UtilitasMathematica

ISSN 0315-3681 Volume 122, 2025

Integrating Deep Learning and Sentiment Analysis for English Premier League Match Forecasting

Omkar Birmole
Computer Engineering
Vidyalankar Institute of Technology
Mumbai, India
omkar.birmole@vit.edu.in

Dr. Amit Nerurkar
Computer Engineering
Vidyalankar Institute of Technology
Mumbai, India
amit.nerurkar@vit.edu.in

Dr. Ravindra Sangle
Computer Engineering
Vidyalankar Institute of Technology
Mumbai, India
ravindra.sangle@vit.edu.in

Dr. Sanjeev Dwivedi
Computer Engineering
Vidyalankar Institute of Technology
Mumbai, India
sanjeev.dwivedi@vit.edu.in

Abstract*— This research presents an ensemble framework that predicts the outcomes of English Premier League matches by integrating historical match statistics with Twitter sentiment analysis. Employing the CRISP-DM methodology, the study combines a Long Short-Term Memory (LSTM) model trained on twenty seasons of match data with sentiment insights derived from tweets collected a week before each game. Feature engineering and dimensionality reduction techniques were applied to enhance model efficiency and address multicollinearity issues. The LSTM model achieved a prediction accuracy of 70%, outperforming other machine learning algorithms like Decision Trees, Random Forests, and SVMs. Sentiment analysis of over 10,000 tweets per week provided additional predictive power. By ensembling the outputs of the LSTM and sentiment models using a weighted average approach (70:30 ratio), the system consistently predicted 7–8 correct match outcomes per week. The results demonstrate that integrating social media signals with historical data significantly improves predictive accuracy, offering a robust approach for forecasting football match results.

***Keywords*—LSTM, KDD, ANN, RNN,**

I. INTRODUCTION

Football is the most widely played and watched sports in the world. Thus the most extensively studied domain in the football is the prediction of football matches. Therefore, a number of scientists have been researching and developing a model to predict match winner and match score outcomes. There are various ways to predict the result by considering several necessary attributes. The most favourite parameters to be considered while forecasting results are previous match statistics, Team form, Home – Away conditions, Player's performance, Yellow – Red cards, Player Injuries etc. Thus, in this research study, we have created an ensembled model by considering some of these parameters and predicted the outcome of English Premier League weekly matches. This ensemble model consists of combining two separate methods which are the Recurrent Neural Network (LSTM model) and the sentiment analysis of Tweets.

Consider the scenario for prediction of the football match, if Team A loses the match against Team B today and if they are scheduled to play again in after few weeks, a model need to remember this defeat because it occurred in a different time frame. Recurrent Neural Networks are widely used in such scenarios. A Recurrent Neural Network can combine different layers of an ANN with the help of a recurrence function. Long Short-Term Memory (LSTM) is a form of artificial recurrent neural network (RNN) and is used in deep learning. LSTM has feedback connections as opposed to traditional feedforward neural networks. When new information confronts RNNs, they completely transform the existing information by applying a function. This information in LSTM flows through a system known as cell states. LSTM also has a special operation that is a forget gate through which the unnecessary information can be omitted. LSTM predictive model is used in various fields such as time series prediction, speech recognition, rhythm learning, handwriting recognition, sign language translation. Therefore, we have used this property of the LSTM model in our study to consider which information is necessary or what kind of data is inadequate. Using LSTM model, we are able to achieve the model accuracy of 70%.

Twitter is one of the leading sources of predictive modelling in various fields such as stock market, manufacturing, sales prediction , disease spread prediction or sporting outcomes. Until now, such an analysis has not been carried out in football. This work aimed to analyse whether data collected from Twitter can be used for this purpose. For the last 7 days before the week of the game, we have considered twitter data and combined twitter prediction with historical data and basic football statistics. In addition, it builds hybrid models using both Twitter and historical data. The final results show that data mined from Twitter can still be a valuable tool for forecasting Premier League games. Thus, This study provides the proof that features derived from Twitter is useful to predict football outcomes.

This research study is completely based on the results and performance of teams which are playing in English Premier League. CRISP – DM methodology is used to build this framework. After constructing a framework, multiple case

studies are performed on the model by predicting weekly outcomes of season matches for the EPL 2019-20. Out of the 10 weekly matches, every time we have correctly predicted the outcomes of 7 – 8 matches. In the below sections, we have demonstrated how the system is developed and used in the prediction of EPL weekly matches.

II. Literature Review

In the recent years the extensive research has happened in the field of match's result and eventually betting odds of the match. Various scientist has been performing statistical analysis on the historical data or previous match results and implement an algorithm which will predict the winner of the match or match score before the match ends. Some of these research papers are studied and surveyed most of the statistical attributes to perform analysis and create a model in this paper.

Pettersson and Nyquist, used LSTM model of Recurrent Neural Network (RNN) to predict football match results. Their main objective as to check whether the RNN can be used for the match prediction as well as is the history of players is important factor and how long it required. In this paper, the results were examined by considering Playing line-ups, Positions, Goal, Card, Substitution, and Penalty as an input parameter. Their dataset includes league matches from 63 different 3 tournaments having a total of 35234 matches of season 2015 – 16 and 2016 – 17. The neural network is constructed with the help of TensorFlow libraries. They have developed 7 case studies with two approaches that is many-to-one and many-to-many. First 6 case studies are based on the many-to-many approach where an accuracy is calculated for all the events in the sequence whereas the last case study is about the many-to-one approach where the accuracy is calculated at the end of the sequence. The LSTM model predicted the accuracy at every 15 minutes of all matches. The accuracy at the start of the game of many-to-many model is 43.96 % which increases up to 88.68 % at the end of the game. Similarly, the accuracy of many-to-one model is 33.35 % at the beginning of the game while it increases up to 98.63% when the game finishes. They have concluded that the LSTM network works well in classification and prediction problems.[3]

In another paper, Joseph, Fenton, and Neil predicted the result by Expert Bayesian Network and compared its accuracy with other machine learning techniques such as Naïve Bayesian Network, K – Nearest Neighbour, Data-Driven BN, MC4 Decision Tree. The dataset consists of total 76 matches of only one team for season 1995/96 and 1996/97. Their analysis is based on few factors like presence of 3 marquee players, playing position of Wilson, the quality of opponent team and venue of the match. Apart from this, they have classified three more attributes which defines the attacking play of a team, team's overall quality of playing and performance compared to opponent's team. They have drawn the relative accuracy of these algorithms using general and expert model data. They have concluded that the expert BN learner has the highest accuracy of 59.21% compared to other learners in both the model. But there is a limitation that this analysis is based on the statistics of only one team having limited matches.[4]

A study titled An Improved Prediction System for Football Match Results by Igiri Chinwe Peace and Nwachukwu Enoch Okechukwu from the University of Port Harcourt, Nigeria, utilized Knowledge Discovery in Databases (KDD) to predict football match outcomes. The researchers employed Artificial Neural Network (ANN) and Logistic Regression (LR) models to analyse factors influencing match results. Their work primarily examined how external cup competitions affect league performance, the impact of injuries to key players on team outcomes, and the influence of home advantage. The study considered variables such as the number of goals scored, the moving average of team performance during a season, performance indices for players and managers, and additional features from existing systems. The dataset comprised 110 matches from the 2014–2015 English Premier League season. They have used input parameters like Home and Away goals (GHA), Home and Away shorts (HAS), Home and Away corner (HAC), Home and Away Odds (HAOD), Home and Away attack strength (HAAT),Home and Away Players' performance index(HAPPI), Home and Away Managers' performance index (HAMPI), Home and Away managers win (HAMW), Home and Away streak (HASTK).The output parameters are Result of match (Home Win, Draw, Away Win). In this paper they have compared Statistical model with Data mining techniques. They have concluded higher accuracy with LR, yielded 85% and 93% prediction accuracy for ANN and LR techniques respectively.[5]

Stylianos Kampakis used Twitter to predict football outcomes. They have built a set of predictive models based on tweets of the English Premier League of few months period and studied whether these models can overcome predictive models which use only simple football statistics and historical data. In this paper they have used model based on Twitter and historical data. In final they have concluded that tweeter data combined with historical match data is useful source for prediction of Premier League matches. They observed that Twitter-based model performs considerably better than simple statistical models. They have collected data of matches of last 3 months of 2014 consisting of the twitter dataset, the historical and simple statistics dataset. For Tweeter data they have used Twitter's open Streaming API for collecting tweets of over 2 million fans about their favourite teams. Input for the model is Home team features, Away team features and Response variable. Model used for predictions are Logistic regression, SVM, Naïve Bayes and Random forests. They have observed that for twitter- Only model with Random forest as a classifier got accuracy between 56.3% and 74.7%. For Historical-only model with Naïve Bayes as a classifier got accuracy between 50.6% and 64.4% but when they combined twitter + historical data with Random forest as a classifier they got accuracy between 64.4% and 74.7%.[6]

The paper Predicting Wins and Spread in the Premier League Using a Sentiment Analysis of Twitter by Robert P. Schumaker, A. Tomasz Jarmoszko, and Chester S. Labedz Jr. explored the use of sentiment analysis to predict match outcomes in the English Premier League. The study analyzed tweets from fans of the league's twenty clubs, focusing on their sentiment content. The authors addressed key questions such as: What signals in Twitter data can aid in predicting match results? How does the magnitude of sentiment affect

prediction accuracy? And what influence does sentiment-based prediction have on wagering returns?[13]

The study of Bunker and Thabtah is focused on the results of historical matches, player performance indicators, and opposition information. They have used ANN for prediction of match results. They proposed SRP-CRISP-DM framework for sports match predictions which is based on the six steps of the standard CRISP-DM framework. Also, their analysis confirmed that the machine learning seems to be an appropriate methodology for sports prediction due to high volume on betting on sport. [8]

In another paper, Sushant and Deepanshu analyzed the football dataset using various machine learning algorithms and compare their accuracies. They have used CRISP-DM framework for exploratory and spiral analysis. The analysis and comparison are based on the Logistic Regression, Extra Gradient Boosting and Support Vector Machine algorithms. Their dataset includes a total of 4560 English Premier League matches with 12 seasons. The input parameter consists of Home/Away Team Goals, Home/Away Team Cumulative Goals Scored and Conceded, Last 5 matches Home/Away Team Performance, Home/Away Team Win and Lose 3 and 5 matches streak, Home and Away Team Goal Difference, Difference in points last year standings. They have concluded that the Logistic Regression is the best suitable model for their dataset with the accuracy of 60% whereas XGBoost and SVM models got the accuracy of 54% and 56% respectively.[13]

In 2018, Pablo Bosch did comparison between Machine Learning and Deep Neural Network to predict the winner of American Football League. He has selected Logistic Regression, SVM and Random Forest models from Machine Learning techniques whereas he has chosen Artificial Neural Network, LSTM and RNN from Deep Learning models. The data obtained for this included a total of 2048 matches from season between 2009 to 2016 with 13 different datasets. He trained the data with different batch sizes and structure with respect to the required model. He has concluded that with his experimental setup, machine learning model got better accuracy than Deep Learning models. Additionally, he found that LSTM works better among all other Deep Learning model with the selected data.[9]

III. Research Methods

The next phase of CRISP – DM methodology is the data understanding and the data preprocessing. This section therefore describes how the data are obtained, what pre-processing is performed on the data and what kind of analysis methods are used in this research.

A. Data Collection

Data collection and understanding the data is very important since the entire study is based on the data form. The collected data could be either qualitative or quantitative in nature. Quantitative refers to the numbers or values in the field whereas quantitative is nothing but the text form or words. In this study, two major datasets are considered, and they are historical dataset on which LSTM model is constructed and tweets on which the sentiment analysis is performed.

The historical dataset used in this study is obtained from football.co.uk website. The historical dataset consists of last 20 years of data that is from 2000-01 season up to the current season, 2019-20. The extracted dataset was available in csv file format. Each season has individual csv file which included a total of 380 league matches. These csv file contains various factors such as Date of match, Playing teams, Match Result, Teams Total Shots, Teams Shots on Target, Teams Fouls, Teams Cards, Betting odds of different websites etc. Overall, it contains the statistics and betting odds about game, but our study is completely focused on the statistics of the game.

Another dataset is based on the tweets which are tweeted by people and this incorporates people's opinion about different teams. This Twitter dataset consisted of about two million fan tweets about each of their favourite teams and was collected through Twitter's open streaming API. We have been considering twitter data set in this research for the last 7 days before every week of the season. The tweets are fetched based on the different hashtags. A list of hashtags is built that are closely connected to each of the 20 Premier League teams. The below table number 1 contains a list of few hashtags which are used to retrieve tweets from the twitter:

#EPL20	#Watford	#Spurs	#goVillans
#PL	#Norwich	#Shefield	#bhafc
#mufc	#Arsenal	#goEverton	#WestHam
#Palace	#Liverpool	#Burnley	#Leicester
#Chelsea	#ManCity	#Wolves	#nufc

Table 1: List of few hashtags used to extract Tweets

For the most part, the compilation of this list was based on many online resources that describe the teams' official hashtags, and any nicknames that a team may have such as '#PremierLeague,'#PL,'#ManUnited,'#Palace,'#Chelsea,'#Norwich,'#Arsenal,'#Liverpool' etc. With respect to these hashtags, we considered around 15,000 tweets.

We came across a set of challenges while generating both the datasets. In case of historical dataset, although the data was available in the csv file format for all the seasons but the number of attributes in each season was varying. Especially, the older seasons had a very less set of attributes compared to the latest seasons. To overcome this challenge, we considered those set of attributes which are common in all seasons as well as very important with respect to the model building. Along with this, we filled couple of missing data cells by taking reference from the internet. In case of twitter dataset, we were restricted for the tweets of last 7 days. The twitter API is not allowed to capture tweets more than 14 days. Additionally, some of the tweets were about the team news due to Covid – 19 pandemics. This issue is resolved by not considering these tweets for the final analysis.

B. Exploratory Analysis:

The exploratory analysis begins by analyzing the trend of match result. The result of the match could be either winning of home team, away team or draw. By analyzing the dataset of past 19 years season, it is identified that the possibility of home team's win is more than away team's win or draw. The

below pie chart (Figure 1) describes the percentages of home team win, away team win and draw result.

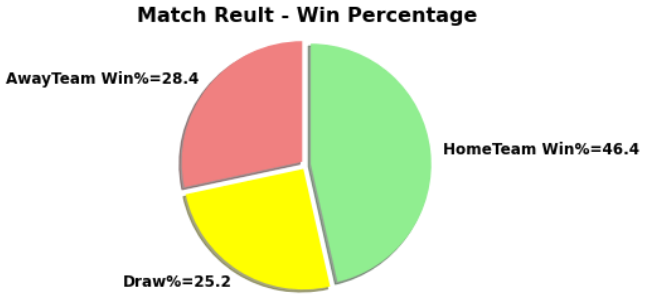

Figure 1: Football Match Results in percentages

Secondly, we focused on the correlation between the total shots on target and the number of goals scored. It's quite clear that the game is won by the team that scores more goals. But sometimes it is possible that after playing shots on target, the teams are unable to score a goal due to opponent's strong defense which indirectly links to the result of the game. This relation is studied by plotting a graph of total shots on target against the number of goals scored.

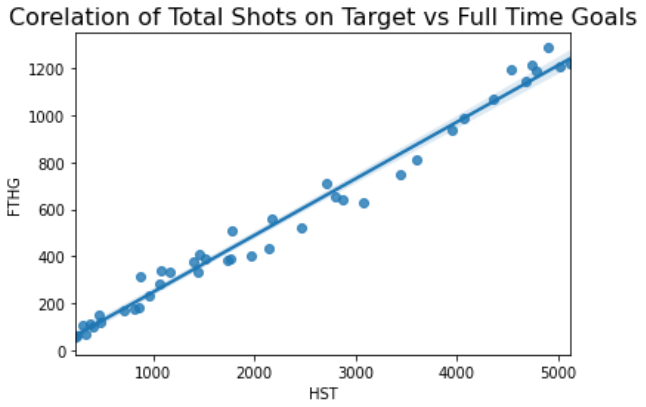

Figure 2: The correlation between Total Shots on Target and Full Time Goals scored

The figure number 2 explains that the there is a strong relationship between the shots on target and the total goals. Consequently, this also meant that the match would win by scoring more shots on target.

The next part of exploratory analysis is the Dimensionality Reduction process. Dimensionality Reduction is the method in which only required features are selected from a set of features and generating a new set of features from the selected attributes. Thus, it directly reduces the large number of columns and brings down to a considerable set of attributes. It is very important to perform a dimensionality reduction operation otherwise it may cause many issues while training the model. Dimensionality Reduction has following list of advantages:

1. Less number of chances for overfitting the model.
2. Less dimensions increases the computation efficiency and reduces the computing time.
3. Less amount of data requires minimal storage space.
4. Eliminates duplicate set of values or noise.

The following set of features which are mainly statistical data are extracted from a total of around 60 columns after the dimensionality reduction operation:

```
['Date', 'Home Team', 'Away Team', 'Full Time Home Goals', 'Full Time Away Goals', 'Full Time Results', 'Home Team Shots', 'Away Team Shots', 'Home Team Shots on Target', 'Away Team Shots on Target']
```

After extracting features, the next important step is to preprocess a dataset to generate more information about data. Data Pre-processing is the stage in which the data is analysed, or Encoded, to get it to such a state that the machine can easily parse it. In other words, it is nothing but converting a raw data into a clean dataset. In this stage, different operations are performed on the dataset and new information is generated. The newly generated features are defined below:

1. Previous Match Results:

In football game, team's performance is very crucial factor in terms of analyzing and predicting the results. The previous match form is very effective and advantageous over the other team. Hence, we have generated a new attribute which will calculate team's form based on the results of their 5 previous matches. It's value ranges from 0 to 1 scale where 1 denotes all previous wins while 0 denotes no win in all previous matches.

2. Shots Accuracy:

In every match, each team played a shot towards a goal such that some of them could be on target while some of them could be off target. Certainly, the shots on target creates a chance of goal and boosts the confidence of team. Thus, we have calculated the accuracy of the shots by creating a below formula.

$$Shot\ Accuracy = \frac{Shots\ on\ Target}{Total\ Shots}$$

The value of this attribute is also ranging from 0 to 1 here 0 means no shot on target while 1 means al shots on target.

3. Home Team Advantage:

In every sports match, the home team always geta an additional advantage over the opponent. Because of the favorable conditions, ground awareness, crowd support etc. In our analysis, we have considered Home Team advantage is one of the primary and important factors. The fig. 1 which shows that the chances of home team win is higher than other two results which is the evidence to support this analysis. Therefore, we have classified the result of the match as home team win or home team does not win.

4. Team Ranking:

Team Ranking expresses where does team ranks among other playing clubs. The top-ranking teams always has a great performance to low ranking teams in EPL. Therefore, we have obtained the previous season's rankings of both Home and Away team.

5. Total Goal Differences:

In each season, the cumulative Goal Scored and Goal Conceded of each team are computed to calculate the goal difference between the goal scored and goal conceded. This describes that the positive goal difference leads towards the strongest team than a negative goal difference.

When the heatmap is generated with all newly generated features, it is identified that the dataset could cause the problem of multicollinearity. Multicollinearity basically occurs when two or more independent variables are highly correlated with each other. Thus, by changing the value of one independent variable, the value of another independent variable can also alter. This will create a highly volatile and unstable model. In our dataset, the multicollinearity mainly occurs because the newly generated variables are highly dependent on other variables. Also, they included some sort of identical data. There are multiple tests to detect multicollinearity in a dataset, but we performed Variance Inflation Factors (VIF) test to identify collinear columns from the dataset. The VIF factor is calculated by the degree of association between the independent variables. Basically, it takes one variable and compares it with some other variable. The VIF expression is denoted by following formula:

$$VIF = \frac{1}{1 - R^2}$$

Therefore, the closer the R^2 value to 1, the higher the VIF value and the greater the multicollinearity with the feature variable in question. The VIF value of column which exceeds than 10 will cause the multicollinearity issue in the model. However, in case of weaker models, the value greater than 2.5 will create a multicollinearity issue.[15]

In VIF test, it is observed that there is a conflict between few columns such that there VIF values are greater than 10. Following are the set of attributes causing multicollinearity issue which are then removed:

```
['Home Team Shots on Target', 'Away Team Shots on Target', 'Home Team Goal Conceded', 'Away Team Goal Conceded', 'Home Team Goal Scored', 'Away Team Goal Scored']
```

After all tests, the final set of attributes are obtained from the created dataset. The heatmap generated for the final set of attributes is attached below[16] (Figure 3):

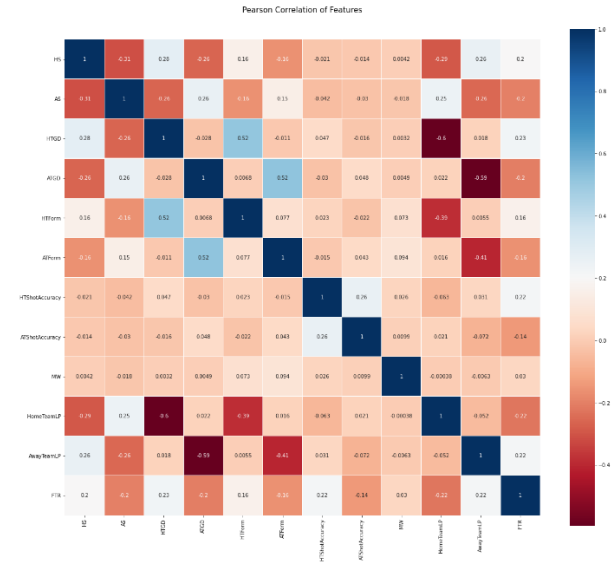

Figure 3: Heatmap of Correlation Matrix

The following table number 2 contains a final list of attributes on which the model is trained.

Attribute	Data Type	Description
HS	Ratio	Home Team total shots
AS	Ratio	Away Team total shots
HTForm	Ratio	Home Team Form
ATForm	Ratio	Away Team Form
HTGD	Ratio	Home Team Goal Difference
ATGD	Ratio	Away Team Goal Difference
HTShotAccuracy	Ratio	Home Team Shot Accuracy
ATShotAccuracy	Ratio	Away Team Shot Accuracy
MW	Ratio	Match Week
HomeTeamLP	Ratio	Home Team Last Year Ranking
AwayTeamLP	Ratio	Away Team Last Year Ranking
FTR	Ordinal	Full Time Results

Table 2: Final List of Features

C. Classification:

Classification is a problem in machine learning that learns how to allocate a class mark to examples from the specific problem. There are various types of classification problem

occurs in machine learning based on the label in target column. In our problem, the target variable has nominal values that is Home Win, Away Win and Draw, hence it is classified into a Multi Label Classification problem. We have converted it into a binary classification problem (Home Team Win, Home Team Not Win) since, our research is based on weather home team will win or lose?[12]

This problem has been addressed using a variety of machine learning algorithms and artificial neural network techniques. Specifically, Decision Tree Classification, Random Forest Classification, SVM Classification, and Extreme Gradient Boosting Classification were selected from machine learning methods, while Multi-Layer Perceptron and Long Short-Term Memory (LSTM) from the Neural Network family were employed. Among these, the primary emphasis is on the Long Short-Term Memory (LSTM) algorithm, a type of Recurrent Neural Network. These algorithms have been widely utilized in prior research, which is why they were chosen for training the model in this study.[1]

We have faced couple of challenges in selecting the primary algorithm for research and analysis of the problem. In previous study, some researcher performed experiments using MLP technique while few of them opted to go ahead with LSTM model. Moreover, they have obtained considerably a better result with both these models. Therefore, we have performed a Proof of Construct (POC) on both these models (MLP and LSTM) by considering a small set of data that is data sampling. We obtained a better result for LSTM model, hence we moved forward to study more about LSTM.[16]

1) Decision Tree Classification:

Decision Tree is used for the structured classification and regression model. This is incremental development as dataset is divided into minor subsets and at the same time decision tree starts constructing incrementally. It gives the end result as a leaf node which is termed as a decision node. A decision node such as Weather has two or more branches such as Sunny, Overcast and Rainy, each shows the value of the attribute. Leaf node such as Hours Played represents numerical value of the decision.

The predictor of the tree is the topmost node called as root node. The solution of our problem emerges as the leaves of tree where each node represents a point of decision. In our problem, full time result is the target element and the Decision Trees are used to identify the result based on other various statistical attributes.[15]

2) Random Forest Classification:

Random decision forest is a classified as a Machine Learning method for data analysis and prediction. Random forest is considered as a Bagging methodology because the creation and execution of tree works in simultaneously. At training time, decision is made by constructing multitude of decisions trees and output is in the form of class. Hyperparameter is the main feature of random forest architecture. In each node, some limited percentage of the total number can be divided into further node. This algorithm selects the k – data points from the training dataset and builds the decision tree associated with these k data points. The above step gets repeated with the number of trees equal to the n_estimators parameter given during the model fitting. It generates new data point at each iteration and predicts the value of target variable for that data point. Finally, all of the predicted target values are assigned by the average of new data point.[14]

3) Extreme Gradient Boosting Classification:

Extreme Gradient Boosting (XGBoost) is a powerful machine learning library in Python that supports regression, classification, and ranking tasks. It enhances the Gradient Boosting model by improving efficiency and performance. By utilizing pseudo-residuals, the algorithm minimizes the loss function iteratively at each step.

The XGBClassifier is specifically designed for classification problems. Written originally in C++, XGBoost is significantly faster than many other ensemble models. Its ability to parallelize computations makes it highly suitable for large datasets. Additionally, it provides built-in parameters for cross-validation, regularization, handling missing values, and configuring tree structures, making it a versatile tool for complex modeling tasks.

4) SVM Classification:

In most of the problem, SVM has obtained higher accuracy than other classifiers such as Decision Tree, Logistic Regression model etc. Basically, SVM separates two different classes to each other by constructing a hyperplane. Here, the task of the classifier is to find the maximum margin that separates two classes. The plotted points which are the closest to the hyperplane are termed as a Support Vectors. A Kernel is used to implement SVM algorithm and this technique is called as a Kernel Trick. A kernel is available in different types. In our study, we have used a linear kernel to train the model.

5) LSTM:

Long Short-Term Memory is classified as a Recurrent Neural Network model. In Recurrent Neural Network (RNN), the output of the present time step is forwarded to the input of the next time step. Therefore, the model not only considers the current input at each element of the series but also identifies the preceding elements. Its memory helps to construct a long-term dependency such that when it considers whole context into account while predicting, it can be a next word of sequence, a sentiment classification or next measurement of temperature. But, RNN suffers a problem of short-term memory. That means if sequence is long enough, RNN fails in carrying information from current time step to the next time step. Thus, it is possible that RNN may left out an important information while processing a paragraph of text. [17]

LSTM solves this problem by including a memory cell in the model that can sustain the information for the longer time. LSTM processes the information when it propagates forward. The operations performed by LSTM model allows it keep or

forget the information when required. The architecture of LSTM model is described below figure 4:

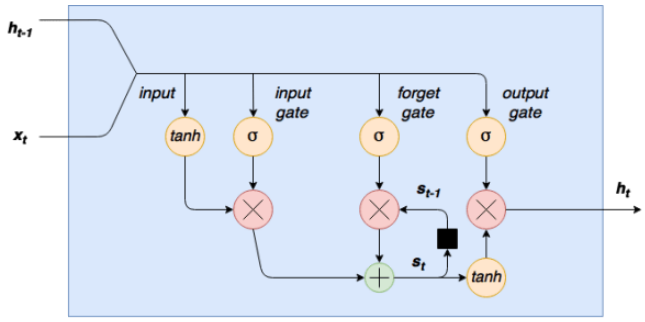

Figure 4: LSTM Cell Diagram

The LSTM is built on the cell state and its different gates. The responsibility of cell state is to transfer the relative information throughout the sequence chain. The gates act as a different neural network which determines which information is relevant and needs to keep forwards or forget.

a) Input Gate:

The input gate consists of the previous hidden state and current input data which is then passed through the sigmoid function. Sigmoid function determines which values are get updated by transforming values between 0 to 1 where 0 defines as not important whereas 1 denotes important to update.[11]

Also, the hidden input state and the current state can be initially passed through the tanh function which squashes the data between -1 to 1 to regulate the network. After this, we can multiply its output with the sigmoid function. Consider the below equations of input gate:

$$i = \sigma(b^i + x_t U^i + h_{t-1} V^i) \quad \dots \text{(i)}$$

$$g = tanh(b^g + x_t U^g + h_{t-1} V^g) \quad \dots \text{(ii)}$$

In the above expressions, U and V are the weights for input and previous cell output respectively. X_t and h_{t-1} are the input values. The input bias is defined by b.

In the first equation, the sigmoid function squashes the expression into the range of 0 to 1 whereas in the equation 2, the tanh function compresses the output between -1 to 1. The output of the input cell is expressed by o operator which denotes the element wise multiplication.[11]

$$g \circ i$$

b) Forget Gate:

Forget gate is the most important gate in the model as it decides which infromation should be kept or thrown away. The information carried out from the previous hidden state and the current input is processed through the sigmoid function. The sigmoid function squeezes output in the range of 0 to 1 where 0 denotes to forget while closer to 1 means to keep. [15]

In our dataset, the forget gate is used to keep the information about teams's performance in last 5 matches. The forget gate is defined by an expression given below:

$$f = \sigma(b^f + x_t U^f + h_{t-1} V^f)$$

Similar to input gate, U and V acts as the weights for input and previous cell output respectively. The input bias will be defined by b and x_t & h_{t-1} are the input values. Further, this forget acts as weightage for the internal state. The output of this state is defined by s_t and expressed by below equation:

$$s_t = s_{t-1} \circ f + g \circ i$$

c) Output Gate:

The output gate is the final state of LSTM model. It provides the information of next hidden state and the next cell state. The output of input gate that is sigmoid function is multiplied with the tanh function which is the output of cell state to produce the hidden state. Thus, basically the output gate describes that what will be the next hidden state. The output gate is defined by given expression[15]:

$$o = \sigma(b^o + x_t U^o + h_{t-1} V^o)$$

Also, the final output of the cell is calculated by multiplying the output of output gate with the output of internal state undergoing through tanh function.

$$h_t = tanh(s_t) \circ o$$

d) Cell State:

The Cell state is calculated with the help of the information available at input gate and the forget gate. Initially, the forget vector multiplies cell state by point wise. After this, it sums up the output of the input state and updates a new value for cell state point wise.

6) Sentiment Analysis:

Sentiment Analysis is the process of identifying the author's opinion about the topic and classifying it as a positive, negative or a neutral tone. The author's opinion is nothing but the attitude or the emotions of the speaker towards a particular subject.[19] Hence, it is also called as an Opinion Mining or the polarity of the content. Today, the sentimental analysis is widely used in the field of business, politics and public opinion sector such as analyzing citizens mind before the voting process etc.

Sentiment Analysis is performed on text data by using natural language processing. It processes important words in the text sentences in terms of "Bag of Words" and classifies that text sentence. Twitter is one of the sources through which the tweets can be fetched and performed a sentiment analysis on the text to determine the tone of the subject. In our project, we have used twitter as an input source to collect tweets related to EPL football matches. A Tweepy library is used to gather tweets before 7 days of the playing week.[20]

IV. Evaluation And Results

Once the primary analysis is performed on the dataset and the methods are selected, the next phase of CRISP – DM methodology is modelling and evaluation. In this section, we have illustrated the system architecture and findings or observations which we have obtained from our experiments.

A. System Architecture:

The architecture of the system is very crucial in terms of modelling and executing the experiments. We have designed our prediction system after multiple trail – error basis and iterative approaches. The entire modelling is based on this architectural design.

The given below is the architectural figure of the model (Figure 5:

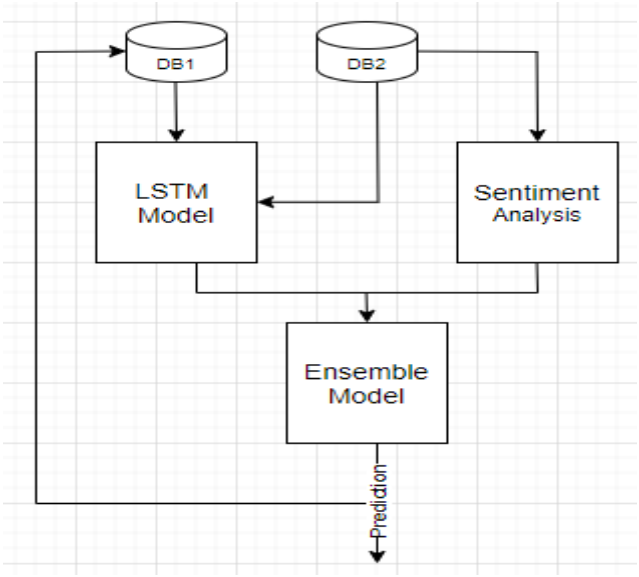

Figure 5: The Prediction model Framework

As described in the figure number 5, the system is divided into 3 primary components that is LSTM model, Sentiment analysis (Tweets) and Ensembled Model. The DB1 dataset contains the data of 19 seasons that means the dataset from 2000-01 season up to current week results. This dataset is further divided into training and validation sets in the LSTM model. The LSTM model gives the possibility of winning chances for both Home Team and Away team.

On the other side, the DB2 dataset refers to the next week matches (10 match per week) and this dataset is used for the testing purpose in the LSTM model. Furthermore, the tweets captured for the next week's matches (DB2 dataset) are undergoing the model of Sentiment analysis. Similar to LSTM model, this model also provides the output as the winning possibilities of both the teams.

Finally, we have ensembled model in the ratio of 70:30 such that 70 % of prediction values considered from LSTM model while Sentiment analysis partnered as a 30 % in the total prediction. Finally, the actual results of next week's game are added into the main historical dataset and new next week's matches have replaced them in DB2 dataset and the cycle repeats for every week.

B. Results:

The evaluation process commences with constructing and training the model. As explained above, out of a total of 19 seasons, we have divided first 15 seasons as a training dataset and remaining 4 seasons as a validation set. Furthermore, next week's 10 matches are considered into a testing dataset. The results obtained from each component are noted below:

LSTM:

In our dataset, we have a total of 11 features extracted and created from the main dataset. Also, our target attribute is the result of the match which denotes whether the home team win, or home team does not win. Thus, it is a binary classification problem where the target column has two values 0 and 1. The LSTM model is composed of a three-layer block with an input layer, an output layer and a hidden layer. Since, it is a binary classification problem, a softmax activation function is used whereas binary crossentropy is used as a loss function. Initially, the model is trained at different epoch size, but the better accuracy is obtained at the 60-epoch size. The below table number 3 describes the accuracy of LSTM model and comparison with another machine learning algorithm:

Classification Algorithm	Model Accuracy
Decision Tree	0.6173
SVM	0.7093
Random Forest	0.6794
Extreme Gradient Boosting	0.6998
LSTM	0.7001

Table 3: Comparison of Different Machine Learning Algorithms

After training and fitting a dataset with LSTM model, the model has achieved a 70 percent accuracy. Compared with other classification algorithms for machine learning, the LSTM model has worked better. The F – measure of the model is 0.70 which describes that combination of precision and recall. The ROC curve of the model is shown below (Figure 6):

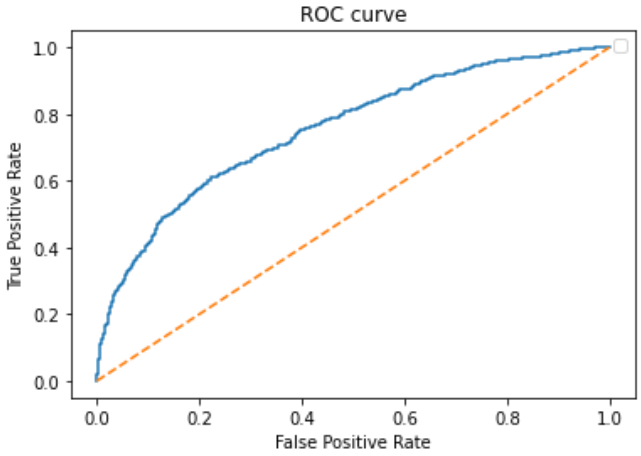

Figure 6: ROC curve of LSTM model

Ensemble Model:

A Machine Learning Ensemble model combines the decision from multiple models to boost overall efficiency. An

ensemble model assists in reducing the error factors from the learning model, including noise, bias and variance. Thus, it gives the model stability and the robustness to the model overall. After the sentiment analysis of tweets, we have constructed an ensemble model which gives the final prediction of matches. The model is built by using the Weighted Averaging technique. It contains 70 % weights of predictions obtained from LSTM model while 30 % weights came from the predictions using sentiment analysis. Finally, this model draws the winning possibility chances of home team and the away team.

Consider below case study of 3rd last week of EPL 2019-20 season. There are 10 matches played in each week, where each team has a match with another club. For a specific club, it may be Home game or Away game. A total of 10000 tweets are extracted from the twitter based on different hashtags and a sentimental analysis is performed on them. This analysis provides the wining chances of both teams. A dataset is created with this up to the game of the previous week and trained with the LSTM model. The predictions are obtained from the LSTM model after it has been checked for the game of the coming week. Finally, both the models are ensembled in the ratio of 70:30. Following are the findings of a few matches of this case study (Table 4):

Date	16-07-2020	15-07-2020
Home Team (HT)	Arsenal	Newcastle
Away Team (AT)	Liverpool	Tottenham
LSTM HT Possibility	0.5768	0.3213
LSTM AT Possibility	0.4231	0.6786
Twitter HT Possibility	0.3768	0.4420
Twitter AT Possibility	0.6231	0.5579
Ensemble HT Possibility	0.5168	0.3575
Ensemble AT Possibility	0.4831	0.6424
Actual Match Score	2 – 1	1 – 3
Actual Team Win	Arsenal	Tottenham

Table 4: Result of Case Study (Few Matches)

Thus, out of 10 matches the result of 7 matches was calculated accurate using this system. One notable outcome came from the Arsenal vs. Liverpool match where Liverpool was the number one team in the table, but it was expected they would lose the match and in fact they lost the match by 1 –2.

After the actual structure was completely developed, we had conducted these case studies several times at the start of each week's game and analyzed the potential winning team based on their winning chances. In each case study, 7 or 8 out of 10 results correctly predicted.

V. Conclusion

In this research study, we analyzed the trends in the English Premier League previous seasons and created an ensembled framework by integrating LSTM model with the sentimental analysis of tweets to predict the outcome of the game before start of the match. The CRISP – DM methodology is followed in constructing this framework. We have observed that the teams which are playing at home ground creates a higher chance of winning the match. Past game statistics had a very important role to play in drawing conclusions about the game. Along with primary statistics attributes, we have developed few of new features, which has helped us to improve the accuracy of the model. We performed preprocessing on the dataset before training the model in order to eliminate the multicollinearity within the model. LSTM model fetched us an accuracy of 70 % which was better than other machine learning algorithms.

The next additional part on which we have worked is the sentimental analysis of tweets which are gathered from Twitter. Overall, we used to collect a tweet of 10000 with respect to different hashtags related to playing clubs before 7 days of week. The sentimental analysis on tweets represented the amateur opinion and classified the tweets of tone that is positive or negative tone as weather.

Finally, we ensembled both the approaches by using the method of Weighted Averaging in the ratio of 70:30. It is concluded that the ensembled model got more accurate results compared to individual approaches. We may further expand this work in the future by considering some other parameters such as Player performances, Player injuries, Other competitions influences etc. and also by more rigorously optimizing feature set.

References

[1] Rodrigues, Fátima, and Ângelo Pinto. "Prediction of football match results with Machine Learning." Procedia Computer Science 204 (2022): 463-470.

[2] Choi, Bing Shen, Lee Kien Foo, and Sook-Ling Chua. "Predicting Football Match Outcomes with Machine Learning Approaches." MENDEL. Vol. 29. No. 2. 2023.I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[3] Sherstinsky, Alex. "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network." *Physica D: Nonlinear Phenomena* 404 (2020): 132306.

[4] Joseph, Anito, Norman E. Fenton, and Martin Neil. "Predicting football results using Bayesian nets and other machine learning techniques." *Knowledge-Based Systems* 19.7 (2006): 544-553.

[5] Igiri, Chinwe Peace, and Enoch Okechukwu Nwachukwu. "An improved prediction system for football a match result." *IOSR journal of Engineering* 4.12 (2014): 12-20.

[6] Kampakis, Stylianos, and Andreas Adamides. "Using Twitter to predict football outcomes." *arXiv preprint arXiv:1411.1243* (2014).

[7] Schumaker, Robert P., et al. "Prediction from regional angst–a study of NFL sentiment in Twitter using technical stock market charting." *Decision Support Systems* 98 (2017): 80-88.

[8] Bunker, R. P., and F. Thabtah. "A machine learning framework for sport result prediction. Applied Computing and Informatics, 15 (1), 27–33." *URL: http://www. sciencedirect. com/science/article/pii S* (2019).

[9] Bosch, Pablo. "Predicting the winner of NFL-games using Machine and Deep Learning." *Vrije universiteit, Amsterdam* (2018).

[10] Goyal, Udgam. *Leveraging machine learning to predict playcalling tendencies in the NFL*. Diss. Massachusetts Institute of Technology, 2020.

[11] Ceron, Andrea, Luigi Curini, and Stefano M. Iacus. "Using sentiment analysis to monitor electoral campaigns: Method matters—evidence from the United States and Italy." *Social Science Computer Review* 33.1 (2015): 3-20.

[12] Kaimakamis, Christos. "Sports Analytics Algorithm for NBA Champion Prediction." (2021).

[13] Schröer, Christoph, Felix Kruse, and Jorge Marx Gómez. "A systematic literature review on applying CRISP-DM process model." *Procedia Computer Science* 181 (2021): 526-534.

[14] Shimaoka, André Massahiro, Renato Cordeiro Ferreira, and Alfredo Goldman. "The evolution of CRISP-DM for Data Science: Methods, Processes and Frameworks." *SBC Reviews on Computer Science* 4.1 (2024): 28-43.

[15] Kalnins, Arturs, and Kendall Praitis Hill. "The VIF score. What is it good for? Absolutely nothing." *Organizational research methods* (2023): 10944281231216381.

[16] dos Santos Canova, Luciana, et al. "An improved successive projections algorithm version to variable selection in multiple linear regression." *Analytica Chimica Acta* 1274 (2023): 341560.

[17] Dataset: https://www.football-data.co.uk/englandm.php

[18] Modi, Astha, et al. "Sentiment analysis of Twitter feeds using flask environment: A superior application of data analysis." *Annals of Data Science* 11.1 (2024): 159-180.

[19] Elbagir, Shihab, and Jing Yang. "Sentiment analysis on Twitter with Python's natural language toolkit and VADER sentiment analyzer." *IAENG Transactions on Engineering Sciences: Special Issue for the International Association of Engineers Conferences 2019*. 2020.

[20] Zahoor, Sheresh, and Rajesh Rohilla. "Twitter sentiment analysis using machine learning algorithms: a case study." *2020 International Conference on Advances in Computing, Communication & Materials (ICACCM)*. IEEE, 2020.

[21] Chen, Ray, and Marius Lazer. "Sentiment analysis of twitter feeds for the prediction of stock market movement." *stanford edu Retrieved January* 25 (2013): 2013.

[22] Wagh, Rasika, and Payal Punde. "Survey on sentiment analysis using twitter dataset." *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. IEEE, 2018.