

Article

Using Convolutional Neural Network and Candlestick Representation to Predict Sports Match Outcomes

Yu-Chia Hsu

Department of Sports Information and Communication, National Taiwan University of Sport,
Taichung 404, Taiwan; ychsu@ntus.edu.tw; Tel.: +886-4-2221-3108

Abstract: The interdisciplinary nature of sports and the presence of various systemic and non-systemic factors introduce challenges in predicting sports match outcomes using a single disciplinary approach. In contrast to previous studies that use sports performance metrics and statistical models, this study is the first to apply a deep learning approach in financial time series modeling to predict sports match outcomes. The proposed approach has two main components: a convolutional neural network (CNN) classifier for implicit pattern recognition and a logistic regression model for match outcome judgment. First, the raw data used in the prediction are derived from the betting market odds and actual scores of each game, which are transformed into sports candlesticks. Second, CNN is used to classify the candlesticks time series on a graphical basis. To this end, the original 1D time series are encoded into 2D matrix images using Gramian angular field and are then fed into the CNN classifier. In this way, the winning probability of each matchup team can be derived based on historically implied behavioral patterns. Third, to further consider the differences between strong and weak teams, the CNN classifier adjusts the probability of winning the match by using the logistic regression model and then makes a final judgment regarding the match outcome. We empirically test this approach using 18,944 National Football League game data spanning 32 years and find that using the individual historical data of each team in the CNN classifier for pattern recognition is better than using the data of all teams. The CNN in conjunction with the logistic regression judgment model outperforms the CNN in conjunction with SVM, Naïve Bayes, Adaboost, J48, and random forest, and its accuracy surpasses that of betting market prediction.

Keywords: convolutional neural network (CNN); time series; pattern recognition; betting odds; Gramian angular field (GAF); National Football League (NFL)



Citation: Hsu, Y.-C. Using Convolutional Neural Network and Candlestick Representation to Predict Sports Match Outcomes. *Appl. Sci.* **2021**, *11*, 6594. <https://doi.org/10.3390/app11146594>

Academic Editor: Iztok Fister

Received: 4 June 2021

Accepted: 15 July 2021

Published: 18 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Sports forecasting research has developed rapidly in recent years and begun to cover different sports, methods, and research questions. Previous studies from various disciplines have extensively investigated various predictive aspects of sports match outcomes. Several review articles [1–7] have synthesized and organized existing predictive models with different perspectives based on human heuristics, quantitative models, ratings and rankings, betting odds, machine learning, and hybrid methods and compared their prediction capabilities.

For sports betting, which is a billion-dollar industry, accurate predictive models are necessary to set the baseline for betting odds [8,9]. Given that an improper setting of betting odds will bring huge losses to betting companies, the win/loss predictions implied in the odds are highly credible. Therefore, these odds are frequently adopted by researchers in economic and financial contexts and manipulated via a time series method to examine efficiency [10,11], study irrational behavior [12], and predict wins–losses [13,14] in betting markets. However, such odds-based models mostly focus on how to beat bookmakers and measure performance based on the returns obtained from the predicted outcome in conjunction with the betting strategy [15], not pursuing the accuracy of the predictions [16].

Similar to assessing risky financial market behavior, sports competition outcome predictions are assessed based on systemic and non-systemic effects [7]. Systematic effects are mainly related to the strength of teams, whereas non-systemic effects are some random factors. In general, systematic effects are relatively mature to analyze. By using the past performance of each team, various quality measures for teams or players and statistical-based metrics for building prediction models have been introduced. Experienced practitioners often use heuristics to measure the strengths and weaknesses of teams and to determine the winner of a match. By contrast, non-systemic effects are usually difficult to predict. Some predictive models that have been comprehensively tested and applied in stock markets over the past decade, such as stochastic processes, probability distribution models, and Monte Carlo simulations, have also been applied in sports forecasting.

However, some scholars argue that stock price fluctuations do not completely follow the stochastic process but rather show preceding and succeeding relationships that highlight behavioral patterns. Consequently, the behavioral patterns of these relationships have been widely explored in recent years through pattern recognition based on technical indicator analyses and knowledge graphs, such as candlesticks for prediction modeling. Candlesticks can clearly show the degree of fluctuation and are commonly used in stock market analysis and prediction, as well as in a few novel studies for the prediction of teens' stress level change on a micro-blog platform [17], prediction of air quality changes [18], analysis of the teaching model effectiveness [19], and prediction of sports match outcomes [20]. While the results of sports games are traditionally considered independent events that do not affect one another, some scholars have recently proposed that these games are not entirely independent events and that certain trends or regularities may occur as a result of previous events [6]. Some scholars have also suggested to exclude many factors that affect winning and losing, and considered only the scores and the match results as the raw data, using the score-driven time series to build prediction model of football match outcomes [21,22]. In addition, in professional sports with intensive schedules, a win or loss in the previous match can bring anxiety for the player and generate a certain impact on the next match, such as the "hot hand" effect [23,24]. Therefore, behavioral patterns may exist in sports matches. Similar to the stock market behavior resulting from the psychological reaction of the investing public, the behavior of a sports team is influenced by the psychological reaction of several players. In team sports, the outcome of a match is easily influenced by the main powerful player on the field and can greatly vary across each player [25]. Identifying these implicit behavioral patterns of sports remains extremely challenging.

This paper proposes an approach for sports match outcome prediction with two main components: the convolutional neural network (CNN) classifier for implicit pattern recognition and the logistic regression model for matching outcome judgment. This novel prediction model cleverly transforms the prediction problem, which is often solved using numerical parametric models, into a time series classification and prediction problem to exploit the strengths of deep learning techniques in visualizing behavior recognition. The significance of this study lies in the interdisciplinarity of integrating sports, econometrics, computer science, and statistics to develop a hybrid predictive model. The most important contributions of this paper are the framework of the hybrid machine learning approach and the design of each processing procedure, as well as the empirical results from real data. With respect to the state of the art, the following three main features are summarized.

First, we propose an alternative approach based on well-established betting market forecasts without considering too many sport performance indicators, thereby ensuring consistency in practice across various sports types. Similar to many previous studies on sports outcome prediction, we use machine learning as our basis for developing predictive models. However, previous studies have mostly focused on the selection of sports performance indicators, their feature capture approach, and the input variables for the prediction models. In this paper, we argue that such models are highly specific to a certain type of sport and that the parameters adopted in different competitions also vary.

Second, our predictive model is capable of learning the behavioral patterns of sports teams from historical scores and betting market odds. To achieve this, we employ CNN to leverage its recent significant success in computer vision and speech data classification. For the 2D matrix data required for image recognition, in which CNN operates excellently, we represent the behavioral characteristics of the team as an image encoded from OHLC time series in candlesticks, which are widely used in financial market forecasting. Unlike candlesticks in the stock market that demonstrate well-known patterns, the behavioral patterns of sports teams in preceding and following games are implicit and not easily detected. CNN has the potential to identify unknown patterns by classifying them through image recognition. Combining candlestick charts with CNN for sports prediction is a novel contribution of this study.

Third, we adopt a two-stage strategy to design our prediction model following a heuristic approach within the domain knowledge of sports. Initially, we assess the current status and recent performance trends of each team, and then adjust its assessment by plus or minus based on the strengths and weaknesses of the opponents in a match and judge the outcome. In this two-stage approach, the CNN classifier in the first stage recognizes the graphical behavioral pattern and computes the classification probability of each team to win in the next match. Meanwhile, the logistic regression in the second stage learns the CNN winning probabilities and the actual results of the two teams by pair in past matches to build the outcome judgment model.

The rest of this paper is organized as follows: Section 2 reviews the relevant literature; Section 3 defines the proposed two-stage approach; Section 4 provides a thorough experimental evaluation of US National Football League (NFL) matches from seasons 1985/1986 to 2016/2017; Section 5 concludes the paper.

2. Related work

2.1. Sports Outcome Prediction with CNN

Most applications of CNN in the field of sports are based on images, videos, or sensor data as data sources. The application examples include classifying six categories of cricket shots from 3600 images [26], recognizing tennis actions based on CNNs with historical long short-term memory network (LSTM) from sports videos [27], and classifying wearable sensor data from beach volleyball players to monitor activity and prevent injury [28]. However, only a few studies have transformed raw sports competitions data to exploit the benefits of CNN. The main reason is that when using CNNs for sports competition data, the greatest challenge is to appropriately convert the non-graphical numerical data into a multidimensional matrix that complies with the CNN input norm.

Lin et al. [29] and Chen et al. [30] used CNN to predict the outcome of NBA games. They used data of more than 4000 games covering three seasons and coded the score of each player from both two teams as 1, 0, and -1 to represent the contributions of each player toward winning. Both teams, each with a maximum of 16 players, form a 16×16 2D matrix by cross-comparison and are used for CNN prediction modeling.

Hubáček et al. [31] applied CNN with NBA data from seasons 2007 to 2014 for match outcome prediction. The convolutional layer takes advantage of a large number of player-related statistics as its input. In the input matrix, the rows represent the players, whereas the columns represent the player-level features grouped by two matrices, namely, home team and away team.

Manivannan and Kausik [32] utilized CNN to predict the outcome of cricket matches by learning the end-to-end mapping between the performance of players and the outcome of matches. The input data are used to build a representation matrix of two match teams as obtained by encoding the feature of each player and then aggregating each player to the team-level data.

Huang and Li [33] used CNNs in predicting the outcomes of Major League Baseball matches. However, instead of exploiting the advantages of 2D CNN in computer vision

recognition, they used a 1D CNN model with a total of about 30 statistical values for batting and pitching as variables, which were treated as signals and time series to analyze the data.

The aforementioned studies on match result prediction have mainly considered the advantages of CNN in analyzing large amounts of data and represented the performance of each player in the team as an encoded feature matrix. As a CNN input, the 2D matrix merely represents multiple features after encoding and is not treated as an image in computer vision. Each sports match is considered a sequence of discrete-time data, including spatiotemporal or statistical information, that lack a consensual visual representation. Even if consistent knowledge graphs are available for representation, the implicit patterns in these graphs are difficult to realize. To exploit the power of CNN in image-based classification for sports outcome prediction, we visualize sports competition data as candlesticks and leverage the time series classification technique.

2.2. Time Series Prediction with CNN

Unlike previous parametric and nonparametric models for building predictive models through machine learning, CNNs leverage their significant achievements in computer vision classification to address the prediction problems. Following this idea, numerical data are transformed into a knowledge graph for the easy extraction of feature matrices, and then these matrices are fed into CNN to mine behavioral patterns.

In stock price prediction, Kim and Kim [34] proposed a feature fusion LSTM-CNN model that combines the features of different representations of financial time series data. They constructed four stock chart images as inputs to examine their predictive performance and found that candlestick charts perform the best. To utilize the power of deep CNN for an algorithmic trading system, Sezer and Ozbayoglu [35] proposed a novel approach that converts 1D financial time series into a 2D image-like data representation from 15 selected technical indicators and labeled them as three categories of investment, namely, buy, sell, and hold, depending on the hills and valleys of the original time series. To predict stock market price trends, Long et al. [36] dealt with the correlation between stocks by building a company knowledge graph and applied the node2vec algorithm [37], which has been adopted in many studies [38], to obtain a vector representation of company nodes and to facilitate the subsequent feeding into the CNN for investment pattern recognition.

Time series classification studies have proposed several techniques for converting the original 1D data into the preferred 2D format to allow CNN to take advantage of the deep CNN classifier. For instance, Hatami et al. [39] used recurrence plots (RP) to transform the 1D signals of time series into 2D texture images. Wang and Oates [40,41] used Gramian angular fields (GAF) and Markov transition fields (MTF) to encode time series signals as images. Yang et al. [42] used Gramian angular summation field (GASF), Gramian angular difference field (GADF), and MTF to encode multivariate time series data into 2D colored images for classification through CNN.

For the time series prediction problem on traffic flow, Ma et al. [43] represented spatio-temporal traffic flow data as images for the time series prediction problem and then examined the performance of the CNN model. Deng et al. [44] investigated an image-based spatio-temporal data representation approach for predicting traffic flow by using convolutional layers and ensemble learning models. Asadi and Regan [45] applied a time series decomposition method to separately feed short-term, long-term, and spatial patterns into different components to identify clusters of neighboring time series residuals and then seeded a single cluster of input time series for each convolutional kernel of neural networks. Conversely, for the time series prediction problem on the stock market, Chen et al. [46] adopted two transformation strategies that test the predictive performance of the CNN model for stock prices by converting the original time series into 2D images via GAF and by computing the moving averages with different time frames to obtain 2D time series expressions and compare them with a candlesticks chart.

In a comprehensive review of the literature on financial time series forecasting [47], several articles have used raw time series, such as “open”, “high”, “low”, and “closing”

(OHLC) prices and volume, as their feature set to predict stock prices. This simple approach inspires us to ignore numerous sports performance indicators and statistical information and merely focus on matching-score-related information as our feature set for predicting sports outcomes.

2.3. OHLC Time Series Representing and Encoding for Sports Matches

CNNs can retrieve implicit patterns from knowledge graphs and time series for classification. Therefore, we feed the matching scores and betting odds of each match into the CNN via candlestick representation with time series.

The sports gaming market shares many similar characteristics with the financial market [48]. Odds in the betting market, which are comparable to commodity prices in the financial market, are implicit representations of various information, including past performance of players and teams, a mix of immediate and asynchronous information, and various components of sports fan psychology. The efficient market hypothesis for commodity prices in the financial market is also observed in the gaming market [49]. Therefore, we propose that the sports competition behavior is applicable to the technical analysis charts of the stock market. Historical data from the betting market, including odds and match scores, can be manipulated and used to chart candlesticks and identify matching characteristics [50,51]. These data can also be utilized in machine learning modeling to develop predictive models [20].

GAF is a transformation method based on a polar coordinate system that enables the visual interpretation of 1D time series by converting them into a 2D image. The original time series is represented in a polar coordinate system by encoding its values as an angular cosine, with the time stamp as the radius, while retaining temporal dependence. GAF is implemented in two forms, namely, GASF and GADF, which differ in their use of the angular perspective to consider the trigonometric sum/difference between each point and identify the temporal correlation across different time intervals. The conversion results of these GAFs do not greatly differ, and both of these approaches have been widely used in various fields to deal with time series data.

GADF and GASF have been applied to map the measured electromagnetic interference time signals to an image for electromagnetic interference discharge source classification [52]. On the one hand, GADF has been used to transform ECG time series into images to extract sparse high-dimensional features that perform well in linear classifiers for detecting myocardial infarction [53]. On the other hand, GASF has been used to convert data from multiple wearable inertial measurement units into images to classify load-carrying modes in construction tasks [54]. For the OHLC time series of stock prices, GAF has been adopted as the image encoding method for image-based classification with CNN [55].

We propose using a candlestick chart to represent the performance of each team in a series of sporting matches. The OHLC time series, which convey the candlesticks representation of sports matches, are encoded into images by using GAF such that the visualization features can be easily recognized for CNN to learn the implied patterns.

3. Methodology

3.1. Procedure of the Proposed Model

We propose a two-stage approach for the sports prediction model. The related framework is illustrated in Figure 1.

whether a bet is won or lost. These spread, total points scored, and actual points scores are plotted for each team’s candlestick according to the method proposed by Mallios [50]. Further details can be found in the article on sports metrics [51] and the previous study [20]. The candlestick chart defined by OHLC is formulated as follows:

$$O = LD \tag{1}$$

$$H = \begin{cases} \max(D, LD) + GST \text{ (if } GST > 0) \\ \max(D, LD) \text{ (if } GST < 0) \end{cases} \tag{2}$$

$$L = \begin{cases} \min(D, LD) \text{ (if } GST > 0) \\ \min(D, LD) - GST \text{ (if } GST < 0) \end{cases} \tag{3}$$

$$C = D \tag{4}$$

$$\text{Body color} = \begin{cases} \text{White (if } D > LD) \\ \text{Black (if } LD > D) \end{cases} \tag{5}$$

where D is the team winning/losing margin (i.e., the difference in points scored between the team and its opponent); LD is the line (or spread) on the winning/losing margin between the team and its opponent (i.e., the side line in the betting market); T is the total points scored by both teams; LT is the line on the total points scored by both teams (i.e., the over/under (O/U) line in the betting market); GSD presents the gambling shock related to the line on the winning/losing margin between the team and its opponent (i.e., $D-LD$); GST provides the gambling shock related to the line on the total points scored by both teams, which can calculate the difference between LT (the line on the total points scored by both teams) and T (the actual total points scored) (i.e., $T-LT$).

The candlestick chart for each team can be obtained in this manner. We take New England with the highest winning percentage and Cleveland with the lowest winning percentage of the 2017/2018 NFL season as examples, as shown in Figures 2 and 3.

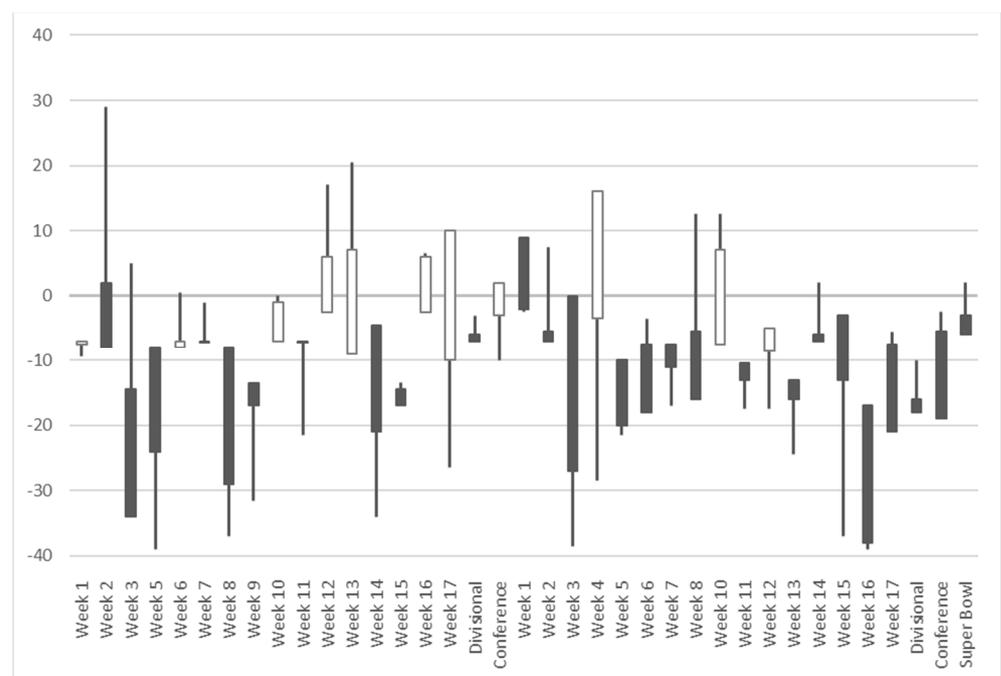


Figure 2. Candlestick chart of New England in 2015/2016 and 2016/2017.

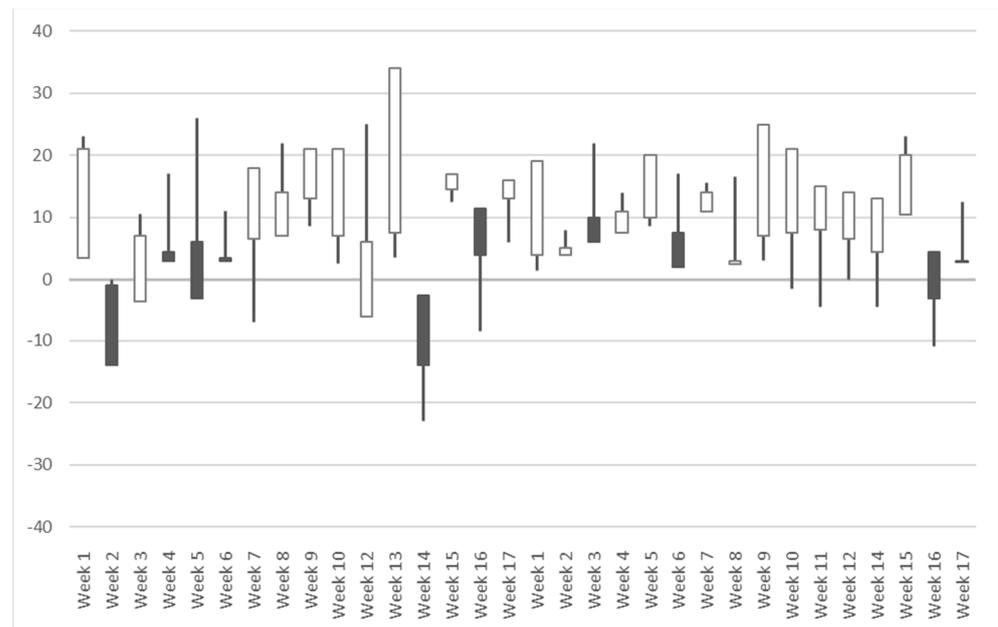


Figure 3. Candlestick chart of Cleveland in 2015/2016 and 2016/2017.

As shown in Figures 2 and 3, the OHLC sports time series include positive and negative values, which greatly differ from the traditional time series of stock prices, where all values are greater than 0. The OHLC sports time series reflect the gap between two teams in the match. These time series are equivalent to the stationary return time series of the stock prices, which are derived by differencing its original non-stationary price time series. Stationary time series forms are beneficial for time series prediction modeling. However, the existence of meaningful patterns in sports candlesticks, such some identified patterns (e.g., morning star, bullish engulfing, etc.), warrants further investigation.

3.3. OHLC Time Series Data Encoding as Images by GAF

Although candlestick charts are examples of knowledge graphs, they cannot be directly fed into CNN because the appearance of candlesticks made of pixels will greatly vary depending on the scale of the drawing, the ratio of the x and y axes, and the size of the image pixels. Therefore, we use OHLC time series to represent candlesticks and address the inconsistency in candlestick appearance. The segmented 1D data of the original time series are converted into a 2D matrix, and the matrix size is treated as the pixel size of the graph. We use GAF to encode the time series as input images for computer vision.

First, time series $X = \{x_1, x_2, \dots, x_n\}$ is normalized to \tilde{X} by using Equation (6). Given that OHLC can either be negative or positive, the data are rescaled between -1 and 1 .

$$\tilde{x}_i = \frac{(x_i - \max(X)) + (x_i - \min(X))}{\max(X) - \min(X)} \tag{6}$$

The normalized data are then encoded into polar coordinates by calculating the angles and radius by using the angular cosine and time stamp. GASF uses the cosine function as defined in Equations (7) and (8). Each element of the GASF matrix represents the cosine of the summation of angles \emptyset . In the image matrix obtained by GASF, the diagonal elements from top-left to bottom-right represent the symmetric axis of the images, and the position corresponds to the original time series.

$$GASF = \begin{bmatrix} \cos(\emptyset_1 + \emptyset_1) & \cdots & \cos(\emptyset_1 + \emptyset_n) \\ \vdots & \ddots & \vdots \\ \cos(\emptyset_n + \emptyset_1) & \cdots & \cos(\emptyset_n + \emptyset_n) \end{bmatrix} \tag{7}$$

$$GASF = \tilde{X}' \cdot \tilde{X} - \sqrt{I - \tilde{X}^2} \quad (8)$$

where I is the unit row vector, \tilde{X} and \tilde{X}' represent different row vectors.

We generate a 4D matrix that comprises the OHLC time series from the candlesticks. Each set of 1D time series data is encoded as 2D images by using GASF. Given that the candlestick representation includes all four OHLC time series, each series can be considered a channel that represents the image color. Given that we use the preceding 10 games to predict the next game, the input values of the CNN are 10×10 2D matrices in four channels as shown in Figure 4.

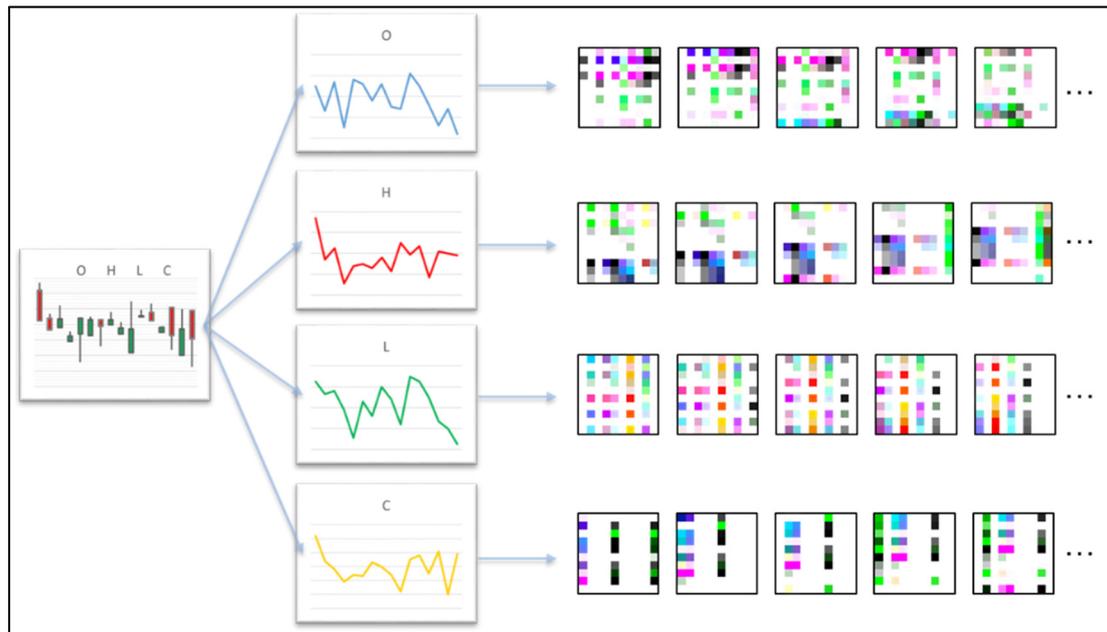


Figure 4. Candlesticks of OHLC time series encoded by GAF into four channels of 10×10 images, which are used as input for CNN classifier.

3.4. Win Probability Estimation Based on CNN Classification

CNN specializes in image classification, and its accuracy has recently surpassed that of humans in various image recognition applications. With regard to the behavioral patterns in sports matches, we represent the odds and scores of matches as candlesticks in order to understand the fluctuations in time series. However, the time series transformed into 2D images by GAF is a highly in-depth knowledge graph, and humans cannot easily observe the features and patterns on this graph given their unknown style. Therefore, we exploit the classification capability of CNN to recognize the implied patterns from these image-based time series.

Figure 5 shows our CNN architecture, which originates from the most popular, small, and easy architecture, LeNet-5 [56], and is enhanced by incorporating the concepts of rectified linear units (ReLU), max-pooling, and dropouts that have been proposed in recent years. After some testing and fine tuning, the adjusted architecture has been proven to work well.

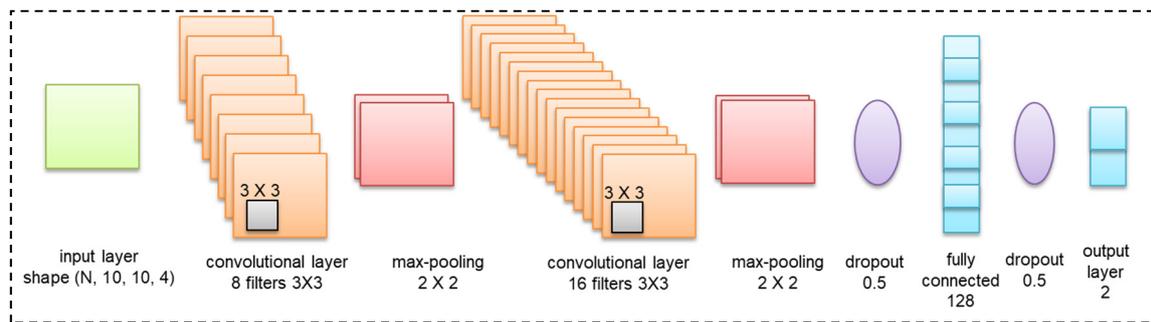


Figure 5. The CNN architecture.

As shown in Figure 5, the CNN comprises two sets of convolutional, activation, and max-pooling layers followed by a dropout, fully connected layer, activation, another dropout, and a softmax classifier. The detailed structure and hyperparameters of each layer are described follows:

3.4.1. Input Layer

We encode the four OHLC time series that constitute the candlestick into 2D images with GADF as the input variable of CNN. We predict the next match outcome based on the information of the preceding 10 matches. We set the window size to 10 and fix the size of the 2D-transformed GADF image to a 10×10 pixel that comprises the four channels O, H, L, and C. After this step, the shape of the data matrices will be (10, 10, 4).

3.4.2. Convolution Layer

Two convolutional layers with a kernel size of 3×3 are used to match the input pixel size of 10×10 , and the excitation function is set to the ReLU function.

3.4.3. Max-Pooling Layer

Two max-pooling layers with a window of 2×2 pixels are used for general image classification by calculating the maximum value for each patch of the feature map.

3.4.4. Dropout Layer

The data used for sports forecasting are expressed on a game-by-game basis. Despite having decades of retrospective data, our amount of data is far less than a million. Having a small amount of data can easily lead to overfitting problems when training CNN models. Therefore, we apply a dropout with a probability of 0.5 after the convolution and FC layers to reduce overfitting.

3.4.5. Fully Connected Layer

We apply two fully connected layers. The first layer is set with 128 dense and ReLU activation functions, and the second FC layer is set with the softmax activation function.

For our deep learning model, we select the adaptive learning optimizer Adam [57] as our optimization algorithm to quickly obtain favorable results. To reduce the learning rate, which is typically necessary to achieve superior performance, we choose the simple learning rate schedule of reducing the learning rate by a constant factor when the performance metric plateaus on the validation/test set (commonly known as ReduceLRonPlateau). The output value of CNN represents the binary classification result of win or loss. We take the classification probability value of win for each team to determine the prediction outcome of the match by using the judgment model in the next stage.

3.5. Outcome Judgment Model of Logistic Regression

We use the CNN classifier to predict the next game based on the past performance of the team by using its own behavior as the learning object. This approach may be explained by the psychological changes in the morale and confidence of the team when facing an opponent and by the effect of the intensity of previous matches on the next match. However, given that the strong team has won many games in the past (thus making the CNN prediction to be mostly won), the CNN prediction for the weak team is mostly lost; this would make it very difficult to predict when a strong team meets another strong team or a weak team meets another weak team. Therefore, when the teams are separately classified as winners or losers in the first stage based on their historical behavior, we need to incorporate a judgment model that considers the win probabilities of both teams to determine the final predicted outcome.

Many studies that predict NFL game wins/losses have used a similar concept and dealt with numerical variables such as using the win probability [58] of both teams as the output or calculating their rating scores and ranking [59–61]. These teams are then compared before deciding the outcome. We take the probability values of the winning and losing classification results from the CNN classifier and feed them into the logistic regression model to learn and evaluate the final result.

The simple outcome judgment model is defined as follows. Considering team A vs. team B, the probability of each team winning the match is estimated by P_a and P_b . Given that P_a and P_b are estimated separately without considering the opponent, both of them take values between 0 and 1, and their total value ranges between 0 and 2 and is not necessarily equal to 1. If P_a is greater than P_b , then we predict that team A wins, and vice versa. The logistic regression judgment model takes the name of team A, the probability of winning P_a , the name of opponent team B, and the probability of winning, a total of four variables, to build a prediction model, as shown in Figure 6.

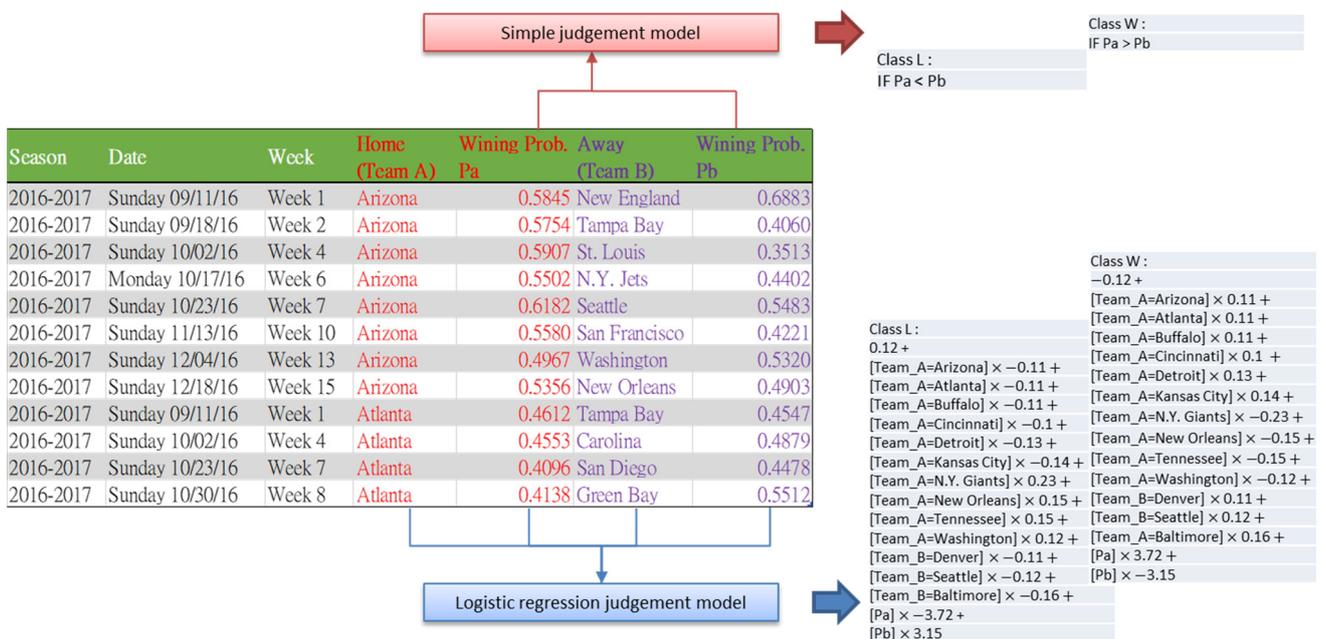


Figure 6. The outcome judgment model in the second stage.

Logistic regression is a classification method that is especially suitable for predictive models with binary outcomes whose output values are restricted between 0 and 1 and can be interpreted as probabilities. Therefore, logistic regression has been used in predicting sports outcomes, such as soccer matches [62], used together with a Markov chain model to predict National Collegiate Athletic Association football rankings [63], or incorporated with data envelopment analysis to predict match outcomes of National Basketball Association [64].

The outcome judgment model of the logistic regression is expressed as follows. Let Y be the variable of the binary reaction (win or loss) and p be the probability of win influenced by independent variable x . The relationship between p and x is given by:

The probability of success for event Y is computed as:

$$p = \frac{e^{f(x)}}{1 + e^{f(x)}} \quad (9)$$

x : Classification output of CNN for teams A and B in the match

Y : Probability of team A winning

When the strengths of both teams are close to each other, the difference between the classification probabilities originally obtained from the CNN is not significant. Such difference can be highlighted through an exponential change in the effect of the independent variables on the dependent ones in the logistic distribution to help determine the winner.

4. Experiments

4.1. Dataset

We collected 32 years' worth of NFL match data from the 1985/1986 to 2016/2017 seasons to investigate CNN match outcome prediction models. These data include the odds of two betting lines and the actual points scored, which are publicly available and can be obtained from covers.com (<https://www.covers.com/sport/football/nfl/teams/main/buffalo-bills/2017-2018> (accessed on 29 October 2020) on a team-by-team, year-by-year basis via web crawling. The data were collected for both regular season and playoff games, covering a total of 18,944 games for 32 teams.

We organized our dataset by team and organized the time sequence of each team in the order of game time. The last game of the previous year was followed by the first game of the following year in the time series. The data for each team include various types of information (e.g., date, number of games, opponents, home or away games, win or loss results) and numerical data (e.g., point spread, line on side, line on total point, and points scored by both teams). These numerical data were converted into candlesticks to obtain four OHLC time series, and each 1D time series was converted into a 2D graph by GAF.

Figure 7 illustrates the data segmentation approach used to build a time series prediction model. First, we used the rolling windows approach by setting the window size to 10 (i.e., the candlestick of the previous 10 matches) and used CNN to find the implied behavioral patterns for predicting the next match. After windows rolling precession, we divided our dataset into the training and test sets by year. We used the first 31 years for the CNN model learning, and 90% and 10% of these data were randomly scattered as training and validation sets to verify the model learning. The last one was the test set, which serves as an out-of-sample assessment for ensuring the accuracy of predictions.

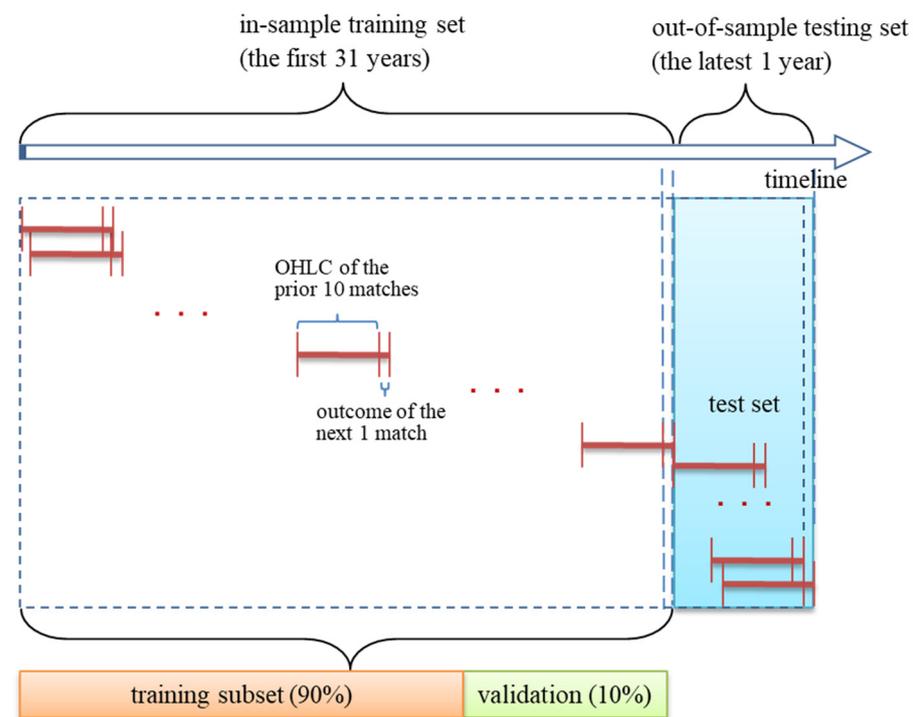


Figure 7. Data segmentation.

4.2. Experiment Designs

We conducted two series of experiments to verify the effectiveness and predictive performance of our proposed prediction model based on the sports candlestick pattern from all and individual teams and to test the approaches we used in the outcome judgment model. In the first experiment, we evaluated the effectiveness of GAF-encoded images derived from the OHLC series of sports candlesticks when classified by CNN in the first stage and tested whether the implied behavioral patterns for providing predictions should be based on the whole team or on individual teams. The second experiment aimed to test the performance of the two-team match outcome judgment model in the second stage. We used different machine learning algorithms to check whether the prediction performance was acceptable when compared with each other and with the comparison group.

To evaluate the experiment results, for the first stage of CNN classification capability, we compared the accuracy of the test sets. For the second stage, we assessed the performance of the outcome prediction models based on different machine learning approaches, which in turn were assessed based on the accuracy of the win/lose outcome. The precision, recall, and F-measure corresponding to the outcome were also computed to reveal further details. A higher value of these measures is generally favored. These measures are defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad (10)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (11)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (12)$$

$$\text{F-measure} = \frac{2 \times TP}{2 \times TP + FP + FN} \quad (13)$$

where TP , TN , FP , and FN refer to true positive, true negative, false positive, and false negative, respectively.

To further compare whether the two-stage approach proposed in this study helps to improve the single-stage approach, we also build a logistic regression model for OHLC

time series in the first stage. We normalize the OHLC time series using Equation (6) and set the window size to 10, making it consistent with the input values used for the GAF conversion in the first stage CNN classifier. Therefore, the input values of the logistic regression model in the first stage are the team name, the opponent team name, the team's OHLC value in the preceding 10 consecutive games, and the opponent team's OHLC value in the preceding 10 consecutive games, a total of 82 variables. The output value is the team's win or loss in the present match.

The experimental results were also compared with "betting" and "home" as benchmarks. Betting refers to the bookmaker's prediction, which is based on the betting odds of the winning/losing margin. A negative value indicates a win, whereas a positive value indicates a loss. Home refers to the well-known home field advantage, where the home team is always predicted to win.

The kappa coefficient is a measure of inter-rater agreement involving binary forecasts, such as win–lose [65]. We used this coefficient to measure the levels of agreement among the proposed prediction model, the comparison groups, and the actual outcome.

4.3. Comparison of Candlestick Pattern Recognition with CNN

Our first task was to examine whether the patterns implied by sports candlesticks can reflect past behavior and if they are useful in predictive models. Several patterns in stock market candlesticks have been proven to be useful in explaining investor behavior and forecasting purposes. These patterns are not limited to a particular stock but can be applied to any stock. Therefore, we checked for common behavioral patterns among different teams based on sports candlesticks. We conducted two experiments by using the data of all teams and the data of each team as our data source for CNN learning. The experiment results reveal that either all teams share a common pattern or that each team is suitable for the prediction based on their historical behavioral patterns. The results are shown in Table 1.

Table 1. Comparison of the accuracy of modeling with all teams' data and each team's data.

	Actual Win Percentage	All Teams			Each Team		
		CNN Classifier	Simple Judgment Model	Logistic Regression Judgment Model	CNN Classifier	Simple Judgment Model	Logistic Regression Judgment Model
New England	89.47%	84.21%	60.00%	70.00%	89.47%	80.00%	80.00%
Dallas	76.47%	64.71%	66.67%	77.78%	23.53%	55.56%	77.78%
Kansas City	70.59%	58.82%	55.56%	66.67%	64.71%	77.78%	77.78%
Oakland	70.59%	64.71%	87.50%	87.50%	82.35%	75.00%	75.00%
Atlanta	68.42%	57.89%	54.55%	45.45%	31.58%	36.36%	36.36%
Pittsburgh	68.42%	68.42%	66.67%	77.78%	68.42%	66.67%	88.89%
N.Y. Giants	64.71%	47.06%	62.50%	75.00%	64.71%	87.50%	87.50%
Green Bay	63.16%	47.37%	22.22%	55.56%	57.89%	44.44%	66.67%
Seattle	61.11%	77.78%	88.89%	77.78%	66.67%	88.89%	100.00%
Miami	58.82%	70.59%	62.50%	75.00%	58.82%	100.00%	87.50%
Denver	56.25%	50.00%	37.50%	50.00%	56.25%	62.50%	75.00%
Tampa Bay	56.25%	43.75%	62.50%	50.00%	43.75%	37.50%	37.50%
Tennessee	56.25%	37.50%	25.00%	37.50%	75.00%	62.50%	50.00%
Houston	55.56%	38.89%	55.56%	55.56%	27.78%	33.33%	66.67%
Detroit	52.94%	23.53%	50.00%	62.50%	47.06%	37.50%	100.00%
Baltimore	50.00%	62.50%	25.00%	50.00%	37.50%	75.00%	75.00%
Indianapolis	50.00%	62.50%	75.00%	75.00%	62.50%	75.00%	75.00%
Minnesota	50.00%	37.50%	50.00%	62.50%	68.75%	25.00%	62.50%
Washington	50.00%	37.50%	62.50%	37.50%	62.50%	62.50%	75.00%
Arizona	43.75%	43.75%	62.50%	62.50%	43.75%	50.00%	62.50%
Buffalo	43.75%	56.25%	75.00%	75.00%	62.50%	87.50%	75.00%

Table 1. Cont.

	Actual Win Percentage	All Teams			Each Team		
		CNN Classifier	Simple Judgment Model	Logistic Regression Judgment Model	CNN Classifier	Simple Judgment Model	Logistic Regression Judgment Model
New Orleans	43.75%	43.75%	62.50%	50.00%	31.25%	50.00%	62.50%
Philadelphia	43.75%	43.75%	37.50%	50.00%	62.50%	50.00%	75.00%
Carolina	37.50%	43.75%	50.00%	50.00%	56.25%	25.00%	25.00%
Cincinnati	37.50%	37.50%	75.00%	62.50%	62.50%	62.50%	50.00%
N.Y. Jets	31.25%	31.25%	75.00%	62.50%	50.00%	62.50%	50.00%
San Diego	31.25%	56.25%	37.50%	37.50%	68.75%	62.50%	75.00%
St. Louis	25.00%	75.00%	87.50%	87.50%	75.00%	62.50%	62.50%
Chicago	18.75%	81.25%	50.00%	25.00%	68.75%	75.00%	50.00%
Jacksonville	18.75%	75.00%	75.00%	62.50%	75.00%	62.50%	87.50%
San Francisco	12.50%	87.50%	62.50%	50.00%	87.50%	75.00%	75.00%
Cleveland	6.25%	75.00%	87.50%	62.50%	93.75%	87.50%	75.00%
Overall average	49.63%	55.99%	59.55%	60.30%	60.11%	62.17%	69.29%

Table 1 lists the teams in a descending order according to their actual win percentage; that is, the percentage of wins out of all matches played during the test period. The prediction accuracy of the two experiments was grouped into three measures: the CNN prediction classification in the first stage, the simple judgment in the second stage, and the logistic regression judgment models in the second stage.

First, we investigated the utility of our two-stage approach by comparing the CNN classifier prediction results in the first stage with the judgment model results in the second stage. Based on the average accuracy obtained in all prediction cases, we found in both experiments that the accuracy is lowest if only the first-stage CNN was used. However, such accuracy improves when the second-stage judgment model was added. This model considers the CNN prediction values of both teams in the match. Only slight improvements in accuracy were observed when the simple judgment model was used, and the best accuracy was obtained when using logistic regression.

Second, we compared the effect of introducing overall and individual behaviors into the prediction model. The prediction accuracy of using overall behavior was less acceptable than that of using individual behavior, and the average accuracy of the best logistic regression judgment model was only 60.30%, which was not meaningful. Meanwhile, the average accuracy of using individual behavior, whether CNN, simple judgment, or logistic regression judgment, was high, with the best logistic regression judgment model reaching an acceptable value of 69.29%.

Third, we addressed the differences across each team in the prediction models. We plotted the accuracies of the CNN prediction classification and judgment model according to the overall and individual behaviors in Figures 7 and 8, respectively. The x-axis denotes the ranking of actual win percentage, which represents the strength of the team, and an x-value of 1 is ranked first. Meanwhile, the y-axis shows the accuracy with values ranging from 0% to 100%. The curves of accuracy were added to the figure with trend lines to highlight the tendency of each team's strength and weakness rankings.

Figure 8 presents the results of the overall behavior experiment. The teams on the left and right sides of the graph reported higher accuracies than those at the center. The trend line showed a smile-like curve, thereby suggesting that the strong and weak teams had fixed behaviors in terms of winning and losing, thereby facilitating predictions.

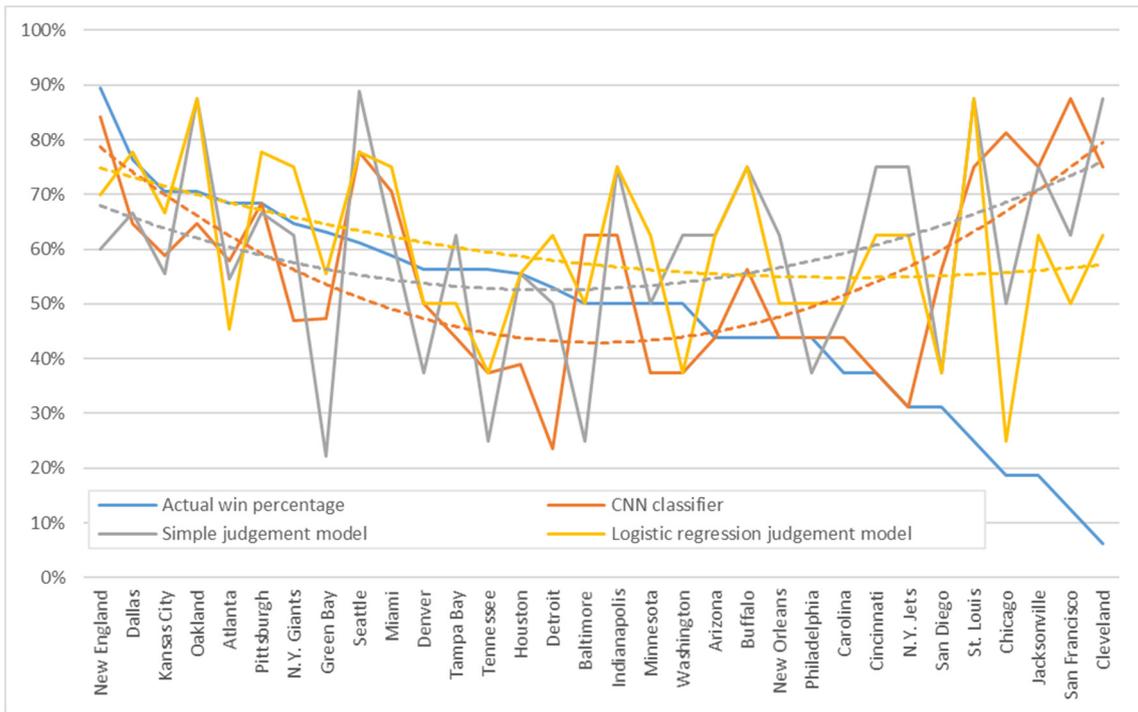


Figure 8. Accuracy comparison of the overall behavior (all teams).

Figure 9 presents the results of the individual behavior experiment. The prediction accuracy of the team at the center of the graph improved compared to that shown in Figure 8. Although such accuracy varied across teams, the trend line was close to the same level, thereby indicating the minimal difference between the strong and weak teams.

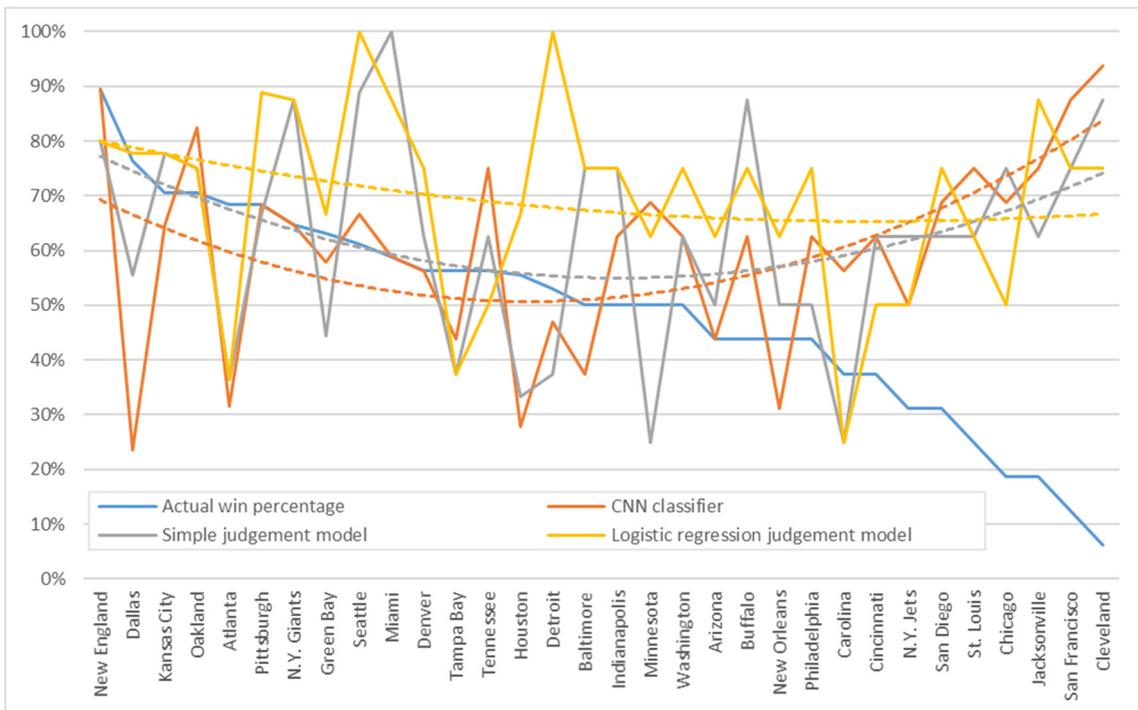


Figure 9. Accuracy comparison of the individual behavior (each team).

4.4. Comparison of Different Approaches in the Outcome Judgment Model

We used the best learning strategy for each team to learn individually based on their own data. After building the CNN classifier in the first stage, we calculated the CNN prediction values of all experimental data, including those for the training and test set. In the second stage of the judgment model, we used the CNN output values of both teams in the training set to learn the judgment model and then verified the final prediction performance in the test period.

In addition to logistic regression and the simple judgment model, we adopted sequential minimal optimization (SMO) for support vector machines (SVM), Naïve Bayes, Adaboost, J48, random forests, and multi-objective evolutionary algorithms (MOEA) for fuzzy classification, which are commonly used in machine learning, in our experiments to compare the prediction accuracy of the judgment model. The results are shown in Table 2.

Table 2 also presents the accuracy of two single-stage models, and these values were used as benchmarks for comparison with two-stage models. As shown in Table 2, the accuracies of the first-stage CNN classification and the first-stage logistic regression were 60.11% and 61.42%, respectively. It revealed that when using OHLC time series to build a prediction model for sports matches, the CNN classifier with graphical classification to process the time series was not as accurate as the logistic regression model that took into account the strengths and weaknesses of the two teams. Although these values of the single-stage model were higher than the 58.05% accuracy of home in the comparison group, they were not as good as the 63.67% accuracy of betting.

Table 2. Comparison of different approaches used in the judgment model.

Approach		Accuracy	Precision	Recall	F Measure	Kappa
First stage	Second Stage					
CNN classifier	Logistic regression	69.29%	0.6923	0.4821	0.5684	0.3417
	SMO for SVM	65.17%	0.6508	0.3661	0.4686	0.2386
	Naïve Bayes	62.55%	0.5638	0.4732	0.5146	0.2135
	Adaboost	67.04%	0.6176	0.5625	0.5888	0.3148
	J48	65.92%	0.6154	0.5000	0.5517	0.2815
	Random forests	59.93%	0.5253	0.4643	0.4929	0.1637
	MOEA for fuzzy classification	66.29%	0.6486	0.4286	0.5161	0.2737
	Simple judgment model	62.17%				
CNN classifier		60.11%				
Logistic regression		61.42%				
Home		58.05%				
Betting		63.67%				

After incorporating the second stage to the output values of the first stage CNN classifier, the accuracy of random forest decreased, whereas that of Naïve Bayes grew slightly larger than that of simple judgment. The rest of the machine learning algorithms increased the prediction accuracy above that of betting (63.67%), and the best-performing logistic regression judgment model reported an accuracy of 69.29%. Table 2 also lists the kappa coefficients for comparing the level of agreement with the actual outcome and facilitating a fine-grained comparison. The kappa coefficients of all approaches were below 0.4, with only the logistic regression and Adaboost obtaining coefficients above 0.3.

We conducted McNemar tests to examine whether the logistic regression model in the second stage significantly improved the first-stage results and outperformed the betting model. This nonparametric test is designed for two related samples and is particularly useful for the before-and-after measurements of the same subjects [66]. Table 3 presents the results. As shown in the table, the logistic regression judgment model significantly differed from the simple judgment model and the first-stage CNN classifier in that both of them rejected the null hypothesis. This proved that by adding the logistic regression judgment model in the second stage, it is significantly different from the one-stage only

CNN prediction model. With the help of judgment in the second stage, the improvement of prediction accuracy is significant. The McNemar value between the logistic regression judgment model and the betting market prediction was 0.468, whereas the p value was 0.494, thereby supporting the null hypothesis that the performances of both models are the same. The logistic regression judgment model outperformed the betting market prediction but did not reach the 0.05 statistical significance level.

Table 3. McNemar test results for the pairwise comparison of the predictive outcome with the logistic regression.

	Betting	CNN Classifier	Simple Judgment Model
McNemar value	0.468	48.188	57.017
p value	0.494	<0.0001	<0.0001

4.5. Discussion

By using CNN to learn the data of all teams, the binary prediction performance was not satisfactory when used to predict the classification of each team in the next match. The second stage of the judgment model, which simultaneously considers both teams in the match, is necessary to improve accuracy. Each team should learn individually with their own data and build their own CNN classifier. A total of 32 CNN classifiers were associated with 32 teams instead of using all data to construct a common CNN classifier. This approach is in line with the concept of some prediction methods in sports, where teams are initially grouped according to their performance and specific models are built for each group [67].

In the second stage, the logistic regression model is significantly more effective than the other machine learning methods probably due to the fact that when only the names and CNN prediction values of the two teams (total of four variables) are used to predict the victory or defeat of the home team, the logistic regression can simply yield different weights and either add or subtract points to adjust the prediction value for each team from the historical data. Other methods may require additional data, such as past performance indicators, and additional variables in the composition of players to be equally effective.

The two-stage approach proposed in this study, incorporating the CNN classifier in the first stage and the logistic regression in the second stage, would yield several benefits. The combination of the two results in a considerable improvement in prediction accuracy over the CNN classifier and logistic regression alone and outperforms the betting market prediction. Another benefit is that the use of match scores and betting odds as raw data is a universal approach to predicting without restrictions on the type of sport. There is no need to think about which performance metrics, such as earned run average and batting average in baseball, or real plus-minus and pace in basketball, to use as features in the model. In contrast to recent two studies on machine learning for NFL game outcome prediction, Hsu [20] similarly used features derived from candlestick charts based on betting odds and scores as input variables to makes one-step ahead predictions. These features include the difference and the proportion of change between the preceding and following match data. A total of 19 and 28 features were selected as the input variables for the classification-based model and the regression-based model, respectively. Both models were examined using five different methods of machine learning. Another study by Beal et al. [68] tested the nine different machine learning classification techniques using a total of 42 features for each team. These 42 features were derived from scores and performance metrics, using the current season average up to that game and the average across the most recent completed season. The total number of features of the two teams in a match was 84, and with considering the home advantage, a total of 85 parameters were used as input values for the prediction model. Although both studies are able to outperform the bookmaker's prediction accuracy, they are accompanied by a number of features that need to be considered as input variables in the prediction model.

However, such time series-based models that we adopt are inevitably challenged as overly simplistic and difficult to represent sports matches. Complementarily, the complicated deep learning algorithms processed in the CNN are used to learn the features of different sports matches and different teams within the time series. The drawbacks of this approach come from the CNN. The network topology and hyperparameter settings of CNNs are not easy to determine with a trial-and-error approach. Moreover, such machine learning methods are more computationally intensive than traditional statistically based methods.

5. Conclusions

CNN and deep learning have made great achievements in the fields of computer vision and natural language processing. However, limited progress has been made in sports analysis and prediction, which mostly rely on visual materials, such as videos of competitions or player position maps. CNN shows difficulties in exploiting its capabilities for 1D data types. Therefore, a predictive model cannot be directly constructed with CNN for sports matches, which are often considered discrete and independent.

To take advantage of the achievements of CNN in computer vision, we proposed a two-stage approach for predicting sports match outcomes. The raw data of the odds and points scored in the betting market were processed through a series of transformations into candlestick expressions comprising time series. Afterward, we used GAF to encode the time series into 2D images in order for them to converge with the CNN of the first stage. The second stage of the judgment model based on logistic regression was then incorporated to consolidate the CNN output for prediction.

Our experimental comparisons reveal that for predicting wins and losses, the historical data imply some associations and behavioral patterns between pre- and post-games that can be used for the prediction. The behavioral patterns implied by individual teams are more useful for the prediction than the behavioral patterns implied by all teams. In addition, predicting the match outcome directly with the CNN of the first stage is not ideal. However, the prediction performance of the CNN was improved significantly by the proposed two-stage approach with the logistic regression judgment model and achieves a prediction accuracy of 69%, which was superior to the betting market.

Future studies are suggested to further investigate the feasibility of using the proposed two-stage model in other sports matches. The hyperparameter tuning problem, which is often encountered when using CNN, can be further tested to improve prediction accuracy. Future research could be directed towards identifying opportunities for incorporating emerging deep learning techniques from traditional sports prediction methods, such as performance metrics and ranking/rating systems, and modifying them to achieve cutting-edge prediction models that keep pace with computational intelligence technology.

Author Contributions: Y.-C.H.: conceptualization, methodology, software, validation, formal analysis, investigation, writing, visualization. No other person is entitled to authorship.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this research are available from the corresponding author upon request.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Haghghat, M.; Rastegari, H.; Nourafza, N. A Review of Data Mining Techniques for Result Prediction in Sports. *Adv. Comput. Sci. Int. J.* **2013**, *2*, 7–12.
2. Sarmiento, H.; Marcelino, R.; Anguera, M.T.; Campaniço, J.; Matos, N.; Leitão, J.C. Match analysis in football: A systematic review. *J. Sports Sci.* **2014**, *32*, 1831–1843. [[CrossRef](#)]

3. Witten, I.H.; Frank, E.; Hall, M.A.; Pal, C.J. *Data Mining: Practical Machine Learning Tools and Techniques*; Morgan Kaufmann: Cambridge, MA, USA, 2016; ISBN 978-0-12-804357-8.
4. Koseler, K.; Stephan, M. Machine Learning Applications in Baseball: A Systematic Literature Review. *Appl. Artif. Intell.* **2017**, *31*, 745–763. [[CrossRef](#)]
5. Beal, R.; Norman, T.J.; Ramchurn, S.D. Artificial intelligence for team sports: A survey. *Knowl. Eng. Rev.* **2019**, *34*, e28. [[CrossRef](#)]
6. Horvat, T.; Job, J. The use of machine learning in sport outcome prediction: A review. *WIREs Data Min. Knowl. Discov.* **2020**, *10*, e1380. [[CrossRef](#)]
7. Wunderlich, F.; Memmert, D. Forecasting the outcomes of sports events: A review. *Eur. J. Sport Sci.* **2020**, 1–14. [[CrossRef](#)] [[PubMed](#)]
8. McHale, I.; Morton, A. A Bradley-Terry type model for forecasting tennis match results. *Int. J. Forecast.* **2011**, *27*, 619–630. [[CrossRef](#)]
9. Baker, R.D.; McHale, I.G. Forecasting exact scores in National Football League games. *Int. J. Forecast.* **2013**, *29*, 122–130. [[CrossRef](#)]
10. Angelini, G.; De Angelis, L. Efficiency of online football betting markets. *Int. J. Forecast.* **2019**, *35*, 712–721. [[CrossRef](#)]
11. Santos, R.M. FIFA World Cup: A Case of (In)efficiency of the Betting Market. *Int. J. Sport Financ.* **2020**, *15*, 110–124. [[CrossRef](#)]
12. Braun, S.; Kvasnicka, M. National Sentiment and Economic Behavior: Evidence From Online Betting on European Football. *J. Sports Econ.* **2013**, *14*, 45–64. [[CrossRef](#)]
13. Strumbelj, E. On determining probability forecasts from betting odds. *Int. J. Forecast.* **2014**, *30*, 934–943. [[CrossRef](#)]
14. Wunderlich, F.; Memmert, D. The Betting Odds Rating System: Using soccer forecasts to forecast soccer. *PLoS ONE* **2018**, *13*, e0198668. [[CrossRef](#)]
15. Wheatcroft, E. A profitable model for predicting the over/under market in football. *Int. J. Forecast.* **2020**, *36*, 916–932. [[CrossRef](#)]
16. Wunderlich, F.; Memmert, D. Are betting returns a useful measure of accuracy in (sports) forecasting? *Int. J. Forecast.* **2020**, *36*, 713–722. [[CrossRef](#)]
17. Li, Y.; Feng, Z.; Feng, L. Using Candlestick Charts to Predict Adolescent Stress Trend on Micro-blog. *Procedia Comput. Sci.* **2015**, *63*, 221–228. [[CrossRef](#)]
18. Xu, R.; Liu, X.; Wan, H.; Pan, X.; Li, J. A Feature Extraction and Classification Method to Forecast the PM2.5 Variation Trend Using Candlestick and Visual Geometry Group Model. *Atmosphere* **2021**, *12*, 570. [[CrossRef](#)]
19. Gerigk, M. Improvements to the STEAM-based teaching of architectural drawing. *World Trans. Eng. Technol. Educ.* **2021**, *19*, 163–168.
20. Hsu, Y.-C. Using Machine Learning and Candlestick Patterns to Predict the Outcomes of American Football Games. *Appl. Sci.* **2020**, *10*, 4484. [[CrossRef](#)]
21. Koopman, S.J.; Lit, R. Forecasting football match results in national league competitions using score-driven time series models. *Int. J. Forecast.* **2019**, *35*, 797–809. [[CrossRef](#)]
22. Rakytyanska, H.; Demchuk, M. Football Predictions based on Time Series with Granular Event Segmentation. In *Proceedings of the Lecture Notes in Computational Intelligence and Decision Making*; Lytvynenko, V., Babichev, S., Wójcik, W., Vynokurova, O., Vyshemyrskaya, S., Radetskaya, S., Eds.; Springer: Cham, Switzerland, 2020; pp. 478–497.
23. Raab, M.; Gula, B.; Gigerenzer, G. The hot hand exists in volleyball and is used for allocation decisions. *J. Exp. Psychol. Appl.* **2012**, *18*, 81. [[CrossRef](#)]
24. Evans, A.E.; Crosby, P. Does a cool head beat a hot hand? Evidence from professional golf. *Econ. Model.* **2021**, *97*, 272–284. [[CrossRef](#)]
25. Robertson, S.; Gupta, R.; McIntosh, S. A method to assess the influence of individual player performance distribution on match outcome in team sports. *J. Sports Sci.* **2016**, *34*, 1893–1900. [[CrossRef](#)]
26. Foysal, M.F.A.; Islam, M.S.; Karim, A.; Neehal, N. Shot-Net: A Convolutional Neural Network for Classifying Different Cricket Shots. In *Proceedings of the Recent Trends in Image Processing and Pattern Recognition*; Santosh, K.C., Hegadi, R.S., Eds.; Springer: Singapore, 2019; pp. 111–120.
27. Cai, J.; Hu, J.; Tang, X.; Hung, T.-Y.; Tan, Y.-P. Deep historical long short-term memory network for action recognition. *Neurocomputing* **2020**, *407*, 428–438. [[CrossRef](#)]
28. Kautz, T.; Groh, B.H.; Hannink, J.; Jensen, U.; Strubberg, H.; Eskofier, B.M. Activity recognition in beach volleyball using a Deep Convolutional Neural Network. *Data Min. Knowl. Disc.* **2017**, *31*, 1678–1705. [[CrossRef](#)]
29. Lin, S.-H.; Chen, M.-Y.; Chiang, H.-S. Forecasting Results of Sport Events Through Deep Learning. In *Proceedings of the 2018 International Conference on Machine Learning and Cybernetics (ICMLC)*, Chengdu, China, 15–18 July 2018; pp. 501–506.
30. Chen, M.-Y.; Chen, T.-H.; Lin, S.-H. Using Convolutional Neural Networks to Forecast Sporting Event Results. In *Deep Learning: Concepts and Architectures, Studies in Computational Intelligence*; Pedrycz, W., Chen, S.-M., Eds.; Springer: Cham, Switzerland, 2020; pp. 269–285, ISBN 978-3-030-31756-0.
31. Hubáček, O.; Šourek, G.; Železný, F. Exploiting sports-betting market using machine learning. *Int. J. Forecast.* **2019**, *35*, 783–796. [[CrossRef](#)]
32. Manivannan, S.; Kausik, M. Convolutional Neural Network and Feature Encoding for Predicting the Outcome of Cricket Matches. In *Proceedings of the 2019 14th Conference on Industrial and Information Systems (ICIIS)*, Kandy, Sri Lanka, 18–20 December 2019; pp. 344–349.

33. Huang, M.-L.; Li, Y.-Z. Use of Machine Learning and Deep Learning to Predict the Outcomes of Major League Baseball Matches. *Appl. Sci.* **2021**, *11*, 4499. [[CrossRef](#)]
34. Kim, T.; Kim, H.Y. Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. *PLoS ONE* **2019**, *14*, e0212320. [[CrossRef](#)]
35. Sezer, O.B.; Ozbayoglu, A.M. Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Appl. Soft Comput.* **2018**, *70*, 525–538. [[CrossRef](#)]
36. Long, J.; Chen, Z.; He, W.; Wu, T.; Ren, J. An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in Chinese stock exchange market. *Appl. Soft Comput.* **2020**, *91*, 106205. [[CrossRef](#)]
37. Grover, A.; Leskovec, J. Node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.
38. Palumbo, E.; Rizzo, G.; Troncy, R.; Baralis, E.; Osella, M.; Ferro, E. Knowledge graph embeddings with node2vec for item recommendation. In Proceedings of the European Semantic Web Conference, Heraklion, Greece, 3–7 June 2018; pp. 117–120.
39. Hatami, N.; Gavet, Y.; Debayle, J. Classification of Time-Series Images Using Deep Convolutional Neural Networks. *Proc. SPIE Int. Soc. Opt. Eng.* **2018**, 10696. [[CrossRef](#)]
40. Wang, Z.; Oates, T. Imaging time-series to improve classification and imputation. In Proceedings of the 24th International Conference on Artificial Intelligence, Buenos Aires, Argentina, 25 July 2015; pp. 3939–3945.
41. Wang, Z.; Oates, T. Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. In Proceedings of the Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 40–46.
42. Yang, C.-L.; Chen, Z.-X.; Yang, C.-Y. Sensor Classification Using Convolutional Neural Network by Encoding Multivariate Time Series as Two-Dimensional Colored Images. *Sensors* **2019**, *20*, 168. [[CrossRef](#)] [[PubMed](#)]
43. Ma, X.; Dai, Z.; He, Z.; Ma, J.; Wang, Y.; Wang, Y. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors* **2017**, *17*, 818. [[CrossRef](#)]
44. Deng, S.; Jia, S.; Chen, J. Exploring spatial-temporal relations via deep convolutional neural networks for traffic flow prediction with incomplete data. *Appl. Soft Comput.* **2019**, *78*, 712–721. [[CrossRef](#)]
45. Asadi, R.; Regan, A.C. A spatio-temporal decomposition based deep neural network for time series forecasting. *Appl. Soft Comput.* **2020**, *87*, 105963. [[CrossRef](#)]
46. Chen, J.-F.; Chen, W.-L.; Huang, C.-P.; Huang, S.-H.; Chen, A.-P. Financial Time-Series Data Analysis Using Deep Convolutional Neural Networks. In Proceedings of the 2016 7th International Conference on Cloud Computing and Big Data (CCBD), Macau, China, 16–18 November 2016; pp. 87–92.
47. Sezer, O.B.; Gudelek, M.U.; Ozbayoglu, A.M. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Appl. Soft Comput.* **2020**, *90*, 106181. [[CrossRef](#)]
48. Levitt, S.D. Why are Gambling Markets Organised so Differently from Financial Markets? *Econ. J.* **2004**, *114*, 223–246. [[CrossRef](#)]
49. Williams, L.V. Information Efficiency in Betting Markets: A Survey. *Bull. Econ. Res.* **1999**, *51*, 1–39. [[CrossRef](#)]
50. Mallios, W.S. *Forecasting in Financial and Sports Gambling Markets: Adaptive Drift Modeling*; John Wiley & Sons: Hoboken, NJ, USA, 2011; ISBN 978-1-118-09953-7.
51. Mallios, W. *Sports Metric Forecasting*; Xlibris Corporation: Bloomington, IN, USA, 2014; ISBN 978-1-4990-4273-3.
52. Mitiche, I.; Morison, G.; Nesbitt, A.; Hughes-Narborough, M.; Stewart, B.; Boreham, P. Imaging Time Series for the Classification of EMI Discharge Sources. *Sensors* **2018**, *18*, 3098. [[CrossRef](#)] [[PubMed](#)]
53. Zhang, G.; Si, Y.; Wang, D.; Yang, W.; Sun, Y. Automated Detection of Myocardial Infarction Using a Gramian Angular Field and Principal Component Analysis Network. *IEEE Access* **2019**, *7*, 171570–171583. [[CrossRef](#)]
54. Lee, H.; Yang, K.; Kim, N.; Ahn, C.R. Detecting excessive load-carrying tasks using a deep learning network with a Gramian Angular Field. *Autom. Constr.* **2020**, *120*, 103390. [[CrossRef](#)]
55. Chen, J.-H.; Tsai, Y.-C. Encoding candlesticks as images for pattern classification using convolutional neural networks. *Financ. Innov.* **2020**, *6*, 26. [[CrossRef](#)]
56. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
57. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980.
58. Lock, D.; Nettleton, D. Using random forests to estimate win probability before each play of an NFL game. *J. Quant. Anal. Sports* **2014**, *10*, 197–205. [[CrossRef](#)]
59. Boulier, B.L.; Stekler, H.O. Predicting the outcomes of National Football League games. *Int. J. Forecast.* **2003**, *19*, 257–270. [[CrossRef](#)]
60. David, J.A.; Pasteur, R.D.; Ahmad, M.S.; Janning, M.C. NFL Prediction using Committees of Artificial Neural Networks. *J. Quant. Anal. Sports* **2011**, *7*. [[CrossRef](#)]
61. Balreira, E.C.; Miceli, B.K.; Tegtmeyer, T. An Oracle method to predict NFL games. *J. Quant. Anal. Sports* **2014**, *10*, 183–196. [[CrossRef](#)]
62. Prasetio, D.; Harlili, D. Predicting football match results with logistic regression. In Proceedings of the 2016 International Conference on Advanced Informatics: Concepts Theory and Application (ICAICTA), Penang, Malaysia, 16–19 August 2016; pp. 1–5.

63. Kolbush, J.; Sokol, J. A logistic regression/Markov chain model for American college football. *Int. J. Comput. Sci. Sport* **2017**, *16*, 185–196. [[CrossRef](#)]
64. Li, Y.; Wang, L.; Li, F. A data-driven prediction approach for sports team performance and its application to National Basketball Association. *Omega* **2021**, *98*, 102123. [[CrossRef](#)]
65. Song, C.; Boulier, B.L.; Stekler, H.O. Measuring consensus in binary forecasts: NFL game predictions. *Int. J. Forecast.* **2009**, *25*, 182–191. [[CrossRef](#)]
66. Kim, K. Financial time series forecasting using support vector machines. *Neurocomputing* **2003**, *55*, 307–319. [[CrossRef](#)]
67. Bilek, G.; Ulas, E. Predicting match outcome according to the quality of opponent in the English premier league using situational variables and team performance indicators. *Int. J. Perform. Anal. Sport* **2019**, *19*, 930–941. [[CrossRef](#)]
68. Beal, R.J.; Norman, T.; Ramchurn, S. A critical comparison of machine learning classifiers to predict match outcomes in the NFL. *Int. J. Comput. Sci. Sport* **2020**, *19*. [[CrossRef](#)]