Utilizing the MultinomialNB class from the sklearn library, the model was configured with standard parameters, and the Near-Miss sampler was used to mitigate class imbalance. Specifically, the Multinomial algorithm was employed for Naive Bayes classification, effectively handling the frequency of occurrence data. Additionally, sevenfold cross-validation was conducted, yielding an accuracy of 0.66. The Near-Miss method, a sampling technique used to address the class imbalance in datasets, played a crucial role by selecting samples from the majority class closest to the minority class samples, creating a more balanced distribution. This comprehensive approach ensured a robust evaluation, addressing class imbalance while maintaining reliable performance.

Subsequently, Vanilla RNN was examined in detail. Utilizing TensorFlow and TensorFlow Addons, the model architecture was designed to comprise three layers of Vanilla RNNs. The first layer comprised 64 units, while the subsequent two contained 128 units each. The ReLU activation function was applied throughout the network, with l_2 weight regularization employed to mitigate overfitting. Batch normalization was integrated to enhance model stability, and a dropout rate of 0.5 was used for regularization. The training process was orchestrated with a batch size of 15 and lasted 1000 epochs. Early stopping was enforced after 50 epochs of no improvement to prevent overfitting. Additionally, the Yogi optimizer was chosen with a learning rate of 0.001, and the learning rate was adjusted after 20 epochs of no improvement. For data augmentation, an input noise of 0.15 was injected. After meticulous configuration and optimization, the model was primed for evaluation, yielding an accuracy rate of 0.71 through sevenfold cross-validation. Detailed evaluation results showed a precision of 0.64, a recall of 0.61, and an F1 score of 0.58, demonstrating stable performance across different categories.

Feedforward Neural Networks were then explored. Employing TensorFlow and TensorFlow Addons, the model architecture was designed to contain 4 connected hidden layers; the first is 64 units, and the remaining 3 are 128. The ReLU activation function was applied throughout the network, with l_2 weight regularization utilized to mitigate overfitting. A dropout rate of 0.5 was employed for regularization, and batch normalization was integrated to enhance model stability. The training process was orchestrated with a batch size of 15 and lasted for 400 epochs. Early stopping was enforced after 50 epochs of no improvement to prevent overfitting. Additionally, the Yogi optimizer was chosen with a learning rate of 0.002 and a learning rate patience of 20. An input noise of 0.15 was injected for data augmentation purposes. After meticulous configuration and optimization, the model was primed for evaluation, yielding an accuracy rate of 0.73 through sevenfold cross-validation.

The Voting Model, an ensemble learning technique, combines predictions from multiple individual models to make the final prediction. Initially, the input data undergoes preprocessing, after which each model's prediction probabilities are aggregated for voting. Specifically, predictions are obtained from each model for each input sample, and their prediction probabilities are averaged. Subsequently, the class with the highest average prediction probability is chosen as the final prediction. Throughout the research, various model combinations were explored, the most successful being the

fusion of XGBoost and Random Forest models. This combination achieved an accuracy of 0.83, indicating its robust performance and generalization capability in addressing the research problem.

A detailed analysis of the model performance data, as shown in Table 7, reveals that the Feedforward Neural Network is one of the effective models, aside from the Voting Model. This conclusion is drawn from its performance across different classes, particularly in the Home category, where it achieved a precision of 0.74 and an F1 score of 0.82. The model also performed well in the Away category, with a recall of 0.74, showcasing its ability to generalize across various results.

The Feedforward Neural Network's metrics, achieved without sampling techniques, illustrate its efficiency and adaptability to diverse data conditions. This makes it particularly suitable for environments where maintaining a balance between accuracy, computational complexity, and robustness is critical. Its performance indicates a robust capability to handle predictive tasks effectively across different scenarios, making it one of the reliable choices for complex predictive applications.

Aside from the Feedforward Neural Network, SVM and Random Forest also demonstrate strong performances, particularly in predicting Home outcomes. SVM achieves a precision of 0.74, a high recall of 0.92, and an F1 score of 0.82 for Home predictions, showcasing its accuracy in this category, while maintaining consistent results in the Away category with both precision and recall at 0.74. Random Forest performs similarly

Table 7 Detailed classification metrics for eight models for result task

Model	Class	Precision	Recall	F1-score
Logistic Regression	Home (0)	0.81	0.76	0.78
	Draw (1)	0.44	0.45	0.44
	Away (2)	0.72	0.77	0.74
XGBoost	Home (0)	0.77	0.85	0.81
	Draw (1)	0.39	0.22	0.29
	Away (2)	0.64	0.74	0.69
Random Forest	Home (0)	0.75	0.92	0.83
	Draw (1)	0.63	0.24	0.35
	Away (2)	0.68	0.76	0.72
SVM	Home (0)	0.74	0.92	0.82
	Draw (1)	0.71	0.35	0.47
	Away (2)	0.74	0.74	0.74
Naive Bayes	Home (0)	0.70	0.83	0.76
	Draw (1)	0.73	0.16	0.27
	Away (2)	0.60	0.79	0.68
Vanilla RNN	Home (0)	0.66	0.86	0.75
	Draw (1)	0.50	0.10	0.16
	Away (2)	0.77	0.87	0.82
FNN	Home (0)	0.74	0.92	0.82
	Draw (1)	0.71	0.35	0.47
	Away (2)	0.74	0.74	0.74
Voting Model	Home (0)	0.77	0.97	0.86
	Draw (1)	0.89	0.55	0.68
	Away (2)	0.93	0.86	0.89

well, with a precision of 0.75, recall of 0.92, and an F1 score of 0.83 in the Home category, along with reliable Away predictions (precision of 0.68 and recall of 0.76). Although both models face challenges in predicting Draw outcomes, these metrics indicate that SVM and Random Forest are effective models for Home and Away predictions, offering strong overall reliability. Thus, while FNN performs well, SVM and Random Forest both demonstrate comparable, if not superior, effectiveness, especially in critical categories like Home predictions.

From a macro perspective, as seen in Table 8, various models generally perform better in predicting Under/Over (U/O) 2.5 goals compared to the Result category. This enhanced performance is likely attributable to the binary nature of the U/O 2.5 problem, which inherently reduces complexity.

The Logistic Regression model demonstrates solid predictive performance, achieving an accuracy of 0.78. Its F1-score of 0.77 supports this further, indicating a well-balanced trade-off between precision and recall. Notably, the model is equally effective in predicting both classes, with precisions of 0.77 for under and 0.79 for over predictions and recalls of 0.66 for under and 0.87 for over predictions. These results underscore the model's robust ability to differentiate between the two classes, even when managing data imbalances. The standardization normalization method combined with SVM-SMOTE sampling ensures reliable predictions, leading to a consistently accurate model.

The XGBoost model, configured with standardization normalization and SMOTE-NN sampling, achieves a solid accuracy of 0.76. This result, supported by the balanced F1-score of 0.76, underscores the model's consistent performance across precision and recall. With precisions of 0.65 and 0.90 and recalls of 0.88 and 0.68 for under and over-predictions, respectively, the model effectively addresses class imbalances, ensuring both classes receive comparable predictive attention. The strategic parameter tuning results in a robust model that balances predictive power and computational efficiency, making it well-suited for handling diverse classification tasks.

The Random Forest model achieves consistent performance across various metrics, demonstrating an accuracy of 0.77 and a macro-average F1-score of 0.77. This balance is particularly evident in the class-specific results, where precision and recall for Class 1 are 0.84 and 0.77, respectively, slightly outperforming Class 0. Configured with 979 estimators and using the entropy criterion to maximize information gain at each split, the model lays a robust predictive foundation. Feature selection, managed by selecting

Table 8 Tuned U/O performance comparison of eight models

Model	Normalizer	Sampler	Accuracy	F1-score	Precision	Recall
Logistic Regression	Std.	SVM-SMOTE	0.78	0.77	0.78	0.76
XGBoost	Std.	SMOTE-NN	0.76	0.76	0.77	0.78
Random Forest	Std.	Random-Oversampling	0.77	0.77	0.77	0.77
SVM	Std.	SVM-SMOTE	0.77	0.76	0.76	0.77
Naive Bayes	Min–Max	Near-Miss	0.75	0.75	0.75	0.76
Vanilla RNN	Std.	None	0.76	0.75	0.74	0.75
FNN	Std.	None	0.75	0.74	0.75	0.74
Voting Model	None	None	0.84	0.84	0.85	0.87

the square root of the total number of features (`sqrt`), helps the model strike an optimal balance between variance reduction and generalization. Additional settings, including a minimum sample leaf of 21, a minimum sample split of 20, and a maximum depth of 40, are strategically chosen to prevent overfitting, ensuring the trees remain general rather than overly specific to the training data. Moreover, implementing balanced class weights addresses potential class imbalances, providing accurate and reliable differentiation between the two classes, thereby enhancing the model's performance in diverse conditions and its overall generalizability.

The Support Vector Machine model achieves strong predictive performance, reaching an accuracy of 0.77 and a macro-average F1-score of 0.76. This balance is evident across classes, with Class 1 achieving precision and recall of 0.85 and 0.74, respectively, slightly outperforming Class 0. Configured with a regularization coefficient (C) of 0.6, a sigmoid kernel, and the gamma parameter set to 'scale,' the model manages non-linearities effectively while ensuring balanced predictions across classes through the balanced class weight parameter. The combination of these well-calibrated settings, together with data standardization and SVM-SMOTE sampling, enables the SVM to deliver accurate and robust predictions by balancing precision and recall.

The Naive Bayes model, using a Multinomial algorithm and Min–Max method, demonstrates reliable performance with an accuracy of 0.75 and a macro-average F1-score of 0.75. Class 1 exhibits stronger precision and recall at 0.84 and 0.72, respectively, while Class 0 shows a precision of 0.66 and a recall of 0.80. The Multinomial algorithm, specifically designed for classification with discrete features, effectively handles the frequency of occurrence data, making it well-suited for text classification and other similar tasks. The application of Min–Max normalization ensures that feature scaling contributes to the model's performance consistency across different data ranges. Using the Near-Miss sampler to address class imbalance further enhances the model's robustness, providing consistent and accurate predictions across diverse data distributions.

The Vanilla RNN model demonstrates notable predictive accuracy, achieving an overall accuracy of 0.76 and an F1-score of 0.75, indicative of a balanced precision-recall ratio. It excels in the "Over" category with a precision of 0.83 and recall of 0.78, outperforming its metrics in the "Under" category. The model's ability to differentiate effectively between these classes highlights its adept handling of data imbalances. Its architecture includes three recurrent layers of 128 units each, supplemented by batch normalization and a dropout rate of 0.5 to prevent overfitting. Starting with a SimpleRNN layer, the network maintains uniform unit configuration and activation through two additional RNN layers, ending in a Dense layer with sigmoid activation for binary classification. Using the Yogi optimizer with a learning rate of 0.001 and an early stopping mechanism, activated after 50 epochs without improvement, optimizes training and ensures robust performance across evaluations.

The Feedforward Neural Network model demonstrates solid predictive performance, achieving an accuracy of 0.75. Its F1-score of 0.74 further supports this, indicating a balanced trade-off between precision and recall. Notably, the model is slightly more effective in predicting the "Over" class, with a precision of 0.81 and recall of 0.77, compared to the "Under" class. These results underscore the model's robust ability to differentiate between the two classes, even when managing data imbalances. The model's architecture

includes multiple dense layers with 64 and 128 units, batch normalization, and dropout for regularization. Specifically, the model starts with a Dense layer of 64 units and ReLU activation, followed by batch normalization. This is succeeded by two Dense layers of 128 units each, with ReLU activation, and a dropout layer with a rate of 0.5 to prevent overfitting. The final layer is a Dense layer with a single unit and sigmoid activation, suitable for binary classification. Using the Yogi optimizer, with a learning rate of 0.001, enhances the model's training process. Additionally, the training regime includes early stopping, which monitors validation accuracy and halts training when performance stops improving, set with a patience of 50 epochs. These strategies ensure reliable and consistent performance, leading to a consistently accurate model.

After experimenting with various combinations, it is noteworthy that the Voting Model combining the predictive strengths of XGBoost and Random Forest once again achieves the highest result, with an impressive accuracy of 0.84. This combination maintains consistent performance across all metrics, with precision, recall, and F1-score each at 0.84. Class 0 achieves a precision, recall, and F1-score of 0.71, while Class 1 achieves a precision, recall, and F1-score of 0.99, 0.76, and 0.86 respectively. This highlights the reliable synergy between these two models, confirming their ability to deliver robust predictions and generalize effectively across different data scenarios.

Upon reviewing the results in Table 9, the Voting Model emerges as the top performer for U/O 2.5 predictions, demonstrating the highest overall performance across various metrics. The Voting Model achieves an F1-Score of 0.83, along with precision of 0.71 for under predictions and 0.99 for over predictions, and recalls of 0.99 for under predictions and 0.76 for over predictions. These metrics show a strong predictive ability.

The Voting Model consistently achieved the highest accuracy across the two prediction tasks by combining Random Forest (RF) and XGBoost. This superior performance likely stems from the complementary strengths of these two algorithms. Random Forest, known for its robust ensemble learning approach, excels in handling diverse data types and complex interactions. XGBoost, an advanced gradient boosting implementation, is highly efficient and precise, capturing subtle data patterns effectively. By integrating RF's robustness and XGBoost's precision, the Voting Model benefits from the strengths, improving overall performance and making it a powerful tool for prediction tasks.

In a comprehensive study comparing seven models (excluding the Voting Model) for predicting soccer game results-Win, Draw, and Loss-the feedforward neural network

Table 9 Detailed classification metrics for eight models for U/O 2.5 Task

Model	Under precision	Over precision	Under recall	Over recall	Under F1-score	Over F1-swcore
Logistic Regression	0.77	0.79	0.66	0.87	0.71	0.83
XGBoost	0.65	0.90	0.88	0.68	0.75	0.77
Random Forest	0.69	0.84	0.78	0.77	0.73	0.80
SVM	0.68	0.85	0.81	0.74	0.74	0.79
Naive Bayes	0.66	0.84	0.80	0.72	0.72	0.78
Vanilla RNN	0.66	0.83	0.73	0.78	0.70	0.80
FNN	0.65	0.81	0.72	0.77	0.68	0.79
Voting Model	0.71	0.99	0.99	0.76	0.83	0.86

notably emerged as the top performer. Closely following was the Vanilla RNN, which secured its position as the second-best model, demonstrating strong robustness and reliability in capturing the intricacies of game results. The exceptional performance of the neural network and the Vanilla RNN highlights their superior ability to handle the complexity inherent in multi-class classification tasks.

For the U/O 2.5 goals prediction task, aimed at determining whether a game will end with under or over 2.5 goals, the Logistic Regression model excelled above the rest, achieving an impressive accuracy of 0.78. The Support Vector Machine (SVM) and Random Forest models were also highly effective, each scoring an accuracy of 0.77. The success of the Logistic Regression model underscores its adaptability and precision in binary classification tasks, making it especially suited for predictions involving discrete, binary outcomes like goal counts. Both SVM and Random Forest demonstrated that they, too, are capable contenders, providing dependable performance for this predictive challenge.

Additional experiments were conducted to substantiate these findings further and ensure consistency and reliability of the results. These will help validate the current conclusions and explore potential improvements in model training and feature selection. By enhancing the methodology and incorporating more diverse data, predictions can be refined, and the applicability of the models can be expanded to a wider range of soccer game scenarios.

Comparison with other leagues

After analyzing the Dutch Eredivisie league, further exploration of other leagues is necessary to validate the conclusions. The Belgian and Scottish leagues closely align with the Dutch league in culture, emphasizing technical prowess and fostering intense competition. Geographical proximity facilitates frequent exchanges and competitions among teams, while impressive performances in European competitions provide valuable reference points. Additionally, the competitiveness between teams in these leagues mirrors that of the Dutch league, reinforcing their suitability as comparable options. Moreover, there is no significant difference in their prominent features for prediction, further enhancing their suitability for inclusion in the comparative framework. These leagues offer striking parallels across various dimensions, enriching the comparative framework for accurate predictions.

The examination of the Belgian Jupiler League data, as detailed in Table 10, illustrates the efficacy of various predictive models. Among these, the Feedforward Neural

Table 10 Tuned result performance comparison of seven models on Belgian Jupiler

Model	Normalizer	Sampler	Accuracy	F1-score	Precision	Recall
Logistic Regression	Std.	None	0.64	0.47	0.42	0.54
XGBoost	Std.	None	0.62	0.54	0.56	0.55
Random Forest	Std.	Random-OverSampling	0.64	0.56	0.58	0.57
SVM	Std.	SVM-SMOTE	0.64	0.55	0.60	0.56
Naive Bayes	Std.	None	0.60	0.50	0.51	0.52
Vanilla RNN	Std.	None	0.62	0.53	0.55	0.54
FNN	Std.	None	0.66	0.54	0.70	0.57

Network stands out with the highest accuracy of 0.66 and an F1-score of 0.54, demonstrating its strong ability to effectively manage complex, non-linear data relationships. Its precision of 0.70 and recall of 0.57 further confirm its robust performance, especially in distinguishing between 'Home' and 'Away' outcomes.

In contrast, when enhanced with random oversampling, the Random Forest model also shows commendable performance with an accuracy of 0.64 and an F1-score of 0.56. This model is a reliable choice for handling prediction tasks, with its precision and recall rates of 0.58 and 0.57, respectively, indicating consistent performance across various game results.

The SVM model, using SVM-SMOTE as a sampling technique, similarly achieves an accuracy of 0.64, underscoring its effectiveness with an F1-score of 0.55. It exhibits a notable precision of 0.60, contributing to its strong recall rate of 0.56. This configuration illustrates SVM's capacity to balance between correct predictions and actual positives, making it a viable contender in this analysis.

Despite the strengths displayed by these models, predicting 'Draw' outcomes remains a universal challenge, as indicated by generally lower scores across all metrics for this category. While not leading in overall accuracy, Logistic Regression maintains a fairly balanced profile with modest precision and recall rates, reinforcing its versatility in handling different outcome classes.

In conclusion, the results from the Belgian Jupiler League reaffirm the superior predictive capabilities of the Feedforward Neural Network, which consistently excels across different metrics. The Random Forest and SVM models also perform robustly, highlighting their potential as reliable predictors in sports analytics. These insights underscore the importance of model selection and data handling strategies in optimizing prediction accuracy in soccer game results.

The tuned U/O performance comparison in Table 11 reflects some notable shifts from previous results, particularly in the context of the Belgian Jupiler League. The Feedforward Neural Network and Vanilla RNN models now stand out as the top performers for U/O predictions in this league, each achieving the highest accuracy of 0.79 and an F1-score of 0.78. These models' performances are enhanced by standardization without additional sampling techniques, demonstrating their robustness and capability to handle complex patterns within the data.

Logistic Regression, previously a top performer, continues to show strong results with an accuracy of 0.73 and an F1-score of 0.73, indicating that while the Feedforward

Table 11 Tuned U/O performance comparison of seven models on Belgian Jupiler

Model	Normalizer	Sampler	Accuracy	F1-score	Precision	Recall
Logistic Regression	Std.	None	0.73	0.73	0.76	0.75
XGBoost	Std.	None	0.72	0.72	0.72	0.72
Random Forest	Std.	None	0.73	0.71	0.73	0.71
SVM	Std.	None	0.70	0.69	0.69	0.70
Naive Bayes	Std.	None	0.70	0.70	0.70	0.70
Vanilla RNN	Std.	None	0.79	0.78	0.78	0.79
FNN	Std.	None	0.79	0.78	0.77	0.78

Neural Network and RNN models have shown superior performance in the Belgian Jupiler League, Logistic Regression remains a reliable model. The Random Forest model also performs well, achieving an accuracy of 0.73 and an F1-score of 0.71. XGBoost exhibits solid performance with an accuracy of 0.72 and an F1-score of 0.72, while the SVM and Naive Bayes models demonstrate reasonable performance with accuracies and F1-scores of 0.70.

In summary, the tuned U/O performance analysis highlights the Feedforward Neural Network and Vanilla RNN as the best models for U/O predictions in the Belgian Jupiler League, demonstrating significant predictive capabilities. Logistic Regression and Random Forest also perform well, but further observation is required to assess the consistency of Logistic Regression, which shows acceptable yet not top-tier performance. This does not completely overturn previous conclusions but suggests a need for a more detailed evaluation. Further research and experiments are crucial to explore and confirm these findings, ensuring the robustness and reliability of the models in different scenarios. This ongoing assessment will help refine understanding of each model's strengths and limitations.

The analysis of the Scottish Premiership, as represented in Table 12, highlights the performance of various models under different normalizers and sampling techniques. Notably, the Feedforward Neural Network model demonstrates strong predictive capabilities with an accuracy of 0.65 and an F1-score of 0.56. Interestingly, the Naive Bayes model also achieves the highest accuracy of 0.65, highlighting that different machine learning and deep learning models can exhibit varying predictive powers across different datasets. However, the Feedforward Neural Network consistently maintains high robustness.

The Vanilla RNN and Random Forest models are the second-best performers, each achieving an accuracy of 0.64. The RNN model notably secures an F1-score of 0.61, while the Random Forest games this accuracy with an F1-score of 0.57. This highlights their effectiveness in handling the prediction tasks within the league. Logistic Regression, with standardization and random oversampling, also shows solid performance but with slightly lower metrics, having an accuracy of 0.63 and an F1-score of 0.52. Similarly, the SVM model shows a comparable level of effectiveness with an accuracy of 0.61 and an F1-score of 0.57. In contrast, the XGBoost model demonstrates lower predictive ability in this dataset, with an accuracy of 0.59 and an F1-score of 0.53.

This comprehensive evaluation underscores the strengths of the Feedforward Neural Network and Naive Bayes models in handling the complexities of the Scottish

Table 12 Tuned result performance comparison of seven models on Scottish premiership

Model	Normalizer	Sampler	Accuracy	F1-Score	Precision	Recall
Logistic Regression	Std.	Random-Over-Sampling	0.63	0.52	0.57	0.54
XGBoost	Std.	None	0.59	0.53	0.54	0.53
Random Forest	Std.	None	0.64	0.57	0.58	0.58
SVM	Std.	None	0.61	0.57	0.57	0.57
Naive Bayes	Min–Max	Near-Miss	0.65	0.59	0.60	0.59
Vanilla RNN	Std.	None	0.64	0.61	0.61	0.61
FNN	Std.	None	0.65	0.56	0.60	0.58

Table 13 Tuned U/O performance comparison of seven models on Scottish premiership

Model	Normalizer	Sampler	Accuracy	F1-score	Precision	Recall
Logistic Regression	Std.	Random-OverSampling	0.76	0.75	0.79	0.76
XGBoost	Std.	Random-OverSampling	0.74	0.73	0.75	0.74
Random Forest	Std.	None	0.76	0.76	0.76	0.76
SVM	Min–Max	Near-Miss	0.73	0.73	0.73	0.73
Naive Bayes	Min–Max	None	0.71	0.71	0.71	0.71
Vanilla RNN	Std.	None	0.75	0.75	0.75	0.75
FNN	Std.	None	0.75	0.75	0.75	0.75

Table 14 Tuned result performance comparison of seven models on merged datasets

Model	Normalizer	Sampler	Accuracy	F1-Score	Precision	Recall
Logistic Regression	Std.	None	0.66	0.65	0.66	0.65
XGBoost	Std.	None	0.65	0.64	0.64	0.64
Random Forest	Std.	None	0.66	0.60	0.61	0.61
SVM	Std.	None	0.67	0.62	0.64	0.62
Naive Bayes	Min–Max	Near-Miss	0.63	0.59	0.59	0.60
Vanilla RNN	Std.	None	0.65	0.61	0.61	0.61
FNN	Std.	None	0.67	0.61	0.62	0.62

Premiership's predictions while emphasizing the strong performances of the Vanilla RNN and Random Forest as reliable models for such predictive tasks. This analysis highlights the need for careful model selection based on the dataset's unique attributes to optimize prediction results.

The tuned U/O performance comparison in Table 13 reflects similar trends to the previous results. Logistic Regression stands out as one of the best models for U/O predictions, achieving a high accuracy of 0.76 and an F1-score of 0.75, demonstrating robustness and simplicity with standardization and random oversampling. The Random Forest model also performs strongly, reaching an accuracy of 0.76 and an F1-score of 0.76, highlighting its predictive power for binary classification tasks.

In summary, previous analyses indicated that Logistic Regression was the best model for predicting U/O 2.5 goals. The latest analysis of the Scotland dataset reaffirms this. However, the performance of the Random Forest model suggests that it is equally capable of predicting U/O outcomes. The Neural Network model, with an accuracy and F1-score of 0.75, demonstrates its comprehensive predictive capabilities.

Comparison with combined leagues

After independently training models on the Dutch, Belgian, and Scottish leagues, further exploration of the findings is necessary. Given the similar competitive levels and playing styles across these three leagues, combining their data may provide a more robust dataset for analysis. Therefore, the datasets from the three leagues were combined, spanning from the 2017 season to the present, resulting in a total of 5329 games. Models are trained on this combined dataset to validate the conclusions

Table 15 Tuned U/O performance comparison of seven models on merged dataset

Model	Normalizer	Sampler	Accuracy	F1-Score	Precision	Recall
Logistic Regression	Std.	None	0.77	0.77	0.77	0.78
XGBoost	Std.	None	0.76	0.75	0.75	0.76
Random Forest	Std.	None	0.76	0.74	0.76	0.74
SVM	Std.	None	0.74	0.71	0.76	0.71
Naive Bayes	Std.	None	0.72	0.71	0.71	0.72
Vanilla RNN	Std.	None	0.73	0.73	0.73	0.73
FNN	Std.	None	0.73	0.71	0.71	0.72

further. Tables 14 and 15 respectively show the training results of the models for predicting the Result and U/O 2.5 goals.

Based on the analysis of the combined dataset, the performance of seven different models for predicting game results was evaluated. As shown in Table 14, the Feedforward Neural Network and SVM Model achieved the highest accuracy of 0.67. The FNN Model had an F1-score of 0.61, while the SVM Model had an F1-score of 0.62. These results align with the previous conclusion that the Feedforward Neural Network is one of the best predictive models for game results, and they also highlight the effectiveness of the SVM Model.

The Logistic Model achieved an accuracy of 0.66 and an F1-score of 0.65, indicating its robustness and simplicity in prediction tasks. The XGBoost Model showed an accuracy of 0.65 and an F1-score of 0.64, performing reliably but slightly below the Feedforward Neural Network and SVM Models. The Random Forest Model demonstrated consistent performance with an accuracy of 0.66 and an F1-score of 0.60, highlighting its capability in handling classification tasks. The Vanilla RNN, with an accuracy of 0.65 and an F1-score of 0.61, shows commendable performance in sequence data prediction but slightly lags behind the top models. Using Min–Max normalization and Near-Miss sampling, the Bayes Model had the lowest performance with an accuracy of 0.63 and an F1-score of 0.59. This suggests that while the Bayes Model performed well in the Scottish Premiership, it again showed poor performance on the combined dataset, proving that it may be less suitable for soccer prediction tasks.

This consistency across individual and combined datasets reinforces the conclusion that the Feedforward Neural Network model is highly effective for predicting game results. The Logistic Regression and SVM Models also show reliable performance, making them suitable alternatives depending on specific requirements and contexts. This comprehensive evaluation underscores the importance of model selection and fine-tuning to achieve optimal predictive performance in different scenarios.

Based on the analysis of the combined dataset, the performance of seven different models for predicting U/O 2.5 goals was evaluated. As shown in Table 15, the Logistic Regression Model achieved the highest accuracy of 0.77 and an F1-score of 0.77. These results confirm the previous conclusion that the Logistic Regression Model is the best predictive model for U/O 2.5 goals.

In earlier analyses of individual league datasets, the Logistic Regression Model consistently demonstrated superior performance for predicting U/O 2.5 goals. The

combined dataset reaffirms this, with the Logistic Regression Model maintaining strong performance. The Random Forest and XGBoost Models also performed well, each achieving an accuracy of 0.76 with F1-scores of 0.74 and 0.75, respectively, but they did not surpass the Logistic Regression Model. The SVM Model showed an accuracy of 0.74 and an F1-score of 0.71. The Feedforward Neural Network and Vanilla RNN each recorded an accuracy of 0.73 and an F1-score of 0.73, demonstrating their effectiveness in modeling this prediction task. However, they perform slightly below the best model. With the lowest performance, the Bayes Model had an accuracy of 0.72 and an F1-score of 0.71, suggesting that it may be less suitable for soccer prediction tasks in this context.

This consistency of the Logistic Regression Model across individual and combined datasets supports the initial conclusion. However, the strong performance of the Random Forest and XGBoost models in the combined dataset suggests they are also strong contenders for predicting U/O 2.5 goals. This analysis highlights the effectiveness of multiple models and underscores the importance of selecting the appropriate model based on the specific dataset and prediction task.

The evaluation of experimental results provides a comprehensive understanding of each model's capabilities. Building on these insights, the discussion section explores the implications of these findings and their practical applications in sports analytics.

Discussion

The analysis of predicting soccer game outcomes using machine learning and deep learning models revealed several key insights. Table 16 summarizes the performance of various predictive models across these leagues.

Across the individual datasets of the Dutch Eredivisie, Belgian Jupiler League, and Scottish Premiership, the Feedforward Neural Network consistently demonstrated superior predictive performance for game results. This aligns with the model's ability to handle complex and non-linear patterns in data, making it particularly effective in scenarios characterized by high variability and unpredictability, such as soccer games.

Table 16 Model training results across different leagues

League	**Result prediction**	
	Best model	**Runner-up model**
Dutch Eredivisie	FNN	RF
Belgian Jupiler	FNN	RF, SVM
Scottish Premiership	FNN, Naive Bayes	RF, SVM
Combined Dataset	FNN, SVM	RF, Logistic Regression
League	**U/O 2.5 Prediction**	
	Best model	**Runner-up model**
Dutch Eredivisie	Logistic Regression	RF, SVM
Belgian Jupiler	FNN, Vanilla RNN	RF, Logistic Regression
Scottish Premiership	Logistic Regression, RF	Vanilla RNN, FNN
Combined Dataset	Logistic Regression	RF, XGBoost

The practical implications of these findings are significant for sports analysts, betting companies, and team strategists. For instance, the consistent performance of the Feedforward Neural Network in predicting game results suggests that sports analysts could use this model to enhance player selection, tactical decisions, and training focus. Betting companies could use these predictive insights to adjust odds and improve customer betting options, potentially increasing betting accuracy and engagement.

To predict whether the total number of goals in a game would be under or over 2.5, the Logistic Regression model initially showed strong performance in the Netherlands league and the Scottish Premiership. However, in the Belgian Jupiler League, the Feedforward Neural Network and Vanilla RNN outperformed Logistic Regression, demonstrating their capabilities to handle complex patterns and variabilities in data. Despite this, when the datasets from the three leagues were combined, the Logistic Regression model emerged again as the best option, highlighting its robustness for this type of prediction with larger and more diverse data.

In practical terms, the adaptability of the Logistic Regression model for U/O 2.5 goal predictions makes it a valuable tool for real-time betting scenarios and dynamic game strategy adjustments. Teams could use predictions from this model to adjust their in-game tactics depending on the predicted likelihood of high-scoring games, which could be crucial for late-game decisions.

The study also highlights the strong performance of the XGBoost and Random Forest models. Both models demonstrated notable predictive capabilities in the initial training phases, with their accuracies consistently trailing the best-performing models by only 0.1 to 0.3. Although neither XGBoost nor Random Forest emerged as the top standalone predictive model, their combined use in the Voting Model proved highly effective. In the Netherlands league, the Voting Model achieved the highest accuracy for both game result prediction and U/O 2.5 goals, and the best combination consistently is XGBoost and Random Forest. This success can be attributed to Random Forest's ability to handle numerous features and reduce overfitting, along with XGBoost's robustness and precision in managing imbalanced data. These findings underscore the potential of ensemble methods in enhancing predictive performance, particularly in complex and variable-rich environments like soccer games.

The deployment of ensemble methods, such as the Voting Model that combines XGBoost and Random Forest, could significantly benefit predictive tasks in sports environments where the stakes and variables are high. This approach could improve the reliability of game outcome predictions, offering a more robust tool for decision-makers in high-pressure scenarios. This would be particularly beneficial for live game analyses and post-game strategy evaluations, where understanding gameplay dynamics is crucial.

The Vanilla RNN, while demonstrating potential in the analysis of soccer game results, did not consistently lead across all the datasets examined. In the Belgian Jupiler League, it achieved notable success with an accuracy of 0.79, an F1-score of 0.78, precision of 0.78, and recall of 0.79. Although it was one of the better-performing models in this context, it did not universally outperform all other models in various predictive tasks. This performance underlines the Vanilla RNN's ability to process sequential and complex data, suggesting that while it is a capable model for sports analytics, further

enhancements could be made to boost its accuracy and reliability in more demanding analytical scenarios.

The Support Vector Machine model displayed commendable performance, demonstrating robustness across varied datasets with its accuracy showcasing moderate fluctuations. This indicates that while SVM might not be the top performer in every scenario, its effectiveness in adapting to different data distributions makes it a reliable and valuable model in sports analytics. In contrast, the Naive Bayes model showed poor predictive performance. In training, Naive Bayes achieved an accuracy comparable to the Feedforward Neural Network's in predicting game results within the Scottish Premiership. However, in other league datasets, its performance was significantly poorer. This inconsistency highlights Naive Bayes' limitations in handling the complexity and variability of soccer game data.

Having discussed the practical implications and strengths of various predictive models, the study concludes by summarizing the key findings and suggesting directions for future research.

Conclusion

The study presented a comprehensive analysis of predicting soccer game outcomes using enhanced machine and deep learning models, focusing on the Dutch Eredivisie League, with comparisons to the Belgian Jupiler League and Scottish Premiership. The potential limitation regarding the generalizability of the findings has been proactively addressed by employing datasets from various leagues and subsequently evaluating a merged dataset. This methodological approach confirms the consistency of the predictive models across different soccer leagues. It demonstrates their robustness when applied to a composite dataset representative of a wider range of playing styles and competitive environments. These efforts significantly mitigate concerns about the generalizability of the results, suggesting that the models are well-suited for broader applications in sports analytics.

The findings highlighted the efficacy of enhanced versions of machine learning and deep learning models, particularly emphasizing the superior performance of Feedforward Neural Networks in predicting game results and the robustness of Logistic Regression for under/over (U/O) 2.5 goal predictions. Through extensive experimentation and validation, several key insights were derived. Feedforward Neural Networks have been identified as the most effective model for predicting game results in the Dutch Eredivisie, Belgian Jupiler, and Scottish Premiership, as well as in combined datasets. This demonstrates FNN's ability to adeptly manage the complex, non-linear relationships inherent in soccer game results' highly variable and unpredictable nature. On the other hand, Logistic Regression excelled in predicting U/O 2.5 goals, particularly in the Dutch Eredivisie and Scottish Premiership. Its robustness, straightforward implementation, and ability to effectively handle overfitting through regularization techniques make it the preferred model for these specific predictive tasks. Together, these findings underscore the specialized strengths of FNN for result predictions and Logistic Regression for goal predictions across diverse soccer contexts. Therefore, among the models evaluated in this study, these two models are considered the best for the different tasks.

In practical applications, these findings offer valuable insights for team strategists and betting market stakeholders. Sports analysts and coaches could integrate predictive models like FNN into pre-game planning by predicting game results and identifying performance patterns that could influence tactical decisions. For instance, understanding the likelihood of an outcome against a specific opponent can assist in tailoring training regimes or adjusting on-field formations to optimize performance. In the betting market, Logistic Regression's strength in U/O 2.5 goal predictions could provide bookmakers and bettors with a more reliable tool to set odds or make informed wagering decisions, particularly when real-time game statistics and live odds adjustments are involved. These models can also aid in refining betting strategies by predicting high or low-scoring games with greater accuracy, thus influencing market behavior and betting options.

Moreover, the runner-up models, such as Random Forest, Support Vector Machines, and Vanilla RNN, have demonstrated significant versatility and robustness. RF and SVM, frequently noted as secondary choices in both result and U/O 2.5 goal predictions, provide strong alternatives when primary models might not perform optimally. These models excel in handling diverse data types and are particularly adept at managing overfitting, making them reliable choices for predictive sports analytics. The Vanilla RNN, highlighted in the Belgian Jupiler League for U/O 2.5 predictions, showcases its utility in handling complex pattern recognitions in soccer data, proving it is a capable model for various predictive tasks.

From a real-world perspective, the Voting Model, which combines the capabilities of Random Forest and XGBoost, could be particularly advantageous for team performance analytics and sports data companies. The high accuracy achieved through ensemble approaches allows teams to track performance patterns over time and evaluate in-game tactical shifts. Sports data companies could use this for more detailed and reliable insights into team performance metrics, offering enhanced services to clubs, broadcasters, and fans. In betting markets, the ensemble approach's ability to integrate various predictive models can provide more consistent results, helping bookmakers reduce risk and improve odd-setting accuracy.

Additionally, the study demonstrated the significant effectiveness of various data augmentation techniques in addressing class imbalances, which is crucial for developing predictive models that minimize inherent biases and improve accuracy. Class imbalance, a common issue in datasets related to soccer game outcomes, can severely affect the performance and reliability of machine learning and deep learning models. Advanced data augmentation methods such as SVM-SMOTE, SMOTE-NN, and Random-OverSampling were employed to mitigate this.

Despite these promising results, the study identified several areas for future research. Expanding the analysis to include leagues with different competitive levels and playing styles could provide more comprehensive insights and validate the models across diverse environments. Incorporating real-time data and external factors like weather conditions, pitch quality, and in-game events could further enhance the predictive capabilities of the models. Future work could also explore using alternative machine learning models or hybrid approaches to further improve the accuracy and robustness of soccer match predictions.

Acknowledgements
We acknowledge the initial support from Liu Fangtian.

Author contributions
E.F.E.A.M., D.Z., and Z.Z conceptualized the studies and wrote the initial manuscript. J.L. validated the revised manuscript.

Funding
The authors acknowledge support from the Student Partnering with Faculty/Staff Research Program at Wenzhou-Kean University. Grant ID: WKUSPF202434.

Availability of data and materials
Data is available upon reasonable request.

Declarations

Competing interests
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Declaration of generative AI and AI-assisted technologies in the writing process
During the preparation of this manuscript, the authors utilized Grammarly and ChatGPT to enhance its readability and language. Following the use of these tools, the authors thoroughly reviewed and made necessary edits to the content. The authors accept full responsibility for the content of the published article.

Received: 3 July 2024 Accepted: 13 October 2024
Published online: 23 November 2024

References

1. Vernon-Carter E, Ochoa-Tapia J, Alvarez-Ramirez J. Singular value decomposition entropy of the standing matrix for quantifying competitiveness of soccer leagues. Physica A Stat Mech Appl. 2023;625: 129007.
2. Horvat T, Job J, Medved V. Prediction of euroleague games based on supervised classification algorithm k-nearest neighbours. In: 6th international congress on support sciences research and technology support; 2018;20:21.
3. Eid AIA, Miled AB, Fatnassi A, Nawaz MA, Mahmoud AF, Abdalla FA, Jabnoun C, Dhibi A, Allan FM, Elhossiny MA, et al. Sports prediction model through cloud computing and big data based on artificial intelligence method. J Intell Learn Syst Appl. 2024;16(2):53–79.
4. Wunderlich F, Memmert D. Analysis of the predictive qualities of betting odds and FIFA world ranking: evidence from the 2006, 2010 and 2014 football world cups. J Sports Sci. 2016;34(24):2176–84.
5. Berrar D, Lopes P, Dubitzky W. Incorporating domain knowledge in machine learning for soccer outcome prediction. Mach Learn. 2019;108:97–126.
6. Teoldo I, Guilherme J, Garganta J. Football intelligence: training and tactics for soccer success. Routledge; 2021.
7. Machado JT, Lopes AM. On the mathematical modeling of soccer dynamics. Commun Nonlinear Sci Numer Simul. 2017;53:142–53.
8. Lucey P, Bialkowski A, Monfort M, Carr P, Matthews I. Quality vs quantity: Improved shot prediction in soccer using strategic features from spatiotemporal data. In: Proceedings of the 8th Annual MIT Sloan Sports Analytics Conference, Boston, MA, USA, 28 February–1 March 2014; pp. 1–9
9. Gomez M-A, Reus M, Parmar N, Travassos B. Exploring elite soccer teams' performances during different match-status periods of close matches' comebacks. Chaos Solitons Fractals. 2020;132: 109566.
10. Marcelino R, Sampaio J, Amichay G, Gonçalves B, Couzin ID, Nagy M. Collective movement analysis reveals coordination tactics of team players in football matches. Chaos Solitons Fractals. 2020;138: 109831.
11. Tan X. Enhanced sports predictions: a comprehensive analysis of the role and performance of predictive analytics in the sports sector. Wirel Pers Commun. 2023;132(3):1613–36.
12. Novillo Á, Gong B, Martínez JH, Resta R, Campo RL, Buldú JM. A multilayer network framework for soccer analysis. Chaos Solitons Fractals. 2024;178: 114355.
13. Charest J, Sleep MG. Impacts on physical performance, mental performance, injury risk and recovery, and mental health; 2020. 15. 2019;5:41–57. https://doi.org/10.1016/j.jsmc
14. Maglo A, Orcesi A, Pham Q-C. Efficient tracking of team sport players with few game-specific annotations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022;3461–71.
15. Beal R, Norman TJ, Ramchurn SD. Artificial intelligence for team sports: a survey. Knowl Eng Rev. 2019;34:28.
16. Pappalardo L, Guerrini L, Rossi A, Cintia P (2019) Explainable injury forecasting in soccer via multivariate time series and convolutional neural networks. In Barça Sports Analytics Summit 2019, Barelona, 13 October 2019; 1–15. https://doi.org/10.13140/RG.2.2.31428.37765
17. Ballı S, Özdemir E. A novel method for prediction of Euroleague game results using hybrid feature extraction and machine learning techniques. Chaos Solitons Fractals. 2021;150: 111119.
18. Dijksterhuis A, Bos MW, Leij A, Van Baaren RB. Predicting soccer matches after unconscious and conscious thought as a function of expertise. Psychol Sci. 2009;20(11):1381–7.
19. Rico-González M, Pino-Ortega J, Méndez A, Clemente F, Baca A. Machine learning application in soccer: a systematic review. Biol Sport. 2023;40(1):249–63.

20. Horvat T, Job J. The use of machine learning in sport outcome prediction: a review. Wiley Interdiscip Rev Data Min Knowl Discov. 2020;10(5):1380.
21. Geurkink Y, Boone J, Verstockt S, Bourgois JG. Machine learning-based identification of the strongest predictive variables of winning and losing in Belgian professional soccer. Appl Sci. 2021;11(5):2378.
22. Bunker R, Susnjak T. The application of machine learning techniques for predicting match results in team sport: a review. J Artif Intell Res. 2022;73:1285–322.
23. Rahman MA. A deep learning framework for football match prediction. SN Appl Sci. 2020;2(2):165.
24. Zare N, Sarvmaili M, Sayareh A, Amini O, Matwin S, Soares A. Engineering features to improve pass prediction in soccer simulation 2d games. In: Robot world cup. Springer; 2021. p. 140–52.
25. Yeung CC, Bunker R, Fujii K. A framework of interpretable match results prediction in football with FIFA ratings and team formation. PLoS ONE. 2023;18(4):0284318.
26. Football-Data.co.uk. https://www.football-data.co.uk/. Accessed 21 May 2024.
27. Zovak T, Šarčević A, Vranić M, Pintar D. Game-to-game prediction of nba playersâ€™ points in relation to their season average. In: 2019 42nd international convention on information and communication technology, electronics and microelectronics (MIPRO); 2019. p. 1266–70.
28. Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd Acm Sigkdd international conference on knowledge discovery and data mining; 2016. p. 785–94.
29. Wickramasinghe I, Kalutarage H. Naive bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. Soft Comput. 2021;25(3):2277–93.
30. Biau G, Scornet E. A random forest guided tour. Test. 2016;25:197–227.
31. Dongare A, Kharde R, Kachare AD, et al. Introduction to artificial neural network. Int J Eng Innov Technol IJEIT. 2012;2(1):189–94.

Publisher's Note